Designing fast quantum gates using optimal control with a reinforcement-learning ansatz

Bijita Sarma[1,*] and Michael J. Hartmann[1,2]

[1] *Department of Physics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen 91058, Germany*
[2] *Max Planck Institute for the Science of Light, Erlangen 91058, Germany*

(Received 10 January 2024; revised 22 November 2024; accepted 25 November 2024; published 6 January 2025)

Fast quantum gates are crucial not only for the contemporary era of noisy intermediate-scale quantum devices but also for the prospective development of practical fault-tolerant quantum computing. Leakage errors, which arise from data qubits jumping beyond the confines of the computational subspace, are the main challenges in realizing nonadiabatically driven, fast gates. In this work, we propose and illustrate the usefulness of reinforcement learning (RL) to generate fast two-qubit gates in practical multilevel superconducting qubits. In particular, we show that the RL controller offers great effectiveness in finding piecewise constant gate-pulse sequences that act on two transmon data qubits coupled by a tunable coupler to generate a controlled-Z (CZ) gate with a gate time of 10 ns and an error rate of approximately 4×10^{-3}. Using a gradient-based method to solve the same optimization problem often does not achieve high fidelity for such fast gates. However, we show that using the gate pulses discovered by RL as an ansatz for the gradient-based controller can substantially enhance fidelity compared to using RL alone. While for a 10-ns pulse, this improvement is marginal, the combined RL + gradient approach decreases the gate errors below 10^{-4} for a gate of length 20 ns.

DOI: 10.1103/PhysRevApplied.23.014015

I. INTRODUCTION

As we are inching closer to building practical quantum computers, the need for developing fast quantum gates has become increasingly relevant [1]. This is critical in the present noisy intermediate-scale quantum (NISQ) era, allowing the reliable execution of quantum algorithms despite the intrinsic noise and fragility of qubits and facilitating fault tolerance through the effective implementation of error-correcting gates in large quantum systems [2–8]. Fault tolerance implies that, as long as the error rates of the physical qubits remain below a certain threshold, quantum computing systems, together with quantum error correction (QEC) and logical gate operations, can be utilized for useful computation [9–12]. One of the crucial requirements for efficient QEC is the realization of fast and efficient quantum gates [13].

As a hardware platform for quantum computing, superconducting circuits have shown remarkable developments and are considered promising for the construction of large-scale quantum devices. However, designing fast quantum gates with high fidelity remains a major challenge. Since superconducting qubits are in fact multilevel systems, the most significant challenge in achieving fast high-fidelity two-qubit gates is avoiding leakage outside the computational subspace during gate execution [14–17]. These leakage errors are extremely difficult to minimize, and methods to prevent them restrict the amplitude of control pulses, consequently extending the gate duration. Another major issue for the accurate functioning of large-scale superconducting systems is qubit crosstalk caused by residual ZZ interactions, leading to undesired disruptions in two-qubit gate operations. A method of minimizing crosstalk involves enhancing the hardware architecture, for instance, by employing qubits with different anharmonicities to generate a crosstalk cancelation effect through quantum interference [18–20]. The more experimentally practical approach is to position the qubits in a highly dispersive regime with significant detunings. However, in these configurations, transitioning from a state of large detuning to an operating frequency zone with lesser detuning results in slower gate operations. In this work, we present a technique to implement a rapid two-qubit gate by starting from a condition with negligible residual coupling. Despite steep ramps, the optimization process, which leverages reinforcement learning (RL) combined with optimal control, facilitates the realization of an accelerated two-qubit controlled-Z (CZ) gate.

When dealing with global optimization problems of complex, nonconvex, and nonlinear systems, machine learning (ML) in combination with deep learning has recently been shown to be extremely successful and is

*Contact author: bijita.sarma@fau.de

considered highly versatile for a wide range of tasks [21–23]. RL is a type of ML that is particularly suited for learning to control sequential decision-making problems [24–26]. In the last couple of years, RL has been utilized to find control protocols for some interesting quantum problems [27,28]. It was first demonstrated for the optimization of quantum phases [29] and QEC [30], and more recently we have seen its applications in other areas, in particular, in quantum state engineering [31], quantum pulse and gate design [32,33], quantum feedback control [34–36], etc. RL controls have also been used in real laboratory experiments recently with quantum systems, demonstrating their potential for challenging decisions and their adaptability to control such systems in real time [37,38].

Quantum gate preparation tasks have traditionally been addressed using various optimal control methods, such as gradient-ascent pulse engineering (GRAPE), which, as its name implies, employs gradient-based optimization to minimize a loss function, typically the gate infidelity [39,40], or using Krotov's method by iteratively updating control fields [41]. Such methods often depend on the system dynamics being differentiable, thereby necessitating an accurate understanding of the quantum system model. In contrast, RL algorithms are commonly employed as a model-free method that only needs the output data as the observation to be used to map into control sequences. Furthermore, it is highly adaptable to variations in system parameters, which helps to mitigate model bias. This adaptability is particularly useful if the RL agent is initially trained on a simulator and adjusted based on insights obtained from experimental data to refine the control sequences. With recent advances in high-speed electronic components such as field programmable gate array (FPGA), RL can now be fully applied to experiments using real-time data [37,38]. While gradient-based methods are typically more straightforward than RL and can achieve the desired accuracy with considerably less effort, they heavily rely on the initial parameter values of the gradient-based optimizers, especially for complex tasks like quantum gate preparation discussed in this work. We show that by combining these two approaches, viz. RL and optimal control, it is possible to take advantage of the strengths of both by first optimizing pulses with RL and then refining them further using optimal control techniques.

II. MODEL AND METHODS

We consider a tunable coupler superconducting circuit setup (see Fig. 1) to design an ultrafast two-qubit CZ gate. Besides being an entangling gate that can be used to generate a universal gate set, the CZ gate is a core operation in QEC with surface codes [10,13]. For engineering high-performance, large-scale quantum processors, tunable superconducting circuits have gained prominence,

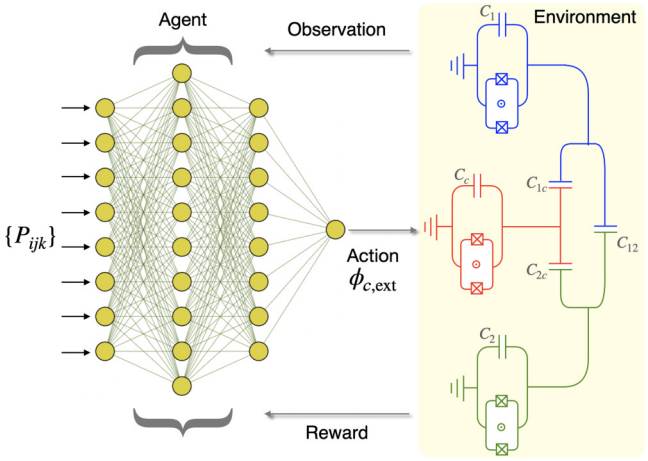

FIG. 1. Schematic diagram showing the RL-controlled gate implementation in a three-qubit tunable coupler circuit. The circuit (right), which constitutes the RL environment, is made of three transmon superconducting qubits, each modeled as a multilevel anharmonic oscillator, with capacitive nearest-neighbor and next-nearest-neighbor couplings. The RL agent is fed with observations $\{P_{ijk}\}$, i.e., the computational and leakage-space populations. Based on these observations, the agent exerts actions to control the coupler frequency $\omega_c(t)$, and receives a reward or penalty in terms of the gate infidelity $\mathcal{I} = 1 - \mathcal{F}$, where $\mathcal{F}$ is the gate fidelity at the end of the sequence.

particularly due to their recently recognized capability for on-demand ON-OFF switching of couplings between qubit pairs via frequency modulation through external fluxes [13,19,33,42,43]. This flexibility allows for precise control over the interactions within the system, making it a valuable resource for implementing high-fidelity gates.

The RL-based optimization protocol is depicted in Fig. 1, where the problem of two-qubit gate design with a tunable coupler superconducting framework is embedded in the RL workflow. The RL agent (shown on the left) is essentially an artificial neural-network model that is responsible for deciding the control sequences (called the actions, $\vec{a}$) by optimizing the weights, $\vec{\theta}$ of the model. These optimizations are directed through the scalar signal of rewards, $\mathcal{R}$ received by the RL agent from the RL environment given it observes some partial information of the system after the application of the control at every step of iteration. These are called the observations, $\vec{s}$ of the RL, and the set of rules it learns by optimizing the parameters $\vec{\theta}$ is called the policy, $\pi(\vec{a}|\vec{s})$ of the RL agent, where $\pi(\vec{a}|\vec{s})$ represents a conditional probability distribution of $\vec{a}$ given $\vec{s}$. In this case, the RL environment is formed by the circuit shown on the right of Fig. 1, comprising two data qubits and a coupler qubit, all of which are modeled as transmon qubits. Explicitly, the observation of the RL agent consists of the computational as well as the leakage-state populations in the eigenstates of the three qubits at the idle points of the gate, i.e., $\vec{s} = \{P_{ijk}\}$, where $\{i, j, k\}$ denotes the qubit

1, coupler, and qubit 2, respectively. The actions of the RL agent are choices of the tunable coupler frequency, ω_c, which are realized by the external flux, $\phi_{c,\text{ext}}$ applied to the coupler, therefore $\vec{a} = \{\omega_c\} \leftarrow \phi_{c,\text{ext}}$. The reward, $\mathcal{R}$ is considered as a function of the process infidelity of the CZ gate defined by $\mathcal{R} = -\log_{10}(1-\mathcal{F})$, where $\mathcal{F}$ is the gate fidelity at the end of the sequence.

The learning process can be divided into iterations called episodes, each with a total duration of τ, which is further segmented into sequences with a duration of $t' = \tau/n$, where n is the number of control steps in the episode. The episode time τ is equivalent to the gate time in our case. If we consider a control problem with N_a control parameters over n control steps in each episode, the complexity of the problem scales exponentially with n as $\prod_{i=1}^{N_a} n_i^n$, where n_i is the number of choices for the ith control parameter, considering a discrete control problem. For the problem under study $N_a = 1$, corresponding to the control parameter ω_c, for which the complexity of the problem scales as n_1^n, where n_1 is the number of choices of the control parameter ω_c. Instead of discrete controls, we consider continuous approximations to stepwise constant shapes of ω_c. This choice of control pulses is motivated by the fact that they are typical pulses generated by arbitrary wave-form generators in current experimental setups.

Despite the fact that we have a single control parameter for the RL agent to learn, this problem turned out to be a formidable task for the RL to learn and we have found that a sophisticated RL algorithm developed in the last few years needs to be employed. Effectively, we used the recently proposed soft-actor-critic (SAC) algorithm for optimization of the RL policy [44]. The SAC algorithm is an actor-critic RL algorithm based on the concept of entropy regularization. The policy π is trained to maximize a trade-off between expected return and entropy. This trade-off determines the balance between exploration and exploitation. The algorithm provides a bonus reward at each time step proportional to the entropy of the policy. This makes the RL policy to spawn actions as randomly as possible due to the inherent stochasticity of the policy, encouraging the agent towards more exploration, prevention of premature convergence to suboptimal solutions, and accelerated learning. The optimal policy π^* is defined as the policy that maximizes the expected return while also maximizing entropy, given by

$$\pi^* = \arg\max_{\pi} \mathop{\mathbb{E}}_{\tau\sim\pi} \sum_{t=0}^{\infty} \gamma^t [\mathcal{R}(s_t, a_t, s_{t+1}) + \alpha\mathcal{H}(\pi(\cdot|s_t))], \tag{1}$$

where $\mathbb{E}_{\tau\sim\pi}$ denotes the expectation value over trajectories τ generated by following the policy π. γ^t is the discount factor raised to the power of t, where γ is a parameter between 0 and 1, representing how much the agent values future rewards relative to immediate rewards. $\mathcal{R}(s_t, a_t, s_{t+1})$ represents the immediate reward obtained when taking action a_t in state s_t and transitioning to state s_{t+1}. The $\alpha\mathcal{H}(\pi(\cdot|s_t))$ term involves the entropy $\mathcal{H}$ of the policy π at state s_t, weighted by a hyperparameter α, that regulates stochasticity of the policy, and encourages randomness in the actions taken by the RL agent [see Appendix A for details].

The tunable coupler circuit depicted in Fig. 1 is described by the Hamiltonian (considering $\hbar = 1$ hereinafter),

$$\begin{aligned} H = &\sum_{i=1,c,2} \left(\omega_i b_i^\dagger b_i + \frac{\alpha_i}{2} b_i^\dagger b_i^\dagger b_i b_i\right) \\ &+ g_{12}\left(b_1 + b_1^\dagger\right)\left(b_2 + b_2^\dagger\right) \\ &+ \sum_{i=1,2} g_{ic}\left(b_i + b_i^\dagger\right)\left(b_c + b_c^\dagger\right), \end{aligned} \tag{2}$$

where $b_i(b_i^\dagger)$ with $i = 1, c, 2$ describe the bosonic annihilation (creation) operators for the transmon qubits ($i = 1, 2$) and the coupler, where the qubits are considered as weakly anharmonic oscillators possessing multiple energy levels, with anharmonicities given by α_i. The qubits interact with one another through capacitive coupling, where the next-nearest-neighbor coupling capacitance, C_{12}, is smaller than the nearest-neighbor coupling capacitances, $\{C_{1c},\ C_{2c}\}$, which in turn are small compared to the transmon qubit capacitances $\{C_1,\ C_c,\ C_2\}$. This leads to the fact that the circuit analysis can be treated perturbatively. The circuit is initialized at the idle point with the eigenstates $|ijk\rangle$ (where $\{i, j, k\}$ label the qubit 1, coupler, and qubit 2, respectively) before application of the gate and is returned back to after the completion of the gate. At the idle point, these instantaneous eigenstates have maximum overlap with the bare states of the circuit, with a slight hybridization between the data qubits due to residual coupling [19]. The experimentally relevant computational subspace for the two-qubit gates between the data qubits consists of the eigenstates at the idle points $|ik\rangle = |00\rangle, |01\rangle, |10\rangle$, and $|11\rangle$, where the coupler is considered to be always in the ground state. All other eigenstates constitute the leakage subspace.

We aim to design the CZ gate utilizing the transverse qubit-qubit coupling to induce a phase of $e^{i\pi}$ in the computational state $|101\rangle$ by using nonadiabatic transitions to the noncomputational eigenstate $|002\rangle$ and back. At the idle point, we consider the qubits to be in the highly dispersive regime where the detuning between the coupler and the qubits is large compared to their mutual couplings, so that both the transverse and longitudinal couplings between the qubits are negligible. We bias the qubits and coupler at the frequencies of $\omega_1/2\pi = 4.2$ GHz, $\omega_2/2\pi = 5.2$ GHz, and $\omega_c/2\pi = 6.38$ GHz. This results in negligible transverse and longitudinal ZZ couplings [see Appendix B for

details]. The other parameters are, $\alpha_1/2\pi = -200$ MHz, $\alpha_c/2\pi = -100$ MHz, $\alpha_2/2\pi = -200$ MHz, $g_{1c}/2\pi = 85$ MHz, $g_{2c}/2\pi = 85$ MHz, and $g_{12}/2\pi = 7$ MHz.

Starting at this dispersive coupling limit, a CZ gate can be obtained by first tuning the frequency of qubit 1 to $\omega_1 = \omega_2 + \alpha_2$, so that the levels $|101\rangle$ and $|002\rangle$ become resonant, and then tuning the coupler frequency close to the data qubit frequencies. Holding the coupler frequency at this point for the time of one oscillation between these two states, the target unitary $U_{CZ} = \mathrm{diag}(1, 1, 1, -1)$ can be achieved up to single-qubit phases, which can be virtually compensated for. However, the gate time for such a Rabi-oscillation-based operation is long as it is given by $t_{\mathrm{gate}} = \pi/\zeta_{XX}$, where ζ_{XX} is the transverse coupling rate, as it needs to satisfy the adiabaticity condition $\int \zeta_{XX}(t)dt \gg 1$ [45]. As discussed previously, the RL agent's task is to tune the coupler frequency $\omega_c(t)$ throughout the duration of the gate within the given interval, $\omega_c(t)/2\pi = [4.2, 6.38]$ GHz to maximize the fidelity of the gate at the end of the gate. Given the extensive exploration abilities of neural networks, RL is expected to outperform standard gradient-based methods that struggle with shallow minima in intricate control tasks. Nevertheless, for challenging problems like the one discussed here, RL might also perform suboptimally, oscillating between minima with nearly identical returns, and frequently abandoning such solutions in pursuit of further exploration, ultimately deviating from the original strategy. We show that using the optimum discovered by an RL stage as the initial ansatz for an optimal control gradient stage, allows us to find a better solution than the one discovered by RL in terms of achievable fidelity within short gate times.

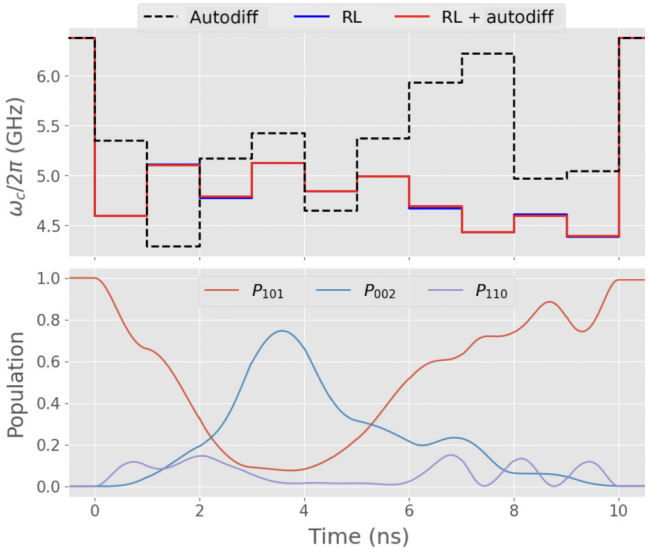

FIG. 2. (Top panel) Piecewise constant gate-control sequence for the coupler qubit frequency obtained from RL (blue), automatic differentiation (in black dashed), and combined RL and autodifferentiation (red). Before implementation of the gate, the qubits are started at the idle point with negligible crosstalk, and also brought to the same resting configuration after the gate. (Bottom panel) The corresponding population of the significantly occupied computational and leakage-space states, P_{ijk}, are shown for the RL + autodiff-derived pulses throughout the gate evolution (see the main text for further details).

III. RESULTS AND DISCUSSIONS

The results are shown in Fig. 2, which illustrates the findings related to the identification of a 10-ns-long CZ gate. The upper panel of the figure presents the control pulses, with the results derived from the gradient with automatic differentiation (autodiff), RL, and combined RL + autodiff methods. The gradient-descent technique seeks to fine tune the values of $\omega_c(t)$ over the period $t = [0, 10]$ ns to reduce the gate's infidelity at the end, while RL utilizes an iterative optimization process, modifying actions based on the feedback obtained after each action $\omega_c[t]$ in order to increase the gate fidelity using a neural-network policy developed through extensive trial-and-error learning. We find that the simple gradient-based method with an intuitive initial ansatz for $\omega_c(t)$ could not find a good solution for the 10-ns CZ gate with a fidelity of about 90% (averaged over runs with different initial ansatz), while the one discovered by RL yields a fidelity of 99.60%. The fidelity is marginally enhanced to 99.63% by employing the RL-derived solution as an ansatz for the gradient-based method. This distinction becomes considerably more prominent for gates with extended gate durations, as depicted in Fig. 3, which illustrates the infidelity as a function of gate times ranging from 10 to 20 ns discovered through the three approaches. RL + autodiff optimization shows an overall improved strategy resulting in consistently better fidelities reaching a value higher than 99.99% for a 20-ns gate time. Although gate sequences shorter than this achieve high speed by partially populating the leakage channels at the expense of lower fidelity [46], the gate with a duration of the order of 20 ns falls below the surface-code Pauli error threshold of approximately 0.01, as well as leakage error threshold of approximately 10^{-4} [47] (see Appendix F for an analysis of leakage population with respect to gate time). Nonetheless, if the leakage is above threshold, there are several specific techniques, such as leakage reduction units or teleportation, that can be applied to get the leakage under control and bring the qubits back into the computational space before regular error-correction cycles start [47–53]. Also, since the decoherence time for state-of-the-art transmons is of the order of $\tau' \sim 60$ μs, this leads to error rates characterized by $\varepsilon_{\tau'} = 1 - \exp(-t_{\mathrm{gate}}/\tau') \approx 3 \times 10^{-4}$ for $t_{\mathrm{gate}} = 20$ ns [42]. Therefore, executing such a rapid gate operation is feasible well before substantial information loss occurs due to decoherence.

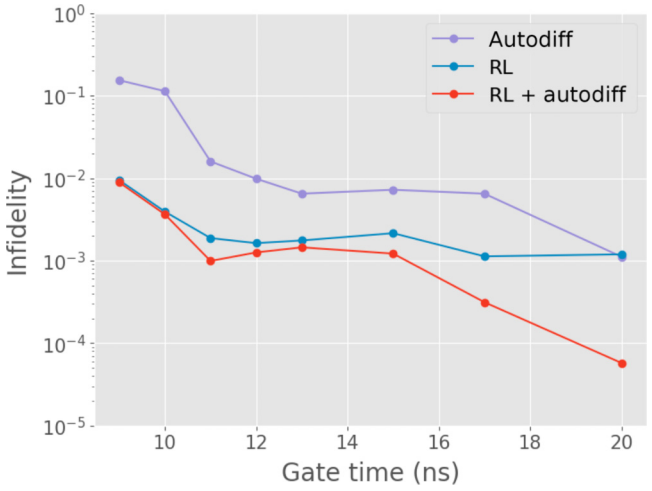

FIG. 3. Comparison of the performance of the three optimization methods, viz. autodiff (purple), RL (blue), and RL combined with autodiff (red), in relation to variations in gate time.

The bottom panel of Fig. 2 shows the populations of the $|101\rangle$, $|002\rangle$, and $|110\rangle$ states during the application of the gate pulses found above. The RL agent finds optimal conditions for the pulse variation between the states $|101\rangle$ and $|002\rangle$ to acquire the desired phase. Throughout the gate operation, the leakage into the coupler is greatly minimized due to the alternating positive and negative shifts in qubit frequency, with a further reduction observed by the end of the gate. The extent of leakage is also reduced substantially for longer-duration gates, as shown in Appendix C.

It is important to note that, in practice, creating pulses with abrupt transitions is challenging and should be substituted with gradual rise and fall transitions. Appendix E examines the effects of finite rise and fall times for each step using a specific example as well as variations in step size, showing that the inclusion of such effects does not drastically reduce gate fidelity.

Finally, we discuss the prospects of the proposed method for experimental implementations. In this context, we investigate the applicability of the pulse shapes to devices where the transition frequencies of the qubits do not exactly match the parameters of the assumed model, cf. Eq. (2). In Fig. 4, we show the robustness of the optimization against frequency fluctuations at the idle point. We consider the optimized pulse that the RL + autodiff method found for a set of initial qubit frequencies (shown with black lines), at which the training was done. Then we apply the trained and optimized pulses to a circuit with variations in the idle point qubit frequency, shown along the x axes. During the gate, the coupler frequency is applied according to the optimized result. The plots show that the fidelity is maintained up to variations of approximately 10% for ω_c and ω_1, while it is more sensitive to ω_2. Although this demonstrates the gate's response to parameter uncertainties, the trained RL model can also be retrained to adjust for such parameter variations. For example, the gate fidelity achieved with the optimized pulse for the coupler's initial frequency of 6.38 MHz remains robust even when the initial frequency drifts from 6.38 to 6.10 MHz, which corresponds to a transverse coupling rate of approximately 0.5 MHz as shown in Fig. 6 (a similar transverse coupling strength of 0.3 MHz was considered in Ref. [54]). This shows that while our optimization scheme is applicable to such bias points while maintaining high fidelity, it can further be improved by retraining at those bias points. In practical scenarios where the gate pulses developed are utilized in actual experiments, it is anticipated that significant model bias will be observed. These biases can be corrected to accommodate minor parameter deviations encountered in practical applications. However, it is useful to recognize that if the system parameter drift is extreme, the resultant pulses may vary substantially. For example, it has been noted that when the RL agent limits the control parameter space to $\omega_c(t)/2\pi = [5.2, 6.38]$ GHz instead of $\omega_c(t)/2\pi = [4.2, 6.38]$ GHz (discussed above), the resulting gate pulses exhibit distinct

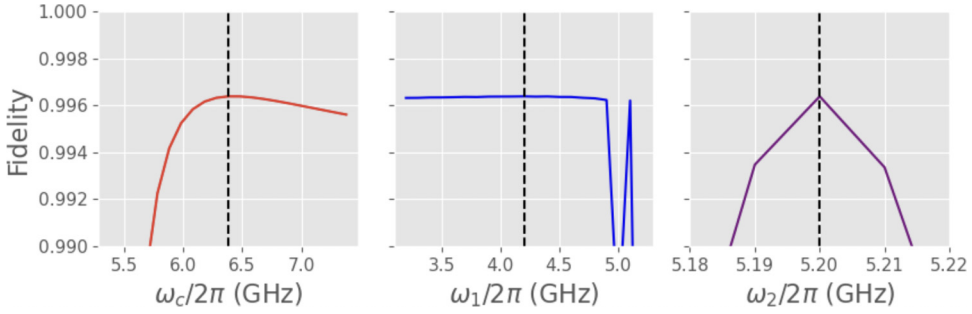

FIG. 4. The variation in gate fidelity for a 10-ns gate in response to model fluctuations, in terms of the variation in the *idling* (initial) frequencies for the coupler qubit (red), qubit 1 (blue), and qubit 2 (purple), respectively, when employing the pulses obtained in Fig. 2 (RL + autodiff) and while the other parameters are kept unchanged. The black dotted vertical lines indicate the idling parameter for which the optimized pulses were found.

characteristics with markedly different leakage behavior [see Appendix D].

IV. CONCLUSION

In summary, we illustrate an RL-driven methodology for the design of rapid and nonintuitive pulse sequences to execute a two-qubit CZ gate within a tunable coupler architecture. We show that, grounded solely on penalty or reward considerations, the artificial agent can assimilate effective strategies and unveil realistic parameter configurations for the modulation of coupler piecewise constant flux pulses. We also combine RL with gradient-based optimal control that results in an improved optimizer culminating in ultrafast CZ gate with high fidelity with error much below the surface-code error threshold while maintaining a very short gate time. This is an improvement in gate duration of approximately 6 and 3 times, respectively, compared to the CZ gate implementations with a tunable coupler in Ref. [42] with a 60-ns-long CZ gate and 34 ns in the surface-code implementation with Google's Sycamore processor [13]. Incorporating experimental data directly into the training process of the RL agent would obviate the necessity for precise simulation models, thereby facilitating the agent's capacity to adjust to device impairments and temporal parameter fluctuations. In this regard, employing the pulse generated by the RL on the simulator as the starting point and subsequently applying a gradient-based method with experimental data would represent a favorable approach.

ACKNOWLEDGMENTS

This work received support from the German Federal Ministry of Education and Research via the funding program Quantum Technologies—from basic research to the market under Contract No. 13N16182 MUNIQC-SC. It is also part of the Munich Quantum Valley, which is supported by the Bavarian state government, with funds from the Hightech Agenda Bayern Plus. B.S. thanks Lukas Heunisch for useful discussions.

APPENDIX A: BASIC THEORY OF REINFORCEMENT LEARNING

1. Reinforcement-learning workflow

RL has emerged as a key domain within ML, notably marked by groundbreaking advancements from DeepMind and Google [22,23]. The unique feature of RL lies in its adaptive and iterative learning process, in which the RL agent adjusts its strategy based on real-time consequences within the dynamic environment. In contrast to supervised and unsupervised ML methods, which rely on (precollected) labeled and unlabeled datasets for training, RL derives its knowledge through continuous interaction with the environment and evaluation via a reward function.

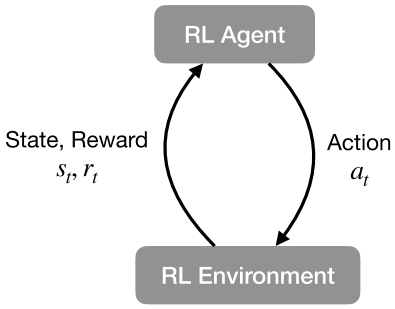

FIG. 5. The basic workflow of RL.

This makes RL particularly well suited for tasks involving control or decision making compared to supervised and unsupervised learning techniques.

The basic RL workflow is shown in Fig. 5. It comprises an RL agent that determines what action to take at a specific timestep t in the RL environment (which encodes the system to be controlled), resulting in a modification of the state of the environment. The agent then observes this altered state (which often includes only partial information about the environment), denoted s_t. The result or impact of the action, whether beneficial or detrimental, is evaluated by the reward r_t calculated based on the system's behavior following the action. The RL agent is generally configured as a neural-network model, trained through extensive agent-environment interactions to achieve optimal control. The cumulative sum of rewards accumulated over an episode (time length for which we want to learn the control sequences) is used as a metric to adjust the model's weights and biases. The combination of optimized parameters encapsulates the complex rules required to adapt actions based on observed state changes. This set of rules, known as the policy, serves as the decision-making algorithm for the RL agent.

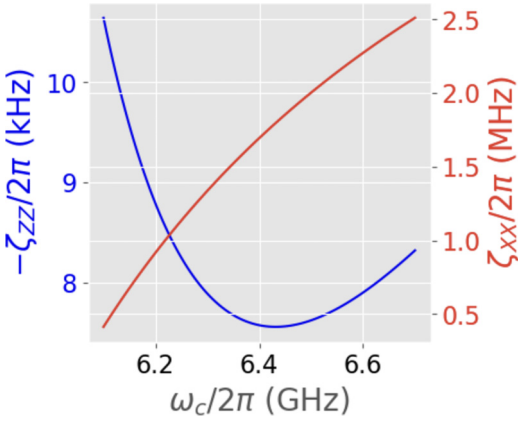

FIG. 6. The effective static transverse (ζ_{XX}) and longitudinal (ζ_{ZZ}) couplings. The circuit parameters are stated in the main text.

2. General theory

The policy can be classified into two forms: deterministic and stochastic. In the deterministic approach, the action of the RL agent is precisely determined by the policy parameters, denoted as θ, given the state s_t at time t. This deterministic relationship is expressed as $a_t = \mu_\theta(s_t)$ (often used for policy evaluation). Alternatively, the policy can be stochastic, where actions are sampled from a probability distribution conditioned on s_t: $a_t \sim \pi_\theta(\cdot|s_t)$. The RL agent's objective is to iteratively refine and optimize these policy parameters (θ) to maximize the cumulative discounted rewards along a trajectory $\tau = (s_0, a_0, s_1, a_2, \ldots)$,

$$R(\tau) = \sum_{t=0}^{T} \gamma^t r_t, \tag{A1}$$

where the discount factor $\gamma \in (0, 1)$ modulates the significance of future rewards. The optimization process entails maximizing the expected return over discounted rewards, denoted by $J(\pi) = \mathbb{E}[R(\tau)]$, with the ultimate goal of achieving the optimal policy $\pi^\star = \text{argmax}[J(\pi)]$.

A related concept is the value function, which is used to predict the expected cumulative discounted future reward and to assess the effectiveness of a given state s or the state-action pair (s, a) in generating a higher net return. The state value $V_\pi(s) = \mathbb{E}[R_t|s_t = s]$ represents the anticipated return when adhering to the policy π from state s. In contrast, the action value $Q_\pi(s, a) = \mathbb{E}[R_t|s_t = s, a_t = a]$ signifies the expected return when action a is executed in state s followed by policy π. The Bellman equations [24] govern the value functions, which can be solved self-consistently. For instance, the action-value function is expressed as

$$Q_\pi(s, a) = \mathbb{E}\left[r(s, a) + \gamma \cdot \max_{a'} Q_\theta(s', a')\right]. \tag{A2}$$

Optimizing the policy encompasses various techniques categorized into three main groups: (a) policy-gradient-based, (b) value-based, and (c) actor-critic-based methods. Value-based approaches, such as Q learning, aim to maximize value functions by solving the Bellman equations. In contrast, policy-gradient methods employ gradient-descent algorithms to optimize policy parameters, given by

$$\nabla_\theta J(\pi_\theta) = \mathbb{E} \sum_{t=0}^{T} [\nabla_\theta \log \pi_\theta(a_t|s_t) R_t], \tag{A3}$$

where $\mathbb{E}$ represents the expectation value over the trajectory τ. This basic approach can be improved by introducing a baseline function, $b(s_t)$, to reduce the variance in gradient estimation, forming the basis of advanced RL actor-critic algorithms. The objective (loss) function for policy-gradient methods to optimize is given by

$$L^{\text{PG}}(\theta) = \hat{\mathbb{E}}_t [\log \pi_\theta(a_t|s_t) A_t], \tag{A4}$$

where π_θ is a stochastic policy, and $\hat{A}_t = Q(s_t, a_t) - V(s_t)$ is an estimator of the advantage function at timestep t, considering R_t as an estimate of $Q(a_t, s_t)$. An actor-critic algorithm simultaneously learns a policy and a state-value function, using the value function for bootstrapping to reduce variance and accelerate learning [24]. The critic updates action-value function parameters, and the actor adjusts policy parameters following the critic's guidance.

3. Soft actor-critic algorithm

The soft actor-critic (SAC) algorithm employed in the current study, is a recently developed actor-critic approach in the realm of RL. What sets SAC apart from other actor-critic methods is its distinctive feature of optimizing the policy in an entropy-regularized manner, rendering it inherently stochastic. In SAC, the policy is trained to strike a balance between expected return and entropy. The intentional introduction of entropy serves the purpose of promoting increased exploration and preventing premature convergence of the policy.

At each time step in RL regularly adjusted for entropy, the agent is compensated according to the entropy of the policy distribution,

$$\pi^* = \arg\max_\pi \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t, s_{t+1}) + \alpha \mathcal{H}\left(\pi(\cdot|s_t)\right)\right)\right], \tag{A5}$$

where $\mathcal{H}(P) = \mathop{\mathbb{E}}_{x\sim P}[-\log P(x)]$ is the entropy of the probability distribution P, and $\alpha > 0$ is the trade-off coefficient. This equation describes the optimal policy π^* that maximizes the expected return over time. The return is the sum of the rewards $R(s_t, a_t, s_{t+1})$ at each time step, discounted by a factor γ^t to prioritize immediate rewards over future ones. Additionally, this formulation includes an entropy term $\alpha\mathcal{H}\left(\pi(\cdot|s_t)\right)$, where entropy $\mathcal{H}$ encourages exploration by maximizing the uncertainty (randomness) of the policy. The trade-off coefficient α balances the significance of this entropy against the immediate reward.

The value functions in this setting, V^π and Q^π, are modified accordingly.

$$V^\pi(s) = \mathop{\mathbb{E}}_{\tau\sim\pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t, s_{t+1}) + \alpha \mathcal{H}\left(\pi(\cdot|s_t)\right)\right)|s_0 = s\right], \tag{A6}$$

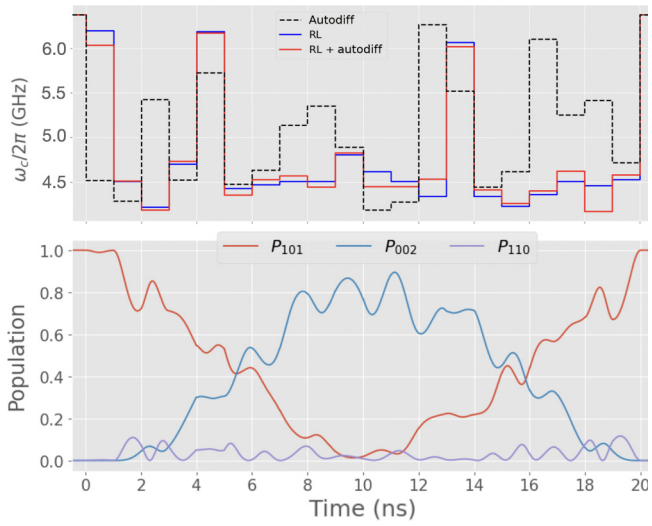

FIG. 7. (Top panel) Control pulses in terms of the coupler frequency derived from the three optimization processes viz. autodiff (black dashed), RL (blue solid), and autodiff on RL-optimized pulses (red solid), for a CZ gate of 20-ns gate time. The parameter search space is restricted to $2\pi \times (4.2 - 6.38)$ GHz. Gate time steps are of 1-ns duration. The circuit parameters are given in the main text. (Bottom panel) The population dynamics in the states $|101\rangle$ (red), $|002\rangle$ (blue), and $|110\rangle$ (purple) during the gate, optimized with RL + autodiff method. Other leakage states are least occupied.

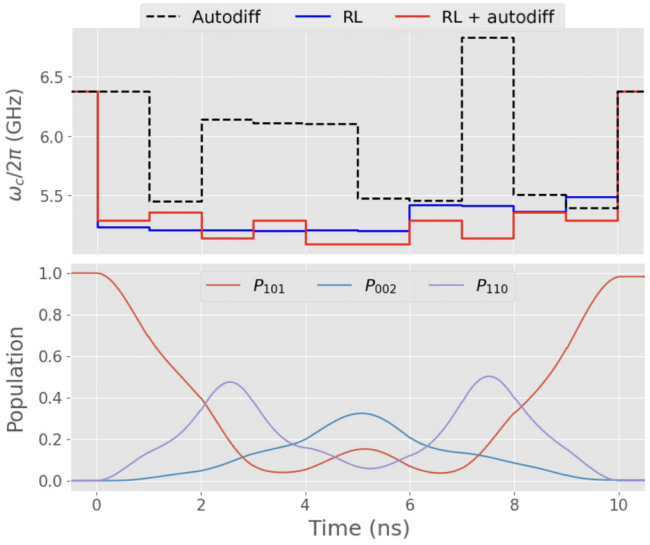

FIG. 8. The control pulses (top panel) and the population dynamics (bottom panel) for a shorter pulse of 10-ns gate time for the limited parameter search space of $\omega_c/2\pi = [5.2, 6.38]$ GHz. The other circuit parameters are considered as specified in the main text. The control pulses are shown for the three optimization methods, viz. autodiff (black dashed), RL (blue solid), and autodiff combined with RL-optimized pulses (red solid). The populations of the states $|101\rangle$ (red), $|002\rangle$ (blue), and $|110\rangle$ (purple) are shown.

$$Q^{\pi}(s,a) = \mathop{\mathbb{E}}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t,s_{t+1}) + \alpha\sum_{t=1}^{\infty}\gamma^{t}\mathcal{H}\left(\pi(\cdot|s_t)\right)|s_0=s,a_0=a\right]. \tag{A7}$$

The state-value function $V^{\pi}(s)$ takes into account the rewards over time and the uncertainty in the policy (via the entropy term). This function essentially tells us how good it is to be in a particular state s under policy π. The action value function $Q^{\pi}(s,a)$ indicates how good it is to take a particular action a in state s. The connection between V^{π} and Q^{π} is shown by the following equation:

$$V^{\pi}(s) = \mathbb{E}_{a\sim\pi}\left[Q^{\pi}(s,a) + \alpha\mathcal{H}\left(\pi(\cdot s)\right)\right]. \tag{A8}$$

It states that the value of being in state s (under policy π) can be computed by taking the expected value of the Q function over all possible actions a the policy might choose, plus the entropy of the policy at that state. This reflects the idea that $V^{\pi}(s)$ considers all potential actions weighted by their probability under π.

The Bellman equation is a recursive equation used to estimate $Q^{\pi}(s,a)$. It states that the Q value for taking action a in state s is approximately equal to the immediate reward r plus the discounted value of the next state's Q value. The subtraction of $\alpha\log\pi(\tilde{a}'|s')$ from this next state Q value accounts for the entropy of the policy, promoting exploration. The Bellman equation for Q^{π} is estimated by

$$Q^{\pi}(s,a) \approx r + \gamma\left(Q^{\pi}(s',\tilde{a}') - \alpha\log\pi(\tilde{a}'|s')\right). \tag{A9}$$

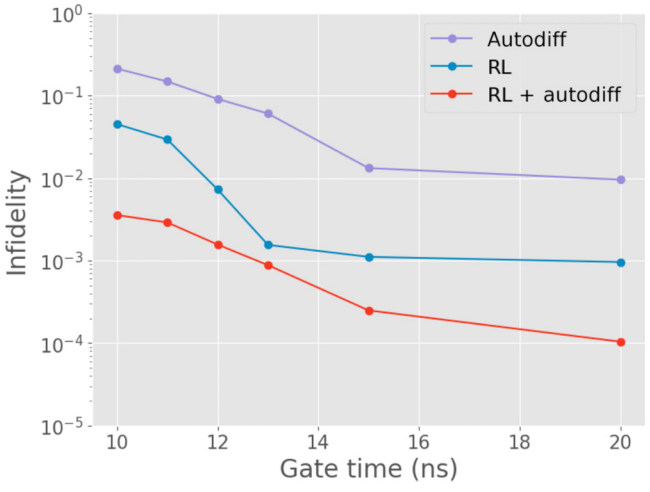

FIG. 9. Comparison of the three optimization methods, viz. autodiff (black dashed), RL (blue solid), and autodiff on RL-optimized pulses (red solid), shown in terms of gate infidelity with respect to variation in gate time for the limited parameter search space of $\omega_c/2\pi = [5.2,\ 6.38]$ GHz.

In this formulation, the expected values over the next states r and the states, s' are taken from the replay buffer, while the subsequent actions $\tilde{a}'$ are sampled from the policy. The SAC algorithm undergoes the concurrent learning of a policy π_θ and two Q functions Q_{ϕ_1} and Q_{ϕ_2}. The loss functions for each Q function, enforcing the minimum Q value between the two Q approximators, are articulated as

$$L(\phi_i, \mathcal{D}) = \underset{(s,a,r,s',d)\sim\mathcal{D}}{\mathbb{E}} \left[\left(Q_{\phi_i}(s,a) - y(r,s',d) \right)^2 \right], \tag{A10}$$

$$y(r,s',d) = r + \gamma \left(\min_{j=1,2} Q_{\phi_{\text{targ},j}}(s',\tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right), \quad \tilde{a}' \sim \pi_\theta(\cdots'). \tag{A11}$$

The loss function $L(\phi_i, \mathcal{D})$ is used to train the Q function Q_{ϕ_i}. The goal is to minimize the difference (squared error) between the Q function's current estimate and the target value $y(r,s',d)$. The target value $y(r,s',d)$ represents the estimated return used to update the Q function. It combines the immediate reward r with the discounted future return from the next state s', considering the minimum value of the two Q functions Q_{ϕ_1} and Q_{ϕ_2}. The entropy term $-\alpha \log \pi_\theta(\tilde{a}'|s')$ encourages exploration by reducing the target value, thus penalizing certainty in the action selection. The data (s,a,r,s',d) are sampled from a replay buffer $\mathcal{D}$, which stores past experiences to break the correlation between consecutive samples during training. Our SAC agent adheres to the implementation as described in Refs. [44,55,56].

APPENDIX B: BIAS POINTS

We aim to design the CZ gate utilizing the transverse qubit-qubit coupling to induce a phase of $e^{i\pi}$ in the computational state $|101\rangle$ by using nonadiabatic transitions to the noncomputational eigenstate $|002\rangle$ and back. Applying a Schrieffer-Wolff transformation, the effective coupler-induced transverse interaction between the data qubits in the dispersive regime is found as, $\zeta_{XX} = g_{12} + g_{1c}g_{2c}\left(\Delta_{1c}^{-1} + \Delta_{2c}^{-1}\right)/2$, where $\Delta_{ij} := \omega_i - \omega_j$ denotes the qubit detunings. One can see that with proper choice of circuit parameters, the effective two-qubit transverse coupling can be tuned. We bias our two-qubit gate circuit in a parameter regime where this effective coupling is small, which is essential for efficient parking of the data qubits (in few MHz range). There is also a residual longitudinal (ZZ) interaction because of dispersive shifts in qubit energies caused by the hybridization of the qubit wave functions, given by $\zeta_{ZZ} = E_{101} - E_{100} - E_{001} + E_{000}$, where E_{jkl} is the energy eigenvalue of the state $|jkl\rangle$. Such residual coupling works as crosstalk in the proper implementation of the gate and should be negligible at the parking point of the circuit. We bias the qubits at the frequencies of $\omega_1/2\pi = 4.2$ GHz, $\omega_2/2\pi = 5.2$ GHz and $\omega_c/2\pi = 6.38$ GHz. This results in a negligible ZZ crosstalk ($\zeta_{ZZ}/2\pi = -7.59$ kHz) (as shown in Fig. 6). As demonstrated in the main text, the optimization remains robust to variations in the coupler frequency at the bias point that corresponds to reduced transverse coupling in the sub-MHz range as well [54].

APPENDIX C: CZ GATE WITH GATE TIME OF 20 ns

In Fig. 7 we show a comparison of the gate optimization for the case of the 20-ns gate found by the three different methods viz. (i) autodiff, (ii) RL and (iii) RL + autodiff, and for this we specify the limits of the control parameter, i.e., the coupler frequency to be in the range of 6.38 ns (the idle point at the dispersive limit) to 4.2 ns (near the lower qubit frequency). The resultant population dynamics due to the RL + autodiff method is shown in the bottom panel. The leakage population is seen to be significantly low throughout the gate operation and further lowered at the end of the gate.

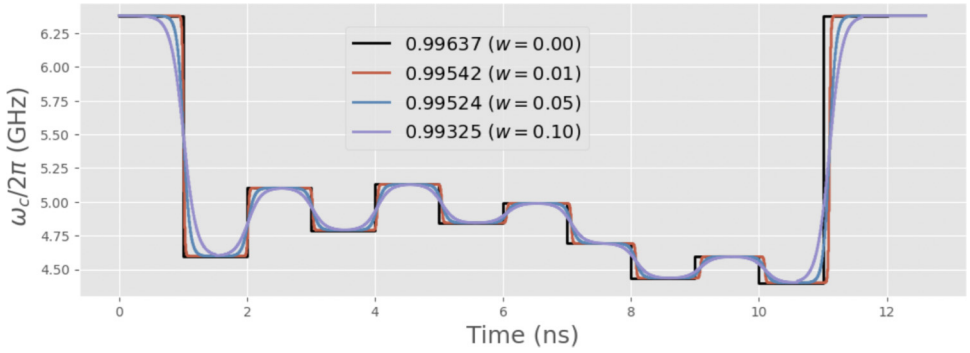

FIG. 10. The influence of finite rise and fall times on the gate fidelity is illustrated as a function of w in Eq. (E1) for a 10-ns gate, where higher values of w indicate smoother edges. The inset shows the fidelities corresponding to the values of w. Other parameters are the same as considered in the main text.

APPENDIX D: EFFECT OF LIMITED CONTROL PARAMETER SEARCH SPACE

It has been observed that the control pulses and the consequent gate dynamics vary notably based on the extent of the parameter space accessible to the RL agent. The pulses and dynamics presented in the main text as well as the ones in Fig. 7 utilize the parameter space where $\omega_c/2\pi \in [4.2, 6.38]$ GHz. In contrast, Fig. 8 shows the pulses and population dynamics for the 10-ns CZ gate found using the autodiff, RL and RL + autodiff methods with a restricted parameter search space of $\omega_c/2\pi \in [5.2, 6.38]$ GHz, where the coupler frequency ranges from the dispersive bias point to the higher data qubit frequency. Compared to this, Fig. 2 in the main text demonstrates a significant reduction in the leakage population in the coupler during the gate operation while the controls are allowed to explore larger parameter space. In Fig. 9, we present a comparison of infidelities resulting from the optimizations using the discussed methods, in relation to the change in gate time for the limited parameter space $\omega_c/2\pi \in [5.2, 6.38]$ GHZ; this can be contrasted with Fig. 3 from the main text.

APPENDIX E: EFFECT OF FINITE RISE AND FALL TIMES AND STEP SIZE

We show the effect of finite rise and fall times for the piecewise constant pulse sequences for a 10-ns gate in Fig. 10. It can be modeled as a function of width w, which controls the scaling of the exponent in the logistic function:

$$f(t) = \left[1 + \exp\left(-\frac{t - t_0}{w}\right)\right]^{-1}. \tag{E1}$$

When w is large, the term $(t - t0)/w$ changes slowly, so the exponential term varies slowly, leading to a gradual transition. When w is small, the term $(t - t0)/w$ changes rapidly, making the exponential term change quickly, leading to a rapid transition. This is a good approximation of practical filtering scenarios similar to also discussed in detail in Ref. [57]. The corresponding gate fidelity shows that for a fractional rise and fall time, the fidelity is not significantly altered. While Ref. [57] reported a quadratic growth of error in rise and fall times, our scheme appears to show a better scaling.

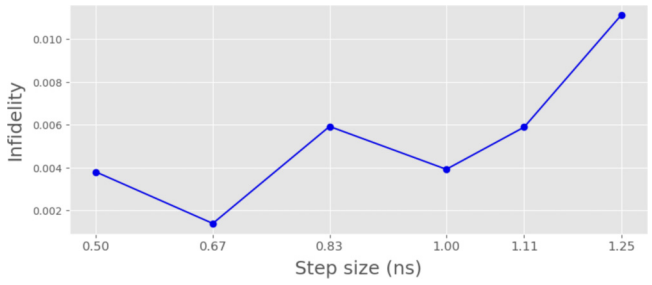

FIG. 11. The effect of step size of the control pulse on the gate fidelity for the 10-ns gate. Circuit parameters are the same as considered in the main text.

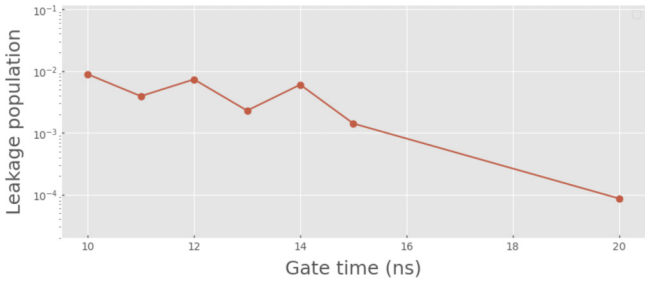

FIG. 12. Leakage population outside the computational subspace for the RL + autodiff optimized pulses plotted with respect to gate duration. Circuit parameters are the same as stated in the main text.

We also show the effect of the step size of the piecewise controls on the optimization with the RL + autodiff method for a 10-ns-long pulse in Fig. 11, which shows an overall increasing behavior of gate error.

APPENDIX F: LEAKAGE POPULATION

Figure 12 shows the leakage population outside the computational subspace at the end of the gate as a function of gate time for the RL + autodiff optimization, demonstrating an overall decreasing trend with a value of 8×10^{-5} at 20 ns.

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, England, UK, 2010).

[2] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[4] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[5] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Superconducting qubits: Current state of play, Annu. Rev. Condens. Matter Phys. **11**, 369 (2020).

[6] S. Rosenblum, P. Reinhold, M. Mirrahimi, L. Jiang, L. Frunzio, and R. J. Schoelkopf, Fault-tolerant detection of a quantum error, Science **361**, 266 (2018).

[7] L. DiCarlo, J. M. Chow, J. M. Gambetta, L. S. Bishop, B. R. Johnson, D. I. Schuster, J. Majer, A. Blais, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, Demonstration

of two-qubit algorithms with a superconducting quantum processor, Nature **460**, 240 (2009).

[8] C. K. Andersen, A. Remm, S. Lazar, S. Krinner, N. Lacroix, G. J. Norris, M. Gabureac, C. Eichler, and A. Wallraff, Repeated quantum error detection in a surface code, Nat. Phys. **16**, 875 (2020).

[9] Y. Ma, Y. Xu, X. Mu, W. Cai, L. Hu, W. Wang, X. Pan, H. Wang, Y. P. Song, C.-L. Zou, and L. Sun, Error-transparent operations on a logical qubit protected by quantum error correction, Nat. Phys. **16**, 827 (2020).

[10] S. Krinner, N. Lacroix, A. Remm, A. Di Paolo, E. Genois, C. Leroux, C. Hellings, S. Lazar, F. Swiadek, J. Herrmann, *et al.*, Realizing repeated quantum error correction in a distance-three surface code, Nature **605**, 669 (2022).

[11] E. Knill and R. Laflamme, Theory of quantum error-correcting codes, Phys. Rev. A **55**, 900 (1997).

[12] A. L. Grimsmo and S. Puri, Quantum error correction with the Gottesman-Kitaev-Preskill code, PRX Quantum **2**, 020101 (2021).

[13] R. Acharya, I. Aleiner, R. Allen, T. I. Andersen, M. Ansmann, F. Arute, K. Arya, A. Asfaw, J. Atalaya, R. Babbush, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, Nature **614**, 676 (2023).

[14] Z. Chen, J. Kelly, C. Quintana, R. Barends, B. Campbell, Y. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Lucero, *et al.*, Measuring and suppressing quantum state leakage in a superconducting qubit, Phys. Rev. Lett. **116**, 020501 (2016).

[15] C. C. Bultink, T. E. O'Brien, R. Vollmer, N. Muthusubramanian, M. W. Beekman, M. A. Rol, X. Fu, B. Tarasinski, V. Ostroukh, B. Varbanov, A. Bruno, and L. DiCarlo, Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements, Sci. Adv. **6**, eaay3050 (2020).

[16] R. Fazio, G. M. Palma, and J. Siewert, Fidelity and leakage of Josephson qubits, Phys. Rev. Lett. **83**, 5385 (1999).

[17] K. S. Chou, T. Shemma, H. McCarrick, T.-C. Chien, J. D. Teoh, P. Winkel, A. Anderson, J. Chen, J. Curtis, S. J. de Graaf, *et al.*, Demonstrating a superconducting dual-rail cavity qubit with erasure-detected logical measurements, ArXiv:2307.03169.

[18] F. Yan, P. Krantz, Y. Sung, M. Kjaergaard, D. L. Campbell, T. P. Orlando, S. Gustavsson, and W. D. Oliver, Tunable coupling scheme for implementing high-fidelity two-qubit gates, Phys. Rev. Appl. **10**, 054062 (2018).

[19] L. Heunisch, C. Eichler, and M. J. Hartmann, Tunable coupler to fully decouple and maximally localize superconducting qubits, Phys. Rev. Appl. **20**, 064037 (2023).

[20] P. Mundada, G. Zhang, T. Hazard, and A. Houck, Suppression of qubit crosstalk in a tunable coupling superconducting circuit, Phys. Rev. Appl. **12**, 054023 (2019).

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016).

[22] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, Mastering the game of Go with deep neural networks and tree search, Nature **529**, 484 (2016).

[23] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, Mastering the game of Go without human knowledge, Nature **550**, 354 (2017).

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

[25] V. Krotov, *Global Methods in Optimal Control Theory* (CRC Press, Boca Raton, FL, USA, 1995).

[26] T. Caneva, T. Calarco, and S. Montangero, Chopped random-basis quantum optimization, Phys. Rev. A **84**, 022326 (2011).

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

[28] V. Gebhart, R. Santagati, A. A. Gentile, E. M. Gauger, D. Craig, N. Ares, L. Banchi, F. Marquardt, L. Pezzè, and C. Bonato, Learning quantum systems, Nat. Rev. Phys. **5**, 141 (2023).

[29] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, Phys. Rev. X **8**, 031086 (2018).

[30] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Reinforcement learning with neural networks for quantum feedback, Phys. Rev. X **8**, 031084 (2018).

[31] R. Porotti, D. Tamascelli, M. Restelli, and E. Prati, Coherent transport of quantum states by deep reinforcement learning, Commun. Phys. **2**, 1 (2019).

[32] B. Sarma, S. Borah, A. Kani, and J. Twamley, Accelerated motional cooling with deep reinforcement learning, Phys. Rev. Res. **4**, L042038 (2022).

[33] L. Ding, M. Hays, Y. Sung, B. Kannan, J. An, A. Di Paolo, A. H. Karamlou, T. M. Hazard, K. Azar, D. K. Kim, *et al.*, High-fidelity, frequency-flexible two-qubit fluxonium gates with a transmon coupler, Phys. Rev. X **13**, 031035 (2023).

[34] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, Measurement-based feedback quantum control with deep reinforcement learning for a double-well nonlinear potential, Phys. Rev. Lett. **127**, 190403 (2021).

[35] Z. T. Wang, Y. Ashida, and M. Ueda, Deep reinforcement learning control of quantum cartpoles, Phys. Rev. Lett. **125**, 100401 (2020).

[36] S. Borah and B. Sarma, No-collapse accurate quantum feedback control via conditional state tomography, Phys. Rev. Lett. **131**, 210803 (2023).

[37] V. V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret, Real-time quantum error correction beyond break-even, Nature **616**, 50 (2023).

[38] K. Reuer, J. Landgraf, T. Fösel, J. O'Sullivan, L. Beltrán, A. Akin, G. J. Norris, A. Remm, M. Kerschbaum, J.-C. Besse, *et al.*, Realizing a deep reinforcement learning agent for real-time quantum feedback, Nat. Commun. **14**, 1 (2023).

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

[40] G. Jäger, D. M. Reich, M. H. Goerz, C. P. Koch, and U. Hohenester, Optimal quantum control of Bose-Einstein condensates in magnetic microtraps: Comparison of gradient-ascent-pulse-engineering and Krotov optimization schemes, Phys. Rev. A **90**, 033628 (2014).

[41] V. F. Krotov, in *Advances in Nonlinear Dynamics and Control: A Report from Russia*, SpringerLink (Birkhäuser Boston, 1993), p. 74.

[42] Y. Sung, L. Ding, J. Braumüller, A. Vepsäläinen, B. Kannan, M. Kjaergaard, A. Greene, G. O. Samach, C. McNally, D. Kim, *et al.*, Realization of high-fidelity CZ and *ZZ*-free iSWAP gates with a tunable coupler, Phys. Rev. X **11**, 021058 (2021).

[43] J. Stehlik, D. M. Zajac, D. L. Underwood, T. Phung, J. Blair, S. Carnevale, D. Klaus, G. A. Keefe, A. Carniol, M. Kumph, M. Steffen, and O. E. Dial, Tunable coupling architecture for fixed-frequency transmon superconducting qubits, Phys. Rev. Lett. **127**, 080505 (2021).

[44] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, ArXiv:1801.01290.

[45] G. T. Genov, S. Rochester, M. Auzinsh, F. Jelezko, and D. Budker, Robust two-state swap by stimulated Raman adiabatic passage, J. Phys. B: At. Mol. Opt. Phys. **56**, 054001 (2023).

[46] B. Khani, J. M. Gambetta, F. Motzoi, and F. K. Wilhelm, Optimal generation of Fock states in a weakly nonlinear oscillator, Phys. Scr. **2009**, 014021 (2009).

[47] M. Suchara, A. W. Cross, and J. M. Gambetta, Leakage suppression in the Toric code, Quantum Inf. Comput. **15**, 997 (2015).

[48] P. Aliferis and B. M. Terhal, Fault-tolerant quantum computation for local leakage faults, Quantum Inf. Comput. **7**, 139 (2007).

[49] M. Werninghaus, D. J. Egger, F. Roy, S. Machnes, F. K. Wilhelm, and S. Filipp, Leakage reduction in fast superconducting qubit gates via optimal control, npj Quantum Inf. **7**, 1 (2021).

[50] M. McEwen, D. Kafri, Z. Chen, J. Atalaya, K. J. Satzinger, C. Quintana, P. V. Klimov, D. Sank, C. Gidney, A. G. Fowler, *et al.*, Removing leakage-induced correlated errors in superconducting quantum error correction, Nat. Commun. **12**, 1 (2021).

[51] K. C. Miao, M. McEwen, J. Atalaya, D. Kafri, L. P. Pryadko, A. Bengtsson, A. Opremcak, K. J. Satzinger, Z. Chen, P. V. Klimov, *et al.*, Overcoming leakage in quantum error correction, Nat. Phys. **19**, 1780 (2023).

[52] E. Hyyppä, A. Vepsäläinen, M. Papič, C. F. Chan, S. Inel, A. Landra, W. Liu, J. Luus, F. Marxer, C. Ockeloen-Korppi, *et al.*, Reducing leakage of single-qubit gates for superconducting quantum processors using analytical control pulse envelopes, PRX Quantum **5**, 030353 (2024).

[53] L. Chen, S. P. Fors, Z. Yan, A. Ali, T. Abad, A. Osman, E. Moschandreou, B. Lienhard, S. Kosen, H.-X. Li, *et al.*, Fast unconditional reset and leakage reduction in fixed-frequency transmon qubits, ArXiv:2409.16748.

[54] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, *et al.*, Superconducting quantum circuits at the surface code threshold for fault tolerance, Nature **508**, 500 (2014).

[55] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Stable-baselines3: Reliable reinforcement learning implementations, J. Mach. Learn. Res. **22**, 1 (2021).

[56] J. Achiam, Spinning Up in Deep Reinforcement Learning (2018), https://github.com/openai/spinningup.

[57] S. Oh, Errors due to finite rise and fall times of pulses in superconducting charge qubits, Phys. Rev. B **65**, 144526 (2002).