Quantum noise modeling through Reinforcement Learning

Simone Bordoni,[1,2,3,*] Andrea Papaluca,[4,2,*] Piergiorgio Buttarini,[1,2,*]
Alejandro Sopena,[5,2,*] Stefano Giagu,[1,3] and Stefano Carrazza[6,7,8,2]

[1] *Dep. of Physics, La Sapienza University of Rome, Piazzale Aldo Moro 2, Rome, 00185, Italy*
[2] *Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, UAE.*
[3] *Sez. di Roma, INFN, Piazzale Aldo Moro 2, Rome, 00185, Italy*
[4] *School of Computing, The Australian National University, Canberra, ACT, Australia*
[5] *Instituto de Física Teórica, UAM-CSIC, Universidad Autónoma de Madrid, Cantoblanco, Madrid, Spain*
[6] *CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland.*
[7] *TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano, Italy*
[8] *INFN, Sezione di Milano, I-20133 Milan, Italy.*

In the current era of quantum computing, robust and efficient tools are essential to bridge the gap between simulations and quantum hardware execution. In this work, we introduce a machine learning approach to characterize the noise impacting a quantum chip and emulate it during simulations. Our algorithm leverages reinforcement learning, offering increased flexibility in reproducing various noise models compared to conventional techniques such as randomized benchmarking or heuristic noise models. The effectiveness of the RL agent has been validated through simulations and testing on real superconducting qubits. Additionally, we provide practical use-case examples for the study of renowned quantum algorithms.

Keywords: Machine Learning; Reinforcement Learning; Quantum Computing; Quantum Noise

I. INTRODUCTION

One important unsolved technological question regards the practical applicability of Noisy Intermediate Scale Quantum (NISQ) [1] computers. Despite the expectation that quantum computers will outperform classical computers in certain computational tasks [2–4], the usability and reliability of NISQ devices are hindered by a large error rate. These errors arise from gate infidelities, unwanted environmental interactions, thermal relaxation, measurement errors, and cross-talk [5–9]. Currently, there are techniques to mitigate these errors [10–14]. However, it has been demonstrated that any quantum circuit for which error mitigation is efficient must be classically simulable [15]. Therefore, it is widely regarded that quantum advantage will only be achieved with future generations of fault-tolerant quantum devices [16–21].

Despite the imperfect results obtained on NISQ devices, numerous algorithms have been developed and implemented on this hardware. Machine learning-inspired models, in particular, have shown promising results in recent years [22–29]. Testing and developing new algorithms in the NISQ era can be challenging due to limited access to quantum chips. Furthermore, the currently accessible quantum computers in the cloud are highly demanded, resulting in lengthy waiting queues [30]. In this context, emulating the noise of these devices emerges as an alternative to accelerate circuit testing.

This work aims to develop a model capable of learning hardware-specific noise for use in circuit simulations. This goal is further motivated by the limited availability of noise modeling or noise prediction techniques [31–34]. Our approach employs Reinforcement Learning (RL) [35–37] to train an agent to add noise channels that replicate the noise pattern of a specific quantum chip. This method minimizes heuristic assumptions about the noise model, thereby enhancing the adaptability and generalization properties. The algorithm has been validated on both simulations and real quantum devices hosted at the Quantum Research Center [38] of the Technology Innovation Institute (TII) in Abu Dhabi, demonstrating its ability to accurately predict different noise patterns.

The Qibo framework [39, 40] was used for the realization of this work. It provides `Qibo` [39, 40] as a high-level language API for crafting quantum computing algorithms, `Qibolab` [41, 42] as a tool for quantum control, and `Qibocal` [43] for conducting quantum characterization and calibration routines.

All the code developed for this work is available on GitHub [1]. It is possible to use the code to reproduce the results as well as testing the algorithm under different noise conditions. The code is intended to be easily customizable in order to allow users to define their own RL agents for specific applications.

The outline is as follows. Section II introduces the basic concepts necessary to understand the proposed algorithm, noise in quantum circuits and reinforcement learning. Section III provides a detailed description of the reinforcement learning algorithm, focusing on its training and use for noise prediction. Section IV presents the results obtained with the proposed algorithm on both simulations and real quantum devices, and compares its per-

* These authors contributed equally to this work.

[1] https://github.com/qiboteam/rl-noisemodel

formance with other noise predictors. Section V demonstrates examples of the algorithm's use-cases for famous quantum algorithms.

II. BACKGROUND

This section provides an overview of the essential concepts required to understand the proposed noise simulation algorithm. Specifically, we briefly discuss noise in quantum circuits and introduce reinforcement learning.

A. Noise in the quantum circuit model

Current and near-term quantum computers lack fault tolerance, and their usefulness is limited by the presence of noise and errors. To understand these limitations, we delve into the quantum circuit model, a framework introduced by Deutsch in 1989 [44]. This model presumes a quantum register composed of near-ideal qubits. Quantum computations are carried out by altering this register through a combination of qubit measurements and unitary operations drawn from a universal set of gates, known as native gate set. Within this context, we distinguish four types of errors. The first type, state preparation errors, arise during the initialization of the quantum register. These errors result from the need for rapid reset protocols, which involve coupling qubits to other elements such as cavities and measurement devices, leading to deviations from the ideal zero state [45]. The second type of error is due to the limited precision of measurements, which requires their representation as POVMs with inherent uncertainties, thereby preventing unlimited repeated measurements on the same qubit. The third type, qubit decoherence, refers to the loss of quantum superposition due to environmental factors. It is typically modeled by relaxation and dephasing times, T_1 and T_2, for each qubit. However, this model can be insufficient when decoherence is correlated, such as when environmental fluctuations or unwanted interactions between the qubits affect multiple qubits similarly. Lastly, gate imperfections arise from intrinsic errors and control limitations in implementing single-qubit and two-qubit unitaries. These imperfections are measured by the gate fidelity.

It is common to make certain assumptions about the inherent errors. One typical assumption is to distinguish the ideal gates from the errors, considering them as separate processes. It is also often assumed that these errors break down into spatially uncorrelated errors that affect the idle qubits, and an average error that is operation-independent, impacting only the qubits that are being manipulated. Within this context, it is useful to classify the various types of noise introduced above into two groups: coherent and incoherent. Coherent noise, typically resulting from minor miscalibrations in control parameters, tends to produce similar errors in successive executions of a quantum circuit, thereby introducing a systematic bias in the output. It is important to note that coherent noise preserves the purity of the state and, once identified, can be corrected [46–48]. On the other hand, incoherent noise can be viewed as processes that cause entanglement between the system and its environment.

The errors and imperfect operations are typically represented using the formalism of quantum channels, i.e., trace-preserving completely positive maps of density matrices into density matrices. In the ideal operation of a quantum computer, a positive map can be just a unitary transformation $\varepsilon(\rho) = U\rho U^\dagger$, where U describes a quantum gate. Coherent noise is unitary, and we model it using single qubit rotation gates (R_x, R_y, R_z). For instance, a coherent error could introduce an unintended deviation ϵ in the x direction during the application of $R_j(\theta)$, altering the state $\rho = R_j(\theta)\rho_0 R_j(\theta)^\dagger$ to

$$\mathrm{Coh}_x(\rho) = R_x(\epsilon)\rho R_x(\epsilon)^\dagger\,. \tag{1}$$

We model incoherent noise as local depolarization and amplitude damping. Depolarization noise tends to drive the state towards the maximally mixed state,

$$\mathrm{Dep}(\rho) = (1-\lambda)\rho + \lambda\frac{\mathrm{Tr}(\rho)}{2^n}\mathbb{I}\,, \tag{2}$$

where n is the number of qubits and λ is the depolarization parameter. Amplitude damping noise models the loss of energy from the qubit to the environment, and it is described by the map

$$\mathrm{Damp}(\rho) = A_1\rho A_1^\dagger + A_2\rho A_2^\dagger\,, \tag{3}$$

where $A_1 = |0\rangle\langle 0| + \sqrt{1-\gamma}\,|1\rangle\langle 1|$ and $A_2 = \sqrt{\gamma}\,|0\rangle\langle 1|$, and γ represents the decay probability from $|1\rangle$ to $|0\rangle$. While it is possible to add more complexity to the modeling of incoherent noise, we aim for an effective description with few parameters to prevent overfitting. Preliminary tests showed no significant performance enhancement when adding other channels.

Multiple noise channels can be combined in order to construct complex noise models [33, 49]. The parameters of these channels can be inferred either from calibration results or directly from the execution of quantum circuits, as demonstrated in this work.

For a simplified noise modeling, we can employ a technique known as Randomized Benchmarking (RB) [50–52]. RB enables us to efficiently estimate the average error magnitude across a set of quantum gates, with resource requirements scaling polynomially with the number of qubits. A RB protocol employs sequences of varying lengths of randomly chosen n-qubit Clifford gates, where the ideal composite operation of the sequence is the identity. To produce such a sequence of depth $m+1$, each of the first m gates in the sequence are picked randomly from the Clifford group. The last gate is then uniquely determined as the Clifford element which inverts the composition of the previous m gates. In the

presence of noise, the actual sequence of Clifford gates does not represent an identity operation. Instead, there is a certain survival probability of measuring the initial state after performing the sequence. By estimating this survival probability across multiple independent random sequences of increasing depth, we can fit an exponential decay of the form

$$P_l = ap^l + b\,. \tag{4}$$

The average error map, therefore, behaves like a completely depolarizing model with an average depolarizing parameter p. This parameter can be used to estimate the average gate fidelity of n-qubit quantum gates. Employing RB as a noise predictor involves extracting the decay parameter and introducing a depolarizing channel with a depolarizing parameter equal to the decay parameter after each gate. We employ this technique to establish a depolarization noise model, serving as a benchmark for our algorithm. Although this model, which projects all noise sources onto the depolarizing channel, may seem unrealistic, it provides a useful baseline for comparison.

B. Reinforcement learning

Reinforcement learning is a powerful Machine Learning (ML) paradigm that trains an agent to make optimal decisions in a dynamic environment. From a mathematical perspective, a reinforcement learning algorithm is described as a Markov Decision Process (MDP) [53]. An MDP is characterized by a tuple (S, A, P, R, γ), where S represents the set of possible environmental states, A denotes the set of actions, $P(s'|s,a)$ is the transition probability of reaching state s' from state s by taking action a, $R(s,a)$ provides the immediate reward of taking action a in state s, and γ is the discount factor that balances the importance of immediate rewards against future rewards. In an MDP, the future state depends solely on the current state and action, regardless of the previous history. Solving MDPs involves finding an optimal policy that maximizes the expected averaged sum of rewards. This policy can be deterministic or stochastic, and it can be represented by a function $\pi(s)$ that returns the action to be taken in state s.

In a reinforcement learning model, the policy $\pi(s)$ is implemented by an artificial Neural Network (NN) [54], which is trained through executing different episodes of agent-environment interaction. At the end of each episode (or batch of episodes), the average episode reward is computed and used to update the weights of the NN using the backpropagation algorithm [55].

Many optimization methods have been developed in recent years to improve reinforcement learning convergence and stability during training [56]. In our work, we have obtained the best results using the Proximal Policy Optimization (PPO) [57]. This optimization method provides increased stability by constraining the policy update to a proximity region. Moreover, PPO can handle a continuous action space, which is necessary to accurately model quantum noise. One drawback is its sensitivity to hyperparameters, as poorly chosen values can impact convergence and overall performance.

We employed the `Stable_Baselines3` library [58] to define and train the algorithm. This library is built on top of `OpenAI Gym` [59], which provides a wide range of customizable environments for reinforcement learning tasks. Gradient optimization was performed with `PyTorch` [60].

III. ALGORITHM IMPLEMENTATION

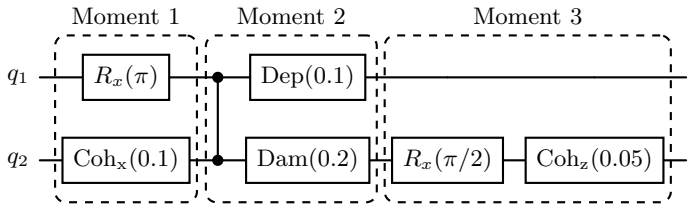
This section provides a detailed explanation of the proposed noise modeling algorithm, with a particular emphasis on the transformation of quantum circuits into input feature vectors suitable for neural network processing, as well as a comprehensive description of the policy, the training process and the datasets.

A. Circuit Representation

To train the RL agent, we need to represent a quantum circuit as an array that can be readily processed by the policy neural network. In the following we refer to this array as the Quantum Circuit Representation (QCR). The QCR has a shape of $[\,qubits, depth, encoding\,]$. The first dimension corresponds to the circuit's qubits, while the second dimension represents the circuit's moments, *i.e.*, a collection of gates that can be executed in parallel. The *encoding* dimension encodes the information regarding gates and noise channels acting on a specific qubit at a specific circuit moment. Its dimension is determined by the total number of native gates and noise channels the model allows for.

In detail, considering a set of n single-qubit native gates $G_{i\,i=1}^{\,n}$, the initial n entries of the *encoding* will contain a *one-hot* encoding of our native set. Namely, in the presence of the gate G_i, all the entries are set to 0 except for the i-th entry that is set to 1. Specifically, in our case, we use $n = 2$ to encode the presence of an R_x or R_z gate. The $n+1$ entry is set to 1 if a two-qubit gate is present and 0 otherwise, with the CZ gate being the native two-qubit gate in this work. For single-qubit circuits, this entry is always zero. The $n+2$ entry encodes the rotation angle of single-qubit gates, normalized to the range $[0, 1]$. The remaining entries detail the parameters of the noise channels in the following sequence: Depolarizing channel, Amplitude damping channel, R_z coherent error, and R_x coherent error. If a noise channel is absent, the corresponding parameter value is set to zero. An example of the QCR for a two-qubit circuit, obtained using this procedure, is illustrated in Figure 1.

To enable the agent to adapt to circuits of varying depths, we further introduce the concept of kernel size k,



		R_z	R_x	CZ	θ	Dep	Dam	$\mathrm{Coh_z}$	$\mathrm{Coh_x}$
Moment 1	q_1	0	1	0	0.5	0	0	0	0
	q_2	0	0	0	0	0	0	0	0.1
Moment 2	q_1	0	0	1	0	0.1	0	0	0
	q_2	0	0	1	0	0	0.2	0	0
Moment 3	q_1	0	0	0	0	0	0	0	0
	q_2	1	0	0	0.25	0	0	0.05	0

FIG. 1. Example of a two-qubit quantum circuit (top) and its vector representation (bottom).

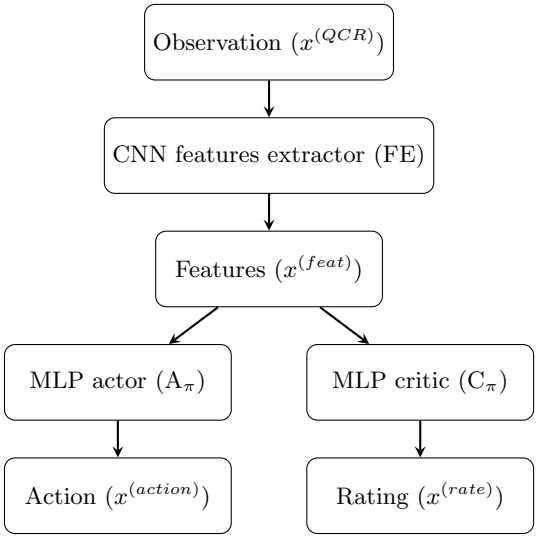


FIG. 2. Schematization of the policy NN for the PPO algorithm.

similar to the kernels used in convolutional neural networks (CNN) [61]. The kernel size establishes a "window" that restricts the number of circuit moments the agent can observe at any given time. For instance, with $k = 3$, the agent only observes the current moment and the immediately preceding and following ones. The window's center starts from the first moment and slides one position at each step until the circuit's end is reached. This approach is based on the heuristic assumption that a gate's noise is most influenced by its temporally proximate gates. At a given moment m, therefore, a window $(m - \frac{k-1}{2}, m + \frac{k-1}{2})$ is extracted from the complete QCR of the circuit, effectively yielding a fixed dimension $[\,qubits,\ k,\ encoding\,]$ tensor that is fed as input to the agent.

B. Policy

Agent's actions can be depicted following the same QCR schema. They are represented as matrices, whose individual rows and columns represent the qubits and the distinct noise channels respectively. Each entry of the action matrix is computed by a forward pass through the policy network, which consists of three main components. The Feature Extractor (FE) takes as input the QCR ($x^{(QCR)}$) and maps it to a high dimensional latent feature space $x^{(feat)}$,

$$x^{(feat)} = \mathrm{FE}(x^{(QCR)})\,.$$

To efficiently capture the correlations between different qubits and moments, we employ a CNN, which should be suited for processing two-dimensional data. The Actor Policy (A_π), is a simple Multi Layer Perceptron (MLP) which takes as input the latent features and computes the actual action the agent is going to take ($x^{(action)}$).

$$x^{(action)} = \mathrm{A}_\pi(x^{(feat)})\,.$$

Moreover, as PPO is an actor-critic policy we have a separated critic policy C_π with the same MLP architecture but independent weights, whose role is to rate the action selected by the actor $x^{(rating)}$. The rating is extracted from the latent features,

$$x^{(rating)} = \mathrm{C}_\pi(x^{(feat)})\,.$$

A schematization of the complete policy NN is reported in Figure 2.

We have performed different tests to determine the optimal architecture and number of parameters for the NNs used in our policy. These hyperparameters slightly change with the number of qubits. Here, we report the best characteristics observed for circuits with one and three qubits. For the feature extractor, we used a single convolutional layer with 16 filters for single-qubit circuits and 32 filters for three-qubit circuits. This convolutional layer is followed by a dense layer with a ReLU activation function. The optimal number of output features for this dense layer is 64 for single qubit circuits and 32 for three qubits circuits. Both the actor and critic policies are implemented as MLPs with a hidden dense layer containing 256 neurons. The total number of trainable parameters in the entire policy NN is on the order of 10^4. We experimented with increasing the number of parameters by adding additional convolutional layers and increasing the number of features in the feature extractor. However, we observed overfitting when the total number of parameters approached the order of 10^5.

As detailed in Section II A, in this work we consider a set of four possible noise channels: depolarizing, amplitude damping, and coherent errors R_x and R_z. This means, that the output of our actor policy A_π is going to be a $(nqubits, 4)$ tensor encoding the predicted parameters of the inserted noise channels at that specific

moment. A value of zero corresponds to no noise channel of that type inserted. For example, in a two-qubit circuit, the action

$$\begin{bmatrix} 0.1 & 0 & 0 & 0.2 \\ 0 & 0.05 & 0.3 & 0 \end{bmatrix}$$

indicates that a depolarizing channel with a depolarizing probability of 0.1 and an R_x coherent error with a rotation angle of 0.2 are added to the first qubit. Simultaneously, an amplitude damping channel with a damping probability of 0.05 and an R_z coherent error with a rotation angle of 0.3 are added to the second qubit.

A sensitive hyperparameter of the algorithm is the maximum allowed value for each noise channel's parameter, P_{max}, which controls in practice the span of the search space. A smaller P_{max} yields a narrower parameter space making convergence faster. However, a too small value might limit the expressive capability of the model, precluding the agent the access to possibly large reward areas of the parameter space. We obtained good results by setting P_{max} as twice the effective depolarizing parameter derived from a RB experiment.

C. Training procedure

The agent is trained to add noise channels to a noiseless quantum circuit in order to reproduce the noise pattern observed when executing the circuit in the presence of some target noise (whether simulated or real when executing on the hardware). This process is outlined in Figure 3.

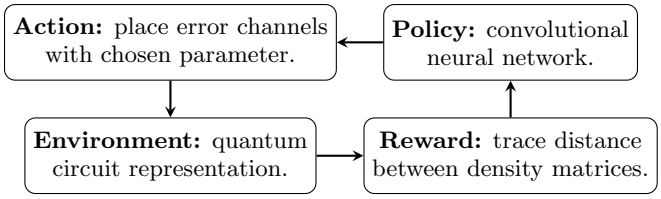


FIG. 3. Training process of the RL algorithm.

Each training episode begins with the agent receiving a randomly selected noiseless quantum circuit from the training set. Then, for each of the circuit's moments, the agent observes the current QCR and takes an action: any combination of the selected set of noise channels, together with their corresponding noise parameters, is inserted in that precise moment, yielding an updated QCR.

The agent receives the reward at end of each episode and the policy's NN parameters are then updated to maximize future rewards. The reward is taken to be a function of the distance between the target Density Matrix (DM) (ρ_{true}) and the DM of the noisy circuit generated by the agent (ρ_{agent}) under some selected metric. We have tested different metrics such as a simple element-wise Mean Squared Error (MSE), density matrix fidelity, and Trace Distance (TD) [62]. Trace distance proved to be the best metric to employ because it is easy to compute and phase invariant. Moreover, it is specifically designed to measure the experimental distinguishability between two quantum states. TD can not be directly used as reward, so we have explored different functional forms. The most effective form has been found to be

$$\mathrm{R}(\rho_{\text{agent}}, \rho_{\text{true}}) = \frac{1}{\alpha\, \mathrm{TD}(\rho_{\text{agent}}, \rho_{\text{true}})^2 + \epsilon}\,, \qquad (5)$$

where ϵ is a small parameter introduced to prevent numerical instabilities and the hyperparameter α can be used to normalize the reward. This equation essentially penalizes high values of the TD with a low reward. The average episode reward, denoted as $\mathcal{R}$, is determined by averaging R over all the training circuits within a batch of episodes. A summary of the training procedure is provided in Algorithm 1. After numerous episodes, the agent is expected to learn the optimal placement of noise channels in a noise-free circuit to reconstruct the final density matrix of the real noisy circuit. Once trained, our algorithm should be capable of generalizing to previously unseen circuits, thereby enabling realistic noisy simulations.

Algorithm 1: Agent training procedure.

```
for episode in n_episodes do
    circuit = random_extraction(training_set);
    for moment in circuit do
        observation =
         agent.make_observation(circuit, moment);
        action = agent.action(observation);
        circuit.add_noise(action);
    end
    generated_dm = extract_density_matrix(circuit);
    reward = compute_reward(generated_dm,
     ground_truth_dm);
    agent.update_policy(reward);
end
```

D. Dataset Generation

The RL algorithm requires training, testing, and evaluation datasets, which consist of ensembles of random quantum circuits and their corresponding final DMs ρ_{true}. These DMs serve as ground truth labels during the training phase of the algorithm (Section III C). In simulations, the ρ_{true} are computed analytically. However, for circuits executed on hardware, they can be obtained using quantum state tomography [63] or more efficient techniques such as classical shadow state reconstruction [64–66].

In this study, we utilize circuits with gates in the native gate set $\{R_x(\pi/2), R_z(\theta), CZ\}$ implemented in the

quantum devices of the Technology Innovation Institute (TII) [38]. We also conducted preliminary tests using the native gates set of IBM quantum hardware [67], which employs the $CNOT$ gate as the two-qubit entangling native gate. Modifying the native gates in our algorithm is a simple process and has not led to any significant changes in performance.

The train set for single qubit circuits is composed of circuits with a fixed number of Clifford gates extracted randomly. Clifford gates are chosen due to, both, their lower simulation cost and their large use in randomized benchmarking and shadow state estimation [52, 65, 66, 68]. In our specific case, since we consider R_x and R_z gates, the only allowed rotation parameters are multiples of $\pi/2$. For three-qubit circuits we have used a more sophisticated training set. Half of the training set is composed of circuits with a fixed number of moments with the gates and parameters extracted randomly. The second half of the training set is composed of Clifford circuits implementing randomly chosen three-qubit Clifford unitaries [69]. These circuits do not have a fixed number of gates or moments. We observed an improvement in the generalization properties of the algorithm when using this mixed training set.

For the performance evaluation we have used two different datasets. The first dataset is composed of non-Clifford circuits with fixed depth. The results obtained with this set demonstrate that the algorithm, even if trained on Clifford circuits, maintains its ability to generalize. The second dataset for performance evaluation, consist of Clifford circuits with varying depths. We have used this dataset to fit the RB noise model and compare it with the RL agent.

For the datasets used in simulations, we defined custom noise models. Specifically, we used two different custom noise models for circuits with one and three qubits. For single-qubit circuits, we applied a depolarizing channel with a depolarizing parameter of $\lambda = 0.02$ after each R_z gate, and an amplitude damping channel with decay parameter $\gamma = 0.03$ after each R_x gate. A coherent $R_x(\theta')$ error with angle $\theta_x = 0.04 \cdot \theta$ is introduced after each $R_x(\theta)$ gate. Similarly, a coherent $R_z(\theta')$ error is added after each $R_z(\theta)$ gate, with $\theta_z = 0.02{\cdot}\theta$. This noise model is not intended to be realistic but to test our algorithm on a gate dependent noise model. For three-qubit circuits, we used a similar noise on rotation gates and added a depolarizing channel with a depolarizing parameter of 0.02 and an amplitude damping channel with a decay probability of 0.03 after each CZ gate. In Section V, we have also used a lower error rate noise model to test the algorithm's ability to generalize to different noise models. A comprehensive summary of the noise parameters used in the simulations is provided in Table I.

Generating the training set is straightforward for simulations where the DMs are computed analytically. However, performing full state tomography to obtain these values on quantum hardware can be time-intensive. For this reason, we conducted preliminary tests on one and three qubit simulated circuits to determine the minimal length of the training set. We trained the algorithm on datasets of varying sizes (from 10 to 10^3), using the noise models introduced in Table I, and assessed the performance on a common evaluation set. We found that for three qubits, it is sufficient to train the algorithm on a dataset with more than 100 circuits to avoid overfitting and achieve optimal performance. For single qubit circuits, a dataset with about 20 circuits is sufficient to obtain nearly optimal performance. This result may vary with the complexity of the noise model; however, it demonstrates that dataset generation is not a bottleneck for quantum chips with a small number of qubits.

Noise	Gates	λ	γ	θ_x	θ_z
1 qubit	$R_x(\theta)$	0	0.03	$0.04{\cdot}\theta$	0
	$R_z(\theta)$	0.02	0	0	$0.02{\cdot}\theta$
3 qubits (high noise)	$R_x(\theta)$	0	0.03	$0.04{\cdot}\theta$	0
	$R_z(\theta)$	0.02	0	0	$0.03{\cdot}\theta$
	CZ	0.02	0.03	0	0
3 qubits (low noise)	$R_x(\theta)$	0	0.01	$0.015{\cdot}\theta$	0
	$R_z(\theta)$	0.015	0	0	$0.02{\cdot}\theta$
	CZ	0.015	0.01	0	0

TABLE I. Noise models used in simulation to train and evaluate the RL algorithm in different conditions. The columns show the noise channels parameters: λ for the depolarizing channel, γ for the amplitude damping channel and θ_x and θ_z for the coherent error R_x and R_z respectively. The rows indicate the gates subject to the noise channel, a zero noise parameter means that the noise channel is not applied to that gate.

IV. RESULTS

The following sections detail the results obtained from applying the proposed algorithm in both simulations and on quantum hardware.

A. Simulations

Our study begins by training the RL agent to emulate the custom noise model, introduced in Section III D, on single-qubit circuits. To train the model, we generated a dataset of 100 random Clifford circuits of depth 10. We have used 80% of the dataset for the training, reserving the remaining 20% for the test set. The performance of the agent is evaluated by determining the average fidelity and TD between the DMs it produces and the actual noisy DMs. These values are computed across all episodes for both the training and test sets, as shown in Figure 4. The training process converges after roughly 4×10^5 episodes, achieving an average fidelity of about 0.99 on the test set. The RL agent effectively learns to simulate the noise, exhibiting no signs of over-

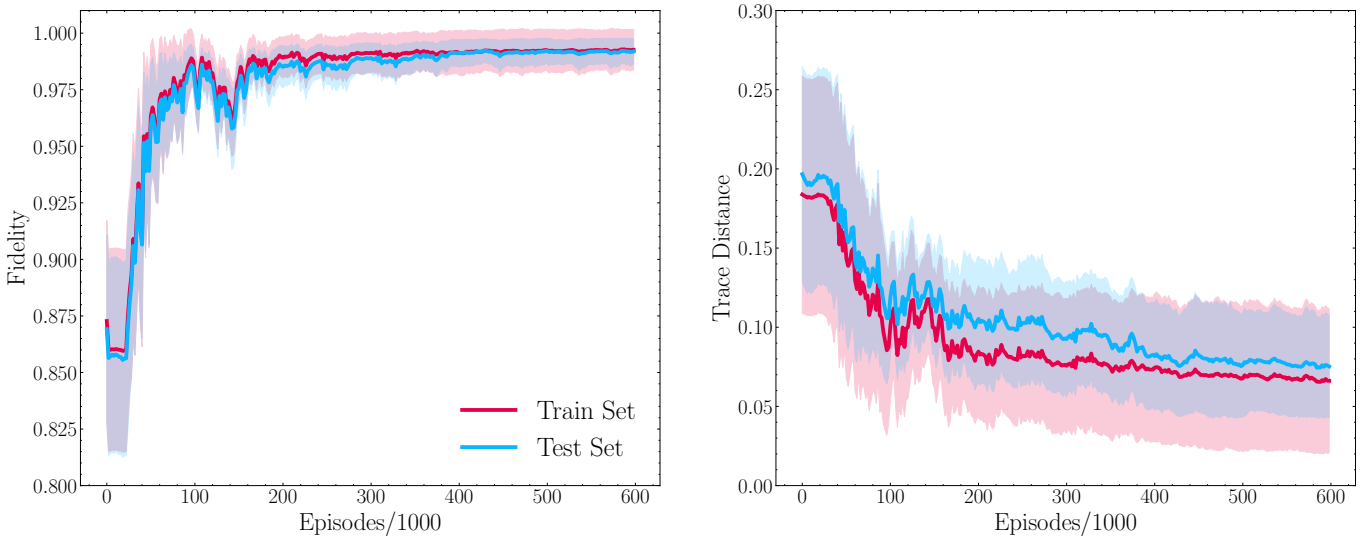


FIG. 4. Average density matrix fidelity (left) and trace distance (right) throughout the training process of the RL agent on single-qubit circuits. The metrics have been evaluated on a dataset of 100 circuits using 80% for the train set and 20% for the test set. Error bars report the standard deviation.

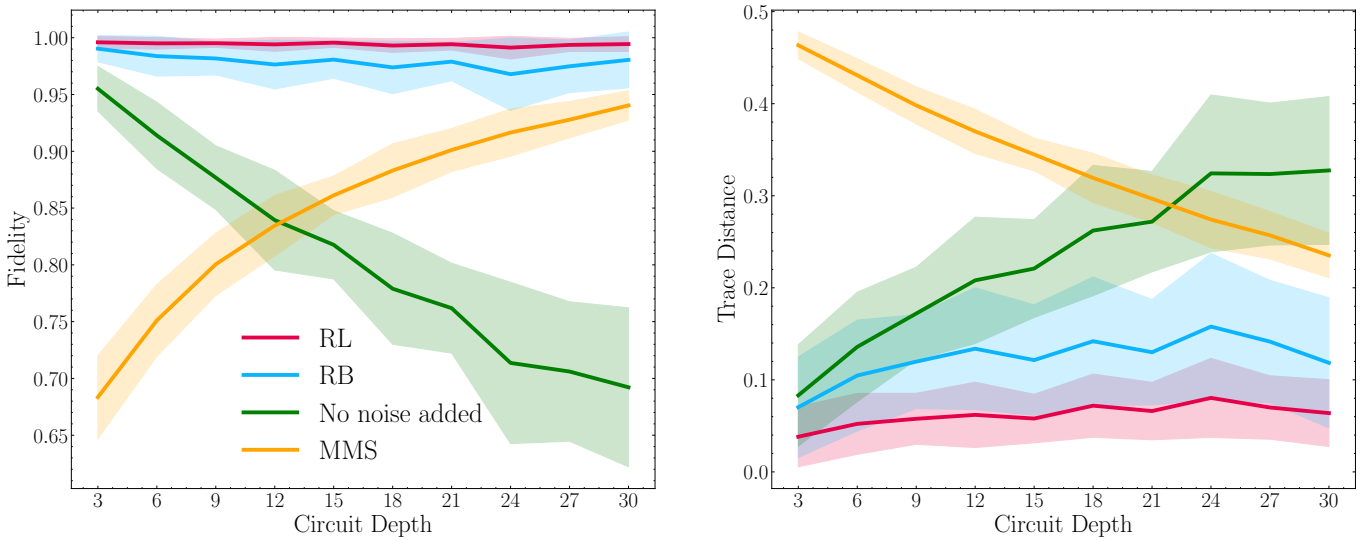


FIG. 5. Performance evaluation of different noise models on Clifford circuits of depths varying from 3 to 30. The performance is evaluated using average DMs fidelity (left) and trace distance (right). The RL agent has been benchmarked against the RB method, the noiseless scenario and the maximally mixed state (MMS). Error bars report the standard deviation.

fitting. For the evaluation set, we used a dataset of 100 random non-Clifford circuits with a depth of 15. The average fidelity of the RL agent on the evaluation set is 0.993, with a standard deviation of 0.003. This result shows that the agent is able to correctly generalize to non-Clifford circuits.

To further assess the model's generalization capability, we evaluate it on random Clifford circuits of varying depths, from 3 to 30. Also in this case we evaluate the performance using the average fidelity and TD between the DMs generated by the model and the actual noisy DMs. Figure 5 compares the performance of the RL agent with the RB noise model described in Section II A). We also offer a comparison with two limit cases: the DMs of noiseless simulation and the DM of the maximally mixed state (MMS). The RL agent demonstrates its adaptability to circuits of different depths, consistently outperforming RB. This performance suggests that while RB categorizes all noise sources as depolarizing, our algorithm can discern the specific characteristics of the noise. The improvement is especially pronounced on shorter circuits, as the circuit depth increases, the noise approximates to a global depolarizing channel, reducing the relative advantage of our RL agent.

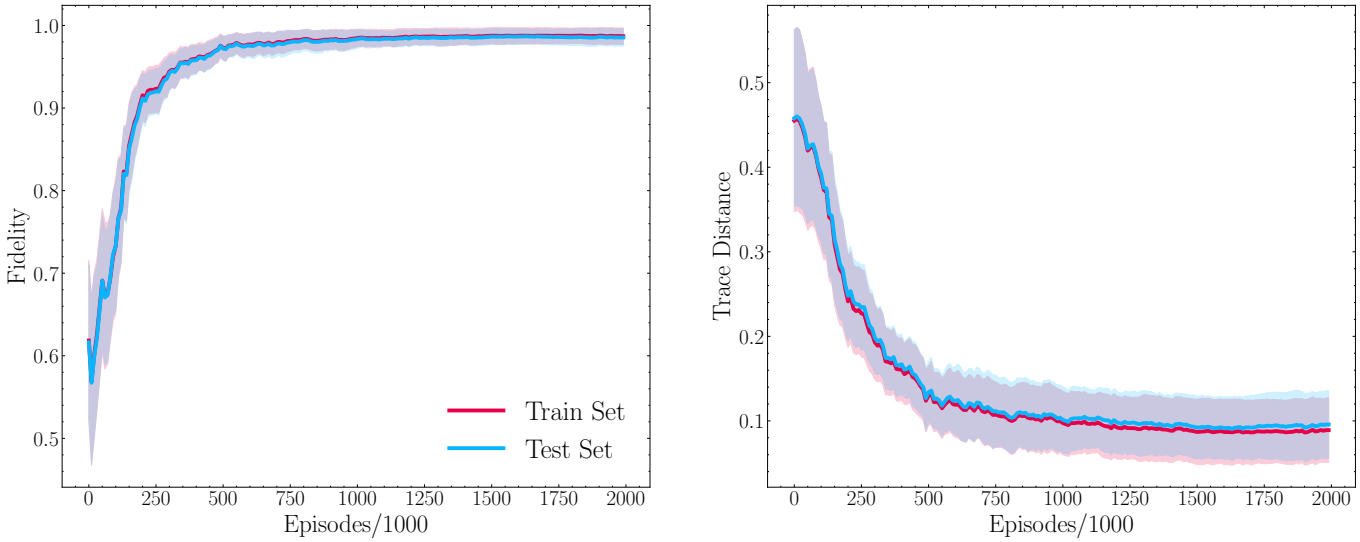


FIG. 6. Average density matrix fidelity (left) and trace distance (right) throughout the training process of the RL agent on three-qubit circuits. The metrics have been evaluated on a dataset of 800 circuits using 80% for the train set and 20% for the test set. Error bars report the standard deviation.

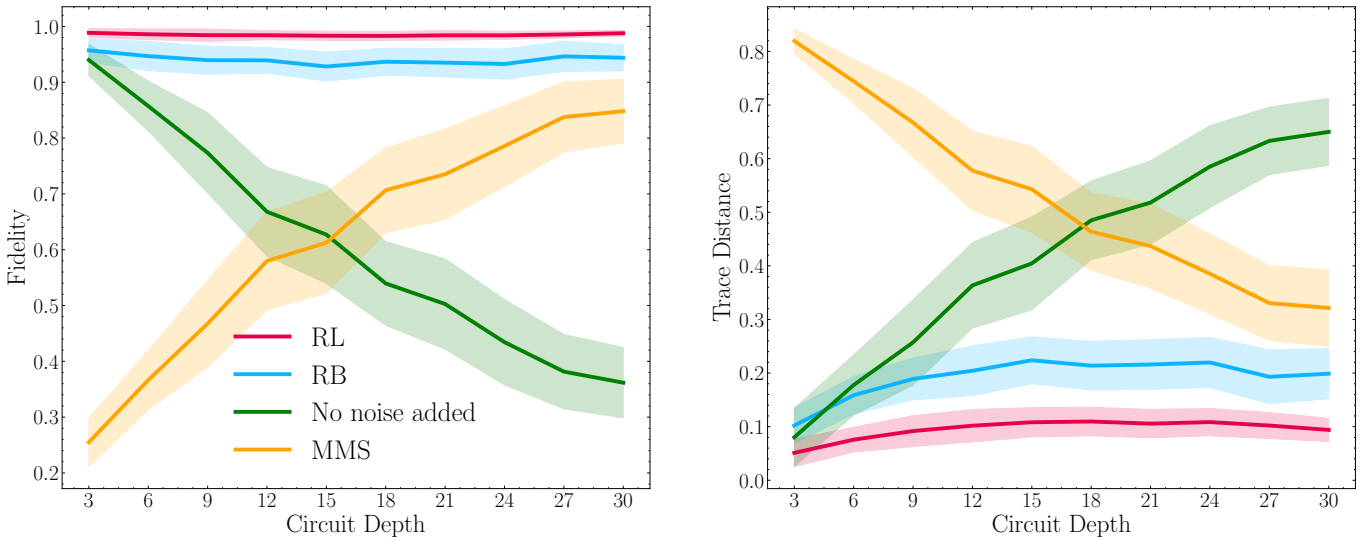


FIG. 7. Performance evaluation of different noise models on three-qubit Clifford circuits of depths varying from 3 to 30. The performance is evaluated using average DMs fidelity (left) and trace distance (right). The RL agent has been benchmarked against the RB method, the noiseless scenario and the maximally mixed state (MMS).

We extended our simulation to three-qubit circuits to evaluate performance in the presence of two-qubit gates. We have used the high noise model on three qubits reported in Table I. The training set consists of 800 three-qubit circuits, divided into 80% for training (640 circuits) and 20% for testing (160 circuits). This training set is composed of both Clifford and non-Clifford circuits as described in Section III D. The evolution of the average DMs fidelity and TD throughout the training process is shown in Figure 6. While the algorithm is capable of learning the noise, the convergence is slower than in single-qubit circuits due to a larger action space requiring more episodes for exploration. Convergence is reached after approximately 1.5×10^6 episodes achieving an average fidelity of about 0.98. No signs of overfitting can be observed during the training phase.
For evaluation, we used a dataset of 200 random non-Clifford circuits of depth 15, where the RL agent achieved an average fidelity of 0.98 with a standard deviation of 0.01.

Mirroring our previous single-qubit circuit experiments, we evaluated the performance of our RL agent against both the RB method and the limit cases of noise-

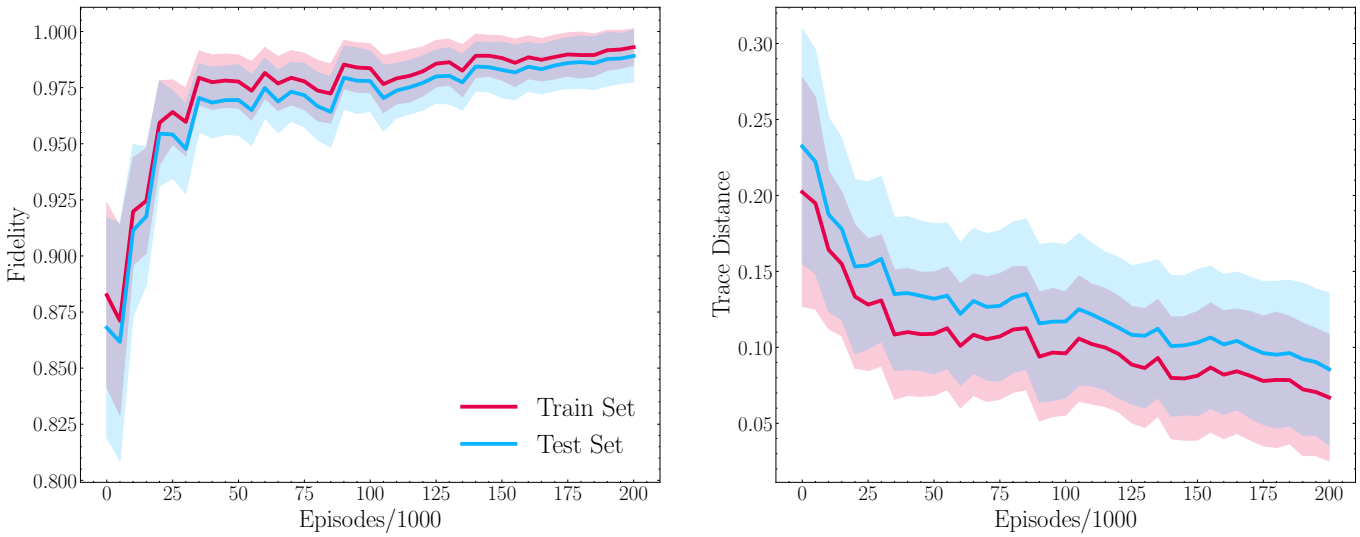


FIG. 8. Average density matrix fidelity (left) and trace distance (right) throughout the training process of the RL agent on single-qubit circuits executed on a superconductive quantum chip. The metrics have been evaluated on a dataset of 60 circuits using 80% for the train set and 20% for the test set. Error bars report the standard deviation.

less circuits and MMS across a variety of circuit depths. This comparison is reported in Figure. 7. In these scenarios, the RL agent consistently demonstrates its adaptability to circuits of different depths, surpassing the performance of the RB method for all circuit lengths.

B. Quantum hardware

To test our algorithm on real quantum hardware, we used a single qubit in a superconducting transmon chip [70]. This 17 qubits chip has been produced by QuantWare [2] and hosted at the Technology Innovation Institute of Abu Dhabi. The single qubit gate fidelity, obtained with RB, is 0.996 with a readout fidelity of about 0.96. To compute the DMs of the circuits, we used state tomography, running 4×10^3 shots to compute each matrix. We attempted to mitigate measurement noise in advance to improve the fidelity of the DMs [71]. However, we observed that the RL agent performs better in learning the noise model without measurement error mitigation.

To train the algorithm, we collected a dataset consisting of 60 random circuits of depth 10, employing 80% of these circuits for the training set and the remaining 20% for the test set. Figure 8 reports the average DM fidelity and TD during training for the first 2×10^5 episodes. Convergence is reached after about $1.5 \cdot 10^5$ episodes, reaching an average fidelity of 0.99, with no evident signs of overfitting. This result is similar to the one obtained with simulations for single qubit circuits.

Using a qubit with high gate fidelity makes the training process more challenging. The ground truth density matrices are affected by both gate errors, which are learned by the reinforcement learning (RL) algorithm, and by shot noise and measurement errors. In the high gate fidelity regime, measurement noise and shot noise can have an impact similar to that of gate noise. These errors in the density matrices make the reward signal less precise, thereby worsening the convergence of the training process. For our dataset, we have computed the average error on the trace distance introduced by measurement errors, shot noise and the additional gates needed to perform state tomography. The obtained value is 0.036, which explains the large error bars observed during training, as shown in Figure 8. To mitigate this problem it would be useful to train the algorithm on longer circuits so that more gate errors can accumulate. In our case, to help the training process, it has been fundamental to set the maximum noise parameter value P_{max} to a low value. As described in Section III B, we have set P_{max} to 0.008, twice the decay parameter obtained with RB.

The performance benchmarking of the RL agent with respect to RB has been performed as described in Section IV A. For this test, we used circuits of length spanning from 5 to 50, 10 circuits for each length. As the noise level of the qubit is quite low, it is necessary to use circuits with higher lengths to extract the decay parameter for the RB method. Figure 9 reports the performance, in reconstructing the noisy DMs, of the trained RL agent compared with RB. The comparison also includes circuits without noise and the maximally mixed state. To better observe the performance differences between the RL and RB noise models, we used a logarithmic scale

[2] https://www.quantware.com

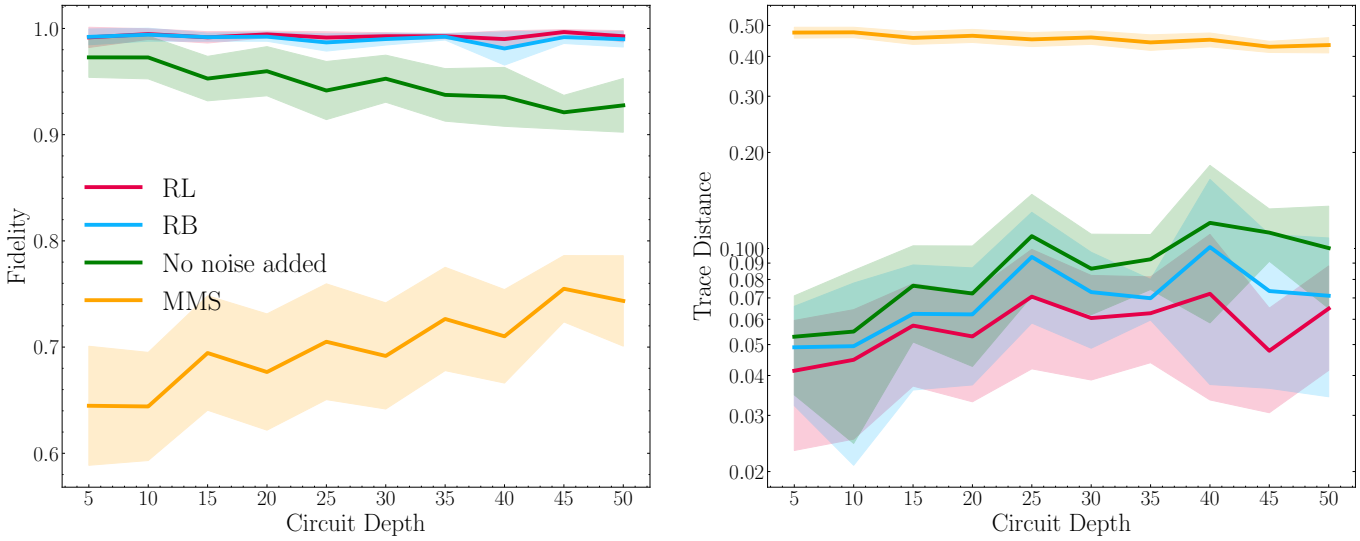


FIG. 9. Performance evaluation of different noise models on single qubit Clifford circuits of depths varying from 5 to 50. The performance is evaluated using average DM fidelity (left) and trace distance (right) with respect to the real noisy DMs obtained from execution on quantum hardware. The RL agent has been benchmarked against the RB method, the noiseless scenario, and the maximally mixed state (MMS). In the plot reporting the trace distance, we used a logarithmic scale to enhance distinguishability between RB and RL.

Noise Model		5	10	15	20	25	30	35	40	45	50
RL	Avg	0.992	0.995	0.992	0.994	0.991	0.993	0.993	0.990	0.997	0.993
	Std	0.009	0.005	0.005	0.003	0.005	0.003	0.002	0.008	0.002	0.005
RB	Avg	0.992	0.994	0.992	0.992	0.987	0.990	0.992	0.981	0.992	0.990
	Std	0.007	0.006	0.004	0.004	0.008	0.006	0.003	0.016	0.006	0.007

TABLE II. Average fidelity and standard deviation in reconstructing noisy DMs of circuits with different depths obtained using the RL agent and RB noise model.

in the plot reporting the TD. On average, the RL agent outperforms the RB method for all circuit lengths when using the trace distance as the metric. The high standard deviation obtained in the plot is mainly due to the uncertainty in the evaluation of the DMs with state tomography. Regarding the fidelity, we present the results obtained with RB and RL in Table II, as the two series are not easily distinguishable from the plot. For this metric as well, the performance of the RL agent is better than or equal to RB for all circuit lengths.

The results obtained in this section differ from those obtained in Section IV A, where the RL agent clearly outperformed the RB method. However, it is important to highlight that the RL agent requires fewer hardware resources than the RB method. In the training of the RL agent, a dataset of 60 circuits with a depth of 10 has been sufficient. The dataset generation process on the employed hardware took just a few minutes, at the cost of some classical resources needed for training the algorithm. Conversely, to estimate the RB parameter with good precision, it is necessary to run many circuits for each depth.

V. APPLICATIONS

This section we assess our model's performance on Quantum Fourier Transform (QFT) and Grover's algorithm circuits under simulated noise. These tests provide valuable insights into the generalization capabilities of the model and serve as a stress test and benchmark for overall performance.

The QFT, a quantum counterpart of the classical Fast Fourier Transform, is a fundamental component of numerous quantum algorithms, including the renowned Shor's algorithm for factoring [2]. The QFT acts on a quantum state $|x\rangle$ of n qubits as

$$\mathrm{QFT}\,|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{k=0}^{2^n-1} e^{2\pi i x k/2^n}\,|k\rangle\ . \tag{6}$$

Being a unitary transformation, the QFT can be implemented as a quantum circuit. For a three-qubit system ($n = 3$), the QFT circuit is reported in Figure 10. The layer of SWAP gates included to reorder the qubits is omitted in our implementation, as this reordering can be handled classically if the QFT is the final operation in an algorithm.

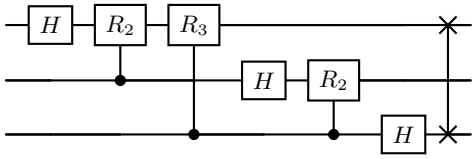


FIG. 10. QFT circuit for three qubits. In this circuit, $R_k = U_1(2\pi/2^k)$ and H denotes the Hadamard gate.

Grover's algorithm, another cornerstone of quantum computing, is renowned for its ability to search unsorted databases with quadratic speedup compared to classical algorithms [3]. The algorithm operates on a superposition of quantum states, and its goal is to find a specific state $|w\rangle$ that satisfies a certain condition defined by an oracle function. The key component of Grover's algorithm is the Grover iterate, a unitary transformation that contain the information of the oracle. The Grover iterate is typically repeated $\mathcal{O}\sqrt{N}$ times to maximize the probability of measuring $|w\rangle$, where N is the dimension of the system. We considered a two-qubit system and the target state $|11\rangle$. This configuration requires only one Grover iteration to find the target state. We utilized an ancillary qubit to construct the oracle, leading to the final circuit reported in Figure 11.

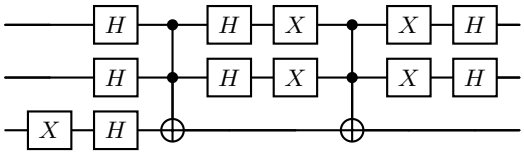


FIG. 11. Grover's search algorithm circuit, the target state is $|11\rangle$ and an ancillary qubit is required.

We transpiled the circuits to utilize the native gates R_z, R_x, and CZ. The gate number after transpilation and other significant circuit parameters are detailed in Table III. It should be noted that the increased depth of the circuits, the lower fraction of two-qubit gates, and the structure, compared to the circuits in the training set, make this generalization test particularly challenging.

Circuit	Total gates	CZ gates	Moments
QFT	23	6	15
Grover	40	7	25

TABLE III. Number of gates and circuit moments of the transpiled circuits for QFT and Grovers's algorithm.

We have tested the algorithms using both the high and low noise models for thee-qubit circuits detailed in Section IV A This approach served a dual purpose. Primarily, we aimed to test the algorithm's generalization capabilities with noise models exhibiting lower error rates. Secondly, the noise model from Section IV A completely masked the final result for Grover's algorithm. By using a noise model with lower error rates, we ensured that a peak at the target state in the final result remained discernible. We evaluated the performance of our RL agent against the RB noise model and the limit case where no noise is added to algorithm's circuits. The fidelity between the reconstructed and original noisy DMs for the different noise models is detailed in Table IV. In all instances, the RL agent obtained the best performance.

Circuit	Noise	RL	RB	No noise added	MMS
QFT	High	0.99	0.97	0.59	0.70
	Low	0.99	0.99	0.78	0.52
Grover	High	0.98	0.95	0.40	0.83
	Low	0.98	0.96	0.65	0.64

TABLE IV. Fidelity between the density matrix reconstructed with a noise model (RL agent, RB, the limit case where no noise channels are added and the MMS) and the ground truth noisy one. The result is reported for QFT and Grover's algorithm circuits for both a high and low noise models.

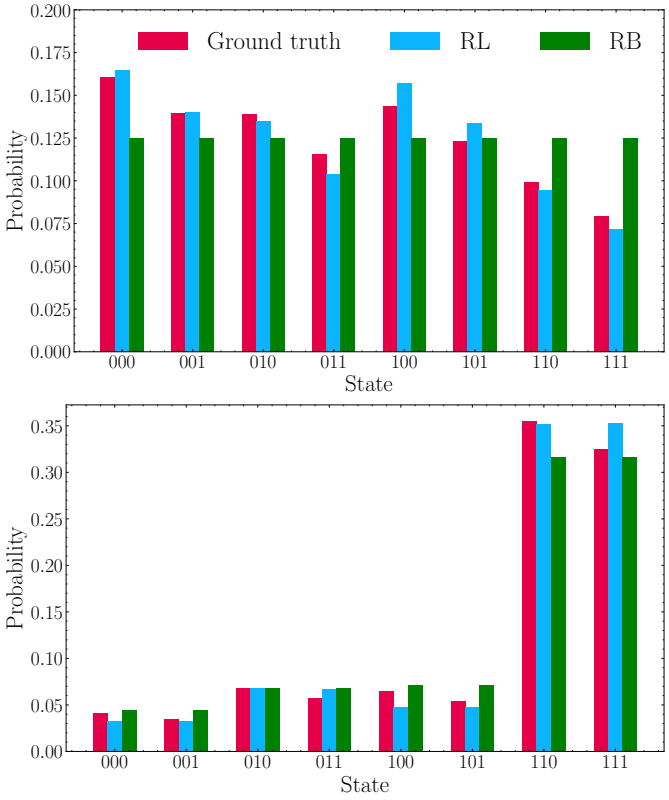


FIG. 12. Computational basis states probabilities for the QFT circuit with a high error noise model (top) and Grover's algorithm circuit with a low error noise model (bottom). The histograms show a comparison between probabilities obtained with the ground truth noise model, the RL agent noise model and the RB noise model.

The final state probabilities for both the QFT and Grover's algorithm circuits, measured in the computational basis, are shown in Figure 12. For the QFT cir-

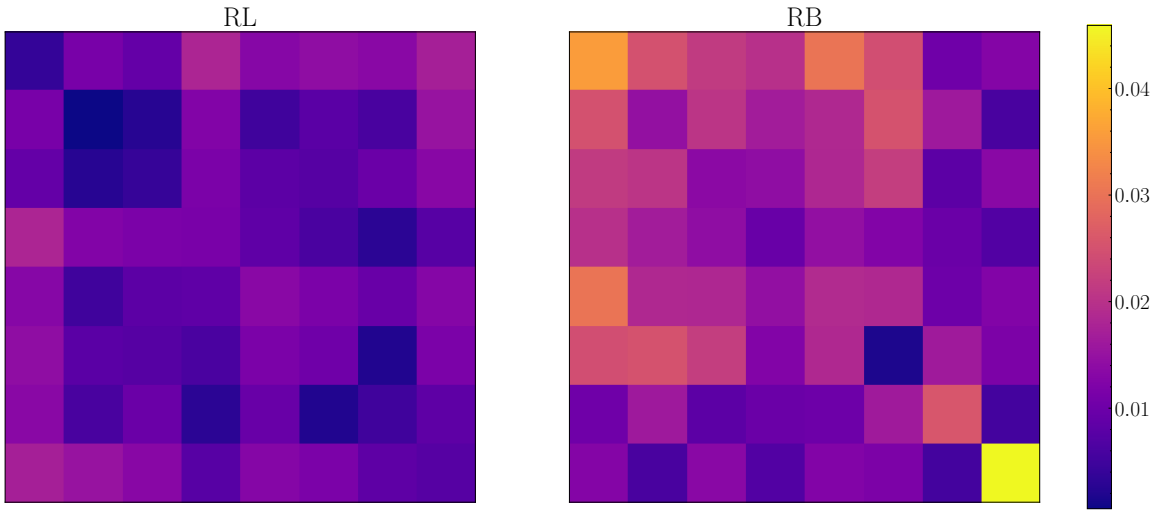


FIG. 13. Heatmap of the absolute error between the ground truth noisy DM and the DM obtained with the RL agent (left) and RB model (right) for the three-qubit QFT circuit.

cuit we report the result obtained with a high error noise model while for the Grover's circuit we report the result obtained with a low error noise model that doesn't destruct the expected result. The histograms compare the probabilities derived from the original noisy circuit, the reconstructed circuit using the RL agent, and the RB noise model. Given the depth of the circuits used, the RB noise model tends to average the output. In contrast, the RL agent, with a few exceptions, aligns more closely with the probabilities obtained from the original noise model. This alignment is particularly noticeable in the QFT simulation counts for the state $|000\rangle$. This state has the higher probability due to the noise model's amplitude damping channels, a feature that is successfully replicated by the RL agent but that is is not possible to replicate with the RB model.

For a more detailed analysis, Figure 13 shows the absolute error between the DMs generated by the RL algorithm and the RB model compared to the ground truth noisy DM. The errors in the RL model are well distributed across the DM, while the RB model's errors tend to be higher along the diagonal. This effect was observed in many circuits, even during preliminary tests with the RL algorithm. As the depth of the circuit increases, the density matrix of noisy circuits tends to approach the MMS, resulting in most of the information being contained in the diagonal. This is one of the main reasons we chose the trace distance as the metric for the reward of the RL algorithm.

The results obtained in this section underscore the generalization capabilities of the proposed RL approach for noise modeling. The RL agent adapts to circuits with structures distinct from the random ones used in the training set and of significantly different depths. This allows the algorithm to be used for circuit simulations for interesting use cases like quantum algorithms and quantum machine learning.

VI. CONCLUSIONS

This work presents a reinforcement learning algorithm for replicating specific noise models in single and multiple qubit quantum circuits. This approach reduces heuristic assumptions about the noise model, enhancing generalization properties. Tests on simulated and quantum hardware circuits have demonstrated the model's ability to learn complex noise patterns and generalize to unseen circuits. For a comprehensive evaluation and to show a possible use case, we tested the model on QFT and Grover's algorithm circuits. In all occasions, the RL model consistently outperformed a common noise characterization method, randomized benchmarking both in the ability of reconstructing the density matrices and in the amount of quantum hardware resources needed.

Possible future applications of the algorithm include not only reproducing the noise pattern of a specific hardware device. By learning the error patterns of qubits for specific gate types, the model could optimize the transpilation process [72], thereby enhancing quantum algorithms fidelity. Furthermore, using the knowledge of the noise for its mitigation could be an interesting approach.

The current model's limitation is its scalability to circuits with many qubits. There are two main challenges to overcome. The first problem is that scaling the model would necessitate a significant increase in the number of actions, A lager action space would complicate and slow down the training process. The second problem regards obtaining the ground truth density matrices via quantum state tomography as it requires exponentially more measurements in the number of qubits. We are considering potential solutions to these challenges. One approach, to solve the second issue, could involve training the model with probability distributions derived from measurements, rather than density matrices. It

is also possible to employ machine learning techniques to reduce the amount of measurements needed for quantum state tomography [73, 74]. To address the first issue, we could partition large circuits into smaller ones, facilitating parallel training of multiple smaller models. Furthermore, it could be beneficial to explore the use of graph neural networks in place of convolutional neural networks to encode qubit connectivity information into the model.

While these ideas require further validation, this work demonstrates that machine learning's application to learn noise patterns within small quantum circuits is a promising proof of concept that could lead to future advancements.

DECLARATIONS

Author contribution: S. Bordoni, A. Papaluca, P. Buttarini, A. Sopena: Methodology, Software, Writing. S. Giagu, S. Carrazza: Project Administration, Review, Funding.

Funding: This work is partially supported by the Technology Innovation Institute (TII), Abu Dhabi, UAE.
S. B. and S. G. are partially supported by ICSC - Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union - NextGenerationEU.
A. S. is supported through the Spanish Ministry of Science and Innovation grant SEV-2016-0597-19-4, the Spanish MINECO grant PID2021- 127726NB-I00, the Centro de Excelencia Severo Ochoa Program SEV-2016-0597 and the CSIC Research Platform on Quantum Technologies PTI-001.

Data Availability: Both the datasets and the code used for this study are available on GitHub: https://github.com/qiboteam/rl-noisemodel .

Conflict of interest: The authors declare no conflict of interest.

[1] J. Preskill, Quantum **2**, 79 (2018).
[2] P. W. Shor, SIAM Journal on Computing **26**, 1484 (1997).
[3] L. K. Grover, A fast quantum mechanical algorithm for database search (1996), arXiv:quant-ph/9605043 [quant-ph].
[4] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, P. Hu, X.-Y. Yang, W.-J. Zhang, H. Li, Y. Li, X. Jiang, L. Gan, G. Yang, L. You, Z. Wang, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Science **370**, 1460 (2020).
[5] P. V. Klimov, J. Kelly, Z. Chen, M. Neeley, A. Megrant, B. Burkett, R. Barends, K. Arya, B. Chiaro, Y. Chen, A. Dunsworth, A. Fowler, B. Foxen, C. Gidney, M. Giustina, R. Graff, T. Huang, E. Jeffrey, E. Lucero, J. Y. Mutus, O. Naaman, C. Neill, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, S. Boixo, R. Babbush, V. N. Smelyanskiy, H. Neven, and J. M. Martinis, Phys. Rev. Lett. **121**, 090502 (2018).
[6] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Annual Review of Condensed Matter Physics **11**, 369 (2020), https://doi.org/10.1146/annurev-conmatphys-031119-050605.
[7] P. Zhao, K. Linghu, Z. Li, P. Xu, R. Wang, G. Xue, Y. Jin, and H. Yu, PRX Quantum **3**, 020301 (2022).
[8] I. Heinz and G. Burkard, Phys. Rev. B **104**, 045420 (2021).
[9] T. Patel, A. Potharaju, B. Li, R. B. Roy, and D. Tiwari, in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis* (2020) pp. 1–15.
[10] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, PRX Quantum **2**, 040326 (2021).
[11] L. Funcke, T. Hartung, K. Jansen, S. Kühn, P. Stornati, and X. Wang, Phys. Rev. A **105**, 062404 (2022).
[12] S. S. Tannu and M. K. Qureshi, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '52 (Association for Computing Machinery, New York, NY, USA, 2019) p. 279–290.
[13] A. Sopena, M. H. Gordon, G. Sierra, and E. López, Quantum Science and Technology **6**, 045003 (2021).
[14] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, PRX Quantum **2**, 040330 (2021).
[15] T. Schuster, C. Yin, X. Gao, and N. Y. Yao, A polynomial-time classical algorithm for noisy quantum circuits (2024), arXiv:2407.12768 [quant-ph].
[16] E. Knill, Nature **434**, 39 (2005).
[17] K. Fukui, A. Tomita, A. Okamoto, and K. Fujii, Phys.

Rev. X **8**, 021054 (2018).
[18] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[19] S. Varsamopoulos, B. Criger, and K. Bertels, Quantum Science and Technology **3**, 015004 (2017).
[20] B. Simone and G. Stefano, Quantum Information Processing **22**, 151 (2023).
[21] A. Cacioppo, L. Colantonio, S. Bordoni, and S. Giagu, Quantum diffusion models (2023), arXiv:2311.15444 [quant-ph].
[22] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).
[23] J. Choi and J. Kim, in *2019 International Conference on Information and Communication Technology Convergence (ICTC)* (2019) pp. 138–142.
[24] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, and L. Cincio, Nature Reviews Physics **3**, 625 (2021).
[25] D. J. Egger, C. Capecci, B. Pokharel, P. K. Barkoutsos, L. E. Fischer, L. Guidoni, and I. Tavernelli, Physical Review Research **5**, 033159 (2023).
[26] S. Bordoni, D. Stanev, T. Santantonio, and S. Giagu, Particles **6**, 297 (2023).
[27] M. Robbiati, J. M. Cruz-Martinez, and S. Carrazza, Determining probability density functions with adiabatic quantum computing (2023), 7 pages, 3 figures, arXiv:2303.11346.
[28] M. Robbiati, S. Efthymiou, A. Pasquale, and S. Carrazza, A quantum analytical adam descent through parameter shift rule using qibo (2022), arXiv:2210.10787 [quant-ph].
[29] J. M. Cruz-Martinez, M. Robbiati, and S. Carrazza, Quantum Science and Technology **9**, 035053 (2024).
[30] G. Ravi, K. N. Smith, P. Gokhale, and F. T. Chong, in *2021 IEEE International Symposium on Workload Characterization (IISWC)* (IEEE Computer Society, Los Alamitos, CA, USA, 2021) pp. 39–50.
[31] A. Zlokapa and A. Gheorghiu, A deep learning model for noise prediction on near-term quantum devices (2020), arXiv:2005.10811 [quant-ph].
[32] C. J. Wood, in *2020 IEEE 38th International Conference on Computer Design (ICCD)* (2020) pp. 13–16.
[33] K. Georgopoulos, C. Emary, and P. Zuliani, Phys. Rev. A **104**, 062432 (2021).
[34] R. Harper, S. T. Flammia, and J. J. Wallman, Nature Physics **16**, 1184 (2020).
[35] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).
[36] L. P. Kaelbling, M. L. Littman, and A. W. Moore, Journal of artificial intelligence research **4**, 237 (1996).
[37] M. A. Wiering and M. Van Otterlo, Adaptation, learning, and optimization **12**, 729 (2012).
[38] qibolab_platforms_qrc (2023).
[39] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, Quantum Science and Technology **7**, 015018 (2021).
[40] S. Efthymiou, M. Lazzarin, A. Pasquale, and S. Carrazza, Quantum **6**, 814 (2022).
[41] S. Efthymiou, A. Orgaz-Fuertes, R. Carobene, J. Cereijo, A. Pasquale, S. Ramos-Calderer, S. Bordoni, D. Fuentes-Ruiz, A. Candido, E. Pedicillo, M. Robbiati, Y. P. Tan, J. Wilkens, I. Roth, J. I. Latorre, and S. Carrazza, Quantum **8**, 1247 (2024).
[42] E. Pedicillo, A. Candido, S. Efthymiou, H. Sargsyan, Y. P. Tan, J. Cereijo, J. Y. Khoo, A. Pasquale, M. Robbiati, and S. Carrazza, An open-source framework for quantum hardware control (2024), arXiv:2407.21737 [quant-ph].
[43] A. Pasquale, S. Efthymiou, S. Ramos-Calderer, J. Wilkens, I. Roth, and S. Carrazza, Towards an open-source framework to perform quantum calibration and characterization (2023), arXiv:2303.10397 [quant-ph].
[44] D. Deutsch, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **425**, 73 (1989).
[45] M. D. Reed, B. R. Johnson, A. A. Houck, L. DiCarlo, J. M. Chow, D. I. Schuster, L. Frunzio, and R. J. Schoelkopf, Applied Physics Letters **96**, 203110 (2010).
[46] G. Cenedese, G. Benenti, and M. Bondani, Entropy **25**, 324 (2023).
[47] O. Kern, G. Alber, and D. L. Shepelyansky, The European Physical Journal D **32**, 153 (2005).
[48] S. Bravyi, M. Englbrecht, R. König, and N. Peard, npj Quantum Information **4**, 10.1038/s41534-018-0106-y (2018).
[49] C. Blank, D. Park, J.-K. Rhee, and F. Petruccione, npj Quantum Information **6**, 41 (2020).
[50] J. Emerson, R. Alicki, and K. Życzkowski, Journal of Optics B: Quantum and Semiclassical Optics **7**, S347 (2005).
[51] M. Heinrich, M. Kliesch, and I. Roth, Randomized benchmarking with random quantum circuits (2023), arXiv:2212.06181 [quant-ph].
[52] J. Helsen, X. Xue, L. M. K. Vandersypen, and S. Wehner, npj Quantum Information **5**, 71 (2019), arXiv:1806.02048 [quant-ph].
[53] M. L. Puterman, in *Stochastic Models*, Handbooks in Operations Research and Management Science, Vol. 2 (Elsevier, 1990) pp. 331–434.
[54] J. Zou, Y. Han, and S.-S. So, Overview of artificial neural networks, in *Artificial Neural Networks: Methods and Applications* (Humana Press, Totowa, NJ, 2009) pp. 14–22.
[55] R. Rojas, The backpropagation algorithm, in *Neural Networks: A Systematic Introduction* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1996) pp. 149–182.
[56] R. Özalp, N. K. Varol, B. Taşci, and A. Uçar, A review of deep reinforcement learning algorithms and comparative results on inverted pendulum system, in *Machine Learning Paradigms: Advances in Deep Learning-based Technological Applications* (Springer International Publishing, Cham, 2020) pp. 237–256.
[57] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms (2017), arXiv:1707.06347 [cs.LG].
[58] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, Journal of Machine Learning Research **22**, 1 (2021).
[59] A. Qiu, D. Wang, S. Partani, and H. Schotten, Modern openai gym simulation platforms for vehicular ad-hoc network systems (2023).
[60] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019) pp. 8024–8035.
[61] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai,

T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, Pattern Recognition **77**, 354 (2018).
[62] Y.-C. Liang, Y.-H. Yeh, P. E. M. F. Mendonça, R. Y. Teh, M. D. Reid, and P. D. Drummond, Reports on Progress in Physics **82**, 076001 (2019).
[63] M. Christandl and R. Renner, Phys. Rev. Lett. **109**, 120403 (2012).
[64] G. Struchalin, Y. A. Zagorovskii, E. Kovlakov, S. Straupe, and S. Kulik, PRX Quantum **2**, 010307 (2021).
[65] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, Nature Reviews Physics **2**, 382 (2020).
[66] S. Chen, W. Yu, P. Zeng, and S. T. Flammia, PRX Quantum **2**, 030348 (2021).
[67] A. C. Santos, Revista Brasileira de Ensino de Fisica **39**, 10.1590/1806-9126-rbef-2016-0155 (2016).
[68] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Phys. Rev. A **77**, 012307 (2008).
[69] S. Bravyi and D. Maslov, IEEE Transactions on Information Theory **67**, 4546 (2021).
[70] M. H. Devoret and R. J. Schoelkopf, Science **339**, 1169 (2013), https://www.science.org/doi/pdf/10.1126/science.1231930.
[71] B. Nachman, M. Urbanek, W. A. De Jong, and C. W. Bauer, npj Quantum Information **6**, 84 (2020).
[72] E. Wilson, S. Singh, and F. Mueller, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020) pp. 345–355.
[73] N. Innan, O. I. Siddiqui, S. Arora, T. Ghosh, Y. P. Koçak, D. Paragas, A. A. O. Galib, M. A.-Z. Khan, and M. Bennai, Quantum Machine Intelligence **6**, 10.1007/s42484-024-00162-3 (2024).
[74] T. Schmale, M. Reh, and M. Gärttner, npj Quantum Information **8**, 10.1038/s41534-022-00621-4 (2022).