Machine Learning based Noise Characterization and Correction on Neutral Atoms NISQ Devices

Ettore Canonici, Stefano Martina, Riccardo Mengoni, Daniele Ottaviani, and Filippo Caruso*

Neutral atoms devices represent a promising technology using optical tweezers to geometrically arrange atoms and modulated laser pulses to control their quantum states. They are exploited as noisy intermediate-scale quantum (NISQ) processors. Indeed, like all real quantum devices, they are affected by noise introducing errors in the computation. Therefore, it is important to understand and characterize the noise sources and possibly to correct them. Here, two machine-learning based approaches are proposed respectively to estimate the noise parameters and to mitigate their effects using only measurements of the final quantum state. Our analysis is then tested on a real neutral atom platform, comparing our predictions with a priori estimated parameters. It turns out that increasing the number of atoms is less effective than using more measurements on a smaller scale. The agreement is not always good but this may be due to the limited amount of real data that are obtained from a still under development device. Finally, reinforcement learning is employed to design a pulse that mitigates the noise effects. Our machine learning-based approach is espected to be very useful for the noise benchmarking of NISQ processors and, more in general, of real quantum technologies.

E. Canonici, S. Martina, F. Caruso
Department of Physics and Astronomy
University of Florence
via Sansone 1, Sesto Fiorentino 50019, Italy
E. Canonici, S. Martina, F. Caruso
European Laboratory of Non-Linear Spesctroscopy (LENS)
via Carrara 1, Sesto Fiorentino 50019, Italy
E-mail: filippo.caruso@unifi.it
R. Mengoni, D. Ottaviani
CINECA
Via Magnanelli 6/3, Casalecchio di Reno, Bologna 40033, Italy
F. Caruso
Istituto Nazionale di Ottica (INO)
Consiglio Nazionale delle Ricerche (CNR)
via Carrara 1, Sesto Fiorentino 50019, Italy

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/qute.202300192

1. Introduction

In the last few years, we are witnessing a revolution in the field of quantum computing. The so called Noisy Intermediate Scale Quantum (NISQ) devices[1] represent the state of the art in this field. The intermediate scale of such devices refers to the fact that, at the best of our technologies, we are still capable of dealing with at most few hundreds of qubits. Several error correction codes have been developed to deal with such noise,[2–4] but they require the adoption of auxiliary qubits further decreasing the resources available for the computation. *Pasqal*[5] has developed a NISQ device called *Fresnel* based on a neutral atom quantum processor capable of using up to 100 qubits,[6] and they provide a *Python* library called *Pulser*[7] that can be used to prepare a setting either to run it on the real machines or to simulate it on a built-in simulator.

Machine Learning (ML) is a field in the context of Artificial Intelligence (AI) that deals with the study and realization of models that learn to make predictions after being trained with data.[8,9] Artificial Neural Networks (ANNs) are ML methods organized in layers of artificial neurons that performs calculations with weighted summations of the inputs followed by non-linear activation functions. ML methods has already been developed in the context of quantum noise characterization,[10–13] and they have already been adopted in the context of error estimation. In [14], the authors train a recurrent neural network to detect if certain errors happened in a quantum circuit, and they use the model to enhance a surface error correction code. Surface error correction codes allows an high error tolerance, however to be implemented they need an high number of physical qubits.[15] By contrast, in our proposed approach for noise mitigation, no additional qubits are needed for error detection. In fact, our purpose is to learn how to modify the pulses in such a way as to minimize the effect of noise without implementing error correction codes. Moreover, we estimate the noise in devices with the analog interface and not with the digital one. In fact, with neutral atoms devices it is possible to take advantage of analog and digital modes. With the former, laser pulses can be used to directly manipulate the Hamiltonian of the system:

$$H = \frac{\hbar\Omega(t)}{2}\sum_i \sigma_i^x - \frac{\hbar\delta(t)}{2}\sum_i \sigma_i^z + \sum_{i<j} U_{ij} n_i n_j \qquad (1)$$

With the digital mode, on the other hand, it is possible to evolve the state of the system through quantum gates thus creating quantum circuits. In [16], the authors consider the noise to have the form of a Pauli channel, and they make the assumption that the error rate is modeled with a Gibbs Random Field (GRF). Those assumptions allows the authors to effectively learn the parameters of the GRF to characterize the noise of a real IBM NISQ device. As discussed below, in our work we use a different noise formalization, in fact we resort on how the noise is implemented in the Pasqal simulator that we use to generate the data to train the deep learning model.

Reinforcement Learning (RL) is a ML methodology that requires the presence of a simulator of an environment where an agent operates.[17] The agent is usually implemented as a neural network that is trained to implement the policy that governs the actions of the agent. Initially, for each episode (the elementary phase of each Reinforcement Learning (RL) algorithm that is repeated over time and that is constituted of a series of actions of the agent and reactions of the environment), the agent and the environment are initialized in some initial state. Then, the agent perceive some information about the environment and, based on that, the policy follows a probability distribution of the possible next actions that the agent can perform to change the state of the environment or the state of the agent within the environment. The episode continue with the choosing of the best action according to the policy, and new steps will follow for a predefined number of steps or until some episode-ending condition. RL have been already used in the context of state preparation and circuit optimization.[18–20] In the context of noise correction, RL have been adopted to correct the noise that degrades a state over time[21] or to optimize existing quantum correction codes.[22] In our work, we instead focus on the task to correct the effects of the noise of a defined quantum dynamics without modifying the base pulse.

2. Noise Benchmarking Protocol

The computation on neutral atom Quantum Processing Units (QPUs) is structured in cycles of three phases: i) the preparation of the atoms register, ii) the quantum processing that modifies the state of the atoms and iii) the register readout that is the measurement process. In particular, on neutral atoms devices, the preparation of the register is obtained using arrays of optical tweezers.[23] This provides the user great freedom in the choice of geometry of the quantum register. Specifically, the register is initialized with atoms in random positions and afterward, the single atoms are moved in the desired positions with the help of highly focused laser beams. The quantum computation is performed analogically using laser pulses that interact with the register atoms. The irradiation of the atoms can induce the excitement from the ground state to the Rydberg state. The laser pulses are characterized by the values and shapes of the Rabi frequency $\Omega(t)$ and detuning $\delta(t)$, and also by the duration of the pulse τ such that $t \in [0, \tau]$.

Finally, the register readout is performed by taking a fluorescence image of the atomic register.

In such process the atoms in the ground state appear bright, while the others appear dark. At the end, an atom that is in the ground state is measured zero, while the Rydberg state is consid-

Table 1. Summary of the main noise parameters with their respective values. We considered the parameters that are expected to have a predominant effect.

Description	Parameter	Value
Laser Intensity fluctuation	σ_R	3%
Laser waist	w	$68 \mu m$
Temperature	T	$30 \mu K$
False positive measurement	ε	3%
False negative measurement	ε'	8%

ered one. In Pasqal NISQ devices, it is possible to prepare registers of maximum 100 atoms with a minimum distance of $4\mu m$ between them arranged in bidimensional structures in an area of maximum radius $50\mu m$.

NISQ devices, as the name suggests, are affected by several noise effects that limit their applicability and the operations that can be reliably executed on them.

All the technology involved in the realization of a quantum computer is not ideal. It unavoidably introduces noise which, in turn, can be responsible for errors in computation. First of all, lasers are not exactly monochromatic, so the frequency of a pulse may not be exactly the desired one. In addition, atoms at the edges of the quantum register may feel a slightly lower frequency than those in the center. Furthermore, atoms in the quantum register are cooled to very low temperatures, but still not zero. This can be the cause of accidental excitations. The overall effect of the aforementioned problems is to introduce errors into the preparation of the quantum system and its manipulation. At the end, the obtained final state is different from the theoretically expected. In general, there are different parameters that can be used to indicate different sources of noise in the device.[24] In the present work, we will focus on five parameters that are considered predominant for their effects: the laser intensity fluctuation σ_R indicates the standard deviation of the fluctuation of the desired Rabi frequency of the laser pulse; the laser waist w is the diameter of the Gaussian laser beam; the false positive measurements ε represents the probability of wrongly measure as excited an atom that was in the ground state; false negative measurements ε' is the probability of measuring an excited atom in ground state. **Table 1** shows those sources of noise and their estimated values provided informally by Pasqal.

The objective of our work is the implementation of ML models to: i) provide a quantitative estimate of the noise parameters and ii) mitigate the effects of the noise. We decide to formulate a supervised regression task to quantitatively estimate the noise[16] and to use a Reinforcement Learning (RL) framework[17] to mitigate the noise effect. Regarding the noise characterization, our aim is to show that it is possible to estimate the noise parameters in the form of mean values and error intervals. As depicted in **Figure 1**, the workflow begins with the simulation of various executions, with different noise parameters, of a quantum dynamic where a global pulse irradiates all the n atoms of a register. Afterward, the atoms occupation probabilities, that we call $\mathcal{P} = \mathcal{P}_1, \ldots, \mathcal{P}_{2^n}$, are collected and used to train ANN models to predict the noise parameters that were used to perturb the dynamics: *temperature*, laser *waist*, false positive measurement rate
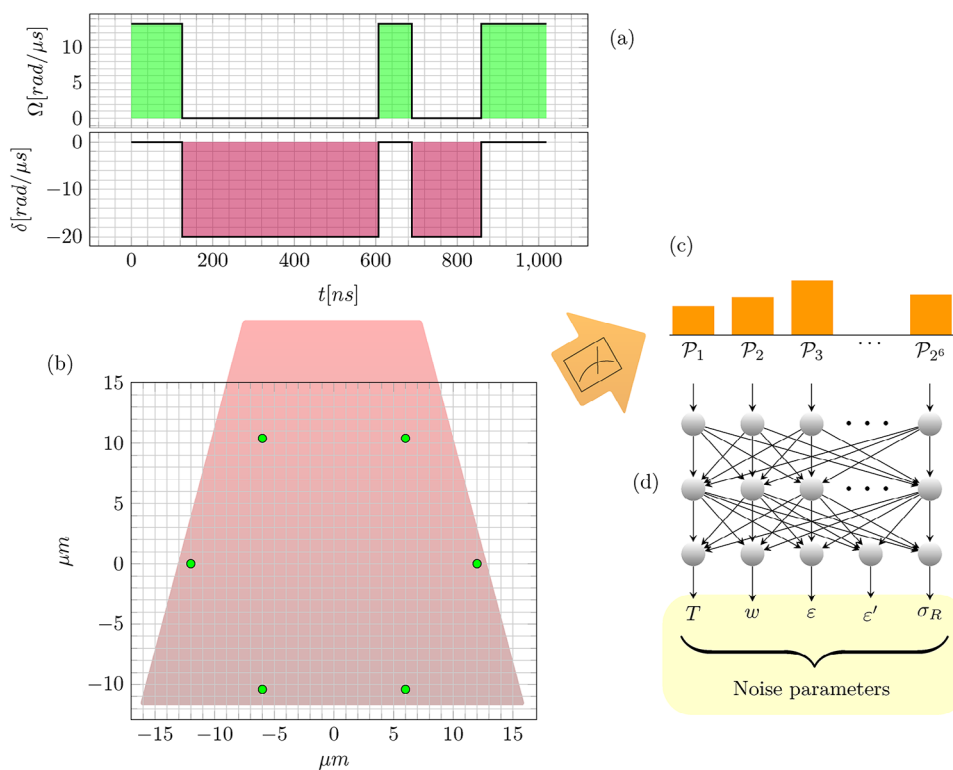
© 2023 The Authors. Advanced Quantum Technologies published by Wiley-VCH GmbH

**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
TECHNOLOGIES**

www.advquantumtech.com

Figure 1. Scheme of the noise estimation pipeline. A global pulse is defined by the shapes of Rabi frequency Ω and detuning δ (a). A register is prepared with the positions of a set of n atoms (six in the specific case) that are irradiated by the laser pulse (b). When the pulse ends, the excitation states of the atoms are measured, and the process is repeated to gather statistics on the occupation probabilities $\mathcal{P} = \mathcal{P}_1, \ldots, \mathcal{P}_{2^n}$ (c). The probabilities are used as input to an Artificial Neural Network (ANN) that predicts the noise parameters (d). The ANN is trained collecting a simulated dataset of probabilities labeled with the corresponding values of noise. The depicted setting is for the more general multiple parameters estimation. The differences for the single parameter estimation are that the neural network have only one output for σ_R and that the adopted pulses and atoms registers are different.

ε, false negative measurement rate ε', and intensity fluctuation σ_R. At the end, the trained models are used on prediction with the real data to obtain an estimation of the noise parameters. For the the simulations used in the generation of the data, and for the training of the models, we use our servers with Nvidia TITAN RTX and GeForce RTX 3090 GPUs. Moreover, we could also make use of the CINECA Marconi100 supercomputer.

The rest of the paper is structured as in the following. First, in Section 3.1 we consider the simpler problem of characterizing only a single noise parameter, then in Section 3.2 we show the results of the characterization of all the aforementioned parameters. Finally, in Section 4 we illustrate the RL error correction protocol that we adopt.

3. Results and Discussion

3.1. Single Parameter Noise Characterization

In this section, we consider the estimation of a single noise parameter. After preliminary analysis, we decide to focus on the noise effects that comes from the laser intensity fluctuations σ_R. In order to perform such characterization, we consider registers with a number of qubits from 2 to 5. We denote by s_i the system composed of i qubits. In addition, we consider six different topologies with four qubits, and we use for those an extra

alphanumeric index from a to f. Specifically, the set of the possible registers is: $s_2, s_3, s_{4a}, s_{4b}, \ldots, s_{4f}, s_5$.

Globally, we collected the measurements of nine different runs on the real Pasqal NISQ device (six different topologies with four atoms and single topologies with 2,3, and 5 atoms). Each run is characterized by a pulse of duration $\tau = 660\,ns$ with constant Rabi frequency $\Omega(t) = 2\pi\,rad/\mu s$ and null detuning $\delta(t) = 0$, for $t \in [0, \tau]$, but with different number and positions of the atoms.

In order to train the ML models to predict the values of σ_R, we proceed by simulating the datasets. Each of the nine datasets (one for each quantum register) is obtained by noisy simulations. However, in order to combine the analyses of the simulations of the nine quantum registers with each other, we need to make sure that the noise (i.e., the value of σ_R) used to run the simulations is the same for each register. For this reason, before simulating, we sample a set of 10 000 values of σ_R from a uniform distribution $\mathcal{U}(0, 0.15)$. These values will be used to run 10 000 simulations for each quantum register. Therefore, in the end the dataset will consist, for each quantum register, of 10 000 pairs of occupation probability and value of σ_R used for the simulation.

We use occupation probabilities as input to the models that are trained to predict ground truth σ_R.

The occupation probabilities, associated with the corresponding values of σ_R, are used to evaluate two different scalings taking advantage of the nine systems: (i) in the quantum register

**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
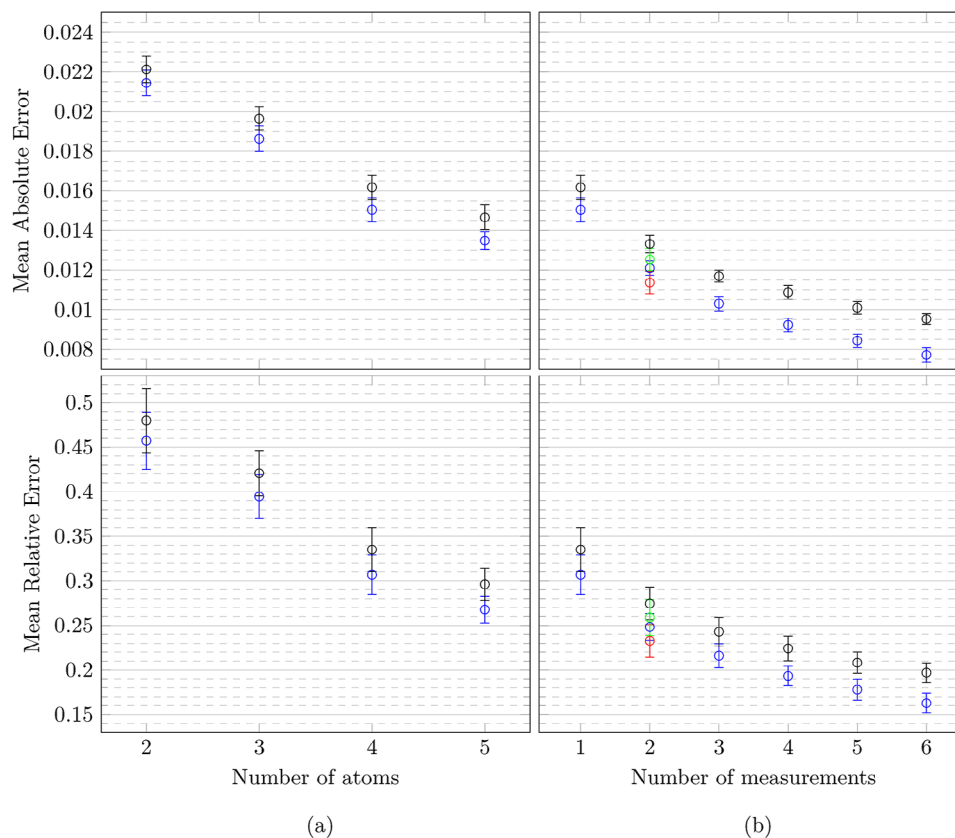TECHNOLOGIES**

www.advquantumtech.com

Figure 2. Scaling of single measurement for systems with an increasing number of atoms (the two left plots denoted by (a)) , and scaling in the number of measurements for systems with four atoms (the two right plots denoted by (b)). We report the average of either the absolute (top plots) and relative (bottom plots) errors with the correspondent standard deviations for 20 linear regression (in black and green) and 20 ANN (in blue and red) models in the predictions of σ_R on the synthetic validation set. The models in (a) uses as input the measurements of s_2, s_3, s_{4a} and s_5. The models in (b) uses as input one or more concatenated measurements of runs of the settings with four atoms (the fourth pair of points in (a) is equal to the first pair in (b)). Indicating with $\cdot \oplus \cdot$ the concatenation of the measurements of the settings, we report in (b) in black and blue s_{4a}, $s_{4a} \oplus s_{4c}$, $s_{4a} \oplus s_{4c} \oplus s_{4d}$, $s_{4a} \oplus s_{4c} \oplus s_{4d} \oplus s_{4e}$, $s_{4a} \oplus s_{4c} \oplus s_{4d} \oplus s_{4e} \oplus s_{4f}$ and in green and red $s_{4a} \oplus s_{4b}$.

size comparing increasingly larger systems of 2,3,4 and 5 qubits and (ii) in the number of measurements of multiple systems with four qubits. In the second scaling, the occupation probabilities of all the systems, simulated with the same values of σ_R, contributes to gather information on the noise effects during the training of the ML models. In detail, we decide to use as input to the ML models the concatenation of the probabilities of the systems and, for two systems s_A and s_B, we indicate the latter with the notation $s_A \oplus s_B = \mathcal{P}_{1,A}, \ldots, \mathcal{P}_{2^n,A}, \mathcal{P}_{1,B}, \ldots, \mathcal{P}_{2^n,B}$. In both scaling, the procedure is always the same: 20 models are trained on each dataset through a 20 fold cross-validation. From the 20 predicted parameter values, the average value and the standard deviation can be calculated to include the variability of the models' predictions. Both analyses are performed with linear regression as baseline model and with ANNs. Regarding ANNs, they are trained for 150 epochs with the Adam optimizer. For a more in-depth discussion of the technical details related to model design and the choice of hyperparameters, refer to Section 6.1.

In the following, the ML models are trained and validated on the simulated data, and subsequently they are also tested on the real measurements. Using the simulated validation data, it is pos-

sible to monitor how the model is capable of generalization to unseen measurements. In this regard, we report in **Figure 2** the Mean Absolute Error (MAE) and Mean Relative Error (MRE) (respectively the two top and the two bottom plots) for the scaling (i) on the two left plots Figure 2a and (ii) on the two right plots Figure 2b. The errors are calculated comparing the predicted values of σ_R with the ground truth that, we recall, is the value of σ_R used to perform the simulation. Overall, we average the errors for all the samples of the validation set. Again, having 20 estimates (one for each model), we calculate mean value and standard deviation of the MAE and MRE to provide more robust results with associated uncertainty. In detail, we calculate the MAE as the average of the absolute values of the differences between the predictions of the models and the noise parameter values used during the simulation. Analogously, we calculate the MRE as the average of the absolute values of the differences divided by the values of σ_R. In order to avoid the divergence of the relative errors for extremely small values of σ_R, we consider in the test only values of $\sigma_R \geq 0.01$ ignoring a negligible part of the data. So far we have tested the protocol on simulated synthetic data. As previously mentioned, our goal is to estimate noise parameters by analyzing
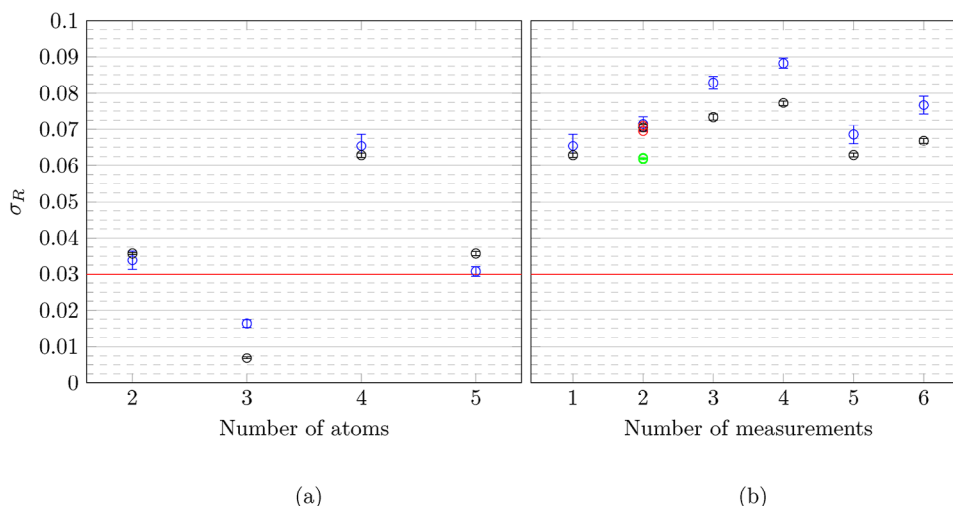
**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
TECHNOLOGIES**

www.advquantumtech.com

Figure 3. Predictions, on real data, of the value of σ_R for the models trained for the scaling in the number of atoms (a) and in the number of measurements (b) reported in Figure 2. We report the average values and standard deviations for the 20 linear regression (in black and green) and for the 20 ANN (in blue and red) models in the predictions of σ_R using a set of real measurements of the settings described in Table 3 run on the Pasqal NISQ devices. The models in (a) uses as input the measurements of s_2, s_3, s_{4a}, and s_5. The models in (b) uses as input one or more concatenated measurements of runs of the settings with four atoms (the fourth pair of points in (a) is equal to the first pair in (b)). We report in (b) in black and blue the incremental concatenation of s_{4a}, s_{4c}, s_{4d}, s_{4e}, and s_{4f}. In green and red we report the concatenation of s_{4a} and s_{4b}. The order of the real measurements for the latter concatenation is irrelevant, thus we report two green and two red points (almost overlapping and not clearly discernible) to consider the two possible concatenations. The horizontal red line indicates the value of 3%, estimated by Pasqal, for σ_R.

samples that come from a real QPU. Therefore, we now analyze real samples provided by Pasqal. We show in **Figure 3** (Figure 3a for the scaling (i) and Figure 3b for the scaling (ii)) the mean values and standard deviations along the 20 models of the predicted values of σ_R. In both Figures 2 and 3, the results of the training of linear regression models are depicted in black and the results of ANN in blue. Additionally, in Figures 2b and 3b we highlight in green for the linear regression and in red for the ANN a specific case: the concatenation of the measurements on two peculiar settings with four atoms, s_{4a} and s_{4b}, that have not only the same amount of atoms but also exactly the same topology. Therefore, the latter can be seen as a special case of the scaling (ii) where multiple measures of the same system are performed. Moreover, for the real measurements we consider both orderings $s_{4a} \oplus s_{4b}$ and $s_{4b} \oplus s_{4a}$ whose prediction results are reported with two couples of green and red points in Figure 3b (not clearly visible in the plot because they are almost overlapping).

As expected, the prediction error is decreasing with the number of atoms in the system because we get more information on the dynamic and thus on the noise influencing it. In Figure 2, we can also observe that ANN are in general more powerful respect to linear regression models (at the cost of more resource-intensive computations). In fact, the errors for the ANN models are always lower respect to the errors of linear regression models, and the difference between the ANN and linear regression errors is more pronounced increasing the number of atoms and measurements. This can be explained with a better capacity of ANNs to model complex dynamics.

Overall, comparing Figure 2a with five atoms and Figure 2b with number of measurements equal to two, seems to be more convenient to consider more measurements respect to increase the number of atoms of the setting. Also, comparing the green and red points with the black and blue ones for the same number

of measurements in Figure 2b, we observe that it can be slightly better to consider multiple measurements of the same setting with the same topology respect to collect measurements of a different setting with the same number of atoms.

We observe in Figure 3 that the values of σ_R predicted for the measurements of settings with two and five atoms are close to the estimated value of 3%, however the prediction for the setting of three atoms is lower, and the predictions for all the settings with four atoms, and concatenation of them, are around 7%. An explanation for this mismatch can be that the real data used for the experiments was collected when the device was still under development. Therefore, it is possible that the declared value of σ_R was not exactly corresponding to the noise effect in the prototype machine and even fluctuating between different runs. Moreover, the predictions consider only σ_R as a variable source for the noise, thus variations of the other noise parameters in the real machine influence the predictions of σ_R. Nevertheless, it is remarkable that the trained models have low standard deviations for the predictions that, even if this does not exclude an high bias error, still suggest a low variance error for the models. We can also observe that the order of the measurement for the settings s_{4a} and s_{4b} do not influence the predicted values – in fact the two green circles and the two red circles in Figure 3 are almost overlapping.

To summarize, noise estimation based on supervised learning is possible. The protocol we presented seems to suggest merging data from multiple similar registers instead of larger registers directly. This may be useful because of the difficulty in simulating larger systems. In addition, the estimates obtained are derived by averaging estimates from 20 models. Moreover, the associated standard deviation is small relative to the predicted value, so all the 20 models converge to very similar values.

**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
TECHNOLOGIES**

www.advquantumtech.com

Table 2. Predicted values on real data expressed as average and standard deviation of 20 models trained on cross-validation. The last column report for convenience the same estimated values of Table 1.

Parameter	Predicted value			Estimated value
σ_R	0.079	±	0.005	0.03
w	$122 \mu m$	±	6	$68 \mu m$
T	$56 \mu K$	±	4	$30 \mu K$
ε	0.082	±	0.010	0.03
ε'	0.078	±	0.005	0.08

Overall, we observe agreement between the predictions and the declared value for σ_R only on few cases. Therefore, it is reasonable to further investigate the reasons for this mismatch on future works when we can have access to more data from real NISQ devices. In fact, the real data used in this work came from a prototype and new analysis suggests lower values for the imperfections on next devices. The latter could possibly lead to an increment in the agreement between the predictions of the ML models and the expected noise values in future works.

Nevertheless, it is remarkable that, on simulated data, we can decrease the relative error from more than 45% for the estimations in the worst case (single measurement of a system with two atoms) to almost 15% using six measurements of a system with four atoms. Moreover, we have shown the interesting evidence that, in order to decrease the error, it is more beneficial to increase the number of measurements respect to increase the number of atoms.

3.2. Multiple Parameters Noise Characterization

In this section, we train a deep learning model in a multioutput regression setting to estimate the values of all the noise parameters in Table 1. We simulated a dataset of 54 000 labeled samples for the six-qubit system whose topology can be observed in Figure 1b. The used pulse sequence, that defines the dynamics, is shown in Figure 1a. Analogously to the scaling experiments in the previous section, the measurement for each simulation is obtained sampling 500 runs. The values used in the simulations for each parameter are: $\sigma_R = \mathcal{U}(0, 0.15)$, $w(\mu m) = \mathcal{U}(0, 200)$, $T(\mu K) = \mathcal{U}(0, 100)$, $\varepsilon = \mathcal{U}(0, 0.15)$ and $\varepsilon' = \mathcal{U}(0, 0.15)$.

After finding the best set of hyper-parameters, 20 models are trained using the cross-validation procedure to exploit the entire dataset and to obtain the standard deviations of the predictions. Each one of the 20 models is trained with early stopping for a maximum of 150 epochs. Further technical details related to ANNs design and hyperparameter optimization can be found in Section 6.2.

In **Table 2**, we show the resulting estimation of the main noise factors. Each reported value is the average of the 20 models, trained on different splits, with the corresponding standard deviation. We observe that the predicted values do not match those estimated by Pasqal, although all 20 models always converge to very similar values of the predictions. In this regard, the same considerations expressed in Section 3.1 are also valid for multi-parameter estimation: i.e., that the parameter predictions can

incorporate other neglected effects (noise sources, influence of other neighboring atoms, etc.). Another possible factor could be that the measurements came from a prototype NISQ, just as in the case of those used in Section 3.1. Therefore, we can expect more agreement in the future as a result of technical improvements. Nevertheless, this inconsistency represents an opportunity to further investigate the sources of this mismatch when we will have access to more data from the real NISQ devices. Finally, it is worth noting that, even if for the experiments in this section the setting and the pulse are different to the ones used in Section 3.1, the predicted value for σ_R is comparable to the ones obtained for the estimation of the same parameter in the settings with four atoms previously illustrated.

3.3. Error Correction

Many techniques have been developed in the theory of classical error-correcting codes.[25,26] The key idea on which they are based is mainly redundancy. Nonetheless, the addition of redundancy is not immediate in NISQ devices because of the no cloning theorem.[27] However, some sort of redundancy can be achieved in quantum devices by expanding the system to more qubits.[28] In fact, all the most used quantum error correction techniques require the use of more qubits than the ones strictly necessary for the computation[29] but, at present time, it is not feasible with NISQ devices. Therefore, we propose to verify that it is possible to mitigate the effects of quantum noise without extra qubits through the use of RL techniques. RL is a ML area where an agent learns which actions to perform in order to maximize a reward.[17] Schematically, we can say that this is a closed-loop problem because the actions of the learning system influence subsequent inputs. In addition, the learner does not know a priori which action to perform and has to find out for himself, through trials and errors, which actions lead to larger rewards. Actions can influence not only the immediate reward but also future rewards. RL, unlike Supervised Learning, does not require labelled input-output pairs, but focuses on finding a balance between exploration of the actions space in an environment and exploitation of the acquired knowledge. The agent must exploit what it already knows in order to obtain reward, but it must also explore in order to make better action selections in the future. The trade-off is that neither exploration nor exploitation can be exclusively pursued without failing in the task. The agent must try a variety of actions and progressively favor those that seem to be the best. Any problem of learning goal-oriented behavior can be reduced to three signals that are exchanged between an agent and its environment: a signal to represent the choices made by the agent (the actions), a signal to represent the basis on which the choices are made (the states) and a signal to define the agent's goal (the rewards). In detail, for each action of the agent at timestep n, its effects on the environment are quantified by a reward r_n. Then the objective of the training is to maximize the discounted cumulative reward $R_{n_0} = \sum_{n=n_0}^{\infty} \gamma^{n-n_0} r_n$, where the discount $\gamma \in (0, 1)$ is an hyperparameter that controls the importance of rewards far in the future respect to the ones immediately after n_0. This objective is implemented with the idea that, if we would have a function $Q^* : State \times Action \rightarrow \mathbb{R}$ that given a state and an action performed over that state returns the

2300192 (6 of 10)
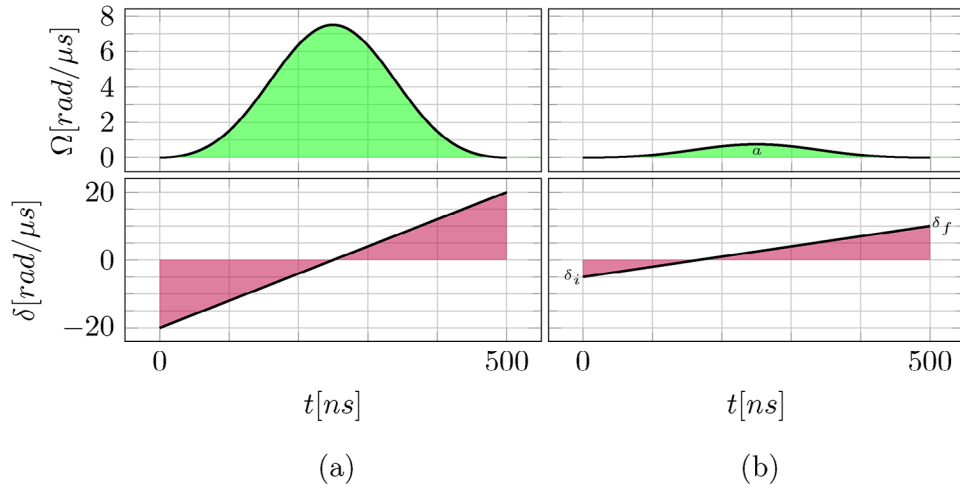
Figure 4. Standard pulse P (a) to be corrected with a correction pulse P' (b) to be added after P to counteract the effects of the noise. The Rabi frequency Ω is depicted in green and the detuning δ in purple. P is a pulse of duration $\tau = 500 ns$, Gaussian Rabi profile with area equal to $\pi/2$ rad and detuning in the form of a ramp from $\delta(0) = -20$ rad/μs and $\delta(\tau) = 20$ rad/μs. P' is a pulse with the same duration and characteristics of P but with variable Rabi area a rad, initial detuning δ_i rad/μs and final detuning δ_f rad/μs.

cumulative discounted reward, then the policy can be implemented with $\pi^*(s) = \arg\max_a Q^*(s, a)$. In general, Q^* is unknown and is approximated by a neural network. For a defined policy π, the Q function obeys the Bellman equation $Q^\pi(s, a) = r + \gamma Q^\pi(s', \pi(s'))$ where r and s' are respectively the reward and the next state obtained after the action a on the state s. The neural network that defines Q, and then the agent, is trained minimizing over a batch of transitions the Huber loss $\mathcal{L}(\xi)$ of the temporal difference error $\xi = Q(s, a) - (r + \gamma \max_a Q(s', a))$.

We choose to correct the standard impulse P depicted in **Figure 4**a applied to a single qubit. P has a Gaussian profile in the Rabi frequency $\Omega(t)$ of duration $\tau = 500$ ns and area $\pi/2$ rad. P also has a ramp profile in detuning $\delta(t)$ of duration $\tau = 500$ ns with $\delta(0) = -20$ rad/μs and $\delta(\tau) = 20$ rad/μs. The chosen approach to correct the noise is to apply the correction pulse Figure 4b to be placed after the pulse to be corrected and having the same characteristics and length of $\tau = 500$ ns. In detail, we choose a Gaussian profile in the Rabi frequency with variable area a rad and a ramp profile in detuning $\delta(t)$ with variable initial $\delta(0) = \delta_i$ and final $\delta(\tau) = \delta_f$. In such a way, the final atoms occupation probabilities with the application of the corrected pulse $\boldsymbol{P}^{noisy}_{P+P'}$ and after the ideal pulse $\boldsymbol{P}^{ideal}_P$ are *closer* than $\boldsymbol{P}^{noisy}_P$ and $\boldsymbol{P}^{ideal}_P$. By the notation $\boldsymbol{P}^i_j$ we denote the measurement $\boldsymbol{P}$ obtained after running a simulation with the pulse j with or without noise (respectively, $i = noisy$ or $i = ideal$). The training allows to find the three optimal parameters a, δ_i, and δ_f for the correction impulse P'.

In our RL framework, the state is represented by the occupation probabilities that are estimated from the average of ten independent noisy simulations whose probabilities are extracted from the amplitudes of 25 quantum states uniformly sampled along the simulated dynamic. At the beginning of each episode we choose $a = \pi/20$ and $\delta_i = \delta_f = 0$, and they can have values in the ranges $a \in [0, \pi/2]$ and $\delta_i, \delta_f \in [-20, 20]$. The agent, implemented with an ANN that have an input layer of 50 units

(two basis for each one of the 25 intermediate states), two ReLU hidden layer of 128 neurons and an output layer of six neurons, selects one among four possible actions: $a^n = a^{n-1} + \Delta a$, $a^n = a^{n-1} - \Delta a$, $\delta_i^n = \delta_i^{n-1} + \Delta\delta_i$, $\delta_i^n = \delta_i^{n-1} - \Delta\delta_i$, $\delta_f^n = \delta_f^{n-1} + \Delta\delta_f$, $\delta_f^n = \delta_f^{n-1} - \Delta\delta_f$. We choose fixed values for $\Delta a = \pi/200$ and $\Delta\delta_i = \Delta\delta_f = 0.2$. Each episode is constituted of a series of steps at increasing values of n. For each step, the chosen action is applied and a correction impulse P'_n, characterized by a^n, δ_0^n, and δ_f^n, is generated and used in a new simulation obtaining a new probability vector $\boldsymbol{P}^{noisy}_{P+P'_n}$ for the final quantum state of the corrected noisy simulation. Afterwards, the reward r_n is obtained before proceeding with the next step. The episode ends when the action causes a, δ_0, or δ_f to go out of boundaries or after 100 steps. The reward is defined as:

$$
r_n = \begin{cases} 1 & \text{if} \quad \left| \boldsymbol{P}^{noisy}_{P+P'_n} - \boldsymbol{P}^{ideal}_P \right|_1 < \left| \boldsymbol{P}^{noisy}_{P+P'_{n-1}} - \boldsymbol{P}^{ideal}_P \right|_1 \\ 0 & \text{otherwise,} \end{cases} \quad (2)
$$

where $|\cdot|_1$ is the ℓ_1 norm. Specifically, the reward is *one* if the last action at step n makes the corrected noisy simulation closer to the ideal one respect to the previous step $n - 1$ and *zero* otherwise. During the training, we monitor the KullbackLeibler (KL) divergence between $\boldsymbol{P}^{noisy}_{P+P'_n}$ and $\boldsymbol{P}^{ideal}_P$:

$$
D_{KL}(\boldsymbol{P}^{noisy}_{P+P'_n}, \boldsymbol{P}^{ideal}_P) = \sum_{i=1}^2 \left(\boldsymbol{P}^{noisy}_{P+P'_n} \right)_i \log\left(\frac{\left(\boldsymbol{P}^{noisy}_{P+P'_n} \right)_i}{\left(\boldsymbol{P}^{ideal}_P \right)_i} \right) \quad (3)
$$

that is averaged for all the steps n within each episode. The evolution of the averaged KL divergence for the 1 000 training episodes is reported in **Figure 5**, where we can observe that it effectively decreases below the reference value of $D_{KL}(\boldsymbol{P}^{noisy}_P, \boldsymbol{P}^{ideal}_P) = 0.0011$

**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
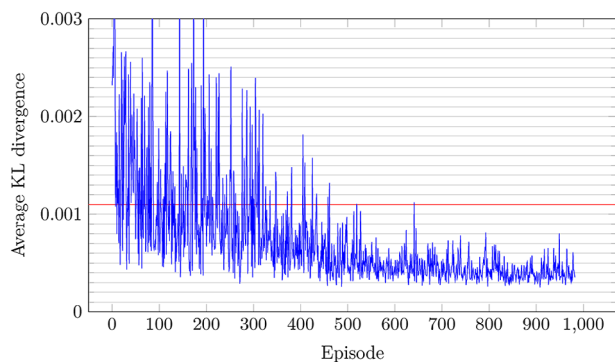TECHNOLOGIES**

www.advquantumtech.com

Figure 5. Evolution of the KL divergence between the corrected noisy simulation and the ideal one averaged for each episode. The red line is the reference value of 0.0011 for the KL divergence between the uncorrected noisy simulation and the ideal one averaged over 100 simulations.

reported with the red line and calculated with the average for 100 noisy simulations without the correction pulse.

4. Conclusion

We presented two applications of ML to the context of quantum noise characterization and correction. To characterize the noise, we collected a dataset of multiple simulated noisy measurement of different settings in Pasqal quantum machines to train ML models, and we test them on real data. For the noise correction, we trained a RL model to find a correction pulse to counteract the effects of the noise affecting a simulated test setting. Regarding the noise characterization, we compared ANN with linear regression models in predicting the value of the laser intensity fluctuation σ_R scaling the number of qubit in the register and the number of measurements of the system. We found that ANNs perform better than linear regression and that the model accuracies increases both with the number of qubits and with the number of measurements. Moreover, we have insights that, in order to better characterize the noise parameters, it is more effective to increase the number of measurements respect to the number of qubits. When we tried to predict the noise parameters on real NISQ devices, we found that, for every set of measurement, 40 different models (ANN and linear regression trained independently in a 20 fold cross validation setting) agree on the predictions, and therefore, the variance error is low. Finally, we trained 20 ANN models in a multiregression setting to predict five different noise parameter values, and, also in this case, the models are in agreement between them when tested on real data. Unfortunately, when we compare the predictions of the noise parameter using measurements of real NISQ devices with the declared values of such parameters, we do not find strong agreement for every measurement. The data was acquired on a NISQ device that was still under development and with high values for the imperfections. Later analysis suggests that the new devices are more robust to errors and noise sources. Therefore, we would like to investigate in future works the ML models predictive capabilities on real data from future Pasqal devices when they will be available. Besides, when it will be possible to gather numerous amount of data on real machines, we would like to train the models with a training set containing also real data.

Regarding the noise correction, the proposed approach successfully learns to correct a simulated noisy pulse and to make the measured probabilities closer to the ideal ones.

We believe that the results presented in this work can be used to better quantify the effects of the noise affecting the Pasqal, and in general neutral atoms, NISQ devices and to counteract those effects. The presented tecniques are dependent on the atoms topology and the pulse shape. Thus, the ML models can be trained to characterize and correct the noise of single quantum gates that compose more complex Hamiltonians. It is important to note that the accuracy of the predicted noise parameters depends on the accuracy of the simulation and in particular on the accuracy of the simulator noise model.

In previous works,[11,13,30,31] and in preliminary experiments using Pasqal simulator, there is an evidence of the improvement of the noise characterization when more temporal statistics are collected. We adopted this strategy in this paper for the noise correction, where the occupation probabilities are obtained from the amplitudes of the intermediate quantum states sampled at regular steps within the simulated dynamic. However, in real NISQ devices, intermediate measurements of the dynamic are less straightforward because of the impossibility of observing a system without changing it. We can obtain the same effect independently measuring incremental subdynamics from $t = 0$ to subsequent time steps of the full dynamic. To implement this approach on Pasqal machines, we can design a full pulse that is subsequently split in sub-pulses at times $[t_0, t_1], [t_1, t_2], \ldots, [t_{m-1}, t_m]$. The measurements at time t_k for $k = 1, \ldots, m$ can then be obtained by initialising the register always to the same initial setting, and performing the computation considering the effects of all the sub-pulses spanning the times $[t_0, t_k]$ from the first to the one before t_k. The ML models can then process all the measurements obtained at times $t_1, \ldots, t_k$ and, in that way, we expect to obtain better results for the characterization of the noise. Moreover, we can also use ANNs more suitable for data organized in temporal sequences, i.e., Recurrent Neural Networks (RNNs).

Finally, in the context of Quantum Machine Learning (QML),[32,33] our work is framed as a classical ML approach to process quantum data. Future research lines may include the design of QML models for the noise characterization and correction implemented directly within the quantum dynamic of neutral atoms devices or of other NISQ devices. For instance, pattern matching QML techniques[34] can be adapted for the identification of noise patterns[13] that are specific to the neutral atoms dynamics.

5. Experimental Section

Single Parameter Characterization: In this section, is described with more details the topologies of the analyzed quantum systems and the ANN models used in Section 3.1.

The registers, summarized in **Table 3**, have an incremental number of atoms from 2 to 5 and some of them are chosen in a way such that the positions of the atoms of every register are included in the subsequent ones as far as possible.

To be precise, with the notation $s_k \subset s_{k'}$ and $k < k'$, is indicated that the quantum register of $s_{k'}$ is the same as that of s_k with the addition of an atom and that, therefore, the coordinates of the atoms in common were the same. In detail, the setting with five atoms (s_5) had atoms in the same positions of the ones of the settings of dimensionality four (s_{4a}) and three (s_3) plus extra atoms in other positions. Moreover, s_{4a} contains all the two atoms of the setting

**ADVANCED
SCIENCE NEWS**

www.advancedsciencenews.com

**ADVANCED
QUANTUM
TECHNOLOGIES**

www.advquantumtech.com

Table 3. Quantum systems used for single parameter estimation σ_R. By the notation s_i we denote the system formed by i atoms. In the case of four atoms, having used six different systems for the spatial arrangement of atoms, we use an additional subscript s_{4j}, with $j = \{a, b, c, d, e, f\}$. Also, when a quantum register of a system s_k is entirely contained in the quantum register of a larger system s'_k, with $k' > k$, we use the notation $s_k \subset s_{k'}$.

Name	s_2	s_3	s_{4a}	s_5	s_{4b}	s_{4c}	s_{4d}	s_{4e}	s_{4f}
Atoms	2	3	4	5	4	4	4	4	4
Properties	$\subset s_{4a}$	$\subset s_5$	$\subset s_5$		$= s_{4a}$				

s_2, but s_3 included only one of the two atoms of s_2 and s_{4a} only two of the three atoms of s_3. The latter properties were denoted with $s_2 \subset s_{4a} \subset s_5$ and $s_3 \subset s_5$. In addition, further measurements of settings with four atoms were also collected. In detail, a second setting s_{4b} with the atoms in the same position of s_{4a} and other four settings with different positions for the atoms: s_{4c}, s_{4d}, s_{4e}, and s_{4f} were run. Those specific settings were chosen to evaluate the two different scaling: i) in the number of atoms, ii) in the number of measurements of different settings with the same number of atoms. Specifically, we consider for (i) s_2, s_3, s_{4a}, and s_5 and for (ii) s_{4a}, s_{4b}, s_{4c}, s_{4e}, and s_{4f}.

The trained ANNs were composed by a single hidden layer of 100 neurons with ReLU activation function and the output layer with a single neuron with sigmoid activation function. The targets were normalized between 0 and 1 before the training and the inverse transformation was applied to calculate the prediction error. The models were developed in *PyTorch*[35] and trained with mini batch gradient descent to minimize the L1 loss using the Adam optimizer[36] with learning rate 0.001 and batch size 512. All models were trained with early stopping for a maximum of 150 epochs.

To perform the scaling (i), we trained four different models using as inputs the 2^2, 2^3, 2^4, and 2^5 measurements of respectively the settings s_2, s_3, s_{4a}, and s_5. To perform the scaling (ii), we considered the measurement coming from the following systems:

- s_{4a} ($2^4 = 16$ measurements)
- $s_{4a} \oplus s_{4b}$ ($2^4 + 16 = 2^5$ measurements)
- $s_{4a} \oplus s_{4b} \oplus s_{4c}$ ($2^4 + 32 = 48$ measurements)
- $s_{4a} \oplus s_{4b} \oplus s_{4c} \oplus s_{4d}$ ($2^4 + 48 = 64$ measurements)
- $s_{4a} \oplus s_{4b} \oplus s_{4c} \oplus s_{4d} \oplus s_{4e}$ ($2^4 + 64 = 80$ measurements)
- $s_{4a} \oplus s_{4b} \oplus s_{4c} \oplus s_{4d} \oplus s_{4e} \oplus s_{4f}$ ($2^4 + 80 = 96$ measurements)

In all the cases, the datasets were split in 20 equal parts to perform a 20 fold cross-validation and it reported the resulting average mean absolute error and its standard deviation for the 20 models.

As a remark, the Pulser simulator allowed to specify the number of samples per run to speedup the computation. In that case, for each run, the final quantum state was preliminary calculated and then the specified number of measurements was obtained from such state. Even if this expedient was useful to spare resources, it was found in preliminary experiments that this speedup was counter-productive in the context of noise estimation. In fact, for all the samples of one run, the Hamiltonian defining the evolution was always the same and also the noise that influences it. For this reason, in this work the number of samples per run were kept equal to one forcing the resampling of the noise at each single measurement.

Moreover, in this context, was better to consider more measurements respect to increase the number of atoms of the setting. In detail, considering both subfigures, the number of data points for the measurement of the setting with five atoms in Figure 2a, i.e. $2^5 = 32$, is equal to the ones for two concatenated measurements of settings with four atoms in Figure 2b, i.e., $2^4 + 2^4 = 32$, but the error in the latter case is lower than the former.

5.0.0.1. Multiple Parameters Characterization: Before training the models, the noise parameters were normalized between 0 and 1 to avoid uneven prediction error during the loss calculation. The models were implemented in Pytorch[35] and were trained with minibatch gradient descent to minimize the L1 loss using Adam.[36] Regarding the architecture of the models, the ANN was a Multi Layer Perceptron (MLP) with the ReLU activation function for all the hidden layers and the sigmoid activation function for the last layer. The best combination of number of neuron layers, number of neurons in each layer, batch size, and learning rate was chosen with an hyper-parameter optimization procedure. The latter was implemented using the python library Ray Tune[37] with the ASHA scheduler[38] and the HyperOpt search algorithm.[39] The ASHA scheduler allowed multiple models to be trained in parallel iteratively interrupting the training of the least promising one and thus reducing the duration of the hyper-parameter optimization. In this case, at each epoch, it halved the models by discarding those with the highest calculated loss on the validation set. HyperOpt search algorithm, on the other hand, choosed the most probable best combinations of hyper-parameters based on the previously trained and/or stopped models. By this procedure, the model with the most promising set of hyper-parameters was chosen from 1000 models trained with the Adam optimizer. The hyper-parameters were sampled in the following ranges: number of hidden layers from 1 to 100, number of neurons in each layer from 5 to 200, batch size in $\{2, 4, 8, 16, 32\}$ and learning rate from $loguniform(10^{-4}, 10^{-1})$. At the end, the best hyper-parameters combination was: one hidden layer of 117 neurons, batch size 16, initial learning rate ≈ 0.069, dropout probability ≈ 0.044 and L2 regularization ≈ 0.0002.

After finding the best set of hyper-parameters, 20 models were trained using the cross-validation procedure to exploit the entire dataset and to obtain the standard deviations of the predictions. In detail, for the cross-validation the dataset was divided into 20 equal parts, 18 were used for training, one for validation and one for testing. The advantage of using the cross-validation procedure was that a different block was used for the test of each model, and also, in this way all the samples of the dataset were exploited for the training. Each one of the 20 models was trained with early stopping for a maximum of 150 epochs.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

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

[1] J. Preskill, *Quantum* **2018**, *2*, 79.

[2] W. Cai, Y. Ma, W. Wang, C.-L. Zou, L. Sun, *Fundam. res.* **2021**, *1*, 50.

[3] Z. Chen, K. Satzinger, J. Atalaya, A. Korotkov, A. Dunsworth, D. Sank, C. Quintana, M. McEwen, R. Barends, P. Klimov, S. Hong, C. Jones, A. Petukhov, D. Kafri, S. Demura, B. Burkett, C. Gidney, A. Fowler, A. Paler, J. Kelly, *Nature* **2021**, *595*, 383.

[4] D. Lidar, T. Brun, *Quantum Error Correction*, Cambridge University Press, **2013**.

[5] Programmable atomic arrays - PASQAL, https://www.pasqal.com/, [Online; accessed 14-2-2023].

[6] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, C. Jurczak, *Quantum* **2020**, *4*, 327.

[7] H. Silvério, S. Grijalva, C. Dalyac, L. Leclerc, P. J. Karalekas, N. Shammah, M. Beji, L.-P. Henry, L. Henriet, *Quantum* **2022**, *6*, 629.

[8] S. Shalev-Shwartz, S. Ben-David, *Understanding machine learning: From theory to algorithms*, Cambridge university press, **2014**.

[9] Y. B. Ian Goodfellow, A. Courville, *Deep learning*, MIT press, Cambridge **2016**.

[10] A. Youssry, G. A. Paz-Silva, C. Ferrie, *Npj Quantum Inf.* **2020**, *6*, 1.

[11] S. Martina, S. Gherardini, F. Caruso, *Phys. Scr.* **2023**, *98*, 035104.

[12] D. F. Wise, J. J. Morton, S. Dhomkar, *PRX Quantum* **2021**, *2*, 010316.

[13] S. Martina, L. Buffoni, S. Gherardini, F. Caruso, *Quantum Mach. Intell.* **2022**, *4*, 1.

[14] P. Baireuther, T. E. O'Brien, B. Tarasinski, C. W. Beenakker, *Quantum* **2018**, *2*, 48.

[15] A. G. Fowler, M. Mariantoni, J. M. Martinis, A. N. Cleland, *Phys. Rev. A* **2012**, *86*, 032324.

[16] R. Harper, S. Flammia, J. Wallman, *Nat. Phys.* **2020**, *16*, 1.

[17] R. S. Sutton, A. G. Barto, *Reinforcement learning: An introduction*, MIT press, Cambridge **2018**.

[18] Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, P. Mundada, T. Merkh, A. Carvalho, M. Biercuk, *PRX Quantum* **2021**, *2*, 040324.

[19] A. J. Goldschmidt, J. L. DuBois, S. L. Brunton, J. N. Kutz, *Quantum* **2022**, *6*, 837.

[20] R. Porotti, A. Essig, B. Huard, F. Marquardt, *Quantum* **2022**, *6*, 747.

[21] R. Sweke, M. S. Kesselring, E. P. van Nieuwenburg, J. Eisert, *Mach. Learn.: Sci. Technol.* **2020**, *2*, 025005.

[22] L. D. Colomer, M. Skotiniotis, R. Muñoz-Tapia, *Phys. Lett. A* **2020**, *384*, 126353.

[23] K.-N. Schymik, V. Lienhard, D. Barredo, P. Scholl, H. Williams, A. Browaeys, T. Lahaye, *Phys. Rev. A* **2020**, *102*, 063107.

[24] S. De Léséleuc, D. Barredo, V. Lienhard, A. Browaeys, T. Lahaye, *Phys. Rev. A* **2018**, *97*, 053803.

[25] R. W. Hamming, *Bell Syst. Tech. J.* **1950**, *29*, 147.

[26] W. W. Peterson, W. Peterson, E. J. Weldon, E. J. Weldon, *Error-correcting codes*, MIT press, Cambridge **1972**.

[27] W. K. Wootters, W. H. Zurek, *Nature* **1982**, *299*, 802.

[28] P. W. Shor, *Phys. Rev. A* **1995**, *52*, R2493.

[29] J. Roffe, *Contemp. Phys.* **2019**, *60*, 226.

[30] S. Martina, S. Hernndez-Gmez, S. Gherardini, F. Caruso, N. Fabbri, *Mach. Learn.: Sci. Technol.* **2023**, *4*, 02LT01.

[31] F. F. Fanchini, G. Karpat, D. Z. Rossatto, A. Norambuena, R. Coto, *Phys. Rev. A* **2021**, *103*, 022425.

[32] V. Dunjko, J. M. Taylor, H. J. Briegel, *Phys. Rev. Lett.* **2016**, *117*, 130501.

[33] E. Aïmeur, G. Brassard, S. Gambs, in *Advances in Artificial Intelligence*, (Eds.: L. Lamontagne, M. Marchand), Springer, Berlin, Heidelberg **2006**, pp. 431–442.

[34] S. Das, J. Zhang, S. Martina, D. Suter, F. Caruso, *Quantum Mach. Intell.* **2023**, *5*, 16.

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., *Advances in neural information processing systems* **2019**, *32*, 8024.

[36] D. P. Kingma, J. Ba, *arXiv preprint arXiv:1412.6980* **2017**.

[37] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, I. Stoica, *arXiv preprint arXiv:1807.05118* **2018**.

[38] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, J. Ben-Tzur, M. Hardt, B. Recht, A. Talwalkar, *Proc. Mach. Learn. Syst.* **2020**, *2*, 230.

[39] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, in *Comput. Sci.Sci. Discov.*, **2015**, *8*, 014008.