

ARTICLE OPEN



Experimental graybox quantum system identification and control

Akram Youssry¹, Yang Yang¹, Robert J. Chapman^{1,2}, Ben Haylock^{3,4}, Francesco Lenzi^{3,5}, Mirko Lobino^{3,6,7} and Alberto Peruzzo^{1,8}✉

Understanding and controlling engineered quantum systems is key to developing practical quantum technology. However, given the current technological limitations, such as fabrication imperfections and environmental noise, this is not always possible. To address these issues, a great deal of theoretical and numerical methods for quantum system identification and control have been developed. These methods range from traditional curve fittings, which are limited by the accuracy of the model that describes the system, to machine learning (ML) methods, which provide efficient control solutions but no control beyond the output of the model, nor insights into the underlying physical process. Here we experimentally demonstrate a ‘graybox’ approach to construct a physical model of a quantum system and use it to design optimal control. We report superior performance over model fitting, while generating unitaries and Hamiltonians, which are quantities not available from the structure of standard supervised ML models. Our approach combines physics principles with high-accuracy ML and is effective with any problem where the required controlled quantities cannot be directly measured in experiments. This method naturally extends to time-dependent and open quantum systems, with applications in quantum noise spectroscopy and cancellation.

npj Quantum Information (2024)10:9; <https://doi.org/10.1038/s41534-023-00795-5>

INTRODUCTION

Quantum technology promises to deliver exponentially faster computation, provably secure communications, and high-precision sensing¹. However, during the fabrication and operation of a quantum device, there are many factors that can significantly impact its functionality, requiring characterization and control techniques to achieve high-level performance. Generally, we are interested in uncovering the unknown relation between the control and the Hamiltonian governing the device, and then utilizing this information to drive the device toward a desired target. Typical targets include unitary gates, a specific Hamiltonian, or certain output probability distributions.

Approaches that directly aim to control the quantum device without first identifying it, includes dynamical decoupling and dynamically corrected gates^{2–5}, as well as direct gradient-based optimization, such as the commonly used GRAPE algorithm⁶ and its variants^{7–14}. These techniques only work when the dependence of the Hamiltonian on the control is known, because they are based on optimizing the fidelity to some target with respect to control. In situations where this dependence is unknown, the fidelity (and, in general other cost functions) and/or its gradient, can be computed iteratively from experimental data. The control is optimized after each iteration and directly applied to the system for the next iteration. The physical system becomes part of a feedback architecture for designing the pulses without a need for a model. Examples of this approach are in^{14–17}, and are sometimes referred to as ‘learning quantum control’¹⁸. Evolutionary algorithms, such as Genetic Algorithm (GA) have also been proposed¹⁹, as optimization techniques with the advantage of

being gradient-free and are more likely to find global minima. These techniques can also be applied with a known model or directly from experimental measurements. Reinforcement learning methods^{20–23}, are also model-free and have been recently explored for the purposes of removing the reliance on assumptions on the physical system and have been successfully applied to controlling quantum systems.

In many situations, it is important to reconstruct a mathematical model of the system from experimental data, a process referred to as ‘system identification’. An identified model can be used to compare the behavior of a fabricated device to its design, or to understand the underlying noise process affecting the system. As the model can predict the behavior of the system, it can be used for control as well.

The traditional approach to characterizing and controlling physical devices is based on theoretical models of the underlying processes governing the relationship between input and output signals. Example of fitting data to a physical model include^{24,25}. These ‘whitebox’ (WB) models are based on parameter estimation via curve fitting and can be computationally expensive, inaccurate, or incomplete. For example, they do not consider unexpected input parameters or dynamics. Moreover, the models commonly used to describe the dynamics of an open quantum system (such as Lindblad’s master equation), is valid only under strict assumptions and approximations of the noise (such as Markovianity) and the control (such as being ideal impulses). These assumptions do not hold for many quantum platforms, making the use of fixed a priori WB models inaccurate.

¹Quantum Photonics Laboratory and Centre for Quantum Computation and Communication Technology, RMIT University, Melbourne, VIC 3000, Australia. ²ETH Zurich, Optical Nanomaterial Group, Institute for Quantum Electronics, Department of Physics, 8093 Zurich, Switzerland. ³Centre for Quantum Computation and Communication Technology (Australian Research Council), Centre for Quantum Dynamics, Griffith University, Brisbane, QLD 4111, Australia. ⁴Institute for Photonics and Quantum Sciences, SUPA, Heriot-Watt University, Edinburgh EH14 4AS, United Kingdom. ⁵Institute of Physics, University of Muenster, 48149 Muenster, Germany. ⁶Department of Industrial Engineering, University of Trento, via Sommarive 9, 38123 Povo, Trento, Italy. ⁷INFN-TIFPA, Via Sommarive 14, I-38123 Povo, Trento, Italy. ⁸Qubit Pharmaceuticals, Advanced Research Department, Paris, France. ✉email: alberto.peruzzo@rmit.edu.au

To increase accuracy and remove some of the limitations of the WB method, supervised machine learning techniques are proving useful for modeling and controlling complex physical systems. In particular, techniques such as neural networks are, in many cases, superior if not the only viable approach. However, this approach, referred to as 'blackbox' (BB), does not provide any information about the underlying physics of the system. Nonetheless, it has been used in many applications such as quantifying Non-Markovianity of quantum systems²⁶, characterizing qubits and environments^{27–30}, quantum control^{31–33}, quantum error correction^{34,35}, optimization of experimental quantum measurements^{36,37}, and calibration of quantum devices³⁸. A closely related approach to system identification is the 'Hamiltonian Learning' problem^{39,40}, where a fixed time-independent Hamiltonian (control is fixed) is inferred from quantum measurements.

In situations where the quantities of interest, such as Hamiltonians, unitaries, and noise operators, cannot be directly accessed from the model, the use of a hybrid WB-BB or 'graybox' (GB) approach allows for both identify and control of these quantities beyond the measurable dataset. Following the standard control engineering definition⁴¹, the aim of GB models is to merge an abstract mathematical structure, such as a neural network, with physical laws. While direct control methods, such as GRAPE, are sometimes referred to as grayboxes, because they are data-driven and may rely on prior knowledge, these methods do not identify a useful mathematical description of the system. The graybox architecture provides access to any physical quantity available from the physical part (WB) of the model.

GB has been proposed to model electrical drift in quantum photonic circuits⁴², as well as open quantum systems subject to classical⁴³ and quantum noise⁴⁴, covering the case of a time-dependent Hamiltonian problem. The GB model was also applied in the context of noise detection in the presence of a spectator qubit that acts as a sensor of the environment⁴⁵, and to geometric quantum gate synthesis⁴⁶. While GB has been used experimentally to characterize superconducting qubits⁴⁷, to date, no quantum device has been characterized using a GB model where the identified model was then used to design optimal quantum control.

Here, we experimentally demonstrate how to model and control a quantum device using the GB architecture when the Hamiltonian dependence on the control is unknown. We report high-fidelity preparation of arbitrary unitaries and output probability distributions of a reconfigurable three-mode integrated photonic device and uncover the Hamiltonian dependence on the control. Our GB approach outperforms the traditional model fitting methods and can successfully prepare unitary operations, which are not accessible from the structure of a BB. Our results show a promising approach to enhance quantum control by understanding the physical processes, and open the way to improve the engineering of quantum devices.

RESULTS

Modeling quantum devices

We consider the class of quantum devices, shown in Fig. 1a, described by a time-independent Hamiltonian undergoing a closed-system evolution (i.e., in the absence of quantum noise). Photonic devices are examples of this class when the Hamiltonian is not modulated faster than the evolution time of a propagating photon. We focus on the case where the system is described by a finite N -dimensional Hilbert space (i.e., a qudit). The Hamiltonian governing the dynamics of the system can be represented in the most general form as an $N \times N$ complex Hermitian matrix that depends on a set of external controls. We encode the set of controls in a $M \times 1$ -dimensional vector $\mathbf{V} = [V_1, V_2, \dots, V_M]^T$, where V_k is the k th control. We assume that during the system evolution,

the control vector is fixed. An example of such controls is the voltages applied to a reconfigurable photonic chip, schematically shown in Fig. 1b. The system starts in an initial state $|\psi_0\rangle$ and evolves to the state $|\psi_T\rangle$ at time $t = T$. The state is then measured on some basis to obtain a set of probability outcomes. A more general form of time-dependent evolution in the presence of unwanted interactions with the environment has been considered in our previous work⁴³.

Our aim is to obtain an ML model that describes the behavior of the device given a set of controls \mathbf{V} and use it for controlling the device. The input to the ML model is the M -dimensional control vector \mathbf{V} , while the output is the set of measured outcomes of the state after evolution. Generally, it is required to have informationally-complete measurements to fully characterize a quantum system. Here, we restrict the initial states as well as the measurement basis to the set of computational basis states $\{|0\rangle, |1\rangle, \dots, |N\rangle\}$ in order to be compatible with our experimental setup. For each of these N -initial states, we have N possible outcomes with an associated probability $P_{j \rightarrow k}$ corresponding to the j th input and k th output, giving a total of N^2 outputs. The approach, however, is independent of this choice, and any set of states can be used. In Supplementary Note 4, we discuss a more general measurement scheme. It is important to emphasize that the model input is the set of controls \mathbf{V} applied to the system, and not the initial state $|\psi_0\rangle$.

Graybox architecture

Our starting point is our theoretical proposal⁴² for modeling and controlling quantum photonic circuits using a GB architecture. The work aimed at stabilizing the effect of electrical drift and preparing quantum gate sequences at the same time. In order to model such an effect, a GB was designed to capture variations over a 'classical' time scale (i.e. slower than the evolution time of a single photon). So, a recurrent neural network was used, particularly a Gated-Recurrent Unit (GRU) as the black part of the model. The inputs and outputs of the model are slowly time-varying waveforms. However, this resulted in optimal voltage pulses that did not belong to the class of pulses in the training set, which is not available in our experimental device, and may not, in general, be available.

Here, we focus only on modeling the unknown Hamiltonian-voltage dependence, and stabilize the drift in hardware using fast pulsing, a well-known technique in integrated photonics. The pulses have fixed frequency and duty cycle, so the only controllable parameters are the amplitude on each electrode. This makes it possible to restrict the controller solution to the space of training pulses. Therefore, in what follows in this paper, we use standard feed-forward neural networks as opposed to recurrent neural networks, as there is no need to model a sequence over time.

The GB structure we propose, shown in Fig. 1c, consists of a BB (in the form of a neural network) followed by a WB part that processes the outputs of the BB into measurable physical quantities. The purpose of the blackbox is to map the controls \mathbf{V} (the model inputs) to the Hamiltonian of the system. The output of the BB then represents the elements of the Hamiltonian matrix. A general $N \times N$ complex matrix has $2N^2$ degree of freedom (N^2 components with real and imaginary parts).

Thus, the output layer of the BB must consist of $2N^2$ neurons. The other BB layers can be designed arbitrarily, and are custom engineered to provide the best performance for a given dataset.

The second layer is a Hamiltonian construction layer that arranges the outputs of the BB into an $N \times N$ matrix. A valid Hamiltonian has to be Hermitian (i.e., $H^\dagger = H$) and this is ensured by calculating the Hermitian part of the constructed matrix and discarding the anti-Hermitian part. This can be done simply by adding the matrix to its Hermitian conjugate. The output of this

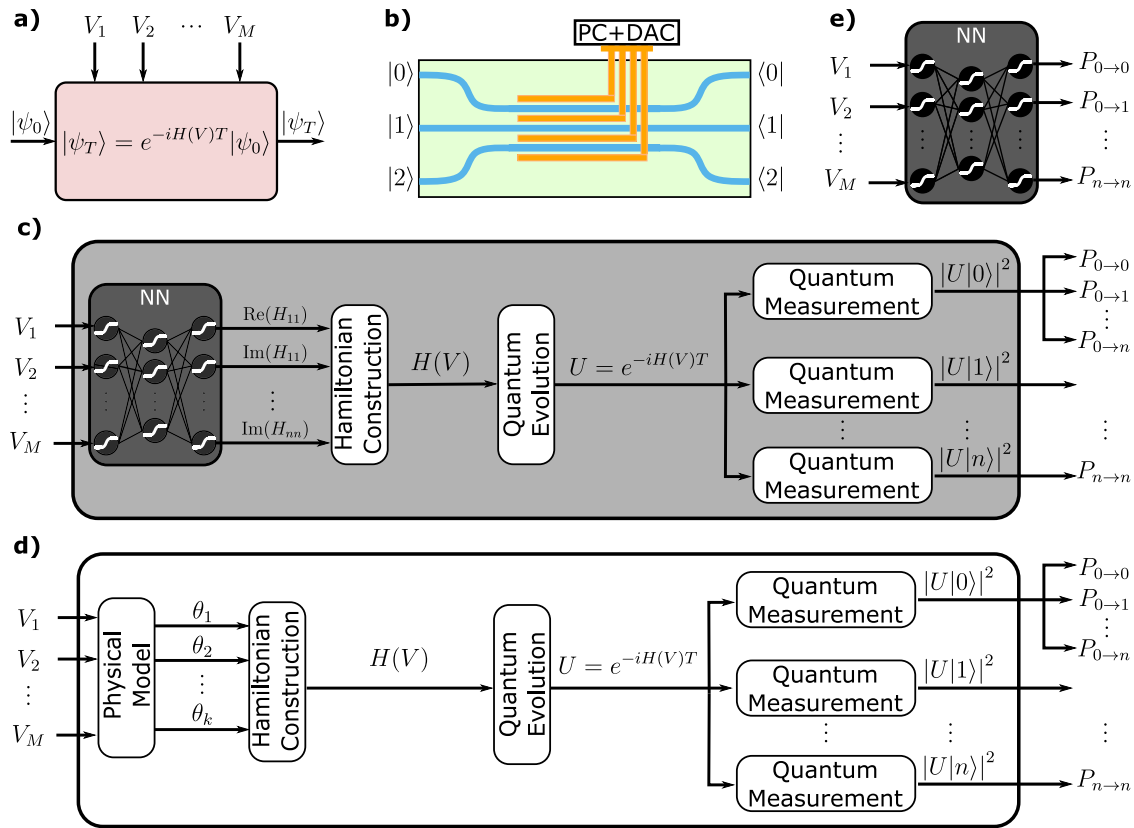


Fig. 1 Physical and machine learning models of the class of quantum devices considered in this paper. These are described by a time-independent Hamiltonian in the absence of interaction with the environment. **a** Representation of the class of quantum devices considered in this work. **b** A schematic of an integrated photonic voltage-controlled reconfigurable waveguide array chip, implementing a noiseless time-independent Hamiltonian. Photons enter from the input port (on the left), undergo a voltage-controlled propagation along the chip, and are then measured at the output port of the chip (on the right). **c** The structure of the proposed graybox model. The input to the model is the set of M controls, while the outputs are the quantum measurements for the set of computational basis as initial states. $P_{a \rightarrow b}$ indicates the transition probability from input port a to output port b . The graybox is a combination of black and white boxes. The blackbox estimates the real and imaginary components of each matrix element of the Hamiltonian. The whitebox layers construct the Hamiltonian matrix and perform the quantum evolution and measurements. **d** A fully whitebox architecture where a physical model is utilized. The first layer generates predefined Hamiltonian parameters that follow a known analytical dependence. The remaining layers perform the quantum evolution and measurements. **e** A fully blackbox model where only a generic neural network is utilized with no physical model.

layer is a valid control-dependent Hamiltonian. We did not enforce any structure or constraint on the Hamiltonian, to enable the best fitting allowed by the rules of quantum mechanics for a given dataset.

The Hamiltonian is then followed by subsequent layers that transform the Hamiltonian matrix into the set of probability outcomes—which can be measured experimentally—utilizing the laws of quantum mechanics. Starting from a valid Hamiltonian, there is no need to further use a BB since the dynamical equations are known. This saves the algorithm from trying to learn the rules of quantum mechanics from experimental data, which would complicate the process and result in a less accurate model. We use WB layers for the remaining steps. In particular, there is a layer that calculates the evolution unitary by the matrix exponentiation of the Hamiltonian $U = e^{-iHT}$, which is the solution of Schrödinger's time-independent equation. The final part of the GB model is a concatenation of N -layers representing the quantum measurement operation for each of the N input $|j\rangle$. In each layer we calculate $|\psi_T\rangle = U|j\rangle$, where $|j\rangle$ is the initial state of the system. After evolving the state to $|\psi_T\rangle$, the probabilities $P_{j \rightarrow k}$ are calculated by taking the absolute value squared of each entry of the state, that is, applying the Born's rule for quantum expectation values.

The added WB layers do not include any trainable parameter; they only exist in the BB part. Therefore, when we train the model

on a dataset, the only updates occur in the BB, generating a set of outputs that can be interpreted as a Hamiltonian.

An important aspect of the GB architecture is that it is independent of the physics of the system, i.e., it provides the most general form of a map between the control vector and the quantum measurements while keeping the most important physical quantities accessible via software, namely Hamiltonian, unitary and evolved state. This is the key aspect needed to perform quantum control as we are usually interested in implementing a quantum gate represented by the unitary or Hamiltonian and not by the measured evolved state. Having access to those quantities, even though the model is trained with quantum measurements, is the key feature of the GB architecture.

Whitebox and blackbox architectures

We benchmark the performance of our GB model against the fully WB and fully BB architectures. The WB approach (Fig. 1d) is equivalent to the standard model (curve) fitting and the details of the architecture depend on the physical system. The assumption is that all the relations between different dynamical variables are exactly known except for the parameters we are fitting. Generally, a WB consists of several layers. The first layer represents the mathematical relations between the controls and the Hamiltonian, with a set of unknown parameters. The input of this layer is the

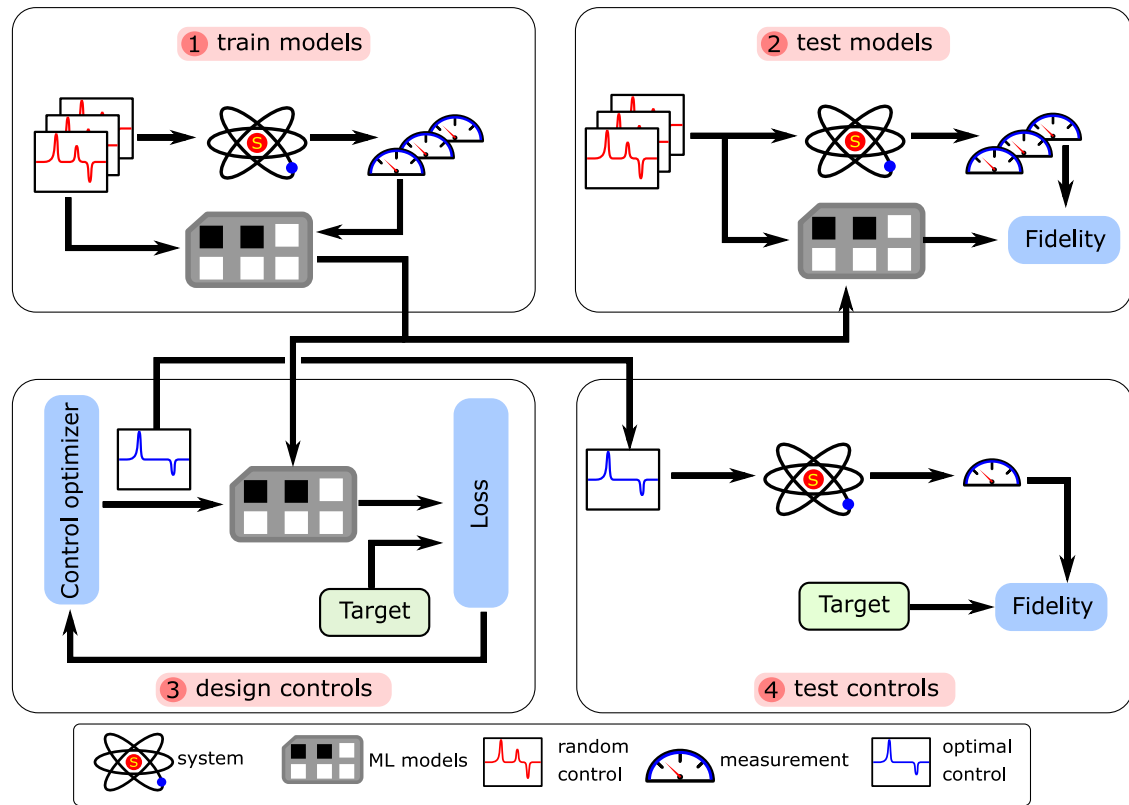


Fig. 2 Protocol schematic. The first step is to construct an experimental dataset by applying controls to the system and measuring the corresponding outputs. The dataset is then used to train the machine learning models (1). Next, the trained models are tested (2) for generalization by comparing their output predictions against a different experimental testing dataset. After that, the trained models can be used to optimize controls (3) to achieve a certain target, which could be a Hamiltonian, a unitary gate, or an output probability distribution. Finally, the obtained controls are tested (4) experimentally and the controlled system output is compared against the desired target.

control vector, and the output is a mathematically valid Hamiltonian. The remaining layers are identical to those of the GB and represent quantum evolution and quantum measurements. For some systems, we might need more layers (e.g., to model the fan-in/out in photonic devices⁴⁸). The WB provides the same access to hidden quantities as the GB and even provides more physics as we know exactly the analytical relations between Hamiltonian and control. However, if we do not know these relations, or they do not match the physical reality, the WB will fail and will not be useful for further applications.

The BB architecture (Fig. 1e) is largely different from the WB model since the relation between the control vector and the quantum measurement is modeled by a neural network. While any structure can be used, in this paper, we consider fully connected neural networks with a softmax output layer of N neurons. This enforces the outputs to form a probability distribution (i.e., positive numbers in $[0, 1]$ whose sum is equal to 1). This is consistent with what the model outputs represent, which are the probability amplitudes of the evolved quantum state. Since we are characterizing with N -initial basis states, we need N of such layers with the initial state chosen accordingly. This will give a total of N^2 outputs. The structure of the other hidden layers can only be determined and optimized by examining the performance on an actual dataset. No other physical quantities can be accessed through this architecture, but it is very good in fitting a dataset since it gives the maximum freedom in terms of representation. If we are only interested in controlling the outputs, a BB would be an efficient solution. But, if the goal is to estimate physical quantities (like unitary gates), then a BB model is not useful at all.

Protocol for training, testing, and controlling

Our protocol for training and testing models and controllers is schematically depicted in Fig. 2. It starts with preparing a dataset that will be used to train and test the ML BB. The dataset consists of examples. Each example is made of the M control inputs \mathbf{V} and the N^2 outputs of the model $P_{j \rightarrow k}$. So we start by generating random values for our control, let the system evolve, then perform the measurements and obtain the probabilities $P_{j \rightarrow k}$. We repeat this for the N input states we consider to obtain all the outputs. The procedure is repeated for multiple examples. The number of examples of the dataset depends on the particular structure of the ML model, the noise level in the experiment, and the acceptable performance level. Generally, the larger dataset is, the better the ML algorithm will perform. In our previous work⁴², only computer-simulated datasets were considered. In this paper, we create and test experimental datasets. This comes with many challenges including performing the experiment itself, the limited dataset size (to be feasible to collect), and the presence of noise not modeled by the quantum dynamics. In particular, initially, we found that statistical noise caused inconsistencies in the dataset, resulting in a poor performance of the method. As a result, we modified the dataset collection protocol, in particular the normalization of output power measurements as discussed in Supplementary Note 3. This extra step improved the performance of the ML significantly.

After the dataset is collected, it is split into the training and testing subsets, and the ML model is trained. The purpose of training is to minimize a loss function that measures the distance between the predictions and actual outputs from the training dataset. In⁴², the loss function measured the similarity between waveforms, because the model was modeling drift, and the

RMSprop algorithm⁴⁹ was used for the training. Here, we use the standard mean square error (MSE) as a loss function and use the ADAM algorithm⁵⁰ for training. Once the model is trained, its parameters are fixed and do not change in any of the remaining protocol stages. Next, the model is evaluated using the testing dataset, and its predictions are compared to the true corresponding values. The testing examples are not included in the optimization procedure, so they provide an unbiased evaluation of the performance of the model.

Let's now consider the trained model for control purposes. In this case, the model acts as a replacement for the actual experimental setup and can be probed via software for any purpose. In⁴², the controller was designed to be a GRU, and therefore the optimal control was not restricted to any class of pulses. Here, we use a different controller that is designed to directly obtain the parameters of a fixed pulse shape (i.e. the voltage amplitudes). We consider two types of applications. The first is for obtaining the values of the control \mathbf{V} to achieve a target output (i.e., probability amplitudes). In this case, we use the MSE as the control cost function, and the optimal control vector \mathbf{V}^* can be expressed as

$$\mathbf{V}^* = \arg \min_{\mathbf{V} \in \mathcal{I}} (\hat{\mathbf{y}}_p(\mathbf{V}) - \mathbf{y}_d)^T (\hat{\mathbf{y}}_p(\mathbf{V}) - \mathbf{y}_d) \quad (1)$$

where $\hat{\mathbf{y}}_p(\cdot)$ is the ML output predictions, \mathbf{y}_d is the desired target, and \mathcal{I} is the control domain, which reflects the maximum allowed range for the controls. To get accurate results, the ML model should be trained with dataset examples that lie in the same control domain as well. Note that the internal ML model parameters that define $\hat{\mathbf{y}}_p(\cdot)$ are not allowed to change during the optimization as they have been fixed after the training.

The other case is for achieving a target quantum gate (i.e., a target unitary). For this application, we use the gate fidelity as the control cost function for the controller defined as

$$F(U, W) = \frac{|\text{tr}(U^\dagger W)|^2}{N^2}, \quad (2)$$

where U and W are two unitary matrices, and N is their dimension. The gate fidelity lies in the range $[0, 1]$, with 1 representing the maximum overlap between the two gates. The optimal control voltages can then be represented as

$$\mathbf{V}^* = \arg \max_{\mathbf{V} \in \mathcal{I}} F(\hat{\mathbf{y}}_U(\mathbf{V}), U_d) \quad (3)$$

where $\hat{\mathbf{y}}_U(\cdot)$ is the evolution unitary obtained from the ML model, and U_d is the desired target gate. In this case, we can only use the GB and WB models, since a BB does not provide access to unitaries as discussed earlier. With this modular approach, the optimization algorithm of the controller can be chosen arbitrarily. While we choose a gradient-based method in this paper, other techniques such as genetic algorithms could also be used.

Finally, the optimal controls for a set of targets are applied experimentally, and the system is measured to construct the 'control' dataset. This dataset is then assessed and compared against the desired targets. Our main goal is to control a quantum system, and thus the assessment of any model should not just rely on its prediction capabilities, but also on how it performs in conjunction with a controller when tested experimentally.

Experimental results

We tailor the design of the models described above around the device used for the experimental verification for our proposal, a voltage-controlled quantum photonic circuit of continuously coupled waveguides based on lithium niobate technology, and schematically shown in Fig. 1b. The details about the chip's fabrication and its physical model are given in Supplementary Notes 1 and 2. The chip has 3 waveguides, corresponding to a qutrit system, and is controlled by 4 electrodes and their

Table 1. Comparison of the performance of each model on the training, testing, and control experimental datasets, as well as the experimental fidelities (average and instances greater than 99%) for 1000 randomly prepared output distributions and unitary gates.

| | GB | WB | BB |
|---------------------------|----------------------|----------------------|----------------------|
| Model | | | |
| Average training MSE | 8.3×10^{-5} | 1.5×10^{-2} | 9.4×10^{-5} |
| Average testing MSE | 9.1×10^{-5} | 1.5×10^{-2} | 1.1×10^{-4} |
| Output controller | | | |
| Average MSE | 2.6×10^{-3} | 1.7×10^{-2} | 2.9×10^{-3} |
| Fidelity | | | |
| Average | 99.53% | 97.47% | 99.48% |
| With >99% | 87.3% | 34% | 86.2% |
| Unitary controller | | | |
| Average MSE | 3.1×10^{-3} | 1.9×10^{-2} | – |
| Fidelity | | | |
| Average | 99.48% | 97.4% | – |
| with >99% | 71.3% | 12% | – |

respective voltages. In principle, there is no or negligible cross-talk in our device. This is guaranteed by the confinement of the electric field within the material due to the shielding effect from neighboring electrodes, as opposed to other technologies such as thermo-optic switching⁵¹. Thus, the electrodes can be activated simultaneously, which is how we perform the experiments in this paper. We implemented the ML models using the TensorFlow Python package^{52,53}, applied the protocol for training the three models, and then verified the performance of the controllers. The details of the implementations are also given in Supplementary Note 3. Moreover, we provide independent results of applying our method to a simulated synthetic dataset in Supplementary Note 4 including a showcase for a 32-mode chip with 33 electrodes.

The results of the models training and testing as well as the control performance, are reported in Table 1. The MSE evaluated at each iteration for training and testing sets are shown in Fig. 3a, b.

DISCUSSION

The plot of the learning curve in Fig. 3a shows the superior performance of the GB in terms of accuracy compared to the WB. This is due to the constraints imposed by the physical model that is used to construct the Hamiltonian in the case of the WB. As detailed in Supplementary Note 2, the commonly used device Hamiltonian is assumed to be tri-diagonal, real-valued, and linearly dependent on voltages. Our results show that these assumptions do not hold for a real device, and thus the degradation of the WB performance. On the other hand, the GB learns a general mathematically valid Hamiltonian and thus is able to better fit the experimental data. Initially, we designed the GB to enforce the Hamiltonian to be real-valued but otherwise arbitrary, and the fitting was not good. When we relaxed this assumption to allow a complex-valued Hamiltonian, the results were improved. Here we note that the Hamiltonian remains Hermitian to allow a unitary evolution, and thus it does not model losses. The normalization procedure (detailed in Supplementary Note 3) that we perform on the power measurements makes it unnecessary to model losses. The BB had a similar performance to GB, with the main drawback of losing the physical picture. In terms of the testing performance, Fig. 3b shows that the three models do not overfit, as the final MSE of the testing set is close to the final MSE of the training set. This means that the models do not memorize the examples of the training set. In other words, the model generalizes—there's no

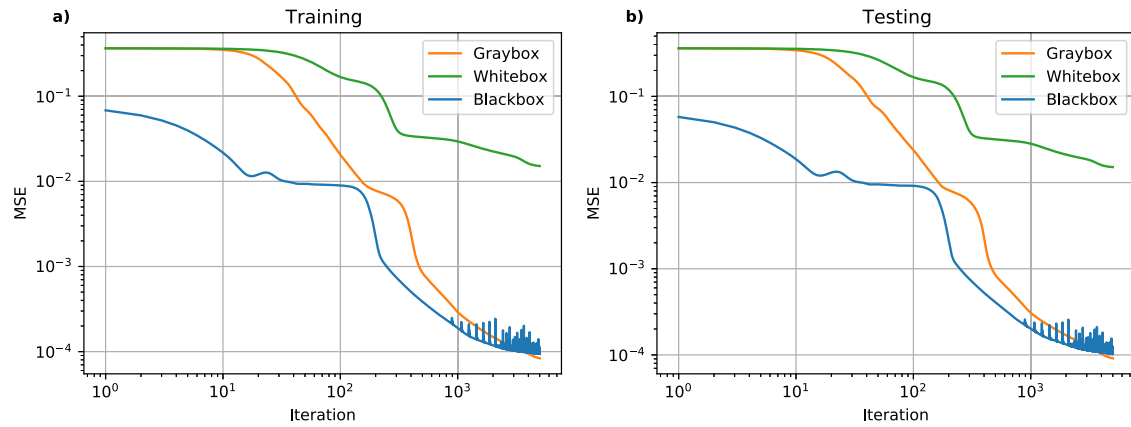


Fig. 3 Experimental performance of the machine learning models. The whitebox model consists of fan-in, reconfigurable, and fan-out sections, each modeled as a real-valued tri-diagonal Hamiltonian in addition to a linear dependence on voltage for the reconfigurable section. **a** Results of training the different models on the experimental dataset. The MSE is plotted versus iteration number. **b** The results of evaluating the different models on the testing set.

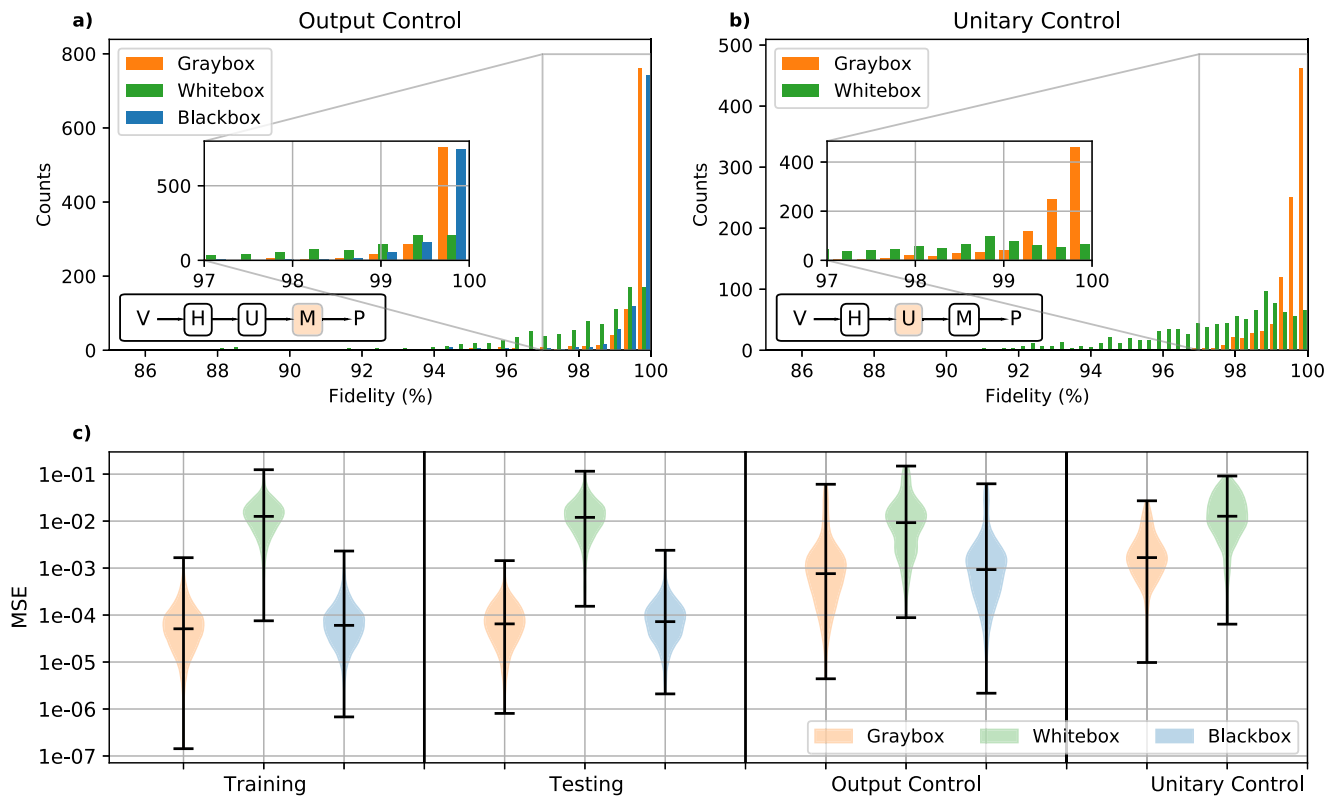


Fig. 4 Experimental quantum control performance. The distribution of the fidelity between the experimentally measured output power distribution and the desired target distribution for the three models. The whitebox model utilizes a real-valued tri-diagonal Hamiltonian with linear dependence on voltages, in addition to fan-in and fan-out sections. The results are for **a** the output controller, and **b** the unitary controller. The reported values are the average over the three distributions corresponding to each possible initial state. **c** Violin plot showing the statistics of the MSE obtained for the training, testing, and control datasets. The error bars depicted as the horizontal lines represent from bottom to top, the minimum, median, and maximum, respectively. The plot also shows an estimated kernel density for the data.

significant loss in prediction accuracy—confirming that the dataset, the model structure, and the training algorithm are well designed. Furthermore, the GB and BB clearly perform better than the WB.

The performance of the models is limited by the size of the dataset. Because the experimental data always suffer from some level of noise, the minimum MSE obtainable without overfitting is limited as well. Usually, the acceptable level of MSE depends on the specific application. In this paper, our application is to control the chip to obtain target power distributions as well as target

unitary operations. The ML models then act as a replacement/simulator of the actual setup. The experimental assessment of the optimal control will determine whether the model performance is accepted or needs improvement. In general, the way to improve models is by constructing very large datasets, which is the standard approach in most typical ML applications. For engineering applications, where we characterize and control a physical device, we are limited by how many measurements we can obtain. Thus, it becomes a tradeoff between the amount of time and

resources needed to construct the dataset experimentally and the accuracy of the trained models, which will also affect the performance of the controller.

The histogram of the controller fidelities between the desired target and the experimental measurements for 1000 randomly prepared output power distributions and 1000 randomly prepared unitary gates are shown in Fig. 4a, b, respectively. The plots show that the WB is particularly skewed towards lower fidelities (minimum is 80.53% compared to a minimum of approximately 91% for GB and BB for the output controller). Similarly, for the gate controller, the minimum fidelity is 86.6% and 94.95% for WB and GB. In Fig. 4c, we summarize the statistics of the MSE between the ML model predictions and actual outputs for the training and testing datasets, in comparison with the MSE between the experimentally controlled measurements and the targets for each of the two controllers.

The results of the controller for obtaining a target power distribution show once again the superior performance of the GB and BB over the WB in terms of both MSE and average fidelity. The same controller/optimization algorithm and cost function are used for the three models. Thus, the lower performance comes from the lower accuracy of the WB model itself. When considering the controller for target unitary, it is only possible to use a WB or a GB as they are the only models that can give access to the overall unitary evolution matrix. A BB cannot be utilized in this case since it only encodes the dynamics in an abstract machine-suitable format, and does not provide any physical picture.

The performance assessment of ML-based algorithms on real rather than synthetic datasets is critical. Different noise sources could affect the data in unpredicted ways, which may also be difficult to simulate. This can affect the performance of the ML algorithm. We see that the final MSE of training and testing for the experimental dataset is two orders of magnitude less than that of the synthetic dataset (shown in Supplementary Note 4). However, the control performance on the experimental dataset is accepted and thus, we also accept the model prediction performance. In other situations, this might not be the case, and the ML design has to be modified to achieve higher performance. Therefore, using a design based on simulations such as ref. ⁴² and applying it directly to an experimental dataset would not result in adequate performance, and so the whole workflow needs to be re-executed.

Determining the required dataset size, as well as NN architecture and complexity for higher-dimensional systems, is generally difficult and has to be studied case by case. In Supplementary Note 4, we show promising results for a simulated 32-mode chip. And while the dataset and neural network sizes

had to be increased, the overall protocol was still feasible to execute.

Finally, the GB provides sufficient physical insights for most purposes, for example, in Fig. 5a, b, where the tunability of the Hamiltonian as a function of a single electrode voltage is explored. The figure shows the GB prediction of the different Hamiltonian elements as a function of the voltage applied to a single electrode. We can use these predictions more generally when more than one electrode is tuned (although it would be more difficult to plot in this situation) for unitary control. We can use the GB to predict the unitary given the set of voltages even though the dataset originally did not include this information but rather the power distribution. Another advantage of using the GB, compared to WB, is the incorporation of unmodelled effects such as cross-talk. While the effect is negligible in our technology, in other situations, it can be difficult to have an exact/accurate WB model.

It is also important to realize that this predicted Hamiltonian, besides not being unique mathematically, represents physically an effective quantity, and so it will differ in structure (such as the existence of an imaginary part) from the ideal Hamiltonian that one may expect for a system. Depending on the purpose of the use of the GB we can control how much we make it 'blacker' or 'whiter'. For control applications, the best architecture is to have this effective Hamiltonian. In another application, such as modeling a device for the purpose of completely understanding the physics, the architecture of the GB might need to be modified to allow Hamiltonians that are closer to some expected structure. In Supplementary Note 5, we explore this idea in more detail. In particular, we explore the relaxation of the WB assumptions gradually until we reach the structure of the GB. The results show that the best architecture that fits the experimental data is a complex non-tri-diagonal Hamiltonian with non-linear dependence on the control voltage. This suggests that the reconfigurable section of the chip has variations along the propagation direction, which could be the result of fabrication imperfections. In other words, the estimated Hamiltonian effectively represents a time-dependent system. Finally, it is worth mentioning that reaching this conclusion was based on interpreting the mathematical structure obtained from the GB. Thus, the GB approach helped us understand better the behavior of our system.

In summary, we have shown how a GB model can be designed for a general quantum device, trained on experimental data, and verified by generating target unitary operations and output distributions with high fidelity. The performance was benchmarked against WB and BB models, showing the superior performance of our approach. Our approach is general and can

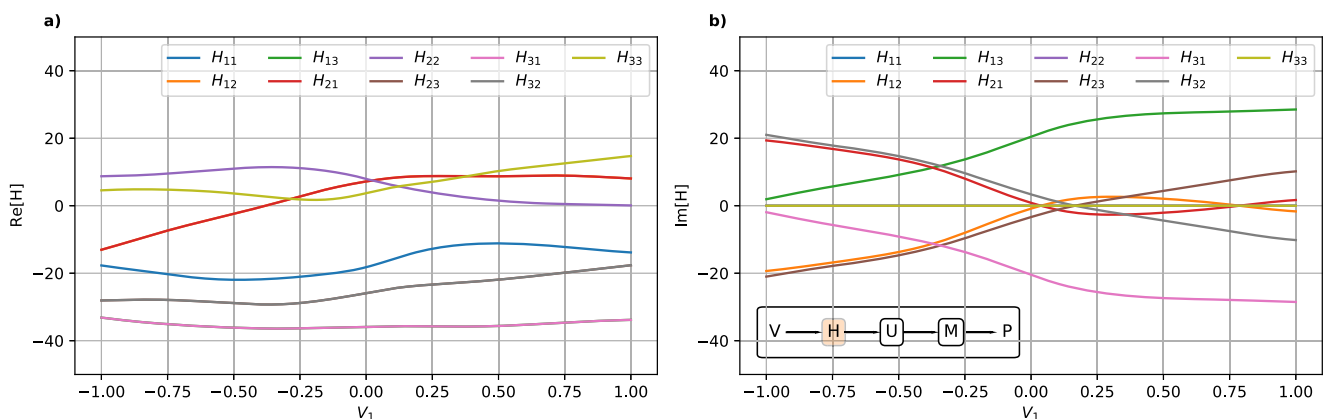


Fig. 5 Dependence of the Hamiltonian elements on a subset of input voltages, as predicted by the graybox model. **a** Real and **b** imaginary parts of the Hamiltonian matrix elements as a function of voltage when all electrodes are grounded except the first electrode. It should be noted that the imaginary parts of H_{11} , H_{22} , and H_{33} are by definition equal to zero. The non-linear dependence, the second off-diagonal elements, and the imaginary components indicate an effective Hamiltonian being estimated for a time-dependent system, attributed to the non-homogeneity of the chip along the propagation direction.

be applied to any quantum system, it can be extended to time-dependent and open quantum systems with the need to modify the ML structure and dedicated dataset-taking process for specific hardware or quantum systems^{43–45}. There are many possible extensions to this work as well. One possibility is to design a GB for other physical models, such as a Lindblad master equation for Markovian open quantum systems. One could also consider a hybrid approach between Hamiltonian learning using Genetic Algorithms (such as ref. ^{54,55}) and our numerical GB for the purpose of obtaining more detailed physical models. In terms of the ML aspects of this application, a study about the scaling requirements for the NN structures of the GB in relation to the dimensionality of the system would be interesting. However, it will be challenging because asymptotic analysis of ML algorithms is difficult or might be impossible. On the other hand, relying on numerical analysis might not be sufficient since the analysis will be restricted to a particular range of the scaling parameter and cannot be generalized outside that range. Another aspect related to any ML algorithm is the requirements of the training dataset size. For complex devices, it can be challenging to collect a large-sized dataset. However, some emerging techniques can facilitate this process, including incremental learning⁵⁶, transfer learning⁵⁷, and adaptive online learning⁵⁸. While these techniques are constantly developing in classical ML literature, there is still a gap in porting such methods to physics-based applications, especially quantum applications.

DATA AVAILABILITY

The data generated in this study is available upon reasonable request from the corresponding author.

CODE AVAILABILITY

The codes developed to generate this study are available upon reasonable request from the corresponding author.

Received: 14 June 2023; Accepted: 29 November 2023;
Published online: 13 January 2024

REFERENCES

- Leuchs, G. & Bruss, D. Quantum information: from foundations to quantum technology applications (John Wiley & Sons, 2019).
- Carr, H. Y. & Purcell, E. M. Effects of diffusion on free precession in nuclear magnetic resonance experiments. *Phys. Rev.* **94**, 630–638 (1954).
- Viola, L., Knill, E. & Lloyd, S. Dynamical decoupling of open quantum systems. *Phys. Rev. Lett.* **82**, 2417–2421 (1999).
- Biercuk, M. J., Doherty, A. C. & Uys, H. Dynamical decoupling sequence construction as a filter-design problem. *J. Phys. B: At. Mol. Opt. Phys.* **44**, 154002 (2011).
- Khodjasteh, K. & Viola, L. Dynamically error-corrected gates for universal quantum computation. *Phys. Rev. Lett.* **102**, 080501 (2009).
- Khaneja, N., Reiss, T., Kehlet, C., Schulte-Herbrüggen, T. & Glaser, S. J. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *J. Magn. Reson.* **172**, 296–305 (2005).
- de Fouquieries, P., Schirmer, S., Glaser, S. & Kuprov, I. Second order gradient ascent pulse engineering. *J. Magn. Reson.* **212**, 412–417 (2011).
- Ciaramella, G., Borzi, A., Dirr, G. & Wachsmuth, D. Newton methods for the optimal control of closed quantum spin systems. *SISC* **37**, A319–A346 (2015).
- Abdelhafez, M., Schuster, D. I. & Koch, J. Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation. *Phys. Rev. A* **99**, 052327 (2019).
- Leung, N., Abdelhafez, M., Koch, J. & Schuster, D. Speedup for quantum optimal control from automatic differentiation based on graphics processing units. *Phys. Rev. A* **95**, 042318 (2017).
- Caneva, T., Calarco, T. & Montangero, S. Chopped random-basis quantum optimization. *Phys. Rev. A* **84**, 022326 (2011).
- Haas, H., Puzzioli, D., Zhang, F. & Cory, D. G. Engineering effective hamiltonians. *New J. Phys.* **21**, 103011 (2019).
- Wu, R.-B., Chu, B., Owens, D. H. & Rabitz, H. Data-driven gradient algorithm for high-precision quantum control. *Phys. Rev. A* **97**, 042122 (2018).
- Wu, R.-B., Ding, H., Dong, D. & Wang, X. Learning robust and high-precision quantum controls. *Phys. Rev. A* **99**, 042327 (2019).
- Li, J., Yang, X., Peng, X. & Sun, C.-P. Hybrid quantum-classical approach to quantum optimal control. *Phys. Rev. Lett.* **118**, 150503 (2017).
- Chen, Q.-M. et al. Combining the synergistic control capabilities of modeling and experiments: illustration of finding a minimum-time quantum objective. *Phys. Rev. A* **101**, 032313 (2020).
- Yang, X. et al. Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses. *Phys. Rev. A* **102**, 062605 (2020).
- Dong, D. Learning control of quantum systems, 1090–1096 (Springer International Publishing, Cham, 2021).
- Judson, R. S. & Rabitz, H. Teaching lasers to control molecules. *Phys. Rev. Lett.* **68**, 1500–1503 (1992).
- Sivak, V. et al. Model-free quantum control with reinforcement learning. *Phys. Rev. X* **12**, 011059 (2022).
- Baum, Y. et al. Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer. *PRX Quantum* **2**, 040324 (2021).
- Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *npj Quantum Inf.* **5**, 33 (2019).
- Erdman, P. A. & Noé, F. Model-free optimization of power/efficiency tradeoffs in quantum thermal machines using reinforcement learning. *PNAS Nexus* **2**, 248 (2023).
- Matthews, J. C. F., Politi, A., Stefanov, A. & O'Brien, J. L. Manipulation of multi-photon entanglement in waveguide quantum circuits. *Nat. Photonics* **3**, 346–350 (2009).
- Erdman, P. A. & Noé, F. Identifying optimal cycles in quantum thermal machines with reinforcement-learning. *Npj. Quantum Inf.* **8**, 1 (2022).
- Fanchini, F. F., Karpat, G., Rossato, D. Z., Norambuena, A. & Coto, R. Estimating the degree of non-markovianity using machine learning. *Phys. Rev. A* **103**, 022425 (2021).
- Flurin, E., Martin, L. S., Hacohen-Gourgy, S. & Siddiqi, I. Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations. *Phys. Rev. X* **10**, 011006 (2020).
- Papič, M. & de Vega, I. Neural-network-based qubit-environment characterization. *Phys. Rev. A* **105**, 022605 (2022).
- Wise, D. F., Morton, J. J. & Dhomkar, S. Using deep learning to understand and mitigate the qubit noise environment. *PRX Quantum* **2**, 010316 (2021).
- Palmieri, A. M., Bianchi, F., Paris, M. G. & Benedetti, C. Multiclass classification of dephasing channels. *Phys. Rev. A* **104**, 052412 (2021).
- Ostaszewski, M., Miszczak, J., Banchi, L. & Sadowski, P. Approximation of quantum control correction scheme using deep neural networks. *Quantum Inf. Process.* **18**, 1–13 (2019).
- Khait, I., Carrasquilla, J. & Segal, D. Optimal control of quantum thermal machines using machine learning. *Phys. Rev. Res.* **4**, L012029 (2022).
- Zeng, Y., Shen, J., Hou, S., Gebremariam, T. & Li, C. Quantum control based on machine learning in an open quantum system. *Phys. Lett. A* **384**, 126886 (2020).
- Bausch, J. & Leditzky, F. Quantum codes from neural networks. *New J. Phys.* **22**, 023005 (2020).
- Baireuther, P., O'Brien, T. E., Tarasinski, B. & Beenakker, C. W. Machine-learning-assisted correction of correlated qubit errors in a topological code. *Quantum* **2**, 48 (2018).
- Lennon, D. T. et al. Efficiently measuring a quantum device using machine learning. *Npj Quantum Inf.* **5**, 1–8 (2019).
- Tranter, A. D. et al. Multiparameter optimisation of a magneto-optical trap using deep learning. *Nat. Commun.* **9**, 4360 (2018).
- Cimini, V. et al. Calibration of multiparameter sensors via machine learning at the single-photon level. *Phys. Rev. Appl.* **15**, 044003 (2021).
- Wiebe, N., Granade, C., Ferrie, C. & Cory, D. Quantum hamiltonian learning using imperfect quantum resources. *Phys. Rev. A* **89**, 042314 (2014).
- Wang, J. et al. Experimental quantum hamiltonian learning. *Nat. Phys.* **13**, 551 (2017).
- Bohlin, T. Practical grey-box process identification (Springer London, 2006).
- Youssry, A., Chapman, R. J., Peruzzo, A., Ferrie, C. & Tomamichel, M. Modeling and control of a reconfigurable photonic circuit using deep learning. *Quantum Sci. Technol.* **5**, 025001 (2020).
- Youssry, A., Paz-Silva, G. A. & Ferrie, C. Characterization and control of open quantum systems beyond quantum noise spectroscopy. *Npj Quantum Inf.* **6**, 1–13 (2020).
- Youssry, A. & Nurdin, H. I. Multi-axis control of a qubit in the presence of unknown non-Markovian quantum noise. *Quantum Sci. Technol.* **8**, 015018 (2023).
- Youssry, A., Paz-Silva, G. A. & Ferrie, C. Noise detection with spectator qubits and quantum feature engineering. *New J. Phys.* **25**, 073004 (2023).

46. Perrier, E., Tao, D. & Ferrie, C. Quantum geometric machine learning for quantum circuits and control. *New J. Phys* **22**, 103056 (2020).
47. Genois, É. et al. Quantum-tailored machine-learning characterization of a superconducting qubit. *PRX Quantum* **2**, 040355 (2021).
48. Peruzzo, A. et al. Quantum walks of correlated photons. *Science* **329**, 1500–1503 (2010).
49. Tieleman, T. & Hinton, G. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw. Mach. Learn.* **4**, 26–31 (2012).
50. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In: Bengio, Y. & LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015).
51. Prencipe, A. & Gallo, K. Electro- and thermo-optics response of x-cut thin film linbo3 waveguides. *IEEE J. Quantum Electron.* **59**, 1–8 (2023).
52. Abadi, M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from <https://www.tensorflow.org/> (2015).
53. Chollet, F. et al. Keras. Software available from <https://keras.io/> (2015).
54. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
55. Gentile, A. A. et al. Learning models of quantum systems from experiments. *Nat. Phys.* **17**, 837–843 (2021).
56. Losing, V., Hammer, B. & Wersing, H. Incremental on-line learning: a review and comparison of state of the art algorithms. *Neurocomputing* **275**, 1261–1274 (2018).
57. Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *J. Big Data* **3**, 1–40 (2016).
58. Hoi, S. C., Sahoo, D., Lu, J. & Zhao, P. Online learning: a comprehensive survey. *Neurocomputing* **459**, 249–289 (2021).

ACKNOWLEDGEMENTS

A.P. acknowledges an RMIT University Vice-Chancellor's Senior Research Fellowship and a Google Faculty Research Award. M.L. was supported by the Australian Research Council (ARC) Future Fellowship (FT180100055). B.H. was supported by the Griffith University Postdoctoral Fellowship. This work was supported by the Australian Government through the Australian Research Council under the Center of Excellence scheme (No: CE170100012), and the Griffith University Research Infrastructure Program. This work was performed in part at the Queensland node of the Australian National Fabrication Facility, a company established under the National Collaborative Research Infrastructure Strategy to provide nano- and micro-fabrication facilities for Australia's researchers. This research was also undertaken with the assistance of resources from the National Computational Infrastructure (NCI Australia), an NCRIS-enabled capability supported by the Australian Government.

AUTHOR CONTRIBUTIONS

A.Y. and A.P. conceived the experiment. B.H., F.L. and M.L. fabricated the integrated photonic device. A.Y., Y.Y., R.J.C. and A.P. carried out the experiments and performed the data analysis. All the authors discussed the results and contributed to the writing of the paper.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41534-023-00795-5>.

Correspondence and requests for materials should be addressed to Alberto Peruzzo.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024