Quantum-tailored machine-learning characterization of a superconducting qubit

Élie Genois,1, * Jonathan A. Gross,1, † Agustin Di Paolo,1, ‡ Noah J. Stevenson,2
Gerwin Koolstra,2, 3 Akel Hashim,2 Irfan Siddiqi,2, 3, 4, 5 and Alexandre Blais1, 5

1 Institut quantique & Département de Physique, Université de Sherbrooke, Québec J1K 2R1, Canada
2 Quantum Nanoelectronics Laboratory, University of California, Berkeley, Berkeley CA 94720, USA
3 Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley CA 94720, USA
4 Materials Science Division, Lawrence Berkeley National Laboratory, Berkeley CA 94720, USA
5 Canadian Institute for Advanced Research, Toronto, Ontario M5G 1M1, Canada

(Dated: June 25, 2021)

Machine learning (ML) is a promising approach for performing challenging quantum-information tasks such as device characterization, calibration and control. ML models can train directly on the data produced by a quantum device while remaining agnostic to the quantum nature of the learning task. However, these generic models lack physical interpretability and usually require large datasets in order to learn accurately. Here we incorporate features of quantum mechanics in the design of our ML approach to characterize the dynamics of a quantum device and learn device parameters. This physics-inspired approach outperforms physics-agnostic recurrent neural networks trained on numerically generated and experimental data obtained from continuous weak measurement of a driven superconducting transmon qubit. This demonstration shows how leveraging domain knowledge improves the accuracy and efficiency of this characterization task, thus laying the groundwork for more scalable characterization techniques.

I. INTRODUCTION

Machine learning (ML) has recently been applied to solve problems in numerous areas of physics, including quantum-information science [1, 2]. For example, ML models have been used to tackle quantum-computing tasks including qubit readout [3–5], quantum control [6, 7] and quantum state tomography [8–10] among others. These “black-box” models enjoy wide applicability to different problems, as their structure is agnostic to the physical processes involved in the targeted task. However, information about the physics of the learned process can also be leveraged in an effort to improve upon these black-box approaches by making them more trainable and interpretable. Recent work has shown promising results in this direction by using “white-box” quantum features, such as physical laws, symmetries and relevant correlations, in the ML approach. For example, including quantum features has been used to improve the characterization of quantum noise [11] as well as to learn quantum states more efficiently [12, 13] and in an interpretable way [14].

Here we are interested in developing quantum-aware ML models in order to characterize the dynamics of a quantum device from physical observations. Accurate device characterization is crucial for producing high-fidelity quantum operations and algorithms [15]. However, just like most techniques in quantum characterization, validation and verification, our ability to perform this characterization is intrinsically limited by quantum properties such as the lack of direct access to the state [16–18].

By tailoring the machine-learning approach to reflect as much of our knowledge of the quantum device as possible, we seek to mitigate the effect of these intrinsic difficulties.

As an example of a generic situation in quantum information, we focus on the circuit QED architecture and characterize the dynamics of a resonantly driven superconducting transmon qubit [19]. The driven qubit is continuously monitored by a weak dispersive measurement [20]. By training a ML model to predict measurement outcomes from the weak-measurement data, we can characterize the qubit parameters via the quantum-trajectory formalism [21].

This physics-inspired approach overcomes two important limitations of generic machine-learning tools that are agnostic to the physics of the problem at hand, such as Ref. [22] which also considered learning quantum trajectories from the continuous weak measurement of a transmon qubit. First, it is more efficiently trainable as significantly less experimental data is needed to train the ML model. Second, the physics-inspired approach allows for a direct device characterization, whereas the black-box approach lacks interpretability as the trained ML model cannot be easily associated with an explicit physical description of the device. We address these two limitations by designing ML models that exploit our domain knowledge of quantum trajectories and dispersive measurements in circuit QED [20]. We also explore the relations between the training efficiency, physical interpretability and achievable accuracy of the ML model predictions.

In the following, we train our ML models on numerically generated and experimental data. The numerical data provides an opportunity to quantify improvements and limitations of different learning approaches, while the experimental data allows us to demonstrate how these approaches can be applied with success to real quantum

* elie.genois@usherbrooke.ca
† jarthurgross@google.com
‡ adipao@mit.edu
systems, even in the presence of noise and imperfections. For example, we show how our approach allows us to learn the Hamiltonian and Lindblad operators of the experimental device as well as the quantum efficiency of the measurement chain, while accounting for state preparation and measurement (SPAM) errors. Our approach is therefore closer to Hamiltonian learning [23–25] than quantum process tomography where a single process or a discrete collection of processes is learned [9, 26–29].

The paper is organized as follows. After presenting the experiment, its numerical model and the generic machine-learning model in Secs. II to IV, we present two main approaches to introduce quantum features in the ML model and show their impact on numerically generated data. Namely, we implement a loss function that encourages quantum mechanical features in a recurrent neural network in Sec. V and we tailor the architecture of the ML model based on the formalism of quantum trajectories in Sec. VI. We then apply these techniques to experimental data in Sec. VII and analyze the parameters resulting from such device characterization approaches in Sec. VIII.

II. EXPERIMENT

As schematically illustrated in Fig. 1, the experiment consists of a superconducting transmon qubit which is continuously probed by a weak dispersive measurement via a microwave cavity. Together with this weak measurement tone, an additional continuous tone at the qubit transition frequency produces Rabi oscillations of the qubit. The interaction-picture Hamiltonian describing the system is $H = H_{\text{int}} + H_R$, where [20]

$$H_{\text{int}} = \frac{\hbar \chi}{2} a^\dagger a \sigma_z,$$

$$H_R = \frac{\hbar \Omega_R}{2} \sigma_x.$$  

Here, $\chi$ is the dispersive qubit-cavity coupling, $\Omega_R$ the Rabi frequency, $a^\dagger$ ($a$) the creation (annihilation) operator for the cavity mode, and $\sigma_{x,y,z}$ are the qubit’s Pauli operators. In our experiment, we have $\Omega_R/2\pi = 0.222$ MHz, $\chi/2\pi = -0.47$ MHz, and cavity linewidth $\kappa/2\pi = 1.56$ MHz.

The experiment is realized in three essential steps. The qubit is first prepared in one of the six cardinal points of the Bloch sphere ($|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$). A weak measurement tone is then applied at the cavity frequency of 6.679 GHz to continuously probe the qubit state in the $\sigma_z$ basis for a variable time $T$ all the while a microwave drive applied at the qubit frequency of 5.473 GHz drives Rabi oscillations. Finally, at the end of the prescribed evolution time, a strong projective measurement of one of the qubit operators $\sigma_{x,y,z}$ is performed. After being amplified and digitized, the weak measurement signal takes the form of a voltage time-series $\Delta M_t$, whereas the strong readout signal is integrated and discriminated to yield a single binary output $Y \in \{+1, -1\}$. In the machine-learning trainings, this final projective measurement is used to quantify the accuracy of the ML model in learning the quantum trajectories of the monitored qubit.

In order to build the dataset used to train our machine-learning models, we perform a total of 3.5 million shots of the three-step experiment. For each of the 18 possible preparation-measurement setups (6 preparations and 3 strong-measurement axes), the weak-measurement records are acquired for different total evolution time $T$ between 0 ps (corresponding to a projective readout immediately following the preparation) and 8 ps in time steps of $\Delta t = 0.04$ ps. The resulting dataset formed by the weak measurements, together with their associated preparation and readout, is then divided into training,
validation and test sets with proportions of 0.75, 0.20 and 0.05 respectively.

III. NUMERICAL MODELING

Single realizations of the experiment can be described by the formalism of quantum trajectories [21]. A quantum trajectory fully captures the evolution of a qubit state $\rho$ over time. Trajectories can be extracted from the continuous weak measurement by integrating a stochastic master equation (SME) of the form [30]

$$\frac{d\rho_t}{dt} = -i[H_R, \rho_t]dt + {\mathcal D}[L]\rho_t dt + \frac{\eta}{2}\left(\mathcal{H}[L]\rho_t dW_t^q + \mathcal{H}[-iL]\rho_t dW_t^q \right).$$

In this expression, the Lindblad operator $L = \sqrt{\Gamma_d/2}\sigma_z$ describes the weak-measurement backaction on the qubit for $\Gamma_d$ the measurement-induced dephasing rate, which is a function of the weak measurement power, and $\eta \leq 1$ is a scaling factor. The quantum efficiency of the measurement chain (including signal amplification and detection) [20]. Moreover, the independent, Gaussian-distributed Wiener increments $dW_t^q$ have mean $E(dW_t^q) = 0$ and variance $\text{Var}(dW_t^q) = dt$, for both quadrature $q \in \{I, Q\}$ of the weak measurement signal at every time $t$. $\mathcal{D}[L]\rho = L\rho L - \frac{1}{2}(L\rho L + \rho L L)$ is the standard dissipator and $\mathcal{H}[L]\rho = L\rho + \rho L - \rho \text{Tr}L\rho + \rho L^\dagger L$ is the measurement superoperator describing the backaction of the weak measurement on the quantum state [30]. In Eq. (3), we have assumed that both quadratures of the measurement record are monitored, something which is known as heterodyne detection [20]. We have also assumed that the cavity-mode dynamics were much faster than that of the qubit, allowing us to adiabatically eliminate the cavity degrees of freedom from the SME [31].

Experimentally, one has access to the weak-measurement record $\Delta M_t^q$ corresponding to a finite sampling in time of the otherwise infinitesimal signal $dM_t^q$. These stochastic variables are related to the Wiener increments through [30]

$$dM_t^q = \sqrt{\frac{\eta}{2}}\text{Tr}\left[\rho_t(c^q + c^q\dagger)\right]dt + dW_t^q,$$

where $c^q = L(c^q = -iL)$ for the $I$ $(Q)$ quadrature. The measured signal $\Delta M_t^q$ is composed of a deterministic portion holding information about the qubit state that scales as $\Delta t$ and a larger stochastic contribution (noise) of order $\sqrt{\Delta t}$.

A. Numerical data

The numerical data is generated by integrating the SME of Eq. (3) using the positivity-preserving scheme introduced by Rouchon and Ralph [32] and implemented in QuTiP [33]. To avoid most numerical integration errors from affecting the numerical datasets, we use a small integration step of 0.001 µs and then coarse grain the weak-measurement results to match the experimental time steps of $\Delta t = 0.04$ µs. In this way, 1.75 million trajectories are generated and divided into training, validation and test sets with proportions of 2/3, 1/6 and 1/6 respectively. Importantly, we generate experimentally-realistic data by using the parameters obtained from independent experimental calibration of the superconducting device. Below, we refer to these as the true parameters $\{\Omega_R, \Gamma_d, \eta\}$ of our physical model given by Eqs. (1) to (3). Using only realistic data allows us to transparently extend the machine-learning improvements drawn from the synthetic results to actual experimental settings.

IV. MACHINE LEARNING

Finding a physical model accurately capturing the dynamics of a quantum device and precisely calibrating all the parameters involved in the model is nontrivial and requires extensive quantum measurements and computational resources [16]. However, evaluating the accuracy of a given model at predicting measurement outcomes is relatively simple. Thus, the task of finding a good model to characterize observed quantum dynamics naturally lends itself to a machine-learning-based approach, where the accuracy of the trainable model can be optimized on a given dataset. In this context, neural networks are particularly attractive as they can serve as universal function approximators [34]. Specifically, recurrent neural network (RNN) architectures can preserve local time correlations [35] that are present in the weak-measurement time-ordered data. This architecture is therefore a good candidate for the task at hand.

A. Recurrent Neural Network

Figure 1 shows the structure of our machine-learning model. The RNN consists of a unit cell that is repeated at every new input $\Delta M_t$ of the time-series data, producing an output $\tilde{h}_t$ known as the hidden state. This hidden state is then combined with the next time-series input $\Delta M_{t+\Delta t}$, allowing information to propagate through the sequence and weight on the outputs at future times. Here, we use a RNN architecture known as gated recurrent unit (GRU) [36]. Further details on the structure of our RNN implementation can be found in Appendix A. To relate the hidden state of the neural network $\tilde{h}_t$ to the targeted state of the qubit $\tilde{r}_t$, we complete our ML model with two feed-forward neural layers named encode and decode layers. The encode layer maps a “one-hot” encoded preparation vector $\vec{p}$ to the initial hidden state $\tilde{h}_0$ of the RNN. The decode layer transforms a hidden state $\tilde{h}_t$ to a prediction for the qubit state $\tilde{r}_t = (\langle \sigma_z \rangle_t, \langle \sigma_y \rangle_t, \langle \sigma_z \rangle_t)$. 

The machine-learning task then consists of inferring an accurate quantum state for the qubit at every time \( t \) during the evolution, given a selected preparation \( \vec{p} \) and the acquired weak-measurement time-series \( \Delta M_t \). The training label corresponds to a one-hot encoded vector \( \vec{m} \) of the projective measurement outcome \( Y \in \{+1, -1\} \) for the measured operator \( \sigma_{x,y,z} \). This label \( \vec{m} \) represents a single bit of information about the qubit state acquired from the projective measurement at the very end of the weak measurement. In other words, the task amounts to reconstructing the quantum dynamics of the qubit by learning its quantum trajectory \( \hat{r}_t \), without having explicit access to the true quantum trajectory \( r_t \).

\[ L_{CE} = -\frac{1}{N} \sum_{n=1}^{N} \left[ \frac{1 + Y_n}{2} \log(\Pi_n^\alpha) + \frac{1 - Y_n}{2} \log(1 - \Pi_n^\alpha) \right], \]

for the pairs \((Y_n, \Pi_n^\alpha)\) of projective measurement outcome \( Y_n \in \{+1, -1\} \) and the probability \( \Pi_n^\alpha \in [0, 1] \) predicted by the ML model of measuring the outcome +1 for the operator \( \sigma_\alpha \).

In addition to using \( L_{CE} \) as a training loss, we use the cross entropy to compare the performance of different models at predicting the experimentally observed (or simulated) measurement outcomes. Using the cross entropy as a performance metric is motivated by the fact that, in the present case where the measurement outcomes are fixed, it is equivalent up to an additive constant to the Kullback-Leibler divergence [34]. Moreover, using the cross entropy allows us to quantify the accuracy of the trained ML model at the level of its individual output trajectories, without having to introduce statistical errors by binning or averaging the trajectories to yield the desired metric.

Figure 2 presents the performance of RNNs trained on numerically generated data – we come back to experimental data in Sec. VII. Panel a) illustrates the deviations of the learned quantum trajectory (light blue line) from the true trajectory (dark blue line) for a single representative component of \( r_t \) and \( \hat{r}_t \) sampled from the test set. As is made clear from this example, because the model is only trained to be accurate at the very end of the quantum evolution where the projective measurement is performed, there are little variations between true and learned trajectories with the generic \( L_{CE} \) black-box neural-network (BBNN) approach at late times. However and for the same reason, the learned quantum trajectories deviate from the true trajectories at early times. This observation is made more quantitative in Fig. 2b), which shows the mean squared error as a function of time between the true trajectories and those learned using the cross-entropy loss (light blue line). The corresponding cross entropy between the generated trajectories of the entire test set and the RNN-learned ones is, moreover, shown in Fig. 2c).

In short, a RNN trained with the loss function \( L_{CE} \) is relatively accurate at predicting the measurement outcomes of unseen data, but fails to precisely capture quantum dynamics during the entire evolution. Extracting information about the device parameters with this approach is therefore expected to lead to flawed results.
B. Physics-Inspired Loss Functions

To improve the situation, we now expend the loss function with terms penalizing learned trajectories which do not respect specific quantum mechanics features. Such regularization of the training allows the RNN to explore the optimization space more efficiently \[42\].

Our first physics-inspired loss is the positivity loss, which is aimed at penalizing ML outputs corresponding to unphysical qubit states living outside of the Bloch sphere. It takes the form

\[
\mathcal{L}_{\text{posit}} = \frac{1}{N(N_t + 1)} \sum_{t=0}^{N_t} \sum_{n=1}^{N} \text{ReLU} \left( |\hat{r}_{t,n}|^2 - 1 \right), \tag{6}
\]

where ReLU\( (x) = \max(0, x) \) is the rectified linear unit and, as above, \( \hat{r}_t = \langle \sigma_x \rangle_t, \langle \sigma_y \rangle_t, \langle \sigma_z \rangle_t \) is the RNN prediction of the qubit state. We note that a BBNN tends to learn states that violate this positivity constraint at early times.

The second loss term proposed is the preparation loss which is intended to make the RNN predictions accurate at the beginning of the quantum trajectory by using our knowledge of the qubit’s initial state. This loss is expressed as

\[
\mathcal{L}_{\text{prep}} = \frac{1}{N} \sum_{n=1}^{N} |\hat{r}_{0,n} - r_{0,n}|^2, \tag{7}
\]

where, when dealing with experimental data, we can set the targeted preparation states \( r_0 \) to be consistent with state preparation and measurement (SPAM) errors. To do so, we consider the subset of data where the projective readout immediately follows the preparation (i.e. the data with a weak-measurement time \( T = 0 \) µs). The states \( r_0 \) are then inferred by extracting the expectation value of the \( \sigma_{x,y,z} \) operators on the prepared cardinal states of the Bloch sphere, thus effectively carrying out quantum state tomography of these initial states \[26\].

This simple pre-processing of the dataset based on our understanding of the dynamics then allows us to train the ML model to output the best initial states given the same measurement data.

The third physics-inspired loss that we introduce is the prediction loss. The objective is to force the model to use directly the deterministic information about the qubit state that is present in the time-series data. To do so, we add an output \( \Delta \tilde{M}_t^q \) to the RNN model that serves as a prediction for the next input of the time-series weak-measurement data \( \Delta M_{t+\delta t}^q \). The prediction loss is implemented as a mean squared error

\[
\mathcal{L}_{\Delta M} = \frac{1}{N_t N} \sum_{t=0}^{N_t} \sum_{n=1}^{N} \left( \Delta \tilde{M}_{t,n}^q - \Delta M_{t+\delta t,n}^q \right)^2, \tag{8}
\]

that pushes the model to make predictions about the next weak measurements that are as accurate as possible. Note that the mean squared error is equivalent to a cross-entropy loss for a Gaussian random variable like \( \Delta M_t^q \).

Adding these different contributions, our physics-inspired loss function takes the form

\[
\mathcal{L}_{\text{PI}} = \mathcal{L}_{\text{CE}} + w_{\text{posit}} \mathcal{L}_{\text{posit}} + w_{\text{prep}} \mathcal{L}_{\text{prep}} + w_{\Delta M} \mathcal{L}_{\Delta M}, \tag{9}
\]

where \( w_j \) are relative weights that can be optimized similarly to other hyperparameters of typical NN trainings.

As illustrated in Fig. 2, the trajectories resulting from training the RNN with the physics-inspired loss function \( \mathcal{L}_{\text{PI}} \) are significantly more accurate than those obtained using only a physics-agnostic \( \mathcal{L}_{\text{CE}} \). This is illustrated qualitatively in Fig. 2a) where the red line correspond to a representative result obtained using Eq. (9). In contrast to the results obtained with \( \mathcal{L}_{\text{CE}} \) (light blue line),
the predicted trajectory now matches the true trajectory (dark blue line) significantly better at early times. Panels b) and c) show the importance of the different physics-inspired loss terms by presenting the mean squared error and the cross entropy, respectively, for RNNs trained on different subsets of $L_{P1}$.

Figure 3 illustrates how $L_{P1}$, and specifically the prediction loss term $L_{ΔM}$, helps the RNN learn significantly better quantum trajectories. Panel a) shows a representative example of an input weak-measurement signal $ΔM$ that was numerically generated (light blue) and the prediction of the same signal $Δ\bar{M}$ (dashed red) from a RNN trained using Eq. (9). We see that these two curves do not directly correspond, which is expected from the fact that $ΔM$ is mostly composed of random noise (proportional to $\sqrt{Δt}$) coming from the Wiener increments in Eq. (4). Figure 3b) shows the agreement between the deterministic portion of this input signal (dark blue) and the same RNN prediction — notice the change of scale. This result shows that our physics-inspired RNN model is able to infer the deterministic information (proportional to $Δt$) hidden in the noisy weak-measurement signal, even if this relevant information is at least an order of magnitude smaller than the noise for our experimentally-realistic parameters. Interestingly, the ML model is then able to use the information associated with this weak-measurement prediction in order to output quantum trajectories that are significantly more accurate throughout the entire quantum evolution.

### VI. QUANTUM-TAILORED ML ARCHITECTURE

Thus far, we have used a generic RNN and shown how leveraging prior knowledge about the structure of a quantum problem allowed us to learn more accurate dynamics while using the same input data. In this section, we follow this intuition further by incorporating features of quantum mechanics in the machine-learning model architecture itself. To do so, we use a physically interpretable model to learn a useful characterization of the device from the weak-measurement data.

The idea consists of directly using our physical model of the quantum dynamics, given by Eq. (3), as the trainable model. This approach makes use of the direct relation between the hidden state of the RNN and the quantum state of the device. Since all measurement predictions are derived from the hidden state of the RNN, the vector corresponding to the hidden state holds an overcomplete representation of the quantum state, and the layers that translate it into Pauli-measurement probabilities can be used to map this representation into the usual density-matrix representation: $h_t \rightarrow \rho_t$. As such, the function $F$ that the RNN is set out to learn is the one updating the quantum state given weak measurement data

$$h_{t+\Delta t} = F_{\text{RNN}}(h_t, \Delta M^q_t),$$

$$\rho_{t+\Delta t} = F_{\text{SDE}}(\rho_t, \Delta M^q_t) = \rho_t + \Delta \rho(\rho_t, \Delta M^q_t),$$

where the last equation is a discretized integration of Eq. (3). Thus, we can use a stochastic differential equation (SDE) integrator, with a set of free parameters, that describe the experimental setup, such as $\{H_R, L, η\}$ that appear in Eq. (3). With this choice, learning accurate quantum trajectories from weak-measurement data becomes a perfectly interpretable device characterization.

We implement the SDE integrator based on the Milstein method [43] using an auto-differentiable library. This implementation allows the model to operate on a GPU and to be trained efficiently using back-propagation in the same way as generic neural networks [34]. In the following, we will refer to this machine-learning approach as SDE learning. We note that the use of a differential equation integrator as a trainable model can be straightforwardly extended to other parameter-estimation problems described by a differential equation.
authors of Ref. [44] independently developed a similar learning approach for control problems, although they only analyze the performance on numerical data with high temporal resolution. We emphasize that realistic experimental restrictions, such as the size of the time increment $\Delta t$, have important consequences on the performance of these learning schemes, which we address in Appendix C.

Using SDE learning on numerically generated data, we have found that this approach is able to reconstruct the true quantum trajectories up to numerical-integration errors (not shown). Here, errors are due to the finite and realistic time step of the weak-measurement data, which is set to $\Delta t = 0.04 \mu s$. The SDE model based on Eq. (3) outperforms the RNN models at the task of learning the correct system dynamics, achieving a total mean squared error of $6.3 \times 10^{-5}$ on the trajectories of the test set. In comparison, the physics-inspired RNN gives an error of $5.1 \times 10^{-3}$ when trained on the same data. Importantly, the SDE-learning approach also directly outputs the device parameters and the generators of the quantum dynamics. The characterization of the device resulting from this approach is further discussed in Sec. VIII. We note that we can also account for SPAM errors in the SDE-learning approach by learning the preparation and projective measurement maps from the initial state tomography subset data, similarly to what can be done with the preparation loss.

While the excellent performance of the SDE learning is to be expected for numerically generated data, the next section explores the application of this approach to experimental data where the model of Eq. (3) does not capture all of the physical dynamics present in the weak measurements.

VII. EXPERIMENTAL RESULTS

We now move from numerically generated data to experimental data obtained from measurements on a superconducting qubit. Because we have already limited the training to experimentally accessible quantities, there are no differences in the training process. However, since the true quantum trajectories are now unknown and the experimental data is affected by nonidealities and noise beyond what can be captured with the physical model of Eq. (3), quantifying the degree of success of the training is more difficult in this case. For this reason, we use the cross entropy to quantify the ability of the trained models to capture the qubit dynamics based on their predictions for the projective measurement acquired at the end of every experiment.

Figure 4a) shows the cross entropy obtained using three different approaches: the physical model of Eq. (3) with independently calibrated parameters (gray), SDE learning (blue), and physics-inspired RNN trained using $\mathcal{L}_\text{PI}$ (red). As an illustration of the outcome of training our ML models on experimental data, Fig. 4b) shows a representative quantum trajectory reconstructed with these three approaches while using the same input weak-measurement data. Comparing the gray and leftmost blue bars in Fig. 4a), we see that the SDE-learning approach allows us to learn physical parameters that describe the experimental outcomes significantly better than inferring these same parameters, $\{\Omega_R, \Gamma_d, \eta\}$ appearing in Eq. (3), using traditional calibration experiments [45]. Part of this discrepancy between the calibrated model and the learned SDE model can be attributed to calibration errors and experimental drifts which are likely to occur during the experiment. The better performance of the SDE learning demonstrates the ability of this approach to learn an accurate physical model of the quantum device.

Figure 4a) also compares the performance of three SDE models (blue bars) that have an increasing number of degrees of freedom from left to right. The leftmost blue bar is obtained by learning the parameters $\{\Omega_R, \Gamma_d\}$ of the Hamiltonian $H_R = \Omega_R/2 \sigma_x$ and Lindblad operator $L = \sqrt{\Gamma_d/2} \sigma_x$, together with the quantum efficiency $\eta$. A significant gain in accuracy can be obtained by leaving unconstrained the form of the Hamiltonian and Lindblad operators and learning these full operators $\{H_R, L\}$ (middle blue bar) directly from the data. Finally, taking advantage of the flexibility of the SDE-learning approach, we can further enrich the model that is learned, for example by adding to Eq. (3) qubit excitation and relaxation described by the additional dissipators $\gamma_1 \mathcal{D}[\sigma_+ \rho] dt + \gamma_2 \mathcal{D}[\sigma_-] \rho dt$. As can be seen from the right-most blue bar, the gain in predictive power is, however, marginal reflecting the fact that this particular addition is not capturing a meaningful contribution to the dynamics of our physical system. As presented with these three models, the ability to directly learn SDEs with growing parameter sets and use model selection to determine the best physical description that balances between predictive power and economy in parameters [46].

The performances of the SDE models are further compared in Fig. 4a) with a physics-inspired RNN trained on the same weak-measurement data (red bar). The RNN has a greater expressivity and is able to achieve a lower cross entropy since it is free from the constraints of the SDE-learning model. This result shows that some qubit dynamics are captured by the RNN but not by the presented SDE models. An advantage of SDE learning remains, however, in that it is a physically interpretable approach which accuracy can be improved by adding dynamics to the physical model describing the device. For example, it would be possible to include the readout cavity mode in the SDE model in order to characterize the non-Markovian qubit dynamics that are likely to be present in the experiment [30].

Another advantage of SDE learning is that it requires significantly less experimental training data. This is illustrated in Fig. 5 which shows the cross entropy as a function of the size of the training dataset. We see that
Cross Entropy δ
ment outcome within a small predicting the same probability Π measure consists in first binning the learned trajectories measure of self consistency used in Refs. [22, 47]. This intuitive way than the cross entropy, Fig. 4c) presents the ing and physics-inspired RNN in a less rigorous but more ever able to infer the known quantum features imposed Given enough training data, the black-box RNN is how- to improve with millions of training samples. Addition- ally, Fig. 5 shows that the physics-inspired loss LPI allows the RNN to learn more accurate quantum trajectories with fewer data than using the physics-agnostic loss LCE. Given enough training data, the black-box RNN is how- ever able to infer the known quantum features imposed to the physics-inspired model in order to perform equally well on the cross entropy, albeit this BBNN outputs non- positive quantum trajectories.

SDE learning (blue line) requires more than a order of magnitude fewer training samples than RNNs (light blue and red lines) to reach the same prediction accuracy. Of course, as already pointed out, the finite expressive power of the SDE model limits its maximal accuracy, which reaches a plateau while the RNN performance’s continue to improve with millions of training samples. Addition- ally, Fig. 5 shows that the physics-inspired loss LPI allows the RNN to learn more accurate quantum trajectories with fewer data than using the physics-agnostic loss LCE. Given enough training data, the black-box RNN is however able to infer the known quantum features imposed to the physics-inspired model in order to perform equally well on the cross entropy, albeit this BBNN outputs non-positive quantum trajectories.

To illustrate the prediction accuracy of the SDE learning and physics-inspired RNN in a less rigorous but more intuitive way than the cross entropy, Fig. 4c) presents the measure of self consistency used in Refs. [22, 47]. This measure consists in first binning the learned trajectories predicting the same probability Πn for the final measurement outcome within a small δ = 0.04. We then compare this approximate prediction to the average final projective measurement outcome (i.e. the tomography result) for the same set of trajectories. Both the SDE learning and RNN models are in excellent correspondence and follow the expected unit slope. The overall agreement between the model predictions and the tomographic mea- surements is quantified by the root-mean-squared error ε weighted by the number of trajectories in each bin. All the same, this measure is not as sensitive as the cross entropy which compares every single learned quantum trajectory to its associated projective measurement.

FIG. 4. Performance of the ML models to predict experimental outcomes. a) Cross entropy measure on the test dataset for the physical model of Eq. (3) using independently calibrated parameters (gray) or learning the parameters from the weak measurement data in a SDE model (light blue), additionally learning the Hamiltonian and Lindblad operators (blue) and qubit relaxation rates (dark blue). The performance of a physics-inspired recurrent neural network (PINN) trained with the loss function of Eq. (9) is also shown (red). b) A quantum trajectory reconstructed from an input weak measurement time-series sampled from the test dataset for three distinct models of panel a). c) More visual comparison of the model predictions with the observed tomography measurements for the same three models. The three tomography axes are used and averaged over. Also presented is the root mean squared error ϵ of the model predictions for all points extracted from the validation dataset (total of 0.68 M trajectories). The error bars are given by the standard deviation of the binomial distribution of model predictions for each point.

FIG. 5. Model performance as measured by the cross entropy on the test set as a function of the number of experimental weak-measurement training samples. The same RNN model is trained on the generic loss function Eq. (5) in light blue and on the physics-inspired loss function Eq. (9) in red. The SDE-learning model in blue has the structure of Eq. (3) with the physical model of Eq. (3) using independently calibrated parameters (gray) or learning the parameters from the weak measurement data in a SDE model (light blue), additionally learning the Hamiltonian and Lindblad operators (blue) and qubit relaxation rates (dark blue). The performance of a physics-inspired recurrent neural network (PINN) trained with the loss function of Eq. (9) is also shown (red).
VIII. ANALYSIS OF DEVICE PARAMETERS

We now study the device parameters learned from our two machine-learning approaches, namely the SDE-learning and the physics-inspired RNN. The SDE model can be trained directly on the weak-measurement data to estimate its free and interpretable parameters. On the other hand, the representation learned by the RNN is not as straightforward to interpret. Here, we propose an efficient way to extract the device parameters using the SDE-learning approach on the RNN-learned quantum trajectories.

An overview of the different parameter inference approaches considered in this work is presented in Fig. 6a), along with the parameters that are learned and the cross-entropy loss achieved on experimental data. Figure 6b) shows the parameters \( \{\Omega_R, \Gamma_d, \eta\} \) obtained from these approaches on numerical data (left panel) and experimental data (right panel). By comparing the darkest blue bar (true values used to generate the numerical data) to the second blue bar (SDE learning) in the left panel, we see that the SDE learning is able to infer the device parameters with an accuracy limited only by numerical errors introduced in the SDE integration. As presented in Appendix C, this error is proportional to the time step \( \Delta t \) which we take to be the same as for the experimental data, namely \( \Delta t = 0.04 \mu s \). The impact of coarse-graining the weak-measurement data is analyzed in Appendix C. Figure 7 further confirms the excellent performance of SDE learning on numerical data by showing the full Hamiltonian (top row) and Lindblad operators (bottom row) that are learned. These Hinton diagrams directly show the learned 2 \( \times \) 2 matrices. Comparing the first two columns of this figure, we confirm that the characterization obtained on numerical data matches very well with the true values.

On the other hand, the parameters obtained from SDE learning on experimental data show larger deviation from the values obtained from independent calibration of the device, see the right panel of Fig. 6b). The same is true for the full Hamiltonian and Lindblad operators shown in Fig. 7; compare the last two columns. The deviations of the inferred parameters from their ideal values are indicative of the need for a more sophisticated model [48]. Further analyzing these discrepancies between the calibrated and learned models could lead to a better understanding of the qubit dynamics. This interpretability of the SDE-learning approach constitutes one of its important advantage as opposed to black-box learning approaches.

Motivated by interpreting the model achieving the lowest cross entropy on the experimental data, we now analyze how we can physically interpret a RNN trained on the weak measurements. The approach to do so used in Refs. [22, 49] consists of two steps. First, the output trajectories are binned within some small expectation value intervals \( \delta \) at every time point. Second, the mean and variance of these distributions are fitted to a SME model using least squares fits, see Refs [22, 49] for more details. Device parameters resulting from this approach are shown in Fig. 6b) with the yellow bars. In the limit of vanishing bin size and many trajectories in each bin this approach will recover the true device parameters. Unfortunately, in the finite-data regime the fits are sensitive to the particular choice of bin size, with larger bin sizes discarding information due to coarse graining and smaller bin sizes suffering from reduced sampling statistics.

To improve upon this method, we employ the SDE-learning approach to interpret the learned representation of the RNN. This is achieved by training the SDE model to output quantum trajectories that maximize the likelihood of matching the learned RNN trajectories. This training task is much simpler than training on the projective measurement outcomes, since we now have direct access to the target quantum trajectories outputted by the RNN. To minimize the distance between the SDE

\[
\begin{align*}
\text{Parameters} & \quad \text{Exp. CEL} \\
\{\Omega_R, \Gamma_d, \eta\} & \quad 0.6505 \\
\{H_R, L\} & \quad 0.6478 \\
\{\Omega_R, \Gamma_d, \eta\} & \quad 0.6517 \\
\{H_R, L\} & \quad 0.6504 \\
\{H_R, L\} & \quad 0.6489
\end{align*}
\]

FIG. 6. Characterizing quantum dynamics from weak-measurement data. a) Schematic of the different approaches to infer qubit parameters. On the right hand side, the cross entropy loss (CEL) of the different models on the experimental test set are presented. A lower CEL implies a better prediction of the actual measurement outcomes. The free parameters of the models are shown in braces. For the last three approaches, a RNN is first trained on the weak-measurement data using the physics-inspired loss function Eq. (9), achieving a CEL of 0.6457, and the parameters are extracted from the RNN output trajectories. b) Device parameters extracted from the different learning approaches on numerically generated data (left panel) and on experimental transmon data (right panel). Note that we consider the parameters \( \Omega_R/2 (\sqrt{\Gamma_d/2}) \) to be the \( \sigma_x \) (\( \sigma_z \)) component of \( H_R \) (\( L \)). The error bars for the SDE approaches are extracted by comparing performance on numerical data, see Appendix C for more details. The error bars of the calibration and binning approaches are extracted directly from curve fits.
We have demonstrated that leveraging quantum mechanics in the design of a machine-learning approach makes the characterization of quantum dynamics more efficient and accurate when using limited experimentally available data. We have analyzed both numerically generated and experimental data obtained with a superconducting qubit, showing the tradeoffs between the accuracy of the learned description, its efficiency in the number of training samples required and its interpretability that allows such a description to be useful for the characterization of the quantum device. We have presented two main approaches to acquire an accurate heuristic of the quantum dynamics from weak-measurement data, namely i) by designing the loss function of a recurrent neural network to make use of our understanding of the quantum formalism and ii) by tailoring the structure of the machine learning model to respect the structure of the stochastic differential equation describing the device dynamics.

Overall, our results demonstrate that useful insights about the physics of quantum systems can be gained by interpreting machine-learning models trained on experimental data, thus going further than using black-box learning approaches. In particular, the SDE-learning constitutes a promising avenue to perform realistic qubit modeling and characterization via a single weak-measurement experiment. Notably, this approach allows us to characterize the quantum efficiency of the measurement chain η, which usually requires involved characterization [51]. The ideas developed in this work can be naturally extended to other quantum tasks involving, for example, noise characterization and optimal quantum control. Our characterization method is also advantageous for continuous quantum feedback control, where controls are applied based on continuous monitoring of the system [52].

An objective of the machine-learning approaches introduced in this work is to acquire an efficient representation of complex quantum devices to improve their characterization, calibration and control. Towards this end, it will be interesting to extend the learning to more qubits and to explore the characterization of additional quantum phenomena occurring in these more complex devices, such as non-Markovian dynamics and crosstalk. The higher training efficiency of our quantum-tailored machine-learning approaches might become especially relevant when scaling up these quantum tasks, where the limitations of available experimental data are increasingly stringent.

ACKNOWLEDGMENTS

We thank Simon Verret for insightful discussions. This work was undertaken thanks in part to funding from NSERC, the Canada First Research Excellence Fund, the
FRQNT, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF-17-S-0008.

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Reviews of Modern Physics 91, 045002 (2019).

[2] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: a review of recent progress, Reports on Progress in Physics 81, 074001 (2018).

[3] B. Lienhard, A. Vepsäläinen, L. C. G. Govia, C. R. Hoffer, J. Y. Qiu, D. Ristè, M. Ware, D. Kim, R. Winik, A. Melville, B. Niedzielski, J. Yoder, G. J. Ribeill, T. A. Ohki, H. K. Krovi, T. P. Orlando, S. Gustavsson, and W. D. Oliver, Deep Neural Network Discrimination of Multiplexed Superconducting Qubit States, arXiv:2102.12481 (2021).

[4] A. Seif, K. A. Landsman, N. M. Linke, C. Figgatt, C. Monroe, and M. Hafezi, Machine learning assisted readout of trapped-ion qubits, Journal of Physics B: Atomic, Molecular and Optical Physics 51, 174006 (2018).

[5] Z.-H. Ding, J.-M. Cui, Y.-F. Huang, C.-F. Li, T. Tu, and G.-C. Guo, Fast High-Fidelity Readout of a Single Trapped-Ion Qubit via Machine-Learning Methods, Physical Review Applied 12, 014038 (2019).

[6] M. Ostaszewski, J. A. Miszczak, P. Sadowski, and L. Banchi, Approximation of quantum control correction scheme using deep neural networks, Quantum Information Processing 18, 126 (2019).

[7] M. Bukov, A. G.-R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, Physical Review X 8, 031086 (2018).

[8] Q. Xu and S. Xu, Neural network state estimation for full quantum state tomography, arXiv:1811.06654 (2018).

[9] G. Torlai, M. Zaccaria, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nature Physics 14, 447 (2018).

[10] J. Carrasquilla, Machine learning for quantum matter, Advances in Physics: X 5, 1797528 (2020).

[11] A. Youssry, G. A. Paz-Silva, and C. Ferrie, Characterization and control of open quantum systems beyond quantum noise spectroscopy, npj Quantum Information 6, 1 (2020).

[12] M. Hibat-Allah, M. Gainahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Recurrent neural network wave functions, Phys. Rev. Research 2, 023558 (2020).

[13] S. Morawetz, I. J. S. De Vlugt, J. Carrasquilla, and R. G. Melko, U(1) symmetric recurrent neural networks for quantum state reconstruction, arXiv:2010.14514 (2020).

[14] A. Valenti, E. Greplova, N. H. Lindner, and S. D. Huber, Correlation-Enhanced Neural Networks as Interpretable Variational Quantum States, arXiv:2103.05017 (2021).

[15] J. Preskill, Quantum Computing in the NISQ era and beyond, Quantum 2, 79 (2018).

[16] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, Quantum certification and benchmarking, Nature Reviews Physics 2, 382 (2020).

[17] I. Šupić and J. Bowles, Self-testing of quantum systems: a review, Quantum 4, 337 (2020).

[18] J. Carrasco, A. Elben, C. Kokail, B. Kraus, and P. Zoller, Theoretical and Experimental Perspectives of Quantum Verification, PRX Quantum 2, 010102 (2021).

[19] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, Charge-insensitive qubit design derived from the cooper pair box, Phys. Rev. A 76, 042319 (2007).

[20] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, Circuit quantum electrodynamics, Rev. Mod. Phys. 93, 025005 (2021).

[21] H. Wiseman and G. Milburn, Quantum Measurement and Control (Cambridge University Press, 2010).

[22] E. Flurin, L. Martin, S. Hacohen-Gourgy, and I. Siddiqi, Using a Recurrent Neural Network to Reconstruct Quantum Dynamics of a Superconducting Qubit from Physical Observations, Physical Review X 10, 011006 (2020).

[23] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory, Robust online hamiltonian learning, New Journal of Physics 14, 103013 (2012).

[24] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Hamiltonian Learning and Certification Using Quantum Resources, Physical Review Letters 112, 190501 (2014).

[25] T. J. Evans, R. Harper, and S. T. Flammia, Scalable Bayesian Hamiltonian learning, arXiv:1912.07636 (2019).

[26] M. Paris and J. Rehacek, eds., Quantum State Estimation, Lecture Notes in Physics (Springer-Verlag, Berlin Heidelberg, 2004).

[27] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, Efficient quantum state tomography, Nature Communications 1, 149 (2010).

[28] D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert, Quantum State Tomography via Compressed Sensing, Physical Review Letters 105, 150401 (2010).

[29] G. Torlai, C. J. Wood, A. Acharya, G. Carleo, J. Carrasquilla, and L. Aolita, Quantum process tomography with unsupervised learning and tensor networks, arXiv:2006.02424 (2020).

[30] J. Gambetta, A. Blais, M. Boissonneault, A. A. Houck, D. I. Schuster, and S. M. Girvin, Quantum trajectory approach to circuit QED: Quantum jumps and the Zeno effect, Physical Review A 77, 012112 (2008).

[31] C. L. Hutchison, J. M. Gambetta, A. Blais, and F. K. Wilhelm, Quantum trajectory equation for multiple qubits in circuit qed: Generating entanglement by measurement, Can. J. Phys. 87, 225 (2009).

[32] P. Rouchon and J. F. Ralph, Efficient Quantum Filtering for Quantum Feedback Control, Physical Review A 91, 012118 (2015).

[33] J. R. Johansson, P. D. Nation, and F. Nori, QuTiP 2: A Python framework for the dynamics of open quantum systems, Computer Physics Communications 184, 1234
A recurrent neural network (RNN) is formed of a unit cell that is repeated at every new time step of the input data $\Delta M_t$, producing an output $\vec{h}_t$ known as a hidden state. This hidden state acts as a memory of the previous inputs and is combined with the next input of the time-series $\Delta M_{t+\Delta t}$ to allow information earlier in the sequence to influence outputs further down in the sequence. As such, the RNN implements a non-linear function, defined in terms of the network parameters $W$, that maps the previous hidden state and current input to the next hidden state. Importantly, the function implemented by the RNN is fully differentiable with respect to these parameters such that they can be updated at each training iteration. This process allows to bring the network output closer to the desired output. It is done by taking the gradients of a chosen loss function $\mathcal{L}$ with respect to the network parameters using back-propagation and then updating them to minimize the loss function via gradient descent [34]

$$\mathbf{W} \rightarrow \mathbf{W} - \zeta \frac{\partial \mathcal{L}}{\partial \mathbf{W}},$$  \hspace{1cm} (A1)$$

where $\zeta$ is the learning rate. In the main text, we are using a specific RNN architecture named gated recurrent unit (GRU). After exploration of different well-known recurrent architecture that avoid the vanishing and exploding gradients problem, the GRU was found to be the most effective for the task at hand.

The specific function implemented by the GRU can be separated into three gates that are used to decide what information should be passed to the output. They are named reset $r_t$, update $z_t$ and new $n_t$ gates and are ex-
pressed as \[ [36] \]

\[
\begin{align*}
    r_t &= \sigma (W^{M,r} \Delta M_t + b^{M,r} + W^{h,r} h_{t-\Delta t} + b^{h,r}) , \\
    z_t &= \sigma (W^{M,z} \Delta M_t + b^{M,z} + W^{h,z} h_{t-\Delta t} + b^{h,z}) , \\
    n_t &= \tanh (W^{M,n} \Delta M_t + b^{M,n} + r_t \ast (W^{h,n} h_{t-\Delta t} + b^{h,n})) ,
\end{align*}
\]  

(A2)

where \( W \) and \( b \) are the free weights and biases parameters optimized during the training stage, \( \sigma(\cdot) \) is the sigmoid function and \( \ast \) is the Hadamard product. The hidden state is then computed by combining these three gates in the following way

\[
h_t = (1 - z_t) * n_t + z_t * h_{t-\Delta t}.
\]  

(A3)

Further information about the origin, possible performances and limitations of this architecture can be found in \([34–36]\].

Appendix B: Machine-learning trainings

In Fig. 8, we present a typical training of a GRU and a SDE-learning model on numerically generated data. The quantum trajectories and their associated weak-measurement data \( \Delta M_t \) were obtained numerically by integrating the stochastic master equation of Eq. (3) for a total time \( T = 8 \) \( \mu \)s for different noise realizations.

Figure 8a) presents the training of the neural network illustrated in Fig. 1 with a GRU containing 16 hidden dimensions and implemented in PyTorch \([53]\). We use the loss function of Eq. (9) with weights \( w_{\text{posit}} = 0.36, \ w_{\text{prep}} = 1.7, \ w_{\Delta M} = 2.1 \). The training is done using the Adam optimizer \([54]\) with a learning rate \( \zeta = 0.001 \) and in batches of 1024 trajectories. We observed that the training quality does not strongly depend on the specific values of these hyperparameters. We see that, within the first few epochs, the GRU learns to do better than the average quantum trajectory prediction, which is given by integrating the master equation (ME) with the true parameters. The ME is given by the first line of the SME Eq. (3), i.e. by dropping the stochastic part. The high expressive power of the GRU allows it to overfit the training data after enough training epochs, which is seen in Fig. 8a) by the increase of the validation loss.

Figure 8b) presents the training of a stochastic differential equation integrator (SDE-learning) model that we implemented in PyTorch using the Milstein scheme presented in \([43]\). The specific SDE used for our model is Eq. (3), which is the same as the equation integrated to generate our weak-measurement data, thus allowing the SDE-learning to learn the quantum dynamics with high accuracy. The small number of parameters of the SDE model allows us to train an ensemble of these models in parallel that each have their own, randomly initialized, free parameters. For example here, 100 models are trained simultaneously on a single Nvidia GeForce 2080 Ti with 11GB of memory. Figure 8b) shows the best model out of this ensemble. The SDE model is also optimized using Adam and the same values for the learning rate and batch size. Since the SDE model automatically satisfies the quantum-mechanical properties we were trying to encourage with additional loss-function terms, the SDE-learning is trained solely on the cross-entropy loss Eq. (5). We highlight the fact that the SDE-learning model is given the same inputs, which are realistically available in an experiment, as the GRU. These inputs are the weak-measurement time-series tensors \( \Delta M_t \), together with
The finite step size $\Delta t$ of the time-series, experimentally limited by the sampling time of the weak measurements, intrinsically limits our ability to reconstruct quantum trajectories by integrating a stochastic master equation. Indeed, some information is lost by integrating the weak-measurement signal in time steps $\Delta t$. Throughout this work, we use $\Delta t = 0.04 \mu s$ for the experimental and numerical data. This coarse-graining of the gathered $\Delta t = 0.001 \mu s$ time-series is motivated by two practical considerations. First, the computing time and memory requirements associated with having a small time step are important for long sequences. Using this time step allows us to train all of our machine-learning models using less than a couple hours and 11 GB of GPU RAM. Second, we want to avoid the correlations in the time-series that are due to the finite memory time of the readout cavity ($1/\kappa \approx 0.10 \mu s$), since our SDE models assume uncorrelated weak measurements.

In order to explore the effect of this coarse graining on the ability of the SDE-learning model to learn accurate device parameters, we train this model on synthetic data coarse grained to different $\Delta t$. The training data contains 75,000 trajectories obtained by integrating the following SME

$$d\rho_t = -i \frac{\Omega_R}{2} [\sigma_z, \rho_t] dt + \frac{\Gamma_d}{2} d[\sigma_z] \rho_t dt + \sqrt{\eta \Gamma_d} \left( \hat{H} [\sigma_z] \rho_t dW_t^I + \hat{H} [-i \sigma_z] \rho_t dW_t^Q \right),$$

(C1)

for variable durations $T$ between 0 $\mu s$ and 8 $\mu s$ and with a resolution $\Delta t = 0.001 \mu s$. From these trajectories, the other training datasets are generated by combining weak measurements in order to coarse grain the data to larger time steps, up to $\Delta t = 0.2 \mu s$. We do the same for the validation set with half as many trajectories as the training set. We note that we generate all of these trajectories with a Milstein integration scheme, which corresponds to the scheme used by our SDE-learning model.

In Fig. 9, we quantify the effect of the integration errors on the SDE-learning achievable accuracy for different coarse graining of the weak-measurement time-series. We present both the relative error of the learned parameters $\{\Omega_R, \Gamma_d, \eta\}$ and the associated prediction accuracy of the SDE model, as given by the validation cross-entropy loss (CEL) error. Figure 9a) shows that we can learn the true physical rates and quantum measurement efficiency within a relative error close to 1% or less. We attribute this finite precision to the limited size of the training dataset and the use of mini-batches during training, which both introduce statistical sampling errors. However, this precision gets worse with increasing $\Delta t$, thus confirming that the integration errors introduced by the coarse-graining is a limitation for the SDE-learning model, and more generally to any physical modeling of the data involving the numerical integration of a differential equation.

In Fig. 9b), we present the isolated effect of the integration errors on the cross entropy loss achieved by a SDE model. To do so, we integrate the validation trajectories using the true values of the parameters $\{\Omega_R, \Gamma_d, \eta\}$ for the differently coarse-grained datasets and compare these trajectories with the expected projective measurement outcomes. We see that the CEL error grows proportionally to the size of $\Delta t$ for the SDE model with true parameters. When leaving these same parameters $\{\Omega_R, \Gamma_d, \eta\}$ free for the SDE model to learn during train-
ing, we observe that the SDE-learning model is able to perform better at predicting the measurement outcomes of the validation set. We attribute this effect to the fact that the model has the freedom to find parameters that account for part of the information-loss due to coarse-graining the data. For example, learning a slightly lower quantum measurement efficiency \( \eta \) than the true value, on coarse-grained data, allows the SDE model to achieve a better CEL. Consequently, the finite size of \( \Delta t \) can bias the parameter estimation and thus limit the ability of the SDE-learning to extract accurate device parameters. As such, we use this bias found in training on numerical data as a lower bound on the error associated with learning the device parameters on experimental data with \( \Delta t = 0.04 \mu s \). We use this error for the error bars of the SDE-learning in Fig. 6 of the main text, since it is larger than the standard deviation of the parameters learned in the ensemble of 100 SDE models.

**Appendix D: Parameter values**

For completeness, we present here the numerical values illustrated in Fig. 6 of the main text.

|                          | True or Calib. | SDE + SDE \{H, L, \eta\} | RNN + SDE + SDE \{\Omega_R, \Gamma_d, \eta\} | RNN + Binning |
|--------------------------|----------------|--------------------------|---------------------------------------------|---------------|
| **Numerical data**       |                |                          |                                             |               |
| \( \Omega_R/2\pi \) (MHz) | 0.222          | 0.220                    | 0.221                                       | 0.221         |
| \( \Gamma_d/2\pi \) (MHz) | 0.187          | 0.194                    | 0.195                                       | 0.193         |
| \( \eta \) (%)          | 14.7           | 13.4                     | 13.5                                        | 13.7          |
| Cross entropy            | 0.64510        | 0.64512                  | 0.64513                                     | 0.64511       |
| **Experimental data**    |                |                          |                                             |               |
| \( \Omega_R/2\pi \) (MHz) | 0.222          | 0.228                    | 0.228                                       | 0.232         |
| \( \Gamma_d/2\pi \) (MHz) | 0.187          | 0.182                    | 0.176                                       | 0.184         |
| \( \eta \) (%)          | 14.7           | 14.5                     | 13.8                                        | 13.3          |
| Cross entropy            | –              | 0.6478                   | 0.6489                                      | 0.6504        |

**TABLE I.** Different approaches to learning device parameters from weak measurements in the cases of synthetic and experimental data. The RNN of the last three columns is a single model trained on the weak measurement data and the parameters are extracted from its output quantum trajectories. The cross entropy is a measure of the accuracy of the model and associated parameters to describe the observed measurement outcomes. Note that the binning approach assumes the same physical model as the RNN + SDE \{\Omega_R, \Gamma_d, \eta\} model, i.e. it assumes that \( H_R = \Omega_R/2\sigma_x \) and \( L = \sqrt{\Gamma_d/2}\sigma_z \) in Eq. (3).