

Noise Detection with Spectator Qubits and Quantum Feature Engineering

Akram Youssry,^{1,2} Gerardo A. Paz-Silva,³ and Christopher Ferrie¹

¹*Centre for Quantum Software and Information,
University of Technology Sydney, Ultimo NSW 2007, Australia*

²*Department of Electronics and Communication Engineering,
Faculty of Engineering, Ain Shams University, Cairo, Egypt*

³*Centre for Quantum Dynamics and Centre for
Quantum Computation and Communication Technology,
Griffith University, Brisbane, Queensland 4111, Australia*

(Dated: March 25, 2021)

Designing optimal control pulses that drive a noisy qubit to a target state is a challenging and crucial task for quantum engineering. In a situation where the properties of the quantum noise affecting the system are dynamic, a periodic characterization procedure is essential to ensure the models are updated. As a result, the operation of the qubit is disrupted frequently. In this paper, we propose a protocol that addresses this challenge by making use of a spectator qubit to monitor the noise in real-time. We develop a quantum machine-learning-based quantum feature engineering approach for designing the protocol. The complexity of the protocol is front-loaded in a characterization phase, which allow real-time execution during the quantum computations. We present the results of numerical simulations that showcase the favorable performance of the protocol.

I. INTRODUCTION

As quantum technology progresses more towards the Noisy Intermediate-Scale Quantum (NISQ) devices era [1], the design and operation tasks become more challenging. One such task is quantum control, where it is required to find a sequence of control pulses that drive a qubit, in order to achieve a desired target such as dynamical decoupling and dynamically-corrected gates [2–8]. The standard approach is to divide the problem into two stages. The first is a characterization stage, which aims to produce models of the noise based on experimental measurements. Quantum Noise Spectroscopy (QNS) is one example where the Power Spectral Density (PSD) of the noise is estimated from coherence time measurements [9–29]. The second stage is Optimal Control (OC), in which a cost function is optimized with respect to the control [30–35].

The main drawback of many techniques is the use of master equations to model noisy dynamics, which are only valid under certain assumptions and/or approximations (such as Markovianity, or weak-coupling). In actual devices, these assumptions and approximations may not hold [36]. Additionally, many techniques assume ideal control pulses (for example instantaneous or unlimited bandwidth). In practice, the non-idealities will affect the performance of the device if not accounted for during the design process. A Machine Learning (ML) approach has been proposed recently to address these drawbacks in [37, 38]. The idea is to construct graybox structures that consists of standard blackbox layers (such as Neural Networks (NN)), as well as custom whitebox layers that encode quantum operations (such as quantum evolution). The advantage of this approach is that it allows having assumptions-free models because of the use of the blackbox layers. At the same time, it allows estimating physically-significant quantities (such as Hamiltonians), satisfying their mathematical constraints (such as Hermiticity) due to the use of whiteboxes. The models are trained from experimental measurements, and can then be used to design the control.

In a typical experiment, the noise behavior can vary over time. In this case, a characterization step at the beginning is not sufficient, and has to be repeated periodically over the course of operation. As a result, the total time during which the qubit operates usefully (i.e. executing quantum computations) reduces. One solution to this problem is through the use of spectator

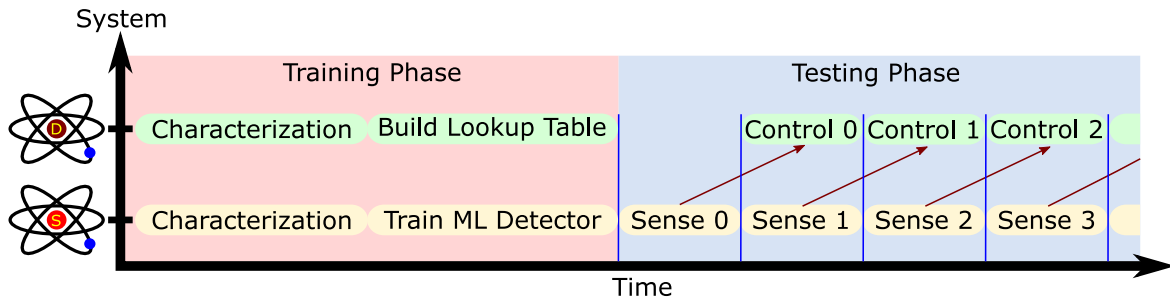


Figure 1: Timing diagram of the proposed protocol. The upper qubit labeled ‘D’ is the data qubit, and the lower qubit labeled ‘S’ is the spectator qubit. At the beginning there is a training phase that is executed once which includes characterization of the qubits and training the machine learning models. This is followed by the testing phase in which a periodic cycle of measurements of the spectator qubit inform the optimal control of the data qubit.

qubits [39, 40]. The idea is to use another qubit, the “spectator”, as a sensor to the noise, instead of stopping the quantum computations on the main “data” qubit to do the sensing. The spectator can be a low-quality qubit (and thus easier to design) as its only role is sensing the noise. Figure 1 shows a typical timing diagram for such a protocol. In this paper, we address the problem of designing a sensing protocol utilizing a spectator qubit for noise detection [41, 42], i.e. there is a discrete set of possible noise profiles that can affect the qubit at any given time. This opposes noise estimation, in which there is an infinite set of possibilities for the noise profiles. We will utilize a “Quantum Feature Engineering” (QFE) approach, extending the proposal in [37] and drawing analogy from the paradigm of “feature extraction” in classical machine learning.

The rest of this paper is organized as follows. The paper starts with the problem setting and the underlying assumptions in Section II. Next, in Section III we introduce the steps of the protocol in detail as well as the design of the ML models it utilizes. After that, we give details on the numerical simulations we implemented to verify the proposed ideas in section IV. Finally, we conclude paper and show the potential extensions of the presented work in Section V.

II. PROBLEM SETTING

The basic setup of our problem is as follows. The data qubit is subject to a Hamiltonian in the form

$$H_D(t) = H_D^{(\text{ctrl})}(t) + H_D^{(1)}(t). \quad (1)$$

The first term is known and deterministic, and represents the drifting and control Hamiltonians given by

$$H_D^{(\text{ctrl})}(t) = \frac{1}{2}\Omega_D\sigma_z + \frac{1}{2} \sum_{k \in \{x,y,z\}} f_D^{(k)}(t)\sigma_k, \quad (2)$$

where Ω_D is the energy gap of the qubit, σ_k are the Pauli matrices, and $f_D^{(k)}(t)$ is the control pulse sequence driving the qubit along the k^{th} direction. On the other hand, the second term $H_D^{(1)}$ is an unknown stochastic Hamiltonian that encodes the noise affecting the qubit due to interaction with the environment. We can express it generally in the form

$$H_D^{(1)}(t) = \sum_{k \in \{x,y,z\}} \beta_D^{(k)}(t)\sigma_k, \quad (3)$$

where $\beta_D^{(k)}(t)$ is realization of the noise random process $\mathbf{B}_D(t)$ along the k^{th} direction. In this paper, we focus on the setting in which there exists a discrete set of possible processes (profiles) that could affect the qubit at any point in time. In other words,

$$\mathbf{B}_D(t) \in \left\{ \mathbf{D}_k \right\}_{k=1}^N. \quad (4)$$

Similarly, the spectator qubit is subject to a Hamiltonian of the form,

$$H_S(t) = H_S^{(\text{ctrl})}(t) + H_S^{(1)}(t), \quad (5)$$

where

$$H_S^{(\text{ctrl})}(t) = \frac{1}{2}\Omega_S\sigma_z + \frac{1}{2} \sum_{k \in \{x,y,z\}} f_S^{(k)}(t)\sigma_k. \quad (6)$$

The spectator qubit energy gap Ω_S and the control pulses $f_S^{(k)}(t)$ need not be the same as the data qubit. The noise Hamiltonian of the spectator can also be written in the form

$$H_S^{(1)}(t) = \sum_{k \in \{x,y,z\}} \beta_S^{(k)}(t)\sigma_k, \quad (7)$$

where $\beta_S^{(k)}(t)$ is a realization of a random process $\mathbf{B}_S(t)$ along the k^{th} direction. There is also a discrete set of possible noise profiles that affects the spectator qubit

$$\mathbf{B}_S(t) \in \left\{ \mathbf{S}_k \right\}_{k=1}^N. \quad (8)$$

We assume the existence of a fixed bijective map \mathcal{F} , known a priori, between the noise profiles affecting the spectator qubit and those affecting the data qubit,

$$\mathcal{F} : \left\{ \mathbf{S}_k \right\}_{k=1}^N \rightarrow \left\{ \mathbf{D}_k \right\}_{k=1}^N. \quad (9)$$

Therefore, if we execute the noise detection protocol on the spectator and find the noise profile to be \mathbf{S}_j , then we can know for certain that the noise profile affecting the data qubit is $\mathcal{F}(\mathbf{S}_j)$. A physical example of this situation is in superconducting qubits. The data and spectator qubits can be tuned differently such that for one of them we have a dephasing noise in the form of $H_S^{(1)} = \beta(t)\sigma_z$ and for the other it is in the form $H_D^{(1)} = c\beta(t)\sigma_z$. It is important to ensure that the map is a bijection, to avoid the situation where multiple spectator noise profiles map to the same data noise profile. In practice, this map can be characterized using physical modeling or using more sophisticated techniques including ML, however this is out of the scope of this paper. We need the following additional assumptions:

1. The size the profiles set N is fixed.
2. The statistical properties of each individual profile \mathbf{S}_k are fixed (can be unknown).
3. The measurements and control are fast enough such that the noise profile does not change during measurements or during execution of a quantum gate.
4. It is possible to characterize each noise profile separately at the beginning of the protocol.

The first two assumptions are necessary in any signal detection problem, that is the existence of fixed well-defined classes to which the signal can belong. The third assumption is fair. If the noise switches between two profiles faster than the measurement or control time, then they should not be treated as separate profiles, but rather one profile with non-stationary statistics.

Note that any kind of statistics are allowed for the profiles \mathbf{S}_k . The last assumption is the most challenging one. In order to characterize each noise profile separately, we must know exactly which profile is currently affecting the qubit. But this is actually the problem that we want to solve, so we have a circular definition situation. However, it is possible to resolve this issue by repeatedly performing the characterization process until we observe all possible noise profiles. This process might take a long time, but it is done only at the beginning before the actual operation of the system. Once we do the characterization, we can address noise detection problem, where we need to know which noise profile is currently active as quick as possible using the minimum number of measurements. A practical example is a solid-state or a superconducting qubit operating at cryogenic temperatures. Every time the device is cooled down, we end up with two-level fluctuators, that effectively act as the environment of the qubit. There might be a change in the physical properties of the fluctuators after each cooling. Thus, we repeatedly cool down the device and characterize it until we observe all possible noise profiles. After that, during the operation of the device (or perhaps other devices of the same type), we do not need to repeat this lengthy procedure.

In an experimental setting, it is only possible to measure the expectation value $\langle O(T) \rangle$ of some observable O at time $t = T$. In what follows, we focus on the spectator qubit, however similar expressions can be written for the data qubit. Assuming the spectator starts in the state $\rho_S(0)$, using a modified interaction picture (see [37] for a detailed derivation), we can write

$$\langle O(T) \rangle = \text{tr} \left(V_O(T) U_0(T) \rho_S(0) U_0^\dagger(T) O \right), \quad (10)$$

where U_0 is the unitary given by time-ordered expression

$$U_0(T) = \mathcal{T}_+ e^{-i \int_0^T H_S^{(\text{ctrl})}(t) dt}, \quad (11)$$

representing the evolution of the system in the absence of noise, and $V_O(T)$ is an operator that encodes all information about the noise and how it interacts with control and is given by

$$V_O(T) = O^{-1} \langle \tilde{U}_I(T)^\dagger O \tilde{U}_I(T) \rangle_c. \quad (12)$$

The classical expectation $\langle \cdot \rangle_c$ is taken the realizations of the noise process, and

$$\tilde{U}_I(T) = U_0(T) U_I(T) U_0^\dagger(T), \quad (13)$$

with

$$U_I(T) = \mathcal{T}_+ e^{-i \int_0^T U_0^\dagger(t) H_S^{(1)}(t) U_0(t) dt}. \quad (14)$$

The use of this formalism allows us to express the dynamics of the open quantum system exactly without approximation or assumptions on the noise or control (as in the case of master equations). The important note is that noise operator V_O depend on both the control and the noise. Thus, if the noise is fixed and the control is allowed to vary, then we can estimate experimentally the operator providing a suitable encoding of the noise.

So, under the aforementioned settings and assumptions, we propose the following research question:

Can we design a protocol to detect which noise profile is affecting the data qubit by performing a measurement on the spectator qubit?

The answer to this question is affirmative. We will develop a protocol that utilizes ML techniques to model the qubits, design the optimal control pulses, and design a noise discriminator. In the next section, we will explain the protocol in detail.

III. METHODS

A. Quantum feature engineering

In classical ML literature, “features” refer to vectors extracted from a signal, that can be used for a variety of applications including classification (assigning labels to signals). Features can be “raw” such as the color of a pixel in an image or the amplitude of an audio signal, or can be more abstract like the amplitude of a frequency component of a signal. There are three basic steps to prepare the features for a classifier:

1. Feature generation: computing the feature vector from a given signal.
2. Feature selection: choosing a subset that best distinguishes the objects we are classifying, based on some ranking criterion.
3. Feature extraction: applying transformations to enhance the distinguishability between the classes and also to reduce the dimensionality of the feature vector.

Once these steps are done, an ML blackbox structure is constructed to perform the classification process. The structure is trained by minimizing a loss function (such as the mean-square error (MSE)) with respect to the model parameters over a training set. The loss function captures the error between the predicted label by the model, and true label, for a given set of training examples. A training set thus consists of representative examples from each class, where an example is a pair of feature vector and the corresponding ground truth label. After the classifier is trained, it can then be used to classify new examples that are not part of the training set. This is referred to as the testing stage of the classifier. The training is usually a computationally expensive process, whereas the testing is very efficient. Traditionally, the process of feature generation, selection, and extraction are done manually using heuristics (see [43] for a standard text on the subject and [44] for an application in image processing). On the other hand, in the modern paradigm of deep learning, these steps become an integrated part of the ML structure, and the inputs become directly the raw signal. The structure includes trainable layers that generates and extracts the features automatically as a part of the training process. The algorithm learns the optimal features that need to be generated and how to transform them such that the loss function is minimized. An example of this structure is the Convolutional Neural Network (CNN).

The noise detection problem can be formulated using the language of features and classification. Since the underlying physical model is based on quantum mechanics, we will refer to this approach as “quantum feature engineering” (see Table I for the analogy with standard ML). We need to train a classifier using features that encode information about the noise, and at the same time accessible experimentally. The only possibility is the expectation of some observables. However, according to Equation 10, this depends on the noise, control, initial state, evolution time, and observable. The dependence on noise is necessary, otherwise we cannot choose the measurements as features in the first place. The evolution time can be chosen such that we are able to obtain useful information from the measurement. This means it is long enough for the state to evolve non-trivially, but not too long that the system completely decoheres. The initial state and the observable could be chosen to form an informationally-complete set. For a qubit, this would be six eigenstates of the Pauli operators as initial states, and all the Pauli operators as observable, which gives a total of 18 measurements. However, in many cases we can reduce this set for efficiency purposes. For example, if we know that the noise is pure dephasing along Z-axis, then it is sufficient to measure the Pauli X operator, with the three positive eigenstates of the Pauli operators as initial states. This gives a total of 3 measurements. A Pauli Z measurement would not reveal any information about the dephasing process. This step is analogous to feature selection in ML. The last item to fix is the control. Now, in principle we could have

two completely different noise profiles, but both result in the same measurements given a particular control. For example, in the case of ideal instantaneous pulses, and Gaussian stationary zero-mean dephasing noise, the measurement $\langle X(T) \rangle = e^{-\int S(\omega)|F(\omega)|^2 d\omega} \langle X(0) \rangle$, where $S(\omega)$ is the PSD of the noise, and $|F(\omega)|^2$ is the filter function. In this case, we could have two different noise spectra, but the filter function happens to overlap with them in a way that results in the same value of the integration, and thus the same measurement. Therefore, the choice of control is important to be able to distinguish between different noise profiles. One possible heuristic to use is the distance between the V_O operators corresponding to each noise profile. We can see that if the control is fixed, then similar to Equation 4 we can define the set of noise profiles using the V_O operators

$$V_O(T)|_{H_S^{(\text{ctrl})}(t)} \in \left\{ V^{(k)} \right\}_{k=1}^N, \quad (15)$$

where each $V^{(k)}$ corresponds to one profile. Therefore, we can use formulate this as a quantum control problem, where it required to design a control that maximally distinguishes between the V_O operators corresponding to each noise profile. The natural question to ask then is how to evaluate the V_O operators to use in an optimization algorithm. Experimentally, this quantity is not accessible. However, we can build estimating models from characterization information (set of random control pulses, and corresponding measurements).

B. Summary of the proposed protocol

1. spectator qubit

To sum up the proposed QFE approach, there are two different pipelines (phases) for training and testing. The training pipeline starts with the characterization information as a raw signal, from which the V_O operators are generated. An optional step would be feature selection where we select the combination of observables and initial states that will maximizes the information obtained from the measurements if we have some prior information about the noise. Next, we design an optimal control sequence to ensure maximum separation between classes (i.e. maximize the distance between each $V^{(k)}$), and evaluate the corresponding optimal measurements. This is the feature extraction step. Finally, a classifier is trained using the simulated optimal measurements and the correct label if the noise profile. The outcomes of this pipeline is the optimal control pulses and the trained classifier. The training pipeline will have a long execution time, however it is done once at the beginning of the protocol. The testing pipeline during which the device operates is more efficient. It starts by the experimentally measuring the observables corresponding to the the optimal control pulses. The measurements are then passed to the trained classifier to predict the current noise profile affecting the system.

We will group the different steps of the protocol into four stages. The first stage (Section III C) includes the characterization and estimation of the V_O operators. We are going to use an ML approach as in [37], but with a different design. The second stage presented in Section III D will include the optimal control pulse design. In Section III E, we explain the third stage which is training the classifier. Those three stages represent the training phase of the protocol. The last stage represents the testing phase of the protocol and is presented in Section III F. The proposed protocol is summarized in Figure 2.

2. data qubit

For the data qubit, We assume that a full characterization procedure is done at the beginning and so we are able to construct a lookup table of optimal control sequences. The lookup table

ML	QFE
Classes	Noise profiles
Raw signal	Characterization data
Feature generation	Estimating V_O
Feature selection	Selecting best observables and initial states
Feature extraction	Designing the optimal control pulses
Features	Predicted optimal measurements

Table I: The analogy between the proposed quantum feature engineering approach for addressing the noise detection problem, and classification

\mathcal{L} would be indexed by the noise profile and the desired quantum gate. In other words, the optimal control pulses $\mathbf{f}_D(t)$ to apply given the noise profile index n_D , and the desired gate G is

$$\mathbf{f}_D(t) = \mathcal{L}(n_D, G). \quad (16)$$

Usually, we optimize a universal set of gates and not every possible gate, and thus it is possible to construct this table. The details about how to construct such a table is out of scope of this paper, however the general method in [37] could be used by performing it for every possible noise profile. The characterization and lookup table construction represent the training phase of the data qubit and is independent of the spectator qubit. The testing phase however depends on the spectator. In this stage, once we perform the aforementioned steps on the spectator and find that the active noise profile is n_S , we simply use the map \mathcal{F} to find the noise profile index n_D that is affecting the data qubit. Consequently, we use the lookup table \mathcal{L} to find the optimal pulses we need to implement a gate G . This process will be repeated periodically over the time. The data qubit is never interrupted while executing the gates, which is the main objective of this paper.

The remainder of this paper, including the numerical simulations, will focus on the spectator qubit, and how the different stages are implemented. The protocol steps related to the data qubit are standard tasks (characterization and control), and so we will not focus on these aspects in this paper.

C. Stage I: Feature Generation

The focus in this paper is on the spectator qubit. We will construct a graybox ML structure [37, 38] to model the qubit. It consists of whitebox layers that performs quantum calculations and is able to generate features such as the V_O operators. Additionally, it has blackbox layers that can be trained to generate information about the noise for example. The combination of blackboxes and whiteboxes result in an overall graybox. The proposed structure is shown in Figure 3. The details are given next.

There are two main paths in the proposed ML structure. The first path is the control path. It starts with the model input which is the control pulse sequence $\mathbf{f}_S(t)$ represented in time domain. The control pulses then passes through a whitebox layer that constructs the Hamiltonian $H_S^{(\text{ctrl})}(t)$. After that, there is a whitebox that computes the control unitary $U_0(t)$ by approximating the time-ordered evolution in Equation 11 as

$$U_0(t) = e^{-iH_S^{(\text{ctrl})}(M\Delta t)\Delta t} \dots e^{-iH_S^{(\text{ctrl})}(2\Delta t)\Delta t} e^{-iH_S^{(\text{ctrl})}(\Delta t)\Delta t} e^{-iH_S^{(\text{ctrl})}(0)\Delta t}, \quad (17)$$

where $\Delta t = t/M$, and M is the number of discrete time steps.

The second path is the noise path. It starts with a custom blackbox that has two outputs. The first is a set of normalized weights $\{w_k\}_{k=1}^K$, such that $0 \leq w_k \leq 1$, and $\sum_k w_k = 1$. That

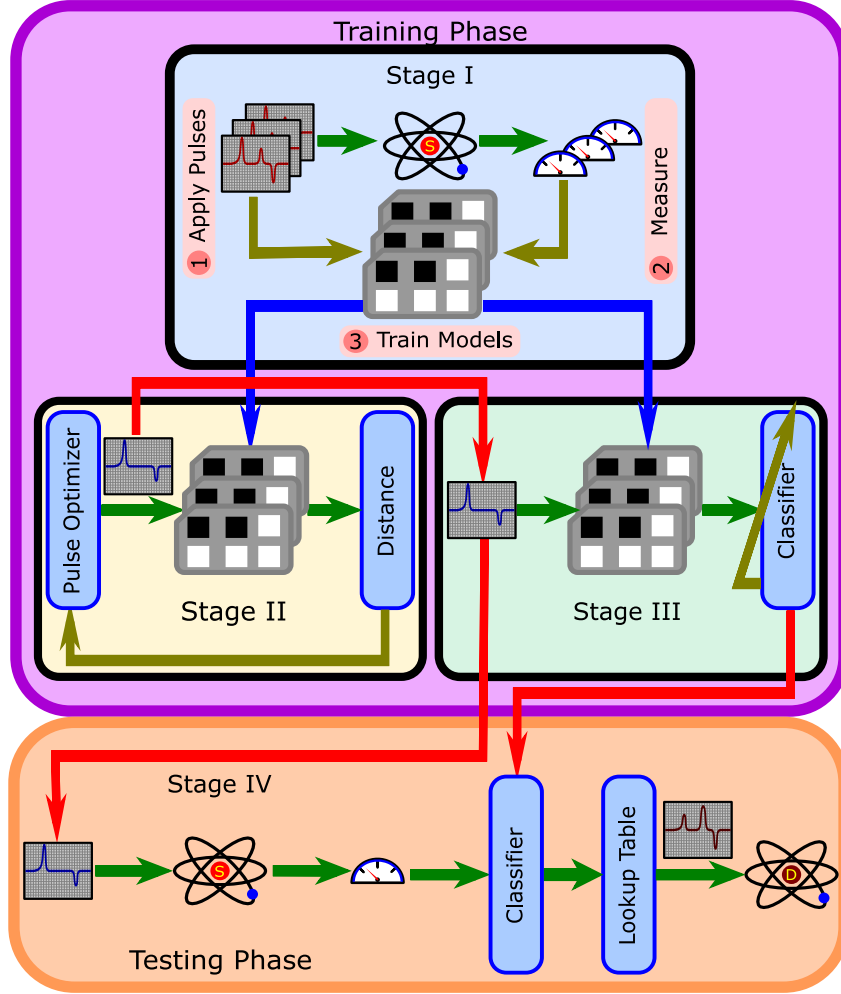


Figure 2: The proposed protocol for controlling a data qubit using measurements from a spectator qubit and quantum feature engineering. The first stage is for characterization and training ML models for estimating the V_O operators. The second stage is using quantum control and the trained ML models to optimize the V_O operator to maximize the distinguishability of the classes. The third stage is training a classical classifier for detecting the noise profile. These three stages form the training phase of the protocol and are executed once at the beginning. The testing phase of the protocol is the actual sense-control periodic cycles. The measurements of the spectator corresponding to the optimal control sequence are used fed into the trained classifier. The predicted label is then used to lookup the optimal sequence that controls the data qubit given the current noise profile.

is, they form a probability distribution. These weights are trainable, so during the training process, the loss function is optimized with respect to those weights. In order to ensure that the conditions hold, we can simply start with a general unconstrained set of weights $\{\tilde{w}_k\}_{k=1}^K$, and then pass them to a standard softmax activation layer that implements the transformation

$$w_k = \frac{e^{\tilde{w}_k}}{\sum_{k=1}^K e^{\tilde{w}_k}}. \quad (18)$$

The second output of the layer is a set of trainable signals $\{\hat{\beta}_k(t)\}_{k=1}^K$ that represents some noise realizations in time domain. For each of these realizations, we construct the Hamiltonian $H_S^{(1)}(t)$ using a custom whitebox implementing Equation 7. Next, the output passes through a *modified* quantum evolution whitebox that calculates the modified interaction unitary $\tilde{U}_I^{(k)}(T)$ for each

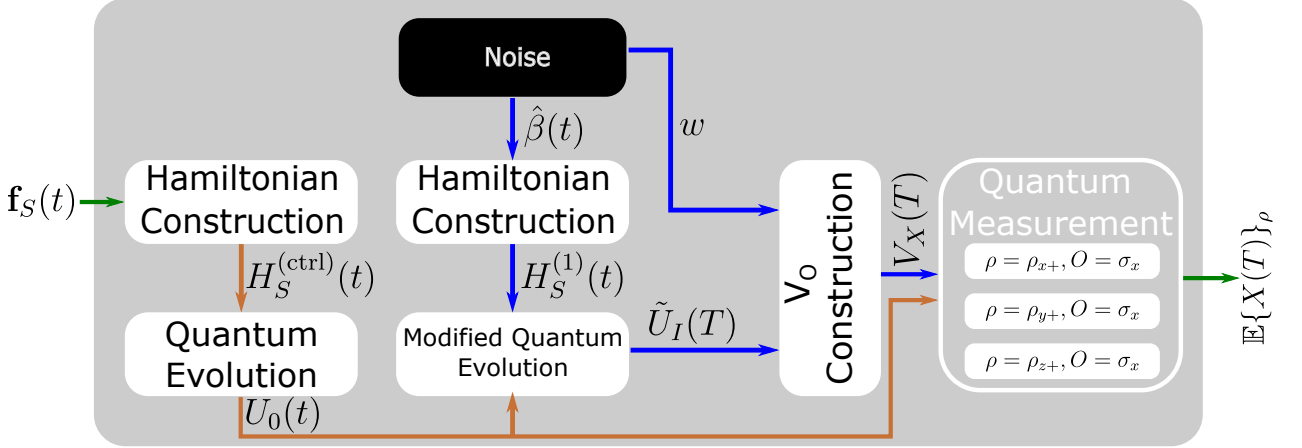


Figure 3: The proposed graybox structure for modeling a qubit. The input to the model is the control pulse sequence, and the output is the observables. The structure consists of two paths.

The first path is the control path which starts with the time domain representation of the control pulse sequence as an input, followed by the construction of the control Hamiltonian and unitary. The second is the noise path that starts with a customized blackbox that generate noise realizations and weights, followed by the construction of the modified interaction picture unitary and finally the V_O operator. The two paths merge at the output layer which calculates quantum observable parameterized by the initial state of the qubit.

noise realization $\hat{\beta}_k(t)$ using Equation 13. The time-ordered evolution is similarly approximated as in Equation 17. The two paths related to the noise then merge into the “ V_O Construction” layer, which is a whitebox that calculates the estimate of the operator as

$$\hat{V}_O(T) = \frac{1}{K} O^{-1} \sum_{k=1}^K w_k \tilde{U}_I^{(k)}(T)^\dagger O \tilde{U}_I^{(k)}(T). \quad (19)$$

This equation represents an approximation of the classical expectation in Equation 12 that is defined over a continuous distribution of all possible noise realizations, by a weighted average over a discrete distribution of K realizations. These special weights and realizations are the only trainable parameters in the model. Therefore, in order to minimize the loss function (which represents the error between the predicted outputs and the actual desired outputs), the training algorithm will be forced to find the optimal values for these parameters such that distance between the actual $V_O(T)$ operator and the estimated one $\hat{V}_O(T)$ is minimized.

The final layer in the model is the output layer which is a whitebox that calculates the quantum measurements using Equation 10. The input to the layer is the estimated $\hat{V}_O(T)$ operator from the noise path, and the $U_0(T)$ from the control path. The initial state of the qubit is parameter of this layer, and so the layer can generate measurements for multiple initial states. As discussed earlier, we can use the full 18 combinations of initial states and observables. In this case, we will need separate V_O construction layer for each observable connected to a quantum measurements layer. However, we do not need multiple noise or control paths, because the control unitary $U_0(T)$ and the modified interaction unitary $\tilde{U}_I(t)$ do not depend on the observables or the initial states. This is a consequence of the linearity of quantum mechanics.

We can see the difference between the proposed structure and the original one presented in [37]. The original design used standard blackboxes to generate a parameterization of the V_O operator. The input to the blackboxes was the control pulses. So, in some sense the model tries to learn the modified interaction picture, which encodes the interaction of control and noise. Whereas in this paper, we directly implement the modified interaction picture with

suitable whiteboxes, and use a customized blackbox to generate noise parameters. Moreover, in this paper we only use the time-domain representation of the control as the model input, instead of having two inputs (the parameterization and time-domain representation) in [37]. This simplifies the implementation of the model. The original design in [37] can still be used in the protocol. However, we choose to showcase a different graybox in this paper to emphasize the idea of building physics-aware ML models. The combination of blackboxes and whiteboxes is not unique and there is flexibility in the way we partition our model. This is similar to standard ML practice. There are some basic blackbox structures that could be combined in many different ways and with many hyperparameters to tune.

The dataset construction and the training and testing processes for the proposed graybox follows very similarly the method of [37]. The MSE is used as a loss function, and an experimental dataset is constructed by applying random control pulses and measuring the observables. The difference between this paper and [37], is that for the noise detection problem under consideration, we need to repeat the whole procedure (dataset construction, model training, and model testing) for each noise profile. Therefore, the output of this stage is a set of N trained ML grayboxes that model the spectator qubit corresponding to each possible noise profile. We do not use the same model for multiple profiles, each profile is associated with a different model. They all have the same aforementioned structure, but they end up with different trained parameters because they were trained on different datasets. A final note is that this process of constructing datasets and training ML models is very lengthy. However, it is only performed once at the beginning of the protocol and not repeated during the execution phase.

D. Stage II: Feature Extraction

The outputs of the first stage of the protocol is a set of trained ML models corresponding to each possible noise profile. These models could be used to predict the measurement outcomes given the control pulses. But, they also can be used to predict the V_O operators given the control pulses by simply probing the output of the V_O construction layer. Therefore, we can use this “reduced” model as a part of an optimization routine to do quantum control tasks. As discussed earlier, the choice of the control is crucial for the success of the detection. So, the target of the second stage of the protocol is to find the optimal discriminating control pulses that maximally separates the noise profiles. In order to do so, we need a criterion for the optimization. Here, we propose a heuristic which is the average distance between the V_O operators for each noise profile. Formally, we are looking for the pulse $\hat{\mathbf{f}}(t)$ that satisfies

$$\hat{\mathbf{f}}(t) = \arg \max_{\mathbf{f}(t)} \sum_{i,j \in \{1,2,\dots,N\}} \|V^{(i)} - V^{(j)}\|, \quad (20)$$

where $V^{(i)}$ is the V_O operator of the i^{th} noise profile and $\|\cdot\|$ is any matrix norm (in this paper we choose the Frobenius norm). Now, these operators are not accessible experimentally, therefore we need to show that the optimal pulses also enhances the separation between the noise profiles in the measurements space. Consider for simplicity we have two noise profiles, the initial state is $\rho(0)$, and the observable is the Pauli X operator. Also assume that the control is ideal with filter function $|F(\omega)|^2$, and the noise has PSD $S(\omega)$. In this case, we can show that

$$V_X = e^{-\int S(\omega)|F(\omega)|^2 d\omega} \sigma_0, \quad (21)$$

where σ_0 is the identity matrix. It is clear that maximizing the distance $\|V^{(1)} - V^{(2)}\|$ is equivalent to maximizing the distance $|I^{(1)} - I^{(2)}|$, where $I^{(j)}$ is the overlap integral of the j^{th} noise

profile, i.e. $I^{(j)} = e^{-\int S^{(j)}(\omega)|F(\omega)|^2 d\omega}$. The distance between the corresponding measurements is

$$|m^{(1)} - m^{(2)}| = \left| \text{tr} \left(V^{(1)} \rho_c(T) \sigma_x \right) - \text{tr} \left(V^{(2)} \rho_c(T) \sigma_x \right) \right| \quad (22)$$

$$= \left| \text{tr} \left((V^{(1)} - V^{(2)}) \rho_c(T) \sigma_x \right) \right| \quad (23)$$

$$= \left| \text{tr} \left((I^{(1)} \sigma_0 - I^{(2)} \sigma_0) \rho_c(T) \sigma_x \right) \right| \quad (24)$$

$$= \left| \text{tr} \left((I^{(1)} - I^{(2)}) \sigma_0 \rho_c(T) \sigma_x \right) \right| \quad (25)$$

$$= \left| I^{(1)} - I^{(2)} \right| \left| \text{tr} (\rho_c(T) \sigma_x) \right|, \quad (26)$$

where $\rho_c(t)$ is the closed-system evolution of the state due to control. The last step follows from the fact the difference between the integrals is just a scalar that can be pulled out of the trace. The first term we get depends on the noise and the control, while the second term depends on initial state and the control. Therefore, the optimization target should be the first term, which is why we chose the aforementioned heuristic in the first place. Maximizing the first term will definitely increase the separability between the measurements. However, in the worst case, the optimal control we obtain might result in an evolved state $\rho_c(T)$ that has a zero projection along the X-axis (any state in the YZ plane). In this case the second term would vanish, and the distance between the measurements would vanish as well. This situation can only be avoided if we use at least three initial states that are non-coplanar in the YZ plane (for example, the positive eigenstates of the Pauli operators). Since, $\rho_c(T)$ evolves using a closed-system unitary, the angles between the initial states have to be preserved after evolution. Therefore, no matter how the control is chosen, there will be at least one evolved state that does not have a vanishing X-component. This means at least one measurement will not vanish and so the worst-case scenario is avoided. In this case, the three measurements would constitute the final extracted feature vector that the classifier would be trained on in the next stage.

Now we can see that the use of the distance between V_O is a useful heuristic. First, it depends only on the noise and control and not the initial state. The V_O formalism itself allows the separation between the evolution due to control, and the effect of noise. If there was no noise, all the profiles would have been the same. Therefore, we have a framework that allows us to do feature extraction (pulse optimization) as well as feature selection (initial state selection). The other possible scenario was to optimize the distance between the measurements directly, In this case we probe the output of the trained ML models instead of the V_O construction layer outputs. Although this is possible, we prefer to use the distance between V_O as a more “natural” heuristic. A final note, is that although this argument depends on the ideal filter function formalism for simplicity, we will show in the numerical simulations that it works as well for general noise and control.

E. Stage III: Classifier Training

After the second stage of the proposed protocol is executed, we obtain the optimal control pulse that best discriminates between the different noise profiles. The next step would be to train a classifier for the noise detection. The inputs to the NN will be the measurements that correspond to the optimal control pulses. These measurements will be estimated from the trained ML models from stage 1. This is the second use of the trained models, besides using them for pulse design. For each noise profile, we use the corresponding trained model to estimate the value of the measurement. The desired output of the classifier will be the class label using one-hot encoding, i.e. the label would be a vector of all zeros except at the position of the correct profile, in which case it takes the value 1. This means the first profile will be labeled as $[1, 0, \dots, 0]$, the second profile as $[0, 1, 0, \dots, 0]$, etc. Therefore, a training example can be defined

as a pair of vectors, the first being the estimated measurements and second is the encoding of the noise profile. This implies the training set will consist of exactly N examples. In practice, this is not sufficient to train a standard ML classifier. Therefore, we construct the training set of the classifier differently. For each of the N basic examples, we generate R replicas each with white Gaussian noise added to the measurements. In signal processing, adding artificial noise to a signal to achieve a useful target is referred to as “dithering”. There are two reasons for performing this step. The first is as we mentioned to increase the size of the training set from N to NR . The second is to model actual errors that we would encounter in the testing phase. These errors manifest themselves as discrepancies between the predicted measurements using the trained models, and the measurements we obtain experimentally. There are two sources of such errors:

- Experimental errors such as State Preparation and Measurement (SPAM) and finite-sampling.
- Prediction errors due to the use of ML models.

And so to increase the robustness of the classifier, we must have noisy examples from each class, and hence the dithering step. Regarding the probability distribution of the dithering noise, there might be better distributions that takes into account models of those errors, however, this out of scope of this paper. The strength of the dithering noise have to be chosen carefully, to avoid the situation where all the classes overlap completely. At the end, the errors we are discussing should be minimal in practical situations. Note, that the dithering noise we are discussing in this context is artificial due to imperfections of the experiments and models. It has nothing to do with fundamental quantum noise that affects the evolution of the qubit, and which we aim to detect.

For the architecture of the classifier, we choose a standard NN blackbox to build the classifier. It consists of three layers: the first has N neurons, the second has $3N$ neurons, and the last layer has N neurons. The two hidden layer have a hyperbolic tangent activation while the output layer has softmax activation. We use an ADAM [45] optimization algorithm and the MSE as loss function. In general, there is a great flexibility to choose the hyperparameters defining the architecture, apart from the output layer which has to be chosen that way to generate a one-hot encoding. Once the training set is constructed for the classifier, the training is performed, and we also generate a very similar testing dataset to check the performance of the classifier.

This is the final stage of the training phase of our proposed protocol. The outcomes of this phase is the optimal control pulse sequence that best discriminated between the different noise profiles, and a trained classifier that acts on the corresponding optimal measurements. All of those stages are performed at the beginning before the actual operation of the device. At this point we are ready to move on the testing phase.

F. Stage IV: Testing

The testing phase of the protocol consists of one stage that is repeated periodically. The steps involved are very efficient and thus can be executed in real-time as opposed to the training phase that requires extensive characterization as well as computations. The first step is to experimentally measure the spectator qubit using the control pulses and initial states obtained from the training phase. This will constitute a feature vector that is then passed to the trained classifier. Although, the classifier is originally trained on simulated measurements, because of the dithering step, we still expect it to perform adequately. The output of the classifier would be a probability distribution of the different noise profiles given the measurements. In the ideal case, we would expect the distribution to be completely concentrated at the correct label. In

practice, this might not happen and we may end up with a broader distribution that still peaks around the correct class. This is still perfectly fine, since we can simply infer the class label as

$$\hat{n}_D = \arg \max_{i \in \{1, 2, \dots, N\}} \hat{y}_i, \quad (27)$$

where $[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N]$ is the output of the classifier. The undesired situation is that when two or more entries have exactly the same value. In this case, the prediction would be chosen randomly between the corresponding noise profiles. This will result probably in a misclassification, which is in fact naturally expected for any detection system. In terms of quantum noise, this means that we cannot differentiate between two or more noise profiles. This implies one of the following possibilities:

- The noise profiles are actually extremely close to each other, that we could consider them in fact one profile without a noticeable effect on the performance of the device.
- The noise profiles might still be different, but under the control constraints and capabilities available in the experiment, there is no way to practically differentiate between them.

In either case, it is impossible to distinguish between the noise profiles unless we change the control. By control constraints we mean maximum bandwidth, maximum amplitude, minimum pulse width, etc. These are imposed by the available hardware in the experiment, and they directly affect every stage in the protocol. When we perform the characterization, we use pulses having these constraints. Also, when we optimize to find the discriminating pulses, we must impose the constraints so that it is possible to implement them experimentally in the testing stage. Therefore, we have to consider the separability of noise profiles in the context of available control. We will give an example of a numerical simulation of this situation later in this paper.

Once, the noise profile is detected, we can use the fixed map \mathcal{F} as discussed earlier to infer the noise profile that is affecting the data qubit. And so we can instantaneously load the proper pulse sequence that implements a desired gate of the data qubit using the pre-built lookup table \mathcal{L} . This ends the stage, which can then be repeated periodically. The data qubit is never interrupted while executing the gates, the spectator is used instead. And so the objective of the proposed protocol is met. The performance of the classifier will be determined completely by the spectator qubit, under the assumption that the map \mathcal{F} is fixed. If the spectator qubit is of low-quality, then it might be an advantage because in this case it might be more sensitive to the quantum noise in the environment. In the next section, we will discuss the numerical simulations that supports the presented ideas.

IV. SIMULATION RESULTS

In this section, we show details about the numerical simulations that were performed to demonstrate the proposed protocol. We implemented the numerical experiments in this paper using Python and Tensorflow [46] and Keras [47]. The source code is publicly available as well as the datasets and the trained models that were used to generate the results in this paper¹. We will focus on the simulations of the stages of the protocol related to the spectator qubit only and not the data qubit. The section starts with an overview on the implementation details including the different simulation parameters and particularities of the protocol for the training phase. Next, we present the results of the testing phase by which we can assess the performance of the overall protocol. We end the section with a discussion on the significance of the obtained numerical results.

¹ <https://github.com/akramyousry/QFEND>

A. Training Phase

1. Stage I

The first stage in the proposed protocol aims to construct a set of characterization data for each possible noise profile, and train a corresponding ML graybox. The dataset will consist of pairs of random control pulses and corresponding measurement outcomes. We will simulate this process numerically. So, we choose a Hamiltonian for the spectator qubit that is a single-axis dephasing in the form

$$H(t) = \frac{1}{2}f_x(t)\sigma_x + \frac{1}{2}(\Omega + \beta(t))\sigma_z, \quad (28)$$

where $f_x(t)$ is the control pulses, Ω is the qubit energy gap, and $\beta(t)$ is the noise process. The evolution time interval is fixed to be $[0, T]$, and is discretized into M steps. The values of the simulation parameters are $T = 1$, $M = 1024$, and $\Omega = 12$. For the quantum observables, we choose the Pauli X , and with three positive eigenstates of the Pauli operators as an initial states. Thus we have only three measurements to perform. We use the same Monte Carlo based technique for the simulating the noisy evolution as in [37]. The number of noise realizations over which we take average is $K = 2000$.

For the noise profiles, we are going to generate realizations for six different random processes and we will choose different subsets to showcase the different possibilities that could occur in an actual experiment. The six profiles are as follows:

1. N0: noiseless (i.e. $\beta(t) = 0$)
2. N1: $\beta(t)$ is defined via its PSD, which take the form of a $1/f$ noise with a cutoff followed by a Gaussian bump. This can be expressed as $S_Z(f) = \frac{1}{f+1}u(15-f) + \frac{1}{16}u(f-15) + 0.5e^{-(f-30)^2/50}$, where $u(\cdot)$ is the unit step function.
3. N2: $\beta(t)$ is a stationary Gaussian colored noise defined via its autocorrelation matrix. The coloring of the noise is simulated by performing a convolution of a white Gaussian noise signal with some deterministic signal.
4. N3: $\beta(t)$ is a non-stationary Gaussian colored noise defined via its autocorrelation matrix. The non-stationarity is simulated by multiplying the stationary noise by some deterministic signal in time domain.
5. N4: $\beta(t)$ is a non-stationary non-Gaussian colored noise defined via its autocorrelation matrix. The non-Gaussianity is simulated by applying a non-linear function (in this paper we choose squaring) to a Gaussian noise.
6. N5: $\beta(t)$ is almost identical with the N1 profile, the only difference is the a slight shift in the location of the bump. The PSD is given by $S_Z(f) = \frac{1}{f+1}u(15-f) + \frac{1}{16}u(f-15) + 0.5e^{-(f-40)^2/50}$.

For each of these noise profiles, we will create a dataset that consists of 10000 examples, where one example is a pair of a control pulse sequence, and the corresponding quantum measurements. The control takes the form of a train of Gaussian pulses of fixed width, and random amplitude and position in the form

$$f_x(t) = \sum_{k=1}^n A_k e^{-\frac{(t-\mu_k)^2}{2\sigma^2}}, \quad (29)$$

where $n = 5$, $\sigma = \frac{1}{6} \frac{T}{2n}$, A_k is chosen randomly in the interval $[-100, 100]$, and μ_k is chosen randomly such that there are no overlapping pulses.

After the datasets are created, simulating the collection of experimental characterization data, we train a separate ML model on each dataset. We select 9000 examples for training and 1000 examples for testing. The number of training iterations is 1000. Supplementary Figure 5 shows the MSE evaluated over the training and testing examples as a function of the iteration number.

2. Stage II

The second stage after training the ML models for each noise profile is finding the optimal discriminating pulse sequence. Here we introduce three different scenarios, where we want to discriminate between $N = 5$ profiles as follows:

1. Scenario 1: The noise profiles are highly separable. In this case we use the N0, N1, N2, N3, and N4 profiles.
2. Scenario 2: Some of the noise profiles are close. Here we use the the N5, N1, N2, N3, and N4 profiles.
3. Scenario 3: The noise profiles are highly separable, but the control is limited to the range $[-1, 1]$. Similar to Scenario 1, we use the the N0, N1, N2, N3, and N4 profiles.

For each of these three scenarios, we run the pulse optimizer for 500 iterations. The optimal pulse sequence we obtain for each scenario is shown in Figure 6. The numerical experiments show in fact that these optimal pulses are not unique. If we run the optimizer multiple times, we can get different pulses.

3. Stage III

After we obtain the optimal control pulses, we proceed to final stage in the training phase of the protocol which is training the classifier on simulated optimal measurements. The first step is to use the trained ML models from stage 1 to simulate the three outcomes when we the input is the optimal pulses. These outcomes are the components of the feature vector. Next, we construct the dithered dataset by generating $R = 10000$ noisy replicas of the feature vector for each class. Since we have five profiles to distinguish, the total number of examples in the dataset is 50000. The examples are then randomly split into training and testing with a split ratio of 0.1. The classifier is then trained for 500 iterations.

This procedure is repeated for each of the three scenarios discussed in the simulations of Stage 2. With this, the training phase of the protocol is concluded. The outcomes are the optimal control pulses, and the trained classifier for each of the three scenarios.

B. Testing Phase

With the training phase of the protocol fully executed, we are ready to simulate the testing phase. This is when we can actually asses the performance of the protocol. The procedure is to experimentally measure the spectator qubit applying the optimal control pulse, and then passing the measurement outcomes to the trained classifier. Then the procedure is repeated periodically over time.

So, there are two main elements to simulate this part of the protocol. The first is simulating the optimal measurements. Here we once more use the Monte Carlo simulation that was used

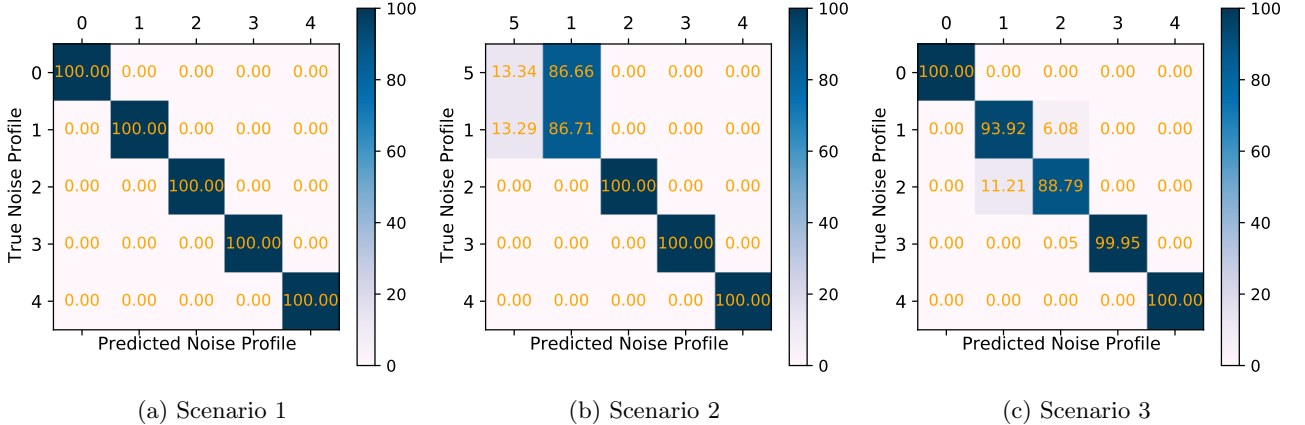


Figure 4: The confusion matrix for the three scenarios evaluated over 10^4 examples. The first scenario is when the noise profiles are highly separable, the second is when some profiles are close and the control is unlimited, and the final scenario is when some profiles are close and the control is limited. Each row in the confusion matrix represents the groundtruth class, while the columns represent the predictions by the classifier. Each entry is the corresponding percentage of times the classifier predicted a particular class given the groundtruth.

to create the datasets. We emphasize that we do not use the trained ML models of stage 1 in the training phase. Additionally, we reduce the number of noise realizations from $K = 2000$ to $K = 1000$. In a real experiment, this corresponds to a decreasing the number of shots we average over, and thus would speed-up the sensing step of the protocol. This is desired because at this stage, the protocol should operate in real-time. Moreover, this decreases the accuracy of the measurement, and thus it will act a good test to the robustness of the trained classifier to artificial noise. On the other hand, when we perform the characterization of the device at the beginning, we sacrifice the time in order to get high quality datasets to enhance the performance of the ML graybox models.

The second element is simulating the process of the noise profile changing in time. For this, we generate a random sequence of integers i_1, i_2, \dots, i_L of length $L = 10000$, where i_k denotes the index of the current noise profile. Then we simply loop over each index, generate the set of noise realizations corresponding to this profile, and run the quantum simulations as described previously. Different noise realizations are generated on the fly at each step in the sequence. This ensures there is enough randomness to mimic an actual experiment. After we run the simulation and get the measurement outcomes, we pass it to the trained classifier and store the predicted profile.

For each scenario, we do the aforementioned procedure, and then we calculate the confusion matrix as a metric for the performance of the protocol for this scenario. This is a widely-used metric in ML to assess the performance of classification algorithms. The confusion matrix C is an $N \times N$ matrix, where the element C_{ij} is defined to be the percentage of times the classifier predicted the label j whereas the groundtruth label is i . Thus, the sum of any row should be 100%. The best case is when the confusion matrix is a diagonal matrix with entries of 100%. In other cases, the confusion matrix can be helpful as it can show which classes are getting mixed by the classifier. Figure 4 shows the confusion matrix for the three scenarios.

C. Discussion

The numerical simulations in this section show a promising performance for the proposed protocol. There are two main results to explore. The first is the performance of the proposed

graybox ML model of the qubit. We can see from the plot of the MSE curves in Figure 5 that the model is able to learn the training examples, reflected by the training MSE curve decreasing with iterations. Additionally, the model is able to generalize as demonstrated by the testing MSE curve also decreasing with iterations, and not being significantly far from the training MSE curve. This was the case for the six noise profiles, which shows the ability of the model to learn diverse types of quantum noise.

The second results is the performance of the classifier as depicted by the confusion matrix in Figure 4. This actually reflects the performance of the whole protocol, since the results do not depend solely on the classifier design, but rather on all other steps. In Scenario 1, the profile N0, N1, N2, N3, and N4 were used, with the control pulses allowed to have the full range. We can see that the protocol was able to successfully classify all the labels correctly, and so the confusion matrix was diagonal with 100%. This means that the noise profiles are separated given the optimal control so there were no misclassifications. The result is also significant because the classifier is trained on the predicted measurements from the graybox, but tested on actual simulated measurements, and yet it succeeded in the task.

In Scenario 2, we have the profiles N5, N1, N2, N3, and N4. Now, profiles N5 and N1 are very close to each other. We see from the confusion matrix that the classifier was able to correctly classify all profiles, but there were misclassifications between those two profiles. This is exactly the expected behavior. This implies that under the constraints of the control pulses, these two profiles almost identical. It is also interesting that the confusion does not propagate into other classes, it is confined between N5 and N1 cases, with bias towards classifying both N5 and N1 examples to N1.

In Scenario 3, it is similar to the first one, with the profiles chosen to be N0, N1, N2, N3, and N4. The difference is that the pulse optimizer now has an extra constraint that the amplitudes should be restricted to $[-1, 1]$. In this case, we see a degradation in the performance of the classifier with misclassifications happening across various classes. This is an interesting result, because we know from scenario 1 that it is possible to distinguish between those profiles. However, due to the constraints of the control, it was not possible. This means that effectively some profiles become indistinguishable. This situation is very similar to standard quantum control. There is always a trade-off between the fidelity of a desired quantum gate, and the constraints of the allowed control (such as amplitude or bandwidth). This result shows that the effect of quantum noise is dependent on the control, and not just its statistical properties. This can be understood as well using the language of frames (see [48]).

V. CONCLUSION

In this paper, we proposed a protocol for noise detection in a data-spectator qubit system. The spectator is used to sense the noise to prevent the interruption of the data qubit during execution of a quantum computation. All the complexity of the protocol is concentrated in the characterization phase, allowing a real-time execution during the quantum computations. The protocol is designed following a quantum feature engineering approach to allow the utilization of machine learning methods. We presented a complete framework consisting of a novel graybox model for feature generation, a quantum control method for feature extraction, and a classifier. The numerical simulations show a promising performance of the protocol and is consistent with intuition about the behavior in various scenarios.

There are limitations and extensions that can be explored in the future. The first and most challenging limitation is the assumption that we have labeled characterization data. In other words, we are able to associate the characterization to a particular noise profile. We have discussed the possibility of doing this in some practical situations, but it is still a complex procedure generally. This cannot be avoided because we used supervised learning methods to design the protocol. However, there is a rich literature about unsupervised learning in which

we can classify examples into classes without the requirement of knowing the labels beforehand. This might work for noise detection applications because we are not interested in the label itself (as is the case in many applications such as object detection in images). Another limitation is the assumption of the known fixed mapping between the data and spectator noise profiles. It will be interesting to design protocol to characterize this map. Alternatively, it might be possible to design a graybox that takes into account both qubits so that we do not need to characterize the map separately.

There are other limitations that come from the use of ML methods such as the requirement of large datasets for training, which can cause problems for large quantum systems. Therefore, a theoretical analysis of the optimal observables would help in reducing the amount of required measurements. In this paper, we chose a heuristic that allowed us to select a small set instead of an informationally-complete set. It would be interesting to explore theoretical tools such as quantum information theory to do this task. Additionally, we saw that in situations where the control is constrained, the distinguishability between noise profiles is affected. It will be an interesting theoretical extension to study this problem and understand how exactly the trade-offs occur. This would facilitate the design of the protocol, because if we know that one or more profiles are not distinguishable under our control constraints, then we do not need to have different graybox models and datasets for them.

Regarding the numeric simulations, we made many choices regarding the design of the various machine learning tools. While our choices show a promising performance, there are many other possibilities that could lead to better results. It is also important to test the presented ideas on an actual experiment. A first simple test would be injecting noise artificially to an almost noiseless qubit, and assess the performance of the protocol. Finally, we presented a new application for the quantum feature engineering approach which is noise detection, besides the original proposal in [37, 38] for characterization and control of quantum systems. It would be interesting to explore further applications in other areas of quantum engineering.

Acknowledgments Funding for this work was provided by the Australian Government via the AUSMURI grant AUSMURI000002. This research is also supported in part by the iHPC facility at UTS. AY is supported by an Australian Government Research Training Program Scholarship.

-
- [1] J. Preskill, [Quantum](#) **2**, 79 (2018).
 - [2] G. Gordon, G. Kurizki, and D. A. Lidar, [Phys. Rev. Lett.](#) **101**, 010403 (2008).
 - [3] M. J. Biercuk, A. C. Doherty, and H. Uys, [Journal of Physics B: Atomic, Molecular and Optical Physics](#) **44**, 154002 (2011).
 - [4] L. Viola, E. Knill, and S. Lloyd, [Phys. Rev. Lett.](#) **82**, 2417 (1999).
 - [5] K. Khodjasteh and D. A. Lidar, [Phys. Rev. Lett.](#) **95**, 180501 (2005).
 - [6] M. H. Levitt and R. Freeman, [Journal of Magnetic Resonance \(1969\)](#) **43**, 65 (1981).
 - [7] C. Kabytayev, T. J. Green, K. Khodjasteh, M. J. Biercuk, L. Viola, and K. R. Brown, [Phys. Rev. A](#) **90**, 012316 (2014).
 - [8] K. Khodjasteh and L. Viola, [Phys. Rev. Lett.](#) **102**, 080501 (2009).
 - [9] J. Bylander, S. Gustavsson, F. Yan, F. Yoshihara, K. Harrabi, G. Fitch, D. G. Cory, Y. Nakamura, J.-S. Tsai, and W. D. Oliver, [Nature Physics](#) **7**, 565 (2011).
 - [10] F. K. Malinowski, F. Martins, L. Cywiński, M. S. Rudner, P. D. Nissen, S. Fallahi, G. C. Gardner, M. J. Manfra, C. M. Marcus, and F. Kuehmann, [Phys. Rev. Lett.](#) **118**, 177702 (2017).
 - [11] K. W. Chan, W. Huang, C. H. Yang, J. C. C. Hwang, B. Hensen, T. Tanttu, F. E. Hudson, K. M. Itoh, A. Laucht, A. Morello, and A. S. Dzurak, [Phys. Rev. Applied](#) **10**, 044017 (2018).
 - [12] C. Ferrie, C. Granade, G. Paz-Silva, and H. M. Wiseman, [New Journal of Physics](#) **20**, 123005 (2018).
 - [13] G. A. Álvarez and D. Suter, [Phys. Rev. Lett.](#) **107**, 230501 (2011).

- [14] P. Szakowski, G. Ramon, J. Krzywda, D. Kwiatkowski, and . Cywiski, *Journal of Physics: Condensed Matter* **29**, 333001 (2017).
- [15] J. Krzywda, P. Szakowski, and . Cywiski, *New Journal of Physics* **21**, 043034 (2019).
- [16] V. Frey, L. M. Norris, L. Viola, and M. J. Biercuk, “Optimally band-limited controls for quantum multi-axis spectral estimation,” (2019), [arXiv:1911.02005 \[quant-ph\]](https://arxiv.org/abs/1911.02005).
- [17] V. M. Frey, S. Mavadia, L. M. Norris, W. de Ferranti, D. Lucarelli, L. Viola, and M. J. Biercuk, *Nature Communications* **8**, 2189 (2017).
- [18] Y. Sung, F. Beaudoin, L. M. Norris, F. Yan, D. K. Kim, J. Y. Qiu, U. von Lüpke, J. L. Yoder, T. P. Orlando, S. Gustavsson, L. Viola, and W. D. Oliver, *Nature Communications* **10**, 3715 (2019).
- [19] L. M. Norris, G. A. Paz-Silva, and L. Viola, *Phys. Rev. Lett.* **116**, 150503 (2016).
- [20] G. Ramon, *Phys. Rev. B* **100**, 161302 (2019).
- [21] G. A. Paz-Silva, L. M. Norris, F. Beaudoin, and L. Viola, *Phys. Rev. A* **100**, 042334 (2019).
- [22] G. A. Paz-Silva, L. M. Norris, and L. Viola, *Phys. Rev. A* **95**, 022121 (2017).
- [23] L. Cywiński, *Phys. Rev. A* **90**, 042307 (2014).
- [24] L. M. Norris, D. Lucarelli, V. M. Frey, S. Mavadia, M. J. Biercuk, and L. Viola, *Phys. Rev. A* **98**, 032315 (2018).
- [25] T. Yuge, S. Sasaki, and Y. Hirayama, *Phys. Rev. Lett.* **107**, 170504 (2011).
- [26] C. L. Degen, F. Reinhard, and P. Cappellaro, *Rev. Mod. Phys.* **89**, 035002 (2017).
- [27] M. M. Mller, S. Gherardini, and F. Caruso, *Scientific Reports* **8** (2018), [10.1038/s41598-018-32434-x](https://doi.org/10.1038/s41598-018-32434-x).
- [28] C. Benedetti, F. Salari Sehdaran, M. H. Zandi, and M. G. A. Paris, *Phys. Rev. A* **97**, 012126 (2018).
- [29] H.-V. Do, C. Lovecchio, I. Mastroserio, N. Fabbri, F. S. Cataliotti, S. Gherardini, M. M. Mller, N. D. Pozza, and F. Caruso, *New Journal of Physics* **21**, 113056 (2019).
- [30] M. Abdelhafez, D. I. Schuster, and J. Koch, *Phys. Rev. A* **99**, 052327 (2019).
- [31] S. Machnes, E. Assémat, D. Tannor, and F. K. Wilhelm, *Phys. Rev. Lett.* **120**, 150401 (2018).
- [32] G. Ciaramella, A. Borzi, G. Dirr, and D. Wachsmuth, *SICOMP* **37**, A319 (2015).
- [33] T. Caneva, T. Calarco, and S. Montangero, *Physical Review A* **84**, 022326 (2011).
- [34] P. de Fouquieres, S. Schirmer, S. Glaser, and I. Kuprov, *J. Magn. Reson.* **212**, 412 (2011).
- [35] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Journal of magnetic resonance* **172**, 296 (2005).
- [36] J. Morris, F. A. Pollock, and K. Modi, [arXiv:1902.07980](https://arxiv.org/abs/1902.07980) (2019).
- [37] A. Youssry, G. A. Paz-Silva, and C. Ferrie, *npj Quantum Information* **6** (2020), [10.1038/s41534-020-00332-8](https://doi.org/10.1038/s41534-020-00332-8).
- [38] A. Youssry, R. J. Chapman, A. Peruzzo, C. Ferrie, and M. Tomamichel, *Quantum Science and Technology* **5**, 025001 (2020).
- [39] R. S. Gupta, L. C. G. Góvia, and M. J. Biercuk, *Phys. Rev. A* **102**, 042611 (2020).
- [40] S. Majumder, L. A. de Castro, and K. R. Brown, *npj Quantum Information* **6** (2020), [10.1038/s41534-020-0251-y](https://doi.org/10.1038/s41534-020-0251-y).
- [41] S. Martina, S. Gherardini, and F. Caruso, [arXiv preprint arXiv:2101.03221](https://arxiv.org/abs/2101.03221) (2021).
- [42] T. L. Scholten, Y.-K. Liu, K. Young, and R. Blume-Kohout, [arXiv preprint arXiv:1908.11762](https://arxiv.org/abs/1908.11762) (2019).
- [43] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Third Edition* (Academic Press, Inc., USA, 2006).
- [44] A. Youssry, A. El-Rafei, and S. Elramly, *Quantum Information Processing* **15**, 2303 (2016).
- [45] D. P. Kingma and J. Ba, in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, edited by Y. Bengio and Y. LeCun (2015).
- [46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” (2015), software available from [tensorflow.org](https://www.tensorflow.org).
- [47] F. Chollet *et al.*, “Keras,” <https://keras.io> (2015).
- [48] B. Tonekaboni, T. Chalermputitarak, Y. Wang, L. Norris, L. Viola, and G. A. Paz-Silva, in preparation (2020).

Appendix A: Supplementary Figures

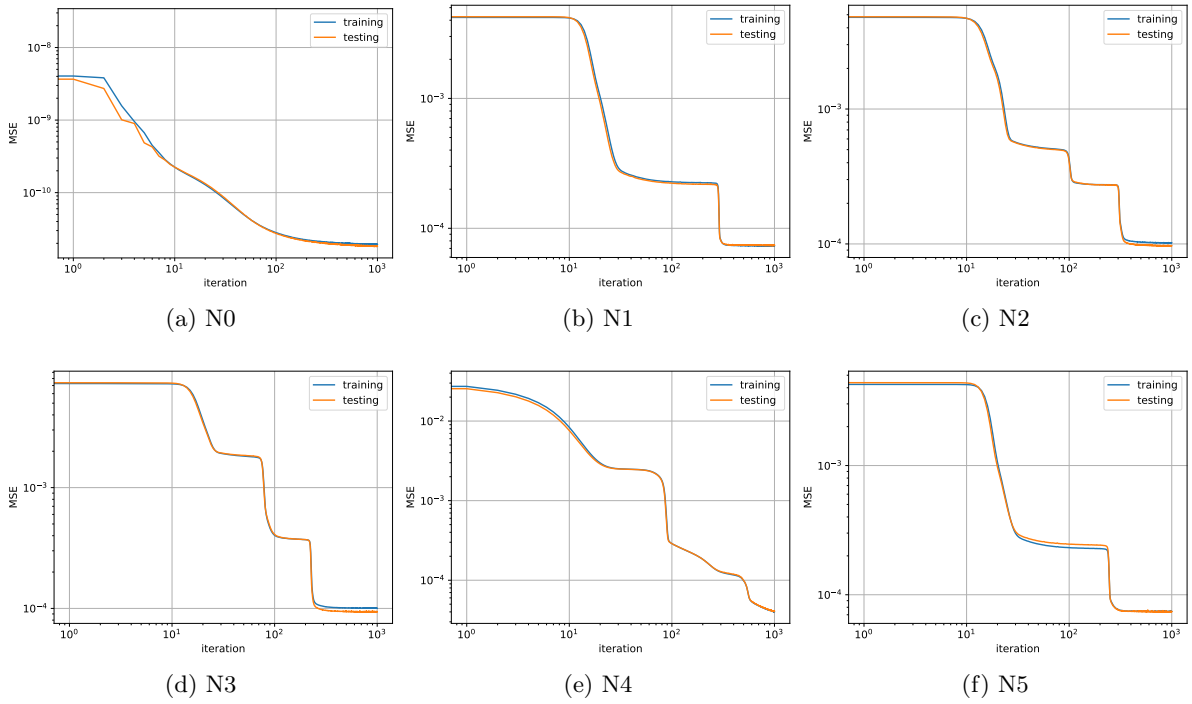


Figure 5: The MSE evaluated for the training and testing examples for each of the simulated datasets corresponding to each noise profile.

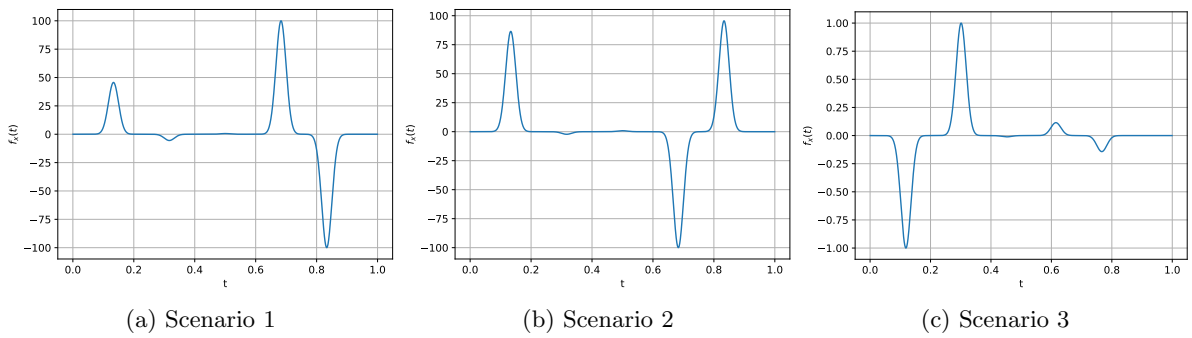


Figure 6: The optimal discriminating pulses for the three scenarios described in the main text.