

# Quantum Feature Space of a Qubit Coupled to an Arbitrary Bath

Chris Wise<sup>1\*</sup>, Akram Youssry<sup>2</sup>, Alberto Peruzzo<sup>2,3</sup>, Jo Plested<sup>1</sup>,  
Matt Woolley<sup>4</sup>

<sup>1</sup>School of Systems and Computing, UNSW, Canberra, ACT, 2601,  
Australia.

<sup>2</sup>Quantum Photonics Laboratory and Centre for Quantum  
Computation and Communication Technology, RMIT University,  
Melbourne, VIC, 3000, Australia.

<sup>3</sup>Quandela, Massy, France.

<sup>4</sup>School of Engineering and Technology and Centre for Engineered  
Quantum Systems, UNSW, Canberra, ACT, 2601, Australia.

\*Corresponding author(s). E-mail(s): [c.wise@unswalumni.com](mailto:c.wise@unswalumni.com);

## Abstract

Qubit control protocols have traditionally leveraged a characterisation of the qubit-bath coupling via its power spectral density. Previous work inferred noise operators that characterise the influence of a classical bath using a grey-box approach that combines deep neural networks with physics-encoded layers. This overall structure is complex and poses challenges in scaling and real-time operations. Here, we show that no expensive neural networks are needed and that this noise operator description admits an efficient parameterisation. We refer to the resulting parameter space as the *quantum feature space* (QFS) of the qubit dynamics resulting from the coupled bath. We show that the Euclidean distance defined over the QFS provides an effective method for classifying noise processes in the presence of a given set of controls. Using the QFS as the input space for a simple machine learning algorithm, we demonstrate that it can effectively classify the stationarity and the broad class of noise processes perturbing a qubit. Finally, we explore how control pulse parameters map to the QFS.

## 1 Introduction

Noise in quantum computers that results in qubits losing coherence (decoherence) is due to unwanted bath coupling [1]. One way to mitigate decoherence is to decouple the qubit from the bath by applying a control field as a series of pulses [2]. This technique was first studied in nuclear magnetic resonance imaging [3, 4]. Within this framework, control pulses are applied for a time  $\tau$  and then turned off for a time  $\tau$ , and this pattern is repeated  $N$  times. This oscillating pulse sequence design is the Carr-Purcell-Meiboom-Gill (CPMG) sequence [5, 6].

Álvarez and Suter (AS) proposed a modified CPMG sequence to perform qubit spectroscopy [7]. AS pulses are part of a family of pulses used for dynamical decoupling noise spectroscopy, and they can be used to infer qubit coherence as a function of time. These empirical coherence data, obtained by AS pulses, are fitted to theoretical coherence curves corresponding to parametric noise spectra [8, 9]. However, this method requires an accurate model of system dynamics [10], which can be challenging to develop in general [11]. Furthermore, if an inappropriate parametric noise spectrum is chosen *a priori*, the estimated noise spectrum will correspond to an inaccurate model of system dynamics, and subsequently, control pulses optimised using this system model will likely be suboptimal [11].

More direct methods of noise spectroscopy include techniques such as two-point correlation, which measures the correlation function of the noise by implementing gates at different times and measuring the change in the qubit state [12]. The Fourier transform of this correlation function gives the noise spectrum. However, this method can be resource-intensive, time-consuming, computationally expensive, and requires that the noise be stationary [8, 10, 13, 14].

The authors of [14] developed a quantum noise spectroscopy method that used the Fourier transform of the measurements of the free-induction decay of the coherence curves. The technique accurately recovered the correct noise spectra and outperformed previous decoupling schemes while significantly reducing experimental overhead. However, the method makes several assumptions that limit the applicability of the work. For example, the authors assume the qubit is only subject to dephasing, where the qubit thermal relaxation process occurs over a much longer time scale than phase randomisation, and that the frequency fluctuations of a qubit are subject to stationary zero-mean Gaussian noise.

In [10], researchers instead used deep learning techniques to infer accurate qubit noise spectra. The method used coherence curves as input to a recurrent neural network with a feedforward neural network head to estimate the noise spectrum. They showed that the model could accurately estimate the qubit noise spectrum using only the coherence curves and outperformed the standard noise spectroscopy techniques. Researchers

in [13] further demonstrated the effectiveness of deep learning by using it to reconstruct the power spectral density (PSD) of an ensemble of carbon impurities around a diamond's nitrogen-vacancy centre.

Youssry et al. [15] introduced a grey-box model separating control and system-bath dynamics. In this work, time-dependent Hamiltonians and unitaries specified the control dynamics, and a form of recurrent neural network (a gated recurrent unit) was used to predict the system-bath dynamics associated with control pulses. The input to this neural network was parameterised control pulses as a sequence of vectors. The model output was the parameterisation of a noise operator. The authors showed that the grey-box model could be used for qubit control pulse optimisation and qubit noise spectroscopy. This grey-box model approach was later used for qubit control in the presence of a non-Markovian bath [16]. The formalism used in this work is discussed in more detail in Sec. 2.

Despite the success of [10, 13, 15, 16], these approaches are limited by their use of deep learning models, specifically the expense of training deep learning models and the need for large datasets. Expensive models are undesirable as noise processes in quantum systems can drift over time [17, 18], necessitating new training data and model retraining potentially daily or even hourly. Additionally, noise profiles may change from one physical qubit to another [19], so a specific model may be required for each qubit, which does not scale. Furthermore, specific to the grey-box method of [15, 16], the white-box aspect (the so-called physics-informed aspect of the network) scales exponentially with the dimensions of the quantum system, adding to the challenge of moving beyond toy problems.

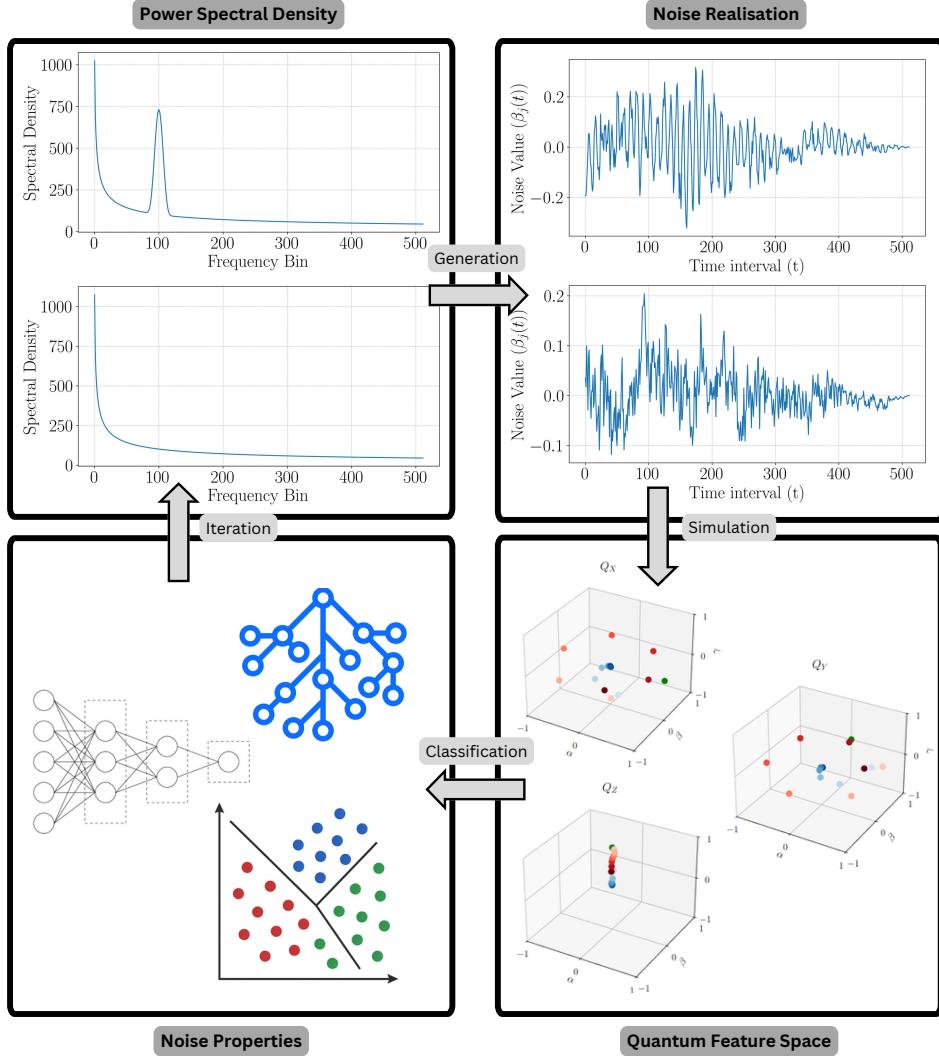
This variability of qubit noise profiles, both in terms of time and between qubits, and the exponential scaling of the grey-box model, motivated the need to move beyond deep learning models for noise spectroscopy and to find feature spaces that can be used to classify qubit noise profiles without deep learning models.

We introduce in this paper the *quantum feature space* (QFS), which encodes the bath's influence on system dynamics while making minimal assumptions about the bath such that it can be used beyond toy problems and have utility for a broad range of physically realised qubits. The QFS is deduced from observable expectations using efficient linear regression models. We then show that no expensive algorithms are needed to classify the noise affecting a qubit; instead, we use the Euclidean distance within the QFS for classification. Fig. 1 demonstrates further uses for the QFS. The feature space can be used as the input space to simple machine learning algorithms for noise classification, or the space can enable control pulse optimisation via visualising how control pulse parameters manifest themselves in the QFS.

## 2 Theoretical Results

### 2.1 Open System Dynamics

We start by modeling a single qubit perturbed by classical noise processes  $\beta_j(t)$  to represent system-bath interactions (using units where  $\hbar = 1$ ). The total Hamiltonian



**Fig. 1:** The diagram illustrates an iterative process for characterising and learning noise properties in quantum systems. Starting with a PSD (top), noise realisations are generated (right) and combined with control pulses in a quantum simulation. The outcomes of these simulations are mapped onto a QFS (bottom), where data points encode the effects of the noise. This feature space is used by regression models such as decision trees to infer key noise properties, including type and stationarity (left). These inferred properties refine the PSD model used in subsequent iterations. This closed-loop framework enables the classification and clustering of noise processes. It facilitates tailored noise mitigation strategies, leveraging machine learning techniques like decision trees, k-means clustering, and neural networks for enhanced quantum control.

is then

$$H(t) = H_{\text{ctrl}}(t) + H_{\text{noise}}(t) \quad (1)$$

with

$$H_{\text{ctrl}}(t) = \Omega \frac{\sigma_z}{2} + \sum_{j=\{x,y,z\}} f_j(t) \frac{\sigma_j}{2} \quad (2)$$

where  $\Omega$  is the energy gap of the qubit and  $f_j(t)$  are externally applied control pulses; and

$$H_{\text{noise}}(t) = \sum_{j=\{x,y,z\}} \beta_j(t) \sigma_j \quad (3)$$

where  $\beta_j(t)$  are stochastic processes that represent the noise.

We are interested in the expected value of a system (a qubit) observable  $O$  at time  $T$  given an initial state of the system  $\rho$ , where  $T$  represents the total time of the control pulse sequence. This is given by,

$$\mathbb{E}\{O(T)\}_\rho = \langle \text{Tr} [U(T)\rho U(T)^\dagger O] \rangle \quad (4)$$

where  $U(T) = \mathcal{T}e^{-i \int_0^T H(s) ds}$  is the time-ordered exponential of the time-dependent Hamiltonian in Eq. (1), and  $\langle \cdot \rangle$  denotes classical averaging over realisations of random processes  $\beta_j(t)$ .

We are interested in decomposing the total evolution into a product of system-bath dynamics and system dynamics. This decomposition allows us to separate the influence of the bath on the system, which is crucial for understanding how noise affects qubit control and dynamics. To do this, we follow the work of Paz-Silva, Norris, and Lorenza [20] and Youssry, Pas-Silva, and Ferrie [15] and move to an interaction picture with respect to the control Hamiltonian.

We start then with the time-ordered control unitary,

$$U_{\text{ctrl}}(t) = \mathcal{T}e^{-i \int_0^t H_{\text{ctrl}}(s) ds} \quad (5)$$

We then define the interaction Hamiltonian as follows,

$$H_{\text{I}}(t) = U_{\text{ctrl}}^\dagger(t) H_{\text{noise}}(t) U_{\text{ctrl}}(t) \quad (6)$$

with the following associated unitary,

$$U_{\text{I}}(t) = \mathcal{T}e^{-i \int_0^t H_{\text{I}}(s) ds} \quad (7)$$

Using this definition of the interaction Hamiltonian allows us to rewrite the total unitary as a product of the interaction unitary and the control unitary without assuming that the control and noise Hamiltonians commute (see Sec. 8.1 for more details). Specifically, we can write the total unitary as,

$$U(t) = U_{\text{ctrl}}(t) U_{\text{I}}(t) = U_{\text{ctrl}}(t) U_{\text{I}}(t) U_{\text{ctrl}}^\dagger(t) U_{\text{ctrl}}(t) = \tilde{U}_{\text{I}}(t) U_{\text{ctrl}}(t) \quad (8)$$

where  $\tilde{U}_I(t) = U_I(t)U_{ctrl}^\dagger(t)U_{ctrl}(t)$  is a modified interaction unitary defined so that we can rewrite the expectation in Eq. (4) as

$$\mathbb{E}\{O(T)\}_\rho = \left\langle \text{Tr} \left[ \tilde{U}_I(T)U_{ctrl}(T)\rho U_{ctrl}^\dagger(T)\tilde{U}_I^\dagger(T)O \right] \right\rangle \quad (9)$$

using the cyclic property of the trace, we have

$$\mathbb{E}\{O(T)\}_\rho = \left\langle \text{Tr} \left[ U_{ctrl}(T)\rho U_{ctrl}^\dagger(T)\tilde{U}_I^\dagger(T)O\tilde{U}_I(T) \right] \right\rangle \quad (10)$$

Bringing the trace inside the expectation and moving the constants out, we have

$$\mathbb{E}\{O(T)\}_\rho = \text{Tr} \left[ U_{ctrl}(T)\rho U_{ctrl}^\dagger(T) \left\langle \tilde{U}_I^\dagger(T)O\tilde{U}_I(T) \right\rangle \right] \quad (11)$$

where we define the QFS operator as

$$\tilde{O}(T) = \left\langle \tilde{U}_I^\dagger(T)O\tilde{U}_I(T) \right\rangle \quad (12)$$

giving us,

$$\mathbb{E}\{O(T)\}_\rho = \text{Tr} \left[ U_{ctrl}(T)\rho U_{ctrl}^\dagger(T)\tilde{O}(T) \right] \quad (13)$$

This shows that the measured expectation depends on the system's evolution under control ( $U_{ctrl}(T)$ ) and the bath's influence, captured entirely by  $\tilde{O}(T)$ . Notice, that at no point did we assume  $[H_{ctrl}(t), H_{noise}(t)] = 0$ ; instead, we used the interaction picture to isolate bath effects, consistent with standard techniques in open quantum systems.

Note, continuing forward, we will omit the time-dependence notation for clarity, and it is understood that the parameters and operators are meant to be evaluated at the end of the control pulse sequence, at time  $T$ , if the time-dependence is not explicitly stated.

## 2.2 Deriving the Quantum Feature Space

The previous grey-box approach to qubit control of [15, 16] used expectations and control pulse amplitudes, both classical feature spaces. This motivated us to develop a method to deduce the  $\tilde{O}$  operator and its parameters from observable expectations as it was hypothesised that the feature space formed from the parameters of  $\tilde{O}$  would be a richer feature space than those used in previous work. By creating a rich feature space, we could classify noise processes without needing expensive deep learning models.

As such, it is important to note some of the mathematical properties of  $\tilde{O}$  as they subsequently inform restrictions on its parameterisation. Specifically,  $\tilde{O}$  has zero trace, is Hermitian, and has real eigenvalues between  $-1$  and  $1$  [15], resulting in the following form,

$$\tilde{O} = \begin{bmatrix} \gamma_O & \alpha_O - \beta_O i \\ \alpha_O + \beta_O i & -\gamma_O \end{bmatrix} \quad (14)$$

with  $-1 \leq \alpha_O, \beta_O, \gamma_O \leq 1$ .

Now  $U_{\text{ctrl}}\rho U_{\text{ctrl}}^\dagger$  must be Hermitian, with trace one, and positive semi-definite, giving us the following form,

$$U_{\text{ctrl}}\rho U_{\text{ctrl}}^\dagger = \begin{bmatrix} a_\rho & b_\rho - c_\rho i \\ b_\rho + c_\rho i & 1 - a_\rho \end{bmatrix} \quad (15)$$

with  $0 \leq a_\rho, b_\rho, c_\rho \leq 1$ .

We can now calculate the generalised form of  $\mathbb{E}\{O\}_\rho$  in terms of the newly introduced parameters,

$$\mathbb{E}\{O\}_\rho = 2b_\rho\alpha_O + 2c_\rho\beta_O + (2a_\rho - 1)\gamma_O \quad (16)$$

For a single qubit (two-dimensional system), for each observable, we measure the observable expectation value for six initial states, specifically the up and down eigenstates for each of the three Pauli matrices ( $\sigma_x, \sigma_y, \sigma_z$ ). This process yields an over-determined system of six linear equations with three unknowns, which we can solve to find the parameters  $\alpha_O, \beta_O$ , and  $\gamma_O$  for each observable  $O$ . The system is represented by the following equation:

$$\begin{bmatrix} 2b_{\rho_{x+}} & 2c_{\rho_{x+}} & (2a_{\rho_{x+}} - 1) \\ 2b_{\rho_{x-}} & 2c_{\rho_{x-}} & (2a_{\rho_{x+}} - 1) \\ \vdots & \vdots & \vdots \\ 2b_{\rho_{z-}} & 2c_{\rho_{z-}} & (2a_{\rho_{z-}} - 1) \end{bmatrix} \begin{bmatrix} \alpha_O \\ \beta_O \\ \gamma_O \end{bmatrix} = \begin{bmatrix} \mathbb{E}\{O\}_{\rho_{x+}} \\ \mathbb{E}\{O\}_{\rho_{x-}} \\ \vdots \\ \mathbb{E}\{O\}_{\rho_{z-}} \end{bmatrix} \quad (17)$$

with  $\rho_{x+}, \rho_{x-}, \dots, \rho_{z-}$  being the standard eigenstates of the Pauli matrices.

The section has also shown how the quantum simulator takes a control sequence and stochastic realisation defined for  $T$  timesteps and produces QFS parameters ( $\alpha_O, \beta_O$ , and  $\gamma_O$ ) and observable expectations ( $\mathbb{E}\{O\}_\rho$ ). For a physical quantum computer, this section has also shown how to use observable expectations to deduce the QFS parameters using the over-determined system of equations in Eq. (17). Properties within the QFS (such as the Euclidean distance) can then be used to estimate the underlying noise model and its parameters. The QFS can also form the input to a simple machine learning algorithm (such as a random forest) or a small neural network, both with nine inputs for the single-qubit case (three parameters for each of the three observables). The output of the machine learning algorithm can be used to classify properties of the noise process, such as its stationarity and type, where the algorithm is trained on labels from the known simulated noise processes. The trained model can then be applied to an unknown noise process from a physical quantum computer to estimate the noise properties and parameters.

### 3 Experimental Results

#### 3.1 Efficiency of QFS Derivation

To find the best fit solution for Eq. (17), we used the Moore-Penrose inverse [21]. We implemented the Moore-Penrose inverse and methodologies of Sec. 2.2 using *Pytorch* [22] and parallelised calculations using across a GPU. The GPU for all experiments and simulations was an NVIDIA GeForce RTX 3060 with 12GB of memory.

To test and verify the methods of Sec. 2.2 we used ground truth expectations and the corresponding  $\tilde{O}$  operators from *QDataSet* [23]. A batch of 1000 expectations was used to derive the parameters  $\alpha_O$ ,  $\beta_O$ , and  $\gamma_O$ , with a total time of  $0.52 \pm 0.05$  ms.

#### 3.2 Quantum Simulation Implementation

Our quantum simulator followed the formalism defined in Sec. 2 and was implemented in *Pytorch* [22]. We noted inefficiencies in the simulator of [23], which led to long execution times, motivating the need for a redesigned simulator to allow frequent and fast experiments. In the redesigned simulator, overall improvement was achieved by using tensor operations whenever possible for the operations presented in Sec. 2.

Like [23], we used the Lie-Trotter product formula to estimate the time-evolution operator, that is,

$$U(t) = \mathcal{T}e^{-i \int_0^t H(s) ds} \approx \prod_{j=0}^{N-1} e^{-iH(t_j)\Delta t} \quad (18)$$

where  $N$  is the number of time steps,  $t_j$  is the time at the  $j^{\text{th}}$  time step, and  $\Delta T = \frac{T}{N}$  is the time step size.

Efficiency gains were made by implementing two major algorithmic changes. Firstly, for Eq. (6), the time-ordered control unitaries ( $U_{\text{ctrl}}(t)$ ) for all time steps needed to be calculated. To do this, first, we created an array of the exponentiated control Hamiltonians

$$[e^{-iH(0)\Delta t}, e^{-iH(1)\Delta t}, \dots, e^{-iH(N-1)\Delta t}] \quad (19)$$

Since the unitary at the  $j^{\text{th}}$  time step is the product of  $j^{\text{th}}$  and all previous exponentiated control Hamiltonians, our array of control unitaries,

$$[U_{\text{ctrl}}(0), U_{\text{ctrl}}(1), \dots, U_{\text{ctrl}}(N-1)] \quad (20)$$

became

$$[e^{-iH(0)\Delta t}, e^{-iH(1)\Delta t}e^{-iH(0)\Delta t}, \dots, e^{-iH(N-1)\Delta t} \dots e^{-iH(1)\Delta t}e^{-iH(0)\Delta t}] \quad (21)$$

Since matrix multiplication is associative, we computed the above array using a Prefix Scan [24] such that only  $O(\log(N))$  iterations of matrix multiplications were required.

A further improvement was that our simulation did not compute the modified interaction unitary ( $\tilde{U}_I(t)$ ) for all timesteps; instead, only the final time step modified

interaction unitary was calculated, as this was the only unitary required for expectation calculations. By enforcing this, again we could exploit the associativity of matrix multiplication and implement a binary tree structure to compute the final unitary, which also only required  $O(\log(N))$  iterations of matrix multiplications.

We benchmarked our simulator and found that what required a few days for single qubit simulation in the previous implementation [23] took 10 minutes with the new simulator, giving over a  $400\times$  improvement. Greater speed-ups can be achieved if computing at lower precision and or using a GPU with more memory, where both options enable larger batch sizes.

Note, for simulator experiments, we derived the QFS parameters directly and did not calculate expectations as we had access to the  $\tilde{O}$  operators. However, when using expectations from a physical quantum computer, one only needs to calculate  $U_{\text{ctrl}}(T)\rho U_{\text{ctrl}}^\dagger(T)$ , for the final time step  $T$  (using the efficient binary tree structure), and then solve Eq. (17), using efficient methods such as the parallelised Moore-Penrose inverse, to find the parameters  $\alpha_O$ ,  $\beta_O$ , and  $\gamma_O$ .

### 3.3 Creation of QFS Data

To create the QFS data, we started with noise models and realisation generation methods following that of *QDataSet* [23]. All noise processes acted along the  $x$  and  $z$  axes, with the noise processes being correlated between the axes. This correlation was achieved by setting the noise values along the  $z$ -axis to be the absolute value of the  $x$ -axis values (similar to noise profile N6 in *QDataSet*).

Our experiments explored the following noise processes:

- a noise process with a  $1/f$  noise PSD (N5 in *QDataSet*)
- a noise process with a  $1/f$  noise PSD with a Gaussian-shaped bump in the PSD (N1 in *QDataSet*)
- a Gaussian coloured noise process (N2 in *QDataSet*)

There were six different noise processes in total. The first three were the above noise processes, and the other three were non-stationary variants of each noise process. To achieve non-stationarity, we multiplied the noise processes by a deterministic triangular time envelope. The envelope was a triangular function with a peak at  $T/2$  and a width of  $T$ .

The second noise process, ‘ $1/f + \text{bump}$ ’, was chosen because it shared similarities with the first noise process ( $1/f$ ). It was hypothesised that these two noise processes would result in parameters close together in the QFS. Conversely, the third process (Gaussian coloured) was chosen because it was hypothesised that it would result in parameters far from the points of the first two noise processes.

Control pulses were applied along the  $x$ -axis for all simulations, where the control pulses were approximations of CPMG pulses. To achieve CPMG pulses, we created

Gaussian-shaped pulses of the form

$$f_j(t) = \sum_{n=1}^{n_{\max}} A_n e^{-\frac{(t-\tau_n)^2}{2\sigma^2}} \quad (22)$$

where  $\sigma = \frac{T}{\lambda M}$ ,  $\lambda \in \mathbb{R}^+$  is the pulse width,  $T$  is the total time,  $M$  is the number of time steps,  $\tau_n = \left(\frac{n-0.5}{n_{\max}}\right)T + \delta_\tau$ , with  $\delta_\tau \in \mathbb{R}$  being used to add jitter to the pulse timing, and  $A_n$  is the amplitude of the  $n^{\text{th}}$  pulse. For all experiments, the number of pulses was  $n_{\max} = 5$ , 1024 discrete time steps were used, and ensemble averaging was performed over 2000 realisations of the noise process for a given control pulse sequence.

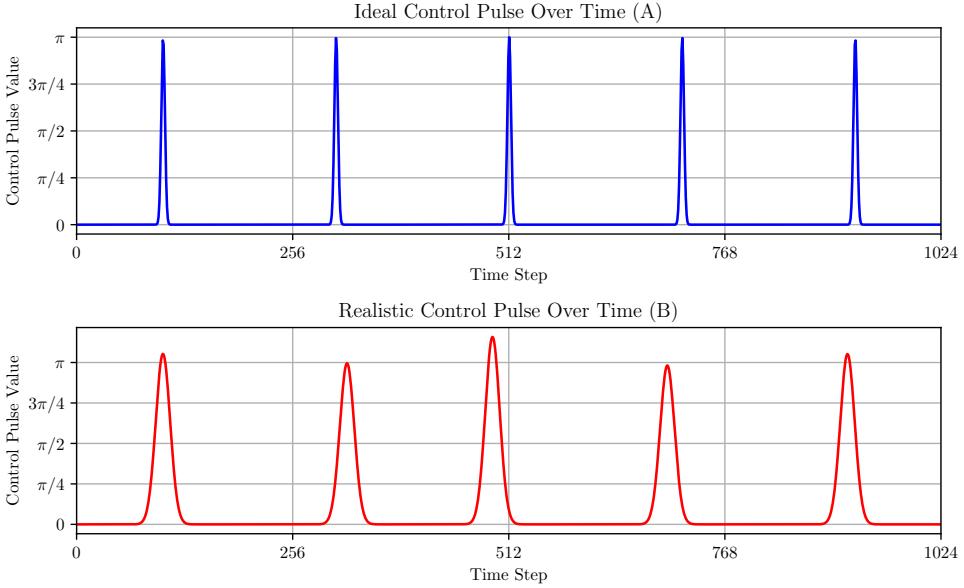
### 3.4 Black Box Classification of Bath Through QFS

We conducted a series of simulations to test the ability of the QFS to classify distinct noise processes. We first simulated qubit dynamics subject to realisations of the above noise processes with the approximated ideal CPMG pulses. We then simulated a ‘1/ $f$  + bump’ process, changing the location of its peak compared to earlier noise processes, which emulated an unknown noise process from a physical qubit that an experimentalist would be trying to classify (the unknown noise process had a peak at the 200<sup>th</sup> frequency bin). The goal was to classify the unknown noise process by comparing it to the known simulated noise processes in the QFS.

While simulating a qubit in the presence of the realisation of an unknown noise process, we used more realistic CPMG pulses; the pulse width was increased, and jitter was added to the peak’s timing and value. From the realistic CPMG pulses, we ended up with a cluster of points in the feature space arising from distinct realisations of the jitter. In a black box manner, we identify which noise process characterises the unknown noise process by measuring which *a priori* noise process the cluster of points was closest to in the QFS.

To simulate ideal CPMG pulses, we set  $\lambda = \frac{1}{96}$ ,  $\delta_\tau = 0$ ,  $n_{\max} = 5$ , and  $A_n = \pi$ . These values were chosen to approximate delta function pulses, shown in Fig. 2(a). For the realistic CPMG pulses, shown in Fig. 2(b), we set  $\lambda = \frac{1}{24}$ ,  $\delta_\tau$  was chosen from a uniform random distribution spanning the interval  $[-\frac{24T}{M}, \frac{24T}{M}]$ , and  $A_n = \pi + \epsilon$  where  $\epsilon$  was drawn from a uniform random distribution spanning the interval  $[-\frac{\pi}{5}, \frac{\pi}{5}]$ . This is s. Fifty such unique realistic control pulse sequences were generated with ensemble averaging over 2000 noise process realisations for each control pulse, resulting in a cluster of 50 unique points in the QFS.

From the first round of simulations, we calculated the average of the Euclidean distances between the realistic data points and the single data point from the *a priori* noise processes for the feature spaces associated with  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$ , with the results shown in Table 1. We found that the ‘1/ $f$  + bump’ noise process had the smallest total average Euclidean distance across all feature spaces, indicating the stationary ‘1/ $f$  + bump’ noise process is likely closest to the unknown noise process.



**Fig. 2:** Control pulses used in the simulations. (a) shows the ideal CPMG pulses, and (b) shows the realistic CPMG pulses.

To refine our classification of the unknown noise process, using the results of the first iteration, we simulated several ‘ $1/f$  + bump’ noise profiles, varying the location of the peak in the PSD. Specifically, peaks were positioned at predefined frequency bins, such as the 15th or 30th bin. The result shown in Table 2 indicated that the noise processes with peaks in the 240<sup>th</sup> and 120<sup>th</sup> bins were the closest matches, as measured by the average Euclidean distances in the QFS to the unknown process, which peaks at the 200<sup>th</sup> bin.

To further refine our estimate of the unknown noise process, we conducted a third round of simulations, this time using the peaks between and including 130<sup>th</sup> and 230<sup>th</sup> bins. As the results show in Table 3, the noise processes with peaks at the 190<sup>th</sup> and 210<sup>th</sup> bins were the closest matches. If we then conclude that the unknown noise process is most similar to the ‘ $1/f$  + bump’ processes with a peak at the 200<sup>th</sup> bin, we have then correctly estimated the unknown noise process.

It is interesting to note the significantly larger distances between the ‘ $1/f$  + bump’ noise processes and the coloured noise processes. This enforces the hypothesis that the QFS encodes the noise process characteristics, as such, all the coloured noise processes are significantly further away from the ‘ $1/f$  + bump’ noise processes than the ‘ $1/f$  + bump’ noise processes are from each other. On the other hand, each noise process has a similar distance metric to its corresponding non-stationary noise process. This indicates that the non-stationary and stationary noise processes result in similar parameters in the QFS, indicating that the QFS is not as sensitive to non-stationarity.

	1/f	1/f (NS)	1/f + bump	1/f + bump (NS)	Coloured	Coloured (NS)
$\tilde{X}$	0.40	<b>0.37</b>	0.38	0.38	0.83	0.92
$\tilde{Y}$	0.30	0.31	<b>0.25</b>	0.27	0.81	0.86
$\tilde{Z}$	<b>0.29</b>	0.30	0.30	0.29	0.69	0.62
Total	0.99	0.97	<b>0.93</b>	0.94	2.33	2.40

**Table 1:** Average Euclidean distances between realistic data points and the corresponding noise process values in the feature spaces of  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$ . For each  $\tilde{O}$  (row), the lowest (best) value is bolded. The total distance is the sum of the distances across all three feature spaces. The overall closest noise process is the ‘1/ $f$  + bump’ noise process, with the smallest total distance as bolded. NS indicates non-stationary noise processes.

	15 <sup>th</sup>	30 <sup>th</sup>	60 <sup>th</sup>	120 <sup>th</sup>	240 <sup>th</sup>	480 <sup>th</sup>
$\tilde{X}$	0.38	0.37	0.37	0.32	<b>0.28</b>	0.56
$\tilde{Y}$	0.24	0.24	0.23	0.15	<b>0.11</b>	0.58
$\tilde{Z}$	0.30	0.28	0.30	0.29	<b>0.28</b>	0.29
Total	0.92	0.89	0.90	0.76	<b>0.67</b>	1.43

**Table 2:** Average Euclidean distances between realistic data points and the corresponding noise process values in the feature spaces of  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$ . For each  $\tilde{O}$  (row), the lowest (best) value is bolded. The total distance is the sum of the distances across all three feature spaces. The overall closest noise process is the 240<sup>th</sup> noise profile, as indicated by the lowest total distance.

We also observed a relationship between the location of the simulated PSD peak and the average Euclidean distance in the QFS to the unknown process (which has a peak at the 200<sup>th</sup> bin). Specifically, noise processes with peaks closer to the 200<sup>th</sup> bin, such as the 190<sup>th</sup> and 210<sup>th</sup> bin profiles, resulted in smaller distances in the feature space (0.60 and 0.661, respectively). Conversely, peaks further away, like the 15<sup>th</sup> or 480<sup>th</sup> bins, corresponded to significantly larger distances (0.92 and 1.43). This correlation demonstrates that the QFS effectively encodes the key characteristics of the noise PSD and that the “distance” between noise processes in terms of their spectral properties (i.e. the location of the peak in the PSD) is meaningfully represented by distance in the QFS. This finding further strengthens the hypothesis that the QFS is a valuable tool for noise spectroscopy.

The results in this section demonstrate that one could simulate a broad class of noise processes to derive quantum features and compare them to quantum features from experimental data with an unknown noise process. Once the general noise process is found, a binary search could be performed to estimate specific parameters of this unknown noise process by minimising distance in the QFS.

	130 <sup>th</sup>	150 <sup>th</sup>	170 <sup>th</sup>	190 <sup>th</sup>	210 <sup>th</sup>	230 <sup>th</sup>
$\tilde{X}$	0.318	0.301	0.305	0.289	0.273	<b>0.267</b>
$\tilde{Y}$	0.155	0.127	0.101	0.081	<b>0.074</b>	0.111
$\tilde{Z}$	0.297	0.282	0.303	0.290	<b>0.278</b>	0.282
Total	0.770	0.710	0.709	0.660	<b>0.625</b>	0.661

**Table 3:** Average Euclidean distances between realistic data points and the corresponding noise process values in the feature spaces of  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$ . For each  $\tilde{O}$  (row), the lowest (best) value is bolded. The total distance is the sum of the distances across all three feature spaces. The overall closest noise process is the 210th noise profile, as indicated by the lowest total distance.

### 3.5 Explicit Comparison to Previous Work

The primary innovation of the QFS method is its dramatic reduction in model complexity and data requirements, directly addressing the key limitations of deep learning strategies [10, 13, 15]. As shown in Table 4, deep learning models employ neural networks with parameter counts in the tens or hundreds of thousands ( $10^4$ - $10^5$ ). Training these models demands a large number of examples (2100 - 3625 in the case of [15]). For noise identification, the QFS method consists of just 10 parameters and in our proof-of-concept, we correctly identified an unknown noise process using only 18 simulated examples, representing a reduction of several orders of magnitude in both model complexity and data needs, making it a far more practical solution for real-time, adaptive characterisation.

In terms of experimental overhead, our method strikes a balance between simplicity and robustness. It requires more control sequences than the minimal approach of Fourier-Transform Noise Spectroscopy (FTNS) [14], which can use a single free-induction decay. However, this trade-off yields significant resilience to experimental non-idealities. Our results were achieved using realistic CPMG pulses with inherent jitter and amplitude errors, demonstrating that the QFS is robust to the types of imperfections common in hardware. This practical robustness is a key advantage over methods that assume ideal conditions. Furthermore, the FTNS method assumes stationary, Gaussian noise, which can be a significant limitation in practical applications. In contrast, our QFS method is designed to handle a broader class of noise processes, including non-stationary and coloured noise.

Furthermore, the QFS framework avoids the “black box” problem of deep learning models. Our results show a clear, interpretable link between distance in the QFS and the physical properties of the noise spectrum (i.e., the location of the PSD peak). This is a step beyond a simple classification label.

It is important to note that the goal of our QFS method differs from that of some prior work. Techniques like FTNS aim for a full, high-resolution reconstruction of a specific

PSD. Our method, by contrast, is designed for efficient and robust classification and characterisation of the dominant noise family. As our results show, it can effectively identify the correct noise model and refine key parameters via a simple iterative search. While we observed that the current feature space is less sensitive to non-stationarity, its overall performance demonstrates that it fills a critical gap: providing a scalable, data-efficient, and computationally trivial tool for practical quantum noise analysis.

Attribute	FTNS [14]	NN Regression [10, 13]	Grey-box ML [15]	QFS (This Work)
Principle	Analytical inversion (Fourier transform)	Deep NN	Deep NN + analytical layers	Euclidean distance
Input Data	Free-induction decay or single-pulse echo	Coherence curves	Observables expectations	Observable expectations
Exp. Overhead	One pulse	One to several sequences	Several thousand examples	Tens of examples
Comput. Cost	Analytical - very low	NN training - high	NN training - high	Linear regression + distance calc - low
Key Advantage	High spectral resolution	High accuracy and robust denoising	High accuracy on non-Markovian noise	Data and computationally efficient, noise agnostic
Key Limitation	Assumes stationary, Gaussian noise; sensitive to pulse errors	Large datasets and "black box" nature	Large datasets required	Not explicitly designed for PSD reconstruction
Model Parameters	N/A	$\sim 10^4 - 10^5$ (estimate)	$\gtrsim 10^5$ (estimate)	10
Training Examples	N/A	Thousands	Thousands	18

**Table 4:** Comparison of qubit noise spectrum estimation and classification methods. Our QFS approach is highlighted for its exceptionally low data and model complexity.

### 3.6 Decision Tree for Noise Characterisation

We then explored the ability of simple ML algorithms to use this QFS. To do this, we generated 600 different noise processes and trained a random forest with  $K$ -Fold cross-validation [25, 26]. For the 600 distinct noise processes, 200 processes were  $1/f$ , 200 were ' $1/f + \text{bump}$ ', and 200 were coloured noise. For each noise process type, half were stationary, and half were non-stationary, i.e. 100  $1/f$  stationary noise processes and 100  $1/f$  non-stationary noise processes. We randomised several parameters between each noise profile to create distinct processes to ensure a diverse and representative dataset. Table 5 summarises the randomised parameters for each noise process and the range of values used. For the  $1/f$  noise processes, the exponent ( $\alpha$ ) was varied, and for the ' $1/f + \text{bump}$ ', the bump location in the frequency domain ( $\mu$ ) was varied alongside  $\alpha$ . For coloured noise, the division factor controlling the filter was randomised. For all non-stationary profiles, the peak position of the triangle envelope as a fraction of the total time ( $T$ ) was randomised.

Noise Profile Type	Parameter	Range/Value
$1/f$	$\alpha$	[0.7, 1.3]
' $1/f + \text{Bump}$ '	$\mu$	[0, 256]
	$\alpha$	[0.7, 1.3]
Coloured	Division Factor	[2, 16]
Non-Stationary	Peak of deterministic signal	[0.1 $T$ , 0.9 $T$ ]

**Table 5:** Summary of noise profile characteristics.

To analyse the noise profiles, we used a random forest classifier from the *scikit-learn* library [26]. The classifier was trained on the combined vector components across all three QFSs, employing a 10-fold cross-validation approach. Each sample was labelled for stationarity and noise type, where two different trees were trained for each classification task. The decision tree classifier for stationarity achieved an average test accuracy of 0.98, and the noise type classifier achieved an average test accuracy of 0.97.

The feature importances for stationarity and noise type classification are shown in Table 6. The feature importance is calculated as the sum of the reduction in the Gini impurity index across all nodes in the tree that use the feature [26]. Interestingly, for both classification tasks, only 3-4 features were needed to explain the majority of the variance in the data, where at least one feature per observable had high importance. This demonstrates how noise process characteristics manifest themselves in the QFS, which may assist in future work to clarify the mapping between a given noise process characteristic and the QFS.

Classification Task	$\alpha_X$	$\beta_X$	$\gamma_X$	$\alpha_Y$	$\beta_Y$	$\gamma_Y$	$\alpha_Z$	$\beta_Z$	$\gamma_Z$
Stationarity	0.22	0.08	0.16	0.08	0.15	0.03	0.05	0.08	0.15
Noise Type	0.09	0.14	0.07	0.15	0.13	0.08	0.03	0.03	0.28

**Table 6:** Feature importances represent the average contributions of the features  $\alpha$ ,  $\beta$ , and  $\gamma$  for each of the QFSs  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$  to the classification task. The feature importances are normalised to sum to 1.

For the sake of completeness, we also trained a K-Nearest Neighbors [27] and Logistic regression algorithms [28]. The results are shown in Table 8. As can be seen there is high accuracy across all the simple and inexpensive ML algorithms in both tasks which demonstrates the effectiveness of the QFS parameters in distinguishing between different noise processes and determining their stationarity. This is, as the feature space is well-defined and captures the essential characteristics of the noise, simple algorithms can easily leverage this structure and achieve high classification accuracy. These results further validate our approach and provides more robust methods for characterising and classifying noise in quantum systems.

**Table 7:** Algorithm Accuracy Comparison for Classification Tasks

Algorithm	Stationarity Classification	Noise Type Classification
Decision Tree	0.98	0.97
Logistic Regression	0.84	0.91
K-Nearest Neighbors	0.96	0.94

**Table 8:** Comparison of classification accuracies for different algorithms. The decision tree classifier achieved the highest accuracy in both tasks, stationarity and noise type classification, with K-Nearest Neighbours following closely. Logistic regression performed well but was less accurate than the other two algorithms.

### 3.7 Properties of the QFS

We sought to understand the mapping between control and noise parameters and the QFS. For these experiments, we explored the impact of widening control pulse widths, which simulated less ideal delta function pulses, the interpolation between noise profiles, and the impact of increasing noise energy.

We first explored the impact of widening control pulse widths on the QFS. The control pulse width was varied by changing the value of  $\lambda$  in Eq. (22). We set  $\lambda = \frac{1}{96}, \frac{1}{48}, \frac{1}{24}, \frac{1}{12}, \frac{1}{6}$ , and  $\frac{1}{3}$ . For these simulations, we used the ‘ $1/f +$  bump’ with a bump in the PSD at the 200<sup>th</sup> bin (the unknown noise process from Sec. 3.4). Fig. 3(a) illustrates the impact of widening control pulse widths. The figure shows that the interpolation between data points arising from widening control pulses is smooth and well-ordered, providing evidence that the QFS is well-behaved, where it is known that smooth interpolation in a feature space is a highly desirable characteristic [29–32].

The results in Fig. 3(b) are similar and show a smooth interpolation between the  $1/f$  noise process with a bump and the coloured Gaussian noise process. These interpolations were achieved by creating various linear combinations of the noise processes at different ratios, progressively transitioning from one to the other.

We then investigated the impact of increasing the signal energy of the ‘ $1/f +$  bump’ and coloured Gaussian noise processes. Signal energy here is defined as,

$$E = \sum_{t=1}^N |n_t|^2 \quad (23)$$

where  $n_t$  is the noise process value at time step  $t$  and  $N$  is the total number of time steps. To implement increasing energy, we scaled all values of a given noise realisation (i.e. all  $n_t$ ) by a constant using the following scale factors: 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75.

Results for increasing noise strength experiments are shown in Fig. 3(c), where increasing colour saturation represents increasing energy. For the coloured noise process and ‘ $1/f +$  bump’, as the noise strength increases, all points tend toward the centre of the feature space, where  $\alpha_O$ ,  $\beta_O$ , and  $\gamma_O$  are all equal to zero. Observe that when  $\alpha_O$ ,  $\beta_O$ , and  $\gamma_O = 0$ ,  $\tilde{O}$  becomes a zero matrix, and looking to Eq. (13), we find that the expectation of the observable is zero. This indicates that as the noise process increases in energy, it saturates the system, and as such, all information about the noise process and control is lost. Conversely, as the noise strength decreases, points move toward the associated identity in the QFS and therefore result in the system dynamics being dominated by the control pulses.

However, the ‘ $1/f +$  bump’ noise process shows a more complex behaviour in Fig. 3(c) when compared to the coloured noise process. The red dots, which represent the ‘ $1/f +$  bump’ noise process, rotate around  $\tilde{X}$  and  $\tilde{Y}$  as the noise strength increases. This rotation is likely due to the spectral bump in the noise profile, which introduces off-diagonal components into the noise operator. In a quantum system under control

pulses, such off-diagonal elements can lead to phase shifts or rotations in the operator representation. Essentially, the result shows that the ‘ $1/f$  + bump’ noise process does not simply scale the operator uniformly; it alters its orientation in the feature space.

It is interesting to note that for the  $1/f$  noise process, the second strongest noise point is closest to the identity (rather than the weakest noise point). One explanation of this phenomenon is that the interplay between the noise’s rotational effect and its contracting effect (i.e., the tendency to “wash out” information) is nonlinear. As such, at a particular noise level, the destructive interference between the control dynamics and the noise perturbations may nearly cancel out.

For both noise processes, behaviour in the  $\tilde{Z}$  feature space is more intuitive; points get closer to the identity as the noise decreases, indicating that the control pulses are dominating system dynamics. Overall, this nuanced behaviour seen in these experiments highlights the richness of the QFS and suggests that further study—perhaps through controlled variations or analytical modelling—could provide deeper insight into these dynamics.

## 4 Conclusion and Future Work

This work has introduced a QFS based on the noise operator formalism. To create this QFS, we extended the noise operator formalism to map experimental measurements directly to system-bath interaction operator parameters. We visualised the QFS, mapping its behaviour to control pulse and noise process parameters. It was found that QFS can be illuminating and used to identify unknown qubit noise processes without making restrictive assumptions, and is a promising tool for future noise characterisation.

Future work could further refine control techniques within the QFS, enhancing the ability to mitigate noise effects. In particular, exploring different types of control pulses and their parameters could provide deeper insights into optimising qubit control for various noise environments. This also includes studying a broader range of noise processes to further validate the developed methodology. Incorporating non-stationary and non-Gaussian noise models further helps us understand the QFS. The methods developed could also be applied and studied in higher-dimensional quantum systems (i.e. multi-qubit systems) in both simulated and experimental settings, to investigate if the QFS can be used for higher-dimensional systems.

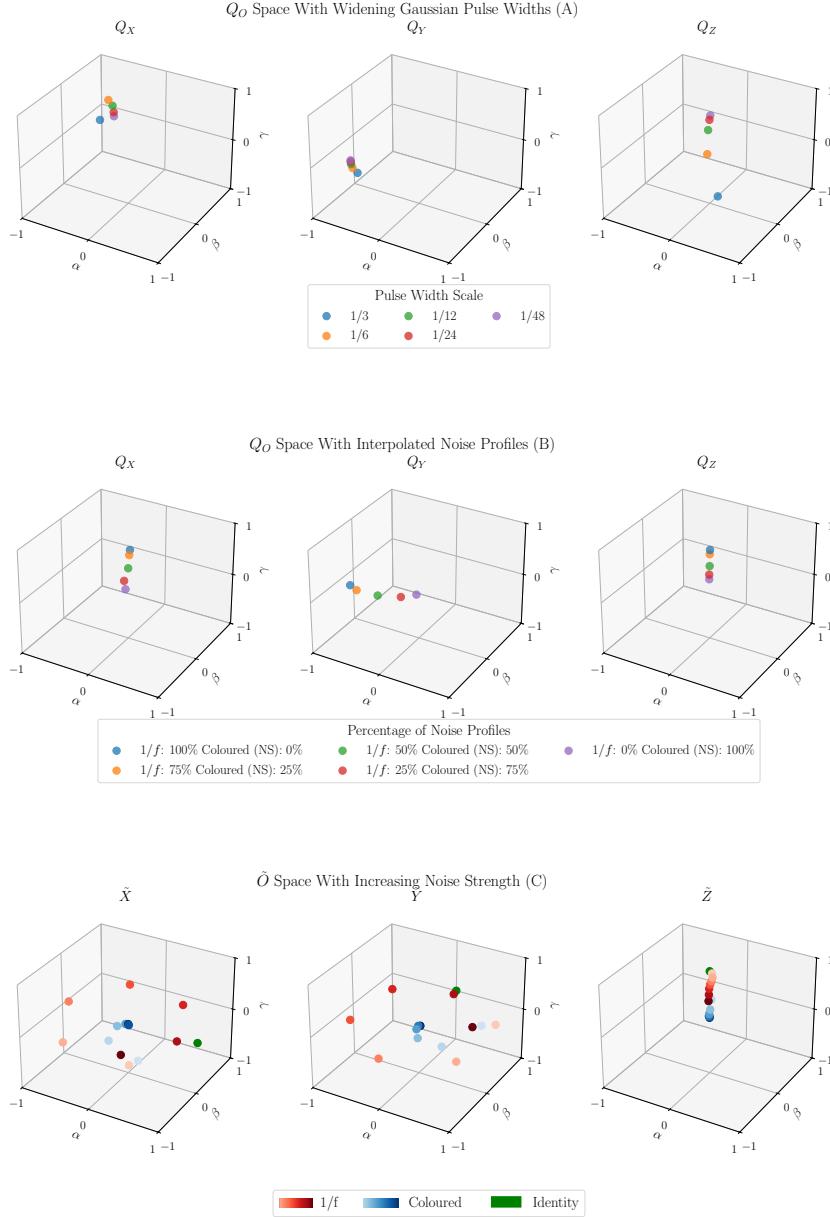
## 5 Data and Code Availability

The data and code are available here:

[https://github.com/ChrisWise07/quantum\\_feature\\_space](https://github.com/ChrisWise07/quantum_feature_space)

## 6 Funding

MW acknowledges support from the Australian Research Council (ARC) Centre of Excellence for Engineered Quantum Systems (CE170100009).



**Fig. 3:** (a) Visualisation of the QFS with widening Gaussian-shaped control pulses, with the fraction indicating the scale factor. (b) Visualisation of the QFS with interpolation between noise processes implemented by utilising a linear combination of the ‘1/f + bump’ and coloured Gaussian noise processes. (c) Visualisation of the quantum parameter space with two different noise processes, ‘1/f + bump’ and coloured Gaussian, with colour saturation representing increasing energy.

AP acknowledges an RMIT University Vice-Chancellor's Senior Research Fellowship and a Google Faculty Research Award. This work was supported by the Australian Government through the Australian Research Council under the Centre of Excellence scheme (CE170100012).

## 7 References

### References

- [1] Bravyi, S., Englbrecht, M., König, R., Peard, N.: Correcting coherent errors with surface codes. *npj Quantum Information* **4**(1), 1–6 (2018)
- [2] Viola, L., Knill, E., Lloyd, S.: Dynamical decoupling of open quantum systems. *Physical Review Letters* **82**(12), 2417 (1999)
- [3] Degen, C.L., Reinhard, F., Cappellaro, P.: Quantum sensing. *Reviews of modern physics* **89**(3), 035002 (2017)
- [4] Boss, J.M., Chang, K., Armijo, J., Cujia, K., Rosskopf, T., Maze, J.R., Degen, C.L.: One-and two-dimensional nuclear magnetic resonance spectroscopy with a diamond quantum sensor. *Physical review letters* **116**(19), 197601 (2016)
- [5] Carr, H.Y., Purcell, E.M.: Effects of diffusion on free precession in nuclear magnetic resonance experiments. *Physical review* **94**(3), 630 (1954)
- [6] Meiboom, S., Gill, D.: Modified spin-echo method for measuring nuclear relaxation times. *Review of scientific instruments* **29**(8), 688–691 (1958)
- [7] Álvarez, G.A., Suter, D.: Measuring the spectrum of colored noise by dynamical decoupling. *Physical review letters* **107**(23), 230501 (2011)
- [8] Lüpke, U., Beaudoin, F., Norris, L.M., Sung, Y., Winik, R., Qiu, J.Y., Kjaergaard, M., Kim, D., Yoder, J., Gustavsson, S., *et al.*: Two-qubit spectroscopy of spatiotemporally correlated quantum noise in superconducting qubits. *PRX Quantum* **1**(1), 010305 (2020)
- [9] Sun, W.K.C., Cappellaro, P.: Self-consistent noise characterization of quantum devices. *Physical Review B* **106**(15), 155413 (2022)
- [10] Wise, D.F., Morton, J.J., Dhomkar, S.: Using deep learning to understand and mitigate the qubit noise environment. *PRX Quantum* **2**(1), 010316 (2021)
- [11] Martina, S., Gherardini, S., Caruso, F.: Machine learning classification of non-markovian noise disturbing quantum dynamics. *Physica Scripta* **98**(3), 035104 (2023)
- [12] Baroni, A., Carlson, J., Gupta, R., Li, A.C., Perdue, G.N., Roggero, A.: Nuclear two point correlation functions on a quantum computer. *Physical Review D*

**105**(7), 074503 (2022)

- [13] Martina, S., Hernández-Gómez, S., Gherardini, S., Caruso, F., Fabbri, N.: Deep learning enhanced noise spectroscopy of a spin qubit environment. *Machine Learning: Science and Technology* **4**(2), 02–01 (2023)
- [14] Vezvae, A., Shitara, N., Sun, S., Montoya-Castillo, A.: Fourier transform noise spectroscopy. *npj Quantum Information* **10**(1), 52 (2024)
- [15] Youssry, A., Paz-Silva, G.A., Ferrie, C.: Characterization and control of open quantum systems beyond quantum noise spectroscopy. *npj Quantum Information* **6**(1), 95 (2020)
- [16] Youssry, A., Nurdin, H.I.: Multi-axis control of a qubit in the presence of unknown non-markovian quantum noise. *Quantum Science and Technology* **8**(1), 015018 (2022)
- [17] Seedhouse, A.E., Stuyck, N.D., Serrano, S., Gilbert, W., Huang, J.Y., Hudson, F.E., Itoh, K.M., Laucht, A., Lim, W.H., Yang, C.H., *et al.*: Wavelet correlation noise analysis for qubit operation variable time series. *Scientific Reports* **15**(1), 11065 (2025)
- [18] Nakamura, K., Ankerhold, J.: Gate operations for superconducting qubits and non-markovianity. *Physical Review Research* **6**(3), 033215 (2024)
- [19] Sung, Y., Beaudoin, F., Norris, L.M., Yan, F., Kim, D.K., Qiu, J.Y., Lüpke, U., Yoder, J.L., Orlando, T.P., Gustavsson, S., *et al.*: Non-gaussian noise spectroscopy with a superconducting qubit sensor. *Nature communications* **10**(1), 3715 (2019)
- [20] Paz-Silva, G.A., Norris, L.M., Viola, L.: Multiqubit spectroscopy of gaussian quantum noise. *Physical review a* **95**(2), 022121 (2017)
- [21] Penrose, R.: A generalized inverse for matrices. In: *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 406–413 (1955). Cambridge University Press
- [22] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., *et al.*: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
- [23] Perrier, E., Youssry, A., Ferrie, C.: Qdataset, quantum datasets for machine learning. *Scientific Data* **9**(1), 582 (2022)
- [24] Harris, M., Sengupta, S., Owens, J.D.: Parallel prefix sum (scan) with cuda. *GPU gems* **3**(39), 851–876 (2007)
- [25] James, G.: *An introduction to statistical learning*. Springer (2013)

- [26] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011)
- [27] Peterson, L.E.: K-nearest neighbor. *Scholarpedia* **4**(2), 1883 (2009)
- [28] Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: *Applied Logistic Regression*. John Wiley & Sons, ??? (2013)
- [29] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828 (2013)
- [30] Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015)
- [31] Guo, J., Xu, X., Pu, Y., Ni, Z., Wang, C., Vasu, M., Song, S., Huang, G., Shi, H.: Smooth diffusion: Crafting smooth latent spaces in diffusion models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7548–7558 (2024)
- [32] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: *International Conference on Learning Representations* (2017)

## 8 Appendix

### 8.1 Derivation of Total Unitary Decomposition

We start with the total Hamiltonian describing a controlled qubit interacting with a bath:

$$H(t) = H_{\text{ctrl}}(t) + H_{\text{noise}}(t)$$

Here, the control Hamiltonian  $H_{\text{ctrl}}(t)$  includes both the qubit's free evolution and external control pulses:

$$H_{\text{ctrl}}(t) = \Omega \frac{\sigma_z}{2} + \sum_{j=\{x,y,z\}} f_j(t) \frac{\sigma_j}{2}$$

where  $\Omega$  represents the qubit's energy gap,  $f_j(t)$  is a control pulse applied along the  $j$ -axis, and  $\sigma_j$  are the Pauli matrices. The noise Hamiltonian  $H_{\text{noise}}(t)$  describes the

interaction with the bath and is given by:

$$H_{\text{noise}}(t) = \sum_{j=\{x,y,z\}} \beta_j(t) \sigma_j$$

where  $\beta_j(t)$  represents a classical stochastic process.

The time evolution of a quantum state vector  $|\psi(t)\rangle$  in the Schrödinger picture is governed by the time-dependent Schrödinger equation (setting  $\hbar = 1$ ):

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

The solution to this equation can be expressed using the total time-evolution operator  $U(t)$ , such that  $|\psi(t)\rangle = U(t)|\psi(0)\rangle$ . The operator  $U(t)$  satisfies its own differential equation:

$$i \frac{d}{dt} U(t) = H(t) U(t)$$

with the initial condition  $U(0) = I$ . For a time-dependent Hamiltonian like  $H(t)$ , the formal solution for  $U(t)$  involves the time-ordering operator such that,

$$U(t) = \mathcal{T} e^{-i \int_0^t H(s) ds}$$

First, let's consider the unitary operator that describes the evolution solely under the control Hamiltonian  $H_{\text{ctrl}}(t)$ , which we denote as  $U_{\text{ctrl}}(t)$ :

$$i \frac{d}{dt} U_{\text{ctrl}}(t) = H_{\text{ctrl}}(t) U_{\text{ctrl}}(t)$$

with the initial condition  $U_{\text{ctrl}}(0) = I$ , where  $U_{\text{ctrl}}(t) = \mathcal{T} e^{-i \int_0^t H_{\text{ctrl}}(s) ds}$

To isolate the effects of the noise Hamiltonian, we define a new state vector in the interaction picture,  $|\psi_I(t)\rangle$ . This definition effectively “rotates away” the time evolution due to the control Hamiltonian  $H_{\text{ctrl}}(t)$ :

$$|\psi_I(t)\rangle \equiv U_{\text{ctrl}}^\dagger(t) |\psi(t)\rangle$$

From this definition, we can express the Schrödinger picture state in terms of the interaction picture state:

$$|\psi(t)\rangle = U_{\text{ctrl}}(t) |\psi_I(t)\rangle$$

Now, we differentiate  $|\psi_I(t)\rangle$  with respect to time using the product rule:

$$i \frac{d}{dt} |\psi_I(t)\rangle = i \left( \frac{dU_{\text{ctrl}}^\dagger(t)}{dt} \right) |\psi(t)\rangle + i U_{\text{ctrl}}^\dagger(t) \left( \frac{d|\psi(t)\rangle}{dt} \right)$$

Simplifying each term, and using the property  $i\frac{d}{dt}U_{\text{ctrl}}(t) = H_{\text{ctrl}}(t)U_{\text{ctrl}}(t)$ ,

$$i \left( \frac{dU_{\text{ctrl}}^\dagger(t)}{dt} \right) = - \left( i \frac{dU_{\text{ctrl}}(t)}{dt} \right)^\dagger = -(H_{\text{ctrl}}(t)U_{\text{ctrl}}(t))^\dagger = -U_{\text{ctrl}}^\dagger(t)H_S^\dagger(t)$$

Since  $H_{\text{ctrl}}(t)$  is Hermitian,  $H_{\text{ctrl}}^\dagger(t) = H_{\text{ctrl}}(t)$ , so this simplifies,

$$i \left( \frac{dU_{\text{ctrl}}^\dagger(t)}{dt} \right) = -U_{\text{ctrl}}^\dagger(t)H_{\text{ctrl}}(t)$$

For the second term, we substitute the Schrödinger equation  $i\frac{d|\psi(t)\rangle}{dt} = H(t)|\psi(t)\rangle$ ,

$$iU_{\text{ctrl}}^\dagger(t) \left( \frac{d|\psi(t)\rangle}{dt} \right) = U_{\text{ctrl}}^\dagger(t)H(t)|\psi(t)\rangle = U_{\text{ctrl}}^\dagger(t)(H_{\text{ctrl}}(t) + H_{\text{noise}}(t))|\psi(t)\rangle$$

Substituting these simplified terms back into the derivative of  $|\psi_I(t)\rangle$ ,

$$\begin{aligned} i\frac{d}{dt}|\psi_I(t)\rangle &= -U_{\text{ctrl}}^\dagger(t)H_{\text{ctrl}}(t)|\psi(t)\rangle + U_{\text{ctrl}}^\dagger(t)(H_{\text{ctrl}}(t) + H_{\text{noise}}(t))|\psi(t)\rangle \\ &= -U_{\text{ctrl}}^\dagger(t)H_{\text{ctrl}}(t)|\psi(t)\rangle + U_{\text{ctrl}}^\dagger(t)H_{\text{ctrl}}(t)|\psi(t)\rangle + U_{\text{ctrl}}^\dagger(t)H_{\text{noise}}(t)|\psi(t)\rangle \\ &= U_{\text{ctrl}}^\dagger(t)H_{\text{noise}}(t)|\psi(t)\rangle \end{aligned} \tag{24}$$

Finally, substitute  $|\psi(t)\rangle = U_{\text{ctrl}}(t)|\psi_I(t)\rangle$  into the expression:

$$i\frac{d}{dt}|\psi_I(t)\rangle = U_{\text{ctrl}}^\dagger(t)H_{\text{noise}}(t)U_{\text{ctrl}}(t)|\psi_I(t)\rangle$$

The above equation is a Schrödinger-like equation for  $|\psi_I(t)\rangle$ . The evolution of the interaction picture state is governed by an effective Hamiltonian, which we define as the interaction Hamiltonian  $H_I(t)$ :

$$H_I(t) = U_{\text{ctrl}}^\dagger(t)H_{\text{noise}}(t)U_{\text{ctrl}}(t)$$

So, the equation of motion for the interaction picture state becomes

$$i\frac{d}{dt}|\psi_I(t)\rangle = H_I(t)|\psi_I(t)\rangle$$

The time evolution of  $|\psi_I(t)\rangle$  is then described by the interaction unitary operator  $U_I(t)$ ,

$$|\psi_I(t)\rangle = U_I(t)|\psi_I(0)\rangle$$

where  $U_I(t)$  satisfies  $i \frac{d}{dt} U_I(t) = H_I(t)U_I(t)$  with  $U_I(0) = I$ , and is formally given by,

$$U_I(t) = \mathcal{T} e^{-i \int_0^t H_I(s) ds}$$

Finally, we know that the state at time  $t$  in the Schrödinger picture is given by:

$$|\psi(t)\rangle = U(t)|\psi(0)\rangle$$

From our definition of the interaction state, we also have:

$$|\psi(t)\rangle = U_{\text{ctrl}}(t)|\psi_I(t)\rangle$$

Substitute the expression for  $|\psi_I(t)\rangle$  from the previous step,

$$|\psi(t)\rangle = U_{\text{ctrl}}(t)U_I(t)|\psi_I(0)\rangle$$

Now at  $t = 0$ , we have  $|\psi_I(0)\rangle = U_{\text{ctrl}}^\dagger(0)|\psi(0)\rangle$ . Since  $U_{\text{ctrl}}(0) = I$ , then  $U_{\text{ctrl}}^\dagger(0) = I$ , which means  $|\psi_I(0)\rangle = |\psi(0)\rangle$ . Substituting this back into the equation for  $|\psi(t)\rangle$ :

$$|\psi(t)\rangle = U_{\text{ctrl}}(t)U_I(t)|\psi(0)\rangle$$

By comparing this with  $|\psi(t)\rangle = U(t)|\psi(0)\rangle$ , we arrive at the desired decomposition,

$$U(t) = U_{\text{ctrl}}(t)U_I(t)$$

This result demonstrates how the total time-evolution operator can be rigorously separated into contributions from the system's free evolution (and control pulses) and the interaction with the noise, without assuming any commutativity between the control Hamiltonian  $H_{\text{ctrl}}(t)$  and the noise Hamiltonian  $H_{\text{noise}}(t)$ . The non-commutativity is naturally accounted for in the definition of the interaction Hamiltonian  $H_I(t)$ .