Quantum Feature Space of a Qubit Coupled to a Classical Stochastic Environment

Chris Wise[1*], Akram Youssry[2], Alberto Peruzzo[2,3], Jo Plested[1], Matt Woolley[4]

[1]School of Systems and Computing, UNSW, Canberra, ACT, 2601, Australia.
[2]Quantum Photonics Laboratory and Centre for Quantum Computation and Communication Technology, RMIT University, Melbourne, VIC, 3000, Australia.
[3]Quandela, Massy, France.
[4]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;

Abstract

Qubit control protocols traditionally characterise qubit-environment coupling using resource-intensive pulse sweeps on quantum hardware. Although previous work attempted to reduce this quantum runtime by inferring noise operators, it relied on a complex grey-box approach combining deep neural networks with physics-encoded layers, posing challenges in training and scaling. Here, we introduce a *quantum feature space* (QFS, denoted $\boldsymbol{\mathcal{F}}$) constructed from a single pulse sequence. By engineering $\boldsymbol{\mathcal{F}}$ to encode qubit-environment dynamics, we eliminate the need for expensive neural networks to learn these physical interactions. This allows us to offload extraction of spectral features to efficient classical machine learning algorithms. Bypassing both the extensive pulse sweeps of traditional quantum noise spectroscopy and the costly neural networks of prior methods, we demonstrate that a simple random forest algorithm on $\boldsymbol{\mathcal{F}}$ classifies broad noise classes with $\sim$ **90%** accuracy under simulated realistic hardware constraints, including pulse jitter and readout errors. Furthermore, Euclidean distance searches over $\boldsymbol{\mathcal{F}}$ can determine spectral resonances. Focusing on noise classification and identification rather than continuous power spectral density reconstruction makes $\boldsymbol{\mathcal{F}}$ an efficient diagnostic tool for quantum hardware.

Keywords: noise spectroscopy, quantum features, classification, NISQ hardware

1 Introduction

Noise in quantum computers that results in qubits losing coherence (decoherence) is due to unwanted environment coupling [1]. One way to mitigate decoherence is to decouple the qubit from the environment by applying a control field in the form of a series of pulses [2]. This technique originated in nuclear magnetic resonance [3–6]. Within this framework, N_p refocusing π-pulses are applied separated by free-evolution delays, forming the Carr–Purcell–Meiboom–Gill (CPMG) sequence [3, 4].

Álvarez and Suter (AS) established dynamical decoupling noise spectroscopy by applying CPMG sequences with varied inter-pulse delays to sample the environmental spectrum [7]. In traditional spectroscopy protocols, empirical coherence data obtained from dynamical decoupling sequences are either inverted via filter-function matrices [7–9] or fitted to theoretical coherence curves corresponding to parametric noise spectra. However, this method requires an accurate model of qubit 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 qubit dynamics, and subsequently, control pulses optimised using this qubit 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 [8, 12]. The Fourier transform of this correlation function yields 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].

Vezvaee et al. [14] developed Fourier transform noise spectroscopy (FTNS), which applies the Fourier transform to free-induction decay (FID) coherence measurements. 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 its applicability. For example, the authors assumed that 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.

Instead of analytical methods, Martina et al. [13] 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.

Wise et al. [10] also used deep learning techniques to infer accurate qubit noise spectra. The method used measurement data obtained from CPMG coherence decay 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 coherence curves and outperformed the standard noise spectroscopy techniques.

Beyond these model-specific limitations, an overarching bottleneck of traditional and deep learning approaches is that constructing full coherence decay curves still requires sweeping evolution times or inter-pulse delays across many separate sequences at inference time. This exhaustive sampling can be expensive in terms of quantum

computer resources, posing a challenge for the frequent calibration required by noisy intermediate-scale quantum (NISQ) devices, as noise processes in qubits can drift over time [15, 16]. To circumvent the need for full coherence curve sweeps, researchers have explored alternative architectures that operate on individual control sequences.

Youssry et al. [17] introduced a grey-box model to separate control and qubit-environment dynamics. In this work, time-dependent Hamiltonians and unitaries specified the control dynamics, and a recurrent neural network was used to predict the qubit-environment dynamics associated with control pulses. Once trained on a library of pulse sequences, such a model requires only a single arbitrary control sequence as input to predict the corresponding noisy evolution.

Despite the success of such frameworks [17, 18], these approaches are ultimately limited by their use of deep learning models. These models are often large and require extensive datasets, making them expensive to train. This is undesirable due to the drift in noise dynamics, which necessitates frequent retraining. Additionally, noise profiles may change from one physical qubit to another [19], requiring qubit-specific models, which do not scale efficiently. Furthermore, specific to the grey-box method of [17, 18], the white-box aspect scales exponentially with the number of qubits.

This variability of qubit noise profiles and the scaling challenges of deep learning models motivate the need for efficient alternatives. To develop such efficient noise protocols, we model a classical stochastic environment. In contrast, recent theoretical developments have characterised quantum environments without relying on classical stochastic realisations [12, 20]. While classical noise produces mixed states via trajectory averaging, a quantum environment naturally captures more complex interactions such as non-unital energy relaxation through entanglement and back-action with the environment. However, a classical stochastic environment is the standard assumption in qubit control [21, 22], and keeps the model simple and computationally efficient. This simplicity aligns with the objective to provide a "cheap and fast" diagnostic tool suitable for practical, real-time noise identification on NISQ hardware, rather than an exhaustive quantum description of the environment.

We distinguish between three objectives: *noise classification*, which assigns the noise to a broad physical category (e.g., $1/f$ versus coloured noise); *noise identification*, which focuses on finding specific parameters (such as the location of a resonance peak) within an assumed model; and *PSD reconstruction*, which involves mapping the continuous functional form of the noise spectrum. Traditional spectroscopy focusing on full PSD reconstruction drives up computational and experimental costs. Here, we shift the focus to classification and identification to enable efficient operations.

For this classification and identification task, we introduce a quantum feature space ($\mathcal{F}$) that encodes the environment's influence on qubit dynamics, avoiding the overhead of continuous parameter extraction. We show that no expensive neural networks are needed; instead, simple machine learning algorithms, such as random forests, combined with a single pulse sequence, can quickly classify the noise affecting a qubit (Fig. 1).

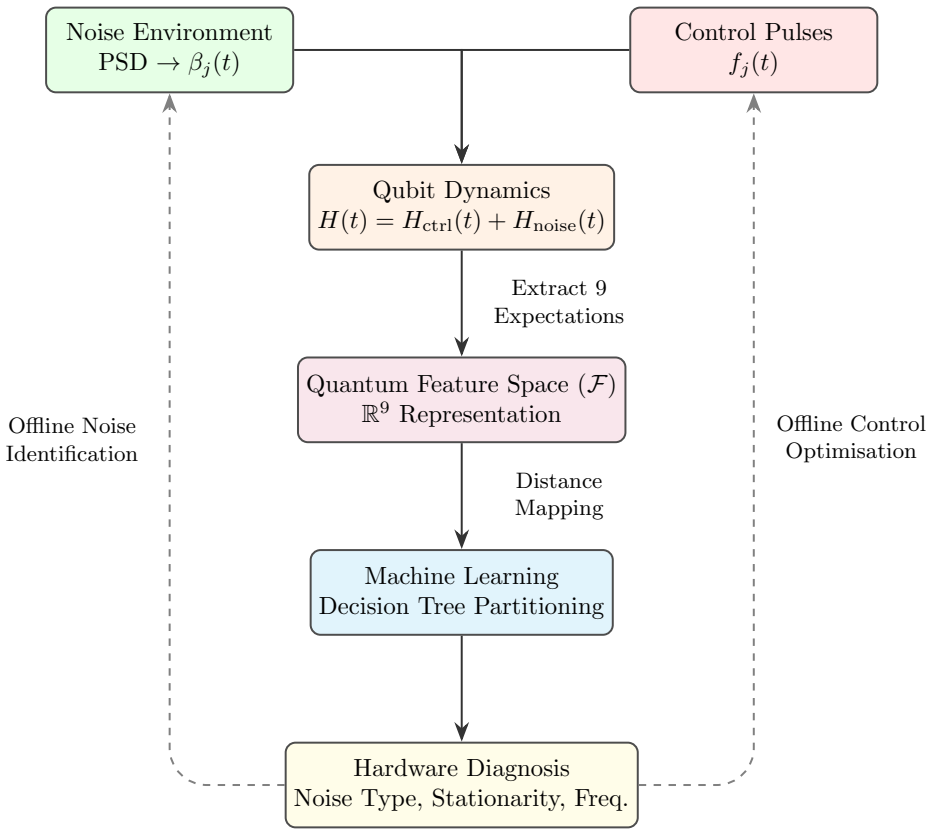


Fig. 1 Schematic of the Quantum Feature Space ($\mathcal{F}$) diagnostic framework. Environmental noise realisations and targeted control pulses govern the qubit dynamics. The resulting expectation values for states prepared along orthogonal Bloch axes and Pauli observables are mapped into a structured 9-dimensional feature space ($\mathcal{F}$). This parameterisation allows efficient classical machine learning algorithms, such as Random Forests, to rapidly classify complex noise types and determine underlying physical parameters without the overhead of continuous spectrum reconstruction. This framework facilitates rapid offline noise identification, which can be leveraged for offline control pulse optimisation via digital simulations of a physical quantum computer.

2 Deriving the Quantum Feature Space

We model a single qubit perturbed by classical noise processes (using units where $\hbar = 1$). The total Hamiltonian is given by

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

where the control Hamiltonian is

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

and we operate in a rotating frame with $f_j(t)$ modelling externally applied control pulses. The qubit-environment interaction is defined as

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

where $\beta_j(t)$ is a stochastic process representing noise.

We are interested in the expected value of a qubit observable O at the final time T of the control pulse sequence, given an initial state ρ,

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

where $U(T) = \mathcal{T}e^{-i\int_0^T H(s)\,ds}$ is the time-ordered evolution operator, and $\langle\cdot\rangle$ denotes classical averaging over the stochastic realisations of $\beta_j(t)$.

We initialise the qubit along the positive principal axes by writing $\rho_{j_+} = \frac{1}{2}(I+\sigma_j)$ with $j \in \{x,y,z\}$, and limit the observable to the Pauli operators, σ_k with $k \in \{x,y,z\}$,

$$\mathbb{E}\{\sigma_k\}_{\rho_{j_+}} = \frac{1}{2}\left\langle \text{Tr}\left[U(I+\sigma_j)U^\dagger\sigma_k\right]\right\rangle = \frac{1}{2}\langle \text{Tr}[\sigma_k]\rangle + \frac{1}{2}\langle \text{Tr}[U\sigma_j U^\dagger\sigma_k]\rangle, \tag{5}$$

where, for brevity, we have omitted T, and operators are understood to be evaluated at the final time T unless otherwise specified. Since $\text{Tr}[\sigma_k] = 0$ for all $k \in \{x,y,z\}$,

$$\mathbb{E}\{\sigma_k\}_{\rho_{j_+}} = \frac{1}{2}\langle \text{Tr}[U\sigma_j U^\dagger\sigma_k]\rangle, \tag{6}$$

yielding nine independent measurements corresponding to $j,k \in \{x,y,z\}$. We then define the observable-specific feature vector as

$$\mathbf{v}_{\sigma_k} = \left[\mathbb{E}\{\sigma_k\}_{\rho_{x_+}}, \mathbb{E}\{\sigma_k\}_{\rho_{y_+}}, \mathbb{E}\{\sigma_k\}_{\rho_{z_+}}\right]^T. \tag{7}$$

The quantum feature space vector $\mathcal{F}$ is then defined as the concatenation of these vectors,

$$\mathcal{F} = \text{vec}\left([\mathbf{v}_{\sigma_x}, \mathbf{v}_{\sigma_y}, \mathbf{v}_{\sigma_z}]\right). \tag{8}$$

This yields a 9-dimensional vector that captures the qubit-environment dynamics and incorporates hardware non-idealities, such as state preparation and measurement (SPAM) and control pulse errors, without the overhead of full state or process reconstruction. Importantly, rather than representing a static quantum state, $\mathcal{F}$ provides a compact, control-conditioned representation of the qubit-environment interaction.

We prepare only the positive eigenstates $\rho_{j_+} = \frac{1}{2}(I+\sigma_j)$ because the negative eigenstates $\rho_{j_-} = \frac{1}{2}(I-\sigma_j) = I - \rho_{j_+}$ yield redundant measurements under unital dynamics. Specifically, for traceless Pauli observables $O = \sigma_k$, any expectation measured from ρ_{j_-} satisfies $\mathbb{E}\{\sigma_k\}_{\rho_{j_-}} = -\mathbb{E}\{\sigma_k\}_{\rho_{j_+}}$.

2.1 Operator Interpretation

Although $\mathcal{F}$ is defined in vector format for efficient storage and direct compatibility with classifiers, it admits a direct physical interpretation. For any unitary $U \in \mathrm{SU}(2)$, the trace inner product $R_{kj} = \frac{1}{2}\mathrm{Tr}[U\sigma_j U^\dagger \sigma_k]$ defines the (k, j) entry of an SO(3) rotation matrix. Averaging over the stochastic realisations yields a 3×3 matrix with entries $T_{kj} = \mathbb{E}\{\sigma_k\}_{\rho_{j+}}$, representing the ensemble-averaged rotation matrix of the channel. While each individual realisation is a unitary rotation, their ensemble average T is a contractive matrix encoding decoherence. Consequently, the Euclidean distance between two feature vectors in $\mathcal{F}$ is identical to the Frobenius distance (and proportional to the Hilbert–Schmidt distance) between their corresponding 3×3 channel matrices.

3 Numerical Simulation Implementation

Our quantum simulator followed the formalism in Sec. 2 and was implemented natively in PyTorch [23]. The codebase is available at https://github.com/ChrisWise07/quantum_feature_space. Following [24], we discretised the time interval $[0, T]$ into $M = 1024$ uniform timesteps of duration $\Delta t = T/M$, defining the discrete time grid $t_m = m\Delta t$ for $m \in \{0, 1, \ldots, M-1\}$, with the total sequence duration normalised to $T = 1.0$.

3.1 Creation of Noise Data

In our simulations, we implemented the following noise Hamiltonian [24],

$$H_{\text{noise}}(t_m) = \beta_x(t_m)\sigma_x + |\beta_x(t_m)|\sigma_z, \tag{9}$$

to couple both transverse and longitudinal noise to a common physical source.

Inspired by Perrier, Youssry, and Ferrie [24], we explored three distinct physical noise profiles:

- **$1/f$ noise:** Representing typical charge and flux noise fluctuations with power spectral density $\propto 1/f^\alpha$ commonly found in superconducting qubits [25].
- **$1/f$ noise with a Gaussian bump:** Simulating a background $1/f^\alpha$ spectrum superimposed with a high-frequency resonance centred at frequency bin μ, modelling a strongly coupled individual fluctuator [26, 27].
- **Gaussian coloured noise:** Serving as a distinctly varied environment to test classification boundaries.

Gaussian coloured noise was generated by applying a moving-average filter of length M/d to white Gaussian noise, where the division factor $d \in \{2, 3, 4, 5\}$ controls the correlation length. The Timmer–König method [28] was used for generating $1/f$ noise frequency-domain realisations, obtaining the time-domain sequences via inverse Fourier transform.

Similar to [24], we created both stationary and non-stationary versions of each noise profile to test how non-stationarity manifests in $\mathcal{F}$ and affects classification performance. Non-stationarity was introduced by multiplying the stationary noise time

series realisations by a Gaussian time envelope $g(t_m) = \exp[-(t_m - cT)^2/(2\sigma_t^2)]$, where $\sigma_t = T/6$, with peak position c sampled uniformly from $[0.2, 0.8]$.

Using this formulation, we generated 600 distinct noise processes (200 of each physical type, divided equally between stationary and non-stationary profiles). For each distinct noise process, an ensemble of up to $K = 4096$ independent realisations was generated.

To guarantee a diverse and representative dataset, the defining parameters for each of the 600 noise profiles were randomised. Tab. 1 outlines the exact parameter bounds utilised in the dataset generation.

Noise Profile Type	**Parameter**	**Range/Value**
$1/f$	α (exponent)	$[0.5, 1.5]$
$1/f$ + Bump	μ (bump frequency bin)	$[5.0, 25.0]$
	α (exponent)	$[0.5, 1.5]$
Coloured	d (division factor)	$\{2, 3, 4, 5\}$
Non-Stationary	cT (envelope peak time)	$[0.2T, 0.8T]$

Table 1 Summary of bounds utilised to randomise the 600 distinct noise profiles. These bounds ensure spectral features lie within the dynamical decoupling filter passbands while providing sufficient variance.

3.2 Energy Normalisation

To ensure that the classification models learnt underlying geometric and spectral structures encoded in $\mathcal{F}$, we prevented the models from trivially classifying datasets based only on total signal energy, defined as

$$E_{\text{signal}} = \sum_{j \in \{x,y,z\}} \sum_{m=0}^{M-1} |\beta_j(t_m)|^2. \tag{10}$$

To do this, we first determined the energy targets (E_{target}) for each of the six noise classes via a binary search protocol. That is, we iteratively scaled the noise profiles $\beta_j(t_m)$ until the uncontrolled FID of the qubit reached $1/e$ at the final sequence time T, at which point we recorded the corresponding total signal energy as E_{target}. Physically, this establishes that the sequence window corresponds to T_2^* for all noise profiles, i.e., $T = T_2^*$.

Then, during data generation, we enforced these target energies across the entire ensemble by applying

$$\beta_j(t_m) \to \beta_j(t_m) \sqrt{\frac{E_{\text{target}}}{E_{\text{signal}}}}. \tag{11}$$

3.3 Control Pulse Implementation

Control pulses were applied solely along the x-axis ($f_y(t_m) = f_z(t_m) = 0$) for all simulations, taking the form of Gaussian-approximated CPMG sequences,

$$f_x(t_m) = \sum_{p=1}^{N_p} (A_p \pm \nu_p) e^{-\frac{(t_m - \tau_p)^2}{2\sigma^2}}, \tag{12}$$

where σ is the Gaussian pulse width, $p \in \{1, \ldots, N_p\}$ is the pulse index, and $\tau_p = \left(\frac{p-0.5}{N_p}\right) T \pm \delta_p$ sets the jittered pulse centre points. Here, $\delta_p \in \mathbb{R}$ explicitly models hardware jitter applied to the pulse timing. A_p dictates the amplitude of the p^{th} pulse (which is dynamically scaled inversely to the pulse width σ to maintain a constant pulse area of exactly π), and ν_p represents the dynamic amplitude jitter for the p^{th} pulse.

For our experiments, the sequence utilised $N_p = 10$ π-pulses (a CPMG-10 sequence). For each control pulse level, we generated K realisations by sampling the dynamic amplitude and timing jitter terms (ν_p and δ_p) according to the hardware parameters in Sec. 4.1. Fig. 2 illustrates level 1.0 and 2.0 pulse sequences with amplitude and timing jitter.

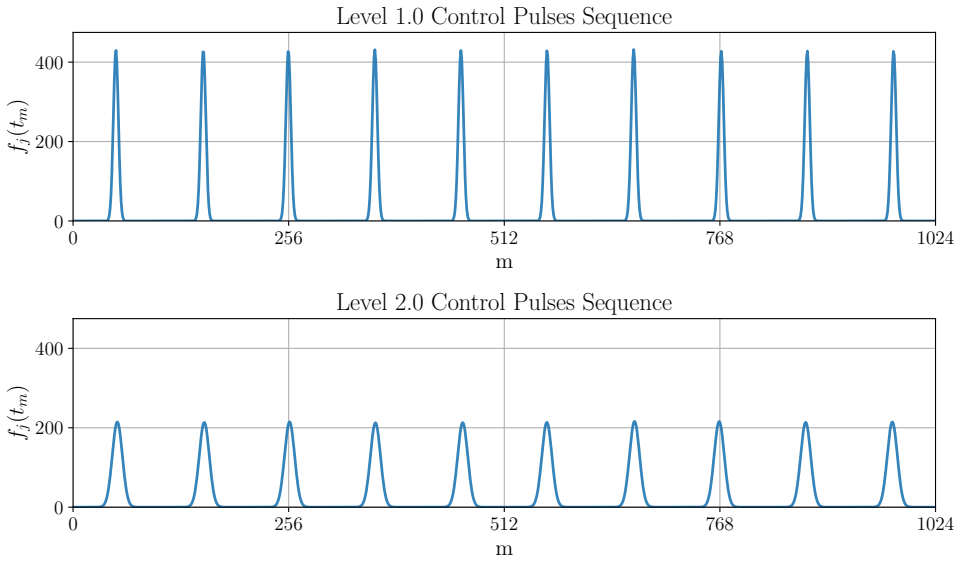


Fig. 2 CPMG-10 control pulse sequences generated for the simulations. (Top) Level 1.0 pulses with minor jitter. (Bottom) Level 2.0 pulses showcasing severe dynamic jitter in amplitude and timing across multiple realisations, representing highly degraded hardware imperfections. Note the reduction in peak amplitude for the wider level 2.0 pulses to preserve the π pulse area.

3.4 Expectation Calculation

We used Eq. (12) to generate K control pulse sequence realisations $f_j(t_m)$ of length M, which were then used to construct the control Hamiltonians, $H_{\text{ctrl}}(t_m)$, across all timesteps following Eq. (2). We then selected K noise realisations (of the 4096 total), each of length M, to construct the noise Hamiltonians, $H_{\text{noise}}(t_m)$, across all timesteps following Eq. (9). The total Hamiltonian $H(t_m)$ was then the sum of the control and noise Hamiltonians at each discrete timestep following Eq. (1).

With the total Hamiltonian $H(t_m)$ defined at each discrete timestep, we computed the total evolution operator $U(T)$ using the Lie–Trotter product formula:

$$U(T) = \mathcal{T} e^{-i\int_0^T H(s)\,ds} \approx \mathcal{T} \prod_{m=0}^{M-1} e^{-iH(t_m)\Delta t} = e^{-iH(t_{M-1})\Delta t} \cdots e^{-iH(t_0)\Delta t}. \tag{13}$$

Eq. (4) was then used to compute the expectation estimates, and by exploiting the batched structure of the tensors, we could simultaneously compute all nine expectation values for a given $U(T)$. These expectation values were calculated by averaging over these K classical noise and jitter realisations.

4 Classification Results

4.1 Hardware Assumptions and Error Modelling

We set the baseline (level 1.0) parameters to represent standard superconducting transmon devices (e.g., IBM Quantum architectures) [27, 29], with a baseline single π-pulse duration of 35 ns. While physical transmons typically exhibit T_2^* times of tens to hundreds of microseconds [27, 29], we set $T = T_2^* = 2$ μs. This choice is governed by numerical resolution: with $M = 1024$ timesteps, $T = 2$ μs provides a timestep of $\Delta t \approx 1.95$ ns, which is fine enough to resolve the narrowest 8.75 ns pulses (level 0.0) used in the experiments; in contrast, $T = 100$ μs would result in $\Delta t \approx 98$ ns, which is too coarse to resolve nanosecond pulse dynamics. Following the Nyquist–Shannon sampling theorem, these grid settings yield a Nyquist frequency of $f_{\max} = \frac{M}{2T} = 256$ MHz and a frequency resolution of $\Delta f = \frac{1}{T} = 0.5$ MHz.

Level	Pulse Parameters			SPAM (%)		Energy	Realisations (K)
	Width (σ)	Amp. Jitter (ν_p)	Time Jitter (δ_p)	ϵ_0	ϵ_1		
0.0	8.75 ns	0	0	0.0	0.0	0.25×	4096
0.5	17.5 ns	$\pm 0.0025\pi$	± 1 ns	1.0	1.5	0.50×	2048
1.0	35.0 ns	$\pm 0.0050\pi$	± 2 ns	2.0	3.0	1.00×	1024
1.5	52.5 ns	$\pm 0.0075\pi$	± 3 ns	4.0	6.0	1.50×	512
2.0	70.0 ns	$\pm 0.0100\pi$	± 4 ns	8.0	12.0	2.00×	256

Table 2 Summary of experimental degradation levels simulating realistic hardware imperfections. The pulse amplitude jitter explicitly varies the target π area, and the time jitter bounds are determined by the 200 ns pulse intervals. Level 1.0 represents standard NISQ baseline capabilities, while level 2.0 models severe calibration drift.

- **Pulse Error Sensitivity:** We introduced Gaussian jitter to both the target pulse amplitude ($A_p \pm \nu_p$) and centre timings ($\tau_p = \left(\frac{p-0.5}{N_p}\right) T \pm \delta_p$), whilst widening Gaussian envelopes representing lower maximum amplitude allowed by the equipment. These errors were escalated across the 5 specific levels outlined in Tab. 2.
- **Noise Energy Sensitivity:** We applied a global scaling to the noise realisations via scalar multipliers, as outlined in Tab. 2, to observe how varying the signal-to-noise ratio affects diagnostic performance.
- **Realisation Sensitivity:** We varied the finite number of noise and control sequence realisations utilised for ensemble averaging. We mapped realisation numbers K to the escalating levels in Tab. 2.
- **SPAM Errors:** We simulated SPAM errors via a transformation to the calculated expectations: $\tilde{\mathbb{E}}\{\sigma_k\}_{\rho_{j_+}} = \mathbb{E}\{\sigma_k\}_{\rho_{j_+}} \cdot (1 - \epsilon_0 - \epsilon_1) + (\epsilon_1 - \epsilon_0)$, where ϵ_0 and ϵ_1 denote the readout misclassification rates, with ϵ_0 representing the probability of incorrectly reading a $|0\rangle$ as $|1\rangle$, and ϵ_1 representing the probability of incorrectly reading a $|1\rangle$ as $|0\rangle$. The rates for ϵ_0 and ϵ_1 were scaled according to the levels detailed in Tab. 2, with error rates modelling those discussed in [29, 30].
- **Combined Errors:** We simultaneously tested all the above error sources by escalating them collectively according to the combined parameter sets defined in Tab. 2. This evaluated the classifier's robustness under compounded experimental imperfections to indicate expected real-world performance.

4.2 Robustness Tests

Using the generated $\mathcal{F}$ parameter datasets, we evaluated a Random Forest classifier, K-Nearest Neighbours (KNN), and Logistic Regression algorithms. For all tests, the models were trained and evaluated using 10-fold cross-validation. The results displayed in the accompanying figures and text denote the mean accuracy across these folds, alongside the standard deviation.

For all three algorithms, we utilised the standard implementations provided by the Scikit-Learn library [31]. The Random Forest classifier was constructed with default hyperparameters, comprising an ensemble of 100 decision trees governed by the Gini impurity split criterion. The K-Nearest Neighbours algorithm was similarly configured to its default of 5 nearest neighbours, evaluated using the standard Euclidean distance metric. The Logistic Regression model was configured with an L2 regularisation penalty ($C = 1.0$) and allowed a maximum of 1000 iterations to ensure optimal convergence.

Across all tests, the Random Forest consistently achieved high test accuracy (averaging $\sim$ 89–90%) for classifying the noise profile type, demonstrating resilience to severe experimental constraints and imperfections. Furthermore, the Random Forest identified noise stationarity with near 100% accuracy across all evaluations.

Logistic Regression underperformed compared to the Random Forest and KNN algorithms (e.g., dropping to $\sim$ 80–82% for noise type, and exhibiting much higher variance). This provides empirical evidence that the mapping from the underlying noise parameters into the 9-dimensional $\mathcal{F}$ is non-linear, creating a data structure that the Random Forest partitions but linear decision boundaries cannot.

4.2.1 Pulse Error Sensitivity

We first tested the sensitivity of the classifiers to pulse errors. For these experiments, we used the following parameters:

- Noise Energy: 1.0× (baseline)
- Realisations: $K = 4096$
- SPAM Errors: level 0.0

where pulse errors varied from level 0.0 (no errors) to level 2.0 (severe errors) as detailed in Tab. 2.

As shown in Fig. 3, varying the physical characteristics of the control pulses had minimal impact on the Random Forest and KNN classifiers. At the realistic standard IBM baseline (level 1.0), the Random Forest achieved a mean noise-type accuracy of $89.5\% \pm 2.4\%$. Even at the most severe level 2.0 degradation, classification accuracy was $90.2\% \pm 2.3\%$, reinforcing the robustness of the $\mathcal{F}$ representation against control pulse imperfections.

The Logistic Regression classifier for noise stationarity deteriorated at narrow pulse widths (particularly level 0.0), indicating possible sensitivity to narrow pulses. This trend continues across all other experiments involving narrow level 0.0 pulses.

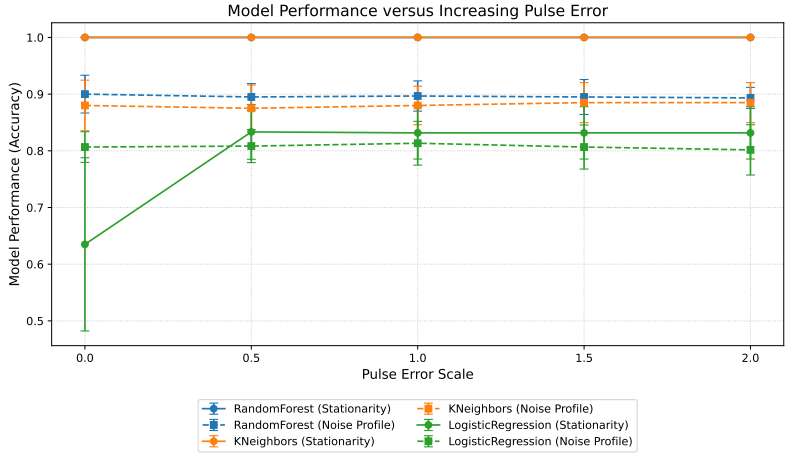


Fig. 3 Classifier performance under escalating levels of pulse amplitude jitter, timing errors, and Gaussian width expansion. The line displays the mean accuracy across the 10-fold cross-validation, and the shaded regions indicate the standard deviation.

4.2.2 Noise Energy Sensitivity

For the noise energy sensitivity tests, the following parameters were used:

- Pulse Errors: level 0.0
- Realisations: $K = 4096$
- SPAM Errors: level 0.0

where the noise energy multiplier varied across the range 0.25× to 2.0×.

Fig. 4 demonstrates the impact of underlying noise power, and because the noise is pre-normalised to T_2^*, varying the energy tests the classifier's reliance on geometric structure rather than absolute amplitude. At the baseline 1.0× multiplier, the Random Forest achieved 89.8% ± 3.2% noise type accuracy. The Random Forest's accuracy remained high across both low-energy (0.25×: 90.2% ± 3.0%) and high-energy (2.0×: 90.8% ± 2.9%) regimes, demonstrating that the signature of the spectral features is retained within $\mathcal{F}$, independent of the signal magnitude.

KNN was moderately sensitive to noise energy variations, and Logistic Regression showed stronger sensitivity, with accuracy increasing as noise energy increased. This suggests that these algorithms may rely on the absolute amplitude of the noise, which could manifest in points separating further apart in the feature space as the noise energy grows.

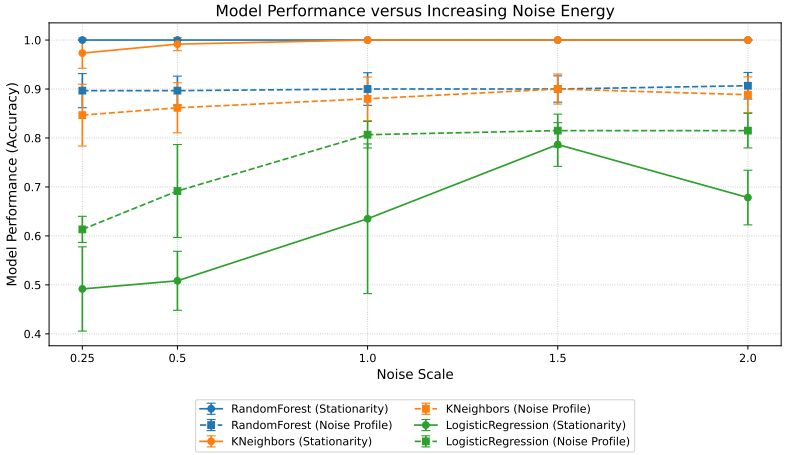


Fig. 4 Classifier performance across scaled environmental noise energy multipliers.

4.2.3 Finite-Ensemble Sensitivity

We then investigated the sensitivity of the classifiers to the number of stochastic noise realisations; to do so, we used the following parameters:

- Pulse Errors: level 0.0
- Noise Energy: 1.0× (baseline)
- SPAM Errors: level 0.0

with the number of realisations varied as $K \in \{256, 512, 1024, 2048, 4096\}$. As illustrated in Fig. 5, at $K = 1024$ realisations, the model achieved a mean noise-type accuracy of 89.2% ± 5.5%. Decreasing the number of ensemble realisations from an ideal $K = 4096$ down to $K = 256$ realisations did not significantly degrade the Random Forest's predictive capability, going from 90.0% ± 3.0% at $K = 4096$ to

$90.0\% \pm 4.0\%$ at $K = 256$, meaning $\mathcal{F}$ extraction could be performed efficiently with limited measurement resources.

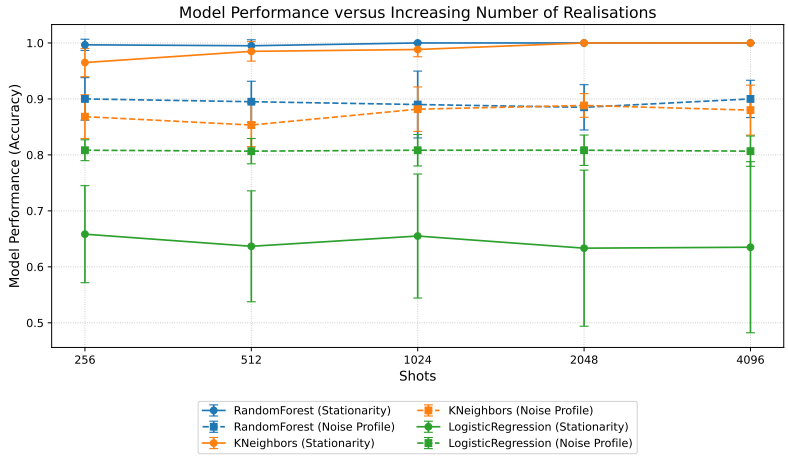


Fig. 5 Classifier accuracy plotted against the number of noise realisations (K) used for ensemble averaging to compute the observable expectations.

4.2.4 SPAM Errors

To evaluate the impact of SPAM errors, we considered the following setup:

- Pulse Errors: level 0.0
- Noise Energy: 1.0× (baseline)
- Realisations: $K = 4096$

with SPAM errors varied from level 0.0 to level 2.0. As shown in Fig. 6, all algorithms demonstrated resilience to SPAM errors (only the Logistic Regression algorithm diagnosing stationarity showed noticeable degradation). At the standard NISQ level 1.0 error rates ($\sim$ 2–3% bit-flips), the Random Forest maintained $90.0\% \pm 3.3\%$ accuracy when diagnosing the noise profile. Despite introducing large readout bit-flips (averaging 10% misclassification across the ground and excited states at level 2.0, with $\epsilon_0 = 8\%$ and $\epsilon_1 = 12\%$), the Random Forest was able to preserve high diagnostic fidelity ($90.0\% \pm 3.2\%$).

4.2.5 Combined Errors

Finally, to understand the likely performance of the QFS framework under realistic error conditions, we evaluated the models using the combined parameter sets defined in Tab. 2. In the preceding tests, we isolated individual error sources by holding all other parameters at their baseline values (e.g., $K = 4096$, 1.0× noise scaling, and zero pulse/SPAM errors for level 0.0). However, in this combined test, we simultaneously escalated all sources of degradation. This meant moving from the most pristine

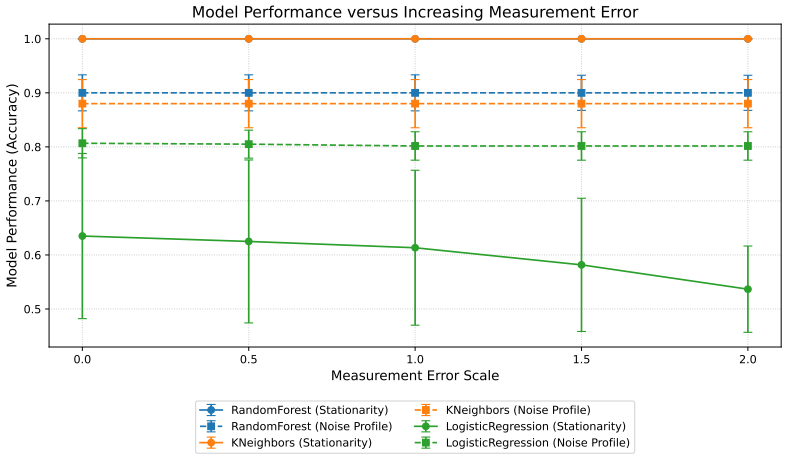


Fig. 6 Evaluation of the feature space under severe asymmetrical readout misclassification (SPAM) affine transformations.

quantum computer at level 0.0 (idealised $K = 4096$ realisations, no pulse errors, no SPAM errors, with a $0.25\times$ noise scaling) down to a severely degraded level 2.0 scenario (encompassing wide pulses, significant jitter, a reduced number of realisations $K = 256$, a $2.0\times$ noise scaling, and 10% SPAM bit-flips).

Fig. 7 demonstrates the practical utility of the QFS framework. Despite increasing error, the Random Forest continued to correctly diagnose the underlying noise family. At the realistic level 1.0, the model achieved $89.2\% \pm 4.6\%$ for noise profile diagnosis, while maintaining a perfect $100.0\% \pm 0.0\%$ stationarity diagnosis. Even under the most severe error conditions at level 2.0, the Random Forest still achieved a diagnostic accuracy of $88.2\% \pm 3.3\%$.

The KNN and Logistic Regression also demonstrate a level of resilience, albeit with lower overall accuracy compared to the Random Forest, even under the combined severe error conditions.

4.2.6 Summary of Robustness Tests

Across all isolated and combined hardware degradations, the 9-dimensional $\mathcal{F}$ combined with ML classifiers, particularly the Random Forest, demonstrated robustness and reliability in diagnosing the underlying noise characteristics despite severe NISQ hardware limitations. The Random Forest model consistently delivered $\sim 90\%$ classification accuracy for the underlying physical noise type and $\sim 100\%$ accuracy for stationarity. The fact that the Random Forest successfully parsed $\mathcal{F}$ under combined, severe NISQ conditions (level 2.0) demonstrates that a single-pass extraction of the expectation values contains sufficiently robust structural information for practical downstream diagnostic tasks.

Furthermore, we speculate that all algorithms performed better for stationarity classification because of the energy normalisation. That is, because a non-stationary

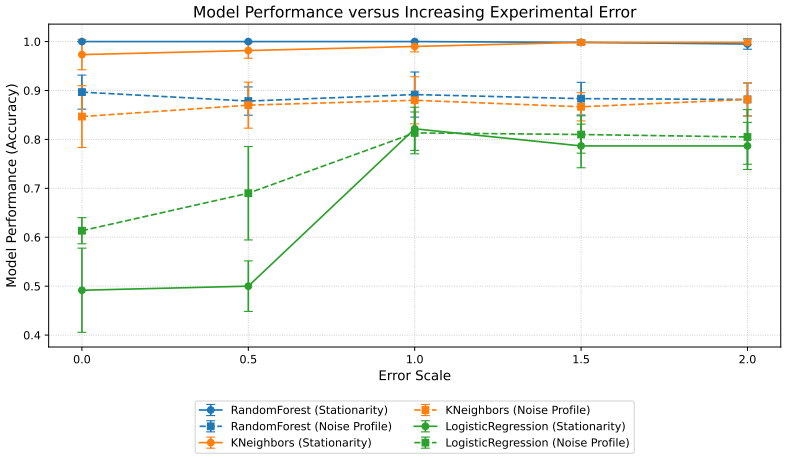


Fig. 7 Classifier performance when subjected to all experimental degradations and hardware constraints simultaneously.

signal is characterised by a Gaussian bump where it is "active", it inherently possesses regions of near-zero amplitude at the edges of the sequence window. To achieve the exact same total sequence energy as a continuously active stationary signal, the non-stationary signal must exhibit a significantly higher peak variance during its active window. This pronounced difference in variance likely translates directly into distinct geometrical divergence within $\mathcal{F}$. Consequently, separating stationary from non-stationary processes becomes an inherently easier classification task.

4.3 Black Box Classification of Noise Environment Through QFS

To demonstrate $\mathcal{F}$ as a practical and efficient diagnostic tool, we conducted a proof-of-concept black box classification experiment. We utilised the Random Forest trained explicitly on the level 1.0 Combined Error dataset, as this level provided the most realistic representation of standard NISQ hardware capabilities.

We started by modelling an "unknown" physical process as a $1/f$ noise spectrum with a resonance bump at a frequency bin of $\mu = 15.0$. The data point in $\mathcal{F}$ corresponding to this unknown environment was successfully diagnosed by the Random Forest as a stationary '$1/f$ + Bump' noise type.

Having identified the broad class of noise, we then aimed to identify the precise resonance frequency. Instead of executing a complex analytical spectral reconstruction, we generated a library of candidate '$1/f$ + Bump' noise profiles spanning various frequency bumps ($\mu \in [5.0, 25.0]$). To isolate the geometric behaviour of $\mathcal{F}$ to the spectral parameter μ, the candidates were simulated using common random numbers where appropriate (i.e., the same control pulse realisations and noise seeds). Consequently, at the true candidate $\mu = 15.0$, the distance is 0.0.

Candidate Freq. (μ)	**Distance in** $\mathcal{F}$	**Candidate Freq.** (μ)	**Distance in** $\mathcal{F}$
5.0	0.032155	16.0	0.000251
6.0	0.023452	17.0	0.000451
7.0	0.016232	18.0	0.000624
8.0	0.010713	19.0	0.000778
9.0	0.006785	20.0	0.000906
10.0	0.004152	21.0	0.001014
11.0	0.002467	22.0	0.001085
12.0	0.001411	23.0	0.001120
13.0	0.000748	24.0	0.001132
14.0	0.000312	25.0	0.001124
15.0	**0.000000**		

Table 3 Euclidean distance in the Quantum Feature Space between the unknown target ($\mu = 15.0$) and various candidate frequency bumps.

As recorded in Tab. 3, the Euclidean distance is smallest at the correct candidate $\mu = 15.0$ (distance ≈ 0.0), correctly identifying the spectral parameters of this unknown noise profile.

The distances exhibit a smooth, nearly monotonic relationship: as the candidate frequency moves away from the target $\mu = 15.0$ in either direction, the Euclidean distance in $\mathcal{F}$ increases across almost the entire candidate range. This monotonic gradient structure suggests that the QFS maps physical parameters (like frequency) smoothly into distance, making this framework highly suitable for optimisation algorithms (like gradient descent or binary search) to rapidly fit noise parameters without exhaustive sweeps.

Here we employ the Euclidean distance over $\mathcal{F}$ as a simple, efficient metric that, as established in Sec. 2.1, is proportional to the Hilbert–Schmidt distance (and identical to the Frobenius distance) between the effective channel matrices. Investigating whether non-linear operator metrics, such as trace distance, Bures distance, or process fidelity, could further improve discrimination between candidate noise profiles remains an interesting direction for future work.

5 Comparison to Previous Work

The primary innovation of the QFS method is its substantial reduction in both model complexity and data requirements, directly addressing the key scaling limitations of deep learning strategies [10, 13, 17]. As shown in Tab. 4, deep learning models employ neural networks with high parameter counts (often $> 10^4$ to 10^5) that require thousands of data points for training. By contrast, the Random Forest used in our experiments has 8,192 parameters and used only 600 training examples.

Previous black-box and grey-box methods utilised neural networks to directly output full continuous PSD reconstructions or noise operator estimates [10, 17]. Because full continuous spectral mapping is an inherently difficult, high-dimensional task, these models are computationally expensive and require large training datasets. Instead, leveraging $\mathcal{F}$—which intrinsically encodes numerous dynamical properties of the qubit-environment interaction—allows us to break the characterisation process into two

simplified phases. First, by applying simple ML algorithms to $\mathcal{F}$, we solve the easier problem of noise classification. Once the overarching noise class is known, we can use simulated data to test various parameters within that specific class in parallel, finding the parameter set that best matches the experimental $\mathcal{F}$ data. This two-phase approach allows for an accurate, targeted diagnosis of the specific noise spectra while reducing the computational and experimental cost.

This work has not exhaustively stress-tested this parameter-fitting approach for full continuous PSD curve reconstruction across an arbitrary variety of complex spectra. However, the results here show that this efficient protocol of classification followed by targeted identification is a promising approach for downstream quantum optimal control, which is orders of magnitude faster than direct neural network approaches.

Attribute	**FTNS [14]**	**NN Regression [10, 13]**	**Grey-box ML [17]**	**QFS (This Work)**
Principle	Analytical inversion	Deep NN	Deep NN + analytical layers	Euclidean distance / Random Forest
Exp. Overhead	FID curve (time sweep)	Decay curves (τ sweeps)	Single sequence	Single fixed sequence
Comput. Cost	Analytical (very low)	NN training (high)	NN training (high)	Distance calculation / Tree training (low)
Fidelity/Accuracy	$\sim 99\%$	~ 95–98%	$> 99\%$	$\sim 90\%$ / Exact*
Model Parameters	N/A	$\sim 10^4$–10^5	$\gtrsim 10^5$	$\sim 8 \times 10^3$

Table 4 Comparison of qubit noise spectrum estimation methods. While previous methods focus on continuous PSD reconstruction, the QFS approach is highlighted for its ability to classify noise types and identify spectral features with low model complexity, making it highly suitable for rapid downstream diagnostics. *$\sim 90\%$ broad classification accuracy under level 1.0 combined errors; resonance peak μ identified exactly on the candidate grid.

6 Scaling

To scale the feature space to a two-qubit gate protocol, the qubit state and observables are represented by 4×4 matrices that span a 16-dimensional operator space (the generalised Pauli basis). Following the same logic as the single-qubit derivation, removing the uninformative identity terms requires $15 \times 15 = 225$ independent expectation measurements. While each configuration requires only a single-pass pulse sequence (not the exhaustive frequency sweeps of traditional spectroscopy), one would then need to classify the behaviour of all physical two-qubit gates across the hardware.

Because physical quantum computers decompose all operations into fundamental single- and two-qubit physical gates, the noise characterisation problem inherently simplifies into local components, avoiding the need to consider the full exponential scaling of the global Hilbert space.

For single-qubit gates, experimental data could be gathered for all physical qubits simultaneously, with the corresponding classical diagnosis computed in parallel using GPU acceleration.

However, diagnosing every physical pair of qubits introduces an experimental overhead as the total number of connected pairs grows rapidly, particularly in globally

connected hardware topologies. This challenge can be mitigated with data collection parallelised across spatially disjoint qubit pairs.

In graph-theoretic terms, characterising the noise of a fully connected N-qubit processor is equivalent to finding the 1-factorisation (or minimum edge colouring) of a complete graph K_N [32]. For an even number of qubits, the $N(N-1)/2$ total two-qubit connections can be partitioned into exactly $N-1$ disjoint matchings. This guarantees that even in the worst-case scenario of a globally connected macroscopic processor, the required number of sequential measurement rounds scales strictly linearly.

For example, fully characterising a 10-qubit processor (45 total pairs) requires 9 parallelised rounds, and a 100-qubit processor (4,950 total pairs) requires 99 rounds. Therefore, fully mapping two-qubit gate noise across a macroscopic NISQ processor avoids exponential bottlenecks, requiring at most N parallelised rounds of experimental data collection (and exactly $N-1$ when N is even).

7 Conclusion

In this work, we introduced a quantum feature space ($\mathcal{F}$) constructed from a single pulse sequence, designed to intrinsically encode the qubit-environment interaction dynamics of a qubit subjected to a classical stochastic environment. By shifting the objective from computationally expensive, continuous full-spectrum reconstruction to targeted discrete classification, we demonstrated that the $\mathcal{F}$ framework eliminates the need for both exhaustive multi-pulse frequency sweeps on quantum hardware and expensive neural networks for classical processing.

Instead, we showed that applying simple, efficient ML algorithms, such as a Random Forest containing fewer than 10^4 parameters, to $\mathcal{F}$ provides a robust and scalable diagnostic tool. Under simulated realistic hardware constraints incorporating pulse jitter and readout errors, our method successfully classified broad noise processes with $\sim 90\%$ accuracy. Furthermore, using simple Euclidean distance metrics over $\mathcal{F}$, we correctly identified spectral resonances. This protocol provides a rapid, computationally cheap diagnostic tool for real-time noise identification on macroscopic NISQ hardware.

Looking forward, this paper opens several promising avenues for future work. First, extending the framework to study non-Gaussian environments, such as random telegraph noise, will address strongly coupled two-level fluctuators common in solid-state devices.

Additionally, extending the framework to capture non-unital dissipative dynamics—where the Bloch sphere undergoes an affine shift and longitudinal relaxation towards a ground or thermal state rather than purely unital dephasing—is an important open direction. This could be achieved by coupling the qubit to a truncated quantum environment, which naturally produces non-unital relaxation through entanglement and energy exchange with the environment. Alternatively, for simpler diagnostic goals, the classical stochastic model could be phenomenologically augmented with an explicit amplitude damping channel to model T_1 relaxation. Any of these extensions would broaden the diagnostic scope of $\mathcal{F}$ to encompass the full range of decoherence mechanisms encountered on real hardware.

Future work could examine the performance of ML classifiers as the number of timesteps M changes, for example $M \in \{256, 512, 1024, 2048, 4096\}$. For a fixed sequence duration T, this evaluates classifier robustness across different noise spectral bandwidths and control pulse discretisation fidelities.

There is also significant potential for direct parameter estimation. By utilising PyTorch's native autograd capabilities, researchers could reparameterise the problem and backpropagate directly through the differentiable $\mathcal{F}$ mapping to estimate experimental noise parameters without relying on heuristic searches. This could involve parameterising the PSD as a weighted sum of basis functions, with the weights treated as optimisable parameters. This approach fuses noise classification and identification, potentially avoiding the need for any ML algorithms.

Applying the framework directly to physical NISQ hardware will be a vital next step. Executing these sequences on physical quantum processors will benchmark real-world fidelities and validate the resilience of the QFS against unmodelled environmental crosstalk.

To further enhance the practical utility of the QFS, future work could also study the robustness of the framework when extending to two qubits, where correlated noise and entanglement effects become significant.

Finally, utilising the established QFS as a lightweight digital twin presents an opportunity for advanced control. By mapping hardware responses to $\mathcal{F}$, we could backpropagate and optimise quantum control pulse sequences natively on classical GPUs. This would entirely bypass executing online hardware optimisations, enabling rapid, closed-loop pulse generation adapted to drifting environmental noise.

8 Data and Code Availability

The data and code are available at https://github.com/ChrisWise07/quantum_feature_space.

9 Funding

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

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] Carr, H.Y., Purcell, E.M.: Effects of diffusion on free precession in nuclear magnetic resonance experiments. Physical Review **94**(3), 630 (1954)

[4] Meiboom, S., Gill, D.: Modified spin-echo method for measuring nuclear relaxation times. Review of Scientific Instruments **29**(8), 688–691 (1958)

[5] Degen, C.L., Reinhard, F., Cappellaro, P.: Quantum sensing. Reviews of Modern Physics **89**(3), 035002 (2017)

[6] 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)

[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] Szańkowski, P., *et al.*: Environmental noise spectroscopy with qubits subjected to dynamical decoupling. Journal of Physics: Condensed Matter **29**(33), 333001 (2017)

[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] Vezvaee, A., Shitara, N., Sun, S., Montoya-Castillo, A.: Fourier transform noise spectroscopy. npj Quantum Information **10**(1), 52 (2024)

[15] 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)

[16] Nakamura, K., Ankerhold, J.: Gate operations for superconducting qubits and

non-markovianity. Physical Review Research **6**(3), 033215 (2024)

[17] 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)

[18] 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)

[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] Winczewski, M., Mandarino, A., Suarez, G., Alicki, R., Horodecki, M.: Intermediate-times dilemma for open quantum system: Filtered approximation to the refined weak-coupling limit. Physical Review E **110**(2), 024110 (2024)

[21] Ball, H., Biercuk, M.J., Carvalho, A., Chen, J., Hush, M., De Castro, L.A., Li, L., Liebermann, P.J., Sastrawan, H., Frey, V., *et al.*: Software tools for quantum control: Improving quantum computer performance through noise and error suppression. Quantum Science and Technology **6**(4), 044011 (2021)

[22] Cywiński, Ł., Lutchyn, R.M., Nave, C.P., Das Sarma, S.: How to enhance dephasing time in superconducting qubits. Physical Review B—Condensed Matter and Materials Physics **77**(17), 174509 (2008)

[23] 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)

[24] Perrier, E., Youssry, A., Ferrie, C.: Qdataset, quantum datasets for machine learning. Scientific Data **9**(1), 582 (2022)

[25] Paladino, E., Galperin, Y., Falci, G., Altshuler, B.: 1/f noise: Implications for solid-state quantum information. Reviews of Modern Physics **86**(2), 361 (2014)

[26] Müller, C., Cole, J.H., Lisenfeld, J.: Towards understanding two-level-systems in amorphous solids: insights from quantum circuits. Reports on Progress in Physics **82**(12), 124501 (2019)

[27] Krantz, P., Kjaergaard, M., Yan, F., Orlando, T.P., Gustavsson, S., Oliver, W.D.: A quantum engineer's guide to superconducting qubits. Applied Physics Reviews **6**(2), 021318 (2019)

[28] Timmer, J., König, M.: On generating power law noise. Astronomy and Astrophysics **300**, 707 (1995)

[29] Jurcevic, P., Dial, O.E., Cross, A.W., Bronn, N.T., Chow, J.M., *et al.*: Demonstration of quantum volume 64 on a superconducting quantum computing system. Quantum Science and Technology **6**(2), 025020 (2021)

[30] Maciejewski, F.B., Zimborás, Z., Oszmaniec, M.: Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography. PRX Quantum **1**(1), 010304 (2020)

[31] 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)

[32] West, D.B.: Introduction to Graph Theory, 2nd edn. Prentice Hall, Upper Saddle River, NJ (2001)