Machine Learning: Science and Technology

LETTER • **OPEN ACCESS**

Deep learning enhanced noise spectroscopy of a spin qubit environment

IOP Publishing Mach. Learn.: Sci. Technol. 4 (2023) 02LT01 https://doi.org/10.1088/2632-2153/acd2a6

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
1 February 2023

REVISED
25 April 2023

ACCEPTED FOR PUBLICATION
4 May 2023

PUBLISHED
17 May 2023

LETTER

Deep learning enhanced noise spectroscopy of a spin qubit environment

Stefano Martina[1,2,6], **Santiago Hernández-Gómez**[2,3,6], **Stefano Gherardini**[2,4], **Filippo Caruso**[1,2,5,*] and **Nicole Fabbri**[2,5]

[1] Dipartimento di Fisica e Astronomia, Università di Firenze, I-50019 Sesto Fiorentino, Italy
[2] European Laboratory for Non-Linear Spectroscopy (LENS), Università di Firenze, I-50019 Sesto Fiorentino, Italy
[3] Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America
[4] Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO), Area Science Park, Basovizza, I-34149 Trieste, Italy
[5] Istituto Nazionale di Ottica del Consiglio Nazionale delle Ricerche (CNR-INO), I-50019 Sesto Fiorentino, Italy
[6] Equally contributed to this work.
[*] Author to whom any correspondence should be addressed.

E-mail: filippo.caruso@unifi.it

Keywords: deep learning, neural networks, machine learning, quantum machine learning, quantum noise, quantum sensing, quantum noise spectroscopy.

Abstract

The undesired interaction of a quantum system with its environment generally leads to a coherence decay of superposition states in time. A precise knowledge of the spectral content of the noise induced by the environment is crucial to protect qubit coherence and optimize its employment in quantum device applications. We experimentally show that the use of neural networks (NNs) can highly increase the accuracy of noise spectroscopy, by reconstructing the power spectral density that characterizes an ensemble of carbon impurities around a nitrogen-vacancy (NV) center in diamond. NNs are trained over spin coherence functions of the NV center subjected to different Carr–Purcell sequences, typically used for dynamical decoupling (DD). As a result, we determine that deep learning models can be more accurate than standard DD noise-spectroscopy techniques, by requiring at the same time a much smaller number of DD sequences.

1. Introduction

Quantum sensing combines theoretical results with experimental and engineering techniques to carry out inference of signals with improved accuracy and/or less computation time by making use of quantum physics [1, 2].

A quantum sensor takes advantage of the fragility of its quantum properties, such as quantum coherence or entanglement, to improve the detection of external perturbations with higher accuracy compared to any classic sensor. However, this same property implies that the quantum sensor is subjected to detrimental noise stemming from the coupling with its environment. For this reason, it is desirable to fully characterize the sensor's environment, either to filter out its detrimental effect, or to take it into account when detecting external signals, for example, in algorithms using quantum optimal control [3–7].

Neural networks (NNs) [8, 9], i.e. algorithmic models provided by the interconnection of a group of nodes commonly called neurons, could be a powerful tool to infer the sensor's environment. In this context, deep learning has been already proposed theoretically for the classification and detection of quantum noise features [10–12], and employed experimentally for the following tasks. *(a)* Estimating the spectra of minuscule amounts of complex molecules [13] for nano nuclear magnetic resonance; *(b)* the sensing of magnetic-field strength at room temperature with high precision [14, 15] by using nitrogen vacancy (NV) centers; *(c)* performing error mitigation [16] and noise learning [17–19]; *(d)* the tracking of quantum trajectories [20]; *(e)* classification of many-body quantum states [21] in superconducting quantum circuits; *(f)* improving quantum error correction [22]. Also quantum NNs are recently investigated in order to solve a given quantum technology task with a greater accuracy than classical NNs [23–25]. However, to our

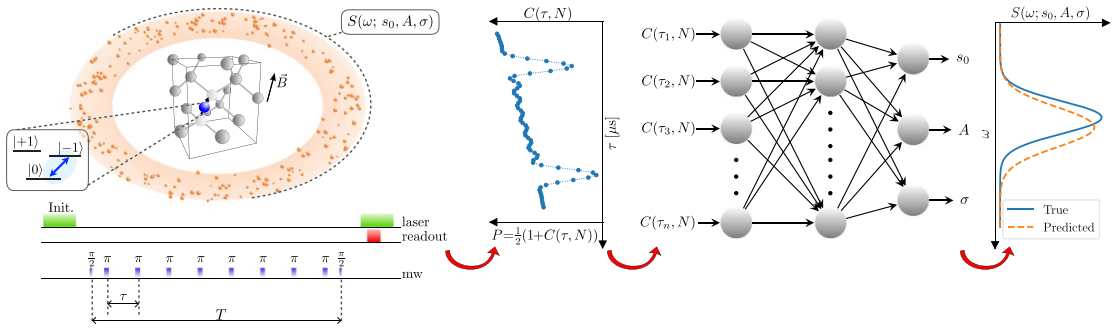

Figure 1. NV center and neural networks for noise spectroscopy. The NV center is surrounded by an ensemble of ^{13}C nuclear spins (orange spheres) that collectively induce dephasing to the NV electronic spin (blue sphere). The NV electronic spin is controlled with a DD sequence (specifically, a Carr–Purcell (CP) sequence) with the aim to measure its dephasing, and therefore characterize the NSD of the nuclear spin bath, i.e. $S(\omega; s_0, A, \sigma)$. The CP sequence is formed by N equidistant π pulses in between an initial and a final $\pi/2$ pulse. The time τ between the π pulses determines the measurement total time $T = N\tau$, given that the time between the first $\pi/2$ and the train of π pulse and the time between the last π and $\pi/2$ pulses are both equal to $\tau/2$. Then, we measure the output of this experiment, which is the probability $P = \frac{1}{2}(1 + C(\tau, N))$ that the NV center remains in the initial state $|0\rangle$. The spin coherence function $C(\tau, N)$—evaluated at previously-determined inter-pulse times in the set $\tau \in \{\tau_1, \tau_2, \ldots, \tau_n\}$ is the input of the designed neural networks (NNs). For illustrative purposes, here we only consider one fixed value of N. In our study we also consider a set of different values of N (section 2.3). After being trained, the NNs return the estimation of the NSD parameters.

knowledge, experimental noise spectroscopy in single color centers in diamond via deep learning is still missing.

In this paper, we demonstrate that NNs can be used to process the data obtained by a qubit, operating as a quantum sensor, and then reconstruct the noise spectrum that induces dephasing into the qubit itself. In particular, we focus on a qubit under dynamical decoupling (DD) control sequences [26, 27] in the presence of classical random noise with an unknown power density spectrum, usually denoted as *noise spectral density* (NSD). Beyond testing numerically our machine learning models, we use a single NV center in diamond as a spin qubit sensor and we perform a spectroscopic reconstruction of the magnetic noise of its local environment. The latter comprises ^{13}C nuclear spins randomly distributed in the diamond lattice [28–30] (see figure 1). The dephasing affecting the qubit sensor is analyzed by applying a set of DD control pulses that realize filter functions [26, 27, 31, 32] in the frequency domain. The filter functions are designed to select specific noise components, without sensing all other system-bath interactions. A widely used DD control pulse is the Carr–Purcell (CP) sequence [1, 33] that is given by N equidistant π pulses, performed between an initial and a final $\pi/2$ pulse. CP sequences act in the frequency domain approximately as Dirac comb filters [34]; hence, they have been used to perform spectroscopy of intricate signals, e.g. for noise spectroscopy [35, 36]. With this protocol, the requirement to achieve high values of the noise reconstruction accuracy is to perform sequences with a high number of pulses meaning $N \in [30, 120]$ (as in [37]) or higher, so that the Dirac comb filter approximation remains valid (in fact, N determines the filter width). This usually leads to long experiments to reconstruct the whole spectrum of the noise. Other techniques using non-equidistant or even more sophisticated DD sequences [4, 38–41] have proved to be effective for noise sensing, but sometimes at the price of a higher computational burden.

For our sensing task, NNs are designed to solve a regression problem, i.e. the reconstruction of the NSD. Here, we assume that the NSD of the bath of spins has a Gaussian profile [37, 42, 43]. The Gaussian NSD is thus parametrized as a function of key parameters, i.e. the mean value, variance, offset and noise power that we aim to reconstruct. Note that our proposal can be adapted to other parametrized NSD functions. The NNs are trained over a set of synthetic data generated by simulating how the coherence of the qubit sensor decays over time under the influence of both the CP control pulses and the NSD. Moreover, to make the measurement statistics as close as possible to the ones obtained from the experiments, extra artificial errors sampled from a normal distribution are added.

Our approach using NNs entails the following advantages that we have proven experimentally. *(i)* NNs have the capability to predict never-before-seen experimental data, and they can work with a better reconstruction accuracy (even up to seven times better, as shown in the section 2 below) than standard noise spectroscopy, as the Álvarez–Suter method [36], by making use at the same time of DD control sequence with a much smaller number of pulses. *(ii)* The training dataset, which can contain both synthetic and experimental data, is generated just once and then it can be applied several times, as long as the new collected data reproduce the physical context under analysis. In connection with *(i)*, we are going to show that the

amount of data used as input to the NNs can be smaller than the one needed to resolve the NSD by means of standard noise spectroscopy methods.

From our knowledge, this work is the first experimental proof of enhanced reconstruction performance with NNs for carrying out noise spectroscopy in single color centers in diamond. We thus expect that the techniques discussed here could fast become a novel standard spectroscopy tool both for such quantum systems and other quantum platforms in which regression problems have to be solved.

2. Results

2.1. Generation of training dataset

The training dataset is composed of synthetic data that are originated by simulating the coherence decay of the qubit sensor in a noise spectroscopy experiment based on DD, as the one depicted in figure 1. This standard sensing procedure, which stems from Ramsey interferometry [1], maps information about the quantum coherence of the sensor into the population in $|0\rangle$ that is then effectively recorded. After having initialized the qubit sensor in the ground state $|0\rangle$, a $\pi/2$ pulse is applied such that the qubit state $|\psi\rangle$ is the superposition $(|0\rangle + |1\rangle)/\sqrt{2}$. Then, we perform a CP control sequence consisting in a train of π pulses that flips repeatedly the qubit, and finally, a second $\pi/2$ pulse is applied in order to map the phase of the qubit into its population. The probability that the state of the quantum sensor is $|0\rangle$, which corresponds to the observable population, equals to [1, 37]

$$P = \frac{1}{2}\left(1 + C(\tau, N)\right), \tag{1}$$

where N is the number of π pulses and τ is the time between them. The coherence function $C(\tau, N)$ is simulated numerically, for a set of different values of τ and N, to generate the training dataset.

Let us now introduce the decoherence function that quantifies how the quantum coherence $C(\tau, N)$ is modified under the action of both the external bath of spins and a set of CP control pulses. The control sequence has the effect to modulate the coherence content of the qubit sensor, while the interaction with the bath, associated to the NSD $S(\omega)$, tends on average to destroy such coherence. Overall, under the joint presence of control fields and a noise source, the coherence decays as $C(\tau, N) \equiv e^{-\chi(\tau, N)}$, where $\chi(\tau, N)$ denotes the *decoherence function* [32, 44–46]:

$$\chi(\tau, N) = \int \frac{\mathrm{d}\omega}{\pi\omega^2} F(\omega, \tau, N) S(\omega)\,. \tag{2}$$

In equation (2), the *filter function* $F(\omega, \tau, N) \equiv |Y(\omega, \tau, N)|^2$ is the square modulus of the Fourier transform of the so-called modulation function $y(t, \tau, N)$. The latter is constant piecewise, with values ± 1, and switches sign at the times $t = \tau/2, 3\tau/2, \ldots, (N - 1/2)\tau$ where each π pulse is applied [2]. Notice that we are assuming that the π pulses are instantaneous, a reasonable assumption for our experimental setup where a π pulse duration is $\sim$0.1 μs and the time between pulses is $\tau \in [3.3, 6.1]$ μs. Let us now recall the expression, in the frequency domain, of the filter function for a CP sequence with even N:

$$F(\omega, \tau, N) = 8 \sin^2\left(\frac{\omega\tau N}{2}\right) \sec^2\left(\frac{\omega\tau}{2}\right) \sin^4\left(\frac{\omega\tau}{4}\right), \tag{3}$$

while for odd N, $\sin^2(\omega\tau N/2)$ has to be replaced with $\cos^2(\omega\tau N/2)$ [2, 31].

In order to generate the training dataset, the NSD $S(\omega)$ is parameterized as

$$S(\omega) = s_0 + A \exp\left(-\frac{(\omega - \omega_c)^2}{2\sigma^2}\right). \tag{4}$$

Thus, being a Gaussian distribution, the NSD is fully described by the *offset* s_0, *amplitude* A, *width* σ and *center* ω_c. For the training dataset in the paper, the values of these parameters are taken from the following intervals: $s_0 \in [4 \times 10^{-4}, 4 \times 10^{-3}]$ MHz; $A \in [0.3, 0.7]$ MHz; $\sigma \in [2 \times 10^{-3}, 9 \times 10^{-3}]$ MHz. Instead, ω_c is kept constant. This is because in our experimental setup the NSD stems from the interaction with a large ensemble of unresolved ^{13}C impurities (nuclear spin bath) around the NV electronic spin. Therefore, the center of the NSD corresponds to the Larmor frequency $\omega_c = \gamma B$, where $\gamma = 1.0705$ kHz G^{-1} is the gyromagnetic ratio of the ^{13}C nuclear spins, and B is the amplitude of a static magnetic field aligned with the NV quantization axis, z. Such static magnetic field is well known during the experimental procedure since it determines the NV electronic spin resonances ($B = 403.2 \pm 2$ G).

The training dataset is generated by uniformly sampling 10^4 sets of parameters within the chosen intervals. Hence, overall we consider 10^4 distinct sequences of NSD parameters that are used to simulate

different coherence curves $C(\tau,N)$. These sequences are taken in the time intervals $\tau \in [3.3, 3.66]$ μs and $[5.5, 6.1]$ μs with sampling time $\Delta\tau = 1$ ns ($\Delta\tau = 20$ ns in the experimental case, see below), and for $N = \{1, 8, 16, 24, 32, 40, 48\}$. These intervals are significant for our study because they include the values of τ at which the coherence decay curve exhibits the first and second order collapses induced on the qubit sensor by the bath of ^{13}C impurities (for the coherence curves, the first and second order of the collapses refer to the harmonics of the filter functions $F(\omega,\tau,N)$, for more details see [37]). Finally, in order to make the synthetic data used to train the NNs closer to the experimental setting, extra artificial errors sampled from a normal distribution with zero expected value and standard deviation equal to 0.05 (comparable with the expected error in our experimental measurements) are added to every point of the generated coherence decay curves. In this way, one may mitigate the over-fitting of the employed machine learning models that are thus expected to better generalize to unseen data. In general, a model trained on synthetic data cannot be successfully applied to real data without fine tuning it. But in our case, it becomes possible, probably due to the fact that the simulated data of the coherence decay are quite close to the experimentally observed decay data induced by the environment.

As final remark, notice that, from the 10^4 simulated curves $C(\tau,N)$, 6000 are used for the training of the NNs and 2000 for their validation. Instead, the test step is performed either by using the remaining 2000 simulated curves, or by using experimental data as described below.

2.2. NNs working principles

Let us describe the main working features of the NNs employed in this paper to carry out noise spectroscopy. Specifically, we are going to use the multi-layer perceptron (MLP) that is composed of fully-connected layers, each of them with a variable number of artificial neurons.

A single artificial neuron returns as output the scalar

$$\hat{y} \equiv \Sigma(\mathbf{w}^T \cdot \mathbf{x} + b) \tag{5}$$

that, by definition, is provided by applying the non-linear function $\Sigma : \mathbb{R} \to \mathbb{R}$ to the weighted sum of the input vector $\mathbf{x} \in \mathbb{R}^k$ to which the bias term $b \in \mathbb{R}$ is added. $\mathbf{w} \in \mathbb{R}^k$ denotes the vector of weights. In our analysis, the activation function Σ is chosen equal to the *rectifier* $\Sigma(x) \equiv \max(0,x)$ [47, 48]. Thus, an MLP layer composed of q neurons (each with k inputs) returns the vector

$$\hat{\mathbf{y}} \equiv \Sigma(W^T\mathbf{x} + \mathbf{b}), \tag{6}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^q$, $W \in \mathbb{R}^{k\times q}$ is the matrix of weights (W collects all the weight vectors of the single neurons), and $\mathbf{b} \in \mathbb{R}^q$ is the vector of the biases. Hence, an MLP with L layers is ruled by the recursion equation

$$\mathbf{h}[\ell] \equiv \Sigma\left(W[\ell]^T\mathbf{h}[\ell-1] + \mathbf{b}[\ell]\right), \tag{7}$$

where $\ell = 1,\ldots,L$ is the index over the number of layers and $\mathbf{h}[0] \equiv \mathbf{x}$. In equation (7), $W[\ell]$ and $\mathbf{b}[\ell]$ are, respectively, the weights and the biases of the ℓth layer. The output vector of the MLP is $\hat{\mathbf{y}} \equiv \mathbf{h}[L]$. It is worth noting that the number, dimension and activation functions (they are usually denoted as the *hyperparameters* ξ) of the NN layers are chosen through a single optimization routine (cf section 5).

Let us now introduce the supervised learning process. Ideally, the purpose of the latter is to find the parameters $\theta^* = \mathrm{argmin}_\theta \mathcal{R}_\mathcal{D}(\theta,\xi)$ that minimize the *theoretical risk function*

$$\mathcal{R}_\mathcal{D}(\theta,\xi) \equiv \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}\left[\mathcal{L}\left(\hat{\mathbf{y}},\mathbf{y}\right)\right], \tag{8}$$

where $\theta \equiv \{W[1],\mathbf{b}[1],\ldots,W[L],\mathbf{b}[L]\}$, and $\hat{\mathbf{y}}$ are the estimated values of $\mathbf{y}$. By definition, $\mathcal{R}_\mathcal{D}$ is the expected value of the *loss function* $\mathcal{L}$ for $(\mathbf{x},\mathbf{y})$ sampled from the distribution $\mathcal{D}$ that generates the dataset [49]. The loss function $\mathcal{L}$ is a differentiable function that measures the distance between the prediction $\hat{\mathbf{y}}$ (output of the MLP) and the desired output $\mathbf{y}$. However, since one can only dispose of a finite set $\mathcal{S} = \{(\mathbf{x},\mathbf{y})_1,\ldots,(\mathbf{x},\mathbf{y})_m\}$ of samples to train, validate and test the employed ML models, the theoretical risk function is approximated by the *empirical risk function*. Considering the partition $\{\mathcal{S}_{\rm tr},\mathcal{S}_{\rm va},\mathcal{S}_{\rm te}\}$ of $\mathcal{S}$ in *training* ($\mathcal{S}_{\rm tr}$), *validation* ($\mathcal{S}_{\rm va}$) and *test* ($\mathcal{S}_{\rm te}$) sets, the empirical risk function is defined by:

$$\mathcal{R}_{\mathcal{S}_{\rm tr}}(\theta,\xi) \equiv \frac{1}{|\mathcal{S}_{\rm tr}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{S}_{\rm tr}}\mathcal{L}\left(\hat{\mathbf{y}},\mathbf{y}\right), \tag{9}$$

where $|\mathcal{S}_{\rm tr}|$ is the cardinality of the training set. In fact, $\mathcal{R}_{\mathcal{S}_{\rm tr}}$ is the arithmetic mean of the loss function $\mathcal{L}$ evaluated on the samples of the training set $\mathcal{S}_{\rm tr}$.

In our paper, we take the loss function $\mathcal{L}$ equal to the *mean squared error* (MSE), also called *L2* loss:

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{q}\sum_{i=1}^{q}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2 \tag{10}$$

for the q outputs of the last layer (in our case three, corresponding to the noise parameters s_0, A, σ). The MLP is trained by minimizing (step-by-step over time) the empirical risk function $\mathcal{R}_{\mathcal{S}_{\rm tr}}(\theta, \xi)$ with respect to θ by means of the *mini-batch gradient descent* method, so as to obtain the optimal value θ^* of the NN parameters. Each gradient descent step is defined by

$$\theta_{t+1} = \theta_t - \eta\nabla_\theta \frac{1}{B}\sum_{b=1}^{B}\mathcal{L}(\hat{\mathbf{y}}_{b,t}, \mathbf{y}_{b,t}), \tag{11}$$

where θ_0 is a randomly chosen starting point, η is the *learning rate* that defines the length of the step and $\nabla_\theta \frac{1}{B}\sum_{b=1}^{B}\mathcal{L}(\hat{\mathbf{y}}_{t,b}, \mathbf{y}_{t,b})$ is the gradient of the loss function. The gradient is calculated for any time t on a batch of B elements taken from the training set, and the subscript θ in ∇_θ indicates that the variables of $\mathcal{L}$ during the gradient evaluation are the weights of the NN. In this paper, $\mathcal{R}_{\mathcal{S}_{\rm tr}}$ is minimized by means of Adam [50] that is a gradient-based optimization algorithm performing the adaptive estimation of lower-order moments. The minimization is stopped when the time-derivative of the risk function evaluated on the validation set $\mathcal{R}_{\mathcal{S}_{\rm va}}(\theta^*, \xi)$ becomes positive (*early stopping* strategy) or after a predefined number of gradient steps using all the data of the training set (called *epochs*). Then, we use $\mathcal{R}_{\mathcal{S}_{\rm va}}(\theta^*, \xi)$ to check if the MLP works also for unseen data and tune the hyperparameters ξ (cf section 5). Finally, the test set $\mathcal{S}_{\rm te}$ is employed to calculate the metrics (discussed in detail below) used to generate the figures with the results that we are going to illustrate.

2.3. Training and numerical test of NNs

We now show the results obtained by using the trained machine learning models to infer the value of the NSD parameters $\{s_0, A, \sigma\}$. As already mentioned, the NNs are tested with 2000 different NSD parameters. For each of these sets of parameters, the curves $C(\tau, N)$ have been simulated as described in the previous subsections.

In order to determine the smallest amount of data required to reconstruct the NSD, we perform the training, validation and test of the NNs with sub-sets of the simulated curves. These sub-sets are defined by introducing the variable $\overline{N}$ that denotes the upper bound for the number of pulses $N \leqslant \overline{N}$ considered during the whole process. For example, for $\overline{N} = 16$ only the curves $C(\tau, N)$ with $N \in \{1, 8, 16\}$ are considered. Note that the sub-sets defined for each value of $\overline{N}$ contain the curves for all the different NSD parameters (6000 for training, 2000 for validation, and 2000 for testing), and for all the times τ in the intervals defined in section 2.1. In detail, the input of the NN is defined as the concatenation of all the values of $C(\tau, N)$, for τ in the intervals defined before and $N = 1, 8, 16, \ldots, \overline{N}$. Specifically, $\mathbf{x} = \{C(\tau_1, 1), C(\tau_2, 1), \ldots, C(\tau_n, 1), C(\tau_1, 2), \ldots, C(\tau_n, 2), \ldots, C(\tau_n, \overline{N})\}$.

The results of this analysis are shown in figure 2 (orange data), where the MSE (the loss function) between the inferred parameters $(\hat{s_0}, \hat{A}, \hat{\sigma})$ and the original parameters (s_0, A, σ) used to generate the dataset is plotted as a function of $\overline{N}$. Remarkably, the MSE seems to achieve its minimum value after $\overline{N} = 16$. This entails that the NNs do not significantly improve their precision on the reconstruction of the NSD by using more data to train them beyond this point.

To establish how accurately an NN reconstructs the NSD, we need to compare the corresponding results with those of a different method. In particular, we concentrate on the method used in [37], which is itself based on references [35, 36]. According to them, the decay of the coherence function $C(\tau, N)$ is analyzed as a function of N, for each fixed value of τ_i, i.e. for each fixed frequency component of the filter functions. In the limit of high N, the decay of the coherence is exponential, with a rate that is inversely proportional to the amplitude of the NSD [35]. In other words, the amplitude of the NSD is directly estimated for a discrete set of frequencies (each proportional to $1/\tau$). In contrast with the original proposals in references [35, 36], the method in [37] demonstrates that it is better to use the harmonics of the filter functions to reconstruct the NSD, in order to avoid extra broadening of the reconstructed spectrum. For this reason, we denote this method as *harmonics spectroscopy* (HS).

We have analyzed the same 2000 different curves $C(\tau, N)$ (used to test the machine learning models) also with the HS method. The results are collected and shown in figure 2 (blue data), where the first point is for $\overline{N} = 16$. This is due to the fact that, by definition, the HS method fits the decay of the coherence as a function of N. This is possible only for a dataset with at least three points (in this case $N = 1, 8, 16$). As one can observe in figure 2, the MSE values for the HS method (blue region) are always above the MSE values for the

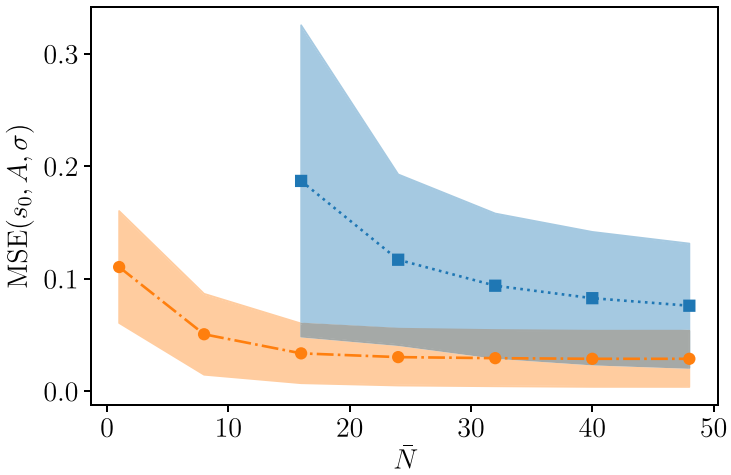

Figure 2. Mean-square-errors (MSEs) between original and estimated NSD parameters for a set of 2000 test cases. Orange bullets with dash-dotted line are the mean values returned by NNs. Blue squares with dotted line are the mean values provided by the HS method. Finally, shaded areas denote the standard deviation, taking into account all the 2000 cases.

NN method (orange region), especially for lower values of $\overline{N}$. These results demonstrate that the NN method can predict the parameters of the NSD with an improved accuracy (up to five times larger) with respect to the HS method. The test presented in this subsection have been performed with simulated data. In the next subsection we are going to repeat the same test but with experimental data.

2.4. Experimental test of NNs

By this point we know that NNs can reliably predict the NSD from *noisy* simulated data. In this section, we want to use the NNs (trained and validated with noisy simulated data) to reconstruct the NSD using experimental data.

As quantum sensor we use a spin qubit encoded in the electronic spin of the ground state of a single NV center in a bulk diamond at room temperature. This system has proven as a sensitive quantum probe of magnetic fields, with outstanding spacial resolution and sensitivity [51, 52]. The diamond sample in our experiments has a natural abundance of ^{13}C impurities (1.1%) that are randomly distributed in the diamond lattice [28–30]. The ^{13}C nuclear spins constitute the external environment of the NV center. They act as a collective bath of spins that induces dephasing into the NV electronic spin, limiting the its coherence time $T_2 \approx 100\,\mu$s. In the presence of strong bias magnetic field ($\geqslant$150 G) [37, 53], the weak coupling of the NV spin with these carbon impurities can be modeled as a classical stochastic field. The latter has a power spectrum density (here called NSD) that follows a Gaussian distribution centered at the Larmor frequency of the ^{13}C nuclear spins. In order to measure the NV spin coherence function $C(\tau, N)$, we apply a train of π pulses (in our case a CP sequence) to the NV spin qubit following the DD protocol described in figure 1. For more details on the experimental implementation and Hamiltonian of the system see [37]. We have performed this experiment for $N = \{1, 8, 16, 24, 32, 40, 48\}$, and for $\tau \in [3.3, 3.66]$ μs and $[5.5, 6.1]$ μs with sampling time $\Delta t = 20$ ns. The results are shown in figure 3(a) (blue bullets). Then, the collected coherence functions have been processed and employed to reconstruct the NSD parameters by means of both the NN (trained with the generated dataset) and the HS methods. In contrast with the test using simulated data in the previous section, in the experimental case we do not know the exact values of the NSD parameters. Therefore, we cannot calculate the MSE to quantify the accuracy of the reconstructed parameters. In order to estimate such accuracy we have used the following procedure: from the inferred NSD, the coherence curves $C(\tau, N)$ are simulated and then compared with the experimental results. An example of this comparison is shown in figure 3(a), where $C(\tau, N)$ is simulated under the assumption that the NSD parameters are inferred either by the machine learning models (orange) or by the HS method (red), both for $\overline{N} = 16$. Qualitatively it is clear that the orange curves are much closer to the experimental data, than the red curves.

There are several options to quantitatively compare the experimental data and the simulation results. Here we use both the *reduced chi-squared* χ^2_ν [54], and the MAE [55] between the experimental data and the predicted coherence functions $C(\tau, N)$ (see section 5 for more details). The results of this comparison are shown in figure 3(b), where χ^2_ν and the MAE are plotted as a function of $\overline{N}$. Remarkably, the NSD reconstructed by the NN for $\overline{N} = 16$ behaves better that any case using the HS method. It is worth observing that the same experimental data used to infer the NSD parameters are partially used to estimate the χ^2_ν and MAE($C(\tau, N)$). For example, for $\overline{N} = 16$, only the data for $N = 1, 8, 16$ are used to reconstruct the NSD, but

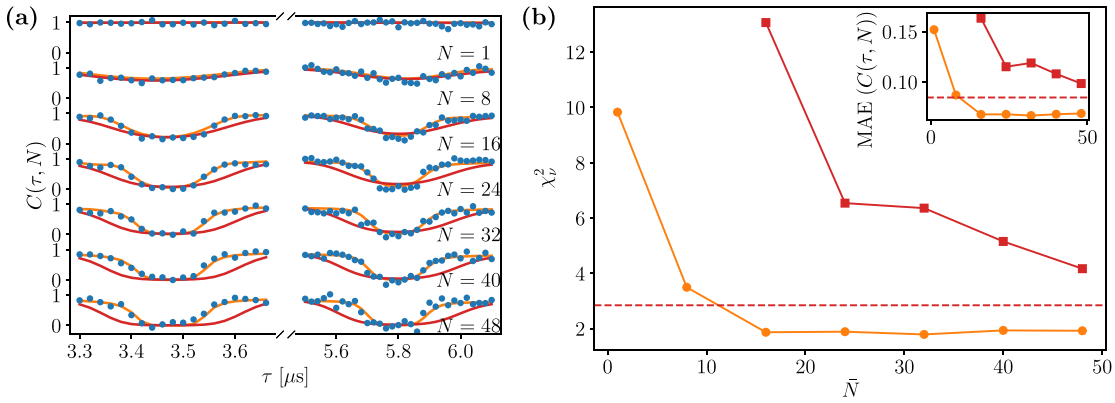

Figure 3. (a) Coherence function $C(\tau, N)$. The experimental data (blue bullets) are shown together with the simulated ones using the NSD predicted respectively by the HS method (red lines) and machine learning models (orange lines), both for $\overline{N} = 16$. (b) Reduced chi-squared χ^2_ν, obtained by comparing simulation and experimental data, as a function of $\overline{N}$. As in panel (a), orange and red curves refer to the NN and HS method, respectively. Instead, the dashed line denotes the value of the reduced chi-squared for the HS method when we employ additional measurements for $N = 56, 64, 72, 80$ in the interval $\tau \in [5.5, 6.1]$ μs. Inset: same results but quantified by the mean-absolute-error (MAE) between the experimental data and the predicted $C(\tau, N)$.

we employ all the data $N = 1, 8, 16, \ldots, 48$ to obtain the χ^2_ν and MAE$(C(\tau, N))$. Overall, we have observed enhanced performance in reconstructing the NSD of the collective bath of spins, with a maximum improvement (about seven times higher) for $\overline{N} = 16$. In other words, for $\overline{N} = 16$, once we reconstruct the NSD, the quantum sensor dynamics can be predicted with an average square deviation of $\simeq$1.86 experimental error-bars by using the NN method, or with an average square deviation of $\simeq$13 error-bars if we use the HS method.

3. Discussion

As shown pictorially in figure 1, the NN takes as input the spin qubit coherence functions (the coherence of the quantum sensor decays due to the presence of the external bath) obtained by using a set of different CP control sequences. The NN returns as output the parameters of the unknown NSD in the frequency domain. One can thus note that the NN, once validated, acts as a 'time-frequency converter' (making use of a quite complicated deconvolution) from the measured signals living in the time domain—the spin coherence functions—to the NSD defined in the frequency domain.

The results shown in the previous section, and summarized in figures 2 and 3(b), demonstrate that NNs can be used to reconstruct the NSD affecting a quantum sensor, achieving higher precision and with considerable less data than the standard HS method. Improved values of the reconstruction accuracy have been obtained with simulated and experimental data. Both the HS and NN methods are comparable—in terms of NSD reconstruction accuracy—for high values of $\overline{N}$, but not for small ones, where NNs give significantly better results. Moreover, the main result of our study is that NNs trained with data obtained for $\overline{N} = 16$ reconstruct the NSD more accurately than the best estimate provided by the HS method with $\overline{N} = 48$. This improvement is remarkable by itself, but it becomes more significant when we consider that the time required to complete these experiments has a growth faster than a linear function with respect to $\overline{N}$, following an arithmetic progression. As an example, the total time to perform all the experiments in the case of $\overline{N} = 16$ and 48 is respectively $\simeq$10 min and $\simeq$1.2 h (for this estimation we consider 10^5 repetitions as in our experiments, we recall that the total time for each repetition of the single experiment is $T = N\tau$). This is an under-estimation of the time difference between methods, because we are only considering the bare measurement time, without taking into account the time delay between different experiments. Furthermore, it is worth stressing that our results also show that deep learning has a predictive power since it can be applied to never-before-seen data. This naturally provides to the employed machine learning models a connotation of *robustness* that is crucial in real applications.

As a general comment, we stress that it is difficult to identify a definite reason why a machine learning model is more accurate (especially in the case of *small* $\overline{N}$) than a standard DD technique for noise spectroscopy. As said above, we observe experimentally that the employed NNs are able to learn non trivial patterns in the sequences of input-output data. What the NN learn is to invert $C(\tau, N)$ as a function of the noise parameters s_0, A and σ (see equations (2) and (4) in the main text) that we aim to reconstruct. It is known that NNs are universal approximators of functions: this can be the reason why they are well suited to find the parameters of the NSD from $C(\tau, N)$. Moreover, NNs do not consider approximations of the filter

function, and they manage to find the noise parameters even for input data containing values of $C(\tau, N)$ with N small. In contrast, the Álvarez–Suter method, as well as the HS methods in general, arises from approximating the filter function as a Dirac comb. This approximation is not valid for a small number of pulses, hence it is expected to poorly reconstruct the noise spectrum using standard DD techniques. In addition, reconstructing the parameters of an NSD using experimental data with an NN trained with synthetic data has been made possible by training the NN over an informative set of noise samples, used to generate the synthetic data. The latter, indeed, are given by a collection of values of $C(\tau, N)$ that implicitly include a parameterization of the NSD that is reasonable for the experimental setting; in our case, a Gaussian distribution whose offset, amplitude, width and center belong to finite-valued intervals estimated from similar experimental conditions (e.g. [37]).

Let us also observe that regression tasks, which are successfully solved by MLPs (one of the easiest form of NN), are less common with respect to the ones to carry out classification; a review of some example datasets and methods for regression is in [56]. Hence, we expect that the synthetic data used in this work could be useful as a test bed also to the audience of machine learning researchers and developers solving regression problems in different contexts. With this in mind, we share the training dataset with synthetic data and our codes for their generation, as well as the code for machine learning experiments and NSD reconstruction [57]. In this way, we promote the improvement of machine learning models for noise sensing purposes and their use to solve different regression tasks in the quantum estimation framework.

4. Conclusions

In this paper, we use NNs to carry out noise spectroscopy with a quantum sensor using DD sequences with a much smaller number of π pulses and, at the same time, achieving a higher reconstruction accuracy than standard methods (e.g. HS protocol). This means that with our proposal the noise spectroscopy procedure will take less time and give better results. More in detail, we experimentally demonstrate the capability of NNs to reconstruct the NSD of the collective nuclear spin bath that surrounds an electronic spin qubit, i.e. the ground state of a single NV center in bulk diamond at room temperature.

To conclude, we outline some possible outlooks for our work. First of all, one may evaluate the performance of NNs that are trained over input data obtained using DD control sequences with more degrees of freedom than the CP ones [58–62]. Secondly, deep learning might be applied to noise spectroscopy techniques beyond the HS methods, as for example optimal band-limited control protocols [39, 40] and even non-Gaussian noise characterization [63–65]. In this regard, notice that the NNs take as input the data associated to the spin coherence, and return as output the parameters of the NSD. Therefore, a new NN for the characterization of spin qubit's environment can be trained with coherence curves obtained from using any kind of coherent control sequences. The study of the performance of NNs trained with data from these more general control protocols is the next step in understanding how machine learning can enhance quantum sensing. In addition, it might be worth investigating how deep learning can be integrated to quantum sensing procedures that rely on the so-called stochastic quantum Zeno effect [66, 67], whereby the quantum probe is subjected to a sequence of quantum measurements that in the ideal case are designed to confine the dynamics of the probe around the initial (nominal) state [38, 68, 69]. We are also confident that the extent of our results can be quite easily replicated in other experimental settings, as e.g. superconducting flux qubits [70, 71], trapped ions [72, 73], cold atoms [74, 75], quantum dots [76, 77], nuclear magnetic resonance (NMR) experiments in molecules [36, 78], and nanoelectronic devices [79]. For such a purpose, one might slightly adapt the deep learning techniques used here to methods tailored for time series.

5. Methods

5.1. Technical details on the training of NNs

The NN models are developed using the PyTorch framework [80] on a machine with 32 CPU cores, 126 GB of RAM and a GeForce RTX 3090 GPU. The training time, including the optimization of the hyperparameters, is around 12 h for each $\overline{N}$.

The hyperparameters optimization is implemented by means of the Ray Tune library [81]. The *Hyperopt* package [82] uses the *Tree-structured Parzen Estimators* [83] algorithm as a Bayesian optimization to search for the best choice of the hyperparameters within a predefined search space. Hyperopt suggest the likely better configurations of the hyperparameters and the underlying model is updated after each trial that is run. The *ASHA* scheduler [84] is then used to stop the run of the least promising trials chosen by the search algorithm, thus speeding up the hyperparameters optimization process.

The optimized hyperparameters are the following. *(1)* The *number* of hidden layers decides the value of $L-1$ in equation (7). The hidden layers are between the input layer $\mathbf{h}[0]$ and the output layer $\mathbf{h}[L]$. *(2)* The

Table 1. Hyperparameters for the employed machine learning models. For each value of $\overline{N}$ (that determines the size of the input layer) we report: the number of hidden layers (h. l. num.), the dimension of each hidden layer (h. l. dim.) and the values of learning rate (learning r.), batch size, dropout and weight decay (weight d.).

$\overline{N}$	h. l. num.	h. l. dim.	Learning r.	Batch size	Dropout	Weight d.
1	1	2	10^{-2}	16	0	10^{-3}
8	5	328	10^{-4}	4	0	10^{-4}
16	2	133	10^{-3}	8	0	10^{-6}
24	3	224	10^{-4}	2	0	10^{-4}
32	3	145	10^{-4}	4	0	10^{-5}
40	3	286	10^{-4}	4	0	10^{-4}
48	3	38	10^{-3}	8	0	10^{-4}

dimension of the hidden layers is the value of q in equation (6) that, for the sake of simplicity, is equal for all the layers in equation (7). Both the number and dimension of the hidden layers are chosen by sampling log-uniformly an integer value from the space $[1, 32)$ and $[1, 1024)$, respectively. *(3)* The *learning rate* is responsible for the length of the gradient descent step and it is optimized with a choice between 10^{-2}, 10^{-3} and 10^{-4}. *(4)* The *batch size* denotes the dimension of the batch on which the loss function is summed for the gradient calculation in a single descent step. The batch size is chosen between 2, 4, 8, 16, 32. *(5)* The *dropout* is a regularization strategy that aims to reduce the overfitting by randomly turn off the NN neurons with a predefined probability. Such probability is one among 0 (no dropout), 0.2 and 0.5. *(6)* The *weight decay* is another regularization technique that adds to the loss function the squared weights of the NN multiplied by a decay value. The latter value is optimized choosing between 0 (no decay), 10^{-6}, 10^{-5}, 10^{-4} and 10^{-3}.

To facilitate the reproducibility of the experiments, we summarize in table 1 the optimal values of the hyperparameters for the trained models. Each value of $\overline{N}$ defines the input size of the NN. Therefore, a different optimization of the hyperparameters is performed for each case.

5.2. Definition of quantifiers for reconstruction accuracy

The accuracy of NN and HS methods can be estimated by using the reconstructed NSD to simulate the coherence function $C(\tau, N)$, and 'measuring' the distance between the simulated data and the experimental values. To do so, we use the *reduced chi-squared* χ^2_ν, and the mean-absolute-error (MAE(C)): we define $C_e \pm \delta C_e$ (C_s) as the experimental (simulated) values of $C(\tau, N)$, where δC_e is the standard deviation of the experimental data. Then we can write *reduced chi-squared* and the MAE as

$$\chi^2_\nu \equiv \frac{1}{\nu} \sum_{n,N} \frac{(C_e(\tau_n, N) - C_s(\tau_n, N))^2}{\delta C_e(\tau_n, N)^2} \tag{12}$$

$$\mathrm{MAE}(C) \equiv \frac{1}{\nu} \sum_{n,N} |C_e(\tau_n, N) - C_s(\tau_n, N)|\,, \tag{13}$$

where $N = \{1, 8, 16, 24, \ldots, \overline{N}\}$, $\{\tau_n\}$ are the values of the time between pulses within the time intervals defined in main text, and ν is the total number of elements in the sum. Notice that χ^2_ν takes into account the experimental precision to scale the difference between experiment and simulation. The results showing both χ^2_ν and the MAE are in figure 3.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/trianam/noiseSpectroscopyNV.

Acknowledgment

This work was supported by the European Commission's Horizon Europe Framework Programme under the Research and Innovation Action GA n. 101070546-MUQUABIS, and by the European Defence Agency under the project Q-LAMPS Contract No. B PRJ-RT-989. S H G acknowledges support from CNR-FOE-LENS-2020. S M acknowledges financial support from PNRR MUR project PE0000023-NQSTI. F C also acknowledges the European Union's Horizon 2020 research and innovation programme under FET-OPEN GA n. 828946-PATHOS.

ORCID iDs

Stefano Martina https://orcid.org/0000-0001-6024-1752
Santiago Hernández-Gómez https://orcid.org/0000-0002-2432-5729
Stefano Gherardini https://orcid.org/0000-0002-9254-507X
Filippo Caruso https://orcid.org/0000-0002-8366-4296
Nicole Fabbri https://orcid.org/0000-0002-0525-0696

References

[1] Degen C L, Reinhard F and Cappellaro P 2017 Quantum sensing *Rev. Mod. Phys.* **89** 035002
[2] Hernández-Gómez S and Fabbri N 2021 Quantum control for nanoscale spectroscopy with diamond nitrogen-vacancy centers: a short review *Front. Phys.* **8** 610868
[3] Poggiali F, Cappellaro P and Fabbri N 2018 Optimal control for one-qubit quantum sensing *Phys. Rev.* X **8** 021059
[4] Müller M M, Gherardini S and Caruso F 2018 Noise-robust quantum sensing via optimal multi-probe spectroscopy *Sci. Rep.* **8** 14278
[5] Rembold P, Nimba Oshnik M M, Müller S M, Montangero S, Calarco T and Neu E 2020 Introduction to quantum optimal control for quantum sensing with nitrogen-vacancy centers in diamond *AVS Quantum Sci.* **2** 024701
[6] Marshall A *et al* 2022 Macroscopic hyperpolarization enhanced with quantum optimal control *Phys. Rev. Res.* **4** 043179
[7] Zhang F, Xing J, Xiaoxiao H, Pan X and Long G 2023 Coupling-selective quantum optimal control in weak-coupling NV-13 C system *AAPPS Bull.* **33** 2
[8] Bishop C M 2006 *Pattern Recognition and Machine Learning* (*Information Science and Statistics*) (Berlin: Springer)
[9] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press) (available at: www.deeplearningbook.org)
[10] Youssry A, Paz-Silva G A and Ferrie C 2020 Characterization and control of open quantum systems beyond quantum noise spectroscopy *npj Quantum Inf.* **6** 95
[11] Martina S, Gherardini S and Caruso F 2021 Machine learning approach for quantum non-Markovian noise classification (arXiv:2101.03221)
[12] Wise D F, Morton J J L and Dhomkar S 2021 Using deep learning to understand and mitigate the qubit noise environment *PRX Quantum* **2** 010316
[13] Aharon N, Rotem A, McGuinness L P, Jelezko F, Retzker A and Ringel Z 2019 NV center based nano-NMR enhanced by deep learning *Sci. Rep.* **9** 17802
[14] Santagati R *et al* 2019 Magnetic-field learning using a single electronic spin in diamond with one-photon readout at room temperature *Phys. Rev.* X **9** 021019
[15] Kyunghoon Jung M H, Abobeih J Y, Yun J, Kim G, Hyunseok O, Henry A, Taminiau T H and Kim D 2021 Deep learning enhanced individual nuclear-spin detection *npj Quantum Inf.* **7** 41
[16] Strikis A, Qin D, Chen Y, Benjamin S C and Li Y 2021 Learning-based quantum error mitigation *PRX Quantum* **2** 040330
[17] Harper R, Flammia S T and Wallman J J 2020 Efficient learning of quantum noise *Nat. Phys.* **16** 1184
[18] Martina S, Buffoni L, Gherardini S and Caruso F 2022 Learning the noise fingerprint of quantum devices *Quantum Mach. Intell.* **4** 1–12
[19] Martina S, Gherardini S, Buffoni L and Caruso F 2022 Noise fingerprints in quantum computers: machine learning software tools *Softw. Impacts* **12** 100260
[20] Koolstra G *et al* 2022 Monitoring fast superconducting qubit dynamics using a neural network *Phys. Rev.* X **12** 031017
[21] Gong M *et al* 2022 Quantum neuronal sensing of quantum many-body states on a 61-qubit programmable superconducting processor *Sci. Bull.* accepted (https://doi.org/10.1016/j.scib.2023.04.003)
[22] Ding L, Wang H, Wang Y and Wang S 2022 Based on quantum topological stabilizer color code morphism neural network decoder *Quantum Eng.* **2022** 9638108
[23] Tilaye G F and Pandey A 2023 Investigating the effects of hyperparameters in quantum-enhanced deep reinforcement learning *Quantum Eng.* **2023** 2451990
[24] Zhang Y and Ni Q 2021 Design of quantum neuron model for quantum neural networks *Quantum Eng.* **3** e75
[25] Wei S, Chen Y, Zhou Z and Long G 2022 A quantum convolutional neural network on NISQ devices *AAPPS Bul.* **32** 2
[26] Viola L and Lloyd S 1998 Dynamical suppression of decoherence in two-state quantum systems *Phys. Rev.* A **58** 2733–44
[27] Faoro L and Viola L 2004 Dynamical suppression of $1/f$ noise processes in qubit systems *Phys. Rev. Lett.* **92** 117905
[28] Taylor J M, Cappellaro P, Childress L, Jiang L, Budker D, Hemmer P R, Yacoby A, Walsworth R and Lukin M D 2008 High-sensitivity diamond magnetometer with nanoscale resolution *Nat. Phys.* **4** 810–6
[29] Goldstein G, Cappellaro P, Maze J R, Hodges J S, Jiang L, Sørensen A S and Lukin M D 2011 Environment-assisted precision measurement *Phys. Rev. Lett.* **106** 140502
[30] Abobeih M H, Cramer J, Bakker M A, Kalb N, Markham M, Twitchen D J and Taminiau T H 2018 One-second coherence for a single electron spin coupled to a multi-qubit nuclear-spin environment *Nat. Commun.* **9** 2552
[31] Cywiński Ł, Lutchyn R M, Nave C P and Das Sarma S 2008 How to enhance dephasing time in superconducting qubits *Phys. Rev.* B **77** 174509
[32] Biercuk M J, Doherty A C and Uys H 2011 Dynamical decoupling sequence construction as a filter-design problem *J. Phys. B: At. Mol. Opt. Phys.* **44** 154002
[33] Carr H Y and Purcell E M 1954 Effects of diffusion on free precession in nuclear magnetic resonance experiments *Phys. Rev.* **94** 630–8
[34] Maze J R *et al* 2008 Nanoscale magnetic sensing with an individual electronic spin in diamond *Nature* **455** 644–7
[35] Yuge T, Sasaki S and Hirayama Y 2011 Measurement of the noise spectrum using a multiple-pulse sequence *Phys. Rev. Lett.* **107** 170504
[36] Álvarez G A and Suter D 2011 Measuring the spectrum of colored noise by dynamical decoupling *Phys. Rev. Lett.* **107** 230501
[37] Hernández-Gómez S, Poggiali F, Cappellaro P and Fabbri N 2018 Noise spectroscopy of a quantum-classical environment with a diamond qubit *Phys. Rev.* B **98** 214307

[38] Do H V, Lovecchio C, Mastroserio I, Fabbri N, Cataliotti F S, Gherardini S, Müller M M, Dalla Pozza N and Caruso F 2019 Experimental proof of quantum Zeno-assisted noise sensing *New J. Phys.* **21** 113056
[39] Frey V M, Mavadia S, Norris L M, de Ferranti W, Lucarelli D, Viola L and Biercuk M J 2017 Application of optimal band-limited control protocols to quantum noise sensing *Nat. Commun.* **8** 2189
[40] Frey V, Norris L M, Viola L and Biercuk M J 2020 Simultaneous spectral estimation of dephasing and amplitude noise on a qubit sensor via optimally band-limited control *Phys. Rev. Appl.* **14** 024021
[41] Wang G, Zhu Y, Li B, Li C, Viola L, Cooper A and Cappellaro P 2022 Digital noise spectroscopy with a quantum sensor (arXiv:2212.09216)
[42] Szańkowski P, Ramon G, Krzywda J, Kwiatkowski D and Cywiński Ł 2017 Environmental noise spectroscopy with qubits subjected to dynamical decoupling *J. Phys.: Condens. Matter* **29** 333001
[43] Szańkowski P and Cywiński Ł 2018 Accuracy of dynamical-decoupling-based spectroscopy of Gaussian noise *Phys. Rev.* A **97** 032101
[44] Gordon G, Erez N and Kurizki G 2007 Universal dynamical decoherence control of noisy single- and multi-qubit systems *J. Phys. B: At. Mol. Opt. Phys.* **40** S75–S93
[45] Gordon G, Kurizki G and Lidar D A 2008 Optimal dynamical decoherence control of a qubit *Phys. Rev. Lett.* **101** 010403
[46] Dalla Pozza N, Gherardini S, Müller M M and Caruso F 2019 Role of the filter functions in noise spectroscopy *Int. J. Quantum Inf.* **17** 1941008
[47] Glorot X, Bordes A and Bengio Y 2011 Deep sparse rectifier neural networks *Proc. 14th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research* vol 15) *(Fort Lauderdale, FL, USA, 11–13 April 2011)* ed G Gordon, D Dunson and M Dudík (PMLR) pp 315–23
[48] Nair V and Hinton G E 2010 Rectified linear units improve restricted Boltzmann machines *Proc. 27th Int. Conf. on Int. Conf. on Machine Learning* (*ICML'10*) (Madison, WI: Omnipress) pp 807–14
[49] Shalev-Shwartz S and Ben-David S 2014 *Understanding Machine Learning: From Theory to Algorithms* (Cambridge: Cambridge University Press)
[50] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
[51] Doherty M W, Manson N B, Delaney P, Jelezko F, Wrachtrup J and Hollenberg L C L 2013 The nitrogen-vacancy colour centre in diamond *Phys. Rep.* **528** 1–45
[52] Rondin L, Tetienne J-P, Hingant T, Roch J-F, Maletinsky P and Jacques V 2014 Magnetometry with nitrogen-vacancy defects in diamond *Rep. Prog. Phys.* **77** 056503
[53] Reinhard F, Shi F, Zhao N, Rempp F, Naydenov B, Jan Meijer L T Hall L H, Jiangfeng D, Liu R-B and Wrachtrup J 2012 Tuning a spin bath through the quantum-classical transition *Phys. Rev. Lett.* **108** 200402
[54] Hughes I and Hase T 2010 *Measurements and Their Uncertainties: A Practical Guide to Modern Error Analysis* (Oxford,: Oxford University Press) pp 107–107
[55] Sammut C and Webb G I (eds) 2010 Mean absolute error *Encyclopedia of Machine Learning* (Boston, MA: Springer) pp 652–652
[56] Fernández-Delgado M, Sirsat M S, Cernadas E, Alawadi S, Barro S and Febrero-Bande M 2019 An extensive experimental survey of regression methods *Neural Netw.* **111** 11–34
[57] Martina S and Hernández-Gómez S 2023 *GitHub* (available at: https://github.com/trianam/noiseSpectroscopyNV)
[58] Uhrig G S 2007 Keeping a quantum bit alive by optimized π-pulse sequences *Phys. Rev. Lett.* **98** 100504
[59] Zhao N, Hu J-L, Ho S-W, Wan J T K and Liu R B 2011 Atomic-scale magnetometry of distant nuclear spin clusters via nitrogen-vacancy spin in diamond *Nat. Nanotechnol.* **6** 242–6
[60] Souza A M, Álvarez G A and Suter D 2011 Robust dynamical decoupling for quantum computing and quantum memory *Phys. Rev. Lett.* **106** 240501
[61] Zhao N, Wrachtrup J and Liu R-B 2014 Dynamical decoupling design for identifying weakly coupled nuclear spins in a bath *Phys. Rev.* A **90** 032319
[62] Casanova J, Wang Z-Y, Haase J F and Plenio M B 2015 Robust dynamical decoupling sequences for individual-nuclear-spin addressing *Phys. Rev.* A **92** 042304
[63] Paz-Silva G A and Viola L 2014 General transfer-function approach to noise filtering in open-loop quantum control *Phys. Rev. Lett.* **113** 250501
[64] Norris L M, Paz-Silva G A and Viola L 2016 Qubit noise spectroscopy for non-Gaussian dephasing environments *Phys. Rev. Lett.* **116** 150503
[65] Sung Y *et al* 2019 Non-Gaussian noise spectroscopy with a superconducting qubit sensor *Nat. Commun.* **10** 3715
[66] Shushin A I 2011 The effect of measurements, randomly distributed in time, on quantum systems: stochastic quantum Zeno effect *J. Phys. A: Math. Theor.* **44** 055303
[67] Gherardini S, Gupta S, Cataliotti F S, Smerzi A, Caruso F and Ruffo S 2016 Stochastic quantum Zeno by large deviation theory *New J. Phys.* **18** 013048
[68] Müller M M, Gherardini S, Dalla Pozza N and Caruso F 2020 Noise sensing via stochastic quantum Zeno *Phys. Lett.* A **384** 126244
[69] Virzì S *et al* 2022 Quantum Zeno and anti-Zeno probes of noise correlations in photon polarization *Phys. Rev. Lett.* **129** 030401
[70] Bylander J, Gustavsson S, Yan F, Yoshihara F, Harrabi K, Fitch G, Cory D G, Nakamura Y, Tsai J-S and Oliver W D 2011 Noise spectroscopy through dynamical decoupling with a superconducting flux qubit *Nat. Phys.* **7** 565
[71] Yoshihara F, Nakamura Y, Yan F, Gustavsson S, Bylander J, Oliver W D and Tsai J-S 2014 Flux qubit noise spectroscopy using Rabi oscillations under strong driving conditions *Phys. Rev.* B **89** 020503
[72] Biercuk M J, Uys H, VanDevender A P, Shiga N, Itano W M and Bollinger J J 2009 Optimized dynamical decoupling in a model quantum memory *Nature* **458** 996
[73] Kotler S, Akerman N, Glickman Y, Keselman A and Ozeri R 2011 Single-ion quantum lock-in amplifier *Nature* **473** 61
[74] Sagi Y, Almog I and Davidson N 2010 Process tomography of dynamical decoupling in a dense cold atomic ensemble *Phys. Rev. Lett.* **105** 053201
[75] Almog I, Sagi Y, Gordon G, Bensky G, Kurizki G and Davidson N 2011 Direct measurement of the system–environment coupling as a tool for understanding decoherence and dynamical decoupling *J. Phys. B: At. Mol. Opt. Phys.* **44** 154006
[76] Chan K W *et al* 2018 Assessment of a silicon quantum dot spin qubit environment via noise spectroscopy *Phys. Rev. Appl.* **10** 044017
[77] Malinowski F K *et al* 2017 Spectrum of the nuclear environment for GaAs spin qubits *Phys. Rev. Lett.* **118** 177702
[78] Fu Y, Wu Y, Dai Y, Qin X, Rong X and Du J 2021 Molecular-spin-qubit noise spectroscopy through dynamical decoupling *Phys. Rev. Appl.* **15** L061001
[79] Muhonen J T *et al* 2014 Storing quantum information for 30 seconds in a nanoelectronic device *Nat. Nanotechnol.* **9** 986

[80] Paszke A *et al* 2019 Pytorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems* vol 32, ed H Wallach, H Larochelle, A Beygelzimer, F d' Alché-Buc, E Fox and R Garnett (Curran Associates, Inc.)
[81] Liaw R, Liang E, Nishihara R, Moritz P, Gonzalez J E and Stoica I 2018 Tune: a research platform for distributed model selection and training (arXiv:1807.05118)
[82] Bergstra J, Yamins D and Cox D 2013 Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures *Proc. 30th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research* vol 28) *(Atlanta, Georgia, USA, 17–19 June 2013)* ed S Dasgupta and D McAllester (PMLR) pp 115–23
[83] Bergstra J, Bardenet R, Bengio Y and Kégl B 2011 Algorithms for hyper-parameter optimization *Advances in Neural Information Processing Systems* vol 24
[84] Li L, Jamieson K, Rostamizadeh A, Gonina E, Ben-tzur J, Hardt M, Recht B and Talwalkar A 2020 A system for massively parallel hyperparameter tuning *Proc. Machine Learning and Systems* vol 2, ed I Dhillon, D Papailiopoulos and V Sze pp 230–46