PAPER • OPEN ACCESS

Spectral density classification for environment spectroscopy

View the article online for updates and enhancements.

MACHINE
LEARNING
Science and Technology

Spectral density classification for environment spectroscopy

J Barr[1,*] ![ORCID], **G Zicari**[1] ![ORCID], **A Ferraro**[1,2] ![ORCID] and **M Paternostro**[1,3] ![ORCID]

1 Centre for Quantum Materials and Technologies, School of Mathematics and Physics, Queen's University Belfast, Belfast BT7 1NN, United Kingdom
2 Dipartimento di Fisica 'Aldo Pontremoli', Università degli Studi di Milano, I-20133 Milano, Italy
3 Dipartimento di Fisica e Chimica - Emilio Segrè, Uiversità degli Studi di Palermo, Palermo 90123, Italy
* Author to whom any correspondence should be addressed.

E-mail: jbarr24@qub.ac.uk and G.Zicari@qub.ac.uk

Abstract

Spectral densities encode the relevant information characterizing the system–environment interaction in an open-quantum system problem. Such information is key to determining the system's dynamics. In this work, we leverage the potential of machine learning techniques to reconstruct the features of the environment. Specifically, we show that the time evolution of a system observable can be used by an artificial neural network to infer the main features of the spectral density. In particular, for relevant examples of spin-boson models, we can classify with high accuracy the Ohmicity parameter of the environment as either Ohmic, sub-Ohmic or super-Ohmic, thereby distinguishing between different forms of dissipation.

1. Introduction

Recent progress in the field of quantum technologies has advanced our capabilities to control quantum systems and exploit their non-classical properties. Yet, this task presents significant challenges. Quantum systems are inherently open, as they inevitably interact with their surrounding environment [1, 2]. They are thus susceptible to gain and losses, as well as to the genuine quantum phenomenon of decoherence [3–5], which disrupts the phase coherence of superposition states, posing a major obstacle in preserving quantum states [6, 7]. If we are to effectively devise strategies for mitigating adverse environmental influences on a system, it is crucial to have a comprehensive understanding of the effects that need to be addressed when facing open quantum dynamics. This, in turn, requires a full characterization of the mechanism governing the system–environment interaction.

To address such challenge, here we tackle the problem of characterizing environmental effects on an open quantum system harnessing recent advances in the field of Machine Learning (ML). The latter has opened up new data-driven approaches, which have shown their effectiveness in various applications in the field of quantum technologies [8]. Among those, some are very close to the spirit of this work. ML-based methodologies have been applied to quantum tomography [9–11], quantum channel discrimination [12], simulation of open quantum systems [13–16], as well as quantum control [17–19]. References [20, 21] reported the deployment of deep-learning methods to the inference of photon correlation functions and phonon blockade effects based on homodyne-detection schemes.

We focus on the typical open quantum system scenario, where we are able to effectively describe and control the reduced system, as opposed to the infinitely many uncontrollable environmental degrees of freedom which are responsible for dissipation and decoherence. In this setting, we focus on the interaction of a given system with an external environment in terms of the Spectral Density (SD), which, by encoding full information about the system–environment coupling, allows us to determine the two-time correlation function of the environment. Having full knowledge of this quantity allows us to predict the temporal behavior of an open quantum system without a full microscopic description of the environment.

The SD for a given system–environment interaction, however, is rarely directly available and challenging to calculate from first principles. The form of a SD is at best phenomenologically inferred through empirical data gathered from experimental observations, and at worst *guessed* using ad hoc assumptions, which might

result in significant discrepancies between the predicted and actual dynamics of the system [22]. In this work, we consider the case of a quantum system interacting with a bosonic thermal bath. Depending on the nature of the system-bath interaction, the system dynamics can manifest as either pure dephasing or amplitude damping [23–25]. The particular choice of the SD in this setting is responsible for possible memory effects. On one hand, we can encounter a scenario where the information is monotonically flowing from the system to the bath, i.e. the usual scenario characterizing quantum Markovian processes [26, 27]. On the other hand, some functional forms of the SD are suitable to model a physical situation in which the system, dynamically interacting with its environment, can partially retrieve the information that was previously lost—these processes are dubbed as non-Markovian instead [28, 29].

Prior works have studied the use of ML for noise characterization in open quantum systems. Various methods have been explored, such as studying the noise in qubit systems using two-pulse echo decay curves [30], and random pulse sequences that are applied to the qubit [31]. Additionally, other studies have focused on constructing the power spectral density for ensembles of carbon impurities around a nitrogen vacancy centre in diamond [32], and inferring the environment affecting superconducting qubits [33].

In this work, we show that an artificial Neural Network (NN) can be used to *classify* the SD characterizing the dynamics of a system, based on its features. Previous research has examined the classification of *aspects* of noise in open quantum systems. For instance, in [34] ML techniques were used to discern between Markovian and non-Markovian noise. More pertinent to the matter at hand, aspects of the problem of distinguishing between Ohmic, sub-Ohmic and super-Ohmic SDs have already been studied: in [35], a scenario where a probe qubit is used to access a second inaccessible one is proposed to infer the Ohmicity class by using NNs and leveraging the special features of quantum synchronization. In [36], a different use of NNs was put forward as tomographic data at just two instants of time were used, rather than a time-series approach. In contrast, this work takes a simpler approach by utilizing the time evolution of a system observable for classification without the need for a probe system or tomographically complete information. We focus on the case of a general Spin-Boson (SB) model to show that, even when the environment cannot be exactly traced out to infer the reduced dynamics of a system, a NN can classify the SD with high accuracy. Furthermore, we discuss the limitations imposed by the fluctuation of the parameters in the SD, the number of sampled points in the time signals, and measurement sampling noise. Our study emphasizes the potential of ML techniques to characterize environments with arbitrary SDs.

The remainder of this paper is structured as follows: in section 2 we provide an introduction to the general setting under consideration, as well as the ML approach utilized. Specifically, we examine an arbitrary system that is interacting with a bosonic environment and we give some background on the ML model used, namely, NNs. Next, in section 3 we detail the physical models that are considered. We investigate two SB models: in the first case, we are able to exactly derive the pure dephasing dynamics starting from the full system-bath unitary evolution; in the second case, we work in the weak coupling limit to approximately derive the reduced dynamics of the system. In both cases, the dynamics can feature non-Markovian effects, depending on the SD we select. In section 4 we discuss the architecture of the NNs, along with a detailed discussion of the results of training and testing for each model. Finally, we give our conclusive remarks and discuss our future outlook in section 5.

2. General setting and methods

Let us consider the general setting of an arbitrary system interacting with an environment which is comprised of infinitely many bosonic modes, as shown in figure 1. This scenario reproduces the ubiquitous Caldeira–Leggett model, which describes the motion of a quantum particle undergoing a Brownian motion [37, 38]. The full (time-independent) Hamiltonian reads as

$$\hat{H} = \hat{H}_S + \hat{H}_B + \hat{H}_I, \tag{1}$$

where $\hat{H}_S$ and $\hat{H}_B$ are the Hamiltonian operators of the system and the environment, respectively. The system–environment interaction term $\hat{H}_I$ is expressed in the form

$$\hat{H}_I = \hat{X} \otimes \hat{B}, \tag{2}$$

where $\hat{X}$ is a generic system operator, while $\hat{B}$ is an operator of the bath. We take the latter as

$$\hat{B} = \sum_k \left(g_k \hat{b}_k^\dagger + g_k^* \hat{b}_k \right), \tag{3}$$

Figure 1. Sketch of a generic open quantum system *S* interacting with a bosonic environment composed of infinitely many harmonic oscillators labelled by the integer *n*. Each oscillator has frequency ω_n and is coupled to the system at a rate g_n.
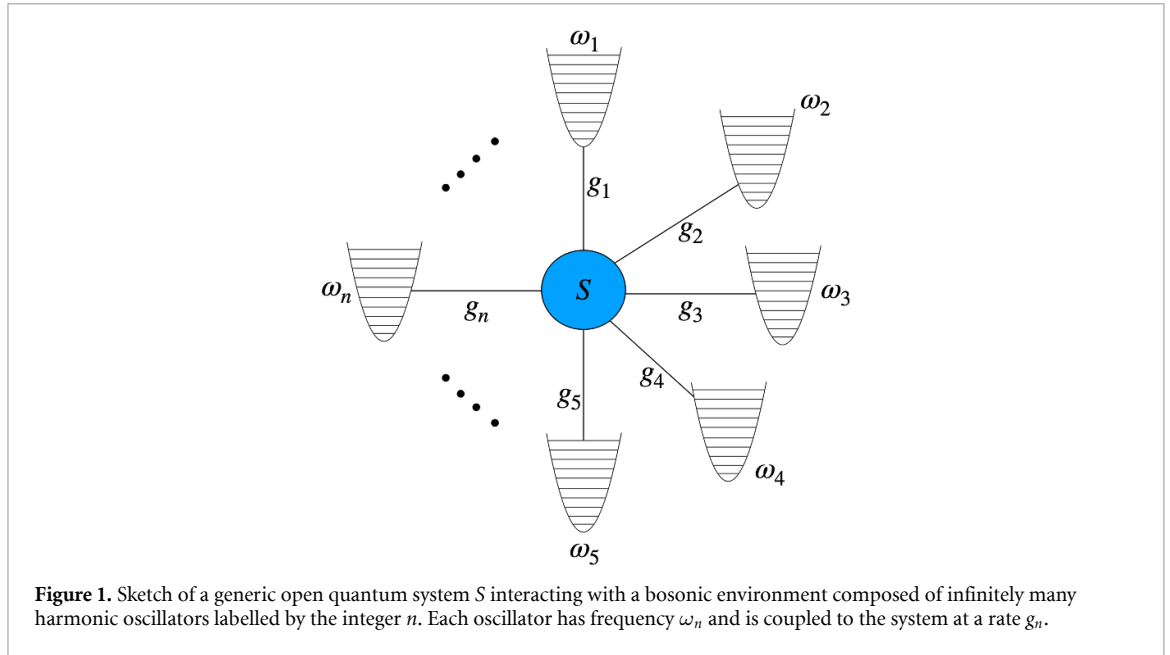
where the coefficient g_k accounts for the interaction strength between the *k*th mode of frequency ω_k, while $\hat{b}_k^\dagger$ and $\hat{b}_k$ are the creation and annihilation operators associated with it. The coupling coefficients enter in the formal definition of the SD, i.e. $J(\omega) = \sum_k |g_k|^2 \delta(\omega - \omega_k)$, the latter encoding all the information about the system–environment interaction. Since we are interested in the typical irreversible open system scenario, we will assume that the distribution of modes forms a continuum, so that the system dynamics does not display recurrences [1, 39, 40]. In this limit, the SD appears in the expression for the correlation function of a bosonic bath, defined as $\alpha_\beta(t) \equiv \langle \hat{B}(t)\hat{B}(0)\rangle_B$, where $\hat{B}(t)$ is the bath operator in the interaction picture with respect to the free Hamiltonian $\hat{H}_0 = \hat{H}_S + \hat{H}_B$. In appendix A, we show that if the environment is in a thermal Gibbs state, the correlation function can also be expressed as

$$\alpha_\beta(t) \equiv \nu(t) + i\mu(t), \tag{4}$$

where

$$\begin{bmatrix} \nu(t) \\ \mu(t) \end{bmatrix} = \int_0^\infty J(\omega) \begin{bmatrix} \cos(\omega t)\coth\left(\frac{\beta\omega}{2}\right) \\ -\sin(\omega t) \end{bmatrix} d\omega, \tag{5}$$
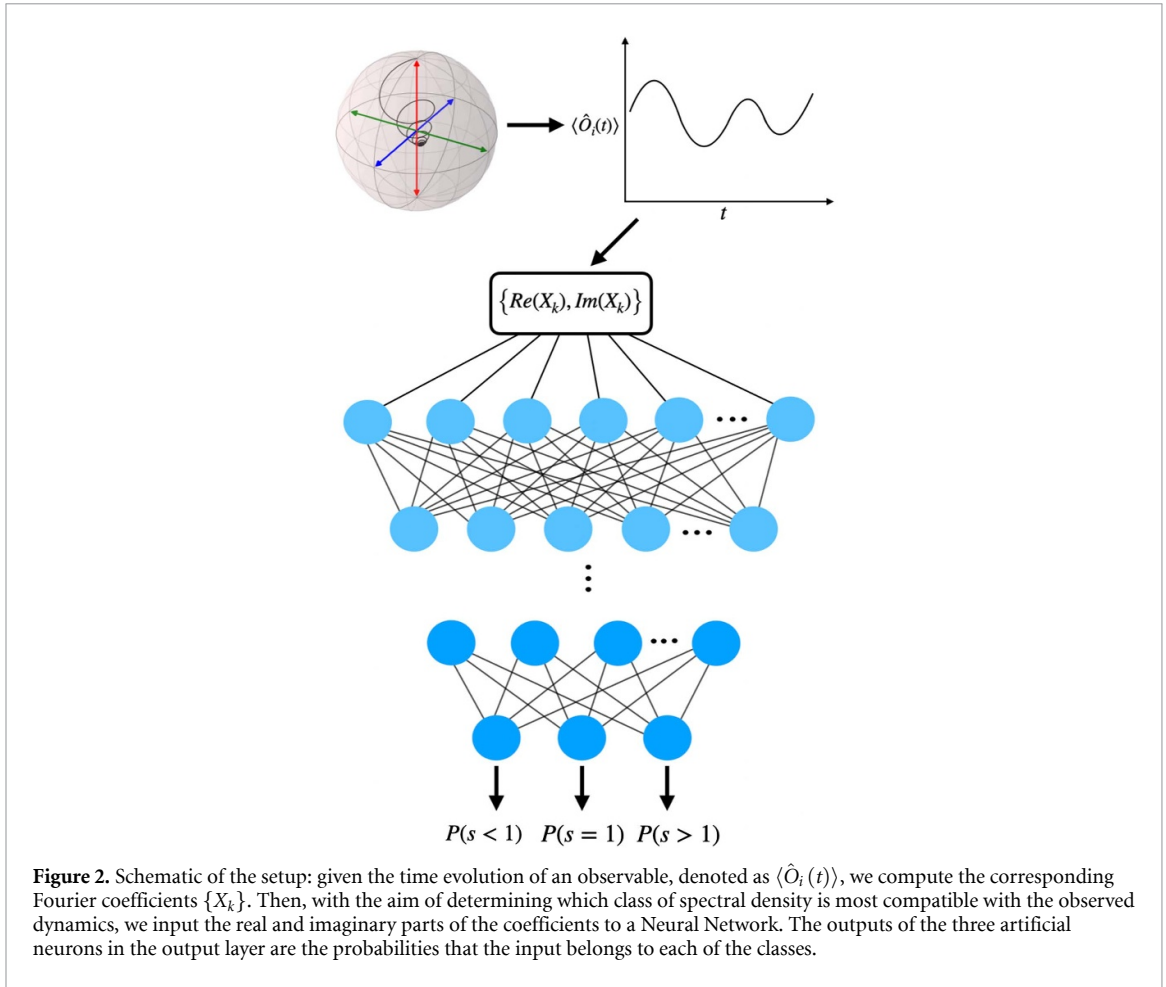
with $\beta = 1/T$. Note that hereafter we will work in units such that $\hbar = 1$ and $k_B = 1$. The two functions $\nu(t)$ and $\mu(t)$ are also referred to as noise and dissipation kernels, respectively: the latter is independent of the temperature of the environment. The effective dynamics of the system, governed by a master equation, crucially depends on the correlation function $\alpha_\beta(t)$, which represents the fingerprint of the environment. The function $\alpha_\beta(t)$ is ultimately determined by the shape of the SD, which essentially contains all of the information about the environment needed to solve the dynamics of the system, and, thus, obtain the time evolution of any of its observables. The expectation value of a generic system observable at time *t* is indeed given by

$$\langle \hat{O}(t)\rangle \equiv \mathrm{Tr}_{SB}\left(\hat{O}e^{-i\hat{H}t}\hat{\rho}_{SB}^0 e^{i\hat{H}t}\right), \tag{6}$$

where $\hat{H}$ is the system–environment Hamiltonian of equation (1), while the global initial state is factorized as $\hat{\rho}_{SB}^0 = \hat{\rho}^0 \otimes \hat{\rho}_B$, with $\hat{\rho}^0$ and $\hat{\rho}_B$ being the initial system and environmental states, respectively. We assume the environment to be given by a large bosonic thermal reservoir, i.e. $\hat{\rho}_B = e^{-\beta\hat{H}_B}/\mathcal{Z}_B$, where $\mathcal{Z}_B \equiv \mathrm{tr}_B(e^{-\beta\hat{H}_B})$ is the reservoir partition function. Under these hypotheses, it can be shown that the only environmental quantity entering in the expression of $\langle \hat{O}(t)\rangle$ is the SD $J(\omega)$.

Here we focus on special classes of SDs, which can be expressed as [38, 41]

$$J(\omega) = \eta\omega_c^{1-s}\omega^s f(\omega, \omega_c), \tag{7}$$

Figure 2. Schematic of the setup: given the time evolution of an observable, denoted as $\langle \hat{O}_i(t) \rangle$, we compute the corresponding Fourier coefficients $\{X_k\}$. Then, with the aim of determining which class of spectral density is most compatible with the observed dynamics, we input the real and imaginary parts of the coefficients to a Neural Network. The outputs of the three artificial neurons in the output layer are the probabilities that the input belongs to each of the classes.

where $s > 0$ is known as Ohmicity parameter, and $\eta > 0$ is the coupling strength between the system and the environment. The constant ω_c is the cut-off frequency, while $f(\omega, \omega_c)$ is the cut-off function, which ensures that $J(\omega) \to 0$ in the limit of large frequencies, i.e. $\omega \to \infty$. In what follows we consider the exponential cutoff, namely $f(\omega, \omega_c) = e^{-\omega/\omega_c}$. Depending on the value of s, we model different system–environment couplings, corresponding to various physical scenarios [38, 41, 42]. SDs with $s = 1$ (i.e. linear in the frequency ω) are called Ohmic, while those for which $s > 1$ ($s < 1$) are known as super-Ohmic (sub-Ohmic).

In this work, we will use the tools provided by ML to classify the SD characterizing the system–environment interaction. Specifically, we use an artificial NN that comprises many artificial neurons—essentially a computational unit—arranged in a series of layers, as in figure 2 [8, 43]. Given a set of inputs $\{x_i\}$, each neuron computes the weighted sum

$$z = \sum_i w_i x_i + b, \tag{8}$$

with weights w_i and a bias term b. A non-linear activation function f is then applied to the result z, yielding the output of the neuron $y = f(z)$. The activation function used in this work is the standard sigmoid function, i.e. $f(z) = 1/(1 + e^{-z})$. The aforementioned weights and biases are free parameters to be optimized. In addition, the outputs from each layer are input to the next layer. In this way, the input data propagates through the network, so that outputs from later layers become increasingly complex functions of the data. The first layer receives the input data and passes it to the subsequent layer, without performing any computation, while the final layer computes the final output of the network. Accordingly, we refer to these layers as the *input layer* and *the output layer*, respectively. The layers between the input and output layers are known as *hidden layers*. Note that we opt for the aforementioned architecture due to its success in accomplishing the intended objective, without necessitating the use of a more complex architecture, such as a recurrent neural network [43].

For the purpose of classifying the SD using ML techniques, let us suppose we have the time evolution of a family of system observables $\langle \hat{O}_j(t) \rangle$ (for a set of indices j) as input. These time signals can be gathered as

outcomes of an experiment carried out in a laboratory, or, as in our case, they can be generated by solving the system dynamics (either exactly or approximately).

As each signal is a time series, we Fourier-decompose the signal. To this end, we compute the Fourier coefficients as

$$X_k^j = \sum_{n=0}^{N-1} \langle \hat{O}_j(t_n) \rangle e^{-2\pi i k n / N}, \tag{9}$$

where N is the total number of time-steps and $\langle \hat{O}_j(t_n) \rangle$ denotes the nth sampled point. We can reconstruct the original signal by inverting equation (9), where, $X_k^j \in \mathbb{C}$, and the sum runs over all the sampled points in the time series, and $k \in [0, N-1]$. We split each coefficient X_k^j into its real and imaginary parts and train the network using the Fourier coefficients rather than the time series $\langle \hat{O}_j(t) \rangle$ directly. Using the resulting dataset, we address the ternary classification problem of distinguishing between three different families (i.e. *classes*) of SDs according to their value of the Ohmicity parameter [cf equation (7)]. In our case, the output layer of the NN has three artificial neurons which compute weighted sums z_j and apply the softmax activation function [44], defined as

$$f(z_j) = \frac{e^{z_j}}{\sum_{j=1}^{N_c} e^{z_j}}, \tag{10}$$

where N_c is the number of classes (in our case $N_c = 3$). It follows that the outputs of the network are the predicted probabilities that the input belongs to a particular class. As is common for classification problems, we use the categorical cross-entropy as a loss function. Given a dataset containing N_t trajectories, let y_{ij} represent the true probability that the ith trajectory belongs to the jth class and let $\hat{y}_{ij}$ denote the predicted probability of the same. Then the categorical cross-entropy is defined as [45]

$$L(\hat{y}, y) = -\frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_c} y_{ij} \log(\hat{y}_{ij}). \tag{11}$$

The task of training the network reduces to an optimization problem where the aim is to find the set of parameters that minimizes the loss function. A schematic view of the setup is shown in figure 2.

3. Generation of the dataset: spin-boson models

Given the general framework outlined in section 2, we now identify the systems to scrutinize. We focus on the dynamics of a Spin-Boson (SB) model consisting of a two-level system interacting with a bosonic bath. Therefore, in equation (1), we choose

$$\hat{H}_S = \frac{\omega_0}{2} \hat{\sigma}_z, \quad \hat{H}_B = \sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k \tag{12}$$

with $\hat{\sigma}_z$ being the z Pauli operator. The choice of the system–environment coupling Hamiltonian $\hat{H}_I$ leads to different physical scenarios, in general requiring different techniques to solve the dynamics. In section 3.1, we introduce an exactly solvable SB model, where the full system–environment unitary dynamics can be accessed, and the system dynamics is obtained by tracing out the environmental degrees of freedom. In section 3.2 we then choose a different form of coupling, which requires further approximations to effectively trace out the environment.

In both cases, the reduced dynamics of the system is governed by a master equation of the form

$$\dot{\hat{\rho}} = \mathcal{L}_t \hat{\rho}, \tag{13}$$

where $\mathcal{L}_t$ is the Liouvillian (super)-operator accounting for both the unitary and non-unitary dynamics, and $\hat{\rho}$ is the reduced density operator. Given the initial state of the system $\hat{\rho}(0) = \hat{\rho}^0$, equation (13) can be formally solved yielding $\hat{\rho} = \hat{\rho}(t) = e^{\mathcal{L}_t t} \hat{\rho}^0$ at any time t. It is thus immediate to obtain the expectation value of a generic observable $\hat{O}$, i.e. $\langle \hat{O}(t) \rangle \equiv \text{tr}_S(\hat{O}\hat{\rho}(t))$. Since we are considering a SB model, a natural choice of the observable would be given by the Pauli operators, i.e. $(\hat{O}_1, \hat{O}_2, \hat{O}_3) = (\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z)$ or a combination thereof.

3.1. Pure dephasing

Let us consider the case in which $\hat{X} = \hat{\sigma}_z$ in equation (2). Owing to this choice, the interaction Hamiltonian commutes with the system Hamiltonian and the populations of the reduced density matrix are left invariant by the dynamics. In this case, we can access the full unitary evolution, and exactly trace out the environmental degrees of freedom, thus yielding an analytical solution for the reduced dynamics [1, 46, 47]. In appendix B, we explicitly solve the dynamics under the standard assumption of an initially uncorrelated system–environment state, where we assume the environment to be in a thermal Gibbs state. Working in the interaction picture, the evolved reduced density matrix at time t can be written in the $\hat{\sigma}_z$ basis $\{|0\rangle, |1\rangle\}$ as

$$\hat{\rho}(t) = \begin{pmatrix} \rho_{00}^0 & \rho_{01}^0 e^{-\Gamma(t)} \\ \rho_{01}^{0*} e^{-\Gamma(t)} & 1 - \rho_{00}^0 \end{pmatrix}, \tag{14}$$

with the decoherence function

$$\Gamma(t) = 4 \int_0^\infty \mathrm{d}\omega J(\omega) \coth\left(\frac{\beta\omega}{2}\right) \frac{1 - \cos(\omega t)}{\omega^2}. \tag{15}$$

From equation (14) we can easily deduce that the interaction with the environment induces pure dephasing in the $\hat{\sigma}_z$ basis, with no dissipation (as deduced by comparing equation (15) with equation (5)). Moreover, it is worth emphasizing that there might be choices of the SD leading to negative values of $\Gamma(t)$. In such intervals of time, the system *re*-coheres as a result of (non-Markovian) memory effects of the dynamics [48].

3.2. Amplitude damping

Alternatively, we can turn to a set-up beyond pure dephasing, just by choosing $\hat{X} = -\hat{\sigma}_x/2$ in the interaction Hamiltonian of equation (2). Unlike the case discussed in section 3.1, the Hamiltonian does not exhibit any explicit symmetry, therefore we are not able to provide an exact solution for the dynamics. We can nevertheless effectively solve the dynamics, provided that we rely on further assumptions. Starting from an initial uncorrelated state, we can derive a master equation in the weak coupling regime, where we are still able to obtain non-Markovian effects. As outlined in the appendix C, we can derive a second-order approximated master equation that is local in time [1, 49, 50] and can be written in terms of dynamical equations for the components of the Bloch vector $\langle \vec{\sigma}(t) \rangle = \left(\langle \hat{\sigma}_x(t) \rangle, \langle \hat{\sigma}_y(t) \rangle, \langle \hat{\sigma}_z(t) \rangle \right)^{\mathrm{T}}$, with $\langle \hat{\sigma}_i \rangle = \mathrm{tr}_S(\hat{\sigma}_i \hat{\rho})$. These equations can be cast in the form

$$\frac{\mathrm{d}\langle \vec{\sigma}(t) \rangle}{\mathrm{d}t} = A(t)\langle \vec{\sigma}(t) \rangle + \vec{b}(t) \tag{16}$$

with $\vec{b}(t) = (0, 0, b_z(t))^T$ and $b_z(t) = \int_0^t \mathrm{d}s\, \mu(s) \sin(\omega_0 s)$. We have also introduced the matrix

$$A(t) = \begin{pmatrix} 0 & -\omega_0 & 0 \\ \omega_0 + a_{yx}(t) & a_{zz}(t) & 0 \\ 0 & 0 & a_{zz}(t) \end{pmatrix} \tag{17}$$

with the time-dependent entries

$$a_{yx}(t) = \int_0^t \mathrm{d}s\, \nu(s) \sin(\omega_0 s), \tag{18}$$

$$a_{zz}(t) = -\int_0^t \mathrm{d}s\, \nu(s) \cos(\omega_0 s). \tag{19}$$

The noise and dissipation kernels $\nu(t)$ and $\mu(t)$ are defined in equation (5).

4. Analysis and results

In this section, we present the results of our numerical experiments. We consider a two-level system, whose open dynamics depends on the choice of the coupling between the system and the bosonic environment, as discussed in section 3. For a given initial state, we generate a set of curves reproducing the time evolution of the expectation value of a system observable, i.e. $\langle \hat{O}(t) \rangle$. Each signal is sampled at N = 400 successive and equally spaced points over a certain time interval $[t_{\min}, t_{\max}]$, to ensure a sufficient resolution of the dynamics. As discussed in section 2, instead of directly using the time series, we input the 2N real and imaginary parts of the Fourier coefficients X_k. For this reason, we build the input layer with 2N input neurons. The NN for each model consists of the input layer followed by 2 hidden layers where the first hidden layer comprises 250

neurons, and the second comprises 80 neurons. The output layer, instead, is made of 3 neurons, which matches the number of classes (Ohmic, sub-Ohmic, super-Ohmic). The choice of network architecture was iteratively refined, adding layers and neurons until the network achieved a high accuracy without overfitting. The code employed for data generation, the datasets, and the code utilized for subsequent analysis are available in the following GitHub respository [51].

In order to evaluate the performance of the NN, we use the classification accuracy which is defined as the percentage of trajectories that are classified correctly. We generate a training dataset containing N_{Train} trajectories which is used to train the model, a validation dataset containing N_{Valid} trajectories which is used to assess the performance during training, and a test dataset containing N_{Test} trajectories which is used to assess the final accuracy of the network. We optimize the NNs using whole batch gradient descent and the Adam optimizer with a learning rate of 1×10^{-4}.
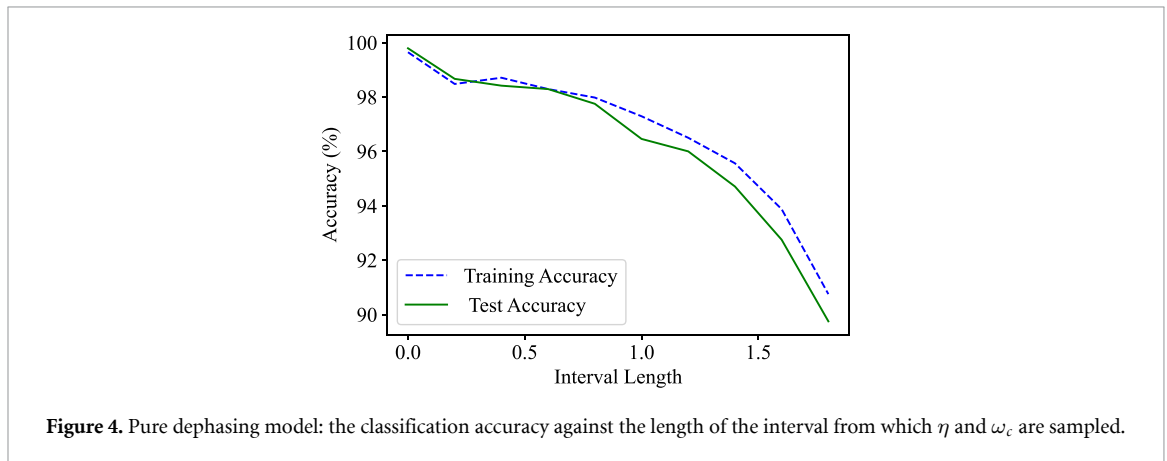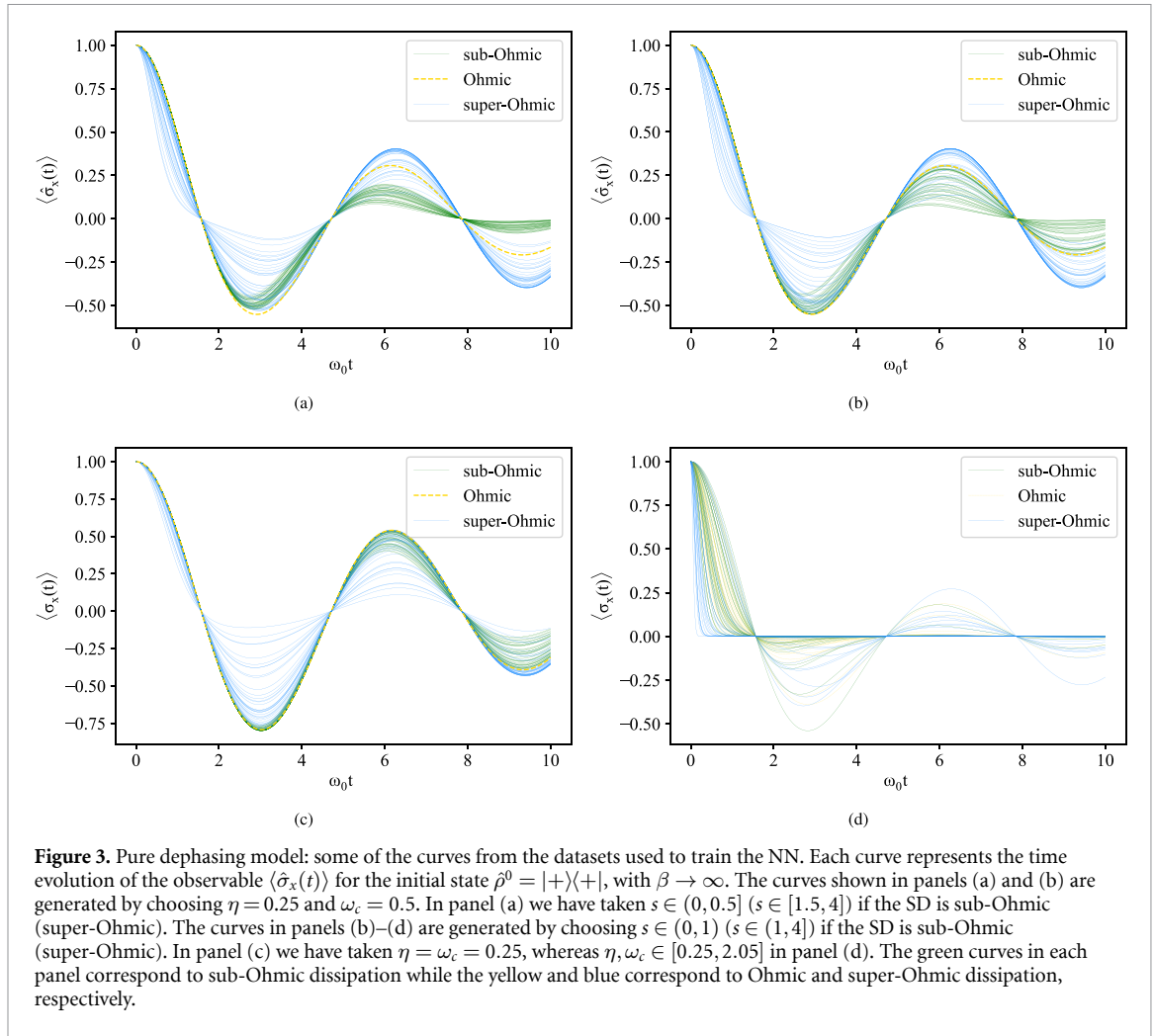
4.1. Pure dephasing

We consider the evolution of the pure dephasing model introduced in section 3.1. We solve the system dynamics choosing the initial state $\hat{\rho}^0 = |+\rangle\langle+|$, with $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, while—without loss of generality—we keep the thermal bath at zero temperature, i.e. $\beta \to \infty$. With this choice, we obtain the expectation value $\langle\hat{\sigma}_x(t)\rangle$ within the time interval $t \in [0, 10]$. It is worth noting that alternative choices for the initial state and the observable can be made, however, it should be recognized that, within the context of the pure dephasing model, the time evolution of $\langle\hat{\sigma}_z(t)\rangle$ will always be trivial. Moreover, should the initial state possess coherences equal to zero, the time evolution of the density matrix, and by extension any observables, will be trivial as well. As input to the NN we use the real and imaginary components of the Fourier coefficients obtained using equation (9). We generate a training, validation, and test set of size $N_{\text{Train}}/2 = N_{\text{Valid}} = N_{\text{Test}} = 2400$. The number of trajectories in the Ohmic, sub-Ohmic and super-Ohmic classes are equal in all datasets.

We assess the performance of the NN in two scenarios: the first being where ω_c and η are fixed, and the second being where they vary. In the first scenario, we consider the case where $\eta = 0.25$, $\omega_c = 0.5$, while the only parameter that varies is s. At first, we want to test how the model performs when the classes are easy to differentiate. To that end, we consider trajectories with $s \in (0, 0.5]$ if the SD is sub-Ohmic and $s \in [1.5, 4]$ if it is super-Ohmic. If the SD is Ohmic then s $= 1$. In figure 3(a) a subset of trajectories from the resulting training set are plotted where the green curves correspond to sub-Ohmic dissipation, while the yellow and blue curves are trajectories characterized by Ohmic and super-Ohmic dissipation, respectively. Given the substantial separation in the permissible values for s across the different classes, we expect that the performance of the NN will be high. In figure 3(a), it is evident that the classes are easily distinguishable due to distinct characteristics exhibited by each of them. Specifically, the super-Ohmic curves exhibit the steepest initial descent. In addition, while the sub-Ohmic and Ohmic curves show a comparable initial rate of descent, their oscillatory patterns differ. Oscillations are exhibited by all three classes, but the amplitude of oscillation for the sub-Ohmic curves appears to reduce more rapidly than that of the Ohmic or super-Ohmic curves as time grows. Confirming our expectation, the accuracy of the network evaluated on both the training and the test set reaches 100% after ≈80 training iterations.

We then make the task a bit more difficult for the network by allowing $s \in (0, 1)$ for the sub-Ohmic dissipation and $s \in (1, 4]$ for the super-Ohmic dissipation. We anticipate that the task will be more difficult in this scenario due to the reduced separation in the allowed values for s across the classes. This is reflected in the resulting training trajectories, a subset of which are plotted in figure 3(b), where we observe that the super-Ohmic curves maintain a more pronounced initial descent relative to the Ohmic and sub-Ohmic curves. However, there are instances where the oscillation amplitudes between the classes are similar. The final training accuracy of the network in this case reaches 99.31% after around 5000 training iterations, while the final test accuracy reaches 99.50%.

Next, to challenge the NN further, we consider the second scenario where we let η and ω_c vary: the idea is to assess the performance as we increase the upper bounds of the intervals from which they are sampled. We let $s \in (0, 1)$ for the sub-Ohmic spectral densities and $s \in (1, 4]$ for the super-Ohmic spectral densities. Initially, we set both η and ω_c equal to 0.25, then we let them vary into the interval $[0.25, 0.45]$. We increase the upper bound in increments of 0.2 until the interval becomes $[0.25, 2.05]$. Figure 3(c) shows some example trajectories from the training set for $\eta = \omega_c = 0.25$, while figure 3(d) shows some for $\eta, \omega_c \in [0.25, 2.05]$. From figure 3(c), we can observe that the scenario closely resembles that depicted in figure 3(b). In particular, the initial decay rates of the super-Ohmic curves are larger than those corresponding to the Ohmic or sub-Ohmic curves. However, the oscillation amplitudes across the three classes are comparable in some cases. In figure 3(d), it is evident that there is considerable overlap both in the initial decay rates and the amplitudes of oscillation between the three classes, indicating that the differences between the behaviors of the classes are less pronounced and that the classification task will be significantly

Figure 3. Pure dephasing model: some of the curves from the datasets used to train the NN. Each curve represents the time evolution of the observable $\langle\hat{\sigma}_x(t)\rangle$ for the initial state $\hat{\rho}^0 = |+\rangle\langle+|$, with $\beta \to \infty$. The curves shown in panels (a) and (b) are generated by choosing $\eta = 0.25$ and $\omega_c = 0.5$. In panel (a) we have taken $s \in (0, 0.5]$ ($s \in [1.5, 4]$) if the SD is sub-Ohmic (super-Ohmic). The curves in panels (b)–(d) are generated by choosing $s \in (0, 1)$ ($s \in (1, 4]$) if the SD is sub-Ohmic (super-Ohmic). In panel (c) we have taken $\eta = \omega_c = 0.25$, whereas $\eta, \omega_c \in [0.25, 2.05]$ in panel (d). The green curves in each panel correspond to sub-Ohmic dissipation while the yellow and blue correspond to Ohmic and super-Ohmic dissipation, respectively.

Figure 4. Pure dephasing model: the classification accuracy against the length of the interval from which η and ω_c are sampled.
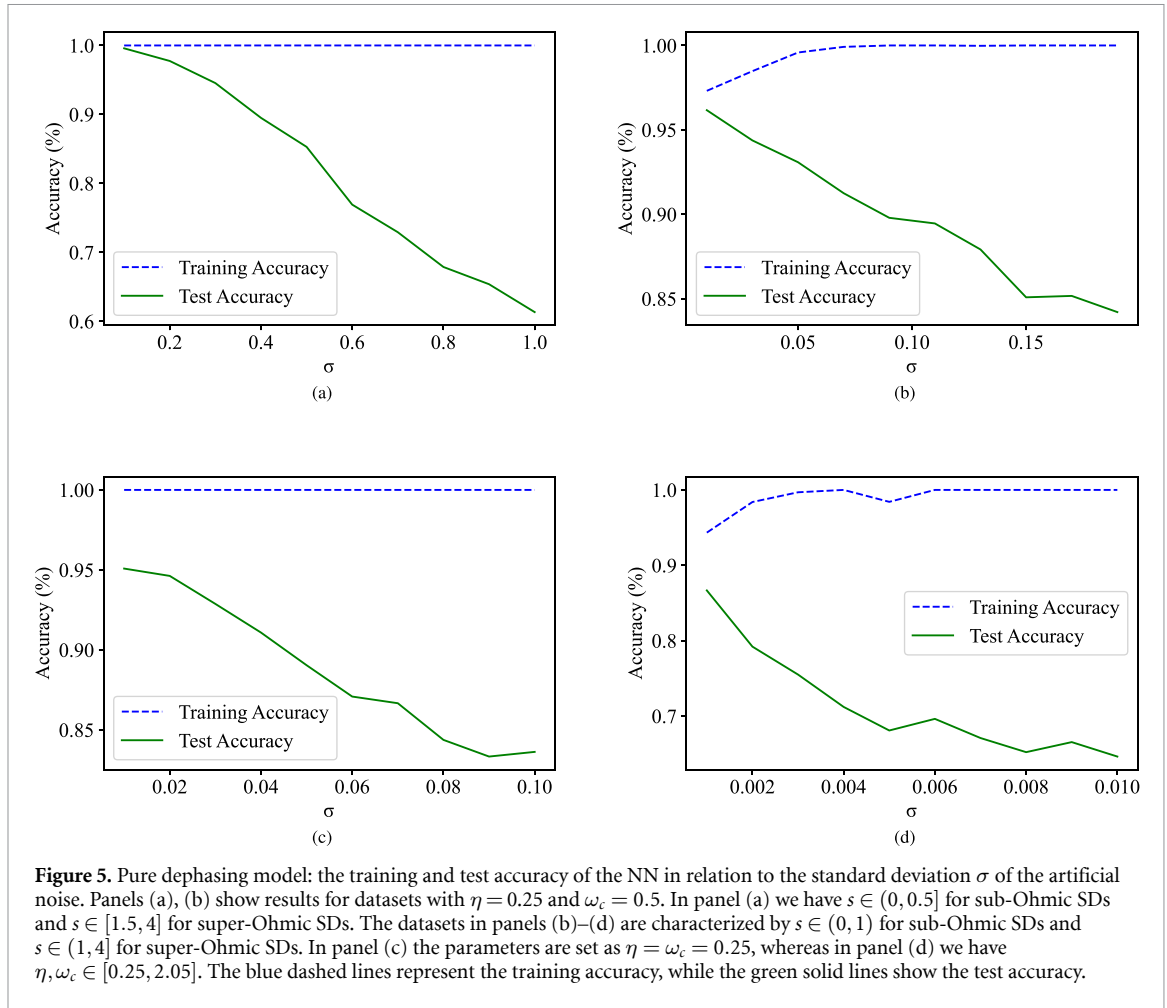
more difficult. The classification results after 2×10^4 training iterations are shown in figure 4 where the blue curve is the accuracy evaluated on the training set and the green curve is the accuracy evaluated on the test set. As expected, we can see that the accuracy decreases as we consider larger intervals η and ω_c. This is indeed the case, as taking larger intervals essentially increases the amount of noise in the dataset. It is worth noting that the accuracy may improve with larger datasets or more training iterations.

4.1.1. Measurement sampling noise

The accurate measurement and classification of experimental expectation values are inherently impacted by various sources of noise. One of the predominant ones is sampling noise, which arises due to the finite number of measurement samples that one can realistically acquire experimentally. In this section, we analyze the impact that sampling noise has on the NN, with the aim of providing a deeper insight into the

Figure 5. Pure dephasing model: the training and test accuracy of the NN in relation to the standard deviation σ of the artificial noise. Panels (a), (b) show results for datasets with $\eta = 0.25$ and $\omega_c = 0.5$. In panel (a) we have $s \in (0, 0.5]$ for sub-Ohmic SDs and $s \in [1.5, 4]$ for super-Ohmic SDs. The datasets in panels (b)–(d) are characterized by $s \in (0, 1)$ for sub-Ohmic SDs and $s \in (1, 4]$ for super-Ohmic SDs. In panel (c) the parameters are set as $\eta = \omega_c = 0.25$, whereas in panel (d) we have $\eta, \omega_c \in [0.25, 2.05]$. The blue dashed lines represent the training accuracy, while the green solid lines show the test accuracy.

performance of the model under realistic conditions. In our approach, we artificially introduce noise into the trajectories by adding a random value to each time point. Such value is drawn from a normal distribution with zero mean and a given standard deviation, σ.

We assess how the performance of the NN varies with σ in the same two scenarios as before: firstly, we hold both $\eta = 0.25$ and $\omega_c = 0.5$ constant. For the dataset with clear separation in the allowed values of s among classes [$s \in (0, 0.5)$ for a sub-Ohmic SD; $s \in [1.5, 4]$ for a super-Ohmic one], the results after 10^3 training iterations are shown in figure 5(a). The training accuracy remains consistently close to 100% for all of the considered σ values. This suggests that the NN can learn from the training data well, regardless of the magnitude of the noise that is introduced. However, the test accuracy decreases from 99.58% for $\sigma = 0.1$ to 61.33% for $\sigma = 1$, thus indicating that the capacity of the model to generalize to unseen data diminishes as the noise intensity increases.

For the dataset with $\eta = 0.25$, $\omega_c = 0.5$ and $s \in (0, 1)$—for a sub-Ohmic SD—and $s \in (1, 4]$—for a super-Ohmic one—the results after 10^3 training iterations are shown in figure 5(b). Mirroring the trends observed for the preceding dataset, the training accuracy remains notably high and close to 100% for the range of σ examined. On the other hand, the test accuracy starts at 96.17% for $\sigma = 0.01$ and decreases to 84.21% for $\sigma = 0.19$. Therefore, relative to the previously examined case, the NNs performance with this dataset exhibits a heightened susceptibility to noise.

We now redirect our attention to the case where η and ω_c vary. To begin with, we analyze the dataset corresponding to the shortest interval length in figure 4 [where $s \in (0, 1)$ if the SD is sub-Ohmic and $s \in (1, 4]$ if it is super-Ohmic] with $\eta = \omega_c = 0.25$. The results of this analysis after 10^4 training iterations, shown in figure 5(c), closely resemble those in figure 5(b), albeit with a noticeable decrease in performance. The training accuracy remains at 100% while the test accuracy starts at 95.08% for $\sigma = 0.01$ and decreases to 83.63% for $\sigma = 0.1$. Lastly, we turn our attention to the dataset corresponding to the longest interval length in figure 4. The results after 2×10^4 training iterations are shown in figure 5(d). While the conditions for s are consistent with those defined for the shortest interval length, η and ω_c vary into the interval $[0.25, 2.05]$. The training accuracy for this dataset starts at 94.33% for $\sigma = 0.001$ and exhibits minor fluctuations across the considered σ range but remains quite close to 100%. Meanwhile, the test accuracy begins at 86.67% for
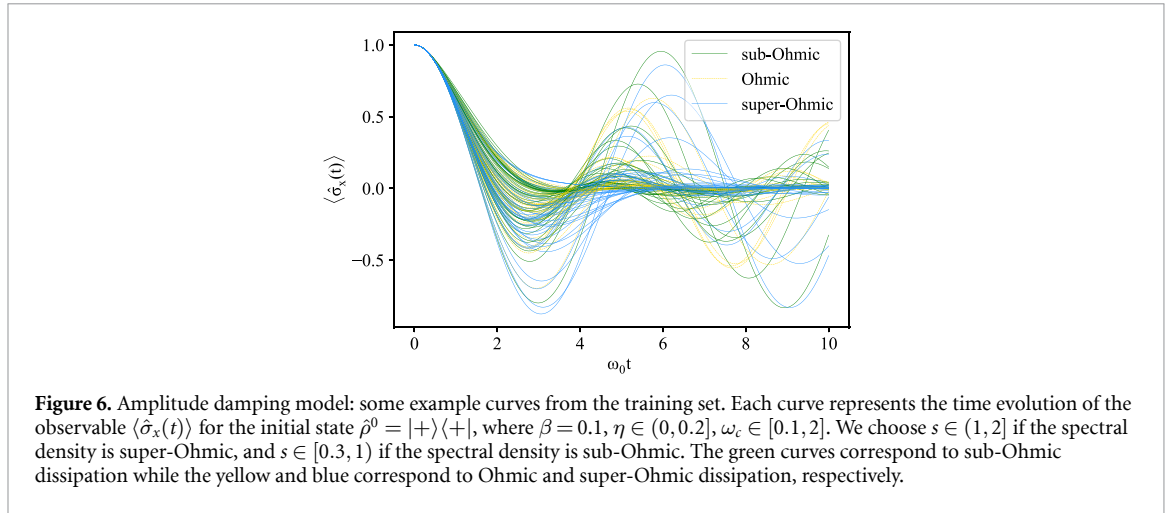
Figure 6. Amplitude damping model: some example curves from the training set. Each curve represents the time evolution of the observable $\langle \hat{\sigma}_x(t) \rangle$ for the initial state $\hat{\rho}^0 = |+\rangle\langle+|$, where $\beta = 0.1$, $\eta \in (0, 0.2]$, $\omega_c \in [0.1, 2]$. We choose $s \in (1, 2]$ if the spectral density is super-Ohmic, and $s \in [0.3, 1)$ if the spectral density is sub-Ohmic. The green curves correspond to sub-Ohmic dissipation while the yellow and blue correspond to Ohmic and super-Ohmic dissipation, respectively.

$\sigma = 0.001$ and drops to 64.63% at $\sigma = 0.01$. This dataset exhibits the greatest sensitivity to noise, leading to the lowest performance metrics. Moreover, the results emphasize the fact that despite the NNs ability to learn from training data, increasing noise levels hamper its generalization to previously unseen data.
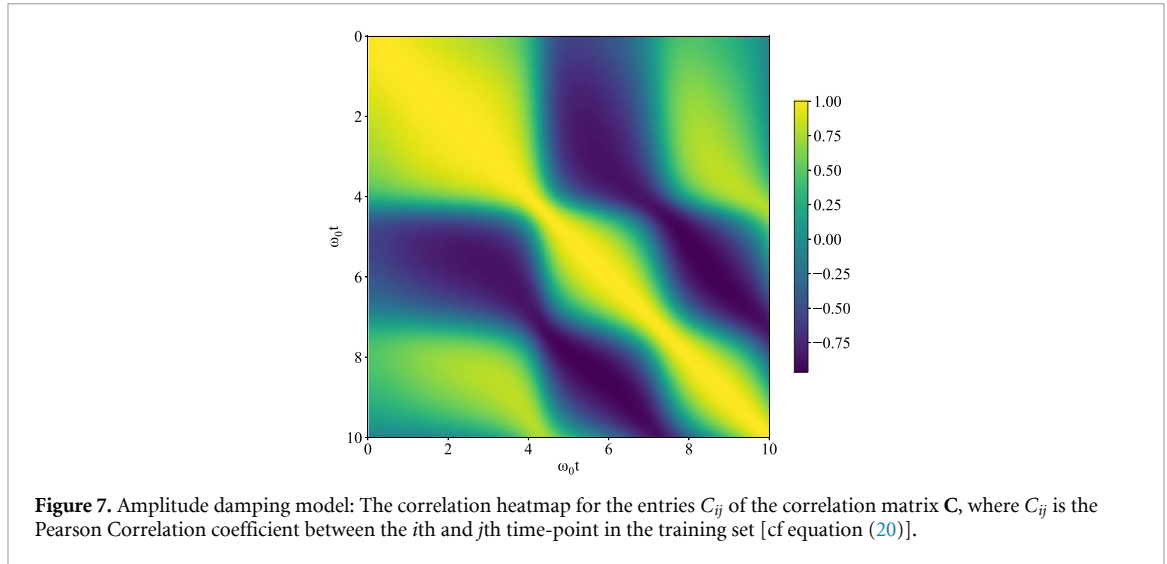
4.2. Amplitude damping

We shall now analyze the amplitude damping model detailed in section 3.2. We choose the initial state $\hat{\rho}^0 = |+\rangle\langle+|$, the bare frequency of the oscillator $\omega_0 = 1$, while we keep the environmental inverse temperature $\beta = 0.1$. We subsequently solve for the dynamics of the system and determine the expectation value $\langle \hat{\sigma}_x(t) \rangle$ within the time interval $t \in [0, 10]$. As for the previous model, we use the real and imaginary components of the Fourier coefficients obtained through equation (9) as input to the NN. We let $\eta \in (0, 0.2]$, $\omega_c \in [0.1, 2]$. In addition, we take $s \in (1, 2]$ [$s \in [0.3, 1)$] if the SD is super-Ohmic [sub-Ohmic] and $s = 1$ if the SD is Ohmic. We generate a training, validation, and test set such that $N_{\text{Train}} = 1500$, and $N_{\text{Valid}} = N_{\text{Test}} = 300$. In all datasets, the Ohmic, sub-Ohmic and super-Ohmic classes have an equal number of trajectories. Figure 6 shows some of the curves from the resulting training set where, as before, the green curves represent sub-Ohmic dissipation while the yellow and blue curves correspond to trajectories characterized by Ohmic and super-Ohmic dissipation, respectively. The final training accuracy of the network in this case reaches 97.93% after 10^4 training iterations while the test accuracy is significantly lower and reaches 93.00%.

Firstly, we would like to assess the number of time-points required to attain a high level of accuracy. It should be noted that it is generally advisable to avoid having highly correlated features in a dataset, whose linear dependence implies that the value of one can be derived from that of the other [44]. Hence, mutually correlated features convey redundant information to the model since each feature provides little or no additional information beyond what the other features already capture. Including all of the features will not improve the ability of the model to discriminate, but will increase the complexity of the algorithm, thus increasing the computational cost.

To this end, we introduce the Pearson correlation coefficient, which is a statistical measure of linear correlation between two variables [52, 53]. It ranges from a value of -1, indicating perfect anti-correlations, to 1, when the variables are perfectly correlated. A value of 0 indicates that there is no linear relationship between the two variables. Let $\langle \hat{\sigma}_x \rangle_n^i$ denote the nth time-point of the ith trajectory in a given dataset. Then the Pearson correlation coefficient between the nth and mth time-points, denoted as C_{nm}, is given by the formula

$$C_{nm} \equiv \frac{\sum_{i=1}^{N} \Delta \langle \hat{\sigma}_x \rangle_n^i \, \Delta \langle \hat{\sigma}_x \rangle_m^i}{\sqrt{\sum_{i=1}^{N} \left(\Delta \langle \hat{\sigma}_x \rangle_n^i \right)^2} \sqrt{\sum_{i=1}^{N} \left(\Delta \langle \hat{\sigma}_x \rangle_m^i \right)^2}}, \tag{20}$$

where $\Delta \langle \hat{\sigma}_x \rangle_j^i \equiv \langle \hat{\sigma}_x \rangle_j^i - \overline{\langle \hat{\sigma}_x \rangle}_j$, with $\overline{\langle \hat{\sigma}_x \rangle}_n$ the average value of the nth time step, and N the total number of trajectories in the dataset. We calculate the Pearson correlation coefficient between each time step in our training set and generate a correlation matrix, $\mathbf{C}$, whose entries quantify the correlation between time-points. The resulting correlation heatmap, a graphical representation of the correlation matrix, is shown in figure 7. From the heatmap, it can be observed that there is a high degree of correlation between adjacent and near-adjacent time points.

Figure 7. Amplitude damping model: The correlation heatmap for the entries C_{ij} of the correlation matrix **C**, where C_{ij} is the Pearson Correlation coefficient between the ith and jth time-point in the training set [cf equation (20)].

To address this issue, a common approach is to perform *feature selection*, identifying a subset of features that are the most informative and non-redundant. Retaining only one of two correlated features may expedite the learning process, without compromising the accuracy of the model. While we ideally want to avoid correlation between the features in a dataset, it is preferable to retain features which are correlated with the dependent variable [54]. Correlations make it possible to use the value of one variable to predict the value of another, meaning that features which are correlated with the output are predictive of the output.

Note that the Pearson correlation coefficient is only suitable for measuring the correlation between two continuous variables. As, in our case, the dependent variable consists of discrete labels, we can instead determine the degree of correlation between a feature and the dependent variable by examining whether the variance of the feature can be explained by the dependent variable. To do this, we group the feature into classes based on the discrete labels, compute the variance of each class, and calculate the difference between the mean of the resulting variances and the overall variance of the feature. If the mean of the class variances is significantly lower than the overall variance, this suggests that the feature and the dependent variable are correlated.

A possible strategy for performing feature selection and identifying the most salient features for learning is thus to sort the entries in the correlation matrix into descending order. Then, starting from the highest correlation, one can remove the contributing feature that exhibits the lowest correlation with the dependent variable. Using the above strategy, we can obtain a ranking of the features based on their importance and determine the order in which to remove features if we are to maintain a high classification accuracy. In this scenario, given that the time intervals between points may not be uniformly distributed, it becomes necessary to compute the Fourier coefficients using the non-uniform discrete Fourier transform [55]

$$X_k = \sum_{n=0}^{N-1} \langle \hat{\sigma}_x(t_n) \rangle e^{-2\pi i k p_n}, \tag{21}$$

where p_n are the non-uniform time points suitably scaled to fall between 0 and 1, while $\langle \hat{\sigma}_x(t_n) \rangle$ denotes the nth sampled point in a given trajectory. As for the discrete Fourier transform, k is the frequency which is an integer number between 0 and $N-1$. Note that if $p_n = n/N$, then this equation reduces to the discrete Fourier transform shown in equation (9).

We compare the results obtained using the proposed feature selection algorithm with the results obtained by selecting time points uniformly, i.e. choosing time points that are evenly spaced throughout the datasets. For example, we might select the first in every 5 points or the first in every 100. Figure 8 shows a plot of the test accuracy against the number of selected time points for the two different selection methods. The blue curve show the results obtained using uniform sampling, while the green curve shows the results obtained using the proposed feature selection algorithm. Firstly, the plot shows that the test accuracy remains consistently high until the number of time points is reduced to approximately 20. Beyond this point, a sharp decline in the accuracy is observed, as shown in the inset of figure 8. Analysis of the plot indicates that the performance of the two time point selection methods is comparable across different ranges of selected time points. Specifically, when we take a number of time points between ≈ 250 and ≈ 400, there is little difference between the accuracy obtained using uniform sampling and the feature selection algorithm. However, in the
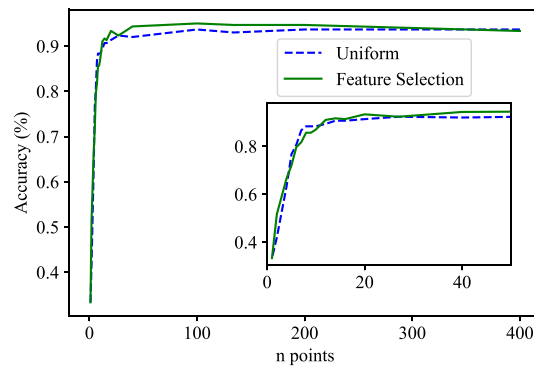
Figure 8. Amplitude damping model: the test accuracy of the NN against the number of time points when the time points are selected uniformly or using the proposed feature selection method described in the main text.
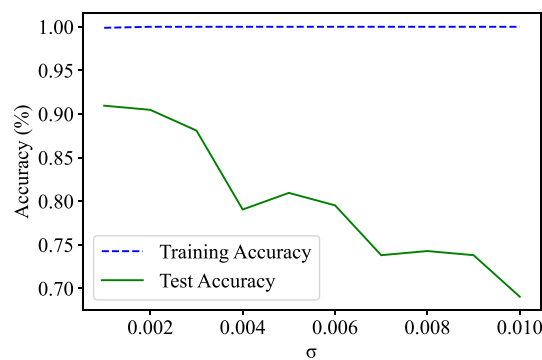
Figure 9. Amplitude damping model: The training and test accuracy of the NN against the standard deviation, σ, of the artificial noise.

range of approximately 40 to 250 time points, the feature selection algorithm shows slightly better results compared to uniform sampling. Lastly, taking less than $\approx$40 points, the test accuracy fluctuates, but we can conclude that the performance of both methods is similar.

For the sake of completeness, we also explored various other methods for feature selection. For instance, after grouping each feature according to the discrete labels, we used one-way-analysis-of-variance (ANOVA) to determine if there were statistically significant differences between the three groups [56]. We also considered the ratio of the mean of the variances of the groups and the overall variance, as opposed to the difference. Lastly, we attempted to assess the importance of each feature using principal component analysis. Specifically, we examined the degree to which each feature contributed to the principal components, as a large contribution to the principal components suggests that a feature is important in explaining the overall variability of the data [57]. We observed that none of the aforementioned methods outperformed the correlation based feature selection algorithm employed in figure 8.

4.2.1. Measurement sampling noise

In this section, similar to the analysis conducted for the pure dephasing model, we assess how the NN performs when subject to realistic conditions. Due to the presence of noise sources such as sampling noise, experimentally obtained expectation values are seldom completely accurate, as it is only feasible to collect a finite number of measurement samples experimentally. Given this, we aim to investigate how the performance of the NN is affected by these realistic challenges. To this end, we simulate the impact of sampling noise by incorporating artificial noise into the trajectories. We use the entire trajectory, consisting of 400 time points, add a random value drawn from a normal distribution with zero mean and a standard deviation σ to each point, and then assess the performance of the NN as σ increases.

The results of our analysis, after 10^4 training iterations are shown in figure 9, where the blue dashed line represents the accuracy evaluated on the training set, and the green solid line corresponds to the accuracy evaluated on the test set. We observe that the training accuracy remains consistently high, at around 100%. In contrast, the test accuracy starts off at 90.95% when $\sigma = 0.001$ and experiences a decline, dropping to 69.05% as the value of σ increases to 0.01. Consequently, our observations are consistent with those obtained for the

pure dephasing model. The model is able to effectively learn from the training data, and maintain a high training accuracy, regardless of the noise levels. However, its capacity to generalize to new, unseen data deteriorates.

5. Conclusions

We have shown that, in a standard open system scenario, a NN can perform SD-classification with high accuracy. First, we have considered an exactly solvable, pure-dephasing model, and assessed the performance of the NN as a classifier, highlighting the limiting role played by the fluctuations of the SD parameters. We have then considered a SB model that, under a number of reasonable approximations, results in a master equation accounting for energy losses and decoherence. We observed that, despite the approximations being invoked, the NN can perform the SD-classification task with high accuracy. Furthermore, we thoroughly discussed the interplay between high accuracy in the classification task and the number of sampled points for the system observable. Lastly, we investigated how the NN's performance for both models withstands the challenge of measurement sampling noise, thus providing insights into its robustness under realistic conditions.

 The methodology introduced in this paper, as well as the case studies analyzed therein, highlight the capability of ML techniques to characterize environments with arbitrary SDs, thus embodying a reliable tool for environment characterization and the provision of useful information for control and process diagnosis. This paves the way to, and leaves great hopes for, the full characterization of an unknown SD through, for instance, regression of the parameters rather than classification. We also stress that the method put forward here does not rely critically on how the information on the dynamics is specifically acquired. In this sense, we expect the method to maintain effectiveness even when considering classes of SDs leading to long-lived correlations that, in turn, would hinder the direct derivation of master equations in Lindblad-like form. In such cases, one should rely on more sophisticated simulation techniques—such as Hierarchical equation Of Motion (HEOM) [58, 59], Time-Evolving Matrix Product Operators (TEMPO) [60], or Time-Evolving Density with Orthogonal Polynomials Algorithm (TEDOPA) [61–63], just to name a few. The combinations of one of these methods with the ML will help achieving the successful characterization and control of the environment affecting a given open system.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/Jessica-Barr/SD-Classification.

Appendix A. The correlation function of a Bosonic bath

Here we explicitly derive the correlation function for an arbitrary quantum system that is interacting with an environment which is made up of infinitely many independent harmonic oscillators, i.e. equations (4) and (5) of the main text. Given an interaction Hamiltonian in the form of equation (2) and the bath operator $\hat{B}$ given by equation (3), we can compute the correlation function which is defined as

$$\alpha_\beta(t) = \langle \hat{B}(t)\hat{B}(0) \rangle_B = \mathrm{tr}_B\left(\hat{B}(t)\hat{B}(0)\hat{\rho}_B\right). \tag{A1}$$

We now move to the interaction picture via the relation $\hat{B}(t) = e^{it\hat{H}_B}\hat{B}e^{-it\hat{H}_B}$, where $\hat{H}_B = \sum_k \omega_k \hat{b}_k^\dagger \hat{b}_k$ is the Hamiltonian of a set of independent harmonic oscillators. Therefore, we have

$$\hat{B}(t) = \sum_k \left(g_k \hat{b}_k^\dagger e^{i\omega_k t} + g_k^* \hat{b}_k e^{-i\omega_k t}\right), \tag{A2}$$

$$\hat{B}(0) = \sum_k \left(g_k \hat{b}_k^\dagger + g_k^* \hat{b}_k \right).$$ (A3)

Thus, the expression for the correlation function reads

$$\langle \hat{B}(t)\hat{B}(0) \rangle_B = \sum_k |g_k|^2 \left(\langle \hat{b}_k^\dagger \hat{b}_k \rangle_B e^{i\omega_k t} + \langle \hat{b}_k \hat{b}_k^\dagger \rangle_B e^{-i\omega_k t} \right),$$ (A4)

where we have utilized the fact that $\langle \hat{b}_k \hat{b}_l \rangle_B = \langle \hat{b}_k^\dagger \hat{b}_l^\dagger \rangle_B = 0$ and that $\langle \hat{b}_k \hat{b}_l^\dagger \rangle_B$ and $\langle \hat{b}_k^\dagger \hat{b}_l \rangle_B$ are non-zero if and only if $k = l$. If we further assume that the environment is thermal equilibrium at a temperature T, then $\hat{\rho}_B$ is represented by a thermal Gibbs state of the form

$$\hat{\rho}_B = \frac{e^{-\beta \hat{H}_B}}{\mathcal{Z}_B},$$ (A5)

where $\mathcal{Z}_B$ is the reservoir partition function. As a result, we find that the quantity $\langle \hat{b}_k^\dagger \hat{b}_k \rangle_B = N_k = (e^{\beta\omega_k} - 1)^{-1}$ is the mean occupation number of the kth mode of the environment. Finally, assuming that the bath modes form a continuum, we obtain the following expression for the correlation function:

$$\alpha_\beta(t) = \int_0^\infty d\omega\, J(\omega) \left[\coth\left(\frac{\beta\omega}{2} \right) \cos(\omega t) - i \sin(\omega t) \right]$$ (A6)

which can be recast in the form of equations (4) and (5).

Appendix B. SB model (pure dephasing)

We now derive the equations governing the dynamics of the system described in section 3.1. We work in the interaction picture and begin by deriving an expression for the unitary evolution operator $\hat{U}(t)$ which acts on the composite system. Let us first notice that the two-time commutator of the interaction Hamiltonian is non-zero, i.e.

$$\left[\hat{H}_I(t), \hat{H}_I(t') \right] = -2i\, \mathbb{1}_S \otimes \sum_k |g_k|^2 \sin(\omega_k(t - t')),$$ (B1)

where $\mathbb{1}_S$ is the identity operator acting on the system only. The latter is useful to evaluate the time evolution operators as

$$\hat{U}(t) = \mathcal{T}_\leftarrow \exp\left[-i \int_0^t \hat{H}_I(\tau)\, d\tau \right],$$ (B2)

where $\mathcal{T}_\leftarrow$ denotes the time ordering operator. Following the ideas in [64] (see also [65]), we can formally discretize the integral in the exponent of the unitary evolution operator and denote $\mathcal{H}_n = -i\hat{H}_I(n dt)$, where $dt = t/N$. Taking the limit as $N \to \infty$ we obtain

$$\hat{U}(t) = \mathcal{T}_\leftarrow \lim_{dt \to 0} \exp\left[\sum_{n=1}^N \mathcal{H}_n\, dt \right].$$ (B3)

We use a generalization of the Baker–Campbell–Hausdorff formula to calculate the exponential

$$e^{\sum_{n=1}^N \mathcal{H}_n} = \left(\prod_{n=1}^N e^{\mathcal{H}_n} \right) \left(\prod_{n<m} e^{-\frac{1}{2}[\mathcal{H}_n, \mathcal{H}_m]} \right),$$ (B4)

which holds since the second order commutators vanish. The unitary evolution operator becomes

$$\tilde{U}(t) = \lim_{dt \to 0} \prod_{n<m} e^{-\frac{1}{2}[\mathcal{H}_n, \mathcal{H}_m](dt)^2} \prod_n e^{\mathcal{H}_n dt},$$ (B5)

where we have noticed that the commutator in the first exponent is just a complex number, so we may omit the time ordering operator. Recombining the exponentials of the operators we find

$$\tilde{U}(t) = \lim_{dt \to 0} e^{-\frac{1}{2}\sum_{n<m}[\mathcal{H}_n, \mathcal{H}_m](dt)^2} e^{\sum_n \mathcal{H}_n dt} = e^{\frac{1}{2}\int_0^t dt_1 \int_0^{t_1} dt_2 [\hat{H}_I(t_2), \hat{H}_I(t_1)]} e^{-i \int_0^t \hat{H}_I(\tau) d\tau},$$ (B6)

where the first exponent—as a consequence of equation (B1)—only applies a global phase to the qubit. As a result, the dynamics of the system are solely governed by the operator

$$e^{-i\int_0^t \hat{H}_I(\tau)d\tau} = e^{\hat{\sigma}_z \otimes \sum_k \left(\alpha_k(t)\hat{b}_k^\dagger - \alpha_k^*(t)\hat{b}_k\right)} \equiv e^{\hat{\sigma}_z \otimes \hat{A}(t)}, \tag{B7}$$

with $\alpha_k(t) = g_k \left(1 - e^{i\omega_k t}\right)/\omega_k$. It is convenient to rewrite this operator in the form:

$$e^{\hat{\sigma}_z \otimes \hat{A}(t)} = I \otimes \sum_{n=0}^\infty \frac{\hat{A}(t)^{2n}}{2n!} + \hat{\sigma}_z \otimes \sum_{n=1}^\infty \frac{\hat{A}(t)^{2n+1}}{(2n+1)!} \tag{B8}$$

$$= I \otimes \cosh\left(\hat{A}(t)\right) + \hat{\sigma}_z \otimes \sinh\left(\hat{A}(t)\right). \tag{B9}$$

The matrix elements of the reduced density matrix are determined by explicitly tracing out the environmental degrees of freedom, i.e.

$$\hat{\rho}_{ij}(t) = \langle i | \operatorname{tr}_B \left\{ \hat{U}(t) \hat{\rho}^0 \otimes \hat{\rho}_B \hat{U}^\dagger(t) \right\} | j \rangle. \tag{B10}$$

It follows that the coherences of the reduced density matrix evolve as

$$\hat{\rho}_{01}(t) = \hat{\rho}_{01}^0 \left\langle e^{2\hat{A}(t)} \right\rangle, \tag{B11}$$

with $\hat{\rho}_{10}(t) = \hat{\rho}_{01}^*(t)$. Resorting to the identity $\langle e^{\hat{A}} \rangle = e^{\langle \hat{A} \rangle^2/2}$, where the operator $\hat{A}$ is a linear combination of creation and annihilation operators [66], we find that

$$\left\langle e^{2\hat{A}(t)} \right\rangle = e^{-2\sum_k |\alpha_k(t)|^2 \langle b_k b_k^\dagger + b_k^\dagger b_k \rangle} = e^{-2\sum_k |\alpha_k(t)|^2 (2N_k+1)}. \tag{B12}$$

Finally, substituting the expressions for α_k and the mean occupation number of the kth mode of the environment, N_k, we obtain

$$\left\langle e^{2\hat{A}(t)} \right\rangle = e^{-\Gamma(t)}, \tag{B13}$$

where we have assumed the bath modes form a continuum. The function $\Gamma(t)$ is the decoherence function given in equation (15) of the main text.

Appendix C. SB model (amplitude damping)

Here, we derive the equations governing the dynamics of the system described in section 3.2. The second-order generator of the TCL master equation leads to the following equation for the reduced density matrix in the interaction picture $\tilde{\rho}$ [1, 2]:

$$\frac{d\tilde{\rho}}{dt} = -\int_0^t ds \operatorname{tr}_B \left[\hat{H}_I(t), \left[\hat{H}_I(s), \tilde{\rho} \otimes \hat{\rho}_B \right] \right], \tag{C1}$$

where $\tilde{\rho}$ and $\hat{H}_I(t) = -\hat{\sigma}_x(t) \otimes \hat{B}(t)/2$ are expressed in the interaction picture with respect to the free Hamiltonian H_S. The form of the bath operator $\hat{B}(t)$ is given by equation (A2). By explicitly performing the calculations, changing the integration variable as $s \to t - s$, and moving to the Schrödinger picture, we are able to rewrite such master equation as

$$\frac{d\hat{\rho}}{dt} = -i\left[\hat{H}_S, \hat{\rho}\right] - \frac{1}{4}\int_0^t ds \left(\nu(s)\left[\hat{\sigma}_x, \left[\hat{\sigma}_x(-s), \hat{\rho}\right]\right] + i\eta(s)\left[\hat{\sigma}_x, \{\hat{\sigma}_x(-s), \hat{\rho}\}\right]\right), \tag{C2}$$

where $\nu(s)$ and $\eta(s)$ are respectively the real and imaginary parts of the correlation function given in equation (5) of the main text. The corresponding dynamical equations for the components of the Bloch vector $\langle \sigma_j(t) \rangle = \operatorname{tr}_S\left[\sigma_j \rho(t)\right]$ read

$$\frac{d\langle \hat{\sigma}_x(t) \rangle}{dt} = -\omega_0 \langle \hat{\sigma}_y(t) \rangle, \tag{C3}$$

$$\frac{d\langle \hat{\sigma}_y(t) \rangle}{dt} = \left(\omega_0 + a_{yx}(t)\right) \langle \hat{\sigma}_x(t) \rangle + a_{yy}(t) \langle \hat{\sigma}_y(t) \rangle, \tag{C4}$$

$$\frac{d\langle \hat{\sigma}_z(t) \rangle}{dt} = a_{zz}(t) \langle \hat{\sigma}_z(t) \rangle + b_z(t), \tag{C5}$$

where the time-dependent coefficients are defined in the main text [cf equations (18) and (19)]. This set of coupled differential equations can be recast in the matrix form of equations (16) and (17).

ORCID iDs

J Barr ⦿ https://orcid.org/0009-0007-2949-0567
G Zicari ⦿ https://orcid.org/0000-0002-2526-9431
A Ferraro ⦿ https://orcid.org/0000-0002-7579-6336
M Paternostro ⦿ https://orcid.org/0000-0001-8870-9134

References

[1] Breuer H-P and Petruccione F 2002 *The Theory of Open Quantum Systems* (Oxford University Press)
[2] Rivas A and Huelga S F 2012 *Open Quantum Systems: An Introduction* (*Springerbriefs in Physics*) (Springer)
[3] Zurek W H 2003 *Rev. Mod. Phys.* **75** 715
[4] Braun D, Haake F and Strunz W T 2001 *Phys. Rev. Lett.* **86** 2913
[5] Strunz W T and Haake F 2003 *Phys. Rev. A* **67** 022102
[6] Unruh W G 1995 *Phys. Rev. A* **51** 992
[7] Chuang I L, Laflamme R, Shor P W and Zurek W H 1995 *Science* **270** 1633
[8] Krenn M, Landgraf J, Foesel T and Marquardt F 2023 *Phys. Rev. A* **107** 010101
[9] Palmieri A M, Kovlakov E, Bianchi F, Yudin D, Straupe S, Biamonte J D and Kulik S 2020 *npj Quantum Inf.* **6** 20
[10] Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R and Carleo G 2018 *Nat. Phys.* **14** 447
[11] Banchi L, Grant E, Rocchetto A and Severini S 2018 *New J. Phys.* **20** 123030
[12] Kardashin A S, Vlasova A V, Pervishko A A, Yudin D and Biamonte J D 2022 *Phys. Rev. A* **106** 032409
[13] Carleo G and Troyer M 2017 *Science* **355** 602
[14] Luo D, Chen Z, Carrasquilla J and Clark B K 2022 *Phys. Rev. Lett.* **128** 090501
[15] Bandyopadhyay S, Huang Z, Sun K and Zhao Y 2018 *Chem. Phys.* **515** 272
[16] Luchnikov I A, Vintskevich S V, Ouerdane H and Filippov S N 2019 *Phys. Rev. Lett.* **122** 160401
[17] Giannelli L, Sgroi P, Brown J, Paraoanu G S, Paternostro M, Paladino E and Falci G 2022 *Phys. Lett. A* **434** 128054
[18] Niu M Y, Boixo S, Smelyanskiy V N and Neven H 2019 *npj Quantum Inf.* **5** 33
[19] Sgroi P, Palma G M and Paternostro M 2021 *Phys. Rev. Lett.* **126** 020601
[20] Zeng Y-X, Gebremariam T, Shen J, Xiong B and Li C 2021 *Appl. Phys. Lett.* **118** 164003
[21] Guo Y, Hu Z, Zhang J, Zhu C and Guo X 2023 *Appl. Phys. Lett.* **123** 051101
[22] Mascherpa F, Smirne A, Huelga S F and Plenio M B 2017 *Phys. Rev. Lett.* **118** 100401
[23] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* 10th Anniversary edn (Cambridge University Press)
[24] Benenti G, Casati G, Rossini D and Strini G 2018 *Principles of Quantum Computation and Information* 2nd edn (World Scientific)
[25] Jagadish V and Petruccione F 2018 *Quanta* **7** 54
[26] Gardiner C 2009 *Handbook of Stochastic Methods* (*Springer Series in Synergetics* vol 13) 4th edn (Springer)
[27] Chruściński D 2022 *Phys. Rep.* **992** 1
[28] Rivas A, Huelga S F and Plenio M B 2014 *Rep. Prog. Phys.* **77** 094001
[29] Breuer H-P, Laine E-M, Piilo J and Vacchini B 2016 *Rev. Mod. Phys.* **88** 021002
[30] Wise D F, Morton J J and Dhomkar S 2021 *PRX Quantum* **2** 010316
[31] Youssry A, Paz-Silva G A and Ferrie C 2020 *npj Quantum Inf.* **6** 95
[32] Martina S, Hernández-Gómez S, Gherardini S, Caruso F and Fabbri N 2023 *Mach. Learn.: Sci. Technol.* **4** 02LT01
[33] Papič M and de Vega I 2022 *Phys. Rev. A* **105** 022605
[34] Martina S, Gherardini S and Caruso F 2023 *Phys. Scr.* **98** 035104
[35] Garau Estarellas G, Giorgi G L, Soriano M C and Zambrini R 2019 *Adv. Quantum Technol.* **2** 1800085
[36] Palmieri A M, Bianchi F, Paris M G A and Benedetti C 2021 *Phys. Rev. A* **104** 052412
[37] Caldeira A and Leggett A 1983 *Ann. Phys., NY* **149** 374
[38] Leggett A J, Chakravarty S, Dorsey A T, Fisher M P A, Garg A and Zwerger W 1987 *Rev. Mod. Phys.* **59** 1
[39] Rivas A, Plato A D K, Huelga S F and Plenio M B 2010 *New J. Phys.* **12** 113032
[40] Pucci L, Esposito M and Peliti L 2013 *J. Stat. Mech.* 04005
[41] de Vega I and Alonso D 2017 *Rev. Mod. Phys.* **89** 015001
[42] Weiss U 2012 *Quantum Dissipative Systems* 4th edn (World Scientific)
[43] Marquardt F 2021 *SciPost Phys. Lect. Notes* 29
[44] Hastie T, Tibshirani R, Friedman J H and Friedman J H 2009 *The Elements of Statistical Learning* 2nd edn (Springer)
[45] Murphy K P 2012 *Machine Learning: A Probabilistic Perspective* (MIT press)
[46] Palma G M, Suominen K-a and Ekert A 1996 *Proc. R. Soc. A* **452** 567
[47] Guarnieri G, Smirne A and Vacchini B 2014 *Phys. Rev. A* **90** 022110
[48] Addis C, Bylicka B, Chruściński D and Maniscalco S 2014 *Phys. Rev. A* **90** 052103
[49] Breuer H-P, Kappler B and Petruccione F 2001 *Ann. Phys., NY* **291** 36
[50] Clos G and Breuer H-P 2012 *Phys. Rev. A* **86** 012115
[51] Barr J 2023 SD-Classification (available at: https://github.com/Jessica-Barr/SD-Classification)
[52] Wilcox R R 2010 *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy* vol 249 (Springer)
[53] James G, Witten D, Hastie T and Tibshirani R 2013 *An Introduction to Statistical Learning* vol 112 (Springer)
[54] Liu H and Motoda H 2012 *Feature Selection for Knowledge Discovery and Data Mining* vol 454 (Springer Science & Business Media)
[55] Greengard L and Lee J-Y 2004 *SIAM Rev.* **46** 443
[56] Kyun K T 2017 *Korean J. Anesthesiol.* **70** 22
[57] Johnson R A and Wichern D W *et al* 2002 *Applied Multivariate Statistical Analysis* (Prentice Hall)
[58] Tanimura Y and Kubo R 1989 *J. Phys. Soc. Japan* **58** 101
[59] Tanimura Y 1990 *Phys. Rev. A* **41** 6676
[60] Strathearn A, Kirton P, Kilda D, Keeling J and Lovett B W 2018 *Nat. Commun.* **9** 3322

[61] Chin A W, Rivas A, Huelga S F and Plenio M B 2010 *J. Math. Phys.* **51** 092109
[62] Prior J, Chin A W, Huelga S F and Plenio M B 2010 *Phys. Rev. Lett.* **105** 050404
[63] Tamascelli D, Smirne A, Lim J, Huelga S F and Plenio M B 2019 *Phys. Rev. Lett.* **123** 090402
[64] Lidar D A, Bihary Z and Whaley K 2001 *Chem. Phys.* **268** 35
[65] Schaller G 2014 *Open Quantum Systems Far From Equilibrium* (*Lecture Notes in Physics* vol 881) (Springer)
[66] Giuliani G and Vignale G 2005 *Quantum Theory of the Electron Liquid* (Cambridge University Press)