Physica Scripta

PAPER

Machine learning classification of non-Markovian noise disturbing quantum dynamics

Stefano Martina[1,*], **Stefano Gherardini**[1,2,3] and **Filippo Caruso**[1,2]

[1] Dept.of Physics and Astronomy, University of Florence, via G. Sansone 1, 50019 Sesto Fiorentino, Italy
[2] EuropeanLaboratory for Non-Linear Spectroscopy (LENS), University of Florence, via Nello Carrara 1, 50019 Sesto Fiorentino, Italy
[3] CNR-INO, Area Science Park, Basovizza, I-34149 Trieste, Italy
[*] Author to whom any correspondence should be addressed.

E-mail: stefano.martina@unifi.it, gherardini@lens.unifi.it and filippo.caruso@unifi.it

OPEN ACCESS

RECEIVED
17 October 2022

REVISED
18 November 2022

ACCEPTED FOR PUBLICATION
16 January 2023

PUBLISHED
15 February 2023

Abstract

In this paper machine learning and artificial neural network models are proposed for the classification of external noise sources affecting a given quantum dynamics. For this purpose, we train and then validate *support vector machine*, *multi-layer perceptron* and *recurrent neural network* models with different complexity and accuracy, to solve supervised binary classification problems. As a result, we demonstrate the high efficacy of such tools in classifying noisy quantum dynamics using simulated data sets from different realizations of the quantum system dynamics. In addition, we show that for a successful classification one just needs to measure, in a sequence of discrete time instants, the probabilities that the analysed quantum system is in one of the allowed positions or energy configurations. Albeit the training of machine learning models is here performed on synthetic data, our approach is expected to find application in experimental schemes, as e.g. for the noise benchmarking of noisy intermediate-scale quantum devices.

1. Introduction

Noise sensing aims at discriminating, and possibly reconstructing, noise profiles that affect static parameters and dynamical variables governing the evolution of classical and quantum systems [1–4]. In the quantum regime, which constitute the main object of our discussion, noise partially destroys the coherent evolution of the investigated open quantum system, interacting with an external environment or simpler with other systems [5, 6]. In such scenario, noise can be generally modelled as a stochastic process, distributed according to an unknown probability distribution [7, 8]. As concrete examples, one may consider the following cases that have recently studied experimentally: (i) Resonant microwave fields with random amplitude and phase for the driving of atomic transitions [9]; (ii) solid-state spin qubits in negatively charged nitrogen-vacancy (NV) centers that are naturally affected by a carbon nuclear spin environment [10]; (iii) single photons undergoing random polarisation fluctuations [11, 12]. In all these experiments, noise stochastic fields sampled from an unknown probability distribution have to be included in the microscopic derivation of the system dynamics under investigation, in order to properly carry out noise sensing and discrimination.

Several techniques, at both the theoretical and experimental side, have been developed for the inference of the unknown noise distribution and to detect, if present, non-zero time-correlations among adjacent samples (over time) of the noise process [9, 10, 13–22]. However, most of them suffer of the need to control the quantum system, by generating multiple control sequences (e.g., dynamical decoupling ones [23–25]), each of them being sensitive to a different component of the noise spectrum [26, 27]. In this regard, in Ref. [28] a diagnostic protocol for the detection of correlations among arbitrary sets of qubits have been tested on a 14-qubit superconducting quantum architecture, by discovering the persistent presence of long-range two-qubit correlations. Moreover, Machine Learning (ML)-models have been also adopted to study non-Markovian open quantum dynamics [29–31]. In particular, in [29] a method is developed to learn the effective Markovian embedding of a non-Markovian process. The embedding is learned by maximising the likelihood function built over successively

observed measurements of the quantum dynamics. The assumption in [29] is that the underlying time-evolution of the system is non-Markovian and the focus of the work is the training of the Markovian embedding. Thus, it is not directly addressed the issue of discriminating the presence of noise sources affecting the dynamics, nor if the noise samples over time are time-correlated. Instead, in [30] a Support Vector Machine (SVM) model is trained to predict the degree of non-Markovianity in open quantum systems. An open quantum system approach is thus employed, but without providing emphasis on quantum dynamics perturbed by a stochastic process of noise, nor on the use of more complex ML-models as neural networks and Recurrent Neural Network (RNN). In Ref. [31] a deep neural network approach is adopted to perform (at the theoretical level) noise regression of qubits immersed in their environment that entails different stationary, Gaussian noise spectra. In [31], deep neural networks are trained with time-dependent coherence decay (echo) curves used as input data.

In this paper, differently to all the aforementioned references, we exploit ML techniques [32, 33] to efficiently carry out high accuracy classification of noise affecting quantum dynamics. The proposed methods are designed to distinguish between Independent and Identically Distributed (i.i.d.) noise sequences and noise samples originated by a non-trivial memory kernel, thus characterised by specific time-correlation parameters. It is worth reminding that, in the latter case, the dynamics of the stochastic quantum system (stochastic due to the presence of fluctuating parameters, e.g. in the Hamiltonian of the analyzed system as in [9]) turns out of being non-Markovian [34, 35], in the sense that samples of its state in different time instants are correlated [36]. This entails that the propagation of the system in subsequent time intervals is highly influenced by its previous states, even occurring in the early stages of the dynamics [37–39]. This effect corresponds to a two-fold exchange of information between the system and the external sources, which has thus applications for quantum sensing [40, 41].

To present our novel approach and demonstrate its efficacy in discriminating Markovian and non-Markovian noise sources, we focus on the dynamics of a quantum particle randomly moving on a graph $\mathcal{G}$ [42–44], as generated by a stochastic Schrödinger equation. Depending on the way the particle is affected by the external noise, the noise-dependent component of its movements within the graph may be time-correlated. In this general context, we are going to propose ML-based solutions for the classification of characteristic noise features. Specifically, by training a properly-designed Machine Learning model via the probabilities that the particle is in each node of the graph $\mathcal{G}$ at discrete time instants (thus, no coherence decay curves need to be measured as in [31]), we will show that it is possible to discriminate accurately between different noise sources and identify the possible presence of time-correlations from observation of the quantum system dynamics.

To perform noise classification, SVMs, Multi-Layer Perceptrons (MLPs) and RNNs [45–48] are successfully trained on six data sets (each of them composed of 20 000 realisations) that have been properly generated to carry out binary classification of noisy quantum dynamics. Once trained, the proposed ML-models are able to reach a classification accuracy (defined by the number of correctly classified realisations over their total number) up to 97%. A pictorial representation of the proposed ML procedure is depicted in figure 1. We share the source codes used to generate the data and to train the ML-models for our results [Q2].

As other existing sensing techniques, the training of our ML-models can be performed preliminary on synthetic data. Specifically, synthetic data are generated by solving a stochastic Schrödinger equation—modeling the noisy quantum dynamics we are analyzing—that exhibits at least one random parameter to be randomly sampled. As a result, we have observed that both i.i.d. and correlated noise sources can be accurately discriminated by means of one single ML architecture. Moreover, our ML-based approach allows for non-Markovian noise classification by processing only measurements of the diagonal elements (even called 'populations') of the density operator ρ_t associated with the quantum system under investigation. Thus, no measurements of the off-diagonal elements of ρ_t, stemming from quantum coherence terms in a given basis of interest, might be required. For example, for the quantum particle case, this means that we just need to record, in discrete time instants, the probabilities (denoted as 'occupation probabilities') that the particle is in the positions (even part of them) identified by the nodes of the graph $\mathcal{G}$. These advantages can find application in experimental setups affected by stochastic noise sources as the ones in [9, 10, 12], and even in the available or coming quantum devices where a noise certification could be crucial before performing any task [14, 49] (see also the subsection 4.3 below).

2. Stochastic quantum dynamics

Let us introduce the general physical framework to which our ML methods will be applied. For this purpose, we consider a quantum particle that randomly moves on a complex graph $\mathcal{G}$ by following the quantum mechanics postulates. The complex graph is described by the pair $(\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes or vertices while $\mathcal{E}$ is the set of links, denoted as $s \leftrightarrow \ell$, coupling pairs of nodes, with $s, \ell = 1, \ldots, d$ and d being the total number of nodes. Each node is associated with a different particle position, while the links correspond to the possibility that

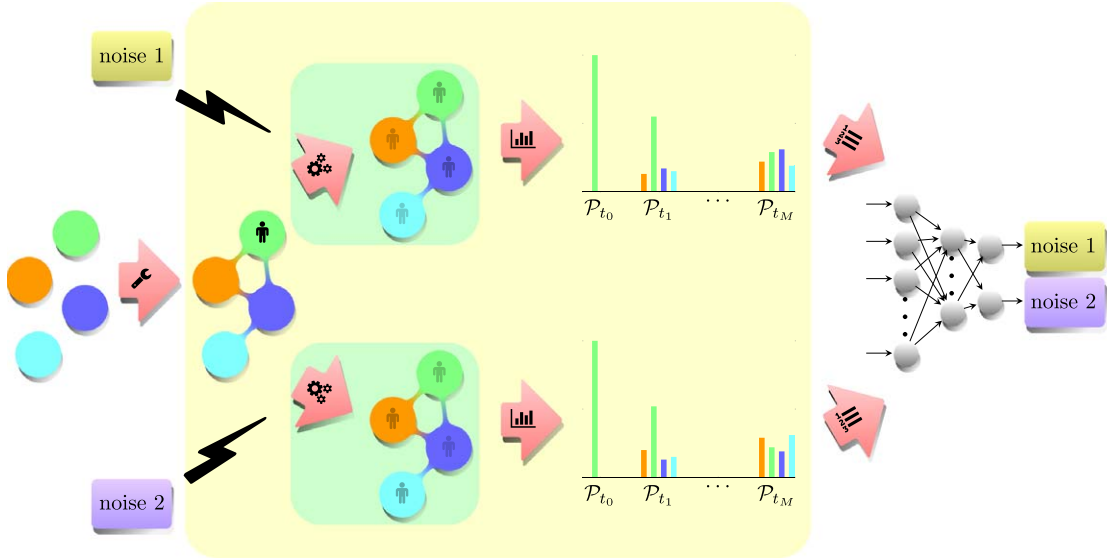

Figure 1. Pictorial representation of the proposed machine learning procedure for noise classification. For a fixed set of nodes $\mathcal{N}$ (coloured circles) we take into account the stochastic evolution of a quantum particle in a network affected by different types of noise sources. Such noisy quantum dynamics are evaluated at M consecutive steps (small green plate). To make the synthetic generated data closer to a possible real setting, the topology $\mathcal{E}$ (edges linking the coloured circles) of the network and the initial state distribution $\mathcal{P}_{t_0}$ (black pawn on the green right circle) are chosen randomly for a predefined number of different configurations (yellow background). After the dynamics, all the distributions $\mathcal{P}_{t_k}$, in correspondence of the $M+1$ time instants t_k with k=0, … ,M, are collected and recorded along with the noise type label. Then, a data set of N different realisations is used to train a ML-model (a neural network in the figure) for the classification of noise sources.

the particles jumps from a node to another. In particular, the links in $\mathcal{E}$ can be summarised in the adjacency matrix A_t (time-dependent operator in the more general case), whose elements are given by

$$A_t^{(s,\ell)} \equiv \begin{cases} g_t & \text{if } s \leftrightarrow \ell \in \mathcal{E} \\ 0 & \text{if } s \leftrightarrow \ell \notin \mathcal{E}. \end{cases} \tag{1}$$

In this way, we are implicitly assuming that all the links are equally coupled with the same weight equal to g_t that is taken as a time-dependent parameter.

Here, the coupling g_t is modelled as a stochastic process defined by the collection of random variables $\boldsymbol{g} \equiv (g_{t_0}, \dots, g_{t_{M-1}})^T$, with $(\cdot)^T$ being the transposition operation, in correspondence of the discrete time instants $t_k, k = 0, \dots, M-1$. At each t_k, g_t is sampled from a specific probability distribution Prob(g) and is assumed to remain constant at the extracted value for the entire time interval $[t_k, t_{k+1}]$. For simplicity, also the value $\Delta \equiv t_{k+1} - t_k$ is taken constant for any $k = 0, \dots, M-1$, and the stochastic process g_t is considered to take D different values $g^{(1)}, \dots, g^{(D)}$ with probabilities $p_{g^{(1)}}, \dots, p_{g^{(D)}}$. In this way,

$$\text{Prob}(g) = \sum_{j=1}^{D} p_{g^{(j)}} \delta(g - g^{(j)}) \tag{2}$$

is provided by a discrete probability distribution with D values, with $\delta(\cdot)$ denoting the Kronecker delta.

If $\boldsymbol{g}$ is provided by a collection of i.i.d. random variables sampled from the probability distribution Prob(g), then the noise sequence that affects the link strength g is uncorrelated over time, and it is denoted as *Markovian*. Conversely, in case the occurrence of the random value $g^{(j)}$, j=1, … ,D, at the discrete time instants t_k, $k = 0, \dots, M-1$, depends on the sampling of g at previous time instants, the noise sequence is time-correlated and the noise is denoted as *non-Markovian* or as a *coloured noise process*. In this regard, notice that the value of the parameters, which define the correlation among different samples of noise in single time-sequences, uniquely set the colour of the noise. Also observe that, known the multi-times distribution Prob($\boldsymbol{g}$) defined over the discrete time instants t_k, one can compute the noise auto-correlation function, whose Fourier transform is by definition the power spectral density of the noise process. In other terms, there is a one-to-one mapping between the representations of the noise in the time and frequency domains respectively. This entails that noise sensing can be performed in one of the two domain at best convenience. Moreover, this also motivates the generality of the stochastic quantum model we are here introducing that, indeed, can be applied to all those problems concerning the transport of single particles within a network [50–52], but also to quantum system dynamics influenced by the external environment as those in [2, 4, 18].

In our model we adopt as correlation model the well-known formalism of time-homogeneous *discrete Markov chains* [53]. The latter can be graphically interpreted as state-machines that assign the conditional probability of 'hopping' from each possible value of g to an adjacent one at consecutive time instants. Each conditional probability is defined, at any time t, by a *transition matrix* T that is a left or right stochastic operator. Let us remind that discrete Markov chains differ by a parameter m named the *order* of the chain. In a Markov chain of order m, future realisations of the sampled random variable (e.g., our g_t) depend on the past m realisations in previous time instants. Here, we will consider ($m = 1$)-order discrete Markov chains, namely correlated noise sequence characterised by a single (1-step) transition matrix T that we aim to discriminate by means of properly-developed ML techniques. This choice is simply dictated by our desire to effectively illustrate the obtained results, and not by intrinsic limitations of the methods we are going to propose. As an example, let us assume $m = 1$ and $D = 2$. In this specific case, by taking the conditional probabilities $p(g_{t_k}|g_{t_{k-1}})$ with g_{t_k} equal to $g^{(1)}$ or $g^{(2)}$ for any k, it holds that $p(g_{t_k}|g_{t_{k-1}})$ is equal to one of the elements within the following transition matrix:

$$T = \begin{pmatrix} p(g_{t_k} = g^{(1)}|g_{t_{k-1}} = g^{(1)}) & p(g_{t_k} = g^{(1)}|g_{t_{k-1}} = g^{(2)}) \\ p(g_{t_k} = g^{(2)}|g_{t_{k-1}} = g^{(1)}) & p(g_{t_k} = g^{(2)}|g_{t_{k-1}} = g^{(2)}) \end{pmatrix}. \tag{3}$$

Thus, the stochastic realisations of g in different time instants are not correlated only if all the elements of T are equal to $1/2$. In addition, we assume that all the nodes of the graph $\mathcal{G}$ have the same energy. Without loss of generality, one is allowed to set such energy to zero, with the result that the Hamiltonian H_t of the quantum particle is identically equal to the adjacency matrix A_t, i.e., H_t=A_t for any time instant t. Moreover, we consider that the state of the particle, moving on a graph with d nodes, is provided by the density operator ρ_t that, by definition, is an Hermitian, positive semi-definite, idempotent operator matrix with trace 1. By using the vectorisation operation vec[·], we convert ρ_t into the column vector

$$\begin{aligned} \boldsymbol{\lambda}_t &\equiv \mathrm{vec}[\rho_t] \\ &= (\rho_t^{(11)}, \ldots, \rho_t^{(d1)}, \rho_t^{(12)}, \ldots, \rho_t^{(d2)}, \ldots, \rho_t^{(dd)}) \in \mathbb{C}^{d^2} \end{aligned}$$

where $\rho_t^{(s,\ell)}$ denotes the $(s\,\ell)$-element of ρ_t. The state $\boldsymbol{\lambda}_t$ is a vector of d^2 elements belonging to the space of complex numbers. Since a quantum particle can live in a superposition of positions, whereby also quantum coherence plays an active role, d elements of $\boldsymbol{\lambda}_t$ corresponds to the probabilities of measuring the particle in each of the allowed positions, while the other elements are *quantum coherence* terms that identify interference patterns between the nodes of the graph. Thanks to the vectorisation of ρ_t, the ordinary differential equation, governing the dynamics of the particle, is recast in a linear differential equation for $\boldsymbol{\lambda}_t$, i.e.,

$$\begin{aligned} &\frac{\partial}{\partial t}\boldsymbol{\lambda}_t = \mathcal{L}_t\,\boldsymbol{\lambda}_t \Longleftrightarrow \boldsymbol{\lambda}_t = e^{\mathcal{L}_t}\boldsymbol{\lambda}_0 \\ &\mathcal{L}_t \equiv -\frac{i}{\hbar}(\mathbb{I}_d \otimes A_t - A_t^T \otimes \mathbb{I}_d) \end{aligned}$$

with $\otimes$Kronecker product and $\hbar$ reduced Planck constant. By construction, $\mathcal{L}_t$ is a skew-Hermitian operator for any time instant t, i.e., $\mathcal{L}_t^\dagger + \mathcal{L}_t = 0\ \forall t$.

3. Problem formulation

Our aim is to identify the presence of noise sources acting on the coupling g_t of the adjacency matrix A_t, and then discriminate among different noise probability distributions Prob(g) and correlation parameters in the samples of the time-sequences $\boldsymbol{g}$. Moreover, we also aim to evaluate if such tasks can be carried out by only measuring the population terms of the particle at the discrete time instants t_k, even by taking into account few runs of the quantum system dynamics.

The population values are collected in the vectors $\mathcal{P}_{t_k} \in \mathbb{R}^d$ that have as many elements as the nodes of the graph. After each stochastic evolution of the quantum particle, $\mathcal{P}_{t_k}$ takes different values depending on the specific realisation of $\boldsymbol{g}$.

At the experimental level, the population distributions $\mathcal{P}_{t_k}$ can be obtained in multiple runs, by stopping the stochastic evolution of the system at each time t_k (with k=1, … ,M), then collecting the measurement records and restarting from the beginning the experimental routine. This means that one does not need to experimentally implement sequential measurements routines, requiring to take into account also the quantum measurement back-action on the state of the system. The measurement outcomes can be just recorded at the end of the quantum system evolution; however, this can be realized at the price of performing multiple runs of the stochastic quantum dynamics under scrutiny.

3.1. Data set generation

For the generation of the data used to train the ML-models, we consider two variants of three different classification problems. Each sample of the data sets is created by first generating a random set of links $\mathcal{E}$ (random topology) for the graph $\mathcal{G}$, and then initialising the particle in a randomly chosen node of the graph. We set $M = 15$ as the number of evaluations (measurements) of the quantum particle dynamics, and $d = 40$ as the number of nodes of the graph $\mathcal{G}$. This means that $\mathcal{P}_{t_0}$ is a Kronecker delta centered in one of the 40 nodes, and the stochastic quantum dynamics is evolved for 15 steps for each simulated noise source of the generated data set. Here, it is worth noting that the choice of $M = 15$ is dictated by the fact that in recent experiments as for instance in [9, 54, 55], the number of intermediate quantum measurements does not exceed 10, and thus $M = 15$ is sufficiently large to represent actual physical setups. Instead, regarding taking $d = 40$, such a value is just able to generate a complex landscape for the particle dynamics and small enough to be numerically manageable. The total considered dynamical time t_M is taken equal to $t_M = 1$ or $t_M = 0.1$ in dimensionless units, each of them corresponding to a specific variant. Notice, indeed, that the values of t_M are expressed consistently with the energy scale of the couplings g_t, whose random values $g^{(j)}$ belong to the set $\{1, 2, 3, 4, 5\}$ in the data set generation, such that $\hbar$ can be reliably set to 1 as usual. All the probability distributions $\mathcal{P}_{t_k}$ for $k = 0, \ldots, 15$ are stored together with the attached label that indicates the associated type of noise.

For each of the two variants of our classification problems, we generate three different balanced data sets of 20 000 samples. The first data set, which we call **IID**, is suitable for a supervised binary classification task that discriminates between two different i.i.d. noisy quantum dynamics, where the noise sources have the same support but different probability distribution Prob(g).

The second data set, named as **NM**, concerns the classification of two different coloured noisy quantum dynamics with the noise sources again having the same support but different Prob(g) (the same ones as in the data set **IID**) and a transition matrix T.

Finally, the third data set, called **VS**, is created for the classification between stochastic quantum dynamics affected respectively by an i.i.d. and a coloured noise with same support and Prob(g).

Note that choosing graphs with random links allows to increase the statistical variability of the input data, with the result that the ML algorithms learn to classify noise sources independently of the graph topology. The aim, indeed, is to prevent that the ML-models rely only on features specific to a small class of topologies. Moreover, taking random initial distributions $\mathcal{P}_{t_0}$ allows to increase the *robustness* of the ML methods, making them less likely to overfit on the synthetic data set.

As it will be explained in the following, some ML-models that we are going to introduce will use as input only the last distribution $\mathcal{P}_{t_{15}}$, while other ML-models will take all the $\mathcal{P}_{t_k}$ for any t_k. Moreover, each data set is balanced split in a *training set* of 12 000 samples, a *validation set* of 4 000 samples, and a *test set* of 4 000 samples.

In tables 1 and 2 we plot the occupation probabilities $\mathcal{P}_{t_k}$ (just for the **IID** case for the sake of an easier presentation), being here interested in looking for the difference between choosing $t_{15} = 0.1$ or 1, which identify the two different variants of the generated data set. In this regard, it is worth noting that the duration $t_{15} = 0.1$ (in dimensionless units) of the quantum system dynamics, as in the example in table 1, is the minimal one to observe the diffusion of the system's population outside the node on which has been initialised. However, as it will be verified by our experiments and explained later, with this choice one has that, by taking $t_{15} = 0.1$, the classification problem results quite straightforward. Indeed, just basic ML-models that are only trained on $\mathcal{P}_{t_{15}}$ (thus, only on the final distribution $\mathcal{P}$) are able to correctly classify between two noisy quantum dynamics. Therefore, it was more interesting to increase the value of t_{15} up to $t_{15} = 1$ (in dimensionless units). As in the example of table 2, it leads to more complex data sets, and only deep learning models, designed to read all the $\mathcal{P}_{t_k}$, can classify the generated noisy quantum dynamics.

As final remark, note that the current synthetic data set is build assuming perfect measurement statistics, as it was obtained from a large enough number of repetitions of the noisy quantum dynamics. Hence, to better adapt the synthetic data set to real data, one should simulate experimental case in which the measurement statistics are estimated from a finite number of dynamics realizations (i.e., measurement shots).

3.2. Classification tasks

We here present the binary supervised classification tasks that we are going to address, by taking $\mathcal{P}_{t_k}$ as input:

Two different probability distributions Prob(g)—specifically, $p_g^{(1)}, \ldots, p_g^{(5)} = (0.0124, 0.04236, 0.0820, 0.2398, 0.6234)$ and $=(0.1782, 0.1865, 0.2, 0.2107, 0.2245)$—both associated with i.i.d. noise sources.

Two different Prob(g) (the same as (i)) and different values of the correlation parameters—identified by transition matrices T as explained in section 2—for coloured (thus, non-Markovian) noise processes.

Table 1. Example of a part of $\mathcal{P}_{t_k}$ for all the discrete time instants t_k for a noisy quantum dynamics affected by i.i.d. noise sources and $t_{15} = 0.1$ (in dimensionless units). In the table, $\mathcal{P}^{(s)}_{t_k}$ denotes the s-th element of the vector $\mathcal{P}_{t_k}$ for any t_k, $k = 0, \ldots, 15$.

	$\mathcal{P}^{(35)}_{t_k}$	$\mathcal{P}^{(36)}_{t_k}$	$\mathcal{P}^{(37)}_{t_k}$	$\mathcal{P}^{(38)}_{t_k}$	$\mathcal{P}^{(39)}_{t_k}$	$\mathcal{P}^{(40)}_{t_k}$
t_0	0.00	0.00	0.00	0.00	1.00	0.00
t_1	0.00	0.00	0.00	0.00	0.99	0.00
t_2	0.00	0.00	0.00	0.00	0.93	0.00
t_3	0.00	0.00	0.01	0.01	0.85	0.01
t_4	0.00	0.01	0.01	0.01	0.78	0.01
t_5	0.01	0.01	0.01	0.01	0.69	0.01
t_6	0.01	0.02	0.01	0.01	0.63	0.00
t_7	0.01	0.02	0.01	0.01	0.57	0.00
t_8	0.01	0.02	0.01	0.01	0.52	0.00
t_9	0.02	0.02	0.01	0.01	0.45	0.00
t_{10}	0.02	0.02	0.02	0.02	0.37	0.01
t_{11}	0.01	0.02	0.02	0.03	0.29	0.01
t_{12}	0.01	0.01	0.03	0.04	0.20	0.02
t_{13}	0.01	0.01	0.04	0.04	0.14	0.01
t_{14}	0.01	0.02	0.04	0.05	0.08	0.01
t_{15}	0.01	0.02	0.04	0.05	0.06	0.01

Table 2. Example of a part of $\mathcal{P}_{t_k}$ for all the discrete time instants t_k for a noisy quantum dynamics affected by i.i.d. noise sources and $t_{15} = 1$ (in dimensionless units). Again, $\mathcal{P}^{(s)}_{t_k}$ denotes the s-th element of the vector $\mathcal{P}_{t_k}$ for any t_k, $k = 0, \ldots, 15$. The topology and the initial state, for this example, are the same of those in table 1.

	$\mathcal{P}^{(35)}_{t_k}$	$\mathcal{P}^{(36)}_{t_k}$	$\mathcal{P}^{(37)}_{t_k}$	$\mathcal{P}^{(38)}_{t_k}$	$\mathcal{P}^{(39)}_{t_k}$	$\mathcal{P}^{(40)}_{t_k}$
t_0	0.00	0.00	0.00	0.00	1.00	0.00
t_1	0.02	0.02	0.01	0.01	0.45	0.00
t_2	0.01	0.01	0.02	0.03	0.05	0.02
t_3	0.00	0.00	0.00	0.00	0.13	0.02
t_4	0.02	0.01	0.01	0.01	0.12	0.02
t_5	0.01	0.01	0.03	0.03	0.06	0.01
t_6	0.01	0.01	0.01	0.01	0.01	0.00
t_7	0.04	0.03	0.01	0.06	0.11	0.00
t_8	0.04	0.00	0.03	0.11	0.11	0.03
t_9	0.03	0.00	0.03	0.01	0.01	0.10
t_{10}	0.05	0.01	0.01	0.04	0.08	0.04
t_{11}	0.01	0.03	0.02	0.00	0.08	0.02
t_{12}	0.00	0.05	0.02	0.04	0.00	0.06
t_{13}	0.01	0.03	0.00	0.02	0.05	0.07
t_{14}	0.00	0.00	0.00	0.01	0.12	0.00
t_{15}	0.00	0.00	0.01	0.04	0.10	0.01

An i.i.d. and a coloured noise process with the same support g and distribution $p_g^{(1)}, \ldots, p_g^{(5)} = (0.0124, 0.04236, 0.0820, 0.2398, 0.6234)$ that thus differ for the presence of non-zero correlation parameters.

The values of both Prob(g) and the transition matrices T, used in our numerical simulations, are chosen randomly.

Table 3. Percent accuracy γ (calculated on the test set) of the ML-models trained in the tasks of binary classification of noisy quantum dynamics with: (i) Two different i.i.d. noise sources (**IID**); (ii) two different coloured noise processes (**NM**) leading to non-Markovian dynamics; and (iii) one i.i.d. vs one coloured noise sources (**VS**). In this regard, let us recall that the coloured noise processes addressed in this paper are such that the probability distributions $\mathcal{P}_{t_k}$ depend both on Prob(g) and 1-step transition matrix T. In the first three columns of the table, the total duration of the dynamics is equal to $t_{15} = 0.1$, while in the last three is $t_{15} = 1$. The first two rows of the table report the results of the ML-models that use as input only $\mathcal{P}_{t_{15}}$, while the models of the other rows take as input all the probability distributions $\mathcal{P}_{t_k}$ for $k = 0, \ldots, 15$. The highest values of the accuracy have been underlined, and a color gradient (from blue to bright red) highlights the difference in their values.

		$t_{15} = 0.1$			$t_{15} = 1$		
		IID	**NM**	**VS**	**IID**	**NM**	**VS**
$\mathcal{P}_{t_{15}}$	**m-SVM-single**	97.0	82.3	96.5	50.3	51.2	49.5
	m-MLP-single	96.9	80.7	96.6	49.5	50.7	50.2
$\mathcal{P}_{t_0}, \ldots, \mathcal{P}_{t_{15}}$	**m-SVM**	96.4	80.1	96.3	73.6	61.9	75.0
	m-MLP	96.7	80.7	96.3	74.0	61.4	70.7
	m-GRU	96.5	91.5	96.7	90.5	73.3	88.2
	m-LSTM	96.8	90.4	96.4	88.6	70.3	86.3
	m-biGRU	96.6	92.2	96.6	91.0	74.6	90.6
	m-biLSTM	96.7	89.7	96.5	90.8	70.6	87.2
	m-biGRU-att	97.0	91.6	96.1	90.9	73.4	87.9
	m-biLSTM-att	96.9	87.9	96.3	89.0	71.6	87.4
	m-biGRU-max	96.6	92.6	96.6	91.8	76.1	90.4
	m-biLSTM-max	96.6	91.4	96.3	91.4	74.9	89.0

To solve the classification tasks above, we have employed in this paper both a more standardized ML-model that is Support Vector Machine (SVM) [33] and more recent Artificial Neural Networks (ANNs) [47, 56–58] models in the form of MLPs and RNNs. For an exhaustive explanation of such a ML-models refer to the appendix.

4. Results

In our work, we consider two SVM models as baseline. The first one is denoted **m-SVM-single** and uses as input only the final probability distribution $\mathcal{P}_{t_{15}}$ (the prefix *m-* stands for 'model', to avoid confusion with the algorithm name; the suffix *-single* means that it is based only on $\mathcal{P}_{t_{15}}$). Instead, the second one, which we call as **m-SVM**, uses the set of all the $\mathcal{P}_{t_k}$ with $k = 0, \ldots, 15$. For both of them, we try the following *kernels* to increase the dimension of the feature-space that makes linearly separable the data-set: linear, polynomial with degree 2, 3 and 4, and Radial Basis Function (RBF).

Then, we denote with **m-MLP-single** a MLP (also refer to equation (7) in the appendix), with $\mathbf{x} \equiv \mathcal{P}_{t_{15}}$ and $\mathbf{y} \equiv (0, 1)$ or $(1, 0)$ to identify the two noisy quantum dynamics that we aim to classify. Differently, **m-MLP** takes as input the set of all $\mathcal{P}_{t_k}$.

Moreover, **m-GRU** and **m-LSTM** are unidirectional RNNs that employ the final hidden representation (see equations (19) and (20) in appendix for more details). They are implemented by exploiting the Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) methods, respectively. The input to the models is $\mathbf{x}_{i+1} \equiv \mathcal{P}_{t_i}$, with $i = 0, \ldots, 15$, while the output $\mathbf{y} \equiv (0, 1)$ or $(1, 0)$ as before. Besides, **m-biGRU** and **m-biLSTM** are the bidirectional versions of **m-GRU** and **m-LSTM**, while **m-biGRU-att** and **m-biGRU-max** are as **m-biGRU** but in addition, respectively, with an attention mechanism and a max pooling (respectively, equations (24) and (25) in appendix) as forms of aggregation of the RNN hidden representations . Similarly, **m-biLSTM-att** and **m-biLSTM-max** are the attentive and max pooling equivalents of **m-biLSTM**, respectively.

In table 3, for each model we report the best classification accuracy that is computed on the predictions performed over the test set. More formally, we define the prediction set

$$\Gamma \equiv \{(\mathbf{y}_1, \hat{\mathbf{y}}_1), \ldots, (\mathbf{y}_n, \hat{\mathbf{y}}_n)\}$$

where $\mathbf{y}_1, \ldots, \mathbf{y}_n$, taken from the data set, denote the true noise sources affecting the quantum system dynamics, and $\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_n$ the corresponding predictions of the ML-model. Hence, the (percent) accuracy γ, function of Γ, is provided by

$$\gamma(\Gamma) \equiv \frac{100}{n} \sum_{i=1}^{n} \mathbb{1}\left\{ \arg\max_{j=1,2} \hat{\mathbf{y}}_i^{(j)} = \arg\max_{j=1,2} \mathbf{y}_i^{(j)} \right\}, \tag{4}$$

where

$$\mathbb{1}\{c\} \equiv \begin{cases} 1, & \text{if } c \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

is the so-called *indicator function*. The accuracy γ defines the correctness of the model and can be used as a metric to identify which solution is better. In detail, for binary classification problems, as in our case, if $\gamma \simeq 50$ the classification is equivalent to perform a random guess, thus the model does not work. Instead, when $\gamma \simeq 100$ the model perfectly classifies all the elements of the test set, thus it is a nearly ideal classifier.

From the table, one can first observe that, by dealing with a total duration of the dynamics (in dimensionless units, by rescaling as the inverse of the couplings g_t) equal to $t_{15} = 0.1$, we can reach the 97% and 96.6% of accuracy for the classification tasks **IID** and **VS** via an SVM using as input only the distribution $\mathcal{P}_{t_{15}}$. Instead, the task **NM** is more difficult: 82.3% of accuracy is achieved by SVMs applied just on $\mathcal{P}_{t_{15}}$. MLP does not provide better results. In this case (**NM** tasks), to obtain an accuracy over 90%, one can resort to RNNs taking as inputs all the $\mathcal{P}_{t_k}$ for $k = 0, \ldots, 15$.

Conversely, for a longer dynamics, i.e., with $t_{15} = 1$, we notice that using only $\mathcal{P}_{t_{15}}$ all the three classification tasks are not solved neither with SVM nor MLP. Indeed, the accuracy γ is always around 50% and the models basically perform random guesses. The accuracy is increased by means of an SVM or an MLP based on all $\mathcal{P}_{t_k}$, with $k = 0, \ldots, 15$ as input. However, to get over 90% of accuracy on the tasks **IID** and **VS**, we need to employ RNNs. The task **NM** with $t_{15} = 1$ is the most difficult among the analysed ones, and just 76.1% of accuracy is obtained using RNNs. It is worth noticing that, for the tasks with $t_{15} = 1$, we have empirically observed that the models adopting GRU perform better with respect to the ones that employ LSTM. Moreover, setting the bidirectionality in the RNNs allows slight improved accuracy, as well as the use of max pooling in aggregation. Instead, the attention mechanism does not seem to be beneficial for those tasks.

Among the proposed solutions, the more-performing is **m-biGRU-max** that is realised by a bidirectional RNN with GRU and max pooling aggregation. However, from our numerical simulations, we have observed that, independently on the employed ML-model, the value of the total dynamical time as well as M and Δ (see also paragraph 4.1) emerge to be crucial for quantum noise classification. Specifically, by taking a quantum dynamics with a short enough duration, also SVMs are able to classify quantum noise sources with very high accuracy. With short enough dynamics we mean short with respect to the time needed to the particle in escaping from the initial node of the graph, which in our case is around $t_{15} = 0.1$. Instead, with t_{15} around 1 only RNNs provide better results, and for $t_{15} \gg 1$ none of the proposed ML-techniques solves quantum noise classification problems (these results have not been reported in table 3 for the sake of better presentation). It is also worth stressing that, if the duration of the quantum dynamics is $t_{15} = 0.1$, ML-models efficiently classify quantum noise sources by only processing the last measured distribution $\mathcal{P}_{t_{15}}$. These findings can be relevant for effective implementation (also at the experimental level), since the training and tuning of SVM is orders of magnitude faster with respect to ANNs (e.g., around minutes vs hours or even days depending on the model and provided that a GPU is used). The reason to that has to be found in the more complex structure of the ANNs than SVMs.

4.1. Scaling of the classification accuracy.

Let us now investigate the scaling of the classification accuracy γ, as a function of both the interval Δ between two consecutive transitions for $\mathbf{g}$ and the number M of discrete time instants. Notice that Δ and M are related to the total dynamical time t_M, since $t_M \equiv M\Delta$.

A possible explanation of the differences observed between the three previously-analysed scenarios, i.e., $t_{15} = 0.1$, $t_{15} = 1$ and $t_{15} \gg 1$ (in dimensionless units), could be that the information on both the noise source and the initial quantum state is lost during the evolution of the system. For such aspect, not only the total dynamical time t_{15} could play a role, but also the time interval $\Delta \equiv t_1 - t_0 \equiv \cdots \equiv t_M - t_{M-1}$. In fact, it is reasonable to conjecture that a ML-model, able to correctly classify our noisy quantum dynamics with $t_{15} = 1$ (thus $M = 15$), can also work with $t_{M'} \gg 1$ for $M' > 15$ and $\Delta' = \Delta$ where $\Delta' \equiv t_1 - t_0 \equiv \cdots \equiv t_{M'} - t_{M'-1}$. In this way, the sequence $\mathcal{P}_{t_1}, \ldots, \mathcal{P}_{t_{15}}$ is contained in $\mathcal{P}_{t_1}, \ldots, \mathcal{P}_{t_{M'}}$. In other terms, we conjecture that the classification problem can be solved even for longer noisy quantum dynamics, but provided that Δ remains small.

To gain evidence on this conjecture, we have performed two additional experiments. Starting from the task **IID** with $t_{15} = 1$ and **m-biGRU-max** as baseline (accuracy 91.8%), the same model (optimised in the same hyperparameters space) is trained on two new data sets. In both data sets, $t_M = 2$ with M equal to 15 for the first data set and 30 for the second one. Thus, in the former $\Delta' > \Delta$ with Δ time interval of the original data set, while in the latter $\Delta' = \Delta$. The first experiment ($\Delta' > \Delta$) provides a classification accuracy of 81.1%, contrarily to the results from the second experiment ($\Delta' = \Delta$), where a better accuracy of 96.3% is achieved. We thus observe that, by taking $\Delta' = \Delta$ and the same ML-model, the classification problem can be solved with

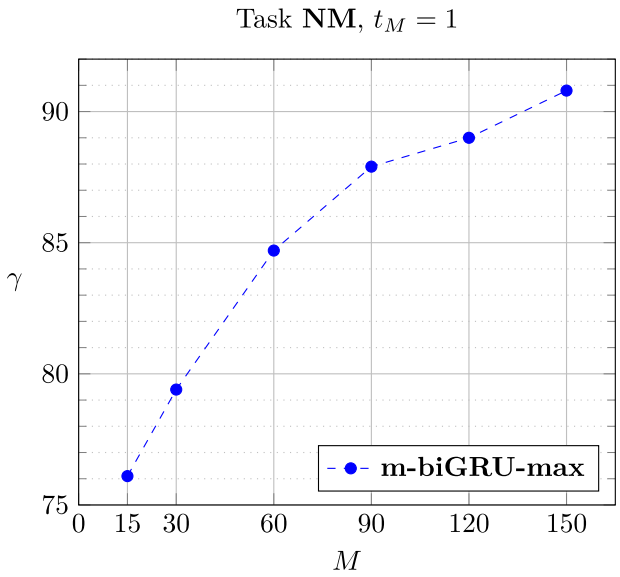

Figure 2. Percent classification accuracy γ versus M. It refers to the test set associated with the model **m-biGRU-max** for the task **NM**, where the value of the total evolution time is fixed to $t_M = 1$. It is worth noting that the first point of the figure corresponds to the value in table 3 obtained for $t_{15} = 1$.

an higher accuracy, but at the price of a longer training time. Indeed, in this case, the length of each sample of the data set is twice the original one.

In another experiment, whose results are shown in figure 2, we vary M by keeping the total evolution time equal to $t_M = 1$. Such tests use as baseline the model **m-biGRU-max** applied on the most difficult task of table 3, i.e., **NM** with $t_{15} = 1$. As a result, the achieved classification accuracy is directly proportional to M and, thus, inversely proportional to the value of the time interval Δ. Indeed, by taking t_M fixed and reducing Δ, the classification accuracy of the same model can be enhanced. Specifically, it is possible to obtain more than 90% of accuracy also for the task **NM** with a total dynamical time equal to 1, at the price of a longer training time as the length of the sequences increases.

4.2. Quantum advantages

Here, we address the following question: Could the proposed ML techniques be applied for the inference of noise sources affecting the dynamics of classical systems, e.g., Langevin equations [59]? Probably yes, but we expect that their application to quantum systems, maybe surprisingly, can be more effective than on classical systems. Both classical (non-periodic) dissipative dynamics [59] and stochastic quantum dynamics (stochastic due to the presence of an external environment, or noise sources as in our case) can asymptotically tend to a fixed-point, whereby the information on the initial state is lost. This means that the states of the system used for this noise classification tend to become indistinguishable as time increases. Classically, this can happen due to energy dissipation introduced by damping terms Instead, quantum-mechanically, a dynamical fixed point can be reached due to decoherence that makes vanishing, at least on average, all quantum coherence terms [5]. Thus, once the transient of the evolution is elapsed, the evaluation of the final state of the system does not bring information neither on the initial state nor on the initial dynamics bringing the system to the asymptotic fixed-point. In our case, we have observed that, by using only $\mathcal{P}_{t_{15}}$ with long total dynamical time, the accuracy of all the classification tasks is always around 50% both for SVM and MLP. Consequently, if one aims to infer/reconstruct the value of parameters, signals or operators that influence the system dynamics by measuring its evolution, the most appropriate time window is during the transient. In this regard, a quantum dynamic, until it is nearly close of being unitary, is able to explore different configurations thanks to linearity and the quantum superposition principle. Conversely, classical dynamics, not being able to propagate superpositions of their trajectories, cannot provide per time unit the same amount of information on the quantity to be inferred.

In conclusion, the application of the proposed methods should be more accurate if applied to quantum systems than classical ones, but during the transient of its dynamical evolution when quantum effects are still predominant and the distance among the state and the fixed point is not negligible.

4.3. Proposal for application to quantum computers

Our techniques are expected to be adopted to witness the (non-)Markovianity of the noise sources in commercial quantum devices, as for example the Q-IBM® [49] or Rigetti®. In fact, such quantum devices, as

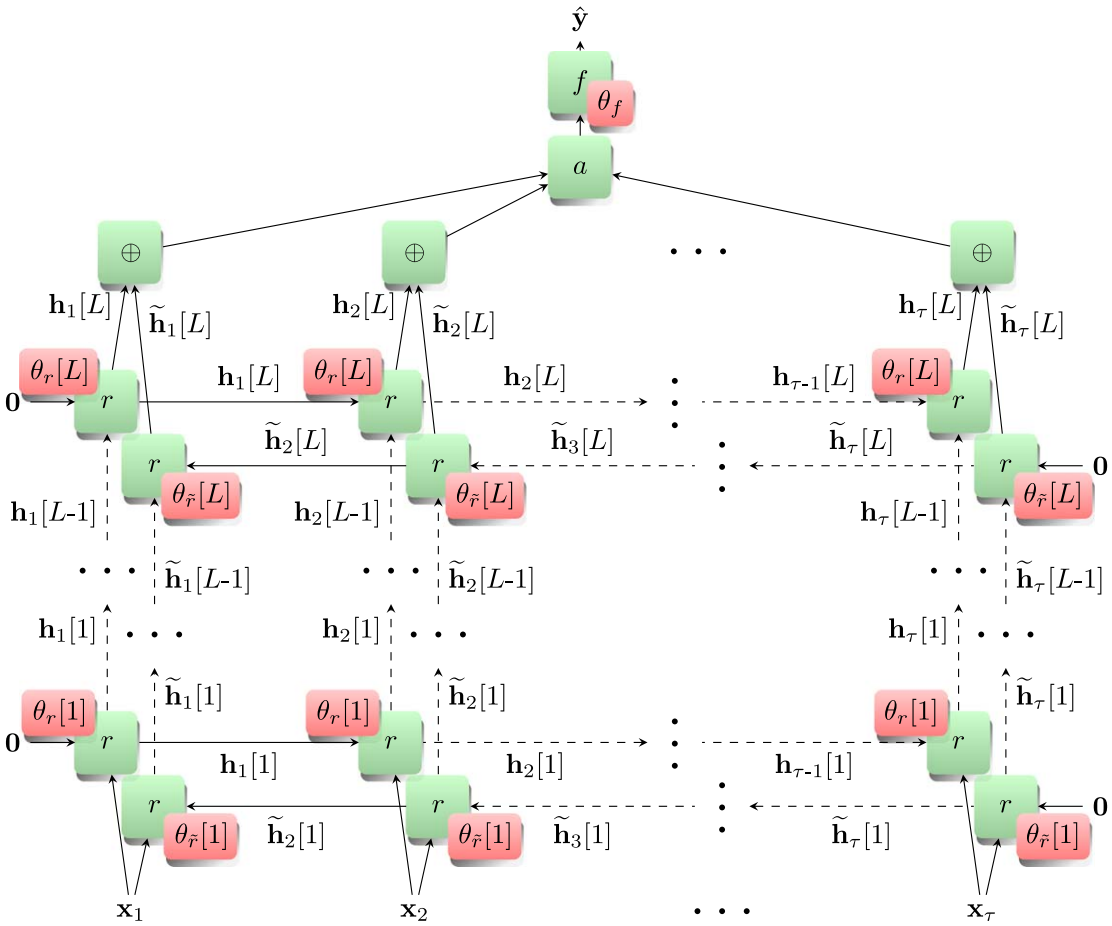

Figure 3. Diagram of a bidirectional multi-layer RNN where the nonlinear function r is defined in equations (17) and (18), a can be defined either with equations (24) or (25), f is provided by equation (20), and $\oplus$denotes concatenation. The input sequence $\mathbf{x}_t$, with $t = 1, \ldots, \tau$, is processed sequentially in both directions by the function r that is parametrised by the shared sets of weights $\theta_r[1]$ and $\theta_{\tilde{r}}[1]$ for the forward and backward directions, respectively. The hidden representations $\mathbf{h}_t[1]$ and $\widetilde{\mathbf{h}}_t[1]$, in turn, are processed by the subsequent layers, parametrised by a different sets of weights, so as to obtain the final hidden representations $\mathbf{h}_t[L]$ and $\widetilde{\mathbf{h}}_t[L]$. Finally, a performs the aggregation of the last hidden representations adopting the attention mechanism (24) or the max pooling in equation (25). The classification is performed by the function f that is parametrised by θ_f. The simpler form of aggregation in equation (19) is not depicted in the figure.

other Noisy Intermediate-Scale Quantum prototypes [60], are unavoidably affected by the external environment that entails random errors. Recently, in [61, 62], it has been shown that it is possible to discriminate different quantum computers by looking at the (unknown) noise fingerprints that characterize each device. Thus, what the ML techniques—presented here—could provide as an added value is to witness whether such noise fingerprints are time-correlated, and possibly how much the time-correlation is non-Markovian. For such experimental noise benchmarking, as in [61], it could be convenient to fix the connections among quantum gates (i.e., the underlying topology), and then consider more realizations of the implemented quantum dynamics affected by noise, so as to avoid the monitoring of the dynamics (cf. section 3). Notice that to make a successful classification among i.i.d. and (non-)Markovian noise samples, one should be able to previously label them as Markovian or not Markovian, and more in general to understand the main features of the noise acting on the machines. However, such task is usually very hard to be carried out. Thus, as a more plausible strategy, we propose to compare the experimental data to the theoretical prediction at the level of multi-time measurement statistics, according to the following two steps:

(1) Discriminate if and how much the measurement statistics (provided by the distributions $\mathcal{P}_{t_k}$ with k=1, … ,M, which have been measured on the real quantum devices on multiple runs) differ from the corresponding theoretical predictions. Such a difference, here on denoted as $\mathfrak{D}$, between theoretical and experimental data returns an effective prediction of the presence of noise on the machines.

(2) Conditionally to step 1), evaluate with ML-models the presence or the absence of functional relations $\mathcal{F}$ that link the difference distributions $\mathfrak{D}_{t_k}$ in correspondence of the time instants t_k. If two consecutive instances of

$\mathfrak{D}$ at times t_{k-1} and t_k are no functionally related, then the noise is originated from a i.i.d. stochastic process. If, instead, there exist a functional $\mathcal{F}_{t_{k-1}:t_k} = f(\mathfrak{D}_{t_{k-1}}, \mathfrak{D}_{t_k})$ that links together $\mathfrak{D}_{t_{k-1}}$ and $\mathfrak{D}_{t_k}$ in a non-trivial way, then the noise would come from a Markovian process. Finally, the noise process would be non-Markovian for functional relations $\mathcal{F}_{t_{k-n}:t_k}$ defined over multi times with $n > 1$.

The empirical characterization of $\mathcal{F}$, as well as the witness of non-Markovianity in the noise samples, can be obtained through the use of generative models. We can train three different models: (i) one model that generates $\mathfrak{D}_{t_k}$ by processing $\mathfrak{D}_{t_{k-n}}, \ldots, \mathfrak{D}_{t_{k-1}}$; (ii) another generative model that returns $\mathfrak{D}_{t_k}$ by taking $\mathfrak{D}_{t_{k-1}}$ as input; (iii) a model that directly generates $\mathfrak{D}_{t_k}$. If these three models have the same accuracy, then the noise process is likely i.i.d.. While, if the generative model (ii) has higher accuracy with respect to (iii) and the same accuracy than (i), then the noise process would be Markovian. Finally, if (i) has higher accuracy with respect to the models (ii) and (iii), then the noise process is non-Markovian.

To conclude, according to our proposal that will be tested in a forthcoming paper, time-correlations in the noisy samples of the distributions $\mathcal{P}_{t_k}$, with k=1, … ,M, can be determined by classifying functional relations $\mathcal{F}$ linking the difference distributions $\mathfrak{D}$, obtained by comparing the theoretical and measured values of $\mathcal{P}$ for a set of time instants. This is equivalent to discriminate coloured noise processes originated by different discrete Markov chain with non-zero transition matrices T. As a remark, it is still worth noting that also the experimental realization of the proposed procedure can be performed on multiple runs without the need to implement sequential measurements routines. Hence, each time a projective measurement is performed and the resulting measurement outcome recorded, the implemented (noisy) quantum circuit shall be executed from the beginning.

5. Conclusions

In this paper, we have addressed non-Markovian noise classification problems by means of deep learning techniques. In particular, the use of RNN—developed for sequence processing—is motivated by the fact that we deal with time-ordered sequences of data. Even without resorting to external driving that may hinder detection tasks, we managed to classify with high accuracy stochastic quantum dynamics characterized by random parameters sampled from different probability distributions, associated with i.i.d. (Markovian) and coloured (non-Markovian) noise processes. For such a purpose, several ML-models have been tested; in this regard, refer to table 3 for a summary of the results in term of the classification accuracy.

Among the proposed solutions, the more-performing is **m-biGRU-max** that is realised by a bidirectional RNN with GRU and max pooling aggregation. In fact, recurrent neural networks are particularly suitable to accomplish temporal machine-learning tasks thanks to their capability to generate internal temporal dynamics based on feedback connections. However, independently on the employed ML-model, different accuracy values are achieved depending on the values of M, Δ and the total dynamical time. The way our ML techniques rely on the parameters of the model has been addressed in the paragraph 4.1.

Overall, all our numerical results have shown that it is easier to classify between two different noisy quantum dynamics both affected by i.i.d. noise sources or by i.i.d. and coloured noise processes than between two noisy quantum dynamics subjected to coloured noise. Again it confirms the relevant role played by time-correlations and how the latter highly influence the value of the classification accuracy. Furthermore, we also expect that the same ML-techniques that we have exploited in this work could be successfully applied to classify among coloured noise with q-step transition matrices $T_{t|t-q}$ with $q > 1$.

5.1. Outlooks

As outlook, we plan to test the ML-models employed in this paper on reconfigurable experimental platforms as the ones in [63, 64], even affected by multiple noise sources. Moreover, we also aim to adapt our ML methods (and especially ANNs) to reconstruct noise processes with time-correlation as key feature in the context of regression task instead of classification. Indeed, our proposal is to provide accurate estimates of both the probability distribution Prob(g) and the transition matrix T, and the analysis would be extended for the prediction of spatially-correlated noise sources. In this way, ML approaches would represent a very promising, and possibly more accurate, alternatives to other noise-sensing techniques, e.g., those recently discussed in [65, 66].

A well-known problem in ML is the generalization to data shift. A model that is trained on a data set sampled from a specific data distribution will work correctly only with data sampled from the same distribution. In this paper, we used only synthetic data to evaluate the correctness of the training process and the ML techniques. Thus, in order to validate this approach to real data, we should first collect them. This is out of the scope of the current work, but, as a remark, we can delineate three possible ways to build a real experimental data set. The

first strategy is to acquire information a-priori on noise sources affecting the quantum system of interest in some experimental contexts by means of standard spectroscopy techniques, so that we can train the proposed ML-models to discriminate between unseen classes of noise. In this way, the initial effort in building a training data set that also contains experimental data is counterbalanced by the possibility to predict noise features by means of faster classification tasks. The second strategy, which has been employed in [61], is to collect experimental data that comes from different noisy measurement statistics whose noise processes are not necessarily known. The ML-models, then, are trained to classify (unknown) noise sources in distinct unseen sets of measurements. Finally, the third strategy, which is aimed to reduce the effort in building an informative experimental data set, is to train the ML-models first on synthetic data and then to fine tune the training on a smaller further data set with only experimental data. In such a case, it is beneficial to adopt a synthetic data set that closely adapts to the real experimental setup. For instance, a simulated extra error can be added to the measurement statistics (in our paper provided by the distributions $\mathcal{P}_{t_k}$, $k = 1, \ldots, M$) to take into account the finite number of measurement shots used for their computation.

Acknowledgments

The authors were financially supported from by the Fondazione Cassa di Risparmio di Firenze through the project QUANTUM-AI, the European Union's Horizon 2020 research and innovation programme under FET-OPEN Grant Agreement No. 828946 (PATHOS), and from University of Florence through the project Q-CODYCES.

Data availability statement

The code used to generate the data and to train the machine learning models that support the findings of this study are openly available at the following URL: https://github.com/trianam/quantumNoiseClassification

Appendix. Details on the employed ML models

In this section, aiming at addressing also an audience not necessarily expert in ML, we describe more in detail the ML-models used in our tasks.

A generic binary data set in input to ML-models is usually represented by a set of n points $\mathbf{x}_q \in \mathbb{R}^p$, with $q = 1, \ldots, n$, each of them living in the p-dimension space of the features. A *feature* is a distinctive attribute of each element of the data set. Each point $\mathbf{x}_q$ is associated with one of two different classes with binary labels $y_q \in \{-1, 1\}$, with $q = 1, \ldots, n$, depending on the specific classification problem that we are solving.

A.1. Support vector machine

Support Vector Machine (SVM) [33] is one of the first ML-model originally used to carry out classification tasks. The SVM training consists in finding the hyperplane that separates the elements $\mathbf{x}_q$ in two groups: one with the label $y_q = 1$ and the other with $y_q = -1$. The final hyperplane, solution of the classification, is the one having the maximum geometrical distance from the two parallel hyperplanes that are defined by the subsets of $\mathbf{x}_q$ called the *support* sets. When the data is not linearly separable, the *kernel trick* allows to increase the dimension of the features space in a way that the data becomes linearly separable in the new space.

Historically, SVM is a generalisation of the Support Vector Classifier (SVC) that, in turn, is an improved version of the Maximal Margin Classifier (MMC) [33]. MMCs aim at finding the hyperplane separating the two aforementioned classes of points, such that the distance between the hyperplane and the nearest points of the classes (commonly denoted as *margin*) is maximised. If the points of the data set are not linearly separable, then the value of the margin is negative. In such a case, the MMCs cannot be adopted. SVCs increase the performance of MMCs, by allowing some points of the data set, called *slack variables*, to be in the opposite part of the hyperplane with respect to the others of the belonging class. If the data set exhibits a non-linear bound between the two classes of points, SVCs are not able to correctly separate them, albeit the method returns a solution. Finally, SVMs extend the capabilities of SVCs by increasing the number of dimensions of the feature-space, such that in the new space the data set becomes linearly separable.

A.2. Multi-layer perceptron

There are several classification problems (as for example the ImageNet Large Scale Visual Recognition Challenge [56] employing millions of images with hundreds of categories) that are solved through SVM but with a quite high residual classification error. For this reason, in order to improve the performance in solving classification

problems, ANNs have been recently (re-)introduced as more-performing tools, and since 2012 have been extensively used [47, 56–58].

A MLP is composed of a variable number of fully connected layers, each of them with a variable number of artificial neurons. A single artificial neuron with I inputs ($\mathbf{x}$) calculates the output as

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

that is the weighted sum of the inputs $\mathbf{x} \in \mathbb{R}^I$ with weights $\mathbf{w} \in \mathbb{R}^I$, plus a bias term $b \in \mathbb{R}$, followed by a nonlinear activation function σ: $\mathbb{R} \to \mathbb{R}$. The most common activation functions $\sigma(\cdot)$ are: The *sigmoid* $\sigma(x) \equiv (1 + e^{-x})^{-1}$; the *hyperbolic tangent* $\sigma(x) \equiv \tanh(x)$; and the *rectifier* $\sigma(x) \equiv \max(0, x)$ [67, 68]. A single MLP-layer, composed of O neurons with I inputs, calculates

$$\hat{\mathbf{y}} \equiv \sigma(W^T \cdot \mathbf{x} + \mathbf{b}),$$

where $\hat{\mathbf{y}} \in \mathbb{R}^O$ is the output vector, $W \in \mathbb{R}^{I \times O}$ is a matrix that collects all the weight vectors of the single neurons, and $\mathbf{b} \in \mathbb{R}^O$ is the vector of the biases. Finally, an MLP with L layers calculates

$$\mathbf{h}[l] \equiv \sigma(W[l]^T \cdot \mathbf{h}[l-1] + \mathbf{b}[l]) \tag{6}$$

with $l = 1, \ldots, L$ (index over the layers) and $\mathbf{h}[0] \equiv \mathbf{x}$. Thus, $\hat{\mathbf{y}} \equiv \mathbf{h}[L]$ is the output of the MLP, where $W[l]$ and $\mathbf{b}[l]$ are, respectively, the weights and the biases of the l-th layer. Also the activation function may change depending on the specific layer. More concisely, the MLP can be denoted by the function

$$\hat{\mathbf{y}} = f(\mathbf{x};\, \theta,\, \xi) \tag{7}$$

of the inputs $\mathbf{x}$. The function f is parametrised by the set $\theta \equiv \{W[1], \mathbf{b}[1], \ldots, W[L], \mathbf{b}[L]\}$ and by the fixed hyperparameters ξ defining the number, the dimension, and the activation functions of the MLP layers.

A.3. Supervised training

Let us now introduce the supervised learning process. For the sake of clarity, we just refer to the training of the MLP; however, the same notions can be applied in general to the supervised learning of vast majority of ANNs.

Equation (7) behaves like a generic function approximator [69]. Ideally, in the training process we would like to find the parameters

$$\theta^* = \arg\min_{\theta} L_{\mathcal{D}}(\theta,\, \xi) \tag{8}$$

that minimise the theoretical risk function

$$L_{\mathcal{D}}(\theta,\, \xi) \equiv \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[\ell(f(\mathbf{x};\, \theta,\, \xi),\, \mathbf{y})], \tag{9}$$

i.e., the expected value of ℓ for $(\mathbf{x}, \mathbf{y})$ sampled from the distribution $\mathcal{D}$ that generates the data set [32]. In equation (9), ℓ: $\mathbb{R}^{O\times O} \to \mathbb{R}^+$ denotes the loss function (usually taken as a differentiable function, apart removable discontinuities) that measures the distance between the prediction $\hat{\mathbf{y}}$ and the desired output $\mathbf{y}$. In general, the distribution $\mathcal{D}$ is unknown; thus, the minimisation problem in equation (8) cannot be neither calculated nor solved. Indeed, one can dispose of a finite set $S = \{(\mathbf{x}, \mathbf{y})_1, \ldots, (\mathbf{x}, \mathbf{y})_n\}$ of samples, to train, validate and test the ML-model. By considering the partition $\{S_{tr}, S_{va}, S_{te}\}$ of S, the theoretical risk function is approximated by the empirical risk function

$$L_{S_{tr}}(\theta,\, \xi) = \frac{1}{|S_{tr}|} \sum_{(\mathbf{x},\mathbf{y})\in S_{tr}} \ell(f(\mathbf{x};\, \theta,\, \xi),\, \mathbf{y}) \tag{10}$$

that is the arithmetic mean of the loss function ℓ evaluated on all the samples of the training set S_{tr} [32]. By minimising the empirical risk function $L_{S_{tr}}(\theta,\, \xi)$ with respect to θ, the MLP is trained and θ^* obtained. Then, the validation set S_{va} is used to compute the empirical risk $L_{S_{va}}(\theta^*,\, \xi)$ that takes as input the optimal parameters attained by the minimisation of $L_{S_{tr}}$ (training stage). This procedure allows to check if the ML-model works also for unseen data. Notice that the minimisation of the training risk function $L_{S_{tr}}(\theta,\, \xi)$ with respect to θ is performed step-by-step over time. After each step (also called *epoch*), the validation risk $L_{S_{va}}(\theta^*,\, \xi)$ is evaluated, and the minimisation procedure is stopped when the time-derivative of $L_{S_{va}}(\theta^*,\, \xi)$ becomes positive for several epochs, thus showing *overfitting* [70]. In case such time-derivative remains negative or constant over time, the procedure is ended after a predefined number of epochs. The validation set S_{va} can be also used to explore other configurations ξ of the ML-model: this process is called *hyperparameters optimization*. In particular, after completing the training procedure using two different set of hyperparameters ξ and ξ', we obtain two minima θ^* and θ'^*, and then compare $L_{S_{va}}(\theta^*,\, \xi)$ with $L_{S_{va}}(\theta'^*,\, \xi')$ to also choose the best hyperparameter. Finally, we use the test set S_{te} to calculate a significant metric (in our case, the classification accuracy) and report the results.

Regarding the hyperparameters optimization, it can be performed in different ways. The most basic technique is called *grid search* whereby the training and validation are carried out on a specific set of hyperparameters configurations. The *random grid search* considers configurations where each hyperparameter is randomly chosen within an a-priori fixed range of values. It has been proved to be more efficient than standard

grid search [71]. A more sophisticated class of optimization methods is the *Bayesian optimization* [72] that updates, after the training of each hyperparameters configuration, a Bayesian model of the validation error. The best hyperparameters configuration is thus chosen as the one allowing for the lower guess validation error.

A.4. Minimisation algorithms

The most used optimisation algorithm to minimise equation (10) is the Stochastic Gradient Descent (SGD) [73–75] and its adaptive variants, such as Adaptive Moment Estimation (ADAM) [76], that changes the value of the learning rate η (i.e., the descent step) at each iteration. After having calculated the predictions $\hat{\mathbf{y}}$, the loss function $\ell(\hat{\mathbf{y}}, \mathbf{y})$ is propagated backwards (*backpropagation*) in the ANN and its gradient in the weight space is calculated. Overall, the optimisation process consists in iteratively updating the value of the weights θ according to the relation

$$\theta_i = \theta_{i-1} - \eta \nabla_\theta L_{S_b}(\theta_{i-1}, \xi),$$

where i is the index for the descent step and $S_b \subset S_{tr}$ denotes the b-th set of samples, taken from the training set and used for the computation of the gradient. If $S_b \equiv S_{tr}$, the algorithm is called *batch* SGD; if S_b contains only one element is called *on-line* SGD; finally, the most common approach (we use it here) is *mini-batch* SGD that consider $|S_b| = B$ with B a fixed dimension [75]. Hence, the update of θ follows the descent direction of the gradient, with a magnitude determined by the learning rate η.

Now, let us introduce the specific loss function ℓ considered in this paper. For classification problems with two or more classes, a common choice for ℓ is the *categorical cross entropy*, which is defined as

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_{j=1}^{O} y^{(j)} \log \hat{y}^{(j)}. \tag{11}$$

This function measures the dissimilarity between two or more probability distributions. Thus, to properly use the categorical cross entropy, it is convenient to choose the desired outputs $\mathbf{y}$ as Kronecker delta functions centered around the indices associated with each class to be classified. The model output $\hat{\mathbf{y}}$, instead, is normalised so that it represents a discrete probability distribution, i.e., a vector of positive elements summing to 1. This operation is obtained by using *softmax* [46] as the activation function of the last layer:

$$\sigma^{(i)}(\mathbf{z}) \equiv \frac{e^{z^{(i)}}}{\sum_{j=1}^{O} e^{z^{(j)}}} \tag{12}$$

where $\sigma(\mathbf{z})$ is the vector having as elements $\sigma^{(i)}(\mathbf{z})$, with $i = 1...,O$, and $\mathbf{z}$ denotes the output of the last layer before the activation function.

In the experiments, the activation functions for the hidden layers of the MLP have been chosen among the sigmoid, hyperbolic tangent and rectifier functions accordingly to the hyperparameters optimization.

A.5. Recurrent Neural Networks

A Recurrent Neural Network (RNN) is an ANN specialised for sequence processing when the data set is expressed as

$$S = \{(\{\mathbf{x}_1, \ldots, \mathbf{x}_{\tau_1}\}, \mathbf{y})_1, \ldots, (\{\mathbf{x}_1, \ldots, \mathbf{x}_{\tau_n}\}, \mathbf{y})_n\}, \tag{13}$$

where τ_r defines the number of elements of the r-th sequence. RNNs can be used in tasks regarding Natural Language Processing (NLP) [48, 77–79], time series analysis [80] and, in general, all the tasks involving ordered set of data [81]. Note that, in general, for *sequence-to-sequence* problems also $\mathbf{y}$ can be a sequence of elements, as for example in machine translation where the inputs and outputs of the RNN are sentences in different languages [82]. In this paper, sequence-to-sequence problems are not considered, and we thus consider $\tau_1 \equiv ... \equiv \tau_n \equiv \tau$.

A RNN is defined by the *recurrent* relation

$$\mathbf{h}_t = r(\mathbf{x}_t, \mathbf{h}_{t-1}; \theta_r, \xi) \tag{14}$$

where $t \in \{1, \ldots, \tau\}$, $\mathbf{h}_t \in \mathbb{R}^d$ is a d-dimensional vector with d being an hyperparameter belonging to ξ and $\mathbf{h}_0 = \mathbf{0}$ (vector of zeros). The recurrent relation (14) defines τ hidden representations $\mathbf{h}_t$ (to be seen as a *memory*) of the input sequence $\{\mathbf{x}_1, \ldots, \mathbf{x}_\tau\}_q$ with $q = 1, \ldots, n$. If the function r is implemented as an MLP (7) that takes as input the concatenation $\mathbf{x}_t \oplus \mathbf{h}_{t-1}$ (usually called 'vanilla RNN'), the model suffers the so-called *vanishing gradient problem* [83, 84] such that the weights of the last layers of the RNN are updated only with respect to the more recent input data. The vanishing gradient problem occurs when the backpropagation is performed on a high number of layers, as it could happen in our case with a large value of τ (thus meaning long input sequences). In this regard, to mitigate the vanishing gradient problem, LSTM [85] and GRU [86] have been introduced. These methods use learned gated mechanisms, based on current input data and previous hidden representations, to control how to update the current hidden representation $\mathbf{h}_t$. Specifically, if LSTM is used, equation (14) needs to be slightly modified as

$$\mathbf{s}_t = v(\mathbf{x}_t, \mathbf{s}_{t-1}; \theta_v, \xi) \tag{15}$$

$$\mathbf{h}_t = r(\mathbf{x}_t, \mathbf{s}_t, \mathbf{h}_{t-1}; \theta_r, \xi) \tag{16}$$

where $\mathbf{s}_0 = \mathbf{0}$ and v, r are, as usual, nonlinear functions. Both for GRU and LSTM, the nonlinearity of the recurrent relations is due to the adoption of the hyperbolic tangent and sigmoid functions, where the latter are employed only for the gating mechanism. It is worth noting that in equation (15) $\mathbf{s}_t$ is a state vector that allows to differently propagate over time specific elements of the hidden representations $\mathbf{h}_t$ depending on the input data. This means that, at any time t, the hidden representation $\mathbf{h}_t$ depends not only on the input $\mathbf{x}_t$ and the previous hidden representation $\mathbf{h}_{t-1}$ but also on the state vector $\mathbf{s}_t$. For further details, refer to [48, 85, 86].

RNNs are usually considered deep-learning models, due to the high number of layers, when they are unfolded on the sequence dimension for $t = 1, \ldots, \tau$. The key aspect of deep learning is the automatic extraction of features by means of the composition of a large number of layers; an increasing (deep) number of layers is typically used to extract features with increasing complexity [87].

Moreover, RNNs can be extended considering more layers [88] and processing the data bidirectionally [89]. Regarding the latter, one can define two sets of hidden representations: One for the forward and the other for the backward direction, where the t-th hidden representation depends respectively on the $(t-1)$-th or $(t+1)$-th one. More formally,

$$\mathbf{h}_t[l] = r(\mathbf{h}_t[l-1], \mathbf{h}_{t-1}[l]; \theta_r[l], \xi) \tag{17}$$

$$\widetilde{\mathbf{h}}_t[l] = r(\widetilde{\mathbf{h}}_t[l-1], \widetilde{\mathbf{h}}_{t+1}[l]; \theta_{\tilde{r}}[l], \xi) \tag{18}$$

with l=1, … ,L and $\mathbf{h}_t[0] \equiv \widetilde{\mathbf{h}}_t[0] \equiv \mathbf{x}_t$.

A.6. Classification with RNNs

Now, let us explain how to use the hidden representations to calculate the prediction $\hat{\mathbf{y}}$ in output from the ML-model. The common approach to calculate the prediction in classification problems is to use the RNN as an encoder of the sequence and to scale the dimension of the last hidden representation $\mathbf{h}_\tau[L] \oplus \widetilde{\mathbf{h}}_1[L]$ (in the more general case of bidirectional models) to the one of the output vector. This scaling can be done through a fully connected layer, or, more in general, by means of an MLP, i.e.,

$$\mathbf{a} \equiv \mathbf{h}_\tau[L] \oplus \widetilde{\mathbf{h}}_1[L] \tag{19}$$

$$\hat{\mathbf{y}} = f(\mathbf{a}; \theta_f, \xi). \tag{20}$$

Then, we can use SGD to minimise an empirical risk function similar to equation (10) of MLPs.

It is possible to consider different forms of aggregation $\mathbf{a}$ for the hidden representations $\mathbf{h}_t[L]$, with $t = 1, \ldots, \tau$, instead of using only the last hidden representation as in equation (19). In this regard, *attention mechanisms* [90–92], also in hierarchical forms [93], perform a weighted average of the $\mathbf{h}_t[L]$ where the weights are learned together with the ML-model. In detail, equation (19) becomes:

$$\mathbf{u}_t = \mathbf{h}_t[L] \oplus \widetilde{\mathbf{h}}_t[L] \tag{21}$$

$$\mathbf{v}_t = \tanh(\mathbf{W}^T \cdot \mathbf{u}_t + \mathbf{b}) \tag{22}$$

$$\alpha_t \equiv \frac{e^{\langle \mathbf{v}_t, \mathbf{c} \rangle}}{\sum_{j=1}^{\tau} e^{\langle \mathbf{v}_j, \mathbf{c} \rangle}} \tag{23}$$

$$\mathbf{a} \equiv \sum_{t=1}^{\tau} \alpha_t \mathbf{u}_t, \tag{24}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product and $\mathbf{c}$ is a learned vector that is randomly initialised and jointly learned during the training process as in [90–93]. Another form of aggregation $\mathbf{a}$ is the *max pooling aggregation*, whereby each element $\mathbf{a}^{(j)}$ of $\mathbf{a}$ just refers to a single value of t. In this case, equation (19) equals to

$$\mathbf{a}^{(j)} = \max_t \mathbf{u}_t^{(j)}. \tag{25}$$

where the expression of $\mathbf{u}_t$ is provided by equation (21). In this way, each element $\mathbf{u}_t^{(j)}$ of the hidden representations (for $t = 1, \ldots, \tau$) learns to detect specific features of the input data within all the interval $[1, \tau]$.

Finally, another approach, which we do not use here, is to consider the RNN as a transducer that produces an output sequence $\hat{\mathbf{y}}_t$ for $t = 1, \ldots, \tilde{\tau}$ (generally $\tilde{\tau} \neq \tau$) in correspondence of the input sequence $\mathbf{x}_t$ with $t = 1, \ldots, \tau$ [48, 82, 94].

A.7. Implementation of the machine learning algorithms

All the ML-models are realized in *PyTorch* and have been trained on the six different data sets using a DELL® Precision Tower workstation with one NVIDIA® TITAN RTX® GPU with 10 Gb of memory, 88 cores Intel® Xeon® CPU E5-2699 v4 at 2.20 GHz and 94 Gb of RAM.

We train the ANN models in mini-batches of dimension 16 by means of the SGD using ADAM [76] and learning rate $\eta = 10^{-3}$. We optimize the hyperparameters ξ with *ASHA* [95] as scheduler and *Hyperopt* [96, 97] (Hyperopt belongs to the family of Bayesian optimization algorithms) as search algorithm in the framework *Ray Tune* [98]. For the MLP models, the hyperparameters optimization defines: (i) the activation functions to be used, (ii) the number of layers, and (iii) their dimension, within the following search space: $\sigma \in \{\text{relu}, \text{sigmoid}, \tanh\}$, $L \in \{2, 3, 4, 5, 6\}$ and $\dim(\mathbf{h}[1]) \equiv ...\equiv \dim(\mathbf{h}[L]) \in \{d \in \mathbb{N} \mid 1 \leqslant d \leqslant 512\}$. Instead, for the RNN models the search space is $L \in \{1, 2, 3, 4\}$ for the number of recurrent layers ($L \in \{1, 2, 3, 4, 5, 6\}$ for the **NM** task with $t_{15} = 0.1$), and $\dim(\mathbf{h}_t[1]) \equiv \dim(\mathbf{h}_t[1]) \equiv ...\equiv \dim(\widetilde{\mathbf{h}}_t[L]) \equiv \dim(\widetilde{\mathbf{h}}_t[L]) \in \{d \in \mathbb{N} \mid 1 \leqslant d \leqslant 512\}$ for the layers dimension. Regarding the ML-models **m-biGRU-att** and**m-biLSTM-att**, the search space includes also the dimension of the attention layer as in equations (22) and (23), i.e., $\dim(\mathbf{c}) \equiv \dim(\mathbf{v}_1) \equiv ...\equiv \dim(\mathbf{v}_\tau) \in \{d \in \mathbb{N} \mid 1 \leqslant d \leqslant 512\}$. In the hyperparameters optimization of all the MLP and the RNN models, we have also used regularization methods as *weight decay* [99] and *dropout* [100]. They are able to mitigate overfitting; in particular, the former adds a penalty (chosen among $\{0, 10^{-4}, 10^{-3}\}$) to the risk function $L_S(\theta, \xi)$ with the aim to discourage large weights. Instead, using dropout, the outputs of the artificial neurons during the training are forced to zero with a probability among $\{0, 0.2, 0.5\}$.

ORCID iDs

Stefano Martina https://orcid.org/0000-0001-6024-1752
Stefano Gherardini https://orcid.org/0000-0002-9254-507X
Filippo Caruso https://orcid.org/0000-0002-8366-4296

References

[1] Cole J H and Hollenberg L C L 2009 Scanning quantum decoherence microscopy *Nanotechnology* **20** 495401
[2] 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** 565–70
[3] Degen C L, Reinhard F and Cappellaro P 2017 Quantum sensing *Rev. Mod. Phys.* **89** 035002
[4] Szańkowski P *et al* 2017 Environmental noise spectroscopy with qubits subjected to dynamical decoupling *J. Phys. Condens. Matter* **29** 333001
[5] Breuer H-P and Petruccione F 2007 *The Theory of Open Quantum Systems* (Oxford: Oxford Academic) (https://doi.org/10.1093/acprof:oso/9780199213900.001.0001)
[6] Caruso F, Giovannetti V, Lupo C and Mancini S 2014 Quantum channels and memory effects *Rev. Mod. Phys.* **86** 1203
[7] Müller M M, Gherardini S and Caruso F 2016 Stochastic quantum Zeno-based detection of noise correlations *Sci. Rep.* 38650
[8] Gherardini S Noise as a resource *PhD thesis* University of Florence, Italy arXiv:1805.01800
[9] 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
[10] 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
[11] Kofman A G, Kurizki G and Opatrný T 2001 Zeno and anti-zeno effects for photon polarization dephasing *Phys. Rev.* A **63** 042108
[12] Virzí S *et al* 2022 Quantum zeno and anti-zeno probes of noise correlations in photon polarization *Phys. Rev. Lett.* **129** 030401
[13] 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
[14] Ball H, Stace T M, Flammia S T and Biercuk M J 2016 Effect of noise correlations on randomized benchmarking *Phys. Rev.* A **93** 022303
[15] Norris L M, Paz-Silva G A and Viola L 2016 Qubit noise spectroscopy for non-Gaussian dephasing environments *Phys. Rev. Lett.* **116** 150503
[16] 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
[17] Müller M M, Gherardini S and Caruso F 2018 Noise-robust quantum sensing via optimal multi-probe spectroscopy *Sci. Rep.* 1–17
[18] Sung Y *et al* 2019 Non-Gaussian noise spectroscopy with a superconducting qubit sensor *Nat. Commun.* **10** 1–8
[19] Krzywda J, Szańkowski P and Cywiński Ł 2019 The dynamical-decoupling-based spatiotemporal noise spectroscopy *New J. Phys.* **21** 043034
[20] Niu M Y *et al* 2019 Learning non-Markovian quantum noise from Moiré-enhanced swap spectroscopy with deep evolutionary algorithm arXiv:1912.04368
[21] Müller M M, Gherardini S, Dalla Pozza N and Caruso F 2020 Noise sensing via stochastic quantum Zeno *Phys. Lett.* A **384** 126244
[22] Youssry A, Paz-Silva G A and Ferrie C 2020 Characterization and control of open quantum systems beyond quantum noise spectroscopy *npj Quantum Information* **6** 95
[23] Álvarez G A and Suter D 2011 Measuring the spectrum of colored noise by dynamical decoupling *Phys. Rev. Lett.* **107** 230501
[24] Yuge T, Sasaki S and Hirayama Y 2011 Measurement of the noise spectrum using a multiple-pulse sequence *Phys. Rev. Lett.* **107** 170504
[25] Poggiali F, Cappellaro P and Fabbri N 2018 Optimal control for one-qubit quantum sensing *Phys. Rev.* X **8** 021059
[26] Cywiński Ł 2014 Dynamical-decoupling noise spectroscopy at an optimal working point of a qubit *Phys. Rev.* A **90** 042307
[27] Pozza N D, Gherardini S, Müller M M and Caruso F 2019 Role of the filter functions in noise spectroscopy *International Journal of Quantum Information* **17** 1941008
[28] Harper R, Flammia S T and Wallman J J 2020 Efficient learning of quantum noise *Nat. Phys.* **16** pages 1184–88

[29] Luchnikov I A, Vintskevich S V, Grigoriev D A and Filippov S N 2020 Machine learning non-Markovian quantum dynamics *Phys. Rev. Lett.* **124** 140502
[30] Fanchini F F, Karpat G, Rossatto D Z, Norambuena A and Coto R 2021 Estimating the degree of non-Markovianity using machine learning *Phys. Rev.* A **103** 022425
[31] 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
[32] Shalev-Shwartz S and Ben-David S 2014 *Understanding Machine Learning: From Theory to Algorithms* (Cambridge: Cambridge University Press) (https://doi.org/10.1017/CBO9781107298019)
[33] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (New York: Springer) (https://doi.org/doi.org/10.1007/978-0-387-84858-7)
[34] Rivas A, Huelga S F and Plenio M B 2014 Quantum non-Markovianity: characterization, quantification and detection *Rep. Prog. Phys.* (https://doi.org/10.1088/0034-4885/77/9/094001)
[35] Breuer H-P, Laine E-M, Piilo J and Vacchini B 2016 Colloquium: Non-Markovian dynamics in open quantum systems *Rev. Mod. Phys.* **88** 021002
[36] von Lüpke U *et al* 2020 Two-qubit spectroscopy of spatiotemporally correlated quantum noise in superconducting qubits *PRX Quantum* **1** 010305
[37] Pollock F A, Rodríguez-Rosario C, Frauenheim T, Paternostro M and Modi K 2018 Non-Markovian quantum processes: complete framework and efficient characterization *Phys. Rev.* A **97** 012127
[38] Milz S and Modi K 2021 Quantum stochastic processes and quantum non-Markovian phenomena *PRX Quantum* **2** 030201
[39] Gherardini S, Smirne A, Huelga S F and Caruso F 2022 Transfer-tensor description of memory effects in open-system dynamics and multi-time statistics *Quantum Science and Technology* **7** 025005
[40] Giarmatzi C and Costa F 2021 Witnessing quantum memory in non-Markovian processes *Quantum* **5** 440
[41] Figueroa-Romero P, Modi K, Harris R J, Stace T M and Hsieh M-H 2021 Randomized benchmarking for non-markovian noise *PRX Quantum* **2** 040351
[42] Kempe J 2003 Quantum random walks: an introductory overview *Contemp. Phys.* **44** 307–27
[43] Venegas-Andraca S E 2012 Quantum walks: a comprehensive review *Quantum Inf. Process.* **11** 1015–106
[44] Dalla Pozza N and Caruso F 2020 Quantum state discrimination on reconfigurable noise-robust quantum networks *Physical Review Research* **2** 043011
[45] Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)
[46] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge: MIT Press) http://www.deeplearningbook.org
[47] Schmidhuber J 2015 Deep learning in neural networks: an overview *Neural Netw.* **61** 85–117
[48] Goldberg Y 2017 *Neural network methods for natural language processing* (Cham: Springer) (*Synthesis Lectures on Human Language Technologies*) (https://doi.org/10.1007/978-3-031-02165-7)
[49] Morris J, Pollock F A and Modi K 2022 Quantifying non-Markovian memory in a superconducting quantum computer *Open Syst. Inf. Dyn.* **29** 2250007
[50] D'Errico C, Moratti M, Lucioni E, Tanzi L, Deissler B, Inguscio M, Modugno G, Plenio M B and Caruso F 2013 Quantum diffusion with disorder, noise and interaction *New J. Phys.* **15** 045007
[51] Viciani S, Gherardini S, Lima M, Bellini M and Caruso F 2016 Disorder and dephasing as control knobs for light transport in optical fiber cavity networks *Sci. Rep.* 1–11
[52] Harris N C *et al* 2017 Quantum transport simulations in a programmable nanophotonic processor *Nat. Photonics* **11** 447–52
[53] Paul W and Baschnagel J 2013 *Stochastic Processes* (Cham: Springer) vol. 1 (https://doi.org/10.1007/978-3-319-00327-6)
[54] Piacentini F *et al* 2017 Determining the quantum expectation value by measuring a single photon *Nat. Phys.* **13** 1191–4
[55] Hernández-Gómez S, Gherardini S, Poggiali F, Cataliotti F S, Trombettoni A, Cappellaro P and Fabbri N 2020 Experimental test of exchange fluctuation relations in an open quantum system *Physical Review Research* **2** 023327
[56] Russakovsky O *et al* 2015 Imagenet large scale visual recognition challenge *Int. J. Comput. Vision* **115** 211–52
[57] Krizhevsky A, Sutskever I and Hinton G E 2017 Imagenet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90
[58] Emmert-Streib F, Yang Z, Feng H, Tripathi S and Dehmer M 2020 An introductory review of deep learning for prediction models with big data *Artif. Intell.* **3** 4
[59] De Groot S R and Mazur P 1984 *Non-Equilibrium Thermodynamics* (Dover Publications)
[60] Preskill J 2018 Quantum computing in the NISQ era and beyond *Quantum* **2** 79
[61] Martina S, Buffoni L, Gherardini S and Caruso F 2022 Learning the noise fingerprint of quantum devices *Quantum Machine Intelligence* **4** 8
[62] Martina S, Gherardini S, Buffoni L and Caruso F 2022 Noise fingerprints in quantum computers: machine learning software tools *Software Impacts* **12** 100260
[63] Nokkala J, Arzani F, Galve F, Zambrini R, Maniscalco S, Piilo J, Treps N and Parigi V 2018 Reconfigurable optical implementation of quantum complex networks *New J. Phys.* **20** 053024
[64] Leedumrongwatthanakun S, Innocenti L, Defienne H, Juffmann T, Ferraro A, Paternostro M and Gigan S 2020 Programmable linear quantum networks with a multimode fibre *Nat. Photonics* **14** 139–42
[65] Paz-Silva G A, Norris L M, Beaudoin F and Viola L 2019 Extending comb-based spectral estimation to multiaxis quantum noise *Phys. Rev.* A **100** 042334
[66] 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 *Physical Review Applied* **14** 024021
[67] Glorot X, Bordes A and Bengio Y 2011 Deep sparse rectifier neural networks *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* 315–23
[68] Nair V and Hinton G E 2010 Rectified linear units improve restricted Boltzmann machines *Proceedings of the 27th International Conference on International Conference on Machine Learning* 807–14
[69] Hornik K *et al* 1989 Multilayer feedforward networks are universal approximators *Neural Netw.* **2** 359–66
[70] Caruana R, Lawrence S and Giles C L 2000 Overfitting in neural nets: backpropagation, conjugate gradient, and early stopping *Proceedings of the 13th International Conference on Neural Information Processing Systems* 381–7
[71] Bergstra J and Bengio Y 2012 Random search for hyper-parameter optimization *The Journal of Machine Learning Research* **13** 281–305
[72] Snoek J, Larochelle H and Adams R P 2012 Practical Bayesian optimization of machine learning algorithms *Proceedings of the 25th International Conference on Neural Information Processing Systems* Volume 2, 2951–9

[73] Bottou L 2010 Large-scale machine learning with stochastic gradient descent *Proceedings of COMPSTAT'2010, Springer* 177–86
[74] Zhang T 2004 Solving large scale linear prediction problems using stochastic gradient descent algorithms *Proceedings of the Twenty-first International Conference on Machine Learning* 116
[75] Bottou L, Curtis F E and Nocedal J 2018 Optimization methods for large-scale machine learning *SIAM Rev.* **60** 223–311
[76] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization arXiv:1412.6980
[77] Martina S, Ventura L and Frasconi P 2020 Classification of cancer pathology reports: a large-scale comparative study *IEEE Journal of Biomedical and Health Informatics* **24** 3085–94
[78] Martina S 2020 Classification of cancer pathology reports with deep learning methods *PhD thesis* University of Florence, Italy
[79] Tang D, Qin B and Liu T 2015 Document modeling with gated recurrent neural network for sentiment classification *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* 1422–32
[80] Jaeger H and Haas H 2004 Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication *Science* **304** 78–80
[81] Lipton Z C, Berkowitz J and Elkan C 2015 A critical review of recurrent neural networks for sequence learning arXiv:1506.00019
[82] Sutskever I, Vinyals O and Le Q V 2014 Sequence to sequence learning with neural networks *Proceedings of the 27th International Conference on Neural Information Processing Systems* Volume 2, 3104–12
[83] Hochreiter S 1998 The vanishing gradient problem during learning recurrent neural nets and problem solutions *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **6** 107–16
[84] Bengio Y, Simard P and Frasconi P 1994 Learning long-term dependencies with gradient descent is difficult *IEEE Trans. Neural Networks* **5** 157–66
[85] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80
[86] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* 1724–34
[87] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[88] Graves A, Mohamed A-R and Hinton G 2013 Speech recognition with deep recurrent neural networks *2013 IEEE international Conference on Acoustics, Speech and Signal Processing, IEEE* 6645–9
[89] Schuster M and Paliwal K K 1997 Bidirectional recurrent neural networks *IEEE Trans. Signal Process.* **45** 2673–81
[90] Bahdanau D, Cho K and Bengio Y 2015 Neural machine translation by jointly learning to align and translate *3rd International Conference on Learning Representations, ICLR 2015*
[91] Luong M-T, Pham H and Manning C D 2015 Effective approaches to attention-based neural machine translation *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* 1412–21
[95] 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 *Proceedings of Machine Learning and Systems* 2, 230–46
[96] Bergstra J S, Bardenet R, Bengio Y and Kégl B 2011 Algorithms for hyper-parameter optimization *Proceedings of the 24th International Conference on Neural Information Processing Systems* 2546–54
[97] Bergstra J, Yamins D and Cox D 2013 Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures *Proceedings of the 30th International Conference on International Conference on Machine Learning* Vol 28, 115–23
[98] 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
[99] Krogh A and Hertz J A 1992 A simple weight decay can improve generalization *Proceedings of the 4th International Conference on Neural Information Processing Systems* 950–7
[100] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *The Journal of Machine Learning Research* **15** 1929–58