Modeling Dynamics of Classical and Quantum Systems Using Machine Learning Techniques

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von

Francesco Carnazza

aus Lecco, Italien

Tübingen

2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	6.12.2024
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Igor Lesanovsky
2. Berichterstatterin:	Prof. Dr. Sabine Andergassen
3. Berichterstatter:	Prof. Dr. Martin Oettel

Summary

In this thesis, we examine the dynamics of many-body systems within both quantum and classical contexts. When dealing with many-body systems, the focus is typically on what are known as order parameters. These parameters indicate whether the physical system is in an ordered phase or not. Order parameters often correspond to the expectation values of local observables, or to the averages of the microscopic degrees of freedom within the system's configuration. Specifically, we leverage machine learning methodologies to develop an interpretable and physically consistent theoretical framework for describing local and macroscopic degrees of freedom within both quantum and classical contexts. In the case of quantum system, we develop a method that is able to approximate a "dynamical generator," that is, an operator which is able to evolve in time few body observables of the system. In particular, this generator preserves some important physical quantities of the state representing the system under investigation, such as its probabilistic interpretation. We explore the capabilities of this method in our **first** and **second scientific paper**. In the first work, we provide full access to the dynamics through the coherence vector, a representation of the quantum state. In our second work, more akin to actual experimental conditions, we supply the variational method with data obtained via state tomography through projective measurements, a routine approach in retrieving the state of quantum devices using qubits. Here, we further develop the methods used in our fist work to address the problem of learning the dynamical generator in the presence of projection noise. Given that projective measurements introduce noise into the dynamics, we naturally consider how to represent physical noise in a machine learning routine. To better understand how noise could be managed by a machine learning routine, we studied the reduced degrees of freedom of classical systems composed of many "binary" spin degrees of freedom whose evolution is probabilistic in our **third scientific paper**. Average quantities of these spin variables are inherently stochastic. By taking the expectation values over numerous stochastic trajectories, order parameters emerge. Specifically, we model the dynamics of this order parameter using a stochastic differential equation of Itô type. Such equations are first-order differential equations in time that include a directed force term, known as the "drift coefficient," and a noisy force term called the "diffusion coefficient." We concentrate on encoding the diffusion term and the drift term of this stochastic differential equation into two separate neural networks. This method proves robust in reproducing the dynamics and allows us to extract valuable information about the system, such as the critical point and the critical exponent.

Zusammenfassung

In dieser Arbeit untersuchen wir die Dynamik von Vielkörpersystemen sowohl im quantenmechanischen als auch im klassischen Kontext. Bei der Betrachtung von Vielkörpersystemen liegt der Fokus typischerweise auf sogenannten Ordnungsparametern. Diese Parameter zeigen an, ob sich das physikalische System in einer geordneten Phase befindet oder nicht. Ordnungsparameter entsprechen oft den Erwartungswerten lokaler Observablen oder den Mittelwerten der mikroskopischen Freiheitsgrade in einer Konfiguration des Systems. In dieser Arbeit nutzen wir Methoden des maschinellen Lernens, um interpretierbare und physikalisch konsistente Bewegungsgleichungen zu finden, die lokalisierte und makroskopische Freiheitsgrade sowohl im quantenmechanischen als auch im klassischen Kontext beschreiben. Im Fall eines Quantensystems entwickeln wir eine Methode, die in der Lage ist, einen lokalen "dynamischen Generator" zu approximieren, also einen Operator, der wenige lokalisierte Observablen des Systems in der Zeit entwickelt. Insbesondere bewahrt dieser Generator einige wichtige physikalische Eigenschaften des Zustands des zu untersuchenden Systems, wie z. B. seine probabilistische Interpretation. In unserer ersten und zweiten wissenschaftlichen Arbeit erforschen wir die Einsatzbereiche dieser Methode. In der ersten Arbeit nehmen wir an, dass wir vollständigen Zugang zum Quantenzustand haben. In unserer zweiten Arbeit, die eher experimentellen Bedingungen entspricht, benutzen wir Daten, die über Zustandstomographie durch projektive Messungen gewonnen wurden. Dies ist ein Routineverfahren zur Bestimmung des Zustands von aus Qubits zusammengesetzten Quantensystemen. Hier entwickeln wir die Methoden aus unserer ersten Arbeit weiter, um das Problem des Lernens des dynamischen Generators unter dem Einfluss von Projektionsrauschen zu behandeln. Dabei untersuchen wir, wie physikalisches Rauschen in im maschinellen Lernalgorithmus implementiert werden kann. Um besser zu verstehen, wie Rauschen in einem maschinellen Lernverfahren behandelt werden kann, untersuchen wir in unserer dritten wissenschaftlichen Arbeit die reduzierten Freiheitsgrade klassischer Systeme, die aus vielen "binären" Spinfreiheitsgraden bestehen, deren Entwicklung probabilistisch ist. Durchschnittswerte dieser Spinvariablen sind von Natur aus stochastisch, und die Mittelung über zahlreiche stochastische Trajektorien ist hier zur Berechnung eines Ordnungsparameters notwendig. Konkret modellieren wir die Dynamik eines solchen Ordnungsparameters mittels einer stochastischen Differentialgleichung vom Itô-Typ. Solche Gleichungen sind zeitliche Differentialgleichungen erster Ordnung, die einen gerichteten Kraftterm, parametrisiert durch den sogenannten ”Driftkoeffizienten“, sowie einen rauschbehafteten Kraftterm, bestimmt durch den ”Diffusionskoeffizienten“, enthalten. Wir konzentrieren uns in dieser Arbeit darauf, den Diffusionsterm und den Driftterm dieser stochastischen Differentialgleichung in zwei separaten neuronalen Netzwerken zu codieren. Diese Methode erweist sich als robust bei der Reproduktion der Dynamik und ermöglicht es uns, wertvolle Informationen über das kollektive Verhalten des Systems zu extrahieren, wie den kritischen Punkt und kritische Exponenten.

List of publications and personal contribution

List of publications

1. F. Carnazza, F. Carollo, D. Zietlow, S. Andergassen, G. Martius and I. Lesanovsky *Inferring Markovian quantum master equations of few-body observables in interacting spin chains* New J. Phys. **24**, 073033 (2022)

2. G. Cemin, F. Carnazza, S. Andergassen, G. Martius, F. Carollo, and I. Lesanovsky *Inferring interpretable dynamical generators of local quantum observables from projective measurements through machine learning* Phys. Rev. Applied **21**, L041001 (2024)

3. F. Carnazza, F. Carollo, S. Andergassen, G. Martius, M. Klopotek, I. Lesanovsky *Machine learning stochastic differential equations for the evolution of order parameters of classical many-body systems in and out of equilibrium* arXiv:2402.03913 (2024)

Personal contribution of the candidate

No.	Author position	Scientific ideas (%)	Data generation (%)	Analysis and interpretation (%)	Paper writing (%)	Status
1	1	50	100	70	80	Published
2	2	50	10	50	30	Published
3	1	80	100	70	50	Published

Acknowledgments

I would like to express my deepest gratitude to Prof. S. Andergassen and Prof. I. Lesanovsky for their supervision throughout the completion of this thesis. My sincere thanks also go to M. Magoni for the enlightening discussions and genuine friendship. I am especially thankful to A. Cabot, A. Cogo, A. Royo, W. Santana Martins, and P. J. Paulino de Souza for their many stimulating discussions and for the warmth and friendship they have shown me. I also want to extend my appreciation to J. Wilkinson, G. Perfetto, B. Olmos-Sanchez, C. Nill, M. Cech, M.B. Svendsen, and M. Boneberg. My thanks go to G. Cemin, S. Walch, and J.E. Fitzner, with whom I had the privilege to collaborate on your thesis projects—you taught me so much. I am equally grateful to my co-authors G. Martius, M. Klopotek, and D. Zietlow for their contributions. Special thanks to I. Estiry and A. Lorösch as well. I owe particular gratitude to F. Carollo for his unwavering patience, deep interest, and invaluable assistance. Lastly, I am forever grateful to my parents, Paola and Eugenio, and my sister, Sara, for making my life so funny and for your support.
And to Chiara, I feel so incredibly fortunate—thank you.

Contents

1 Introduction

1.1 Many-body systems

One of the most captivating areas in modern physics concerns the understanding of the collective behavior that arises from the interactions of a vast number of particles [1, 2, 3]. This is because the interactions among many particles can lead to ”emergent phenomena”. Emergent phenomena are collective behaviors and properties that arise from the (complex) interactions within many-body systems and cannot be straightforwardly predicted from the fundamental laws governing individual particles [4, 5]. An illustrative example of such emergent behavior is crystalline solids’ (beautiful) formations. These formations arise due to large quantities of interacting atoms. Here, we see that new properties and behaviors, i.e. a breaking of spatial symmetry, emerge due to the collective interactions within a many-body system. These properties cannot be directly inferred from those observed of just two atoms: as summarized by the famous quote of Anderson, “more is different” [4].

The emergent behavior of many-body systems is of primary concern in the fields of condensed matter physics and statistical mechanics. In fact, these disciplines encompass a broad spectrum of phenomena and theoretical frameworks, making significant contributions to both material science and fundamental physics providing deep insights into the properties of materials and their underlying physical principles [6]. In principle, classical physics offers a formalism to describe the equations of motion for the positions and momenta of a large number of particles through Hamiltonian mechanics. However, solving these equations of motion is extraordinarily challenging, even for a system as seemingly simple as three interacting particles [7]. The complexity and computational difficulty grow exponentially with the number of particles involved, necessitating alternative approaches [8]. For example, instead of solving exact equations, the general properties of the system, such as its relaxation time to its stationary state or its equilibrium state, when existing, can be described by probabilistic methods: therefore, probabilistic and or statistical methods become essential. In classical systems, the approach of statistical mechanics is to select an observable and consider that the dynamical evolution of the systems’ particles probabilistically affects this observable. Probability plays a fundamental role in quantum mechanics [9]. In a wide range of scenarios of theoretical and practical importance, such as the behavior of materials at very low temperatures, the classical treatment of the system fails, necessitating a quantum mechanical description of the system. This means that some of the methods of simulations that can be safely used in classical systems can fail or must be revised, such as Monte Carlo simulations [10, 11]. In comparison with classical systems, additional correlations between particles arise in quantum systems, due to entanglement [12, 13, 14]. Entanglement occurs in quantum systems when particle can only be described globally, by means of a so called many-body wave-function, which is akin to the square of a probability distribution describing their configuration. A physical configuration includes the spatial arrangement of electrons in an atom, the projection of a spin operator on a given axis for a spin system, and possibly other quantum numbers such as angular momentum, orbital number etc.. Because each configuration has a nonzero probability of being measured, the amount of information the wave function carries tends to grow faster than polynomially with the system size. This results in a probability distribution with an extremely large support, typically growing exponentially with the number of particles involved. Consequently, developing efficient methods for storing and managing this information is of great experimental and theoretical interest [15].

1.2 Phase transitions and critical phenomena

One of the main concepts of many-body system in statistical mechanics is the so-called "thermodynamic limit", [8, 7]. Another is the transition from magnetic to paramagnetic state in ferromagnetic materials such as iron. In this transition, the state of the material moves from an ordered (ferromagnetic) state, where its dipoles point in the same direction, to a disordered (paramagnetic) state at the Curie temperature.

The mathematical framework in which this phenomena are phrased and understood is the renormalization group [16]. The explanation that this general framework gives for the anomalies in the physical observables, and in particular for the divergence of correlation lengths, is that near the critical point, the behavior of the system is self-similar at different time and space scales. This is effectively translated mathematically in the *scaling hypothesis* [17, 18]. This hypothesis states that at the critical point physical observables, such as the correlation length and susceptibility, can be described by homogeneous functions known as scaling functions. These functions exhibit power law divergences at the critical point. The exponents of these power laws are termed "critical exponents". Empirical observations [19] have shown that these critical exponents can be remarkably consistent across vastly different physical phenomena. This surprising consistency led to the concept of "universality classes," where models that share the same scaling functions and critical exponents are considered to belong to the same class. In this thesis, in particular in [20], we set out to study the directed force acting on an order parameter in classical models of statistical mechanics. This force can be interpreted as the gradient of a "dynamical function" or, in equilibrium systems, the free energy. The properties of this function, particularly its zeros, are expected to scale according to power laws, linking to the scaling functions utilized in the scaling hypothesis. Our paper [20] (see also Sec. 2.1)is aimed at using machine learning techniques to approximate the dynamical generators of the system, that is, the equations governing the time evolution of macroscopic physical observables of the system. This generators are related to the scaling functions and allow to gain insight into the physics of the problem.

1.3 Machine learning dynamics of reduced observables

As already mentioned in the summary, the main scope of this thesis is to explore the capabilities of machine learning methods to approximate generators of the dynamics of order parameters in quantum and classical systems. Machine learning employs a variety of variational trial functions, or Ansätze, which can be trained for different tasks, from distinguishing images of cats and dogs [21, 22] to approximating the ground state of a quantum many-body problem [23, 24]. A key aspect of machine learning routines is that the optimization procedure makes use of automatic differentiation to compute the gradients of a loss function, which measures how well the variational Ansatz predicts the desired output. The loss function is either minimized or maximized along the direction of larger variation dictated by the gradients. This optimization is typically performed using stochastic gradient descent, which estimates the loss function by stochastically sampling from the set of desired outputs, enabling efficient handling of large datasets, and eschewing problem associated with local minima, by allowing the loss to “jump” past them. Among the most common variational Ansatz in machine learning are neural networks, which come in various forms, such as fully connected feed-forward networks, convolutional neural networks, and tensor networks (also known as tensor trains). Notably fully connected feed-forward networks, consist of interconnected linear transformations and nonlinear activation functions inspired by biological neural networks.

Despite their biological inspiration, the primary interest in artificial neural networks today is computational and mathematical. Universal approximation theorems [25, 26] guarantee that these neural networks can approximate a wide range of functions, provided they have enough variational parameters. However, a significant drawback of neural networks is the difficulty in determining an upper bound on the number of variational parameters needed, which is crucial to avoid overfitting—a common issue where the network learns features specific to data used to optimize it rather than generalizable patterns. Machine learning methods are appealing in physics because they allow to treat phenomena that at first sight are disparate, in an unified manner. In our works [27] and [28], we employed models that are physically constrained to minimize the number of variational parameters, enhancing the interpretability of the model. These models are based on “one-layer feed-forward neural networks”. In contrast, the method we develop in [20] uses deep feed-forward neural networks, sacrificing some interpretability for greater representation power and the ability to approximate a broader class of functions. In particular, we learn the equation of motion governing some noisy processes in statistical mechanics. The fact that we employ differentiable functions, allows us to easily find the minima of the effective potential. This effective potential governs the relaxation times and the value of the stationary point. In all three studies, the application of machine learning techniques facilitated the retrieval of a ″dynamical generator″ employed to predict distinct classes of physical dynamics. The following subsections provide a more detailed explanation of the methods utilized and the specific physical scenarios.

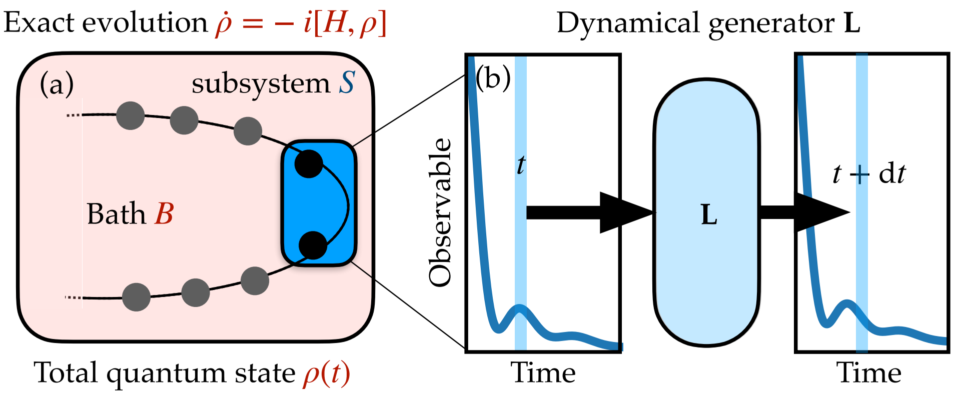
Exact evolution $\dot{\rho} = -i[H,\rho]$
Dynamical generator **L**
(a)
subsystem S
Bath B
Total quantum state $\rho(t)$
(b)
Observable
t
Time
L
$t + \mathrm{d}t$
Time

Figure 1: **The architecture for the dynamical generator L** The full quantum state $\rho(t)$ of a spin chain evolves according to the Hamiltonian H. A complete basis of observables, specifying the reduced state of a two-spin subsystem, is used to optimize the variational parameters of a "dynamical generator" **L**.

1.3.1 Inferring quantum master equations in quantum spin chains

The dynamics of open quantum systems is an active area of research, driven by recent advancements in experimental realizations of artificial quantum systems, such as Rydberg and utra-cold atoms [29, 30, 31, 32]. In this thesis, we investigate how machine learning techniques can be utilized to infer some of the properties of open quantum systems [33, 24, 34, 35, 36, 23, 37, 38]. Specifically, we use a single-layer network whose parameters mimic the form of a dynamical generator **L**. The functional form and the properties of this generator will be described in detail in Sec. 2.2.6. We allow each of the variational parameters in our single-layer model to represent an entry in the generator **L**. Our approach is data-driven, training the variational generator to align with the reduced dynamics of a two-body spin system within a larger environment. The spins under consideration are spin-$\frac{1}{2}$ particles which can stay in two states "up" $|\uparrow\rangle$, or "down" $|\downarrow\rangle$. Their state is described by a "coherence vector" formed by expectation values of local correlation functions of just two spins — see. Fig. 1 [a], which form a subsystem S. The degrees of freedom of the other spins are overlooked, and serve as a source of dissipation, or "bath" B. We obatin data by evolving the quantum state ρ of the entire system $S + B$ according to an Hamiltonian H, (cf. Fig 1): $\dot{\rho} = -i[H, \rho]$, using exact diagonalization. This global Hamiltonian comprises both the bath and the subsystem. At each discrete time step, we trace out the bath's degrees of freedom and calculate the coherence vector from the quantum state of the subsystem. The network excels at predicting time windows included in the training data, but its accuracy diminishes over longer time periods. The learned generator performs best at predicting dynamics when the coupling between spins in the Hamiltonian is weak, but its accuracy declines at higher coupling strengths. This decline is not due to a flaw in the network but rather because the Markovian description of the system becomes invalid at these higher couplings. The generator is interpretable: for instance, one can obtain the predicted relaxation time of the system as the smallest (in absolute value) nonzero real part of the eigenvalues of the generator. This quantity qualitatively predicts the time at which, after a transient period, the system reaches an almost stationary value. Due to finite size effects, small oscillations are still present in the exact dynamics at all times, whereas the dynamics predicted by our generator relax to a proper stationary state.

1.3.2 Inferring quantum dynamics via projective measurements

In our scientific paper [28], we used noisy synthetic data from larger spin chains to train the dynamical generator as in our work [27]. To obtain this data, we used tensor network methods instead of exact diagonalization. These methods [40, 41] enable the simulation of slightly entangled spin models [42] for up to tens or even hundreds of particles. They achieve this by efficiently storing each coefficient of the wave function as a product of tensors, with each tensor associated with a single particle. By truncating the rank of these tensors during time evolution, the amount of information required to simulate these quantum systems scales only linearly with the system size. Specifically, we reconstructed the density matrix by first performing the evolution via the Time-Evolving Block Decimation (TEBD) algorithm [40], which allows us to handle larger spin systems than in our initial work, storing the wave function of the system as a tensor network. This is crucial because a Lindbladian description of the subsystem is expected to be more appropriate for larger baths. Additionally, we did not limit the network input to exact expectation values as done in our previous work. To validate our approach in more realistic experimental settings, we selected M random times in the simulation. At each time, we performed a set of N measurements in each basis, and from these measurements, we estimated the value of the density matrix (cf. Fig. 2).

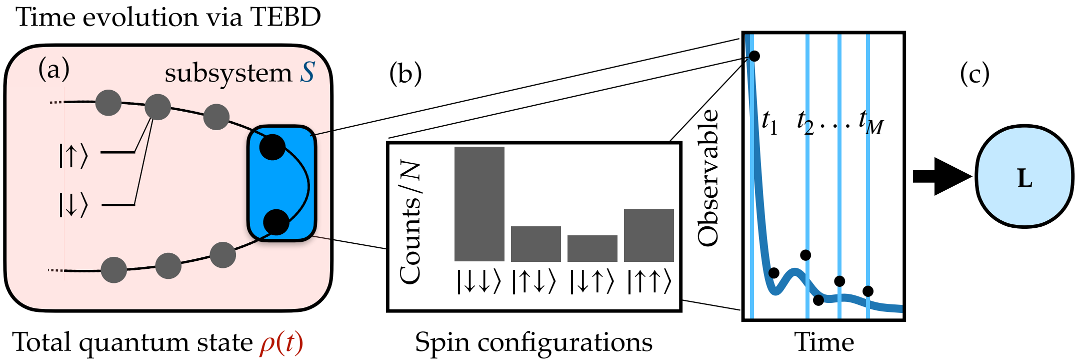

Figure 2: **Machine learning the generator L of quantum dynamics subject to projective noise.** In the scientific paper [39], the dynamics are approximated using the TEBD algorithm, which allows for the simulation of a larger spin chain [see figure (a)]. At each of a set of M randomly selected times, the system's state is estimated by performing a series of N measurements. The results of these measurements are then used to reconstruct the state (illustrated in histogram (b)). For instance, if the observable is the spin correlation function between two adjacent spins, such as $\langle\sigma_x^1\sigma_y^2\rangle$, then $|\uparrow\downarrow\rangle$ indicates that the measurement of the first spin along the x-axis yields an outcome of 1, while the measurement of the second spin along the y-axis gives an outcome of 0. Finally, the dynamical generator **L** [cf. Fig. 1] is trained using synthetic data to best replicate the system's dynamics [shown in (c)]. The form of the observables σ_x, σ_y is explained in detail in Sec. 2.2.7.

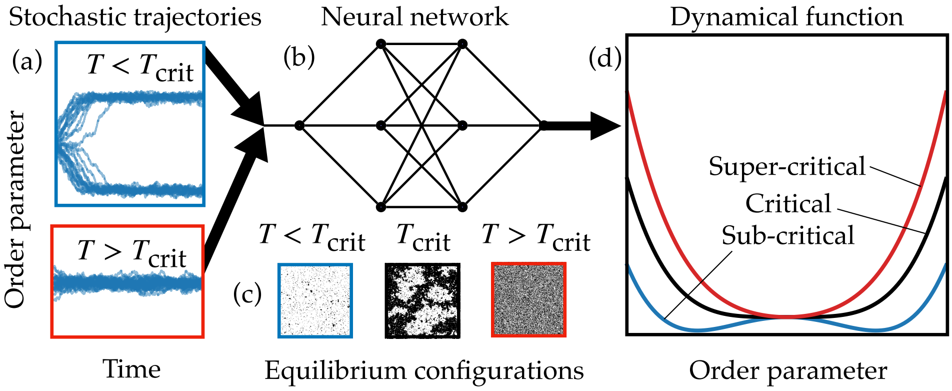

Figure 3: **Machine learning of the dynamical function for the two-dimensional Ising model.** The stochastic evolution of the order parameter (a) is fed to a neural network (b) that learns the form of a "dynamical function" (d). Similarly to a potential in classical mechanics, the minima of this function specify the stationary values for the order parameter, corresponding to those of the configurations in equilibrium (c). In the case of the Ising model, for sub-critical temperatures, $T < T_{\text{crit}}$, the stationary value of the order parameter is finite, while in the case of higher temperatures $T > T_{\text{crit}}$, the order parameter vanishes. This is reflected in the functional form of the dynamical function, which exhibits two distinct minima at sub-critical temperatures, but shows a single minimum at zero for super-critical temperatures.

This density matrix was then used to reconstruct "noisy" coherence vectors, where the noise arises from the probabilistic nature of the measurements. For a spin system, the measurements involved projections on the spatial axes. Due to the smaller and noisier dataset used for training, we added an additional regularization term to the loss function in this case. Similarly to our previous work, we devised an error measure between the exact and predicted dynamics to determine the regions in time and Hamiltonian coupling strength where the method can be used reliably.

1.3.3 Machine learning stochastic differential equations for order parameters

The dynamics of open quantum systems is closely linked to the Markovian master equation formalism, which connects the Fokker-Planck equation with the Itô formalism, as described in detail in Sec. 2.1.5. This connection inspired our exploration of encoding stochastic dynamics within neural networks. Drawing on the approach of Kidger et al. [43, 44, 45, 46], we aimed to learn the drift and diffusion terms of an Itô equation. Unlike their methods, we do not learn these quantities simultaneously; instead, we solve two separate ordinary differential equations. One equation represents the first moment over many trajectories of the random variable of interest, thereby learning the directed force, while the other equation represents the second moment, thereby retrieving the form of the diffusion term. The primary focus of this work is the dynamics of order parameters in the dynamical Ising model and the contact process [47, 48]. These paradigmatic models of statistical mechanics are defined in terms of ”digital” spins on a lattice, meaning that each spin variable can only take one out of two possible values at each time step. While the Ising model is an equilibrium model, with the distribution of its stationary configurations dictated by the Boltzmann distribution (see Fig. 3 (c)), the contact process is defined by a set of dynamical rules for updating the values of the spins over time and lacks a notion of free energy. We provide the networks with the average value of the spins, computed from the simulated dynamics obtained through Monte Carlo simulations. The networks can get a very good result for the drift term, which allows to use these networks to infer the critical point and the critical exponent of the order parameter. To evaluate the accuracy of the retrieved diffusion coefficient, we used it alongside the drift to numerically integrate the dynamics and compared the result with the original dynamics used for training.

1.4 Structure

The thesis is structured as follows. In chapter 2 we introduce the general class of physical models we are interested in studying and the tools necessary for their theoretical understanding. The first section of this chapter is devoted to statistical mechanics and stochastic processes, while the second section deals with the theory of open quantum systems. The theory of stochastic processes in physics is of particular importance for our third work [20], where the stochastic equations emerging from a few paradigmatic models of statistical mechanics are retrieved and analyzed using machine learning methods. The theory of open quantum systems is of main interest in the case of our first and second work [27, 39], where the objects of study are few-body observables subject to dissipative dynamics. In chapter 3 we present the machine learning tools employed in the analysis of the physical systems of interest. In particular, we cover the topics of artificial neural networks, which are used in our works, and describe the standard tools of stochastic gradient descent and back-propagation, which are used to optimize our variational trial functions, that we use to retrieve the generators of the dynamics. In chapter 4 we summarize the results presented in detail in the papers. Finally, in chapter 5 we present the conclusion of the thesis and provide some future directions.

2 Theoretical Background

2.1 Stochastic processes in physics

2.1.1 Introduction

The role of noise in physics is fundamental, theoretically, experimentally and computationally [49, 50, 51]. Experimental data is naturally subject to noise. In many areas of physics, such as in classical systems, this is due to the fact [49] that any observed system is interacting with a large number of *discrete* particles [52]. These particles interact deterministically among themselves, but keeping track of their motion is unfeasible, and their collective effect can only be modeled in a probabilistic fashion. This is why in statistical mechanics the macroscopic dynamics of many-body systems, such as the one of order parameters, is often modelled by means of "stochastic differential equations", where a deterministic drift term in the time evolution of a random variable is complemented by a noisy force, also-called diffusion term, whose value at each point in space and time is extracted according to a probability distribution. Adding a noisy term, even small, can have strong consequences on the behavior of macroscopic quantities such as order parameters or fields, with respect to the treatment where only the deterministic term is considered. For example, "unstable" stationary points of a deterministic dynamics would vanish when adding a stochastic term. Moreover, noise is an intrinsic feature of an important class of numerical methods used to simulate many-body systems, namely, Markov Chain Monte Carlo methods. These methods are iterative. At each iteration, an attempt is made to update each degree of freedom within the system in a probabilistic manner. As a result, the physical quantities computed from these simulations are influenced by probabilistic fluctuations [53, 54, 55]. In this thesis, specifically in our paper [20], we develop a method to compute the drift and diffusion term for data retrieved from Markov Chain Monte Carlo simulations. In this section, we present the general theory for stochastic processes in physics, with a focus on how to obtain the diffusion and the drift term from the dynamics itself.

2.1.2 Markovian master equations

Stochasticity in classical physics arises from the random collisions of a large number of discrete particles. This is due to the fact that these collisions are discrete events, causing abrupt changes in, for example, the direction of motion of the particles.

Given that the time interval between two observations is necessarily much longer than the time interval between two consecutive interactions, the position of the particle can be modeled as a continuous variable that is effectively nowhere differentiable. Moreover, the collisions experienced by the observed particle are modelled as random events. The position of a particle subject to random collisions is a *random process*. A random process in physics [49] is a random variable Z_h depending on time t_h, with an associated probability distribution P, that in general depends on all previous observation of the variable at times $t_1 < t_2 < ... < t_h$:

$$P(Z_h, t_h) = P(Z_h, t_h | Z_1, t_1; ...; Z_{h-1}, t_{h-1}). \tag{1}$$

When the random variable Z_t represents observations of a single particle experiencing numerous random interactions within a vast environment, it is reasonable to assume that the particle's relaxation time is significantly slower than that of the environment. In other words, the dynamics of this single particle are a minor, negligible disturbance to the noise-generating environment of particles. To model the fact that the particle is a minor disturbance to the environment, the assumption is made that this environment "loses memory" on a scale that is much larger then the relaxation time of the particle. When this fact holds, the process is *Markovian*. This is to say, the process can be described by a random variable Z, whose probability distribution has the Markov property. The Markov property for discrete states that the probability P of extracting a random variable Z at time t only depends on its current state, and not on its history, that is, on which variables where extracted at previous times. More precisely, suppose a random process Z_t is measured at times $t_1 < t_2 < ... < t_{N-1}$, with outcomes Z_i at each of the times t_i. Then the conditional probability of obtaining a value Z_N at time $t_N > t_{N-1}$ is [49]:

$$P(Z_N, t_N | Z_1, t_1; ...; Z_{N-1}, t_{N-1}) = P(Z_N, t_N | Z_{N-1}, t_{N-1}). \tag{2}$$

In particular, for three times $t_1 < t_2 < t_3$ [49]:

$$P(Z_1, t_1; Z_2, t_2; Z_3, t_3) = P(Z_1, t_1)P(Z_2, t_2|Z_1, t_1)P(Z_3, t_3|Z_2, t_2). \tag{3}$$

Dividing both sides in last equation by $P(Z_{t_1})$ and integrating over Z_{t_2} one gets the Champman-Kolmogorov equation for the probability $P(Z_3, t_3|Z_1, t_1)$:

$$P(Z_3, t_3|Z_1, t_1) = \int dZ_2 P(Z_3, t_3|Z_2, t_2)P(Z_2, t_2|Z_1, t_1). \tag{4}$$

The quantities $P(Z_i, t_i|Z_j, t_j)$ in the last equations are called the transition probabilities of the process.

2.1.3 The Fokker-Planck equation

The form of $P(Z_i, t_i|Z_j, t_j)$ in Eq. (4), is a self-convolution, so a natural solution to it is given in terms of transition probabilities that are Gaussian in Z [49]. A possible solution is:

$$P_{\mathrm{B}}(Z, t|Z_0, t_0) = \frac{1}{\sqrt{2\pi(t - t_0)}} \exp\left[-\frac{(Z - Z_0)^2}{2(t - t_0)}\right]. \tag{5}$$

This process for Z models the position Z in time t of a particle whose motion is purely chaotic, i.e. a "Brownian particle" [56], [cf. Sec 2.1.4]. For initial conditions $P(Y, 0) = \delta(Y)$ this lead to a probability

$$P_{\mathrm{B}}(Z, t) = \frac{1}{\sqrt{2\pi t}} \exp\left[-\frac{Z^2}{2t}\right]. \tag{6}$$

This probability distribution explicitly depends on time. To obtain a probability distribution $P(Z, t)$ that is time independent, $P(Z, t) = P(Z)$, the Gaussian solution of Eq. (4) must have, thank to a theorem of Doob [57], the following form, for $u = t - t_0$:

$$P_{\text{O-U}}(Z, t|Z_0, t_0) = P_u(Z|Z_0) = \frac{1}{\sqrt{2\pi(1 - e^{-2u})}} \exp\left[-\frac{(Z - Z_0 e^{-u})^2}{2(1 - e^{-2u})}\right]. \tag{7}$$

In this case, the probability $P(Z)$ becomes:

$$P_{\text{O-U}}(Z, t) = P(Z) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{Z^2}{2}\right]. \tag{8}$$

This means that here, the transition probabilities of the process depend only on the time difference u, like in Eq. (5), but the corresponding probability $P_{\text{O-U}}(Z, t)$ is completely time-independent. This type of process was first invented to describe the velocity of a Brownian particle, and it is known as Ornstein-Uhlenbeck process [58]. Processes of this kind, whose probability $P(Z)$ is time-independent, and whose transition probabilities depend only on the interval of time u, are deemed *stationary*. A very important property of stationary processes is that they are not conditional on time, contrary to those appearing in the Chapman-Kolmogorov equation (4). In fact, the Chapman-Kolmogorov equation, being an integral equation for conditional probabilities, is very difficult to handle, but for stationary processes, a much simpler differential equation, namely, the Fokker-Planck equation, may be devised. Moreover, stationary processes are of crucial importance in physics, as one expects a large class of observed phenomena to reach a stationary state [49, 59]. To get a flavor of how a differential equation arises for stationary processes let us note that at $u = \infty$, Eq. (7) satisfies the usual differential equation for a Gaussian:

$$0 = ZP_u(Z|Z_0) + \frac{\partial}{\partial Z}P_u(Z|Z_0). \tag{9}$$

For nonvanishing values of u, one can check by directly plugging in the expression Eq. (7), that the differential equation satisfied by the transition probability of the Ornstein-Uhlenbeck process is:

$$\frac{\partial}{\partial u}P_u(Z|Z_0) = \frac{\partial}{\partial Z}\Big[ZP_u(Z|Z_0) + \frac{\partial}{\partial Z}P_u(Z|Z_0)\Big]. \tag{10}$$

This is the Fokker-Planck equation for the Ornstein-Uhlenbeck process. In what follows we derive the general form of the Fokker-Planck equation starting from the Chapman-Kolmogorv equation, following the standard treatment of [49].

In the case of stationary Markov processes, it is possible to rewrite the Chapman-Kolmogorov equation (4) as:

$$Q_{u+s}(Z_3|Z_1) = \int \mathrm{d}Z_2 Q_s(Z_3|Z_2)Q_u(Z_2|Z_1). \tag{11}$$

In the limit of u going to 0, one expects that no changes occur in the state of the system, that is, $Q_u(Z|Z_0)$ approaches a Dirac delta $\delta(Z_i - Z_j)$ continuously and in a "differentiable" manner. Expanding Q_s around the delta $\delta(Z_i - Z_j)$ one has:

$$Q_s(Z|Z_0) = (1 - \omega_0 s)\delta(Z_i - Z_j) + s\Omega_s(Z|Z_0) + o(s). \tag{12}$$

Here, $\Omega(Z|Z_0$ represents the part of the time derivative of $Q_u(Z|Z_0)$ that is not proportional to a delta function and serves as the transition probability per unit time, also referred to as the *transition density*. The term

$$\omega_0 = \int \mathrm{d}Z_j \Omega(Z|Z_0) \tag{13}$$

is instead the part in the time derivative of $Q_u(Z|Z_0)$ that multiplies the Dirac delta in the expansion. Inserting the expansion Eq. (12) and the expression Eq. (13) in the stationary Chapman-Kolmogorov equation (11), one obtains [49]:

$$Q_{u+s}(Z_3|Z_1) - Q_u(Z_3|Z_1) = s \int \mathrm{d}Z_2 \left[\Omega(Z_3|Z_2)Q_u(Z_2|Z_1) - \Omega(Z_2|Z_3)Q_u(Z_3|Z_1)\right]. \tag{14}$$

In the limit of s going to zero, one gets a differential equation by dividing both sides by s:

$$\frac{\partial}{\partial u}Q_u(Z_3|Z_1) = \int \mathrm{d}Z_2 \left[\Omega(Z_3|Z_2)Q_u(Z_2|Z_1) - \Omega(Z_2|Z_3)Q_u(Z_3|Z_1)\right]. \tag{15}$$

Since Z_1 only appears as a condition in the probabilities Q_u in this last equation, it is convenient to write $p(Z,t) \equiv Q_t(Z|Z_1)$ and obtain:

$$\frac{\partial}{\partial t}p(Z,t) = \int \mathrm{d}Y \left[\Omega(Z|Y)p(Y,t) - \Omega(Y|Z)p(Z,t)\right]. \tag{16}$$

This equation is an example of master equation, a term used to refer to first order differential equations describing the temporal evolution of probabilites. In later sections we will discuss their importance in the case of dissipative quantum processes. If only infinitesimal differences between Z and Y are allowed in the previous equation, an expansion in the “jumps” $x = Z - Y$ is possible. It is this expansion that is called the Fokker-Planck equation, and has the form [cf. (10)]:

$$\frac{\partial}{\partial t}p(Z,t) = \frac{\partial}{\partial Z}[\mu(Z)p(Z,t)] + \frac{\partial^2}{\partial Z^2}[D(Z)p(Z,t)]. \tag{17}$$

It was derived by Planck [60] from the master equation Eq. (16) under the assumptions that i) only small jumps $Z - Y$ are allowed by $\Omega(Z|Y)$ and ii) $p(Z,t)$ varies slowly in Z. These assumptions lead to the following derivation. If one introduces a jump size $x = Z - Y$, it is possible to rewrite $\Omega(Z|Y)$ as

$$\Omega(Z|Y) = w(Y,x). \tag{18}$$

The master equation (16) then takes the form:

$$\frac{\partial}{\partial t}p(Z,t) = \int \mathrm{d}x w(Z-x,x)p(Z-x,t) - p(Z,t)\int \mathrm{d}x w(Z,-x). \tag{19}$$

The assumption that only small jumps x take place translates [49] in the fact that the transition density $w(Z,x)$ vanishes fast for jumps x bigger than a threshold ξ, and stays constant for small variations in Z. More precisely, apart for terms of order $o(\xi)$, the quantity $w(Z-x,x)$ is replaced by $w(Z,x)$ in Eq. (19), which becomes:

$$\frac{\partial}{\partial t}p(Z,t) = \int \mathrm{d}x w(Z,x)p(Z-x,t) - p(Z,t)\int \mathrm{d}x w(Z,-x). \tag{20}$$

The second assumption is that $p(Z,t)$ varies slowly for small increments of Z. This is to say, one assumes that $p(Z,t)$ is continuous, and two times differentiable with respect to Z. In this way, a small variation in Z can be Taylor expanded. Then it is possible to rewrite Eq. (20), apart for terms that are order $o(\xi^2)$ as:

$$\begin{aligned}
&\frac{\partial}{\partial t}p(Z,t) = \\
&\int \mathrm{d}x\Big[w(Z,x)p(Z,t) + \\
&x\frac{\partial}{\partial Z}[w(Z,x)p(Z,t)] + \frac{x^2}{2}\frac{\partial^2}{\partial Z^2}[w(Z,x)p(Z,t)] - p(Z,t)w(Z,-x)\Big].
\end{aligned} \tag{21}$$

Since the first and the last term cancel out, and $p(Z,t)$ is not integrated over, one recovers the Fokker-Planck equation (17):

$$\frac{\partial}{\partial t}p(Z,t) = \frac{\partial}{\partial Z}[\mu(Z)p(Z,t)] + \frac{1}{2}\frac{\partial^2}{\partial Z^2}[D(Z)p(Z,t)], \tag{22}$$

where we have defined:

$$\mu(Z) = \int \mathrm{d}x[w(Z,x)x] \tag{23}$$

and

$$D(Z) = \int \mathrm{d}x[w(Z,x)x^2], \tag{24}$$

as the first and the second moments of x with respect to the transition probability $w(Z,x)$.

2.1.4 The Langevin equation

The solution Eq. (5) to the Chapman-Kolmogorov equation Eq. (4), corresponds to the simplest, (yet very important [49]), example of stochastic, Markovian dynamics, that is, the Brownian motion. This motion is named after the botanist Robert Brown, who conducted some of the earliest studies of the motion of a particle of pollen suspended in water. The first mathematical description of such a motion is due to Einstein [52] and Smoluchowski [61]. The pollen particle is much heavier than the fluid molecules and it collides randomly with them. Each of the collisions changes both the direction of motion and the velocity v. Since the fluid molecules are much lighter than the pollen particle, they move much faster, and between each observation, they change the velocity v many times, such that the two-times correlation function of the velocity decays much faster than the time scale on which measurements are made. Based on this, Einstein and Smoluchowski developed a theory for the probability distribution of the position of the particle as a function of time. Equivalently, Langevin developed a more direct approach [49, 62, 58], in which the equation of motion of the particle are explicitly written as function of a directed, deterministic force, or drift term, and a stochastic, probabilistic force, or diffusion term. The theory reported here follows the Langevin approach. I restrict to the treatment of the one-dimensional motion, which is simpler yet allows to highlight the main ideas behind this approach. The starting point of Langevin's treatment is that, given the velocity v of a particle of unit mass, its motion is dictated by a directed force due to Stoke's law, which is proportional to the velocity itself, and a random force that changes direction and strength randomly with time:

$$\frac{\mathrm{d}}{\mathrm{d}t}v(t) = -\gamma v(t) + \xi(t), \qquad v(t=0) = v_0. \tag{25}$$

Here γ is a constant that represents the damping of the motion of the particle. As will be shown at the end of this section, this equation is the "Langevin" formulation of the Ornstein-Uhlenbeck process, Eq. (10). The noisy force $\xi(t)$ has the following properties: (i) its expectation value over many realization is zero

$$\mathbb{E}[\xi(t)] = 0, \tag{26}$$

(ii) its two time correlation function is a constant

$$\mathbb{E}[\xi(t_1)\xi(t_2)] = \Gamma\delta(t_2 - t_1). \tag{27}$$

A third assumption (iii), is that the noise is Gaussian, i.e. all odd moments of the noise vanish, while even moments reduce to two-point functions:

$$\mathbb{E}[\xi(t_1)...\xi(t_N)] = \mathbb{E}[\xi(t_1)\xi(t_2)]...\mathbb{E}[\xi(t_{N-1})\xi(t_N)] + ... \tag{28}$$

The Gaussian assumption (iii) is not required in the discussion that follows, but it is often needed if one wants to get an approximate description of expectation values involving higher order moments. Moreover, this assumption is required to show that this process is equivalent to the Ornstein-Uhlenbeck process of Eq. (7). It is worth noting that a purely Gaussian noise is not expected to exist in nature, and Eq. (28) should be regarded as an approximation to the actual physical dynamics [49]. The formal solution of Eq. (25) is:

$$v(t) = v_0 e^{-\gamma t} + \int_0^t du \xi(u) e^{-\gamma(t-u)}. \tag{29}$$

The one-point correlation function at time t is obtained by averaging this last expression over noise:

$$\mathbb{E}[v(t)] = v_0 e^{-\gamma t}. \tag{30}$$

The expectation value for the square of the velocity $v(t)$ instead reads:

$$\mathbb{E}[v(t)^2] = e^{-\gamma 2t} v_0^2 + \frac{\Gamma}{2\gamma}(1 - e^{-2\gamma t}). \tag{31}$$

At very long times the energy of the particle is given by the equipartition theorem [49], so that

$$\frac{\Gamma}{\gamma} = 3 k_{\mathrm{B}} T. \tag{32}$$

Where $k_{\mathrm{B}} = 1.38... \times 10^{-23} \frac{\mathrm{m}^2 kg}{\mathrm{s}^2 K}$ is the Boltzmann constant. The velocity v can be integrated in time to obtain the displacement in space $x(t)$:

$$x(t) = \frac{1 - e^{-\gamma t}}{\gamma} v_0 + \frac{1}{\gamma} \int_0^t du \xi(u)(1 - e^{-(t-u)}). \tag{33}$$

This implies that the mean square displacement of the particle has the following form:

$$\mathbb{E}[x^2(t)] = \frac{(1 - e^{-\gamma t})^2}{\gamma^2} v_0^2 + \frac{\Gamma}{\gamma^2} t - 2\frac{\Gamma}{\gamma^2}(1 - e^{-\gamma t}) + \frac{\Gamma}{2\gamma^2}(1 - e^{-2\gamma t}). \tag{34}$$

For long times $t \gg 1/\gamma$ the mean square displacement is proportional to time, $\mathbb{E}[x^2(t)] \sim \frac{\Gamma}{\gamma^2} t$, and one recovers the celebrated Einstein's relation

$$D = 3\frac{1}{\gamma} k_{\mathrm{B}} T. \tag{35}$$

Here D is the diffusion constant, while $1/\gamma$ is the relaxation time.

2.1.5 Equivalence with Fokker-Planck equation

To see that the Langevin equation (25) and the Fokker-Planck equation (10) are equivalent, let us introduce the characteristic function of a stochastic process X, which is defined as:

$$\phi_X(s) = \mathbb{E}[e^{isX}] \equiv \int dx e^{isx} P_t(x|v_0). \tag{36}$$

For simplicity we consider $v_0 = 0$. By expanding in series the term on the left side, and plugging in the solution for the velocity $v(t)$ of the Langevin equation (29), one gets:

$$\phi_{v(t)}(s) = \sum_{n=0}^{\infty} \frac{(is)^n}{n!} \int_0^{t_1} \mathrm{d}t_1 ... \int_0^{t_n} \mathrm{d}t_n e^{\gamma(t-t_1)} ... e^{\gamma(t-t_n)} \mathbb{E}[\xi(t_1)...\xi(t_n)]. \tag{37}$$

Since the noise $\xi(t)$ is Gaussian, only terms with an even $n = 2k$ survive in the series. From (28) one has that the Dirac deltas obtained from taking expectation values of the product of the noise variable $\xi(t_i)$, with $i = 1, .., n$ lead to $(n-1)!! = \frac{(2k)!}{2^k k!}$ equal terms. One thus obtains:

$$\phi_{v(t)}(s) = \sum_{k=0}^{\infty} \frac{(-1s^2)^k}{(2k)!} \frac{(2k)!}{2^k k!} \Gamma^k \left[\int_0^t \mathrm{d}u e^{2\gamma(t-u)} \right]^k. \tag{38}$$

Performing the integration in du leads to:

$$\phi_{v(t)}(s) = \sum_{k=0}^{\infty} \frac{(-1s^2)^k}{k!} \left[\frac{\Gamma}{2\gamma}(1 - e^{-2\gamma t}) \right]^k. \tag{39}$$

If one now performs the summation over all terms, the result is a Gaussian distribution:

$$\phi_{v(t)}(s) = \exp\left[-\frac{\Gamma}{2\gamma} s^2 (1 - e^{-2t}) \right]. \tag{40}$$

To obtain $P_t(v|v_0)$ an inverse Fourier transform is needed:

$$P_t(v|v_0) = \frac{1}{2\pi} \int \mathrm{d}s e^{-ivs} \phi_{v(t)}(s). \tag{41}$$

Since the Fourier-transform of a Gaussian is still a Gaussian, but with inverse variance, one is left with:

$$P_t(v|v_0 = 0) = \frac{1}{\sqrt{2\pi(\Gamma/\gamma)(1 - e^{-2\gamma t})}} \exp[-\frac{v^2}{2(\Gamma/\gamma)(1 - e^{-2\gamma t})}]. \tag{42}$$

Having set $v_0 = 0$ at the start of the calculation is equivalent to a shift in Eq. (29) from $v(t)$ to $v(t) - v_0 e^{-\gamma t}$, by restoring this difference in Eq. (42), one obtains:

$$P_t(v|v_0) = \frac{1}{\sqrt{2\pi(\Gamma/\gamma)(1 - e^{-2\gamma t})}} \exp[-\frac{(v - v_0 e^{-\gamma t})^2}{2(\Gamma/\gamma)(1 - e^{-2\gamma t})}], \tag{43}$$

which is identical in form to Eq. (7).

2.1.6 Itô processes

An important feature of the Langevin equation for the velocity of a Brownian particle is its linearity. This linearity maintains the equation solvable for the first and the second moments of the position x, cf. Eqs. (33), (34). In more general scenarios, nothing precludes the force term to be nonlinear, or the noise $\xi(t)$ to be multiplied by a nonlinear function. In this case, the Langevin equation for a time-dependent random variable Z_t is modified and becomes:

$$\frac{\mathrm{d}}{\mathrm{d}t} Z_t = \mu(Z_t, t) + \sigma(Z_t, t)\xi(t). \tag{44}$$

The term $\mu(Z_t, t)$ is deemed the drift term, while $\sigma(Z_t, t)$ is called the diffusion term. nonlinearity leads to several difficulties, in fact, given a general form of $\mu(Z_t, t)$ and $\sigma(Z_t, t)$ in Eq. (44), a closed-form solution for the moments of Z_t is usually an open question. Moreover, an intrinsic problem associated with nonlinear stochastic differential equations is that since the noise term ξ is a sequence of Dirac delta peaks, the term $\sigma(Z_t, t)$ is either multiplying zero or infinity. When the latter happens it is not clear if one should keep the random variable Z_t at the time t of the arrival of the last delta peak, or at the time $t' > t$ of the current one. In the present work we adopt the assignment due to Itô, who uses the time of the last peak. This means that for an infinitesimal time interval $\Delta t \to 0$, Eq, (44) can be written as:

$$Z_{t+\Delta t} - Z_t = \mu(Z_t, t)\Delta t + \sigma(Z_t, t)\mathrm{d}W_t. \tag{45}$$

where

$$\mathrm{d}W_t = \int_t^{t+\Delta t} \mathrm{d}u \xi(u). \tag{46}$$

In this thesis, we focus on the stochastic dynamics of some "reduced" degrees of freedom in a many-body system, such as the average magnetization of an Ising magnet. No assurance is given that an equation of the form (44) governs the dynamics of such observables, and even so, the form of both σ and μ is *a priori* unknown.

2.1.7 The drift and the diffusion terms

In this thesis, a main area of interest [20] is to infer the form of the drift term μ and the diffusion term σ from stochastic observables of physical systems that evolve in time. The underlying assumption is that this data can be modeled with an Itô equation of the form of Eq. (44). We make use of a closed formula for the drift term. In fact, it can be proven that the generator of the dynamics is [63, 64, 65].

$$\mu(x) = \lim_{\Delta t \to 0^+} \frac{\mathbb{E}_{Z_t=x}[Z_{t+\Delta t}] - x}{\Delta t}. \tag{47}$$

In this equation $E_{Z_t=x}[Z_t]$ is the expectation value over the possible trajectories, with initial conditions $Z_t = x$ for a fixed value x of the dynamics. In general, for a twice differentiable function f, the "infinitesimal generator" $\mathcal{G}$ of the dynamics is

$$\mathcal{G}f(x) = \lim_{\Delta t \to 0^+} \frac{\mathbb{E}_{Z_t=x}[f(Z_{t+\Delta t})] - f(x)}{\Delta t}. \tag{48}$$

Itô [66] proved that, if the rule for the assignment of the times in the diffusion term is as in Eq. (45), then the infinitesimal generator takes the form:

$$\mathcal{G}f(x) = \mu(x)\frac{\partial}{\partial x}f(x) + \frac{1}{2}\sigma^2(x)\frac{\partial^2}{\partial x^2}f(x). \tag{49}$$

One sees that taking this particular Itô choice for the assignment of the time in the noise term of Eq. (45) leads to a "change" in the usual rules for differentiation, with a term proportional to σ^2 appearing in the differential $\mathcal{G}f(x)$. Comparing Eqs. (48) and (49) one notices that the noise has to be proportional to a "square root" of the infinitesimal increment of time Δt. This fact can be used to compute the diffusion term as a sort of "second moment" of the stochastic process Z_t under consideration. In fact, it can be proven [63, 64, 65] that if one define the *quadratic variation* $[Z]_t$ of the process Z_t as:

$$[Z]_t \equiv \sum \lim_{\Delta t \to 0^+} (Z_{t+\Delta t} - Z_t)^2. \tag{50}$$

then the following applies:

$$[Z]_t = \int_0^t du \sigma^2(Z_u). \tag{51}$$

In our work [20] we use this formula to infer the form of the diffusion term for the postulated Itô equation that we assume governs the dynamics of the order parameters of some paradigmatic models of statistical physics.

2.2 Open quantum systems

2.2.1 Introduction

For classical many-body systems, probabilistic approaches such as those described in the previous section arise because the collisions between a large number of discrete

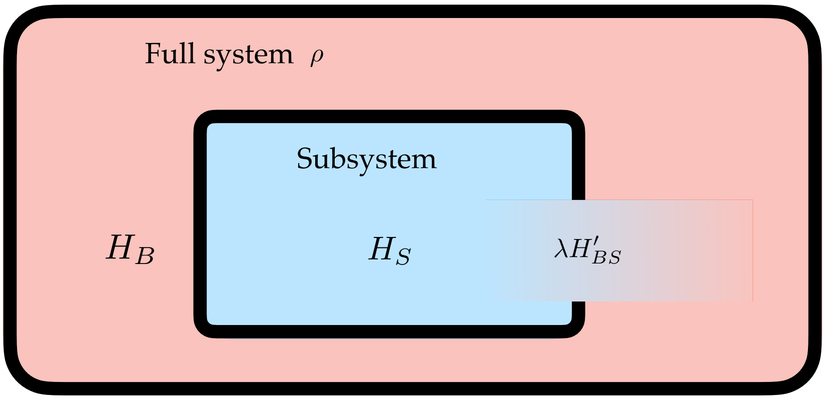

Figure 4: **Interaction between the environment and the subsystem.** The state of the full system, ρ, evolves according to a unitary dynamics given by the solution of the von Neumann equation, Eq. (56), with Hamiltonian $H = H_B + H_S + \lambda H'_{BS}$. When restricted to the quantum state ρ_S of the subsystem S, this dynamics ceases to be unitary, but under certain assumptions, can be made to follow a so-called Lindblad dynamics, Eq. (69).

particles can be regarded as a probabilistic phenomena. This is to say, collisions can be modelled as happening with a certain collision rate. From this, a deterministic differential equation can be extracted for the probability density of the, say, position of a given particle. Analogously, in the case of quantum particles, the main object of study is the wave-function $|\mathbf{\Psi}\rangle$, which has a definition of a "probability amplitude". A probability amplitude acts as the "square root" of a probability density and can take negative or complex values, giving rise to interference patterns that are not seen in classical mechanics. Open quantum systems serve as a bridge between classical and quantum many-body systems, enabling the integration of quantum effects, such as noise, and classical stochastic dynamics within a unified formalism. In open quantum systems, the primary object of study is the quantum state ρ [cf. Fig. 4]. This state represents the probability distribution of the entire system being in a specific configuration. However, in the context of open quantum systems, only a subsystem S of the entire "universe" is experimentally accessible, while the remaining part is referred to as the "bath" B. The term "open" indicates that there are channels through which information can be exchanged between the subsystem S and the bath B. Physically, this means that the total Hamiltonian governing the full system H can be decomposed in three terms: $H = H_S + H_B + \lambda H'_{BS}$. The terms H_S and H_B act only on the subsystem S and on the bath B respectively, while H'_{BS} serves as a "coupling term" between the bath and the subsystem.

2.2.2 Time evolution of open quantum systems

Open quantum systems refer to quantum systems coupled to an environment. The environment usually consists of a macroscopic collection of degrees of freedom at thermal equilibrium. The coupling between system and environment leads to the exchange of energy and information between the system and environment, usually resulting in irreversible effects as dissipation and decoherence. This makes the dynamics of an open system qualitatively different from that of a closed quantum system, which only evolve unitarily. Their study is important because no quantum system in nature is completely closed. A coupling with an external environment is not only inevitable but also necessary to observe the system itself, in the sense that an experimental apparatus acts as an external perturbation to the quantum system under observation [67, 68, 49]. Open quantum systems are defined as a generalization of closed ones, so it is important to start defining what a closed quantum system is, and what are its properties. Full information about a closed quantum system is contained in its wave function $|\psi(t)\rangle$. The time evolution of $|\psi(t)\rangle$ is given by Schrödinger's equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left|\psi(t)\right\rangle = -\frac{i}{\hbar}H(t)\left|\psi(t)\right\rangle . \tag{52}$$

In the following natural units are used, in which $\hbar = 1$. The Hamiltonian operator H is a quantum observable, and thus it is Hermitian ($H^\dagger = H$), so that its expectation values $\langle H\rangle = \langle\psi| H |\psi\rangle$ are real. Because of the Hermiticity of H in Eq. (52), the expectation value $\langle H\rangle$ is conserved in time, and has the meaning of the energy of the system. The formal solution of Eq. (52) is given by the time-ordered exponential:

$$|\psi(t)\rangle = \mathcal{T}_{\leftarrow}\mathrm{exp}\left[-i\int_{t_0}^{t} dsH(s)\right]|\psi(t_0)\rangle . \tag{53}$$

In an experimental setting, access to information is restricted to a system coupled with an environment. However, one cannot monitor *all* the environment and inevitably information is lost. This means that the quantum state associated with the accessible system does not carry all the existing information.

The information associated with the larger environment in which the subsystem is embedded has been "traced out". The system under investigation is thus not closed anymore, but one is confronted with an open quantum system. To see how the Schrödinger equation translates in the case of open quantum systems, it is convenient to phrase the time evolution in terms of a mixed quantum state, (or density matrix):

$$\rho(t) = \sum_i p_i \left|\psi_i(t)\right\rangle \left\langle\psi_i(t)\right| . \tag{54}$$

The density matrix $\rho(t)$ is a *convex* sum of normalized pure states $|\psi_i(t)\rangle$. By definition, a convex sum is a weighted sum where the weights p_i form a probability distribution, $\sum_i p_i = 1$ and $p_i > 0$ for all i. Each $P_{\psi_i} = |\psi_i\rangle \langle\psi_i|$ in the sum in Eq. (54) is a projector to the pure state $|\psi_i\rangle$. The requirement that a mixed state ρ is a convex sum of projectors establishes a statistical interpretation to quantum mechanics: the probability $p(x)$ of making a measurement x on the system, that is, of finding it in state $P_x = |x\rangle \langle x|$ is:

$$p(x) = \text{tr}\left[P_x \sum_i p_i \left|\psi_i(t)\right\rangle \left\langle\psi_i(t)\right|\right] = \sum_i p_i \left\langle\psi_i(t)\right| P_x \left|\psi_i(t)\right\rangle . \tag{55}$$

A density matrix ρ as in (54) has three important properties: i) it is Hermitian, ii) its trace sums to one when properly normalized and iii) it is positive (more precisely, semi-definite positive). This is to say, its eigenvalues are greater or equal to zero. From Eq. (52), one can obtain the time derivative for $\rho(t)$ as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho(t) = \sum_i p_i \frac{\mathrm{d}}{\mathrm{d}t} \left|\psi_i(t)\right\rangle \left\langle\psi_i(t)\right| + \left|\psi_i(t)\right\rangle \frac{\mathrm{d}}{\mathrm{d}t} \left\langle\psi_i(t)\right| = -i[H(t), \rho(t)]. \tag{56}$$

This last equation is known as the von Neumann equation. Again, the formal solution of the von Neumann equation is a time-ordered exponential but given in terms of a linear "Liouville generator" $\mathcal{L}(t)$. The action of $\mathcal{L}(t)$ on $\rho(t)$ is given by

$$\mathcal{L}(t)\rho(t) = -i[H(t), \rho(t)]. \tag{57}$$

The solution $\rho(t)$ to Eq. (56), for fixed initial conditions $\rho(t_0) = \rho_0$ is then:

$$\rho(t) = \mathcal{T}_{\leftarrow}\text{exp}\left[\int_{t_0}^{t} ds\mathcal{L}(s)\right]\rho_0. \tag{58}$$

The main advantage of working with a quantum state $\rho(t)$ is that the unobserved degrees of freedom of the environment can be readily taken into account, by means of a partial trace. This is to say, the density matrix of the observed system ρ_{S} is the partial trace over the degrees of freedom of the environment (or bath) B, of the full density matrix in Eq. (56):

$$\rho_S = \mathrm{Tr}_B \rho. \tag{59}$$

The general dynamical equation for ρ_S, known as Nakajima-Zwanzig equation, is complicated by the fact that it is an integrodifferential equation, such that there appear *nonsecular terms*, *memory effects* and time dependency in ρ_B. nonsecular terms appear as terms in ρ_S whose evolution has the same time scale of the relaxation to equilibrium of the bath B. Memory effects are encoded by terms in the equation that are nonlocal in time. When the unobserved environment can be idealized as an infinitely large source of excitations, the time dependency of ρ_B can be cured assuming an initial separable state, that is, a product state $\rho_S \otimes \rho_B$ [67]. If the bath is assumed to be infinite, its state is then already in a stationary state unaffected by the dynamics of the system S, and that they interact weakly, a regime known as Born approximation. Similarly, nonsecular terms can be eliminated, under the assumption that such a large bath relaxes to stationarity much faster than the system, a setting known as secular approximation. Finally, memory effects can be neglected if one perform the Markov approximation. In this approximation, one assumes that the system relaxes exponentially to stationary, on a time scale that is much longer than the oscillations due to the bath.

A setting where the Born Markov approximation usually works very well is in quantum optical experiments. This is because the relaxation timescales of the electromagnetic environment are extremely short as compared to those of the system [69].

These three approximations, when applicable, lead to a simplified version of the general dynamical equation, called the Lindblad equation, for the reduced density matrix ρ_S. In the following, I first present the Nakajima-Zwanzig equation, and then proceed to apply the mentioned approximations to obtain the Lindblad equation.

2.2.3 The Nakajima-Zwanzig equation

In this section, following the standard treatment of [68, 70], I present the dynamical equation for a reduced density matrix ρ_S. The result of this derivation, which is known as Nakajima-Zwanzig equation, makes it apparent the problems outlined in the previous section. To find this equation, it is convenient to define a projector P_S from the global density matrix ρ, that describes the state of the system S coupled with the environment B to $\rho_S \otimes \rho_B$, which is instead the product state of the partial trace ρ_S in Eq. (59), and a reference state ρ_B for the bath, e.g. a stationary state. The action of this projector is:

$$P_S \rho = \mathrm{Tr}_B \rho \otimes \rho_B = \rho_S \otimes \rho_B. \tag{60}$$

Let us also define its complement $P_B = \mathbb{1} - P_S$, such that their sum is the identity: $P_B + P_S = \mathbb{1}$. One notices that the time evolution for $P_S \rho$ is:

$$\frac{\mathrm{d}}{\mathrm{d}t} P_S \rho = P_S \mathcal{L}(t)(P_S + P_B)\rho(t), \tag{61}$$

while the equation for the time evolution of $P_B \rho$ is given by:

$$\frac{\mathrm{d}}{\mathrm{d}t} P_B \rho = P_B \mathcal{L}(t)(P_S + P_B)\rho(t). \tag{62}$$

The equation (62) has the form of a nonhomogeneous linear equation like:

$$\frac{\mathrm{d}}{\mathrm{d}t} \alpha(t) = z\alpha(t) + F(t). \tag{63}$$

Recalling that the solution to such an equation, for fixed conditions at $\alpha_0 = \alpha(t = 0)$, is:

$$\alpha(t) = e^{zt}\alpha_0 + \int_0^t ds e^{z(t-s)} F(s), \tag{64}$$

one can write the equivalent of the exponential term e^{zt} as a forward propagator $G_{\leftarrow}(t, t_0)$:

$$G_{\leftarrow}(t, s) = \mathcal{T}_{\leftarrow}\mathrm{exp}\left[\int_s^t du P_B \mathcal{L}(u)\right], \qquad G_{\leftarrow}(t, t) = \mathbb{1}. \tag{65}$$

This propagator is solution to the linear differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} G_{\leftarrow}(t, s) = P_B \mathcal{L}(t) G_{\leftarrow}(t, s). \tag{66}$$

then the formal solution to Eq. (62) is, in analogy to Eq. (64):

$$P_B \rho(t) = G_{\leftarrow}(t, 0) P_B \rho(0) + \int_0^t G_{\leftarrow}(t, s) P_B \mathcal{L}(t) P_S \rho(s). \tag{67}$$

Having fixed the initial conditions for $\rho(t)$ at $t = 0$. Plugging in the solution Eq. (67) for the inhomogeneous equation Eq. (62) into eq. (61) one obtains the Nakajima-Zwanzig equation:

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t} P_S \rho = \\ &P_S \mathcal{L}(t) \left[P_S \rho(t) + G_{\leftarrow}(t, 0) P_B \rho(0) + \int_0^t ds G_{\leftarrow}(t, s) P_B \mathcal{L}(t) P_S \rho(s)\right]. \end{aligned} \tag{68}$$

One can see here the problems highlighted at the end of Sec. 2.2. In particular, the time dependence of the propagator $G_{\leftarrow}$ on the history of the system, makes it subjects to memory effects.

2.2.4 The Lindblad equation

The Nakajima-Zwanzig equation (68) encodes memory effects. In many instances, these effects are important and cannot be neglected. This is what often happens in the case of strongly correlated systems of solid-state physics [6]. However, there are numerous relevant situations where these memory effects can be disregarded, particularly in atomic systems and quantum optics [68]. In these cases, it is crucial to determine when it is necessary to consider memory effects, as opposed to when a simplified, *Markovian* description of the quantum system can be used instead of the Nakajima-Zwanzig equation (68). This simplification is achieved by the "Lindblad equation," which, unlike Eq. (68), is local in time. To ascertain whether this description is applicable for a quantum system interacting with an external bath, it is essential to first establish whether a clear separation between the bath and the system can be made. Such a separation is often justified by the fact that correlations in the bath decay on a timescale τ_B that is much smaller than the typical time scale τ_S of the correlations in the system, $\tau_S \gg \tau_B$. In such a scenario, it makes sense to ask if a linearized version of Eq. (68) can describe the system, and what is its form. Starting from the latter question, physically, a linear equation for the evolution of $\rho(t)$ must necessarily preserve the trace, Hermiticity, and positivity of the density matrix. If one performs the Born-Markov and secular approximations described in Sec. 2.2 on the Nakajima-Zwanzig equation, what is obtained is not simply a map that preserves the previously stated properties, but a map with the more stringent condition of *complete positivity*. A completely positive map $\mathcal{J}$ does not simply preserve the positivity of an operator ρ, defined on an Hilbert space H_1 but is such that [70, 71], for any Hilbert space H_2 also the map $\mathcal{J} \otimes \mathbb{1}$ preserves the positivity of any operator χ on $\mathsf{H}_1 \otimes \mathsf{H}_2$. It can be proved [72, 73, 71] that the general form of a linear (i.e. Markovian) equation for the density matrix ρ, preserving its Hermiticity and trace has the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho = -i[H,\rho] + \sum_{ij} \kappa_{ij} \left([F_i, \rho F_j] + [F_i \rho, F_j] \right) = \mathcal{L}[\rho], \tag{69}$$

where $\{F_i\}_{i \text{ in } \{1,2,..,N\}}$ where $N = d^2 - 1$ form a basis for the algebra of operators on the Hilbert space of the system, which here is assumed to be of finite dimension d, while the matrix H is Hermitian. The matrix κ is called the Kossakowsky matrix, and the condition of complete positivity rests on it. It can be proven [72] that the dynamics is com-

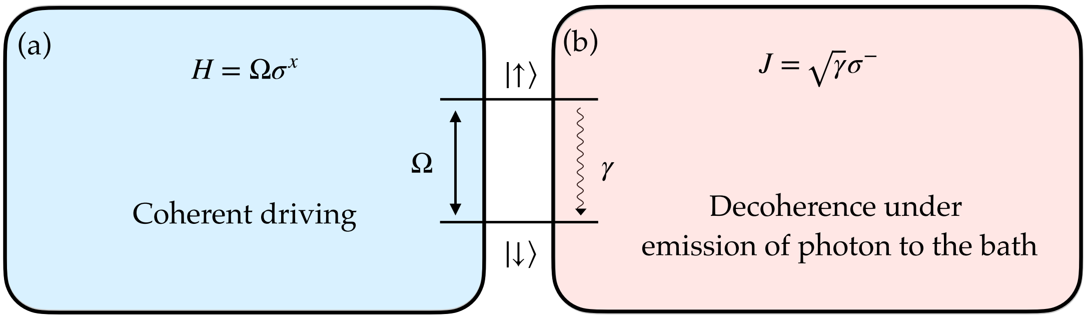

Figure 5: **Disspative dynamics for a driven two level system.** An example of Lindbladian dynamics is given in terms of a two-level system. On one side the system oscillates between the two levels with the so-called "Rabi frequency" Ω (a). On the other, the system is subject to decoherence under the emission of a photon to the bath with "jump frequency" γ (b).

pletely positive if and only if the matrix κ is semi-definite positive. The Kossakowski matrix κ can be diagonalized via an unitary transformation $\mathcal{U}$ to get its eigenvalues $(\gamma_1, ..., \gamma_N)$. By performing a change of basis $J_i = \mathcal{U}^\dagger F_i \mathcal{U}$ Eq. (69) is transformed into the "canonical" form which is known as the Lindblad equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho = -i[H,\rho] + \sum_i \gamma_i \left(J_i \rho J_i^\dagger - \frac{1}{2}\{J_i^\dagger J_i, \rho\} \right). \tag{70}$$

Here, the operators J_i are called jump operators, and the (positive) eigenvalues $\{\gamma_i\}$ can be interpreted as rates at which quantum jumps occur in the system. A simple example of such a dynamics —cf. Fig. 5 is provided by a driven two level system. In this case the state of the system is either "up" $|\uparrow\rangle$ or "down" $|\uparrow\rangle$. An Hamiltonian for this kind of system is the so called "Rabi Hamiltonian":

$$H = \Omega\sigma^x \equiv \Omega \left(|\uparrow\rangle \langle\downarrow| + |\downarrow\rangle \langle\uparrow| \right), \tag{71}$$

which determines the frequency Ω with which the system oscillates between one state and the other. The coupling to the bath is determined by a jump operator of the form:

$$J = \sigma^- \equiv |\downarrow\rangle \langle\uparrow|, \tag{72}$$

with associated frequency of decay γ.

2.2.5 Microscopic derivation

In the following a standard microscopic derivation of the Lindblad equation (69) based on [68, 70] is reported. This derivation is especially important since it serves as a physical justification to the use of a completely positive map for the time evolution of open quantum systems. The first important assumption that is made is that the system is weakly interacting with an external environment. This "weak coupling" limit assumes that the full Hamiltonian governing the bath and the system can be split in three terms (cf. Fig. 4):

$$H = H_S + H_B + \lambda H'_{BS}. \tag{73}$$

As already mentioned, the first term H_S in Eq. (73) models only the system S, the second term H_N only the bath B and the third H'_{BS} represents the interaction between the two, mediated by a coupling λ. Assuming that λ is small, a second-order expansion on it is performed. The coupling Hamiltonian is assumed to be of the form:

$$H'_{BS} = \sum_\alpha S_\alpha \otimes B'_\alpha, \tag{74}$$

where the self-adjoint operators S_α and B'_α act on the system S or on the bath B only, respectively. It is convenient to redefine the operators B'_α to have zero expectation value on the bath:

$$B_\alpha = B'_\alpha - \mathrm{Tr}_B[\rho_B B'_\alpha], \qquad \mathrm{Tr}_B[\rho_B B_\alpha] = 0, \tag{75}$$

such that the Hamiltonian H in Eq. (73) becomes:

$$H = H_S + H_B + \lambda H_{BS} \tag{76}$$

with:

$$H_S^\lambda = H_S + \lambda \sum_\alpha S_\alpha \otimes \mathrm{Tr}[\rho_B B_\alpha], \qquad H_{BS} = \sum_\alpha S_\alpha \otimes B_\alpha. \tag{77}$$

The Hamiltonian H_S^λ has undergone a so-called Lamb shift, that is, its energy levels are renormalized by a mean field factor accounting for a $O(\lambda)$ interaction with the bath. For simplicity, I consider a time-independent Hamiltonian H. In this case, the commutator $\mathcal{L} = [H, \cdot]$ splits in three parts:

$$\mathcal{L} = \mathcal{L}_S \otimes \mathbb{1}_B + \mathbb{1}_S \otimes \mathcal{L}_B + \lambda \mathcal{L}_{BS}, \tag{78}$$

with $\mathcal{L}_S^\lambda = -i[H_S^\lambda, \cdot]$, $\mathcal{L}_B = -i[H_B, \cdot]$ and $\mathcal{L}_{BS} = -i[H_{BS}, \cdot]$. It is also convenient to define the generator for the nonshifted system Hamiltonian $\mathcal{L}_S = -i[H_S, \cdot]$. Now, to arrive at the desired Markovian form for the equation governing the dynamics one must first take the so-called Born approximation. This consists in fixing the action of P_S on $\rho(t)$ to be such that:

$$P_S\rho(t) = \rho_S(t) \otimes \rho_B \tag{79}$$

—-apart from terms that can be neglected in λ. Moreover, the reference state ρ_B is assumed to be stationary with respect to the bath dynamics:

$$[H_B, \rho_B] = 0. \tag{80}$$

The assumption made by Eq. (80) represents the scenario of a large bath that remains unaffected by the dynamics of the system S, while the system itself is influenced by the presence of the bath. The steadiness of the bath, as described by Eq. (79), causes the second term in Eq. (68) to vanish, leaving us with the following equation:

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t} P_S[\rho_S(t) \otimes \rho_B] = \\ &P_S\mathcal{L}P_S[\rho_S(t) \otimes \rho_B] + \int_0^t ds P_S\mathcal{L}P_B \exp[(t-s)P_B\mathcal{L}P_B](P_B\mathcal{L}P_S)[\rho_S(s) \otimes \rho_B].\end{aligned} \tag{81}$$

Here the identity of the bath $\mathbb{1}_B = P_B^2$ has been inserted before and after the propagator: $G_\leftarrow(s,t) = \exp[(t-s)P_B\mathcal{L}P_B]$ in the integral term. Now, the first term on the right hand side of Eq. (81) is

$$P_S\mathcal{L}P_S[\rho_S(t) \otimes \rho_B] = \mathcal{L}_S[\rho_S(t)] \otimes \rho_B. \tag{82}$$

This occurs because $P_S\mathcal{L}_{BS}P_S = 0$, due to the condition $\mathrm{Tr}[\rho_B B_\alpha] = 0$, and the stationarity of ρ_B with respect to H_B. The term $P_S\mathcal{L}P_B[\rho(t)]$ becomes:

$$P_S\mathcal{L}P_B[\rho(t)] = \lambda \mathrm{Tr}_B[\mathcal{L}_{BS}P_B] \otimes \rho_B \tag{83}$$

while the term in $P_B\mathcal{L}P_S$ becomes:

$$P_B\mathcal{L}P_S[\rho_S(t) \otimes \rho_B] = \lambda P_B\mathcal{L}_{BS}[\rho_S(t) \otimes \rho_B] \tag{84}$$

since $P_B[\rho_S(t) \otimes \rho_B] = 0$. Plugging this results in Eq. (81), tracing out ρ_B and using the commutators of the Hamiltonians, we are left with:

$$\frac{\mathrm{d}}{\mathrm{d}t}[\rho_S(t)] = \mathcal{L}_S^\lambda \rho_S(t) - \lambda^2 \int_0^t ds \mathrm{Tr}_B[H_{BS}, e^{(t-s)P_B\mathcal{L}P_B}[H_{BS}, \rho_S(s) \otimes \rho_B]] \tag{85}$$

This equation still incorporates in $\rho_S(t)$ terms that are nonlocal in time, given by the fact that $\rho_B(t)$ is integrated in time in the last term on the right-hand side of Eq. (85). To obtain an equation that is local in time, one notices that the appearance of the square of the coupling constant λ^2 in front of the integral term in Eq. (85) restricts the integration to slow time scales t of order $O(\lambda^{-2})$. The approximation made here involves performing a second-order expansion and retaining terms up to $O(\lambda^2)$. Hence, if one takes the formal solution to Eq. (85):

$$\begin{aligned}\rho_S(t) =& e^{t\mathcal{L}_S^\lambda}\rho_S(0) - \lambda^2 \int_0^t ds e^{(t-s)\mathcal{L}_S^\lambda} \\ &\times \int_0^s dw \mathrm{Tr}_B[H_{BS}, e^{(s-w)P_B\mathcal{L}P_B}\left[H_{BS}, \rho_S(w)\otimes\rho_B\right]],\end{aligned} \tag{86}$$

one notices that in the propagator $e^{P_B\mathcal{L}P_B}$, the term in $\lambda\mathcal{L}_{BS}$ would lead to terms order $O(\lambda^3)$, that can be neglected. This propagator is thus replaced by $e^{\mathcal{L}_S+\mathcal{L}_B}$. Moreover, by switching the order of integration and introducing the new variable $u = s - w$ one gets:

$$\begin{aligned}\rho_S(t) =& e^{t\mathcal{L}_S^\lambda}\rho_S(0) - \lambda^2 \int_0^t dw e^{(t-w)\mathcal{L}_S^\lambda} \\ & \int_0^{t-w} du \mathrm{Tr}_B e^{-u\mathcal{L}_S^\lambda}[H_{BS}, e^{u(\mathcal{L}_B+\mathcal{L}_S)}\left[H_{BS}, \rho_S(w)\otimes\rho_B\right]]\end{aligned} \tag{87}$$

In the case that most of the dynamics of the system S takes place at slow time scales, $t \sim 1/\lambda^2$the effects of the system are not negligible only when $t-w$ is large. Analogously, since the bath correlation functions decay very fast, integrating them up to large times will not change the result of the integration. One is thus allowed to substitute $t - w$ with infinity in the upper limit of the second integral in Eq. (87), to get:

$$\begin{aligned}\rho_S(t) =& e^{t\mathcal{L}_S}\rho_S(0) - \lambda^2 \int_0^t dw e^{(t-w)\mathcal{L}_S} \\ & \int_0^{\infty} du \mathrm{Tr}_B e^{-u\mathcal{L}_S}[H_{BS}, e^{u(\mathcal{L}_B+\mathcal{L}_S)}\left[H_{BS}, \rho_S(w)\otimes\rho_B\right]],\end{aligned} \tag{88}$$

which is formal solution to

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S(t) = \mathcal{L}_S^\lambda\rho_S(t) - \lambda^2\mathcal{D}_{BS}\left[\rho_S(t)\right]. \tag{89}$$

Here a dissipation term $\mathcal{D}_{BS}$ has been introduced:

$$\mathcal{D}_{BS}[\rho_S(t)] = \int_0^{\infty} du \mathrm{Tr}_B e^{-u\mathcal{L}_S}[H_{BS}, e^{u(\mathcal{L}_B+\mathcal{L}_S)}\left[H_{BS}, \rho_S(t)\otimes\rho_B\right]]. \tag{90}$$

Equations like Eq. (89) are known as Redfield equations [74]. Although this equation does guarantee the preservation of the trace and Hermiticity of the density matrix, it does not preserve its positivity [70, 68]. Nevertheless, it is useful to compute, for example, stationary values. To adsorb the propagators $e^{u\mathcal{L}_S}$ and $e^{u\mathcal{L}_B}$, in the operators appearing in $H_{BS} = \sum_\alpha S_\alpha \otimes B_\alpha$ one introduces the time-evolved operators

$$S_\alpha(t) = e^{\mathcal{L}_S}[S_\alpha] = e^{itH_S} S_\alpha e^{-itH_S}, \qquad B_\alpha(t) = e^{\mathcal{L}_B}[B_\alpha] = e^{itH_B} B_\alpha e^{-itH_B}. \tag{91}$$

Moreover, in Eq. (89), the degrees of freedom of the bath B are traced out, but there, it appears a double commutator of the interaction Hamiltonian $H_{BS} = \sum_\alpha S_\alpha \otimes B_\alpha$. It is thus convenient to consider the two-point function of the bath operators $B_\alpha(t)$:

$$G_{\alpha\beta}(s) = \mathrm{Tr}_B \left[\rho_B B_\alpha(s) B_\beta\right] = \mathrm{Tr}_B \left[\rho_B B_\alpha B_\beta(-s)\right]. \tag{92}$$

Plugging the definitions in Eqs. (91) and (92) in the dissipation term (90) one gets:

$$\begin{aligned}&\mathcal{D}_{BS}[\rho_S(t)] = \\ &\sum_{\alpha,\beta} \int_0^\infty ds \left(G_{\alpha\beta}(s)[S_\alpha(s), S_\beta \rho_S(t)] + G_{\beta\alpha}(-s)[\rho_S(t) S_\beta, S_\alpha(s)]\right).\end{aligned} \tag{93}$$

Now, to get rid of non-secular terms in $S_\alpha(t)$, that is, terms that oscillate very fast with respect to the typical (slow) relaxation time of the system S, one needs to decompose the operators S_α on the eigenbasis of H_S^λ, to separate between *secular* terms, whose modes are slow, and nonsecular ones. To do this, let us introduce the projectors P_{ϵ_λ}, which project onto the eigenspace of H_S^λ associated with the eigenvalue ϵ_λ. Then, we can define operators as follows:

$$S_\alpha(\omega) = \sum_{\epsilon_\lambda - \epsilon'_\lambda = \omega} P_{\epsilon_\lambda} S_\alpha P_{\epsilon'_\lambda} \tag{94}$$

such that

$$e^{t\mathcal{L}_S}[S_\alpha(\omega)] = e^{-i\omega t} S_\alpha(\omega), \qquad e^{t\mathcal{L}_S}[S_\alpha^\dagger(\omega)] = e^{i\omega t} S_\alpha^\dagger(\omega). \tag{95}$$

Inserting this expressions in Eq. (93), one obtains:

$$\begin{aligned}\mathcal{D}_{BS}[\rho_S(t)] = \sum_{\alpha,\beta} \sum_{\omega,\omega'} \int_0^\infty ds e^{i(\omega-\omega')s} \big(&G_{\alpha\beta}(s)[S_\alpha(s,\omega), S_\beta(\omega')\rho_S(t)] + \\ &G_{\beta\alpha}(-s)[\rho_S(t) S_\beta(\omega'), S_\alpha(s,\omega)]\big).\end{aligned} \tag{96}$$

The relaxation time of the system is of the order of the inverse of the typical difference between the modes $\omega - \omega'$. Because this relaxation time is expected to be much larger compared to that of the environment, these differences are expected to be small. This results in very rapid oscillations that can be averaged out, leaving us only with terms where $\omega = \omega'$ in the sum in Eq. (96). Moreover, for very small λ it is safe to assume that the energy levels ϵ_λ do not cross between themselves, and one can substitute the projectors of P_{ϵ_λ} for H_S^λ with the ones P_ϵ for the eigenspaces of H_S. One is thus left with:

$$\begin{aligned}\mathcal{D}_{BS}[\rho_S(t)] = \sum_{\alpha,\beta}\sum_{\omega}\int_0^\infty ds\big(e^{i\omega s}G_{\alpha\beta}(s)[S_\alpha(\omega), S_\beta^\dagger(\omega)\rho_S(t)] \\ +e^{-i\omega s}G_{\beta\alpha}(-s)[\rho_S(t)S_\beta(\omega), S_\alpha^\dagger(\omega)]\big).\end{aligned} \tag{97}$$

The time integration now affects only the two-point function of the bath which can be rewritten as:

$$\int_0^\infty dte^{i\omega t}G_{\alpha\beta}(t) = \frac{\kappa_{\alpha\beta}(\omega)}{2} + is_{\alpha\beta}(\omega), \tag{98}$$

where the quantities $s_{\alpha\beta}(\omega)$ and $\kappa_{\alpha\beta}(\omega)$ are defined as follows:

$$s_{\alpha\beta}(\omega) = \frac{1}{2\pi}\mathcal{P}\int_{-\infty}^\infty dz\frac{\kappa_{\alpha\beta}(z)}{z-\omega} = s_{\beta\alpha}^*(\omega). \tag{99}$$

The Hermitian matrix $\kappa_{\alpha\beta}(\omega)$ is defined as:

$$\kappa_{\alpha\beta}(\omega) = \int_{-\infty}^\infty dte^{-i\omega t}G_{\alpha\beta}(t) = \kappa_{\beta\alpha}^*(\omega). \tag{100}$$

This matrix is ensured to be semi-definite positive due to a theorem by Bochner [75, 68], which states that a continuous function $f(\omega)$ is definite positive if there exists a probability measure $\mu(t)$ such that

$$f(\omega) = \int_{-\infty}^\infty \mathrm{d}\mu(t)e^{i\omega t}. \tag{101}$$

To show that $\kappa_{\alpha\beta}$ is semi-definite positive, let us notice that for a fixed complex vector $z = (z_1, z_2, ...)$ the quantity

$$\sum_{\alpha,\beta} G_{\alpha\beta}(t)z_\alpha^* z_\beta = \sum_{\alpha,\beta}\mathrm{Tr}\big(B_\alpha(t)B_\beta(0)\big)z_\alpha^* z_\beta, \tag{102}$$

is a valid probability measure $\mu(t)$. To see this, let us expand each of the elements B_α on a basis of H, which we assume to be complete:

$$B_\alpha = \sum_i \hat{B}_\alpha^j, \qquad H_B B_\alpha^j = E_j B_\alpha^j. \tag{103}$$

Then the following applies:

$$\mathrm{Tr}\left[\rho\sum_{i,\alpha}e^{itE_i}B^i_\alpha z_\alpha\sum_{j,\beta}e^{-itE_j}B^j_\alpha z_\beta\right]=\mathrm{Tr}\left[\rho\left(\sum_{i,\alpha}e^{itE_i}B^i_\alpha z_\alpha\right)\left(\sum_{i,\alpha}e^{itE_i}B^i_\alpha z_\alpha\right)^\dagger\right]. \tag{104}$$

The last term is manifestly a positive real number, guaranteeing that the hypotheses of the Bochner theorem are met. Since the vector z_α is arbitrary, it follows that $\sum_{\alpha,\beta}\kappa_{\alpha\beta}(\omega)z^*_\alpha z_\beta$ is always positive, that is to say, $\kappa_{\alpha\beta}$ is a semi-definite positive matrix.

Plugging this expression in Eq. (97) one arrives at a formula for the dissipation term:

$$\begin{aligned}\mathcal{D}_{BS}[\rho_S]=&i\sum_{\alpha,\beta,\omega}s_{\alpha\beta}(\omega)S_\alpha(\omega)S^\dagger_\beta(\omega)\rho_S]-\\&\sum_{\alpha,\beta,\omega}\kappa_{\alpha\beta}(\omega)S^\dagger_\beta(\omega)\rho_S S_\alpha(\omega)-\frac{1}{2}\{S_\alpha(\omega)S^\dagger_\beta(\omega),\rho_S\}.\end{aligned} \tag{105}$$

The first on the right hand side in Eq. (105) is a commutator of a self-adjoint operator and can be adsorbed in the Hamiltonian H^λ_S while the second gives the dissipating nature of Eq. (69). Putting them together one has:

$$\frac{\mathrm{d}}{\mathrm{d}t}\rho_S=-i\mathcal{H}[\rho_S]+\mathcal{D}[\rho_S], \tag{106}$$

where:

$$\mathcal{H}[\rho_S]=[H'_S+\lambda\sum_\alpha S_\alpha\otimes\mathrm{Tr}[\rho_B B_\alpha]+\lambda^2\sum_{\alpha,\beta,\omega}s_{\alpha\beta}(\omega)S_\alpha(\omega)S^\dagger_\beta(\omega)]\rho_S. \tag{107}$$

The term $\mathcal{D}[\rho_S]$ is instead defined as:

$$\mathcal{D}[\rho_S]=\sum_{\alpha,\beta,\omega}\kappa_{\alpha\beta}(\omega)S^\dagger_\beta(\omega)\rho_S S_\alpha(\omega)-\frac{1}{2}\{S_\alpha(\omega)S^\dagger_\beta(\omega),\rho_S\}. \tag{108}$$

acts as a dissipative contribution to the dynamics.

2.2.6 The coherence vector

For systems with a finite $d-$dimensional local Hilbert space, such as those composed of spin-$\frac{1}{2}$ degrees of freedom, which are those that we study in our scientific papers [27] and [39], the density matrix can be represented as a vector rather than as a matrix, a representation that takes the name of “coherence vector”. This formulation of the density matrix is very amenable to numerical treatment, as it allows to express the action of the linear super-operator $\mathcal{L}$ appearing in the Lindblad equation (69) in terms of a matrix $\mathbf{L}$. In general, any $N \times N$ density matrix can always be decomposed on a basis $\{F_i\}_{i=1}^{N^2}$ of the algebra of operators of the Hilbert space H_S. If one fixes F_{N^2} to be proportional to the identity, the other matrices are traceless, orthogonal, and Hermitian. The algebra of the basis $\{F_i\}$ is given by [76]:

$$F_i F_j = \frac{2}{N}\delta_{ij}\mathbb{1} + i\sum_k f_{ijk}F_k + \sum_k d_{ijk}F_k \tag{109}$$

Where the so-called structure constant f_{ijk} is the completely anti-symmetric tensor components defined by the commutation relations between the matrices F_i:

$$\{F_i, F_j\} = 4\delta_{ij}\frac{\mathbb{1}}{d} + 2\sum_{k=1}^{d^2-1} d_{ijk}F_k, \qquad [F_i, F_j] = 2i\sum_{k=1}^{d^2-1} f_{ijk}F_k. \tag{110}$$

Since the normalization of F_i is:

$$\text{Tr}(F_i F_j) = 2\delta_{ij}, \tag{111}$$

one gets:

$$f_{ijk} = -\frac{i}{4}\text{Tr}\left([F_i, F_j]F_k\right). \tag{112}$$

Analogously, the completely symmetric tensor d_{ijk} is defined as:

$$d_{ijk} = \frac{1}{4}\text{Tr}\left(\{F_i, F_j\}F_k\right). \tag{113}$$

Notice that for self-adjoint basis operators $F_i = F_i^\dagger$, the structure constants are real. The expansion on this basis of the density matrix can be written as:

$$\rho = \frac{1}{N}\left(\mathbb{1} + \sqrt{\frac{N(N-1)}{2}}\sum_i v_i F_i\right), \tag{114}$$

where the vector n is the expansion on the basis F_i of the density matrix ρ. This vector is known as the *coherence* vector and as the *Bloch* vector in the case of single qubit, i.e. 2×2 density matrix. This vector v is given by the expectation values of the operators F_i on ρ:

$$n_i = \frac{1}{\mathcal{N}}\text{Tr}(\rho F_i), \qquad \mathcal{N} = \sqrt{\frac{N}{2(N-1)}}. \tag{115}$$

In terms of the structure constants f_{ijk}, d_{ijk} and the coherence vector v, the Lindblad equation (69) takes the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}v = \mathbf{L}v. \tag{116}$$

The matrix $\mathbf{L}$ can be split into an "Hamiltonian" matrix $\mathbf{H}$ and a dissipation matrix $\mathbf{D}$:

$$\mathbf{L} = \mathbf{H} + \mathbf{D}. \tag{117}$$

For a real vector $\boldsymbol{\omega}$, the skew-symmetric matrix $\mathbf{H}$ is given by

$$\begin{aligned} \mathbf{H}_{ij} &= \frac{1}{4\mathcal{N}} \sum_{k=1}^{N^2-1} f_{ijk} \boldsymbol{\omega}_k, \quad i,j \in \{1,2..,N^2-1\}, \\ \mathbf{H}_{iN^2} &= \mathbf{H}_{N^2 i} = 0, \quad i \in \{1,2..,N^2\}. \end{aligned} \tag{118}$$

whereas the real valued matrix $\mathbf{D}$ id given by:

$$\begin{aligned} \mathbf{D}_{mn} &= -\frac{8}{\mathcal{N}} \sum_{i,j,k=1}^{d^2-1} \left(f_{mik} f_{njk} \mathrm{Re}(\kappa)_{ij} + f_{mik} d_{njk} \mathrm{Im}(\kappa)_{ij} \right), \quad m,n \in \{1,2..,d^2-1\}, \\ \mathbf{D}_{md^2} &= -\frac{8}{\mathcal{N}} \sum_{i,j=1}^{d^2-1} f_{imj} \mathrm{Im}(\kappa)_{ij}, \qquad \mathbf{D}_{d^2 m} = 0, \quad m \in \{1,2..,N^2\}. \end{aligned} \tag{119}$$

In the following we report the derivation for the equations of the dynamics of the coherence vector $v = (v_1, ..., v_{N^2})$ [cf. (118) and (119)] as it appears in [27], and then apply them to the concrete case of a single qubit. They can be obtained by directly considering the time derivative of v_h:

$$\frac{d}{dt} v_h(t) = \frac{1}{\mathcal{N}} \mathrm{Tr} \left(F_h \frac{d}{dt} [\rho_S(t)] \right) = \frac{1}{\mathcal{N}} \mathrm{Tr} \left(F_h \mathcal{L}[\rho_S(t)] \right) = \frac{1}{\mathcal{N}} \mathrm{Tr} \left(\mathcal{L}^*[F_h] \rho_S(t) \right). \tag{120}$$

The first equality comes from Eq. (115), while the second comes from (69). In the previous equation, we introduce the dual map $\mathcal{L}^*$ was introduced. This map evolves observables and leaves the state ρ_S of the subsystem invariant. Its action is obtained from the one of $\mathcal{L}$ by applying the cyclic property of the trace. Substituting

$$\mathcal{L}^*[F_h] = \sum_{k=1}^{d^2} \mathrm{Tr}(F_k \mathcal{L}^*[F_h]) F_k \tag{121}$$

in Eq. (120), one gets:

$$\frac{d}{dt} v_h(t) = \frac{1}{\mathcal{N}} \sum_{k=1}^{d^2} \mathrm{Tr}(\mathcal{L}^*[F_h] F_k) \mathrm{Tr}(F_k \rho_S(t)) = \left[\mathbf{L} v(t) \right]_h. \tag{122}$$

The matrix $\mathbf{L}$ in Eq. (122) is:

$$\mathbf{L}_{hk} = \frac{1}{\mathcal{N}} \mathrm{Tr}(\mathcal{L}^*[F_h] F_k). \tag{123}$$

Explicitly, we obtain

$$\mathrm{Tr}(\mathcal{L}^*[F_h]F_k) = \sum_{h=1}^{d^2} \mathrm{Tr}\left(i[H, F_h]F_k + \frac{1}{2}\sum_{i,j=1}^{d^2-1} \kappa_{ij}(F_j^\dagger[F_h, F_i]F_k + [F_j^\dagger, F_h]F_iF_k) \right). \tag{124}$$

Comparing this equation with Eq. (69), one obtains:

$$\begin{aligned} \mathbf{H}_{kh} &= \frac{i}{\mathcal{N}}\mathrm{Tr}([H, F_k]F_h), \\ \mathbf{D}_{kh} &= \frac{1}{2\mathcal{N}}\sum_{i,j=1}^{d^2-1} \kappa_{ij}\mathrm{Tr}\left(F_j^\dagger[F_k, F_i]F_h + [F_j^\dagger, F_k]F_iF_h\right). \end{aligned} \tag{125}$$

We expand the Hamiltonian term over the basis F_i as $H = \sum_{i=1}^{N^2-1} F_i\boldsymbol{\omega}_i$, where $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, ..., \boldsymbol{\omega}_{N^2-1})$ is a $N^2 - 1$ dimensional vector, which, for Hermitian basis F_i, is real. The matrix **H** is then given element-wise by:

$$\mathbf{H}_{mn} = \frac{i}{\mathcal{N}}\sum_{k=1}^{d^2-1} \mathrm{Tr}([F_k, F_m]F_n)\boldsymbol{\omega}_k. \tag{126}$$

Given the definition of the structure constants f_{ijk} in Eq. (112), the Hamiltonian can be expanded as:

$$\mathbf{H}_{ij} = \frac{4}{\mathcal{N}}\sum_{k=1}^{N^2-1} f_{ijk}\boldsymbol{\omega}_k, \qquad \mathbf{H}_{id^2} = \mathbf{H}_{d^2i} = 0, \tag{127}$$

with $i, j \in \{1, 2.., N^2-1\}$. Thus, the Hamiltonian part is anti-symmetric, as it is expected for the matrix representation of the commutator of a Hermitian operator [70].

On the other side, the dissipative term **D** has an expansion over f_{ijk} and d_{ijk} [cf. Eqs. (112), (113)]

$$\begin{aligned} \mathbf{D}_{mn} &= \tfrac{1}{2\mathcal{N}}\textstyle\sum_{i,j}^{N^2-1} \kappa_{ij}\mathrm{Tr}\left([F_m, F_i]F_nF_j + [F_j, F_m]F_iF_n\right) \\ &= -\tfrac{4}{\mathcal{N}}\textstyle\sum_{i,j,k=1}^{N^2-1}(f_{mik}f_{njk}\mathrm{Re}(c)_{ij} + f_{mik}d_{njk}\mathrm{Im}(c)_{ij}) \end{aligned} \tag{128}$$

for $1 \leq m < N^2, 1 \leq n < N^2$. In the last line of Eq. (128) it is used the fact that the real (imaginary) part of a Hermitian matrix like κ is (skew-)symmetric. For the matrix elements of **D** with $1 \leq m < N^2$, $n = N^2$, one has:

$$\mathbf{D}_{mN^2} = \frac{1}{2}\sum_{i,j,k=1}^{d^2-1} \kappa_{ij}\mathrm{Tr}\left(F_i[F_m, F_j] + [F_i, F_m]F_j\right) = -\frac{4}{\mathcal{N}}\sum_{i,j=1}^{d^2-1} f_{imj}\mathrm{Im}(\kappa)_{ij}, \tag{129}$$

where, in the second equality, we made use of the cyclic property of the trace and again the fact that f_{imj} and $\mathrm{Im}(\kappa)_{ij}$ are anti-symmetric in the indices (ij), but $\mathrm{Re}(\kappa)_{ij}$ is symmetric. Lastly, $\mathbf{D}_{N^2n} = 0$ for $1 \leq n < N^2$.

2.2.7 The single qubit case

In the case of a two-level systems, the natural choice for F_i is given by the Pauli matrices, $\{F_i\} = \{\sigma_1, \sigma_2, \sigma_3, \mathbb{1}\}$, with:

$$\sigma^x \equiv \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma^y \equiv \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma^z \equiv \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{130}$$

The algebra of the Pauli matrices is given by:

$$\sigma_i \sigma_j = \delta_{ij} + i \sum_k \epsilon_{ijk} \sigma_k. \tag{131}$$

Where ϵ_{ijk} is the completely anti-symmetric Levi-Civita symbol, for which $\epsilon_{123} = 1$. The density matrix for a single qubit can be decomposed in terms of the Bloch vector v as:

$$\rho = \frac{1}{2}\left(\mathbb{1} + \sum_i v_i \sigma_i\right). \tag{132}$$

Since the Pauli matrices are Hermitian and traceless, a density matrix written as (132) is naturally Hermitian and with unit trace. The condition of positivity translates in its determinant being positive, which can be written as

$$1 - v \cdot v \geq 0. \tag{133}$$

In particular, the condition that a pure state represents a projector: $\rho^2 = \rho$, implies that for a pure state the norm of the Bloch vector is one. Thus the Bloch vector for pure states lies on a sphere known as the "Bloch sphere". Since the symmetric structure constants are zero for a two-level system, the condition that an evolution of the Bloch vector like Eq. (116)

$$\frac{\mathrm{d}}{\mathrm{d}t} v = \mathbf{L} v = (\mathbf{H} + \mathbf{D}) v, \tag{134}$$

is:

$$v \cdot \mathbf{L} \cdot v \leq 0. \tag{135}$$

This condition is obtained by observing that condition Eq. (133) must hold after an infinitesimal step in time [Eq. (116)] even if the initial state is pure. Plugging in Eq. (135) the expressions in Eqs. (118) and (119), one obtains:

$$||v||^2 \mathrm{Tr}(\kappa) - v \cdot \mathrm{Re}(\kappa) \cdot v - \frac{1}{2} \sum_k \epsilon_{ikj} \mathrm{Im}(\kappa)_{ij} v_k \geq 0, \tag{136}$$

where the skew-symmetric matrix $\mathbf{H}_{ij}$ has vanished because it is contracted with the symmetric matrix $v_i v_j$. The inequality (136) must hold independently of the basis $\{F_i\}$ of choice, which could be such that in it κ is diagonal. Since κ is Hermitian its eigenvalues are real, and in this situation the last term in the inequality (136) vanishes, since κ had been chosen diagonal from the beginning. If κ is semi-definite positive the inequality is automatically satisfied because of Cauchy-Schwartz inequality. One notes that the condition of κ being semi-definite positive is a sufficient, but not necessary condition for the positivity of the map $\mathbf{L}$. In fact complete positivity is a stricter requirement than positivity for a linear map. The characterization of simply positive map is in fact a difficult task [76]. In our works [27, 28], we study completely positive maps, which are much simpler than generally positive maps and have a direct physical interpretation in terms of coherent unitary evolution and dissipation effects.

3 Methods and machine learning

Introduction

As outlined in the introduction, this thesis focuses on employing machine learning methods to simulate the behaviors of various physical systems. Machine learning encompasses a wide range of computational techniques aimed at training a computer program to recognize patterns in data or to manipulate it to make predictions based on those patterns [77]. Machine learning algorithms are *variational* in nature. A variational algorithm is designed to adjust the internal parameters of an "Ansatz" function to achieve a desired result. This means that while the functional form of the Ansatz is fixed, it has a set of internal parameters that can be varied to optimize the function's behavior to closely match the desired outcome. A comparison that is often made to explain how a variational method works, is with the fitting of the coefficients of a polynomial to a given curve using the method of least squares [78, 21]. In this routine, for a fixed degree of the polynomial, the coefficients are adjusted to obtain the minimum distance with the data points. Similarly, the optimization of the internal free parameters of a trial wave function, to minimize the energy of a quantum many-body problem is performed in a method known as "variational Monte Carlo" [79, 80]. Machine learning algorithms are optimized by minimizing or maximizing a "loss" function [21]. This function indicates how well the model performs, reaching its optimal value when the model provides the most accurate predictions for the given task. This function is usually approximated via stochastic sampling. Stochasticity translates in the fact that only a random sample of the entire space of data is handled at a given time. This random sample is used to estimate the loss function and adjust it by approximating its gradient. This approach enables machine learning to optimize a large number of variational parameters and avoids the issue of getting stuck in local minima. These variational parameters are used as arguments of arbitrary differentiable functions (the activation functions), and thus the routine must come with the tools to automatically compute gradients of these functions. Artificial neural networks are a widely used model for variational Ansätze. These functions were initially developed as models for the computational study of neuron behavior in the brain [81], which is why they were named as such. However, they were later found to be of computational interest as “universal function approximators” [26]. This means that, given enough variational parameters, a neural network can approximate any function on a given interval with arbitrary precision.

An artificial neural network is built to process vector data, like image pixels, and generate a vector output, such as another image or a classification label. In general, an artificial neural network consists of a sequence of linear maps applied to an input, with each map followed by a nonlinear activation function acting element-wise on the output vector. Originally, the form of activation functions was chosen to describe the firing behavior of neurons. A neuron activates and “fires” when it detects a rapid change in the membrane potential of another neuron to which it is connected. The firing means that it propagates the sudden change in membrane potential on itself too. This kind of signal is called “action potential” [82]. Activation functions were hence originally chosen to be vanishing or “quiescent” for arguments smaller than a certain threshold and to be nonzero or “firing” afterwards.

3.1 Artificial neural networks

As just mentioned, artificial neural networks are inspired by the way animal brains process information. Similarly to how brains operate, these networks are composed of many interconnected functions, each receiving input from the previous one. The role of these basic functions mimics that of neurons in the brain.

Artificial neural networks accept an $n-$dimensional vector as input, such as an image, and map it to an output, like a label (e.g., ”cat” or ”dog”). They consist of various linear transformations and nonlinear ”activation” functions. The linear transformations roughly correspond to synaptic connections between neurons, while the activation functions represent the threshold for firing action potentials in a neuron. Just as neurons in actual brains can be ”trained” by experience to perform tasks, the parameters in the linear transformations of the networks can be trained using a variational principle. This training process optimizes the network to perform specific tasks effectively. In the following sections, we describe two important models of neural networks [83, 22]. The first is the perceptron, which models a single neuron. The second is the fully connected feed-forward neural network, of which the perceptron is the fundamental component. Despite their relative simplicity, these models are crucial for understanding the general functioning of neural networks and serve as a foundation for more sophisticated architectures. Moreover, in this thesis, the only kind of neural networks that are used are feed-forward architectures. Even if we presented artificial neural networks by making an analogy with the biological form of the neuron and the brain, the analogy must be taken with caution [22]. Modern neuroscience incorporates elements way beyond the scope of this thesis, and the form of neural networks should not be confused with the very sophisticated models used to actually gain understanding on the functioning of the brain. Moreover, modern machine learning, while originating in computational neuroscience, does now encompasses and is largely associated with the fields of statistics, physics and mathematics.

3.1.1 The perceptron

The fundamental unit of artificial neural networks is known as the "perceptron" [81]. The functional design of the perceptron is influenced by the biological structure of a single neuron in the brain. For an incoming signal $s^{(0)}$ the perceptron computes a weighted sum of the entries of $s^{(0)}$ with weights w, adds to it a parameter b known as the bias, and applies to this quantity a nonlinear activation function f. Given an input vector $s^{(0)}$, the mathematical expression for a perceptron is:

$$s^{(1)} = f\left(\sum_{i=1}^{n} w_i s_i^{(0)} + b\right) . \tag{137}$$

The rationale for this functional form is that like all animal cells, neurons have a difference in voltage between the interior and the exterior of their cell membrane, known as the "membrane potential." In biological terms, neurons can sense changes in the membrane potential of other neurons through their terminal appendices called dendrites (cf. Fig. 6 (b)), and propagate the information to other neurons. In the perceptron model, this feature is represented by taking an n-dimensional input vector $s^{(0)}$, where the discrete quantity n represents the number of dendrites. The individual properties of each dendrite are captured by multiplying each entry $s_i^{(0)}$ by the weight

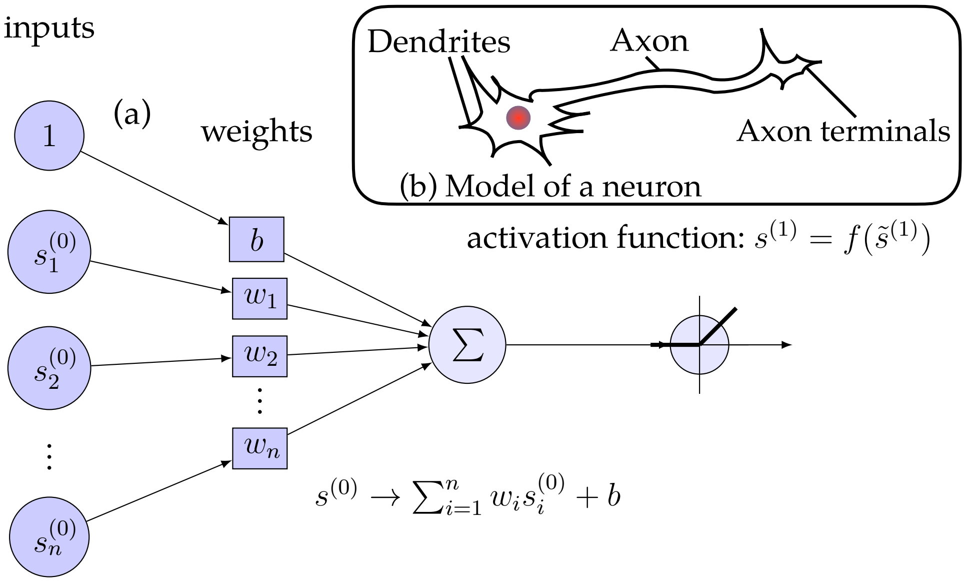

Figure 6: **Model of the perceptron.** In a perceptron (a), a $n-$dimensional input vector $s^{(0)}$ is multiplied element-wise by a vector of weights, each representing the action of a dendrite in a neuron. The specific action of the neuron's body is encoded by a bias b. The general functional form of this model is inspired by the biological structure of neuron cells in animal brains (b). These cells are formed by a main body connected to various dendrites which receive information and transmit it along a wire-like part of the cell called axon.

w_i, specific to the i-th dendrite. If the weight w_i is very small, the intensity of the incoming signal must be large for it to be propagated. Conversely, for highly sensitive ”dendrites” with very large weights (relative to the typical value of incoming signals), even a small incoming signal is easily propagated. These neurons can then propagate the incoming signal to other neurons by transmitting the change in membrane potential along a wire-like part of the cell called the axon. Each dendrite in a neuron has an individual threshold for propagating the incoming signal. This thresholding is modeled in the perceptron by the activation function f. Therefore, the most commonly used activation functions in machine learning are of “sigmoid” form, that is, characterized by exactly one inflection point at zero.

The activation function also introduces nonlinearity to the model. Without the activation function, $s^{(1)}$ would simply be an inner product multiplication between two vectors $s^{(0)}$ and w plus a bias b, which would be insufficient to capture the nonlinear behavior necessary for representing functions of arbitrary complexity. In the perceptron model, the primary purpose of the bias b is to adjust the output along with the weighted inputs, allowing the activation function to shift to the left or right. For activation functions that have inflection points at zero, such as sigmoids, this shifting enables the perceptron to model functions beyond those that pass through the origin. Moreover, during the training process, the bias helps the perceptron to learn and adapt to the data more effectively [22].

3.1.2 Feed-forward neural networks

When m perceptrons $(s_1^{(1)}, ..., s_m^{(1)})$ act in parallel on an incoming vector $s^{(0)}$ they form a “layer”. The i-th perceptron in the layer applies element wise an activation function $f^{(1)}$ to the linear application of the weights and biases, to obtain:

$$s_i^{(1)} = f^{(1)}\left(\sum_{j=1}^{n} w_{ij} s_j^{(0)} + b_i\right) \tag{138}$$

The action of the weights and biases is that of a linear map W acting on the vector $x \equiv (1, s^{(0)})$, and for simplicity, this notation is adopted in what follows. The element 1 at the beginning of the vector is multiplied by the biases $(b_1, ..., b_m)$ of the layer. These biases form the first row of of the matrix W. The other entries of W are given by the

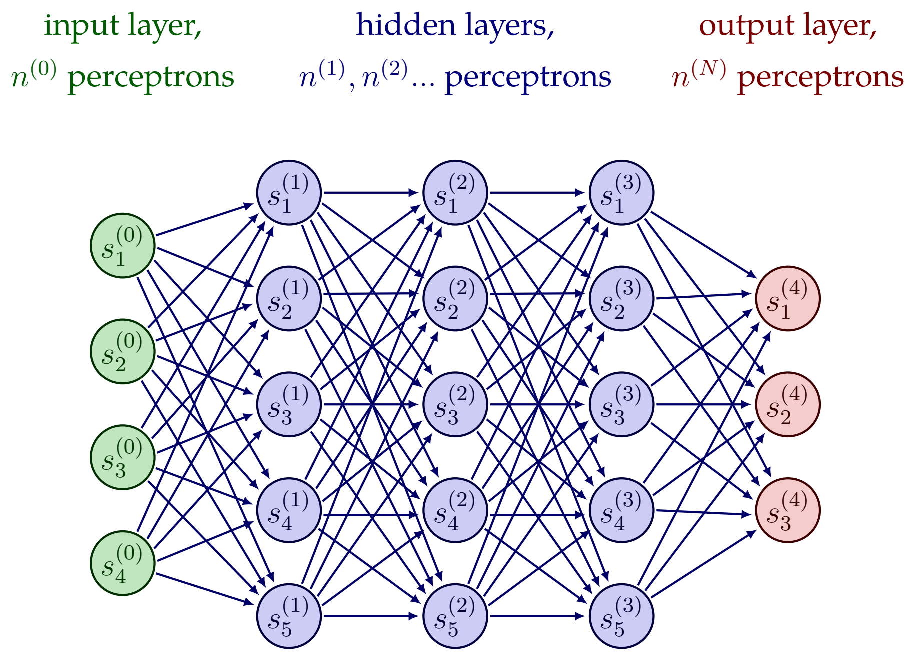

Figure 7: **Fully connected feed-forward neural network**. In the depicted neural network, there are $N = 5$ layers of perceptrons [cf. Eq. (141)]. Arrows incoming to the perceptron s^l_m and originating from a perceptron s^k_l indicate a multiplication by a specific weight and summation by the respective bias. The $i-$th layer has a number $n^{(i)}$ of perceptrons, with $i = 0$ for the input layer, and $i = N$ for the output layer. Deep neural networks have large representational power due to the combinatorially large number of possible paths that information can take in the networks. This is reflected in the figure by the number $n^{(0)}...n^{(N)}$ of possible paths that one can take staying on the arrows and only moving from left to right.

weights w. In total, the linear transformation W in terms of the biases and the weights reads as:

$$W_{ij} = \begin{cases} 1 \text{ for } i = 1,\ j = 1, \\ 0 \text{ for } i = 1,\ j > 1, \\ W_{i1} = b_i \text{ for } i > 1, \\ W_{ij} = w_{ij} \text{ for } j > 1,\ i > 1 \end{cases} \tag{139}$$

Since the same function f is applied by all perceptrons in the layer, Eq. (138) can be made simpler by introducing the Hadamard product notation "$\odot$" for the element-wise application of an activation function f:

$$s^{(1)} = f \odot W \cdot x, \tag{140}$$

Having computed $s^{(1)}$ from $s^{(0)}$, nothing prevents continuing this procedure to obtain successive perceptron layers $s^{(2)}$ from $s^{(1)}$ and so on. When many layers of perceptrons are "fed" one to the other they form a very important model of machine learning, namely, the feed-forward fully connected network [22]. Its importance rests in its ease to train, and also in its versatility. The label "feed-forward" means that there is no recurrence, i.e. the same perceptron is used only once in the computation of the output. This contrasts with another family of neural networks called "recurrent neural networks" [84] in which the flow of information in the neurons can occur recursively. We will not discuss this latter class of neural networks in the present work, as the results reported were achieved using only feed-forward neural networks. These networks, being simpler than their recurrent counterparts, allow for greater interpretability from a physical standpoint, thereby enabling more direct inference of the properties of the physical system under study.

Let us call all the entries of the various weights and biases in an N layers network as θ. The output $s^{(N)}$ of a feed-forward network F_θ is given by:

$$F_\theta(x^{(0)}) \equiv f^i \odot W^i ... f^N \odot W^N \cdot x^{(0)}, \tag{141}$$

where the input of the network is intended to be $s^{(0)}$, and $x^{(0)} = (1, s^{(0)})$.

3.2 Universal approximation theorem

The reason why neural networks are widely adopted in machine learning is justified by a series of results that go under the name of "universal approximation theorems" [26, 25]. These theorems ensure that both single-layer and multi-layer feed-forward neural networks, provided with enough perceptrons, are able to approximate any function on a closed and bounded interval in an n-dimensional real space, within any nonzero error interval[1]. On one side, this theorems are of great theoretical importance, as they give confidence that neural networks have in principle enough representational power to approximate any function, and can thus serve as variational Ansätze for an extremely wide class of functions. On the other, the theorems proved in [25] and [26] do not provide an upper bound on the number of variational parameters needed for the network to be able to approximate the objective function within the desired error bound. Results on the number of the width of the layers, and the depth of the network needed to achieve universality in the power of representing functions exist [85, 86, 87], but often require assumptions on the regularity of the loss function, which are not always possible to satisfy. In practice, the number of layers and their width is still manually adjusted, and treated as a free parameter of the problem. There are instances of functions, that one wants to approximate, where there is enough knowledge of the problem and a precise bound on the number of variational parameters needed is available. In particular, in one of the papers of the present thesis [27], the network employed was effectively one-dimensional, and this also made the results largely interpretable from a physical standpoint. In [20] instead more representational power was needed to learn the desired functions, and *deep* feed-forward neural networks were used. Deep neural nets [22, 88] have high power in representing functions with a limited number of perceptrons. In such networks many layers of perceptrons that act in parallel on each of their inputs are fed forward one to the other. To intuitively understand why deep networks are highly capable of representing functions, consider that if only one perceptron is active in each layer at any given time, there is an exponentially large number of possible activation paths. For example, if the input layer has $n^{(0)}$ perceptrons, perceptrons, the first hidden layer has $n^{(1)}$ and so on, the number of possible activation paths is $n^{(0)}n^{(1)}...n^{(n)}$ [cf. Fig. 7], where an activation path is a possible selection of

[1] More generally, for any Borel measurable function.

perceptrons that got activated for a given input. As highlighted in [22], while neural networks theoretically possess sufficient representational power to approximate the objective function accurately, the practical implementation of an algorithm that achieves this is a delicate matter. Different tasks in machine learning require specific neural network architectures and appropriate training procedures to achieve effective results. A good practice is to use fewer internal parameters in neural networks. This approach not only reduces computational time and resource requirements but also helps minimize overfitting. Overfitting occurs when a neural network learns specific details and noise in the training data rather than the general features and functional forms that it should ideally learn [22, 21, 89]. This leads to poor generalization where the model performs well on training data but fails on unseen data. Overfitting can be likened to fitting a dataset with a high-degree polynomial. While such a polynomial can perfectly fit noisy training data points, it often fails to generalize to new data points from the same dataset that were not part of the training process.

In [27], the neural network was deliberately designed to be one-dimensional and constrained with minimal parameters. This choice was made to ensure the model focused on capturing the essential physical aspects of the problem, thereby reducing the risk of overfitting and improving its ability to generalize to new data.

3.3 The loss function

The variational optimization of the parameters of a neural network is achieved by minimizing a "loss" function. A very common choice for a loss function L is a distance $d(\cdot,\cdot)$ between the output $s^{(N)} = F_\theta(s^{(0)})$ of the neural network F_θ, with variational parameters θ and an objective y:

$$L[\theta] = d(F_\theta(x), y). \tag{142}$$

The general task of machine learning algorithms is to minimize this loss. A typical example is a classification task, such as one where the object x to be classified are photos of cats and dogs, according to the animal that is represented. The desired output y of the network is a label, for example, 0 for cats and 1 for dogs. Since F_θ is a composition of continuous maps, the output of the network will be continuous, but it can be restricted, with an appropriate choice of the last activation function, to be bounded, $F_\theta(x) \in [0, 1]$. After a successful training, the network can correctly classify unseen images, if the labels it assigns them are nearer to 0 in the case of cats, and to 1 in the case of dogs. For this case of tasks, what one is looking for is a generalization property, that is, to enable the network to correctly classify previously unseen photos of cats and dogs.

3.4 The gradient descent algorithm

A tool that makes variational algorithms particularly compelling, and has wide application beyond machine learning, is the gradient descent method [90, 21]. Gradient descent is an iterative algorithm used to minimize an objective function $L[\theta]$, whose arguments are a set of variational parameters θ. The algorithm requires a suitable choice of a "learning rate" η. At each iteration $h+1, h+2, ...$, the computation of the gradient $\nabla L[\theta_h]$ is needed. This gradient is employed to update the parameters θ_h by varying them as:

$$\theta_{h+1} = \theta_h - \eta \nabla L[\theta_h]. \tag{143}$$

If the learning rate η is small enough so to justify a first-order Taylor expansion in it, one has:

$$L[\theta_{h+1}] = L[\theta_h] - |\eta \nabla L[\theta_h]|^2, \tag{144}$$

meaning that one is left with a monotone sequence $L[\theta_{h+1}] \leq L[\theta_h] \leq ... \leq L[\theta_1]$. Initialization $\theta \leftarrow \theta_1$ is extremely important for the gradient descent method to converge, and its order of magnitude should match the features of the loss at hand. Various schemes have been developed to initialize weights and biases in neural networks. A common approach is to initialize them with random numbers, typically drawn from either a uniform distribution over the interval [0, 1] or a normal distribution centered around zero with unit variance. A general caveat of this method is that there is no guarantee of reaching a global minimum (or maximum). This is due to the fact that the convergence of the gradient descent algorithm is only ensured in the case of a *convex* function [2] [90, 21] . The algorithm may seem to converge, in the sense that no change is made in the value of $L[\theta_i]$ and $L[\theta_{i+1}]$; but only because it is stuck in a local minimum of the

[2] If the line segment connecting any two distinct points on the graph of the function lies above or on the graph between those two points, then the function is convex [90]

loss $L[\theta]$, or in a saddle point. However, in practical scenarios, the adoption of a *stochastic* version of gradient descent, makes the algorithm more susceptible to saddle points than local minima [91, 88]. A significant issue about using gradient descent methods in machine learning is the impact of the learning rate on the overall performance. In principle, a larger learning rate implies faster convergence, because one is taking "large jumps" towards a minimum in the loss function's. The fact of taking very large steps in the optimization process makes the algorithm more prone to overlook finer details and fail to reach the true minimum, getting stuck on plateaus. Conversely, a smaller learning rate can provide a finer resolution of the loss landscape but leads to longer convergence times and might be too small to overcome barriers in the loss. Parameters like the learning rate, denoted as η in Eq. (143), which are not adjusted during training, are referred to as hyperparameters in the machine learning community. Other examples of hyperparameters include the number of layers in a fully connected feed-forward network and the number of perceptrons in each layer.

To improve the convergence of the loss, alternative methods involving second-order derivatives can be employed [92]. The predominant optimization algorithms in this category draw inspiration from the Newton-Raphson method for locating function zeros [90, 93]. These techniques generally offer quicker convergence and enhanced stability. In the case of multivariate functions, their second-order derivatives form Hessian matrices. In these algorithms, it is often necessary to invert the Hessian matrices. This poses a significant challenge because diagonalizing large matrices is computationally intensive. In many practical machine learning applications, gradient descent-based methods are powerful enough to achieve the desired outcomes, rendering second-order derivative approaches unnecessary.

3.5 The back-propagation algorithm

An important feature of feed-forward neural networks is that after the element-wise application of an activation function, the output vector is manipulated *linearly* by the successive layer. This enabled the creation of an efficient algorithm known as back-propagation [94], to compute the derivatives of the loss function $L[\theta]$. Specifically, for a feed-forward neural network of N layers:

$$F_\theta(x) = f_N \odot W_N \cdot ... \cdot f_1 \odot W_1 \cdot x, \tag{145}$$

the loss between the objective y and the input x has the form:

$$L[\theta](x, y) = \tilde{L}[\theta](F_\theta(x), y). \tag{146}$$

The total derivative with respect to the argument x of $L[\theta](x, y)$ or the parameters θ can be efficiently computed by repeated application of the chain rule. In the following we describe more in detail this routine, which is called "the back propagation algorithm". Let us first consider the case where the loss function is just the Euclidean distance between the output computed by the neural network from the input $s^{(0)}$ and the desired output y. Moreover, consider a neural network with just one layer: $F_\theta(x) = f \odot W \cdot x$, and as before $x = (1, s^{(0)})$. Then the loss function has the form:

$$L[\theta](x, y) = ||f \odot W \cdot x - y||^2. \tag{147}$$

Here the Euclidean norm of a vector u is intended as $||u||^2 \equiv \sum_i u_i^2$. The total derivative of the expression in Eq. (147) with respect to x is:

$$\frac{\mathrm{d}}{\mathrm{d}x} L[\theta](y, x) = 2(f \odot W \cdot x - y) \odot (\hat{f}') \cdot W, \tag{148}$$

while the total derivative with respect to W is:

$$\frac{\mathrm{d}}{\mathrm{d}W} ||f \odot W \cdot x - y||^2 = 2(f \odot W \cdot x - y) \odot (\hat{f}') \otimes x. \tag{149}$$

Here we introduced the outer product of two vectors as $[x \otimes y]_{ij} \equiv x_i y_j$. The derivative f' of f is evaluated at Wx, that is, I write $\hat{f}' \equiv f' \odot Wx$. The derivative of the loss $\frac{\partial}{\partial z}\tilde{L}'(z, y) = 2(z - y)$ is evaluated at $z = f \odot W \cdot x$. The gradient, which has the interpretation of the direction of highest variation in the loss function, is the dual of the total derivative, that is, its transpose. Calling the transpose of the vector $2(f \odot W \cdot x - y)$, simply as ∇L, one gets that the gradient of the loss *with respect to* x is:

$$\nabla_x L = [2f \odot (Wx - y) \odot \hat{f}' \cdot W]^T = W^T \cdot \hat{f}' \odot \nabla L. \tag{150}$$

For feed-forward networks, the argument of the loss is a neural network of N layers, evaluated at the input x. Each of the layers $1, 2, ..., N$ can be passed to a different activation function $f^1, f^2, .., f^N$ as:

$$L[\theta](x) = L(f^N \odot W^N ... f^1 \odot W^1 \cdot x, y). \tag{151}$$

Let now $\hat{f}^l$ be the "activation" of the $l-$th layer:

$$\hat{f}^l \equiv f^l \odot W^l \cdot f^{l+1} \odot W^{l+1} ... f^N \odot W^N \cdot x, \tag{152}$$

Notice that at the zeroth layer $f^0 \equiv x$. Now, let us define g^l as the gradient of L with respect to $\hat{f}^l$:

$$g^l \equiv \left(\frac{\partial}{\partial \hat{f}^l} L \right)^T = (W^l)^T \cdot (\hat{f}^l)' ... (W^N)^T \cdot (\hat{f}^N)' \odot \nabla L. \tag{153}$$

This "relative error" at the layer l serves as a generalization of the gradient of L with respect to x in Eq. (150). Since the transpose of the external product of two vectors x and y is simply $[x \otimes y]^T = x^T \otimes y^T$, by comparing Eqs. (153) and (149) one notices that:

$$\nabla_{W^l} L[\theta](x) = g^l \otimes (\hat{f}^{l-1})^T. \tag{154}$$

From the definition of g^l in Eq. (153), the quantities g^l can be computed in a recursive manner as

$$g^{l-1} = (W^{l-1})^T \cdot (\hat{f}^{l-1})' \odot g^l. \tag{155}$$

This gradients g^{l-1} are computed in descending order of l, that is starting from $g^{N-1} = (W^N)^T \cdot (\hat{f}^N)' \nabla L$ and then successively multiplying by the vector $(W^{l-1})^T \cdot (\hat{f}^{l-1})'$ on the left. This fact justifies the name "back-propagation": the gradients are computed in reverse order with respect to the succession of layers. Moreover, back-propagation is done after a *forward pass*, which means it is performed after the computation of $F_\theta(x)$ has been completed. What makes the algorithm efficient is that during the forward pass, also the activation functions $\hat{f}^l$ and their derivatives $\hat{(}f^l)'$ are computed and stored in memory. These derivatives will be successively used to compute the gradients $\nabla_{W^l} L[\theta](x)$. These gradients are then utilized for updating the weights W (cf. Eq. (143)).

3.6 Stochastic gradient descent

When training a neural network with large datasets, computing the gradients necessary for minimization of the loss function can pose significant challenges. To address this issue, a stochastic version of the gradient descent algorithm was introduced [95]. This stochastic gradient descent (SGD) algorithm is applicable in scenarios where the sample space is extensive and can be sampled. Specifically, for a loss function of the form:

$$L[\theta] = L[\theta](x, y), \tag{156}$$

stochastic gradient descent operates by using only a subset (or mini-batch) of the sample space at each iteration. The sample space can represent a dataset, but it can also extend to a probability space. The latter case is particularly relevant in areas such as reinforcement learning, where neural networks are often employed as probability distributions and sampled accordingly. In our current context, we focus solely on scenarios where the sample space consists of a given dataset. The utilization of a random subset from the entire sample space introduces stochasticity into the gradient computation, resulting in potentially noisy gradients. To mitigate this variability, a family of optimization algorithms known as "optimizers" has been developed. These optimizers aim to stabilize the training process by computing a moving average of the steepest descent direction and using it to regularize the behavior of the loss function during training. These moving averages are called "momentum terms", because they give "momentum" to the gradient direction, helping to minimize the loss even when it encounters sudden increases.

Optimizers like Adam [96], RMSprop [3], and Adagrad [97] are examples of such techniques that enhance the efficiency and stability of gradient descent in deep learning. They achieve this by adjusting learning rates adaptively and incorporating momentum terms. To elucidate the fundamental concept behind optimizers like Adam, as adopted in the present work in [27], we briefly review its core principles. The objective of the Adam optimizer is to minimize the loss function $L[\theta](x, y)$ iteratively. At each step t of the iteration, the gradient

$$G_t = \nabla_\theta L[\theta](x, y). \tag{157}$$

is computed. The gradient is computed with stochastic sampling of the data, meaning that the values y vary randomly at each iteration. The loss function itself thus fluctuates randomly, resulting in a noisy gradient. The random fluctuations in the gradient

[3] Unpublished. First explained in a Coursera course by G. Hinton at https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

can cause it to point in suboptimal directions within the loss landscape. To address this issue, adaptive optimizers replace the gradient with a form of moving average. In Adam, this is achieved by computing the first moment M_t of the gradient G_t, weighted to prioritize recent gradients:

$$M_t \sim \mathbb{E}[G_t], \qquad M_t = \beta_1 M_{t-1} + (1 - \beta_1) G_t. \tag{158}$$

Here, β_1 is a hyperparameter controlling the exponential decay rate of gradient estimates that were computed previously. Additionally, Adam computes a moving average of the second moment of the gradient, referred to as the uncentered variance Σ_t. This quantity is defined such that $\Sigma_t \sim \mathbb{E}[G_t^2]$, where G_t^2 denotes the element-wise square of G_t:

$$\Sigma_t = \beta_2 \Sigma_{t-1} + (1 - \beta_2) G_t^2. \tag{159}$$

Both M_t and V_t are initialized as vectors of zeros. Without correction, M_t would be biased towards G_t, essentially mimicking standard gradient descent. To correct this bias, Adam [96] introduces bias-corrected estimates of the first and second moments:

$$\hat{M}_t = \frac{M_t}{1 - \beta_1^t}, \qquad \hat{\Sigma}_t = \frac{\Sigma_t}{1 - \beta_2^t}. \tag{160}$$

The parameters θ_t are updated using these corrected moment estimates, normalized by the square root of the second moment estimate:

$$\theta_t = \theta_{t-1} - \eta \frac{\hat{M}_t}{\sqrt{\hat{\Sigma}_{t-1}} + \epsilon}. \tag{161}$$

Here η denotes the learning rate, and ϵ is a small constant added to make the algorithm less prone to numerical instability. This normalization constrains the size of parameter updates by bounding them within the learning rate η, since $|\hat{M}_t / \hat{\Sigma}_t| \leq 1$, ensuring stability during optimization. As detailed in the original publication [96], Adam establishes a region around the current parameter values θ_t within which the update based solely on raw gradient estimates cannot be used (as it is not trustworthy) and should not be used.

4 Results and publications

In this thesis, we aim to describe two classes of different, yet related, dynamics: the dynamics of open quantum systems and the dynamics of stochastic order parameters. To this aim, we employ machine learning techniques, which offer a powerful avenue for studying these problems due to their significant representational capacity. A primary drawback of machine learning techniques is their "black box" nature, which means there is no clear understanding of the physical interpretation that can be applied to these models. Our goal was to maintain the model's architecture as simple as possible and to ensure it was physically motivated, allowing for an understanding of the models in terms of physical quantities.

In our **first work** [27], building on the work [35], we employed a variational algorithm that heavily draws from machine learning to optimize the parameters of a dynamical generator for an open quantum system. The open quantum system under consideration is composed of two $\frac{1}{2}-$spins immersed in a spin chain. The spin chain evolves according to a few paradigmatic models of condensed matter physics and quantum optics. The dynamics we aim to learn are highly constrained by physical considerations. Specifically, we manually restrict the set of dynamical generators that the algorithm can learn to those that exhibit Lindbladian dynamics. This ensures that the generator of the dynamics is completely positive, and thus the quantum state preserves its physical properties during the time evolution. The restriction to a Lindbladian generator allows for a direct interpretation of the parameters in the network. We use this approach to investigate the parameter regime in the Hamiltonian of choice for the time evolution, where a Markovian approach dictated by a Lindbladian generator can correctly predict the dynamics. At higher coupling strengths of the Hamiltonians, as expected, the Markovian approximation needed for a Lindbladian description of the dynamics breaks down. We adopt an error metric between the exact dynamics and those predicted by the network to investigate where this breakdown occurs. The interpretability of the learned parameters in physical terms allows us to compare the stationary state predicted by the learned generator (which corresponds to the eigenstate with a vanishing eigenvalue of the generator) with the average value of the observables of the dynamics at higher times. Surprisingly, even in cases where the Markovian approximation does not appear to hold, the predicted stationary state does not differ too much from the time averaged observables. We note that time averaging is needed because of small oscillations remaining at long times due to finiteness of the systems under consideration. Additionally, we examine the relaxation time of the dynamics to equilibrium, which is dictated by the smallest nonzero eigenvalue of the learned generator, known as the gap.

In experimental settings, the expectation values of local observables are inevitably affected by noise due to the limited number of measurements possible. Surprisingly, even when using synthetic data subject to this kind of projective noise, our method provided robust results, as reported in our **second work** [28]. In that study, we develop further the methods designed in our first work, such that they can be applied to synthetic data that more accurately simulate actual experimental conditions. Specifically, we developed a more suitable machine learning routine to train the dynamical generator, allowing it to learn the reduced quantum dynamics from simulated noisy projective measurements. We applied the developed method to the one-dimensional quantum Ising model with both longitudinal and transverse fields. The data provided to the network correspond to the evolved coherence vector measured via N projective measurements at M random points in time. Surprisingly, even though the training dataset is much smaller and noisier than the one used in our first work, when the hypotheses justifying the use of a Markovian dynamical generator are met, the method consistently reproduces the original dynamics. We demonstrate the applicability of the method across a wide range of parameters in the Hamiltonian. This work not only demonstrates the method's accuracy in reproducing the dynamics of local quantum degrees of freedom but also shows its applicability in inferring the mechanisms of decoherence that affect devices used for quantum computation and quantum simulation.

The handling of noise by neural networks sparked our interest in scenarios where noise is naturally present, and whose source differs from the projection noise of quantum measurements. Hence, we turned our attention to classical systems both in and out of equilibrium and studied the capabilities of machine learning techniques to encode and analyze systems whose evolution at the microscopic level is probabilistic. We focused on two paradigmatic models of statistical physics both in and out of equilibrium. Specifically, in our **third work** [20], we employed deep neural networks to learn the form of the drift (the directed force) and the diffusion term (the noise) in the stochastic dynamics of two different classical dynamical systems. Specifically, we focused on the Ising model with Metropolis Hasting sampling and Glauber dynamics [47], and the classical contact process [48]. The Ising model is an equilibrium model, as it inherently possesses an energy functional, that is, its Hamiltonian. Glauber dynamics involves flipping individual spins at rates that satisfy detailed balance, ultimately leading the system to reach the canonical equilibrium state at stationarity. We "choose" a dynamics, because the Ising model is not inherently dynamical, it is just provided with a Hamiltonian. The stationary points of this model are thus dictated by the "minima" of its free energy functional. For the contact process, which simulates the infection spread in a non-immunized population, the model isn't defined by a Hamiltonian. Instead, it relies on a set of dynamic rules governing the probabilistic evolution of the spins. Consequently, there is no energy functional involved, making the model inherently a nonequilibrium system. In both cases, the devised machine learning techniques were able to accurately recover the original dynamics, when integrated via an Euler-Maruyama scheme [98]. Moreover, the drift term gave valuable information about the stationary values of the dynamics, as its zero correspond to a steady state. Using this information, we were able to give an estimate of the critical point and the critical exponent for the order parameter. In the following section, we report the full publications.

First publication

New Journal of Physics

The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
4 February 2022

REVISED
13 June 2022

ACCEPTED FOR PUBLICATION
4 July 2022

PUBLISHED
21 July 2022

PAPER

Inferring Markovian quantum master equations of few-body observables in interacting spin chains

Francesco Carnazza[1],[*], Federico Carollo[1], Dominik Zietlow[2], Sabine Andergassen[1], Georg Martius[2] and Igor Lesanovsky[1,3,4]

[1] Institut für Theoretische Physik and Center for Quantum Science, Universität Tübingen,Auf der Morgenstelle 14, 72076 Tübingen, Germany
[2] Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany
[3] School of Physics and Astronomy, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[4] Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: francesco.carnazza@uni-tuebingen.de

Keywords: Lindblad dynamics, machine learning, interacting spins, completely positive dynamics

Abstract

Full information about a many-body quantum system is usually out-of-reach due to the exponential growth—with the size of the system—of the number of parameters needed to encode its state. Nonetheless, in order to understand the complex phenomenology that can be observed in these systems, it is often sufficient to consider dynamical or stationary properties of local observables or, at most, of few-body correlation functions. These quantities are typically studied by singling out a specific subsystem of interest and regarding the remainder of the many-body system as an effective bath. In the simplest scenario, the subsystem dynamics, which is in fact an open quantum dynamics, can be approximated through Markovian quantum master equations. Here, we formulate the problem of finding the generator of the subsystem dynamics as a variational problem, which we solve using the standard toolbox of machine learning for optimization. This dynamical or 'Lindblad' generator provides the relevant dynamical parameters for the subsystem of interest. Importantly, the algorithm we develop is constructed such that the learned generator implements a physically consistent open quantum time-evolution. We exploit this to learn the generator of the dynamics of a subsystem of a many-body system subject to a unitary quantum dynamics. We explore the capability of our method to recover the time-evolution of a two-body subsystem and exploit the physical consistency of the generator to make predictions on the stationary state of the subsystem dynamics.

1. Introduction

Artificial neural network methods have established themselves as a versatile tool in many areas of physics [1–3]. Just to mention few examples, their application ranges from the classification of phases of matter [4] over scattering and reflectivity analysis [5, 6] all the way to the learning of topological states [7]. Moreover, neural networks have been effectively employed to encode the quantum state of both closed [1, 8] and open [9–12] quantum systems.

Recently, machine learning methods have also been applied to obtain effective dynamical generators for local degrees of freedom of a many-body quantum system [13]. The idea is the following: let us assume that one is interested in the time-evolution of a subsystem S of a larger quantum system, as depicted in the sketch in figure 1(a). For convenience, we refer to the complement of S, i.e. the remainder of the quantum system, as the 'bath' denoted by B. The full information about properties of S is contained in the so-called reduced quantum state ρ_S, obtained by 'integrating out' the bath B. Assuming that the full quantum system is subject to a Hamiltonian quantum dynamics implemented by the unitary operator U_t this simply means that the time-evolved reduced quantum state of S is given by $\rho_S(t) = \mathrm{Tr}_B(U_t\rho U_t^\dagger)$ (see figure 1(b)), where

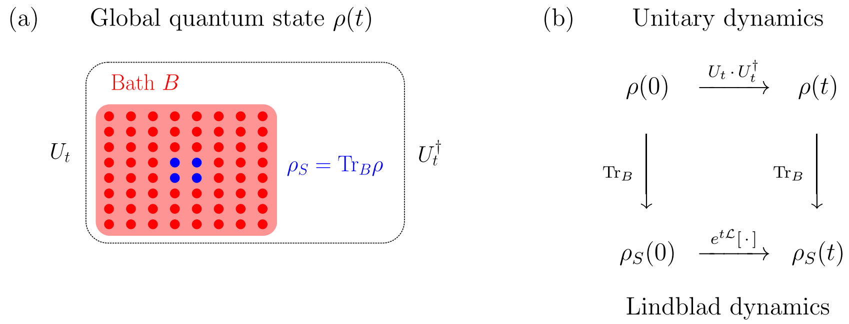

Figure 1. Effective Lindblad generator for the dynamics of a subsystem embedded in a many-body quantum system. The full quantum state ρ_S of a many-body system subject to a unitary quantum dynamics evolves according to the unitary (Schrödinger) operator U_t as $\rho(t) = U_t\rho(0)U_t^\dagger$. (a) Properties about a subsystem S of such a many-body system are described by the so-called reduced quantum state ρ_S, which is obtained by tracing out the remainder of the system (or, as we call it here, the bath), $\rho_S = \mathrm{Tr}_B\rho$. (b) The time-evolution of the reduced subsystem state $\rho_S(t)$ can thus be obtained by tracing out the bath in the evolved many-body quantum state: $\rho_S(t) = \mathrm{Tr}_B\rho(t)$ (see rightmost arrow in the diagram). If the emergent subsystem dynamics is Markovian, the time-evolution of any initial state of the subsystem, ρ_S, can be formulated as an open quantum dynamics, characterised by a time-independent generator $\mathcal{L}$, which effectively accounts for the presence of the bath (see bottom arrow in the diagram).

Tr_B indicates the trace over the degrees of freedom of B. Under certain physical conditions, e.g. weak S–B coupling and very large bath B, such a subsystem time-evolution is well captured by Markovian open quantum dynamics [14–16]. Formally, this entails the existence of a time-independent generator $\mathcal{L}$ of a dissipative quantum dynamics such that (see also figure 1(b))

$$\rho_S(t) = \mathrm{Tr}_B\left(U_t\rho(0)U_t^\dagger\right) \cong e^{t\mathcal{L}}[\rho_S(0)], \tag{1}$$

where $\rho(0) = \rho_S(0) \otimes \rho_B(0)$ and $\rho_{s/B}$ is the initial state of the subsystem/bath, respectively. Interestingly, by exploiting simple neural network architectures, it was shown that such an approximate generator $\mathcal{L}$ can be found also beyond the typically considered settings, and that it can satisfactorily describe the subsystem dynamics whenever non-Markovian effects are negligible [13]. These findings are relevant since they allow one to infer, from the time-evolution of local degrees of freedom, the physical processes underlying the dynamics of quantum (sub)systems. This can, for instance, deliver insight into the dynamical effects behind an order-parameter time-evolution in non-equilibrium phase transitions or shed light on relaxation and thermalization effects in closed systems [17]. Moreover, as we show here, since the neural network provides the dynamical generator of a Markovian time-evolution, the latter can be used to directly target stationary properties of the subsystem. However, the generator $\mathcal{L}$ learned by the neural network architecture of reference [13] is not guaranteed to implement a valid, i.e. physical, dynamics (see below for details). This is a problem since it can lead to physically inconsistent predictions on the dynamics or on stationary properties of the subsystem state [18–20].

In this paper we use a matrix parametrization of the Lindblad generator, and optimise it using data generated with exact numerical simulations. One can see this as an encoding of the matrix elements of the generator into the variational parameters of a simple neural network (see figure 2 below). This interpretation allows us to 'learn' the optimal variational parameters by means of the standard tools of machine learning. For actual implementation and optimization of the network, we used the automatic differentiation toolbox Pytorch [21]. Importantly, our parametrization is such that the learned dynamical generator $\mathcal{L}$ is constrained to be physically consistent. Such a generator is meant to approximate the dynamics of the subsystem state $\rho_S(t)$ in the sense of equation (1). While the neural network, which is agnostic to the fact that the dynamics of $\rho_S(t)$ is a result of tracing out the degrees of freedom of the bath B, always retrieves a time-independent generator $\mathcal{L}$. How well such an approximation can reproduce the dynamics of the subsystem S (see equation (1)) clearly depends on the dynamical regime considered, e.g. strong or weak interactions, determining whether the subsystem dynamics is effectively Markovian or not. Here, we show how the learned generator $\mathcal{L}$ can be used both to extrapolate the subsystem dynamics to times which have not been used to train the network and to predict stationary state properties of the subsystem. We illustrate our ideas by applying our neural network architecture to the reduced dynamics of a two-spin subsystem embedded in a larger quantum spin chain.

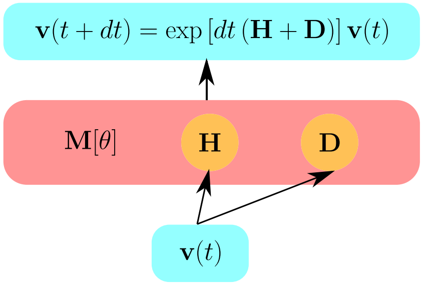
$\mathbf{v}(t+dt) = \exp\left[dt\left(\mathbf{H}+\mathbf{D}\right)\right]\mathbf{v}(t)$
$\mathbf{M}[\theta]$
H
D
$\mathbf{v}(t)$

Figure 2. Architecture of the interpretable artificial neural network layer. The network that we exploit takes as an input the coherence vector at time t, $\mathbf{v}(t)$, and outputs its value at the next discrete time step: $\mathbf{v}(t + \mathrm{d}t) = \mathbf{M}[\theta]\mathbf{v}(t)$. Here $\mathbf{M}[\theta] = e^{dt(\mathbf{H}+\mathbf{D})}$. The goal is to find the optimal parameters θ defining the matrices and $\mathbf{D}$ by training the network with exact simulation data on the input and output coherence vector. The network is parametrized in a way that the learned generator is automatically guaranteed to be physically consistent. The elements of the matrices $\mathbf{H}$ and $\mathbf{D}$ are functions of the variational parameters—the weights of the network—$\theta = \{\boldsymbol{\omega}, X, Y\}$ through equations (10) and (11) and must be optimized according to the loss function, equation (12).

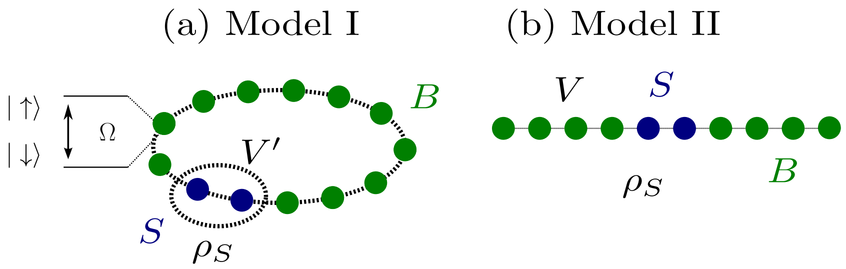

Figure 3. Spin-chain models. We exploit our neural network architecture to learn the reduced dynamics of a two-spin subsystem (highlighted in the panels) embedded into two different many-body quantum spin chains, one with closed boundary conditions (model I) and one with open boundaries (model II). In both settings, the global time-evolution is governed by a Hamiltonian, H_{I} (cf equation (15)) and H_{II} (cf equation (16)), respectively. (a) Quantum spin chain with closed boundary conditions. In this case, the Hamiltonian H_{I} contains only nearest-neighbour interactions. As free parameters for the model, we consider the inverse temperature β for the initial state of the bath spins, and the interaction strength of the subsystem within itself and with the bath V'. These parameters are varied and the error I_{Err}, between the exact dynamics and the (Markovian) one predicted by the neural network model is quantified through equation (19). (b) Quantum spin chain with open boundary conditions. The full system Hamiltonian H_{II} features long-range interactions, which decay with a power-law with exponent α. This exponent and the coupling strength between the spins V are varied in order to investigate the quality of the approximation of the dynamics obtained from the generator calculated by the network. The subsystem of interest contains two contiguous spins located at the centre of a chain of (even) length N.

2. Formulation of the problem

We consider a many-body quantum system partitioned into a subsystem S and the remainder B, which in our setting plays the role of a bath.

The separation implies that the full Hilbert space H is obtained as the tensor product $\mathsf{H} = \mathsf{H}_S \otimes \mathsf{H}_B$ of the Hilbert space of the subsystem (H_S) and of the bath (H_B). Here, we work under the assumption that H is finite-dimensional with dimension m. For instance, for the spin-1/2 models considered later (see figure 3 and equations (15) and (16)) $m = 2^N$, where N is the total number of spins. The quantum state of the full system is described by means of a density matrix $\rho(t)$, which must be positive semi-definite and must have trace equal to one in order to comply with the probabilistic interpretation of quantum mechanics. As such, the space of all possible states of the many-body system is the (convex) subspace of all positive semi-definite unit-trace matrices, $\mathsf{S}(\mathsf{H}) \subset \mathbb{M}(m)$, where $\mathbb{M}(m)$ is the algebra of square matrices of dimensions m. Under the assumptions that the many-body system is subject to a unitary dynamics implemented by the full system Hamiltonian H_{S+B}, its state at time t is given by $\rho(t) = U_t\rho(0)U_t^\dagger$, where $U_t = \mathrm{e}^{-\mathrm{i}tH_{S+B}}$.

The full information about the time-evolution of the degrees of freedom belonging to subsystem S is contained in the reduced density matrix $\rho_S(t)$. As mentioned in the introduction, in certain settings the dynamics of the subsystem state can be approximated through a Markovian open quantum dynamics implemented by a so-called Lindblad generator $\mathcal{L}$ [15, 16], see equation (1). In these cases, $\rho_S(t)$ would effectively obey a quantum master equation

$$\frac{\mathrm{d}\rho_S}{\mathrm{d}t} = \mathcal{L}[\rho_S] = \mathcal{H}[\rho_S] + \mathcal{D}[\rho_S], \tag{2}$$

where we have dropped the explicit time dependence. The map $\mathcal{H}$ accounts for the Hamiltonian contributions to the dynamics while dissipative effects, uniquely associated with the interaction between S and B, are encoded in $\mathcal{D}$. The general form of these terms (for finite-dimensional systems) is

$$\mathcal{H}[\cdot] = -i[H,\cdot], \qquad \mathcal{D}[\cdot] = \frac{1}{2}\sum_{i,j=1}^{d^2-1} c_{ij}\big([F_i,\cdot F_j] + [F_i\cdot, F_j]\big). \tag{3}$$

The operator H is the Hermitian Hamiltonian of the subsystem only and describes its quantum coherent evolution. The operators F_i, with $i = 1, 2, \ldots, d^2$ and d being the dimension of the subsystem Hilbert space, form an orthonormal basis of the subsystem algebra of operators $\mathbb{M}(d)$. Here, without loss of generality, we choose this basis with the property that all its elements are Hermitian $F_i = F_i^\dagger$ and that F_{d^2} is proportional to the identity, $F_{d^2} = \frac{1}{\sqrt{d}}\mathbb{1}$. (Note that this latter term is not included in the double sum appearing in $\mathcal{D}$.) The orthonormality condition thus reads $\mathrm{Tr}(F_iF_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta, showing that all elements F_i but F_{d^2} are traceless. The matrix c, with elements c_{ij} is called the Kossakowski matrix, and it describes dissipative effects on S due to its interaction with B. Such a matrix must be positive semi-definite in order for the dynamical map implemented by $\mathcal{L}$, $\mathcal{T}_t = \mathrm{e}^{\mathcal{L}t}$, to be completely positive [15, 16]; we recall here that a map $\mathcal{T} : \mathbb{M}(d) \to \mathbb{M}(d)$ is said to be completely positive if, for any dimension k, the map $\mathcal{T} \otimes \mathbb{1}_k : \mathbb{M}(d) \otimes \mathbb{M}(k) \to \mathbb{M}(d) \otimes \mathbb{M}(k)$ is also positive, that is, it preserves the semi-positivity of the matrices it acts upon. Since c is semidefinite positive, it can be diagonalized by means of an unitary transformation h: $h^\dagger ch = \mathrm{diag}(\gamma_1, \ldots, \gamma_{d^2-1})$, with non-negative eigenvalues γ_i. The Lindblad generator can then be represented in its 'diagonal' form, via the operators $J_i = \sum_{i=1}^{d^2-1} h_{ij}F_j$ equation (3),

$$\frac{\mathrm{d}\rho_S}{\mathrm{d}t} = -i[H,\rho] + \sum_{i=1}^{d^2-1}\gamma_i\left(J_i\rho J_i^\dagger - \frac{1}{2}\{J_i^\dagger J_i, \rho\}\right), \tag{4}$$

where the operators J_i are called jump operators. The problem we address in this work is that of learning—by means of neural network methods—a time-independent physical dynamical generator $\mathcal{L}$ for the dynamics of a subsystem embedded in a unitarily evolving many-body system (see figure 1). We are moreover interested in investigating in which parameter regimes the dynamics implemented by such a generator satisfactorily reproduces the time-evolution of the subsystem. From a technical perspective, it is convenient to pass from a representation of $\mathcal{L}$ as a map acting on the density matrix ρ_S to that of a matrix acting on a vector representation of ρ_S itself. We therefore discuss in the following how the information contained in ρ_S can be encoded in a vector using the coherence-vector formalism [19], and subsequently derive the ensuing matrix-representation of the Lindblad generator $\mathcal{L}$. We note that while there are several ways of such vectorizing a the density matrix, the one adopted here allows us to rewrite the full quantum dynamical problem in terms of real numbers only (see below), which is convenient for the implementation of neural network algorithms.

Employing the (Hermitian) orthonormal basis of $\mathbb{M}(d)$ introduced above, we can write any density matrix $\rho_S \in \mathbb{M}(d)$ as a linear combination of the F_i. The coherence vector $\mathbf{v} = (v_1, v_2, \ldots, v_{d^2-1}, 1\sqrt{d})$ is then the vector in $\mathbb{R}^{d^2}$ that gathers the corresponding expansion coefficients:

$$\rho_S = \frac{\mathbb{1}}{d} + \sum_{i=1}^{d^2-1} F_i v_i. \tag{5}$$

We note that the trace-normalization condition, $\mathrm{Tr}(\rho_S) = 1$, implies $[\mathbf{v}]_{d^2} = 1/\sqrt{d}$ which has been taken out of the sum. Furthermore, we note that $\mathbf{v}$ is a real vector since both ρ_S and the F_i are Hermitian. Equation (5) shows the one-to-one correspondence between ρ_S and the coherence vector $\mathbf{v}$, which we can exploit to conveniently represent the subsystem's density matrix. To make this idea more concrete, we discuss the example of spin-1/2 particle, whose Hilbert space dimension is $d = 2$. In this case, the basis $\{F_i\}_{i=1}^4$ can be chosen to be proportional to the Pauli matrices $\{\sigma^x/\sqrt{2}, \sigma^y/\sqrt{2}, \sigma^z/\sqrt{2}, \mathbb{1}/\sqrt{2}\}$ with

$$\sigma_x \equiv \sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y \equiv \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z \equiv \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{6}$$

such that the density matrix takes the form

$$\rho_S = \frac{\mathbb{1}}{2} + \frac{\sigma_1 v_1 + \sigma_2 v_2 + \sigma_3 v_3}{\sqrt{2}}. \tag{7}$$

The coherence vector parametrization is then $\mathbf{v} = (v_1, v_2, v_3, 1/\sqrt{2})$, which is reminiscent of the usual Bloch vector (apart for the last unimportant element and the different normalization of the Pauli matrices). To obtain the coherence vector elements from the density matrix one computes the traces

$$v_i = \mathrm{Tr}(F_i \rho_S) \equiv \langle F_i \rangle; \tag{8}$$

as highlighted in the above equation, knowing the coherence vector basically amounts to knowing the expectation value of all possible observables of the subsystem.

We now need to understand how the quantum master equation (2) translates into an evolution equation for the elements of the coherence vector. This can be done by taking a generic density matrix ρ_S as in equation (5), computing the action of the generator on it, $\mathcal{L}[\rho_S]$, and expanding this matrix into the basis formed by the operators F_i. This procedure, detailed in appendix A, provides the matrix representation $\mathbf{L}$ of the map $\mathcal{L}$ which evolves the coherence vector $\mathbf{v}$ through the equation

$$\frac{\mathrm{d}\mathbf{v}(t)}{\mathrm{d}t} = \mathbf{L}\mathbf{v}(t), \qquad \mathbf{L} \equiv \mathbf{H} + \mathbf{D}. \tag{9}$$

Here, analogously to what done for $\mathcal{L}$, we have decomposed the matrix representation of the generator $\mathbf{L}$ into a coherent Hamiltonian part, $\mathbf{H}$, and a dissipative one, $\mathbf{D}$. We note that, since the generator $\mathcal{L}$ is hermiticity-preserving, the matrices $\mathbf{H}$ and $\mathbf{D}$ are real-valued. In particular, as shown in appendix A, the matrix $\mathbf{H}$ is given by

$$\begin{aligned} &\mathbf{H}_{ij} = -4\sum_{k=1}^{d^2-1} f_{ijk}\boldsymbol{\omega}_k, \quad i,j \in \{1,2,\dots,d^2-1\},\\ &\mathbf{H}_{id^2} = \mathbf{H}_{d^2 i} = 0, \quad i \in \{1,2,\dots,d^2\}, \end{aligned} \tag{10}$$

where $\boldsymbol{\omega}$ defines the expansion of the Hamiltonian matrix H in equation (3) over the orthonormal set $\{F_i\}_{i=1}^{d^2}$: $H = \sum_{i=1}^{d^2-1}\boldsymbol{\omega}_i F_i$. The so-called antisymmetric structure constants f_{ijk} are defined as $f_{ijk} = -\frac{i}{4}\ \mathrm{Tr}([F_i, F_j]F_k)$. The dissipative part is instead given by (see detailed derivation in appendix A)

$$\begin{aligned} &\mathbf{D}_{mn} = -8\sum_{i,j,k=1}^{d^2-1}\left(f_{mik}f_{njk}\,\mathrm{Re}(c)_{ij} + f_{mik}d_{njk}\,\mathrm{Im}(c)_{ij}\right), \quad m,n \in \{1,2,\dots,d^2-1\},\\ &\mathbf{D}_{md^2} = -4\sum_{i,j=1}^{d^2-1} f_{imj}\,\mathrm{Im}(c)_{ij}, \qquad \mathbf{D}_{d^2 m} = 0, \quad m \in \{1,2,\dots,d^2\}, \end{aligned} \tag{11}$$

where the so-called symmetric structure constants d_{ijk} are given by $d_{ijk} = \frac{1}{4}\ \mathrm{Tr}(\{F_i, F_j\}F_k)$, and the matrix c is the Kossakowski matrix appearing in equation (3).

As we show below, by exploiting a neural network we can learn the matrix representation $\mathbf{L}$, which propagates the degrees of freedom of a subsystem of a many-body system undergoing unitary quantum dynamics. The learned Lindblad generator $\mathcal{L}$ will by construction be physically valid, i.e. it will implement a completely positive and trace-preserving dynamics. However—while it is always possible to find $\mathbf{L}$—how well the learned subsystem dynamics reproduces the exact one depends on how much the latter can be actually approximated by a Markovian open quantum dynamics.

3. The architecture and the training procedure

In the following we discuss how the matrix $\mathbf{L}$ is learned by a neural network through training it with data obtained from simulating the exact unitary many-body quantum dynamics, see sketch in figure 2. According to equation (9) we have the decomposition $\mathbf{L} = \mathbf{H} + \mathbf{D}$. Here, $\mathbf{H}$ is defined in terms of the $d^2 - 1$ unconstrained real parameters $\boldsymbol{\omega}$ appearing in equation (10). The matrix $\mathbf{D}$ is parametrised by the complex Hermitian matrix c (see equation (11)), which is constrained to be positive semi-definite. To enforce this constraint by construction, we express c as $c = Z^\dagger Z$ for some complex matrix $Z = X + iY$, where X and Y are real matrices. The parameters of our neural network which are to be found via training are thus $\theta = \{\boldsymbol{\omega}, X, Y\}$, and our parametrization automatically ensures the complete positivity of the dynamics generated by $\mathbf{L}$. Note that, since we want to learn a Markovian dynamics, we assume the parameters $\theta = \{\boldsymbol{\omega}, X, Y\}$ to be time independent. Moreover, only the symmetric real matrix $\mathrm{Re}(c) = XX^t + YY^t$ and the skew symmetric real matrix $\mathrm{Im}(c) = X^tY - Y^tX$ appear in equation (11) (see also equations (A.9) and (A.10)), showing that the network indeed only makes use of real parameters.

Given an input coherence vector at time t, $\mathbf{v}_{\mathrm{in}} = \mathbf{v}(t)$, the network $\mathbf{M}$ (see figure 2), which is a function of parameters θ, is trained to output $\mathbf{v}_{\mathrm{out}} = \mathbf{M}[\theta]\mathbf{v}_{\mathrm{in}}$ such that $\mathbf{v}_{\mathrm{out}}$ is close to $\mathbf{v}(t + \mathrm{d}t)$, that is the coherence vector after a discrete time step of length $\mathrm{d}t$. In this way, the network learns how to propagate the coherence vector by an infinitesimal time-step and this information fully specifies the matrix $\mathbf{L}$ (cf equation (9)).

Training data consist of a set of trajectories of the time-evolution of $\mathbf{v}$, starting from an initial state $\mathbf{v}_0$ up to a fixed time T.[5] Each trajectory is thus a list of snapshots $(\mathbf{v}_0, \mathbf{v}(\mathrm{d}t), \ldots, \mathbf{v}(T))$. These trajectories are obtained by exactly evolving the full many-body quantum state ρ under the action of the global Hamiltonian H_{S+B}, tracing out the degrees of freedom of the bath at each time step, and computing $\mathbf{v}$ as in equation (8). For each choice of parameters in the Hamiltonian, n trajectories were used as training data D. The network parameters θ are optimised through the Adam optimization algorithm[6] [22] to minimize the common mean squared error cost function

$$l(\theta) = \mathbb{E}_{\mathbf{v}(t),\mathbf{v}(t+\mathrm{d}t)\sim D}[\|\mathbf{M}[\theta]\mathbf{v}(t) - \mathbf{v}(t + \mathrm{d}t)\|^2]. \tag{12}$$

Here, the expectation $\mathbb{E}$ is taken over the training dataset D.

To summarise, the parameters of the network are $\theta = \{\boldsymbol{\omega}, X, Y\}$, where $\boldsymbol{\omega}$ is a real vector defining $\mathbf{H}(\boldsymbol{\omega})$ and X, Y are real matrices defining $\mathbf{D}(X, Y)$. In particular, $\theta = \{\boldsymbol{\omega}, X, Y\}$ provides the 'weights' of the network. One needs first to use the matrices X and Y to compute the semidefinite positive matrix $c = (X - iY)^t(X + iY)$. After a choice of the basis for the Hilbert space of the subsystem has been made, one can compute the structure constants and use them to map the variational parameters to the matrices $\mathbf{H}$ and $\mathbf{D}$ via equations (10) and (11). Then $\mathbf{H}(\boldsymbol{\omega})$ and $\mathbf{D}(X, Y)$ are applied to $\mathbf{v}_{\mathrm{in}}$ as

$$\mathbf{M}[\theta]\mathbf{v}_{\mathrm{in}} \equiv \mathrm{e}^{\mathrm{d}t(\mathbf{H}(\boldsymbol{\omega})+\mathbf{D}(X,Y))}\mathbf{v}_{\mathrm{in}} \tag{13}$$

and the variational parameters are optimized by minimizing the loss function in equation (12). Here $\mathrm{d}t$ is the time step used in the exact integration of the dynamics upon which the network is trained. In general, in order to learn all the relevant dynamical features of the subsystems one should make sure that $\mathrm{d}t$ is sufficiently small, so that all timescales are captured. We have considered $\mathrm{d}t = 0.01/\Omega$, as we observed that with such a time step the details of the exact evolution were well reproduced. Since creating the artificial dataset though exact diagonalization was the most time consuming task. Therefore, it not efficient to use a too small $\mathrm{d}t$. Once the network has been trained, it is possible to retrieve the Hamiltonian part $\mathbf{H}$ and the Kossakowski matrix c, we give a concrete example of this in appendix C, where we also give a brief explanation of their forms.

4. Many-body models and subsystem of interest

For testing our ideas, we use the above-discussed neural network architecture to learn the generator of the reduced dynamics for two neighbouring spins embedded in a one-dimensional chain. The spin chain is thus partitioned into two parts as shown in figure 3. The first one is formed by two nearest-neighbouring sites which for convenience are labelled as spin 1 and spin 2. This part acts as the 'subsystem of interest'. The other part is the remainder of the lattice and is regarded as the bath. We assume that a Hamiltonian H_{S+B} governs the dynamics of the full system quantum state $\rho(t)$. Training data is computed by evolving this state according to the unitary evolution operator $U_t = \mathrm{e}^{-\mathrm{i}tH_{S+B}}$ and tracing out the degrees of freedom of the bath. As initial state, $\rho_0 = \rho(t = 0)$, of the unitary many-body dynamics we consider product states of the form

$$\rho_0 = \rho_S(0) \otimes \rho_B(0). \tag{14}$$

The reduced density matrix at time t is given by $\rho_S(t) = \mathrm{Tr}_B\, U_t\rho_0 U_t^\dagger$.

We consider two different quantum spin chain models, as shown in figure 3, and refer to them as model I and II. Model I has closed boundaries, i.e. the spins form a ring, while model II has open boundaries. For model I, the full subsystem–bath dynamics is governed by a nearest-neighbour interaction Hamiltonian

$$H_{\mathrm{I}} = \frac{\Omega}{2}\sum_{i=1}^{N}\sigma_i^x + V\sum_{i=3}^{N-1}n_i n_{i+1} + V'(n_L n_1 + n_1 n_2 + n_2 n_3). \tag{15}$$

[5] More specifically, the training data is formed by 50 exactly evolved trajectories, each with the initial conditions as discussed in subsection 4.1. The batch size is 256 with 512 batches per epoch, over 20 epochs. Of the generated data, 80% is used as training set, while 20% is employed as validation, to check that the model does not over fit.

[6] Hyperparameters used in Adam were learning rate Alpha $= 10^{-3}$, Beta$_1 = 0.9$, Beta$_2 = 0.999$ and Epsilon $= 10^{-8}$.

The operator $n_i = \frac{1+\sigma_i^z}{2}$ denotes the projector onto the spin up state of the ith spin. The constants V and V' are the strength of the interaction among neighbouring spins: V is associated with interactions among the bath spins, while V' is associated with interactions between the subsystem sites and between the subsystem and the bath sites next to it (see also figure 3(a)). The contribution proportional to Ω, which is the same for all sites, parameterises a transverse 'laser' field and drives Rabi oscillations between single spin states. The motivation behind studying the Hamiltonian H_I is that the first and the second spin define the subsystem and their interaction with the bath is assumed to be controllable.

Model II, which features open boundary conditions, is governed by a Hamiltonian that features power-law interactions:

$$H_\mathrm{II} = \frac{\Omega}{2}\sum_{i=1}^{N}\sigma_i^x + V\sum_{i<j}\frac{n_i n_j}{|i-j|^\alpha}, \tag{16}$$

where $\alpha \geqslant 0$ defines the power with which the interactions decay over distance. Note, that unlike in model I, there is no specific choice for the coupling constant among bath and system spins. In appendix C we adopt Hamiltonian II to give an example of the form of the retrieved Hamiltonian part and Kossakowski matrix of the Lindblad generator. There, we deem the restriction of equation (16) to a system of two spins as $H_\mathrm{II}^{(2)}$, and undertake a brief study of how the bath affects the form of this Hamiltonian. Hamiltonians (15) and (16) are variants of the Ising model with transverse and longitudinal field as well as to the so-called PXP model [23, 24]. Experimentally they can be realized, for example, with Rydberg atoms [25–29].

Given that we focus on a system whose reduced density matrix is that of two spins, the dimension d of the reduced Hilbert space H_S is four, and a natural choice for the basis $\{F_i\}_{i=1}^{d^2}$ is given by the two-spin Pauli group generators $\{\sigma^x/\sqrt{2}, \sigma^y/\sqrt{2}, \sigma^z/\sqrt{2}, \mathbb{1}_2/\sqrt{2}\}^{\otimes 2}$. Hence, $\{F_1, F_2, \dots, F_{d^2}\} = \{\mathbb{1}_2 \otimes \sigma^x/2, \mathbb{1}_2 \otimes \sigma^y/2, \dots, \mathbb{1}_4/2\}$.If N is the length of the spin chain, the subsystem of interest is identified by the spins at positions 1 and 2, such that each element in $\mathbf{v}$ corresponds to either a two-body expectation value $\langle\sigma_m^1\sigma_n^2\rangle$, or to a single-body expectation value $\langle\sigma_m^1\rangle$, $(\langle\sigma_m^2\rangle)$, with $m, n \in \{1, 2, 3\}$, and

$$\sigma_l^k = \mathbb{1}_2^{\otimes k-1} \otimes \sigma_l \otimes \mathbb{1}_2^{\otimes(N-k)}. \tag{17}$$

4.1. The initial conditions

For both models, the initial conditions of the system are chosen as the product of a thermal state for the bath $\rho_\mathrm{Bath} \propto \mathrm{e}^{-\beta H_{\mathrm{I/II}}}$, with β the inverse temperature, and a valid density matrix ρ_S for the system S. Model II was studied to investigate whether and how the algorithm is able to capture the dynamics in the presence of longer-ranged interactions. To be able to focus on this aspect, we decided to take a fixed initial state for the bath (without changing the temperature) and, for simplicity, we considered the infinite-temperature ones ($\beta = 0$). For the study of both models we choose the initial density matrix ρ_S in the following way: we consider two random $d \times d$ real matrices M and N whose entries are taken from a Gaussian distribution centred at zero and with standard deviation one and we construct the density matrix according to

$$\rho_S(0) = \frac{(M+iN)^\dagger(M+iN)}{\mathrm{Tr}\left[(M+iN)^\dagger(M+iN)\right]}. \tag{18}$$

This ensures that $\rho_S(0)$ is a Hermitian positive semi-definite unit-trace matrix.

5. Results and discussion

In the standard treatment of open quantum systems, the dynamics of the reduced state of a subsystem can be approximated by means of Markovian open quantum dynamics only when certain conditions are met [14] (see e.g. reference [30] and references therein for a different derivation of a Markovian quantum master equations). These include, for instance, a weak subsystem–bath coupling, an infinitely large bath with a continuous dispersion relation and a large separation of time-scales between the subsystem and the bath dynamics.

For the models I and II (see equations (15) and (16)), not all of the above conditions are met. For instance, while it is possible to tune the parameters in a way that the interaction between S and B is weak, the bath B will always be, in our setting, a finite-dimensional object whose Hamiltonian possesses a discrete spectrum. Nonetheless, given that our network is capable of retrieving a time-independent generator for the subsystem from training data, it is natural to ask whether the dynamics implemented by such a generator can provide an approximate description of the subsystem state also in the considered settings.

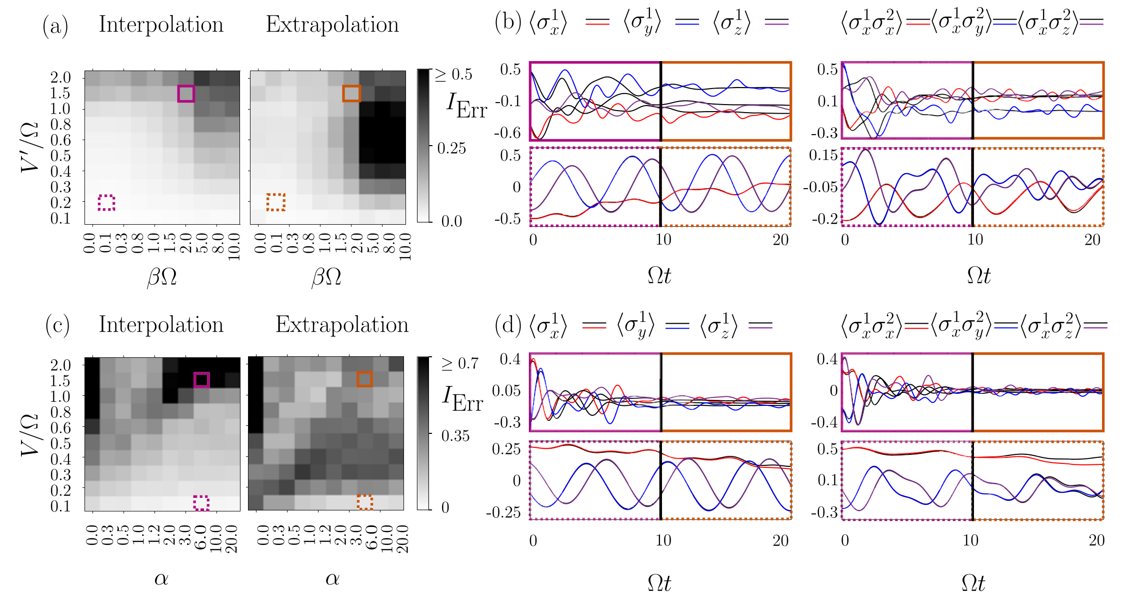

Figure 4. Interpolation and extrapolation error. (a) Scan of the error, I_{Err}, defined by equation (19) for different values of the parameters β and V' for model I. The considered spin-chain length is $N = 11$. The interpolation error I_{Err} is computed from $T_{\mathrm{in}} = 0$ to $T_{\mathrm{fin}} = 10/\Omega$, where T_{fin} is the time up to which the network is trained on the exact trajectories. The extrapolation error I_{Err} refers to the region in time t where the network is used to extrapolate from $T_{\mathrm{in}} = 10/\Omega$ up to unseen times $t \leqslant T_{\mathrm{fin}}$. We choose $T_{\mathrm{fin}} = 20/\Omega$. The discrete time step used in the calculations is $\mathrm{d}t = 0.01/\Omega$. (b) Examples of how the network model performs, for model I, in predicting the time-evolution of expectation values of quantum observables. We study both single-site (left panels) and two-site (right panels) spin observables. The top panels show a case taken from the parameter region characterised by large errors I_{Err}, while the bottom ones show a case taken from a region in which the error is small (see squares in panel (a)). (c) Scan of the error for different values of the parameters α and V for model II. The considered spin-chain length is $N = 10$. (d) Same as in panel (b) but for model II. In panels (b) and (d), the black lines are predictions from the network model, and coloured lines are showing exact simulation results obtained from the dynamics of the full many-body system state.

To determine whether, and for which parameter range, a Markovian dynamical description is a valid one for the subsystem and, thus, whether the network can be used to predict the dynamical behaviour of its observables, we investigate the accuracy of the Markovian approximation obtained through the network by varying the parameters (β, V') and (α, V) for model I and II, respectively. To quantify the error made in the approximation, we define the error measure

$$I_{\mathrm{Err}}(T_{\mathrm{in}}, T_{\mathrm{fin}}) = \frac{1}{T_{\mathrm{fin}} - T_{\mathrm{in}}} \int_{T_{\mathrm{in}}}^{T_{\mathrm{fin}}} \mathrm{d}t \|\rho_{\mathrm{exact}}(t) - \rho_{\mathrm{network}}(t)\|_1. \tag{19}$$

Here, ρ_{exact} is the (time-evolved) quantum state for S obtained from the exact diagonalization of the full many-body problem, while ρ_{network} is the subsystem dynamics as predicted by the network. The trace norm $\|\sigma\|_1$ for a $d \times d$ complex matrix σ, with eigenvalues $\lambda_1, \ldots, \lambda_d$ is given by

$$\|\sigma\|_1 = \mathrm{Tr}\,\sqrt{\sigma^\dagger\sigma} = \sum_{i=1}^{d} |\lambda_i|. \tag{20}$$

In order to remove the dependence of the error from the specific initial condition considered, we consider the quantity I_{Err}, defined in equation (19), averaged over 10 different trajectories, each with an independent initial condition (18) (shown in figures 4 and 5).

5.1. Subsystem dynamics

We start our discussion by considering model I, see equation (15). As apparent from figure 4(a), for relatively low values of the coupling between the subsystem and the bath, the generator learned by the neural network can provide a good approximation of the reduced subsystem dynamics. This is also manifest from the bottom panels in figure 4(b); the time-evolution of the expectation values of subsystem observables, as predicted by the network, is in very good agreement with exact numerical data, both for single-site observables and for two-site ones. On the other hand, figure 4(a) clearly shows that the neural network approximation becomes less and less reliable upon increasing the coupling between the subsystem and the bath. We stress that this is not witnessing an issue occurring in the training of the neural network for these parameter regimes. The reason for this increase of the error is that, consistently with what

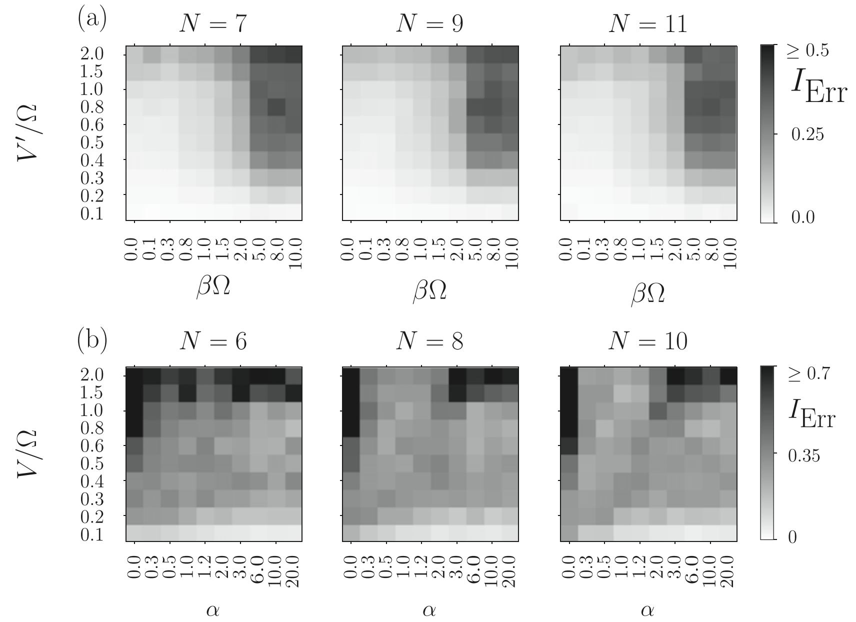

Figure 5. Scaling of the error with the bath size N for model I (a) and model II (b). (a) Scans of the error I_{Err} for different values of the parameters β, V' for model I, produced for different choices of the chain length $N = 7, 9, 11$. (b) Error I_{Err} for model II, for different choices of the parameters V, α and different chain lengths $N = 6, 8, 10$. For both models, the error is computed from time zero up to time $T_{\mathrm{fin}} = 20/\Omega$. Here, in contrast to figures 4(a)–(c), no distinction is made between extrapolation and interpolation region. For model II the error is the highest for the smallest of the system sizes investigated, as one may expect. No dependence on system size is discernible instead for model I.

expected form the weak-coupling approximation, for large enough coupling the reduced subsystem dynamics becomes non-Markovian and cannot be approximated by the time-independent generator learned by the network. We also note that a similar worsening of the Markovian approximation of the dynamics can be observed when lowering the temperature of the bath (increasing β), as becomes transparent from figure 4(a).

As a first application of the learned generator of the subsystem dynamics, one can use it to extrapolate the dynamics of the subsystem observables to times which have not been seen during the training procedure. This is in particular true for the range of parameters for which the error in approximating the dynamics with the one retrieved by the neural network is relatively small, as shown in figure 4(b). The rationale is that whenever the subsystem dynamics is Markovian, the dynamical generator is time-independent and, once this is learned, it can also be used outside the training time-window.

For model II we find that when the interaction between sites decays fast enough ($\alpha \geqslant 1$) and when the coupling is low ($V/\Omega \approx 0.1$), the error I_{Err} remains relatively small (see figure 4(c)). In this regime, just as before, the learned generator can be used also for making predictions beyond training times, as shown in figure 4(d). On the other hand, when the power α decreases, we observe an increase in the approximation error (see figure 4(c)). This is due to the fact that the concomitant increase of the range of the interactions amounts to increase the interaction between S and B. It is thus reasonable to expect that non-Markovian effects become more pronounced.

In the following we investigate the behaviour of the error, for both models, when changing the length of the spin chain, i.e. modifying the size of the bath, see figure 5. For model I, we observe that the error remains very similar for the different system sizes explored. This may suggest that, in the case of nearest-neighbour interactions and for the model at hand, the considered sizes can be already considered sufficiently large for the remainder of the many-body system to act as a proper bath. For model II, instead, there appears to be an improvement in the approximation upon increasing the size of the bath. This may be related to the fact that for long-range interactions, revivals in the subsystem dynamics—due to the back-flow of information from the bath into the subsystem—have stronger effects for smaller system sizes.

For both models, our observations suggest that a Markovian approximation may be indeed justified, also for a bath formed by discrete degrees of freedom, provided that the latter is sufficiently large. Moreover, we observe that for all parameter regimes, even in those in which the Markovian approximation is not quite satisfactory, the error tends to stay bounded. This means that the predicted long-time value of local observables seems to overall capture the actual behaviour of the subsystem, at least within the considered time-window (cf figures 4(b)–(d)). This seems to suggest that even for large errors in the interpolation, one

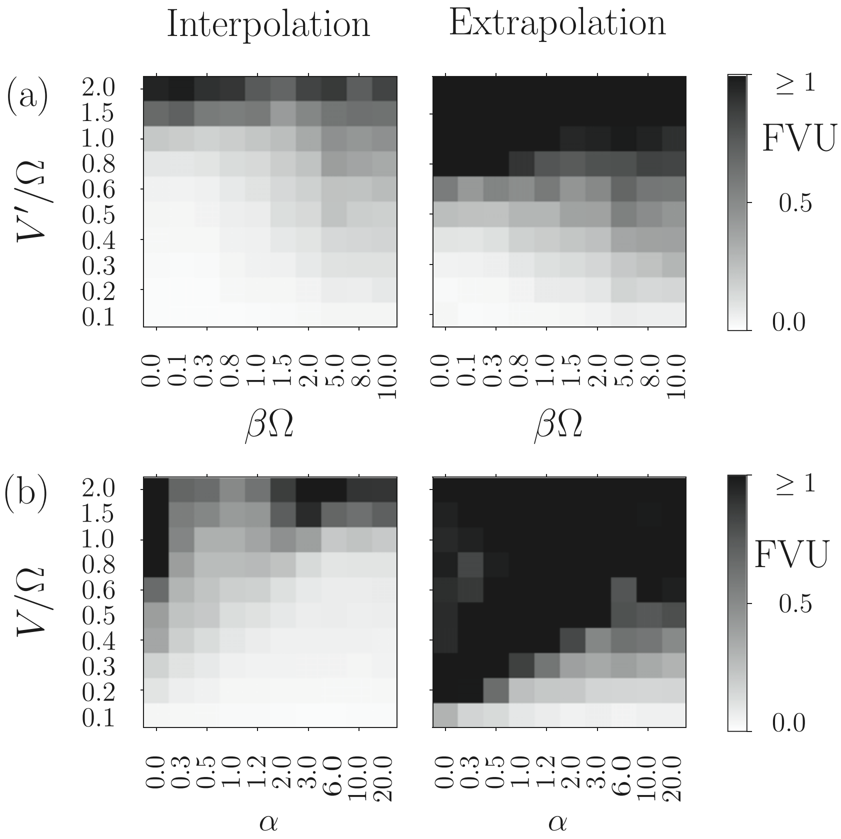

Figure 6. Fraction of variance unexplained (FVU) (a) FVU of the interpolation and extrapolation error for model I and a system size $N = 11$ as a function of the system-bath interaction V'/Ω and the inverse temperature $\beta\Omega$. (b) FVU of the interpolation and extrapolation error for model II as function of the interactions strength V/Ω and the power-law exponent α of the interaction potential.

may find actually a relatively small error in the extrapolation, see figure 4. This effect is seen in several parameter regimes, where the expectation values of subsystem observables is almost constant, but features small amplitude oscillations in the extrapolation window, as shown in figure 4.

In order to have a fairer comparison of the error between initial transient regime with large oscillations and the long-time one in which oscillations have very small amplitude, we introduce a relative error measure. We consider the so-called fraction of variance unexplained, or fraction of variance unexplained (FVU). We compute this quantity between the coherence vector predicted by the network v_i^{network} and the one retrieved from the exact simulation v_i^{exact}, as in the following equation

$$\text{FVU} = \frac{1}{d^2 - 1}\sum_{i}^{d^2-1}\sqrt{\frac{\text{Var}(v_i^{\text{exact}} - v_i^{\text{network}})}{\text{Var}(v_i^{\text{exact}})}}. \tag{21}$$

The set over which the variance Var is computed is given by the time snapshots with discrete time step dt, $v_i = (v_i(t_0), v_i(t_0 + \mathrm{d}t), \dots)$. The fraction of variance unexplained normalises an error to the variance of the signal (here the exact coherence vector). The results for model I and II are plotted in figure 6 for system sizes $N = 11$ and $N = 10$, respectively, while results for smaller system sizes are reported in appendix B. Comparing figure 6 with figure 5, we note that considering the FVU provides a smoother behaviour of the error. In particular, it allows to distinguish between the regions for which the error in the approximation I_{Err} is small simply because the overall signal has small variance, from those where the error is small because the neural network dynamics correctly captures the behaviour of the subsystem. The latter situation occurs for small values of the coupling strength in both models I and II and for fast decaying interactions in model II (cf figure 6). In this region the Markovian approximation is correct, and the network is able to reconstruct the dynamics almost exactly.

5.2. Stationary behaviour

Once the network has learned the matrix-representation **L** of the Lindblad generator, it is also possible to investigate stationary properties of the system [9, 31]. In our case, we can do this by studying the stationary-state coherence vector, which is nothing but the eigenvector $\mathbf{v}_{\text{st}}$ associated with the zero

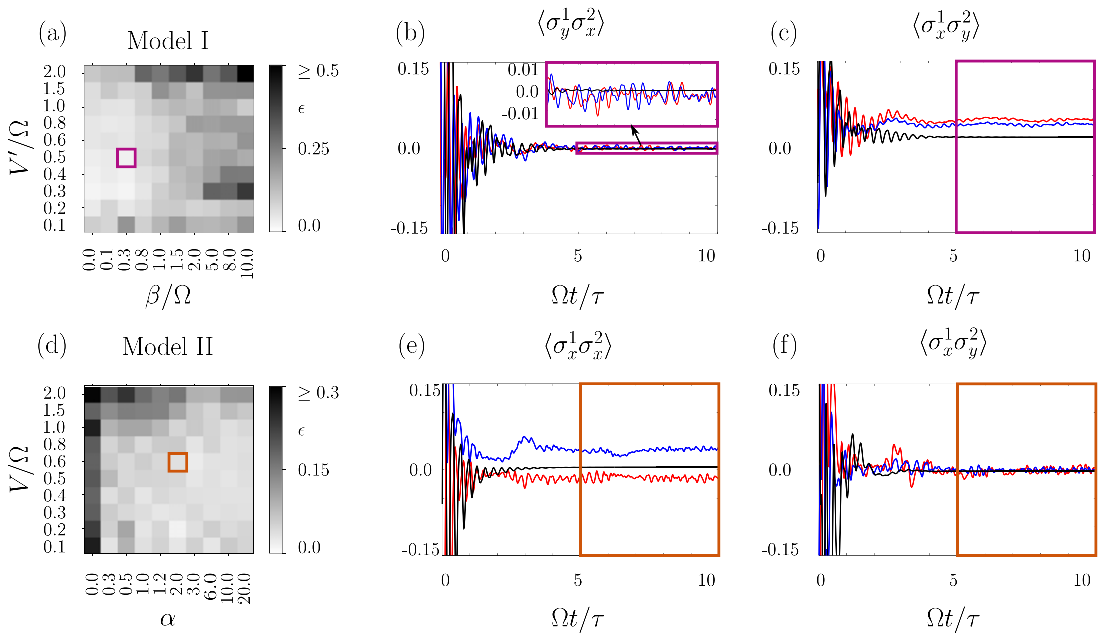

Figure 7. Stationary state dynamics. In panels (a) and (d) the error ϵ, defined by equation (22), is depicted for model I with $N = 11$ and model II with $N = 10$, respectively. The error is computed for different parameter combinations (β, V') for model I and (V, α) for model II. For specific parameter choices the dynamics of some two-spin observables is shown in panels (b) and (c) for model I and panels (e) and (f) for model II. Red and blue lines indicate the same expectation value for two different initial conditions for the quantum state ρ_{exact}. The black line indicates the evolution according to the learned $\mathbf{L}$. The regions in time where the error ϵ is computed are marked by boxes. As the behaviour of single body observable does not differ much from the two-body expectation values, we did not report it.

eigenvalue of $\mathbf{L}$: $\mathbf{L}\mathbf{v}_{\text{st}} = 0 = \frac{\mathrm{d}\mathbf{v}_{\text{st}}}{\mathrm{d}t}$. Such a vector provides the stationary density matrix of the subsystem S, see equation (23), which in our case is by construction ensured to describe a physically consistent quantum state. In principle, there is no reason to expect that the long time coherence vector $\mathbf{v}_{\text{exact}}(t)$ will converge to $\mathbf{v}_{\text{st}}$. This is because (i) we are training the network for a finite time-window and (ii) the bath is finite, and thus one would expect to observe, for long times, recurrence and revivals in the time-evolution of system observables. These effects are associated with a re-entering of the information scrambled from the subsystem into the bath into the subsystem again, and are associated with non-Markovian behaviour. Nonetheless, what we observe (shown in figure 7), is that in some parameter regimes the agreement between $\mathbf{v}_{\text{st}}$ and the long-time behaviour of the exact dynamics is remarkably good.

To be more quantitative, let us define E_{gap} as the smallest, in modulus, among the real parts of the non-zero eigenvalues of $\mathbf{L}$ and let τ be $1/E_{\text{gap}}$. This latter quantity represents the time-scale of the approach to stationarity in a Markovian open quantum system. Moreover, in order to circumvent the issue that the exact subsystem dynamics always presents oscillations (also due to the finiteness of the bath), we define the following error measure

$$\epsilon = \|\rho^*_{\text{exact}} - \rho_{\text{st}}\|_1, \tag{22}$$

where

$$\rho_{\text{st}} = \frac{\mathbb{1}}{d} + \sum_{i=1}^{d^2-1} [\mathbf{v}_{\text{st}}]_i F_i, \qquad \rho^*_{\text{exact}} = \frac{1}{(b-a)\tau}\int_{a\tau}^{b\tau} \mathrm{d}t\rho_{\text{exact}}(t). \tag{23}$$

Thus, the error ϵ measures the distance between the stationary behaviour predicted from the network results and an averaged long-time behaviour of the exact solution. In particular, in order to be sure to address the stationary behaviour, at least in relation to what predicted by the network, we choose $a = 5$, $b = 10$. Within this time window—provided that the exact dynamics is correctly captured by the time-independent generator—the expectation value of local observables should have already converged to their stationary value. Given the finiteness of the bath, these will—as mentioned above—display residual oscillations around such an average stationary value which would be described indeed by ρ^*_{exact}. The discrete time step $\mathrm{d}t$ for the exact integration and for the training of the network is chosen to be $\mathrm{d}t = 0.01/\Omega$ as before. Once the training was complete, we could retrieve the Lindblad generator $\mathbf{L}$ and its eigenvalues, and in particular E_{gap} and the time scale $\tau = E_{\text{gap}}$. In figure 7 the error in equation (23) is shown. It is computed by averaging over 10 different initial conditions for ρ_{exact}. This error is rather different both from the error of equation (19) and more importantly from the FVU of equation (21). The FVU is indeed used to compare the oscillations of the predicted and of the exact dynamics. However, by construction, the predicted dynamics is bound to reach a stationary state, and the FVU will thus tend to be large. This signals that the learned dynamics is not able to reproduce the small oscillations around the stationary values. Nonetheless, we can investigate how well the learned dynamics can capture the average value of this oscillations, which should provide their stationary value. To this end, a more appropriate error measure is the one given in equation (22). One notices that while there are observables that approach the learned steady state $\mathbf{v}_{\mathrm{st}}$, e.g. $\sigma_y^1\sigma_x^2$ in model I, others do not. In certain cases, the exact time-evolution of the expectation of observables converges towards a long-time behaviour which depends on the initial conditions. This may be related to the fact that we are considering a finite bath. On the other hand, the learned dynamics always approaches the same stationary behaviour since, due to the existence of a finite gap E_{gap}, the steady state of the Lindblad generator is unique.

6. Conclusions

We have introduced a simple neural network whose parameters can be exactly mapped onto those of a Lindblad generator. Importantly, such a generator which is learned by the network from exact dynamical data is automatically ensured to be a physically consistent generator of a quantum Markovian dynamics. We have investigated the applicability of such an architecture to two different classes of spin models. Even though the considered physical settings are rather different from those known to give rise to Markovian subsystem dynamics, we find that, in certain parameter regimes, the network model provides a faithful approximation of the subsystem time-evolution.

Future developments in the same spirit may include the adaptation to architectures capable of encoding time correlations in time series, such as long short term memory (LSTM) [32] or transformers [33], which would allow for the learning of a time-dependent generator. A different path to achieve time dependence would be that of learning the analytical solutions of the differential equations of motion [34], or by numerically solving them by means of neural networks [35].

We have exploited the time-independent generator learned by the network in order to investigate stationary properties of the reduced subsystem state. This idea looks promising as a path towards capturing relevant long-time features such as thermalization effects [17, 36]. Finally, instead of learning the linear evolution of the density matrix, one may think of directly learning the evolution of an order parameter, such as the magnetization or particle density [37]. This would entail machine learning of an effectively non-linear dynamical evolution within the state space of the order parameter. This directly leads to the question whether and how more involved neural network architectures permit an increased performance in determining effective generators and possibly allow an improved quantification of time dependence and non-Markovianity.

The presented results open routes towards the understanding of complex non-equilibrium dynamics through a reduced number of (collective) degrees of freedom. This may ultimately allow to develop simplified descriptions of complex dynamical non-equilibrium processes which is not only of interest in fundamental research but may also be of importance when harnessing many-body phenomena in technological applications.

Acknowledgments

The authors thank P Mazza and M Klopotek for valuable discussions. We acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-Number 2064/1—Project Number 390727645. IL acknowledges funding from the 'Wissenschaftler R'éuckkehrprogramm GSO/CZS' of the Carl-Zeiss-Stiftung and the German Scholars Organization e.V., and through the DFG Project Number 449905436. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting DZ.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Representation of the matrix L

In this appendix we derive equations (10) and (11) for the dynamics of the coherence vector $\mathbf{v} = (v_1, \dots, v_{d^2})$. The derivative of v_h with respect to t can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} v_h(t) = \mathrm{Tr}\left(F_h \frac{\mathrm{d}}{\mathrm{d}t}[\rho_S(t)] \right) = \mathrm{Tr}(F_h \mathcal{L}[\rho_S(t)]) = \mathrm{Tr}(\mathcal{L}^*[F_h]\rho_S(t)), \tag{A.1}$$

where in the first equality we used equation (8) and in the second one (2). The map $\mathcal{L}^*$ indicates the dual map of $\mathcal{L}$, i.e. the one which evolves observables and not the subsystem state. The action of the superoperator $\mathcal{L}^*$ can be obtained from that of $\mathcal{L}$ via the cyclic property of the trace. Expanding $\mathcal{L}^*[F_h] = \sum_{k=1}^{d^2} \mathrm{Tr}(F_k \mathcal{L}^*[F_h]) F_k$, one finds

$$\frac{\mathrm{d}}{\mathrm{d}t} v_h(t) = \sum_{k=1}^{d^2} \mathrm{Tr}(\mathcal{L}^*[F_h]F_k)\, \mathrm{Tr}(F_k \rho_S(t)) = [\mathbf{L}\mathbf{v}(t)]_h, \tag{A.2}$$

where we the matrix $\mathbf{L}$ is defined as $\mathbf{L}_{hk} = \mathrm{Tr}(\mathcal{L}^*[F_h]F_k)$. Explicitly, we obtain

$$\mathrm{Tr}(\mathcal{L}^*[F_h]F_k) = \sum_{h=1}^{d^2} \mathrm{Tr}\left(i[H, F_h]F_k + \frac{1}{2} \sum_{i,j=1}^{d^2-1} c_{ij} (F_j^\dagger [F_h, F_i] F_k + [F_j^\dagger, F_h] F_i F_k) \right). \tag{A.3}$$

The comparison with equation (3) yields

$$\mathbf{H}_{kh} = i\, \mathrm{Tr}([H, F_k]F_h), \qquad \mathbf{D}_{kh} = \frac{1}{2} \sum_{i,j=1}^{d^2-1} c_{ij}\, \mathrm{Tr}\Big(F_j^\dagger [F_k, F_i] F_h + [F_j^\dagger, F_k] F_i F_h \Big). \tag{A.4}$$

We can now introduce the structure constants d_{ijk} and f_{ijk} for the basis $\{F_i\}_{i=1}^{d^2}$. The structure constants characterise the commutation and anti-commutation relations of $\{F_i\}_{0 \leqslant i < d^2}$ as

$$\{F_i, F_j\} = 2\delta_{ij} \frac{\mathbb{1}}{d} + \frac{1}{4} \sum_{k=1}^{d^2-1} d_{ijk} F_k, \qquad [F_i, F_j] = \frac{-i}{4} \sum_{k=1}^{d^2-1} f_{ijk} F_k. \tag{A.5}$$

Since $\mathrm{Tr}(F_i F_j) = \delta_{ij}$, one obtains

$$d_{ijk} = \frac{1}{4}\, \mathrm{Tr}(\{F_i, F_j\} F_k), \qquad f_{ijk} = -\frac{i}{4}\, \mathrm{Tr}([F_i, F_j] F_k). \tag{A.6}$$

As a consequence, d_{ijk} is fully anti-symmetric under exchange of two indices, while f_{ijk} is fully symmetric. Moreover, for self-adjoint $F_i = F_i^\dagger$, the structure constants are real.

For the Hamiltonian part, we consider the expansion $H = \sum_{i=1}^{d^2-1} F_i \omega_i$, with $\boldsymbol{\omega} = (\omega_1, \dots, \omega_{d^2-1})$ a $d^2 - 1$ dimensional real vector. Then, the matrix elements of $\mathbf{H}$ are determined by

$$\mathbf{H}_{mn} = i \sum_{k=1}^{d^2-1} \mathrm{Tr}([F_k, F_m] F_n) \boldsymbol{\omega}_k. \tag{A.7}$$

The Hamiltonian part can thus be rewritten as

$$\mathbf{H}_{ij} = -4 \sum_{k=1}^{d^2-1} f_{ijk} \boldsymbol{\omega}_k, \qquad \mathbf{H}_{id^2} = \mathbf{H}_{d^2 i} = 0, \tag{A.8}$$

with $i, j \in \{1, 2, \dots, d^2 - 1\}$. Thus, the Hamiltonian part is skew-symmetric and can be parametrised by a single vector $\boldsymbol{\omega}$.

The dissipative part instead takes the form

$$\begin{aligned} \mathbf{D}_{mn} &= \frac{1}{2} \sum_{i,j}^{d^2-1} c_{ij}\, \mathrm{Tr}\big([F_m, F_i] F_n F_j + [F_j, F_m] F_i F_n \big) \\ &= -8 \sum_{i,j,k=1}^{d^2-1} \left(f_{mik} f_{njk}\, \mathrm{Re}(c)_{ij} + f_{mik} d_{njk}\, \mathrm{Im}(c)_{ij} \right) \end{aligned} \tag{A.9}$$

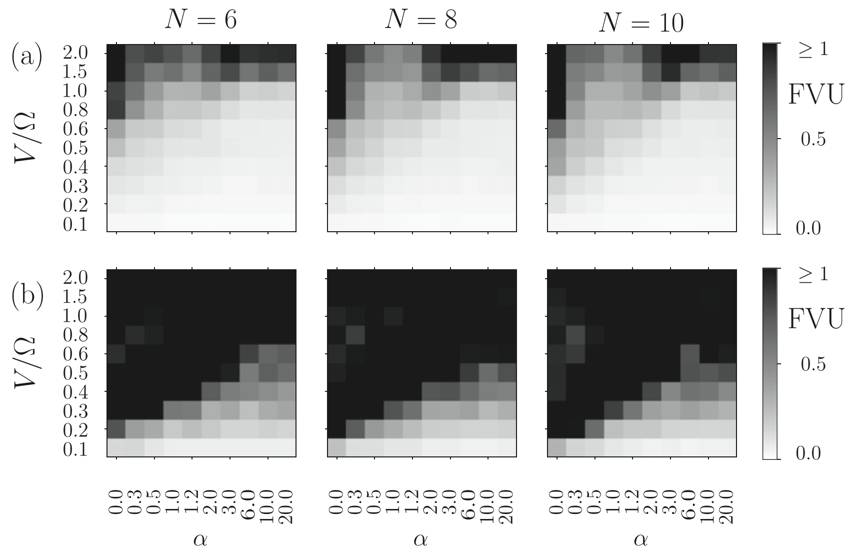

Figure B1. FVU in various spin chain lengths for model I. In the upper row (a) is reported the interpolation and in (b) the extrapolation region.

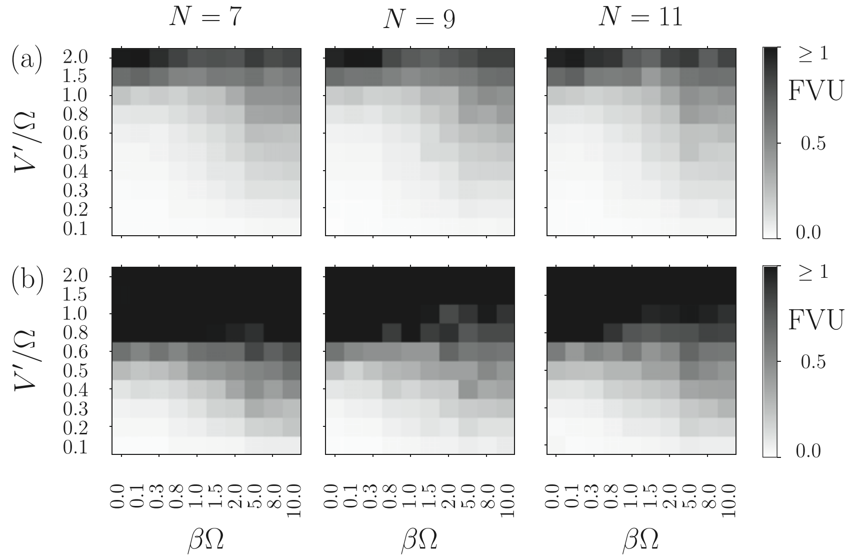

Figure B2. FVU in various spin chain lengths for model II. In rows (a) and (b) the FVU is shown in the interpolation and extrapolation regime respectively.

for $1 \leqslant m < d^2, 1 \leqslant n < d^2$. In the last line we used the hermiticity of the Kossakowski matrix c, i.e. that its real (imaginary) part is (skew-) symmetric. For the matrix elements of $\mathbf{D}$ with $1 \leqslant m < d^2$, $n = d^2$ one has

$$\mathbf{D}_{md^2} = \frac{1}{2}\sum_{i,j,k=1}^{d^2-1} c_{ij}\ \mathrm{Tr}\big(F_i[F_m, F_j] + [F_i, F_m]F_j\big) = -4\sum_{i,j=1}^{d^2-1} f_{imj}\,\mathrm{Im}(c)_{ij}, \tag{A.10}$$

where we used the cyclic property of the trace in the second equality, and the fact that f_{imj} and $\mathrm{Re}(c)_{ij}$ are antisymmetric in the indices (ij), while $\mathrm{Re}(c)_{ij}$ is symmetric. Finally, for $1 \leqslant n < d^2$, one obtains $\mathbf{D}_{d^2n} = 0$.

Appendix B. The fraction of variance unexplained (FVU)

In this appendix we report additional results on the FVU for different system sizes N. In figure B1 we show results for model I both in the interpolation (see panel figure B1(a)) and in the extrapolation (see panel figure B1(b)) regimes. In figure B2 we report analogous results for model II.

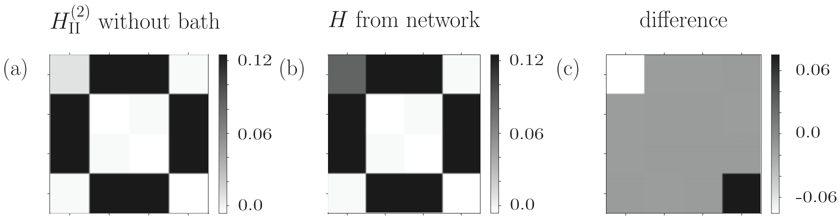

Figure C1. Two-spin Hamiltonian with and without bath. (a) Matrix elements of the Hamiltonian II for a system of two spins $H_{\mathrm{II}}^{(2)}$ with parameters $V'/\Omega = 0.1$ and $\alpha = 0.3$. In order to compare this with the learned Hamiltonian, we have removed the irrelevant contribution proportional to the identity, i.e. we consider $H_{\mathrm{II}}^{(2)} \to H_{\mathrm{II}}^{(2)} - \mathrm{Tr}\left(H_{\mathrm{II}}^{(2)}\right)$. (b) Matrix elements of the Hamiltonian H learned by the generator for same parameters as in (a) and a system of six spins. (c) Difference between the exact Hamiltonian in (a) and the learned one in (b), which shows the emergence of a term proportional to $\sigma_z \otimes \mathbb{1} + \mathbb{1} \otimes \sigma_z$.

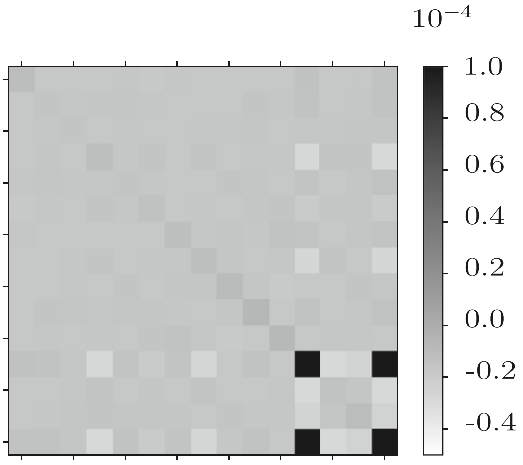

Figure C2. Real part of the Kossakowski matrix. Real part of the Kossakowski matrix retrieved from a six spin system evolved according to Hamiltonian II with parameters $V'/\Omega = 0.1$ and $\alpha = 0.3$. This shows that the largest values are associated with combination of operators $\sigma_z \otimes \mathbb{1}$ and $\mathbb{1} \otimes \sigma_z$ with equal weights. This means that the relevant dissipative effect is a dephasing noise, associated with a jump operator $J \propto \sigma_z \otimes \mathbb{1} + \mathbb{1} \otimes \sigma_z$. Note that, in this case, the entries of the Kossakowski matrix are small. This is due to the fact that we are considering small values of V'/Ω and, in this regime, dissipation is expected to be of second-order in V'/Ω.

Appendix C. Interpretability of the network

We here give an example of how one can extract information about the subsystem dynamics from the network and interpret it as the Hamiltonian and dissipative part of the Lindblad equation. In particular we focus here on model II for the parameters $V^{'}/\Omega = 0.1$ and $\alpha = 0.3$. In figure C1(a) we plot the matrix elements of the Hamiltonian of model II for a chain of only two spins, already discussed below equation (16). In figure C1(b) is plotted the one retrieved from the network, that is, the Hamiltonian part H of the Lindblad generator. Their difference is given in figure C1(c). As shown, this difference is proportional to a term $\mathbb{1} \otimes \sigma_z + \sigma_z \otimes \mathbb{1}$.

This contribution can be explained by means of a 'mean-field' treatment of the Hamiltonian involving the density–density interactions between the bath and the subsystem. Indeed, it results from taking terms like $\langle n_i \rangle n_j$, where i is a site of the bath and $\langle n_i \rangle$ stands for expectation value, while j a site of the two-spin subsystems, once terms proportional to the identity are removed.

For the same system it is possible to also retrieve the Kossakowski matrix. We plot the real part of its entries in figure C2. The basis of choice for the Hilbert space of subsystem is the same as in the main text, and it is given by $\{F_1, F_2, \ldots, F_{d^2}\} = \{\mathbb{1}_2 \otimes \sigma^x/2, \mathbb{1}_2 \otimes \sigma^y/2, \ldots, \mathbb{1}_4/2\}$. This shows that the most relevant entries are associated with dephasing implemented by a 'collective' jump operator (cf equation (4)) of the form $J \propto \sigma_z \otimes \mathbb{1} + \mathbb{1} \otimes \sigma_z$.

ORCID iDs

Francesco Carnazza https://orcid.org/0000-0002-1458-8701
Igor Lesanovsky https://orcid.org/0000-0001-9660-9467

References

[1] Carleo G and Troyer M 2017 *Science* **355** 602–6

[2] Mehta P, Bukov M, Wang C-H, Day A G R, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1–124

[3] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002

[4] Carrasquilla J and Melko R 2017 *Nat. Phys.* **13** 431

[5] Greco A, Starostin V, Karapanagiotis C, Hinderhofer A, Gerlach A, Pithan L, Liehr S, Schreiber F and Kowarik S 2019 *J. Appl. Crystallogr.* **52** 1342–7

[6] Greco A, Starostin V, Hinderhofer A, Gerlach A, Skoda M W A, Kowarik S and Schreiber F 2021 *Mach. Learn.: Sci. Technol.* **2** 045003

[7] Deng D-L, Li X and Das Sarma S 2017 *Phys. Rev.* B **96** 195145

[8] Valenti A, van Nieuwenburg E, Huber S and Greplova E 2019 *Phys. Rev. Res.* **1** 033092

[9] Hartmann M J and Carleo G 2019 *Phys. Rev. Lett.* **122** 250502

[10] Nagy A and Savona V 2019 *Phys. Rev. Lett.* **122** 250501

[11] Reh M, Schmitt M and Gärttner M 2021 *Phys. Rev. Lett.* **127** 230501

[12] Yoshioka N and Hamazaki R 2019 *Phys. Rev.* B **99** 214306

[13] Mazza P P, Zietlow D, Carollo F, Andergassen S, Martius G and Lesanovsky I 2021 *Phys. Rev. Res.* **3** 023084

[14] Breuer H P and Petruccione F 2002 *The Theory of Open Quantum Systems* (Oxford: Oxford University Press)

[15] Gorini V, Kossakowski A and Sudarshan E C G 1976 *J. Math. Phys.* **17** 821–5

[16] Lindblad G 1976 *Commun. Math. Phys.* **48** 119–30

[17] D'Alessio L, Kafri Y, Polkovnikov A and Rigol M 2016 *Adv. Phys.* **65** 239–362

[18] Deen S M, Kabir P K and Karl G 1971 *Phys. Rev.* D **4** 1662–6

[19] Byrd M S and Khaneja N 2003 *Phys. Rev.* A **68** 062322

[20] Kimura G 2003 *Phys. Lett.* A **314** 339–49

[21] Paszke A *et al* 2019 *Proceedings of the 33rd Int. Conf. on Neural Information Processing Systems* ed H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox and R Garnett (Red Hook, NY: Curran Associates Inc.)

[22] Kingma D P and Ba J 2017 Adam: A method for stochastic optimization *Computing Research Repository* (arXiv:1412.6980)

[23] Sun B and Robicheaux F 2008 *New J. Phys.* **10** 045032

[24] Ates C, Garrahan J P and Lesanovsky I 2012 *Phys. Rev. Lett.* **108** 110603

[25] Bloch I, Dalibard J and Nascimbène S 2012 *Nat. Phys.* **8** 267

[26] Kim H, Park Y, Kim K, Sim H-S and Ahn J 2018 *Phys. Rev. Lett.* **120** 180502

[27] Ebadi S *et al* 2021 *Nature* **595** 227–32

[28] Browaeys A and Lahaye T 2020 *Nat. Phys.* **16** 132–42

[29] Bloch I, Dalibard J and Zwerger W 2008 *Rev. Mod. Phys.* **80** 885–964

[30] Ciccarello F 2017 *Quantum Meas. Quantum Metrol.* **4** 53

[31] Vicentini F, Biella A, Regnault N and Ciuti C 2019 *Phys. Rev. Lett.* **122** 250503

[32] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735–80

[33] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 arXiv:1706.03762

[34] Sahoo S S, Lampert C H and Martius G 2018 Learning equations for extrapolation and control *Proc. 35th Int. Conf. on Machine Learning, ICML (PMLR)* (Stockholm, Sweden) vol 80 pp 4442–50

[35] Chen R T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* vol 31 ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Red Hook, NY, USA: Curran Associates, Inc.) pp 6571–83

[36] Polkovnikov A, Sengupta K, Silva A and Vengalattore M 2011 *Rev. Mod. Phys.* **83** 863–83

[37] Kharkov Y, Shtanko O, Seif A, Bienias P, Regemortel M V, Hafezi M and Gorshkov A V 2021 arXiv:2111.02385

Second publication

Letter

Inferring interpretable dynamical generators of local quantum observables from projective measurements through machine learning

Giovanni Cemin,[1,*] Francesco Carnazza,[1] Sabine Andergassen,[2] Georg Martius,[3,4]
Federico Carollo,[1] and Igor Lesanovsky[1,5]

[1] *Institut für Theoretische Physik, Universität Tübingen, Auf der Morgenstelle 14, Tübingen 72076, Germany*
[2] *Institute for Solid State Physics and Institute of Information Systems Engineering, Vienna University of Technology, Vienna 1040, Austria*
[3] *Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, Tübingen 72076, Germany*
[4] *Wilhelm Schickard Institut für Informatik, Maria-von-Linden-Straße 6, Tübingen 72076*
[5] *School of Physics and Astronomy and Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, The University of Nottingham, Nottingham NG7 2RD, United Kingdom*

(Received 21 June 2023; revised 22 November 2023; accepted 29 February 2024; published 3 April 2024)

To characterize the dynamical behavior of many-body quantum systems, one is usually interested in the evolution of so-called order parameters rather than in characterizing the full quantum state. In many situations, these quantities coincide with the expectation value of local observables, such as the magnetization or the particle density. In experiment, however, these expectation values can only be obtained with a finite degree of accuracy due to the effects of the projection noise. Here, we utilize a machine-learning approach to infer the dynamical generator governing the evolution of local observables in a many-body system from noisy data. To benchmark our method, we consider a variant of the quantum Ising model and generate synthetic experimental data, containing the results of N projective measurements at M sampling points in time, using the time-evolving block-decimation algorithm. As we show, across a wide range of parameters the dynamical generator of local observables can be approximated by a Markovian quantum master equation. Our method is not only useful for extracting effective dynamical generators from many-body systems but may also be applied for inferring decoherence mechanisms of quantum simulation and computing platforms.

DOI: 10.1103/PhysRevApplied.21.L041001

a. Introduction. Reconstructing Hamiltonian operators or dynamical generators from physical properties of a quantum system is a problem of current interest. For instance, inverse methods can be applied to identify quantum Hamiltonians associated with a given ground state [1] and interacting many-body theories can be obtained from the knowledge of correlation functions [2,3]. In many settings, one is merely interested in reconstructing effective equations of motion for a subsystem S embedded in a larger "environment" E—as it happens, for open quantum systems (OQSs) [4–8]. Furthermore, in the framework of quantum simulation [9–17], it is important to understand the (effective) dynamical equations under which artificial quantum systems actually evolve and by how much these differ from the desired ones [18,19]. This is relevant for improving state-of-the-art hardware [20,21] and the identification of noise models. Another interesting instance concerns the evolution of order parameters, often constructed from local observables, such as the particle density or the magnetization.

Machine-learning (ML) approaches appear to be particularly suited for this task [22]. Quantum process tomography with generative adversarial methods [23], neural networks [24], and recurrent neural networks [25,26] have been developed. These approaches are promising but have two main drawbacks: they require a great number of measurements and they treat ML algorithms as black boxes, thus lacking in physical interpretation. Simpler methods are capable of learning Hamiltonians from fewer local measurements [27–33], yet they typically rely on a *a priori* ansatz for the functional form of the Hamiltonian or of the dissipation. A more general approach is to fit OQS dynamics by learning the Nakajima-Zwanzig equation [34,35] through transfer-tensor techniques [36–38] or by learning convolutionless master equations [39]. However, these approaches require a full state tomography at different time steps, which is prohibitive to achieve in experiments. Ultimately, current methods thus either rely on an *ad hoc* ansatz or demand data that are not experimentally accessible or that are lacking in physical interpretability (which is actually becoming highly desirable [40]).

In this work, we use ML methods to infer the effective dynamical generator of a subsystem from a finite

[*] Corresponding author. gcemin@pks.mpg.de

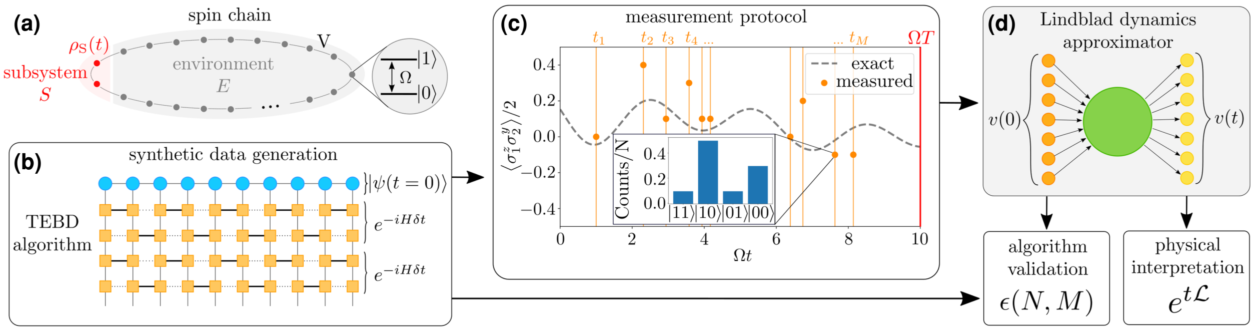

FIG. 1. The system and the measurement protocol. (a) The system is composed of the two-spin *subsystem* S, which is embedded in the *environment* E consisting of $L-2$ spins. The Hamiltonian describing the system [cf. Eq. (1)] features a driving field with Rabi frequency Ω and nearest-neighbor (NN) interactions with coupling strength V. (b) Synthetic data generation using the time-evolving block-decimation algorithm. The time-evolution operator is approximated using Trotterization: $U_t = e^{-iHt} \approx \prod e^{-iH\delta t}$. (c) A sketch of the measurement protocol. The gray dashed line represents the *exact* dynamics of the observable. For each trajectory, M random points are selected in the time window $[0, \Omega T]$ (here, $\Omega T = 10$). For each point in time, we measure N times in a suitable basis; here, the z (y) basis for the first (second) spin. The inset shows an example histogram of measurement outcomes: $|10\rangle$ means that the measurement on the first spin along z basis has outcome 1 and the measurement on the second spin along the y basis has outcome 0. From the histogram, the expectation values (depicted as orange dots) are obtained. (d) The synthetic data are fed to the ML algorithm, namely, a single-layer perceptron that learns the dynamical generator $\mathcal{L}$. The algorithm is validated by comparison to the exact data, where the error $\epsilon(N, M)$ is calculated for different values of N and M. The ML method is interpretable and can be used to "read out" the underlying dynamical processes.

set of local measurements at randomly selected times, which inevitably produce noisy data due to projection noise. To illustrate our approach, we consider a many-body spin system [cf. Fig. 1(a)], which is ubiquitous in the context of experiments with trapped-ion or Rydberg-atom quantum simulators [21,41,42]. By using synthetic (experimental) data generated with tensor-network-based algorithms, we infer a physically consistent Markovian dynamical generator [43,44] governing the evolution of a small subsystem. Our method, which works reliably across a wide range of parameters—even, in some instances, outside the weak-coupling limit—yields interpretable results that may be used to infer noise models on quantum simulators or to study thermalization dynamics in many-body systems.

b. Setting. The system that we consider is a one-dimensional (1D) quantum spin chain consisting of L spins arranged on a circular lattice, as depicted in Fig. 1(a). The chain is partitioned into a *subsystem* S, here formed by two adjacent spins, and the *environment* E, i.e., the remainder of the spin chain. We assume that whole system evolves unitarily, through the many-body Hamiltonian

$$H_{S+E} = \frac{\Omega}{2}\sum_{i=1}^{L}\sigma_i^x + V\left(\sum_{i=1}^{L-1} n_i n_{i+1} + n_L n_1\right). \qquad (1)$$

The first term describes a transverse "laser" field, while the second one accounts for nearest-neighbor (NN) interactions. Here, σ_i^α denotes the α Pauli matrix for the i^{th} spin and we have defined the projector $n = (1 + \sigma^z)/2$. The above Hamiltonian is of practical interest for experiments with Rydberg atoms [42] and essentially encodes an Ising model in the presence of transverse and longitudinal fields. We simulate the time evolution of the whole system by means of the time-evolving block-decimation (TEBD) algorithm [see Fig. 1(b)].

In our setting, the information on the state of S is obtained by a finite number, N, of projective measurements, taken at randomly selected times $t_1, \ldots t_M$ [see Fig. 1(c)]. From this *noisy* data, we want to infer the open quantum dynamics of the reduced state $\rho_S(t)$ of subsystem S. Formally, these dynamics are obtained as the partial trace of the evolution of the full many-body state, i.e., $\rho_S(t) = \mathrm{Tr}_E\left(U_t \rho_{S+E} U_t^\dagger\right)$, where $U_t = e^{-iHt}$, ρ_{S+E} is the initial state of the system, and Tr_E denotes the trace over the environment. In general, such dynamics are rather involved and may show non-Markovian effects or they may be nonlinear for generic initial states ρ_{S+E} [45]. Here, we restrict ourselves to learning Markovian dynamics for $\rho_S(t)$ but more general approaches are possible [43]. The goal is then to identify the time-independent generator $\mathcal{L}$, yielding the Markovian quantum master-equation evolution [45–47]

$$\dot{\rho}_S(t) = \mathcal{L}[\rho_S(t)], \qquad (2)$$

which *optimally* describes the dynamics of S. This simple form has the advantage that it is interpretable, i.e., it allows us to read off the Hamiltonian and decoherence processes (see further below).

c. Data generation. We simulate the time evolution of the system for times $t \in [0, \Omega T]$ by means of the TEBD algorithm [48–50] [see Fig. 1(b)], which allows us to study systems of up to 50 spins. We generate 30 trajectories obtained by initializing the system in state $\psi = \bigotimes_{k=1}^{L} |0\rangle$, with $\sigma^z |0\rangle = -|0\rangle$ and perturbing the subsystem S through a random two-spin unitary $\hat{U}_{\text{rand}}$ distributed with the Haar measure. As the system evolves in time, at intervals of $\Omega dt = 0.01$, we compute the expectation value of the 15 independent observables $\{\mathbb{1}_1 \otimes \sigma_2^x, \mathbb{1}_1 \otimes \sigma_2^y, \dots, \sigma_1^x \otimes \sigma_2^y, \dots, \sigma_1^z \otimes \sigma_2^z\}/2$ of the subsystem S, which uniquely identify the reduced subsystem state. We then emulate experimental measurements of these local observables. For each trajectory, we select M random times in the time window $[0, \Omega T]$ and, for each of these points in time, we perform N measurements in all the relevant bases in order to produce a noisy estimate of the reduced state. Such a procedure is sketched in Fig. 1(c), where the histogram depicts the counting of $N = 10$ measurement outcomes for a single time point, whereas the orange dots represent the experimental expectation values.

d. Ml Architecture and training. The generator $\mathcal{L}$ defining the quantum master Eq. (2) can be parametrized as $\mathcal{L} = \mathcal{H} + \mathcal{D}$, where

$$\mathcal{H}[\cdot] = -i[H, \cdot], \quad \text{with} \quad H = \sum_{i=2}^{d^2} \theta_i^H F_i,$$
$$\mathcal{D}[\cdot] = \frac{1}{2} \sum_{i,j=2}^{d^2} c_{ij} \left([F_i, \cdot F_j^\dagger] + [F_i \cdot, F_j^\dagger]\right). \tag{3}$$

Here, we have introduced the Hermitian orthonormal basis $\{F_i\}_{i=1}^{16}$ for the operators of the subsystem S, where we choose F_1 to be proportional to the identity operator, $F_1 = \mathbb{1}/\sqrt{d}$. Moreover, we write the Hamiltonian in terms of the "fields" θ_i^H and the dissipative contribution $\mathcal{D}$ in a nondiagonal form, fully specified by the so-called Kossakowski matrix c_{ij}. The latter must be positive semidefinite in order for the open quantum dynamics to be completely positive. This constraint can be "hard coded" by setting $c = Z^\dagger Z$, for a complex matrix $Z = \theta^X + i\theta^Y$, with θ^X and θ^Y being real valued.

We decompose the reduced state ρ_S in the basis $\{F_i\}_{i=1}^{16}$ as [51]

$$\rho_S = \frac{\mathbb{1}}{d} + \sum_{i=2}^{16} F_i v_i, \tag{4}$$

which defines the *coherence vector* $v_i = \text{Tr}(F_i \rho_S)$. Note that the condition $\text{Tr}(\rho_S) = 1$ implies $v_1 = 1/\sqrt{d}$, which we take outside the sum. The coherence-vector representation is quite convenient, from a numerical point of view, as it allows us to write the action of the generator on states as the action of the matrix $\mathbf{L}$ on coherence vectors. The quantum master equation [Eq. (2)] becomes

$$\frac{d\mathbf{v}(t)}{dt} = \mathbf{L}\mathbf{v}(t) = (\mathbf{H} + \mathbf{D})\mathbf{v}(t), \tag{5}$$

where $\mathbf{H}_{ij} = -\text{Tr}(\mathcal{H}[F_i]F_j)$ and $\mathbf{D}_{ij} = \text{Tr}(\mathcal{D}[F_i]F_j)$ are real-valued matrices. As explicitly shown in the Supplemental Material [52], the matrix $\mathbf{H} = \mathbf{H}(\theta_i^H)$ depends linearly on the parameters θ_i^H, while the matrix $\mathbf{D} = \mathbf{D}(\theta_{ij}^X, \theta_{ij}^Y)$ depends quadratically on θ_{ij}^X and θ_{ij}^Y.

We build a simple neural network [53], here called the *Lindblad dynamics approximator* (LDA), as

$$\mathbf{M}(\theta, t) = e^{t[\mathbf{H}(\theta_i^H) + \mathbf{D}(\theta_{ij}^X, \theta_{ij}^Y)]}, \tag{6}$$

which is the structure of the Lindblad time propagator.

We train the LDA to learn the Lindblad representation $\mathbf{L}$ from (synthetic) experimental data. In the training procedure, we feed the LDA with the initial conditions $\mathbf{v}_{\text{in}} = \mathbf{v}(0)$ and the time of the measurement t and optimize the parameters $\theta = \{\theta_i^H, \theta_{ij}^X, \theta_{ij}^Y\}$ such that $\mathbf{v}_{\text{out}} \simeq \mathbf{v}(t) = \mathbf{M}(\theta, t)\mathbf{v}_{\text{in}}$ [54]. Training over a finite time t is crucial when working with experimental data. Indeed, training the LDA to propagate the coherence vector only over an infinitesimal time step dt [43,44], i.e., $\mathbf{v}_{\text{in}} = \mathbf{v}(t)$ and $\mathbf{v}_{\text{out}} = \mathbf{v}(t + dt)$, is bound to fail as soon as the noise is larger than the variation of the coherence vector (for further details, see the Supplemental Material [52]).

To test the correctness of the learned generator, we produce r new exact trajectories and compare them with the LDA prediction. A quantitative measure of the performance of the ML algorithm is given by the following error function:

$$\epsilon(N, M) := \frac{1}{r} \sum_{i=1}^{r} \frac{1}{T} \int_0^T \frac{\|\rho_{\text{ML}}^i(t) - \rho_S^i(t)\|_2^2}{\|\rho_S^i(t)\|_2^2} \text{d}t, \tag{7}$$

where $\rho_{\text{ML}}^i(t)$ is the prediction for the state of the subsystem obtained from our ML algorithm, $\rho_S^i(t)$ represents the synthetic data for a given choice of N and M, and $\|O\|_2^2 = \text{Tr}\left(O^\dagger O\right)$.

e. Benchmarking the algorithm. Before training the algorithm on data for the many-body model in Eq. (1), we benchmark their ability to learn a Lindblad generator within a well-controlled setting. We consider a two-spin Lindblad generator, which in its diagonal form is specified by the following Hamiltonian and jump operators [cf. Fig. 2(a)]:

$$H = \frac{\Omega}{2}(\sigma_1^x + \sigma_2^x) + V n_1 n_2,$$
$$J_1 = \sqrt{\gamma}\sigma_1^-, \qquad J_2 = \sqrt{\gamma}\sigma_2^-,$$
$$J_3 = \sqrt{\kappa} n_1, \qquad J_4 = \sqrt{\kappa} n_2. \tag{8}$$

The jump operators effectively describe the effects of the environment on the subsystem: J_1, J_2 encode decay from $|1\rangle$ to $|0\rangle$ while J_3, J_4 encode dephasing. The data-generation procedure follows the protocol described above, with two exceptions: the initial conditions are given by a random density matrix ρ_{rand} and the data for testing have a doubled time window $[0, 2\Omega T]$. In this way, we can also test the ability of the algorithm to extrapolate to unseen times. Since the network is, in principle, able to perfectly learn a Markovian Lindblad generator, we expect the extrapolation to be accurate. The results are reported in Figs. 2(b) and 2(c). The color map [Fig. 2(b)] shows the error, $\epsilon(N, M)$, averaged over $r = 10$ test trajectories, for 100 different combinations of N and M. First, we observe the overall trend of the error to decrease as $N \times M$ increases. However, the plot is not symmetric with respect to the line $M = N$. In fact, for higher values of M (and fixed $N \times M$), the LDA has a more stable training, and hence a better performance. This is because the choice of the M time points is random and a small value of M has a high probability of yielding data that are not representative of the dynamics. On the other hand, higher values of M yield more representative data of the trajectory, despite relatively small values of N. For high values of N and M, as expected, the error becomes small. The ML algorithm can thus successfully learn a Lindbladian, with a precision that approximately depends on the product $N \times M$.

e. Many-body setting. Having established the capability of the LDA to *exactly* learn Lindblad generators from experimental data, we now address the many-body scenario described by Eq. (1). In this case, the reduced dynamics of $\rho_S(t)$ can feature non-Markovian effects. The LDA will thus, by construction, learn the "optimal" Markovian description of the system. Whether or not this description will accurately describe the subsystem depends on the relevance of non-Markovian effects.

In Fig. 3(a), we show results for weak interactions $V = 0.1\Omega$. Here, the ML algorithm learns a dynamical generator that faithfully reproduces the time evolution of the coherence vector. This suggests that the subsystem dynamics is, in this weak-coupling regime, essentially Markovian. Notably, the Lindblad generator can be inferred even when N and M are small. As already observed during the benchmarking, the training procedure is less stable for the models in the bottom-right corner of Fig. 3(a), compared to the top left, confirming that larger M values are better than larger N values at fixed $N \times M$.

In Fig. 3(b), we show results for strong interactions, $V = 2\Omega$. Quite surprisingly, also in this case, the LDA learns an effective Lindbladian that reproduces the subsystem dynamics very well. Also here, the latter has a Markovian character. A possible explanation for this is that strong interactions lead to a faster decay of time correlations in the environment, which thus renders the subsystem dynamics Markovian. Due to the faster oscillations in this regime, more sampling points in time are needed than the weakly interacting case [Fig. 3(a)], for a same accuracy. In both cases, for small $N \times M$, while the model cannot recover the exact dynamics, it nonetheless provides an average over the fast oscillations [cf. Figs. 3(a) and 3(b)]. The data for the whole coherence vector are reported in the Supplemental Material [52], where we also show additional results for the case $V = 0.5\Omega$. In the latter case, the error is higher due to non-Markovian effects, which appear to be non-negligible [55] in this intermediate regime.

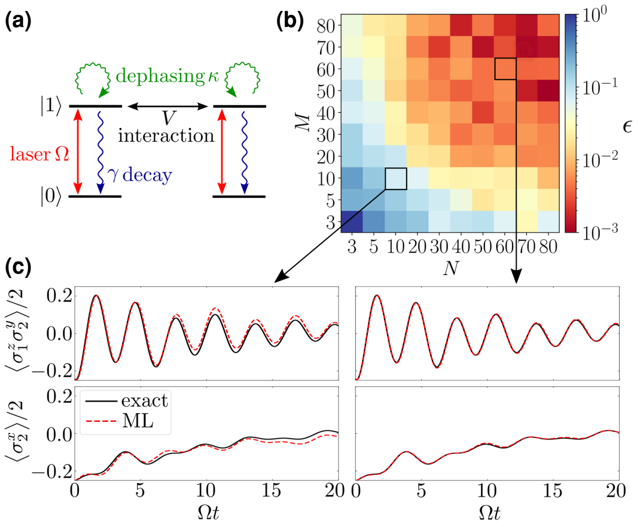

FIG. 2. Benchmarking. (a) The OQS used to benchmark the LDA, described in terms of a Hamiltonian and jump operators [cf. Eq. (8)]. The Hamiltonian consists of a driving (Ω) and an interaction (V). The jump operators describe decay (with rate γ) and dephasing (with rate κ). (b) The error $\epsilon(N, M)$, as in Eq. (7), plotted for different values of N and M. The error is averaged over $r = 10$ out-of-sample trajectories with a doubled time window $[0, 2\Omega T]$ (see text for details). (c) Dynamical curves for two selected observables and different combinations of N and M. From the plot, one can appreciate how the predicted dynamics are already quite reliable for smaller values of N and M. The parameters are $V = 0.5\Omega$, $\gamma = 0.01\Omega$, and $\kappa = 0.05\Omega$.

Conclusions. We have presented a simple ML algorithm able to learn a physically consistent and interpretable dynamical generator starting from (synthetic) experimental data. We have shown that it can yield faithful results for both weak and strong interactions. A wider range of systems could be included by relaxing the assumed Markovianity of the learned generator, namely, by allowing it to be time dependent ($\mathcal{L}(t)$). Some approaches already exist (see, e.g., Refs. [36,39]) but they require an enormous amount of data and lack physical interpretability.

In this work, we have restricted the study to 1D systems with NN interactions. However, our ML method yields physically consistent generators regardless of the

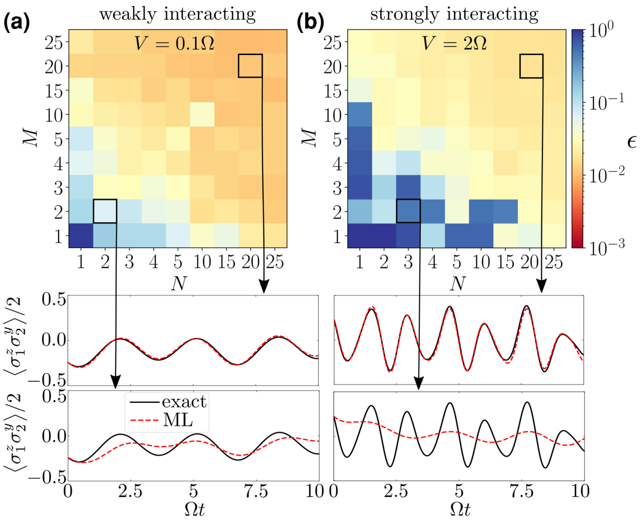

FIG. 3. Learning the subsystem dynamics. The error $\epsilon(N, M)$ as in Eq. (7) for different values of N and M, for two different combinations of the parameters. The error is calculated over $r = 10$ trajectories with $t \in [0, \Omega T]$. (a) The data for the case $V = 0.1\Omega$. Here, the dynamics are slow and the ML algorithm learns an effective $\mathcal{L}$ already for small values of N and M. (b) The data for $V = 2\Omega$. In this case, two different regimes can be distinguished neatly: either the model does not learn the dynamics (blue squares) or it learns it (yellow squares). In both cases, the errors are due to the non-Markovianity assumed for the learned model. The plots beneath the color maps report the time evolution of an observable for the two different models and different sets of (N, M).

dimensionality and the range of the interactions, to which the model is completely agnostic. Moreover, it is interpretable; hence it can be used to "read out" the underlying dynamical processes (see the Supplemental Material [52]). In fact, the learned matrix $\mathbf{L}$ gives direct access to the parameters $\theta_i^H, \theta_{ij}^X, \theta_{ij}^Y$, which represent the Hamiltonian and the jump operators of the subsystem S. In the future, it would be interesting to understand whether feeding the ML algorithm with a sampling of the full coherence vector is necessary or whether *bona fide* dynamics can still be learned when leaving out information about certain observables.

The code for the generation of the artificial data and the training of the ML algorithm is available via GitHub [56].

Acknowledgments. We acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-Number 2064/1—Project No. 39072 7645, under Project No. 449905436 and through the Research Unit FOR 5413/1 (Grant No. 465199066). This project has also received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101046968 (BRISQ). F.C. is indebted to the Baden-Württemberg Stiftung for the financial support of this research project by the Eliteprogramme for Postdocs.

[1] E. Chertkov and B. K. Clark, Computational inverse method for constructing spaces of quantum models from wave functions, Phys. Rev. X **8**, 031029 (2018).

[2] T. V. Zache, T. Schweigler, S. Erne, J. Schmiedmayer, and J. Berges, Extracting the field theory description of a quantum many-body system from experimental data, Phys. Rev. X **10**, 011020 (2020).

[3] C. Merger, A. René, K. Fischer, P. Bouss, S. Nestler, D. Dahmen, C. Honerkamp, and M. Helias, Learning interacting theories from data, Phys. Rev. X **13**, 041033 (2023).

[4] B.-H. Liu, L. Li, Y.-F. Huang, C.-F. Li, G.-C. Guo, E.-M. Laine, H.-P. Breuer, and J. Piilo, Experimental control of the transition from Markovian to non-Markovian dynamics of open quantum systems, Nat. Phys. **7**, 931 (2011).

[5] C. Navarrete-Benlloch, I. de Vega, D. Porras, and J. I. Cirac, Simulating quantum-optical phenomena with cold atoms in optical lattices, New J. Phys. **13**, 023024 (2011).

[6] J. Ma, Z. Sun, X. Wang, and F. Nori, Entanglement dynamics of two qubits in a common bath, Phys. Rev. A **85**, 062323 (2012).

[7] B. Everest, I. Lesanovsky, J. P. Garrahan, and E. Levi, Role of interactions in a dissipative many-body localized system, Phys. Rev. B **95**, 024310 (2017).

[8] U. Hoeppe, C. Wolff, J. Küchenmeister, J. Niegemann, M. Drescher, H. Benner, and K. Busch, Direct observation of non-Markovian radiation dynamics in 3D bulk photonic crystals, Phys. Rev. Lett. **108**, 043603 (2012).

[9] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

[10] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, A Rydberg quantum simulator, Nat. Phys. **6**, 382 (2010).

[11] A. Friedenauer, H. Schmitz, J. T. Glueckert, D. Porras, and T. Schätz, Simulating a quantum magnet with trapped ions, Nat. Phys. **4**, 757 (2008).

[12] J. I. Cirac and P. Zoller, Goals and opportunities in quantum simulation, Nat. Phys. **8**, 264 (2012).

[13] R. Blatt and C. F. Roos, Quantum simulations with trapped ions, Nat. Phys. **8**, 277 (2012).

[14] R. Babbush, W. J. Huggins, D. W. Berry, S. F. Ung, A. Zhao, D. R. Reichman, H. Neven, A. D. Baczewski, and J. Lee, Quantum simulation of exact electron dynamics can be more efficient than classical mean-field methods, Nat. Commun. **14**, 4058 (2023).

[15] S. Jin, N. Liu, X. Li, and Y. Yu, Quantum simulation for quantum dynamics with artificial boundary conditions, arXiv:2304.00667.

[16] X. Zhang, E. Kim, D. K. Mark, S. Choi, and O. Painter, A superconducting quantum simulator based on a photonic-bandgap metamaterial, Science **379**, 278 (2023).

[17] F. Schäfer, T. Fukuhara, S. Sugawa, Y. Takasu, and Y. Takahashi, Tools for quantum simulation with ultracold atoms in optical lattices, Nat. Rev. Phys. **2**, 411 (2020).

[18] J. Carrasco, A. Elben, C. Kokail, B. Kraus, and P. Zoller, Theoretical and experimental perspectives of quantum verification, PRX Quantum **2**, 010102 (2021).

[19] P. Hauke, F. M. Cucchietti, L. Tagliacozzo, I. Deutsch, and M. Lewenstein, Can one trust quantum simulators?, Rep. Prog. Phys. **75**, 082401 (2012).

[20] E. Altman *et al.*, Quantum simulators: Architectures and opportunities, PRX Quantum **2**, 017003 (2021).

[21] M. Morgado and S. Whitlock, Quantum simulation and computing with Rydberg-interacting qubits, AVS Quantum Sci. **3**, 023501 (2021).

[22] V. Gebhart, R. Santagati, A. A. Gentile, E. M. Gauger, D. Craig, N. Ares, L. Banchi, F. Marquardt, L. Pezzè, and C. Bonato, Learning quantum systems, Nat. Rev. Phys. **5**, 141 (2023).

[23] P. Braccia, L. Banchi, and F. Caruso, Quantum noise sensing by generating fake noise, Phys. Rev. Appl. **17**, 024002 (2022).

[24] C.-D. Han, B. Glaz, M. Haile, and Y.-C. Lai, Tomography of time-dependent quantum Hamiltonians with machine learning, Phys. Rev. A **104**, 062404 (2021).

[25] N. Mohseni, T. Fösel, L. Guo, C. Navarrete-Benlloch, and F. Marquardt, Deep learning of quantum many-body dynamics via random driving, Quantum **6**, 714 (2022).

[26] É. Genois, J. A. Gross, A. Di Paolo, N. J. Stevenson, G. Koolstra, A. Hashim, I. Siddiqi, and A. Blais, Quantum-tailored machine-learning characterization of a superconducting qubit, PRX Quantum **2**, 040355 (2021).

[27] F. Wilde, A. Kshetrimayum, I. Roth, D. Hangleiter, R. Sweke, and J. Eisert, Scalably learning quantum many-body Hamiltonians from dynamical data, arXiv:2209.14328.

[28] E. Bairey, I. Arad, and N. H. Lindner, Learning a local Hamiltonian from local measurements, Phys. Rev. Lett. **122**, 020504 (2019).

[29] C. Di Franco, M. Paternostro, and M. Kim, Hamiltonian tomography in an access-limited setting without state initialization, Phys. Rev. Lett. **102**, 187203 (2009).

[30] J. H. Cole, S. G. Schirmer, A. D. Greentree, C. J. Wellard, D. K. Oi, and L. C. Hollenberg, Identifying an experimental two-state Hamiltonian to arbitrary accuracy, Phys. Rev. A **71**, 062312 (2005).

[31] S. J. Devitt, J. H. Cole, and L. C. Hollenberg, Scheme for direct measurement of a general two-qubit Hamiltonian, Phys. Rev. A **73**, 052317 (2006).

[32] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, and A. Laing *et al.*, Experimental quantum Hamiltonian learning, Nat. Phys. **13**, 551 (2017).

[33] A. A. Gentile, B. Flynn, S. Knauer, N. Wiebe, S. Paesani, C. E. Granade, J. G. Rarity, R. Santagati, and A. Laing, Learning models of quantum systems from experiments, Nat. Phys. **17**, 837 (2021).

[34] S. Nakajima, On quantum theory of transport phenomena: Steady diffusion, Prog. Theor. Phys. **20**, 948 (1958).

[35] R. Zwanzig, Ensemble method in the theory of irreversibility, J. Chem. Phys. **33**, 1338 (1960).

[36] J. Cerrillo and J. Cao, Non-Markovian dynamical maps: Numerical processing of open quantum trajectories, Phys. Rev. Lett. **112**, 110401 (2014).

[37] A. Gelzinis, E. Rybakovas, and L. Valkunas, Applicability of transfer tensor method for open quantum system dynamics, J. Chem. Phys. **147**, 234108 (2017).

[38] F. A. Pollock and K. Modi, Tomographically reconstructed master equations for any open quantum dynamics, Quantum **2**, 76 (2018).

[39] L. Banchi, E. Grant, A. Rocchetto, and S. Severini, Modelling non-Markovian quantum processes with recurrent neural networks, New J. Phys. **20**, 123030 (2018).

[40] M. Link, K. Gao, A. Kell, M. Breyer, D. Eberz, B. Rauf, and M. Köhl, Machine learning the phase diagram of a strongly interacting Fermi gas, Phys. Rev. Lett. **130**, 203401 (2023).

[41] X. Wu, X. Liang, Y. Tian, F. Yang, C. Chen, Y.-C. Liu, M. K. Tey, and L. You, A concise review of Rydberg atom based quantum computation and quantum simulation, Chin. Phys. B **30**, 020305 (2021).

[42] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with Rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[43] P. P. Mazza, D. Zietlow, F. Carollo, S. Andergassen, G. Martius, and I. Lesanovsky, Machine learning time-local generators of open quantum dynamics, Phys. Rev. Res. **3**, 023084 (2021).

[44] F. Carnazza, F. Carollo, D. Zietlow, S. Andergassen, G. Martius, and I. Lesanovsky, Inferring Markovian quantum master equations of few-body observables in interacting spin chains, New J. Phys. **24**, 073033 (2022).

[45] H. P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2002).

[46] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, Completely positive dynamical semigroups of N-level systems, J. Math. Phys. **17**, 821 (1976).

[47] G. Lindblad, On the generators of quantum dynamical semigroups, Commun. Math. Phys. **48**, 119 (1976).

[48] G. Vidal, Efficient simulation of one-dimensional quantum many-body systems, Phys. Rev. Lett. **93**, 040502 (2004).

[49] M. Zwolak and G. Vidal, Mixed-state dynamics in one-dimensional quantum lattice systems: A time-dependent superoperator renormalization algorithm, Phys. Rev. Lett. **93**, 207205 (2004).

[50] J. Gray, QUIMB: A PYTHON library for quantum information and many-body calculations, J. Open Source Software **3**, 819 (2018).

[51] M. S. Byrd and N. Khaneja, Characterization of the positivity of the density matrix in terms of the coherence vector representation, Phys. Rev. A **68**, 062322 (2003).

[52] See the Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevApplied.21.L041001 for the derivation of the matrix representation of the Lindbladian, details on the training procedure, and further results.

[53] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, PYTORCH, Programming with TensorFlow: Solution for edge computing applications, 87 (2021).

[54] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[55] J. Roos, J. I. Cirac, and M. C. Bañuls, Markovianity of an emitter coupled to a structured spin-chain bath, Phys. Rev. A **101**, 042114 (2020).

[56] https://github.com/giovannicemin/lindblad_dynamics_approximator

Third publication

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
6 June 2024

REVISED
10 September 2024

ACCEPTED FOR PUBLICATION
13 September 2024

PUBLISHED
7 October 2024

PAPER

Machine learning stochastic differential equations for the evolution of order parameters of classical many-body systems in and out of equilibrium

Francesco Carnazza[1,*], Federico Carollo[1], Sabine Andergassen[2], Georg Martius[3,4], Miriam Klopotek[5] and Igor Lesanovsky[1,6]

[1] Institut für Theoretische Physik and Center for Quantum Science, Universität Tübingen, Auf der Morgenstelle 14, 72076 Tübingen, Germany
[2] Institute of Information Systems Engineering and Institute for Solid State Physics, Vienna University of Technology, 1040 Vienna, Austria
[3] Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, 72076 Tübingen, Germany
[4] Wilhelm Schickard Institut für Informatik, Universität Tübingen, Maria-von-Linden-Straße 6, 72076 Tübingen, Germany
[5] Stuttgart Center for Simulation Science, SimTech Cluster of Excellence EXC 2075, University of Stuttgart, Universitätsstraße 32, 70569 Stuttgart, Germany
[6] Centre for the Mathematics and Theoretical Physics of Quantum Non-Equilibrium Systems, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[*] Author to whom any correspondence should be addressed.

E-mail: francesco.carnazza@uni-tuebingen.de

Keywords: stochastic processes, dynamical Ising model, contact process, machine learning

Abstract

We develop a machine learning algorithm to infer the emergent stochastic equation governing the evolution of an order parameter of a many-body system. We train our neural network to independently learn the directed force acting on the order parameter as well as an effective diffusive noise. We illustrate our approach using the classical Ising model endowed with Glauber dynamics, and the contact process as test cases. For both models, which represent paradigmatic equilibrium and nonequilibrium scenarios, the directed force and noise can be efficiently inferred. The directed force term of the Ising model allows us to reconstruct an effective potential for the order parameter which develops the characteristic double-well shape below the critical temperature. Despite its genuine nonequilibrium nature, such an effective potential can also be obtained for the contact process and its shape signals a phase transition into an absorbing state. Also, in contrast to the equilibrium Ising model, the presence of an absorbing state renders the noise term dependent on the value of the order parameter itself.

1. Introduction

Stochastic processes are fundamentally important in physics [1–3]. For instance, random microscopic fluctuations can strongly impact the evolution of macroscopic physical observables, e.g. order parameters close to phase transitions. Monte Carlo methods [4–6] are often the 'benchmark' for the computational treatment of classical many-body dynamics, allowing for efficient sampling of stochastic microscopic configurations and trajectories. The large-scale dynamics of the order parameter are instead typically modeled by a stochastic differential equation. The latter contains both a force term, leading to a deterministic drift, and a noise term yielding diffusive behavior[7]. [7, 8] However, establishing a connection between fluctuating microscopic stochastic trajectories and the coarse-grained evolution of the order parameter is a challenging task that can rarely be accomplished analytically.

[7] The drift and the diffusion represent the most basic ingredients for a coarse-grained dynamics. More general forms might include memory kernels or other non-Markovian time dependencies [7, 8].

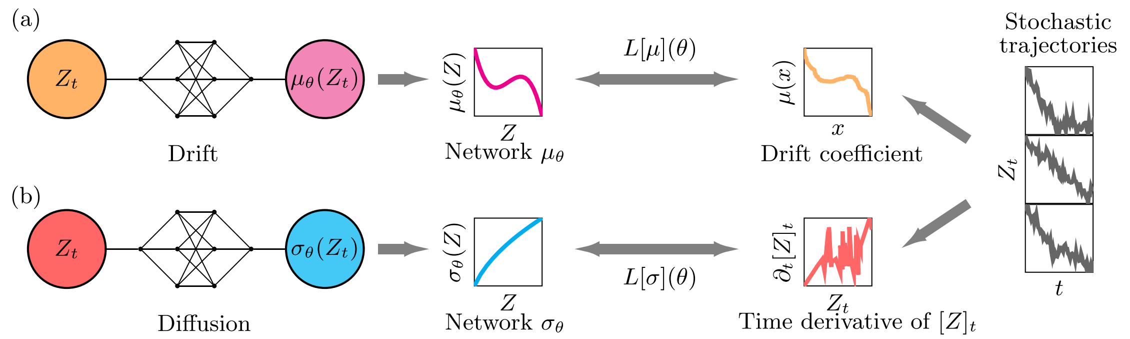

Figure 1. Training of the drift coefficient μ_θ and the diffusion coefficient σ_θ from the stochastic trajectories of the order parameter. (a) The network for the drift coefficient, μ_θ, is trained by minimizing a distance, $L[\mu](\theta)$ in equation (4), between $\mu_\theta(x)$ and the drift coefficient μ, cf equation (3). For a given value x of the order parameter, we consider all the values Z_t close to it in the observed trajectories, and their update Z_{t+dt}. The finite difference $Z_{t+dt} - Z_t$ is then used to locally approximate the drift coefficient μ. (b) The network for the diffusion coefficient σ_θ is trained by minimizing a distance, cf equation (10), between $\sigma^2_\theta(Z_t)$ and the derivative of the quadratic variation $\partial_t[Z]_t$, see $L[\sigma](\theta)$ in equation (9). The quadratic variation $[Z]_t$ itself, cf equation (6), is locally approximated from the observed trajectories.

In this paper, we develop a machine learning approach [9–14] to bridge this gap. To illustrate our method, we consider two paradigmatic classical many-body systems: the 2D Ising model evolving under Glauber dynamics [15–17] and the nonequilibrium contact process in 1D. The dynamics considered for the Ising model obey detailed balance, which eventually takes the system to a state of thermal equilibrium. As a function of temperature, this state shows a transition from a paramagnetic to a ferromagnetic state, characterized by a zero and non-zero value of the order parameter, respectively. As we will show, this transition manifests in the structure of the learned drift term cf figure 1(a), from which one can reconstruct an effective potential that exhibits a characteristic double-well shape below the critical temperature. Both the paramagnetic and ferromagnetic phases are fluctuating, which is also reflected in the learned noise term. In contrast to the scenario of the Ising model, the contact process represents a genuine out-of-equilibrium system [18–21], i.e. its dynamics does not obey detailed balance and its stationary state is non-thermal. The model features a phase transition between a non-fluctuating absorbing state in which the order parameter is strictly zero and a fluctuating active phase with a non-vanishing order parameter. Interestingly, we show that also for this genuine nonequilibrium process, an effective potential governing the deterministic drift of the order parameter can be constructed using our machine learning approach. Unlike for the Ising model, however, where the learned noise is such that both phases are fluctuating, a noise term is inferred whose strength depends on the order parameter. In particular, the noise strength tends to zero for vanishing values of the order parameter, see sketch in figure 1(b), signalling an approach to the (non-fluctuating) absorbing state.

Our method is applicable to a wide range of many-body processes in and out of equilibrium. It provides a way to determine a stochastic equation for order parameters which is intuitive and directly interpretable, as in mean-field theories. Remarkably, it also carries information about the exact low-dimensional physics of the considered model, as we demonstrate through estimates of critical exponents. Moreover, our method should also be applicable for inferring effective stochastic differential equations for the evolution of order parameters from experimental data. Our approach is based on learning ordinary differential equations, which are relatively straightforward to handle. In contrast, alternative methods focus on learning the full probability distribution governing the stochastic process [22–27]. These approaches often require assumptions about the functional form of the probability distribution, such as modeling it with a set of Gaussian distributions. Other techniques learn the stochastic differential equation by integrating the stochastic dynamics and optimizing over the probability distribution of the variable [28–34], but they encounter difficulties in the scenarios we focus on, where the data is subject to significant inherent noise. As background, we explain more in detail what are the physical quantities that we examine. These quantities describe macroscopic properties of the microscopic spin models, which are referred to as ”order parameters” in statistical mechanics. Order parameters, as discussed in [35], are named for their role in measuring the degree of order within a physical system. Typically, they have non-zero values in phases where some form of order exists, and zero values in disordered phases. The coarse-grained quantities we focus on are the averages of the binary spin values over the lattice size at each time step. We assume that their dynamics can be described by a stochastic differential equation of the Itô type whose drift and diffusion coefficient are time-independent. In figure 2, we present various configurations of the microscopic spin system for the Ising model. Specifically, in figure 2(a), we illustrate several configurations of the microscopic spin system and the

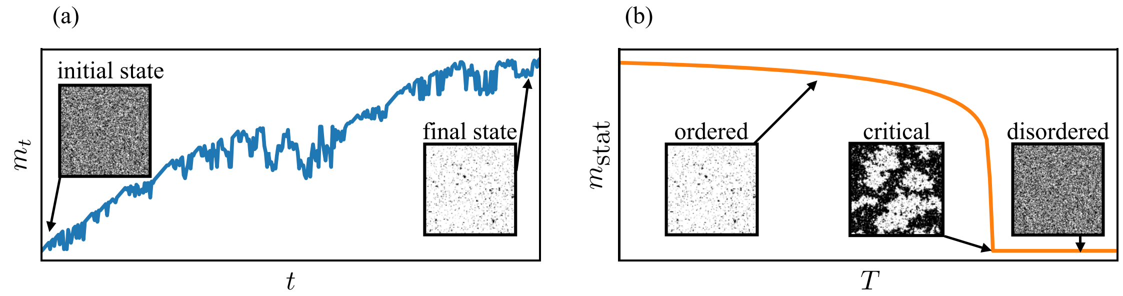

Figure 2. The stochastic dynamics of the order parameter and its steady state for the dynamical Ising model. In panel (a), the evolution of the order parameter, specifically the magnetization m_t, for the dynamical Ising model is depicted as a function of time t at a specific value of the control parameter, namely, the temperature T. The insets show the actual corresponding configurations of the entire spin system at the initial and final states. In the Ising model, varying the temperature results in three distinct phases: the ordered phase, the critical phase, and the disordered phase. The critical phase appears at the critical temperature T_c (see panel (b)). Below the critical temperature, the configurations evolve towards a stationary magnetization per single spin m_{stat} that is non-zero.

corresponding time evolution of the order parameter. In figure 2(b), we display the stationary values of the states as a function of the control parameter, which in this case is the temperature.

2. Many-body stochastic processes

2.1. The evolution of stochastic observables

For the sake of concreteness, we focus on many-body lattice systems of N sites, each of which is associated with a classical spin variable. We denote the system state, or system configuration, through the vector s containing the values s^i of the variables at the different sites i. We furthermore assume the system to be subject to a discrete-time Markovian stochastic spin-flip dynamics.

Relevant information about the above many-body system is provided by so-called order parameters, which encode properties of the whole configuration. A paradigmatic example is given by an average of the form $Z_t \equiv Z(s_t) = \frac{1}{N}\sum_i s_t^i$, where s_t is the time-evolving state of the system. As a consequence of the stochastic nature of s_t, also the effective dynamics of Z_t is stochastic. For large systems and at a continuous coarse-grained time scale, Z_t becomes a continuous random variable that may be expected to obey an emergent stochastic differential equation of the form

$$\mathrm{d}Z_t = \mu\left(Z_t, t\right)\mathrm{d}t + \sigma\left(Z_t, t\right)\mathrm{d}W_t\,. \tag{1}$$

Here, the function μ is referred to as the drift coefficient, while σ is called diffusion coefficient. W_t is a standard Wiener process [1] and $\mathrm{d}W_t$ is its increment satisfying the relations $\mathbb{E}[\mathrm{d}W_t] = 0$ and $\mathbb{E}[\mathrm{d}W_t^2] = \mathrm{d}t$, with $\mathbb{E}$ denoting expectation over the noise. Despite the simple form of equation (1), understanding the functional form of μ and σ is in general a difficult task. In the following, we propose a method to learn an approximation to the analytical form of the drift and the diffusion coefficients by means of neural networks. We determine two artificial neural networks μ_θ and σ_θ (see sketch in figure 1), which describe the dynamics of Z_t, given the network parameters (weights and biases) θ. We restrict ourselves to the Markovian case in which μ_θ and σ_θ do not depend on time

$$\mathrm{d}Z_t = \mu_\theta\left(Z_t\right)\mathrm{d}t + \sigma_\theta\left(Z_t\right)\mathrm{d}W_t. \tag{2}$$

To approximate the functions μ and σ we use a data-driven method, i.e. the networks μ_θ and σ_θ are trained on a data set composed of trajectories Z_t, which we call ground truth data, see also figure 1. Note that restricting to the Markovian case of equation (2) is an assumption since, even if the dynamics of the system configuration s_t is Markovian at the microscopic scale, the emergent dynamics of the order parameters—i.e. macroscopic quantities—may feature non-Markovian effects.

2.2. Neural network representation of the drift and diffusion coefficients

Our approach consists of training the networks μ_θ and σ_θ with separate routines, independently from each other.

As we discuss below, this means that the data sets used for training, despite being derived from the same sets of trajectories, differ from each other. Specifically, in the case of the drift coefficient, the network is trained using the first moments of the (infinitesimal) finite differences between adjacent time steps in the trajectories. For the diffusion coefficient, the network is trained using the time derivative of the quadratic variation, which is akin to the second moment of the finite differences between two different times in the trajectories. In fact, nothing prevents us from training the drift and diffusion terms using completely different sets of trajectories.

The drift term can be quantified by exploiting averages over trajectories of the infinitesimal increment $\mathrm{d}Z_t$ in equation (1). More precisely, starting from equation (1) it is possible to show that the function μ at point x can be obtained as the limit [36–38]

$$\mu\left(x\right) = \lim_{\mathrm{d}t \to 0+} \frac{\mathbf{E}_{Z_t = x}\left[Z_{t+\mathrm{d}t}\right] - x}{\mathrm{d}t}, \tag{3}$$

where $\mathbf{E}_{Z_t=x}$ denotes expectation conditional on the process being in x at time t. In the theory of stochastic processes, the above limit also provides the action of the so-called infinitesimal generator $\mathcal{W}$ on the function x, $\mu(x) = \mathcal{W}[x]$ [38].

The limit in equation (3) can be estimated from the data set, as sketched in figures 1–3. To this end, we generate batches $X_i = \{x_1, \ldots, x_{d_{\mathrm{batch}}}\}$ of size d_{batch}. Each x_j in X_i is extracted randomly between the minimum and maximum values of the trajectories Z_t. For each x_j, we consider all the n_j points Z_t^j, in all trajectories, which belong to the interval of width δ around x_j, see figure 3(a). The value of δ has to be chosen in such a way that all bins associated with the different x_j are sufficiently populated, ensuring the smoothness of the learned $\mu(x_j)$. We check *a posteriori* that the predicted dynamics, learned with such a δ, corresponds to the ground truth (see appendix A)[8].. We optimize μ_θ by minimizing the following loss function, cf figure 3(b)

$$L\left[\mu_\theta\right]\left(\theta\right) = \sum_{j=1}^{d_{\mathrm{batch}}} \left| \mu_\theta\left(x_j\right) - \frac{1}{n_j} \sum_{Z_t^j} \Delta_1 Z_t^j \right|, \tag{4}$$

where $\Delta_1 Z_t \equiv (Z_{t+\mathrm{d}t} - Z_t)/\mathrm{d}t$. We consider the coarse-grained adimensional time t to correspond to the number of discrete-time updates of the system normalized by a suitable factor τ and thus $\mathrm{d}t = 1/\tau$.

In our data sets, the observed noise is often larger than the drift, cf figures 3(c) and 7(c), especially near the stationary state, where the drift coefficient vanishes altogether. This is why computing the targets $\frac{1}{n_j}\sum_{Z_t^j} \Delta_1 Z_t^j$ in equation (4) is essential. In fact, no learning would be possible without taking the targets to be arithmetic averages, due to the above-mentioned large fluctuations.

Since our task is to understand the order-parameter dynamics, we restrict ourselves to the problem of learning one-dimensional data. This allows for an efficient estimate of the drift coefficient in equation (2). In one dimension, the stochastic quantity Z_t indeed hits the different intervals sufficiently many times during the evolution, which is needed for proper sampling and computing $\mu(x)$. To reduce over-fitting, we train n_{train} different models $\mu_\theta^i(x)$, with the loss function (4). To each of these models, we assign a weight w_i equal to the inverse of the mean square error between the data estimate of $\mu(x)$ and the network result μ_θ^i. As a reference model μ_θ, we take the weighted average over this 'ensemble' of models:

$$\mu_\theta = \sum_{i=1}^{n_{\mathrm{train}}} \frac{w_i\, \mu_\theta^i}{\sum_{i=j}^{n_{\mathrm{train}}} w_j}. \tag{5}$$

Specifically, the values of $n_{\mathrm{train}}, d_{\mathrm{batch}}, n_{\mathrm{epochs}}$ and δ we adopt for the considered models are reported in table A1 (see appendix A).

In order to learn the diffusion coefficient, we use the 'second moment' of $\mathrm{d}Z_t$, which is the quadratic variation $[Z]_t$. For stochastic processes as in equation (1), this is given by [39, 40]

$$[Z]_t = \int_0^t \mathrm{d}Z_s^2 = \int_0^t \mathrm{d}s \sigma^2\left(Z_s\right), \tag{6}$$

which is nothing but the integral version of the differential equation

$$\partial_t\, [Z]_t = \sigma^2\left(Z_t\right). \tag{7}$$

[8] This interval is defined as $B_j^\delta \equiv \{\text{All } Z_t \text{such that } |Z_t - x_j| < \delta\}$, and its cardinality is $n_j \equiv \#B_\delta^j$. We denote the points in this interval as Z_t^j.

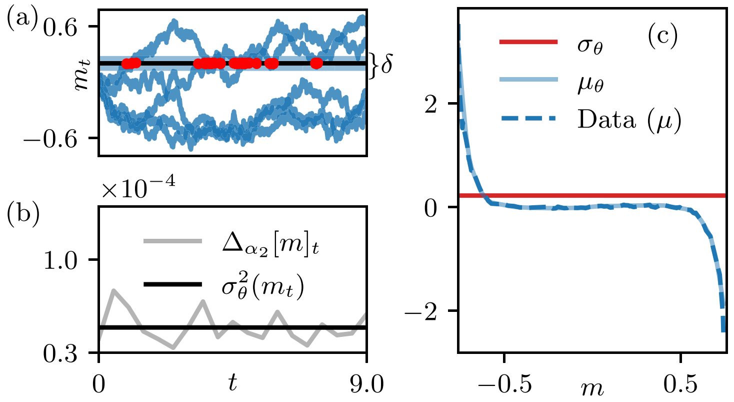

Figure 3. Estimation of the drift and diffusion coefficients μ and σ. In panel (a), some exemplary trajectories Z_t for the Ising model on a 128 × 128 square lattice are depicted. Here, the order parameter Z_t is the magnetization m_t. These trajectories are generated using the dynamical rules reported in equation (14) and provide the variable m_t. For a given m_t (black solid line), all the n_i intersections (red dots) within the shaded area are used to compute the quantity $\mu(x)\mathrm{d}t$. The latter is plotted in panel (c) (dashed line), while the solid line is obtained from the trained network $\mu_\theta(m)$. In panel (c) the network for the diffusion coefficient $\sigma_\theta(m)$ is also shown, in order to compare its dependence on the magnetization m with the one of the drift. This can also be seen in panel (b), where the values for the finite difference of the quadratic variation $\Delta_{\alpha_2}[m]_t$, upon which the network is trained, are provided together with the values of the corresponding predictions of the neural network $\sigma^2_\theta(m_t)$.

To train the network for the diffusion coefficient σ_θ, we devise a coarse-graining procedure that makes the spin-flip noise of the stochastic many-body dynamics look like a Wiener process. To this end, we first compute the quadratic variation from trajectories as

$$[Z]_t \cong \sum_{u\in[0,t]} \left(Z_{u+\alpha_1 \mathrm{d}u} - Z_u\right)^2 . \tag{8}$$

Here, the integer factor $\alpha_1 \geqslant 1$ may allow one to magnify the variation at the different times. Furthermore, we approximate equation (7) by

$$\partial_t [Z]_t \approx \Delta_{\alpha_2} [Z]_t = \frac{[Z]_{t+\alpha_2 \mathrm{d}t} - [Z]_t}{\alpha_2 \mathrm{d}t}. \tag{9}$$

The factor $\alpha_2 \gg 1$ allows one to coarse-grain the noise over many discrete time-steps, which proved necessary for convergence during the training procedure. This is mainly due to the fact that the finite difference in equation (9) is stochastic. For this reason we need an average in order to obtain valuable information for the training. Equation (9) will still be a good approximation of a time derivative if we consider a time window $\alpha_2 dt$ much smaller than the time during which relaxation to stationarity takes place. The optimization of the network parameters is then performed by minimizing the loss function

$$L[\sigma_\theta](\theta) = \sum_t |\Delta_{\alpha_2} [Z]_t - \sigma^2_\theta (Z_t)|. \tag{10}$$

Note that this loss function is insensitive to the sign of σ_θ. This is not a problem since the stochastic increment $\mathrm{d}W_t$ is symmetric under a change of sign.

To summarize, we explicitly selected the following hyper-parameters to control convergence: the number of hidden layers, the learning rate, the number of epochs, the batch size d_{batch}, and the number of models used to compute an ‘average model’ n_{train}. Additionally, we chose a threshold δ from which to compute finite increments from a given initial condition, and a time step dt. We also set two constants α_1 and α_2. The constant α_1 is used in approximating the quadratic variation: $[Z]_t = \sum_u (Z_{u+\alpha_1 \mathrm{d}u} - Z_u)^2$. The constant α_2 is used in computing its finite difference: $\sigma^2(Z_t) = ([Z]_{t+\alpha_2 \mathrm{d}t} - [Z]_t)/\alpha_2 \mathrm{d}t$. For further details on the training procedure, we refer to the appendix A and to table A1.

3. The kinetic ising model

3.1. The model and its dynamics

The Ising model is a paradigmatic model of statistical mechanics. It provides a qualitative description of the behavior of molecular magnetic dipoles in a metal. The crystalline structure of the metal is modeled as a lattice of N sites. At each site $i = 1, 2, \ldots, N$, a magnetic dipole is represented as a spin variable $s^i = \pm 1$. The spins interact with each other according to the following energy functional (Hamiltonian)

$$H(s) = -\frac{1}{2}\sum_{\langle ij \rangle} s^i s^j. \tag{11}$$

Here, the notation $\langle ij \rangle$ restricts the sites i and j in the sum to be nearest neighbors on the lattice. We consider a two-dimensional square lattice. This Hamiltonian presents a $\mathbf{Z}_2$ symmetry since it is invariant under sign change of every spin variable $s^i \to -s^i$. At thermal equilibrium at a given temperature T, each spin configuration has a probability described by the Boltzmann distribution $\pi_{\mathrm{B}}(s) \propto e^{-H(s)/k_{\mathrm{B}}T}$, where k_{B} stands for the Boltzmann constant. Given the magnetization

$$m = \frac{1}{N}\sum_i s^i, \tag{12}$$

the order parameter of the model is the expectation of the absolute value of m in the Boltzmann distribution. The system undergoes a continuous transition from an ordered phase with finite magnetization at sufficiently low temperatures, to a disordered one with vanishing magnetization. While in one dimension the model predicts a finite magnetization only at zero temperature, in two dimensions the critical temperature T_{c} corresponding to the phase transition is finite. Close to T_{c}, the value of the average magnetization $\bar{m}$ follows a power-law behavior

$$\bar{m} \propto |T - T_{\mathrm{c}}|^{\beta}, \tag{13}$$

where β is a so-called critical exponent.

The Ising model discussed above does not possess inherent dynamics. In order to apply our ML method to this model we can endow it with Glauber dynamics using Metropolis–Hastings sampling, which is usually utilized for sampling the Boltzmann distribution of the model. Such a dynamic is defined by the single spin-flip probabilities $P = P(s^i \to -s^i)$, updating the spin variables in the lattice according to

$$P\left(s_t^i \to -s_t^i\right) = \begin{cases} \exp\left(-\Delta E/k_{\mathrm{B}}T\right) & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leqslant 0 \end{cases} \tag{14}$$

where $\Delta E = H(s_t^1, \ldots, -s_t^i, .., s_t^N) - H(s_t^1, \ldots, s_t^i, .., s_t^N)$ is the energy change associated with the transition. For the completion of a single discrete time step $s_t \to s_{t+1}$, a single spin-flip is attempted N times at a random site. For such a *dynamical* Ising model, the (stochastically evolving) order parameter m_t is defined as in equation (12) for an evolving configuration s_t. We choose each of the spins in the initial configuration to be up or down with equal probability, so that for large systems $m_0 \approx 0$. For further detail about the model and its field theoretical representation, see appendix B.1.

3.2. Neural network results

Given a set of trajectories for m_t at temperature T, we learn the corresponding drift term μ_θ^T using the approach explained above and the loss function $L[\mu^T](\theta)$ in equation (4). The drift term essentially acts as a directed force on the order parameter and it is thus natural to define an effective potential driving the motion of m_t via the integral

$$M_\theta^T(m) \equiv -\int_0^m \mathrm{d}x \mu_\theta^T(x). \tag{15}$$

Our results reported in figure 4 show that upon increasing T the effective potential undergoes a transition from a functional form exhibiting a double well to a single well potential. This fact is connected with the equilibrium Ising phase transition which can be understood as follows. The stationary values of the expectation of the order parameter $\bar{m}_{\mathrm{stat}}$ correspond to the minima of the effective potential M_θ^T, see figure 4, and thus to zeroes of the drift coefficient, $\mu_\theta^T(\bar{m}_{\mathrm{stat}}) = 0$, cf figure 5. Since the considered discrete-time dynamics samples the Boltzmann distribution at stationarity, one essentially has that the stationary values $\bar{m}_{\mathrm{stat}}$ should approximate the equilibrium order parameter $\bar{m}$, thus connecting the retrieved potential to the Ising transition.

To benchmark the results from the trained networks μ_θ^T, we can thus extract the critical temperature T_{c} and the critical exponent β of the order parameter and compare them with the known values for the Ising

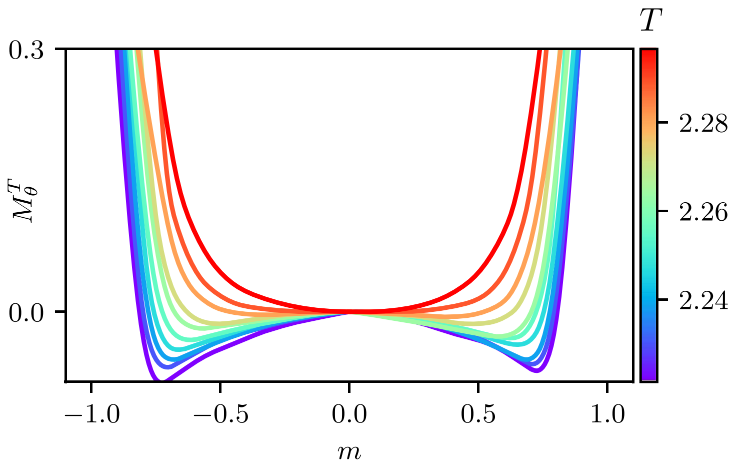

Figure 4. Effective potential M_θ^T of the Ising model. The effective potential M_θ^T in equation (15) is determined by the integral of the drift μ_θ with respect to the magnetization m. One can observe that for increasing temperature this effective potential changes its functional form from having two distinct minima in the double-well potential to having only one minimum at $m = 0$.

model. We fit the stationary magnetization $\bar{m}_{\text{stat}}$ to the scaling form of equation (13) by minimizing the function

$$\epsilon\left(\tilde{c}_1, \tilde{T}_{\text{c}}, \tilde{\beta}\right) = \sum_T |\mu_\theta^T\left(\tilde{c}_1|T - \tilde{T}_{\text{c}}|^{\tilde{\beta}}\right)|^2. \tag{16}$$

The positive function $\epsilon(\tilde{c}_1, \tilde{\beta}, \tilde{T}_{\text{c}})$ vanishes when $\tilde{c}_1|T - \tilde{T}_{\text{c}}|^{\tilde{\beta}} = \bar{m}_{\text{stat}}$. We consider the values of c_1, T_{c} and β that minimize ε in equation (16). To find them, the zeroes of the drift coefficient μ_θ^T, are computed using the exact derivatives via automatic differentiation. This is possible since we use differentiable neural networks. We find the following values: $\beta = 0.156 \pm 0.001$, $T_{\text{c}} = 2.271 \pm 0.001$, and $c_1 = 1.076 \pm 0.002$. Note that the errors reported are only those related to the fit and do not consider finite-time and finite-size errors. For the Ising model, the analytical values are $\beta = 1/8$ and $T_{\text{c}} = 2/\ln(1 + \sqrt{2}) \cong 2.269$ [41]. Our results are thus in good agreement with the exact values and show that the networks are able to provide a sound description of the critical behavior encoded in the data they are trained on.

Close to the critical point, the Ising model with Glauber dynamics is expected to fall in the model A class according to the Halperin classification [42]. This is a pure relaxation model for a time dependent field in a double well potential, subject to uncorrelated white noise [43–46]. The latter feature is indeed reflected in our results on the learned diffusion coefficient σ_θ, shown in figures 3(b) and (c). There, we present σ_θ for $T = 2.269$, which is in proximity to the critical temperature. As can be seen, the diffusion coefficient σ_θ is essentially constant when compared with the drift coefficient, entailing white noise in the dynamics of m_t.

We thus showed how the learned networks are able to encode significant information about the statics, i.e. the order-disorder phase transition (see figure 5) and the dynamics, i.e. the form of the noise, for the process under investigation, through a simple equation.

4. The contact process

4.1. The model

We now apply our method to a paradigmatic nonequilibrium process, the so-called contact process [47, 48]. It was introduced to describe epidemic spreading in the absence of immunization. It is not defined via an energy function but solely via dynamical rules. The contact process shows a nonequilibrium continuous phase transition which belongs to the directed percolation universality class [49–53].

Within the epidemic spreading interpretation of the model, each lattice site i represents an individual which can either be found in the healthy state $s_t^i = 0$ (inactive site) or in the infected state $s_t^i = 1$ (active site). We consider here the case of a one-dimensional lattice. The dynamics occur in discrete time as follows: first, given the configuration s_t at time t, we calculate the probability that each spin flips through the rules

$$\begin{aligned} P\left[0 \to 1, n_t^i\right] &\propto \kappa \mathrm{d}t n_t^i/2\,, \\ P\left[1 \to 0, n_t^i\right] &\propto \gamma \mathrm{d}t\,. \end{aligned} \tag{17}$$

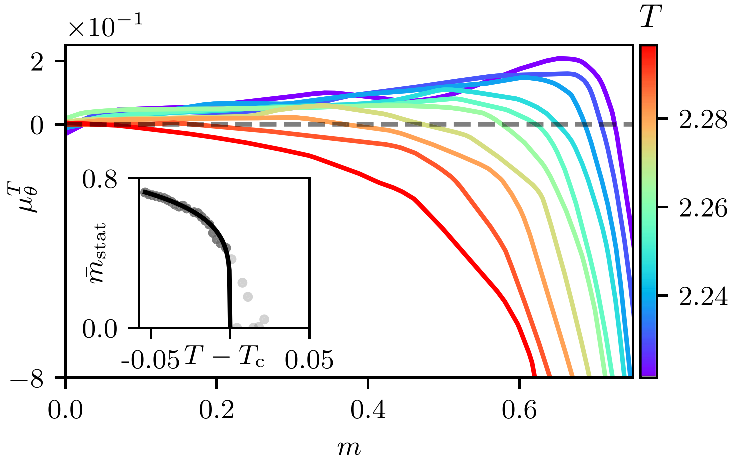

Figure 5. Neural networks for the drift coefficients μ_θ^T for the Ising model. Due to the $\mathbb{Z}_2$ symmetry of the Hamiltonian, only the behavior for $m > 0$ is shown. Warmer tones correspond to higher temperatures. The higher the temperature, the closer the non-trivial zero of μ_θ^T corresponding to $\bar{m}_{\text{stat}}$ is to zero. In the main figure, the dashed line at zero intercepts the graph of the function μ_θ^T, at the stationary value $\bar{m}_{\text{stat}}$ for each temperature T. Specifically, these values correspond to points where $\mu_\theta^T(\bar{m}_{\text{stat}}) = 0$. In the inset, $\bar{m}_{\text{stat}}$ values are shown as a function of their respective temperatures as grey bullets. Additionally, the inset displays the power law behavior of $\bar{m}_{\text{stat}}$ relative to the distance from the critical temperature T_{c}, derived by minimizing the function $\epsilon(\tilde{c}_1, \tilde{T}_{\text{c}}, \tilde{\beta})$ in equation (16) as a solid black line.

Here, we introduced the healing rate γ, the infection rate κ and n_t^i indicates the number of infected nearest neighbors of i. Then, according to the above probability, a spin is extracted, and the corresponding flip is performed. The order parameter is the number density of infected sites

$$\rho_t = \frac{1}{N}\sum_i s_t^i, \tag{18}$$

with N being the total number of sites. We here consider $\rho_0 = 1$ as the initial value for the density (all sites infected).

From the dynamical rules in equation (17), one can see that the state with all healthy sites is a stationary state. In fact, this is a so-called absorbing state since it can be reached during the dynamics but it cannot be left. For any finite system, there is always a finite probability of hitting the absorbing state, which is the unique stationary state of the system. In the thermodynamic limit ($N \to \infty$) and for sufficiently large infection rates, a phase with a finite density of infected sites, usually called fluctuating phase [52–54], becomes stable. In finite systems, this phase eventually dies out and only appears within a meta-stable timescale. The absorbing phase and the fluctuating phase are separated by a continuous phase transition occurring at a finite critical value of the infection rate κ_{c}, above which the system features a nonzero expectation of the stationary density $\bar{\rho}_{\text{stat}}$. In proximity to the phase transition, the density follows a power-law behavior

$$\bar{\rho}_{\text{stat}} \propto |\kappa - \kappa_{\text{c}}|^{\beta}. \tag{19}$$

In the following, we focus on a one-dimensional lattice made of 100 sites and measure the infection rate in units of γ.

4.2. Neural network results

We start by discussing the results for the drift term of the contact process. As for the kinetic Ising model, we train the network for many data sets of trajectories. For each data set at infection rate κ, we train a model μ_θ^κ. The results for learned drifts μ_θ^κ are shown in figure 6, for different values of κ. Decreasing κ, the zero crossings $\mu_\theta^\kappa(\bar{\rho}_{\text{stat}}) = 0$ occur at progressively smaller values of $\bar{\rho}_{\text{stat}}$. In the inset, we illustrate how these can be used to extract the critical infection rate κ_{c} and the associated critical exponent β. As for the kinetic Ising model, we can fit the density of infected sites to the power law of equation (19) by minimizing the function $\epsilon(\tilde{c}_1, \tilde{\kappa}_{\text{c}}, \tilde{\beta})$:

$$\epsilon\left(\tilde{c}_1, \tilde{\kappa}_{\text{c}}, \tilde{\beta}\right) = \sum_\kappa \left|\mu_\theta^\kappa\left(\tilde{c}_1 |\kappa - \tilde{\kappa}_{\text{c}}|^{\tilde{\beta}}\right)\right|^2. \tag{20}$$

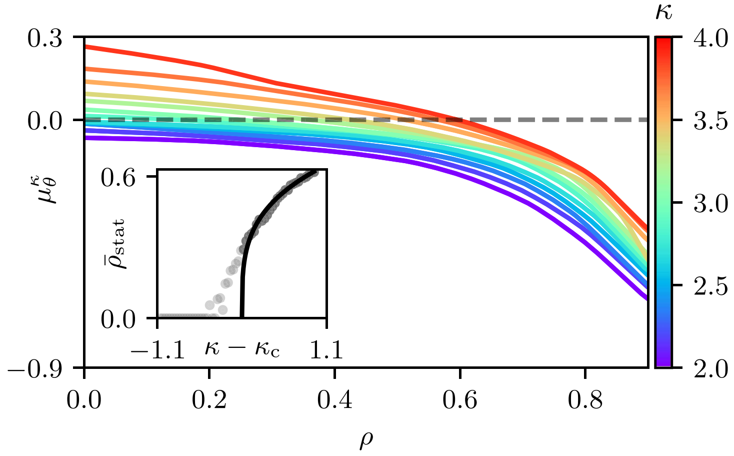

Figure 6. Drifts μ_θ^κ as a function of ρ, for different values of the infection rate. In the main figure, the drifts, which are color coded according to the infection rate κ, are depicted. The zeros of the drift term correspond to the stationary values of the stable stationary points $\bar{\rho}_{\mathrm{stat}}$, where $\mu_\theta^\kappa(\bar{\rho}_{\mathrm{stat}}) = 0$. The dashed line in the figure indicates where these zeros intersect with the drift coefficient. As the infection rate decreases, the stationary density collapses to $\bar{\rho}_{\mathrm{stat}} = 0$. From these values, represented as grey bullets in the inset, the critical exponent β can be extracted in the proximity of the critical infection rate κ_c. The solid black line shown in the inset represents the power law behavior of $\bar{\rho}_{\mathrm{stat}}$ as a function of the infection rate κ. This power law is obtained by minimizing the function $\epsilon(\tilde{c}_1, \tilde{\kappa}_\mathrm{c}, \tilde{\beta})$ in equation (20). The drift terms in the active phase do not vanish when approaching $\rho = 0$ as one may expect. As discussed in the main text, this is a consequence of the fact that trajectories in the active phase do not visit sufficiently often values below the stationary values $\bar{\rho}_{\mathrm{stat}}$, which affects the training procedure.

The values c_1, κ_c and β that we find are $\beta = 0.28 \pm 0.03$, $\kappa_\mathrm{c} = 3.062 \pm 0.003$. These values should be compared with the values obtained by means of Monte Carlo or series expansion $\kappa_\mathrm{c} = 3.29785(8)$ [55–57], $\beta = 0.276486(8)$ [58, 59].

Albeit this agreement, there is in fact a problem with the shape of the learned drifts μ_θ^κ, as shown in figure 6. Given that the contact process features an absorbing state at density $\rho = 0$, one should expect that the drift vanishes for this density. This is evidently not the case here. The reason lies in the fact that the physics actually influences the way in which training data can be produced. In our case, we train the network considering trajectories starting from the state with all sites infected. For such initial condition and being in the active phase, the density of infected sites will decrease with time until it reaches a (meta)stable finite value around which it will fluctuate. This implies that during the learning process values of the density smaller than the (meta)stable one, including the absorbing-state value $\rho = 0$, are not visited sufficiently often. Therefore, it is not possible to appropriately learn the drift term below such values.

In figure 7, we report the results for the learned diffusion coefficient $\sigma_\theta(\rho)$, together with the network prediction for $\mu_\theta(\rho)$ and the time derivative of the quadratic variation $\Delta_{\alpha_2}[\rho]_t$, which the network learns (details on the network parameters are given in table A1). We consider a value for the infection rate, $\kappa = 3.36$, in the proximity of the critical point κ_c. In contrast to the Ising model, where both phases above and below the critical point are fluctuating, the presence of an absorbing phase dictates that the diffusion coefficient must vanish at zero density. This means that the noise must be multiplicative. In fact, it can be proven that the diffusion coefficient is proportional to the square root of the density [54, 60, 61], which is a consequence of the central limit theorem and the fact that only active sites can contribute to fluctuations (for details we refer to appendix B.2). Both the learned diffusion coefficient σ_θ and the drift μ_θ are not constant and approach zero for small ρ, see figure 7(c). They are not strictly zero at $\rho = 0$ due to the above-discussed limitations of the learning procedure.

In figure 7(a) we show a selected trajectory, for which we display ρ_t, We see that σ_θ yields a time averaged value of the (coarse-grained) derivative of the quadratic variation $\Delta_{\alpha_2}[\rho]_t$ on which it was trained. Moreover, we also see that the learned noise vanishes as the system enters the absorbing state, i.e. $\rho_t = 0$ (cf figure 7(b)).

5. Conclusions

We have shown how to encode a simple stochastic equation in an artificial neural network and applied this method to two paradigmatic models of statistical mechanics, both in and out of equilibrium. Both studied systems, the kinetic Ising model and the contact process, exhibit a continuous phase transition which also is captured by the network. For both models we identified the critical point and retrieved the static critical exponent β.

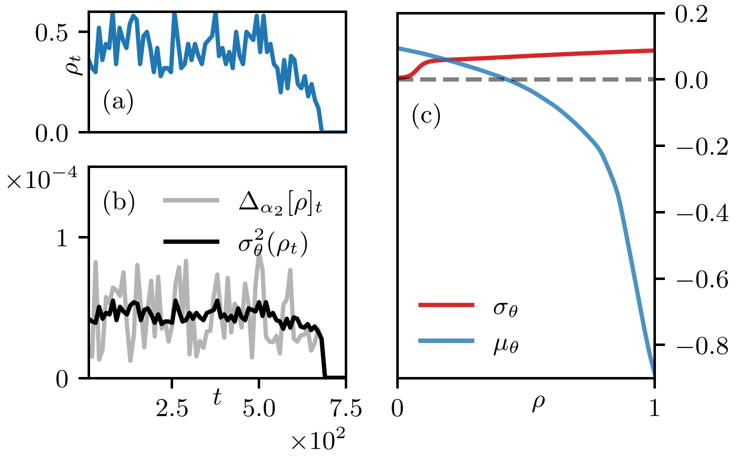

Figure 7. Diffusion in the contact process. (a) A sample trajectory ρ_t for $\kappa = 3.36$. Such trajectories ρ_t are used to compute the derivative of the quadratic variation $\Delta_{\alpha_2}[\rho]_t$. This time derivative is used to train the network σ_θ, via the loss in equation (10). (b) The network takes as input the values of ρ_t of a given trajectory and outputs the values $\sigma^2_\theta(\rho_t)$, which approximate the derivative of the quadratic variation $\Delta_{\alpha_2}[\rho]_t$, see equation (6). Unlike in the Ising model, σ_θ is not a constant function of the order parameter ρ_t. The reason is that in the absorbing phase (where the order parameter is strictly zero) no fluctuations take place, as shown the same trajectory in panel (a). (c) Learned diffusion coefficient σ_θ as a function of the density ρ. We observe that σ_θ indeed goes to zero for vanishing order parameter ρ, signalling multiplicative noise. For comparison, we also provide the results for the corresponding drift μ_θ.

It is important to note that within the chosen approach the network does not learn the order parameter from raw configurations. Rather, it is fed with a one-dimensional average value of an order parameter (density or magnetization) and outputs the one-dimensional drift and diffusion coefficients for a given order parameter value. The network thus learns one-dimensional quantities which simplifies the training process. In the case of the contact process, a multiplicative form of the noise is retrieved, while for the kinetic Ising model, the network learns a noise form that is approximately constant, i.e. independent of the value of the order parameter.

A natural future development would be to use the learned drift as a scaling function and to obtain all the critical exponents. This approach might also prove useful in classifying universal behavior of different processes, as two models are expected to belong to the same class, not only if they share the same set of critical exponents, but also if they share the same scaling function. Another point for future exploration is to go beyond the inherently Markovian assumption in equation (1), as the success of the results reported here, even under this assumption, could be attributed to the one-dimensional character of the training data. Future aims include the application of our approach to trajectories of open quantum processes and the utilization of machine learning methods that automatically infer the relevant order parameter [34].

Data availability statement

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

Acknowledgments

We thank M. Magoni for insightful discussions. We acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-Number 2064/1-Project Number 390727645 (the Tübingen Machine Learning Cluster of Excellence), EXC Number 2075-Project Number 390740016 (the Stuttgart Cluster of Excellence SimTech), under Project No. 449905436, and through the Research Unit FOR 5413/1, Grant No. 465199066. This project has also received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No. 101046968 (BRISQ). F. Carollo is indebted to the Baden–Württemberg Stiftung for the financial support of this research project by the Elite Programme for Postdocs. We acknowledge support from the Open Access Publication Fund of the University of Tübingen.

Table A1. Details of the networks' architecture and of the training procedure. All the trained networks are fully connected feed-forward multi-perceptron networks [63, 64]. In the first column, the number of neurons n_i in the i^{th} layer is reported, as $n_1 \times \ldots \times n_{\text{out}}$, with i in $\{1, 2, \ldots, \text{out}\}$. To train each network, the optimizer RMSprop algorithm in the PyTorch implementation is used, where only the learning rates were modified. The specific activation functions are also displayed. The quantity n_{train} refers to how many models are trained on each dataset, weighted averages of which are taken to compute the reference model. The dimension of the batch use to train is d_{batch}, while n_{epochs} is the number of epochs. The width of the interval from which μ is computed (cf figure 3) is δ, while the coarse graining of the discrete time is τ. The constant α_1 is used in computing the approximation for the quadratic variation $[Z]_t = \sum_u (Z_{u+\alpha_1 \mathrm{d}u} - Z_u)^2$. The constant α_2 is instead using in computing its finite difference: $\sigma^2(Z_t) = ([Z]_{t+\alpha_2 \mathrm{d}t} - [Z]_t)/\alpha_2 \mathrm{d}t$.

Network details						
Model	Layers architecture	Learning rate (RMSprop optimizer)	Activation function	n_{train}, d_{batch}, n_{epochs}	δ,τ	α_1, α_2
μ_θ Ising	$1 \times 50 \times 50 \times 1$	0.3×10^{-3}	ReLU	10, 100, 7000	0.01, 1000	—
σ_θ Ising	$1 \times 64 \times 1$	10^{-3}	Tanh (intra-layers), Sigmoid (output)	10, 100, 5000	—	1, 500
μ_θ Contact process	$1 \times 50 \times 1$	0.5×10^{-3}	ReLU	10, 100, 2000	0.05, 100	—
σ_θ Contact process	$1 \times 64 \times 1$	10^{-3}	Tanh (intra-layers), Sigmoid (output)	10, 100, 5000	—	10 100

Appendix A. Neural network details and integration of the learned stochastic equations

Because of the different properties of the two models considered in the present work, the kinetic Ising model and the contact process, the employed networks and the hyper-parameters adopted to train them are slightly different. In the following, we specify the details of the networks and how the integration of the Itô equation is performed. The code we use is available at [62].

A.1. Neural network and training details

We model the drift μ_θ as a fully connected feed-forward neural network. The network is trained by employing back propagation methods to optimize the loss function (4) This optimization minimizes the distance between the function $\mu_\theta(x)$ and the drift coefficient μ. The back propagation lets us compute the gradients used in an optimization routine. This routine requires as input a constant, namely, the learning rate, which amounts to the optimization step in the gradient descent algorithm. The order of magnitude of the learning rate should be small enough to learn the data's essential details yet not too small to avoid learning the noise effects. Moreover, lower learning rates make the optimization procedure slower. The learning rate we choose is thus a compromise between the optimization velocity and the accuracy of the results. We optimize the network to learn $\Delta_1 Z_t \tau$. The learned μ_θ then has to be multiplied with τ to make it comparable with the training data. For the Ising model, the time scale τ is set to 1000. For the contact process, it is 100. Similarly, the network σ_θ is a fully connected feed-forward network. As for μ_θ, the input and output dimensions are one-dimensional. Both for μ_θ and σ_θ the adopted optimizer is the PyTorch implementation of the RMSprop algorithm [65]. The architecture and training details for the networks and σ_θ are reported in the table A1, both for the kinetic Ising model and the contact process. For both of this processes the power law for the stationary values $\bar{m}_{\text{stat}}$ and $\bar{\rho}_{\text{stat}}$ only applies in the vicinity of the critical point, and only in the ordered and the active phase respectively. For the Ising model, the sum in equation (16) is computed for 15 values of the temperature equally spaced in between a minimum value $T_{\text{min}} = 2.2214$ and maximum value $T_{\text{max}} = 2.2759$. Similarly, for the contact process, in the sum in equation (20), we use 31 equally spaced values of the infection rate κ, from $\kappa_{\text{min}} = 2.0$ as lowest value to highest value $\kappa_{\text{max}} = 2.9831$.

To find the best critical values in equations (16) and (20), we use the minimization library [66] that allows to compute exact gradients on the (differentiable) networks μ_θ and σ_θ and minimize them numerically.

A.2. Integration of the Itô equation

In the present work, we extract an approximation to the drift coefficient μ and the time derivative of the quadratic variation $[Z]_t$ from the ground truth data. It is interesting to numerically solve the learned Itô equation (2) and compare the results with the ground truth Z_t. This can be readily done with the machine learning library Torchsde [67], which we adopt here. The numerical integration of Itô equations requires small time steps to achieve convergence. The fictitious time scale dt introduced to train the drift μ_θ thus comes in handy for the integration. One needs to pay attention to the increments $W_{t+u} - W_t$ of the Wiener

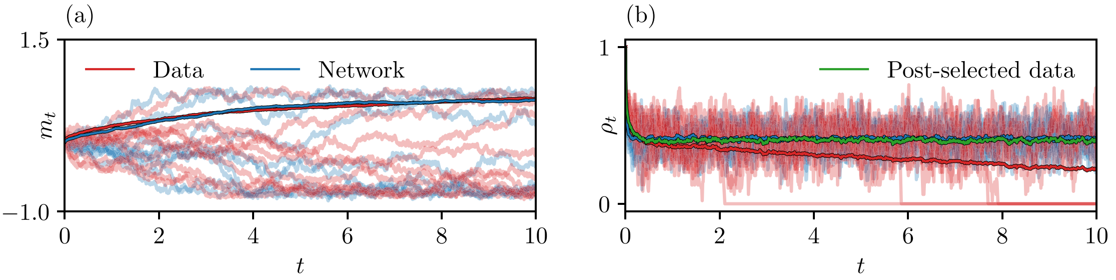

Figure A1. Dynamical trajectories for the kinetic Ising model (a) and the contact process (b). We report the ground truth trajectories ('Data', in red) as well as those generated by numerically integrating the learned Itô equation (2) ('Network', in blue). In panel (a), the time evolution of the magnetization of a two-dimensional lattice of size $N = 128 \times 128$ is shown for a temperature $T = 2.2214$. Some samples for the ground truth trajectories and the ones obtained from the network in lighter colors are displayed together with their averages over 100 trajectories in darker ones. Analogously, in panel (b), the time evolution of the density of active sites ρ_t on a one-dimensional chain of length $N = 100$ is shown for $\kappa = 3.36$, both for the ground truth and for the data generated by the network together with the respective order parameter estimated from 100 trajectories. The green line ('Post-selected data') represents the order parameter computed only from trajectories which do not decay into the absorbing state $\rho = 0$ (for the Ising model, the order parameter is the trajectory average of the absolute value of the magnetization m_t at time t).

process in equation (1), which should be distributed with a probability p following a normal distribution $\mathcal{N}(0,u)$ centered around zero and with variance u such that $p(W_{t+u} - W_t) = \mathcal{N}(0,u)$. The learned σ_θ, which is trained without using the time scale dt, has thus to be divided by $\sqrt{\mathrm{d}t}$ to make it comparable with the values obtained from the drift.

The results obtained by integrating the learned Itô equation (2) present a similar behavior to the ground truth dynamics, see figure A1(b).

For the contact process, our goal is to describe the dynamics up to the non-absorbing stationary state. For this reason, we restrict the training of the drift function μ_θ to $\rho_t > \delta$ in the data set, neglecting the region near the absorbing state ($\rho_t = 0$). This allows us to consider short trajectories while retaining important information about how the active (non-absorbing) stationary state ρ_{stat} is reached. This implies that the learned drift function μ_θ has only one stationary state, that is, a zero, in ρ_{stat}, but not in $\rho_t = 0$. When integrating the Itô equation, no trajectory thus goes to the absorbing state, something that instead happens to the ground truth data. The average of the ground truth data (referred to as 'Data' in figure A1) thus slowly decreases towards zero, while the average of only those trajectories in the ground truth data that do not end in the absorbing state exhibits a non-zero stationary state (indicated by 'Post-selected data' in figure A1). The latter agrees with the average obtained from integrating the learned Itô equation (indicated by 'Network' in figure A1). In this figure, the first moment of the dynamics is reported in darker tone both for the magnetization of the Ising model $\mathbf{E}[m_t]$ and the density of active sites in the contact process $\mathbf{E}[\rho_t]$. A possible measure $\mathbf{d}(m_t^{\text{data}}, m_t^{\text{net}})$ of the accuracy of the model is the averages difference between the first moment of the magnetization (density) m_t^{data} (ρ_t^{data}) and the one predicted by the neural network m_t^{net} over the time window $[0,T]$:

$$\mathbf{d}\left(m_t^{\text{data}}, m_t^{\text{net}}\right) = \frac{1}{T}\int_{[0,T]} \mathrm{d}t \left(\mathbf{E}\left[m_t^{\text{data}}\right] - \mathbf{E}\left[m_t^{\text{net}}\right]\right)^2. \tag{A.1}$$

In the reported figure, we find values $\mathbf{d}(m_t^{\text{data}}, m_t^{\text{net}} = 0.0013$ for the Ising model, and $\mathbf{d}(\rho_t^{\text{data}}, \rho_t^{\text{net}} = 0.0159$ in the case of the contact process.

Appendix B. Field-theoretic formulations

B.1. Kinetic Ising model

The field theory for the kinetic Ising model [42] is introduced by coarse-graining in space the originally discrete value of the spins s^i in the critical regime. Averaging over some mesoscopic spatial volume, a real-valued spin density field ψ is defined over a $(d+1)$-dimensional continuous space-time. Specifically, the configurations Σ_t have been coarse-grained so that at each point, a density $\psi \in \mathbf{R}$ is defined. Then the stochastic time evolution for the density ψ is provided by

$$\partial_t \psi(\mathbf{x}, t) = -\frac{\delta \mathcal{F}[\psi]}{\delta \psi(\mathbf{x}, t)} + \eta(\mathbf{x}, t), \tag{B.1}$$

with an effective potential functional of the form

$$\mathcal{F}[\psi] = \Gamma_0 \int \mathrm{d}^d\boldsymbol{x} |\nabla\psi(\boldsymbol{x},t)|^2 + u_0\psi^2(\boldsymbol{x},t) + r_0\psi^4(\boldsymbol{x},t), \tag{B.2}$$

and a Gaussian noise with correlations

$$\langle \eta(\boldsymbol{x}_1,t_1)\eta(\boldsymbol{x}_2,t_2)\rangle = 2\Gamma_0\delta(\boldsymbol{x}_1-\boldsymbol{x}_2)\delta(t_1-t_2). \tag{B.3}$$

The diffusion constant Γ_0 and the coupling constants in (B.2) are functions of the model parameters.

B.2. Contact process

For the field-theoretic formulation of the contact process, the average over some mesoscopic box in the lattice defines a coarse-grained density field $\rho(\mathbf{x},t)$ (instead of taking the average over the whole lattice $\bar{\rho}$). The Langevin equation for its time evolution can be derived directly from the master equation of the contact process and reads [54, 60]

$$\partial_t\rho(\mathbf{x},t) = D\nabla^2\rho(\mathbf{x},t) + \iota\rho(\mathbf{x},t) - \lambda\rho^2(\mathbf{x},t) + \zeta(\mathbf{x},t). \tag{B.4}$$

The noise $\zeta(\mathbf{x},t)$ exhibits a multiplicative form

$$\langle \zeta(\boldsymbol{x}_1,t_1)\zeta(\boldsymbol{x}_2,t_2)\rangle = \Gamma\rho(\boldsymbol{x}_1,t_1)\delta(\boldsymbol{x}_1-\boldsymbol{x}_2)\delta(t_1-t_2). \tag{B.5}$$

D is the diffusive constant, while the coupling constants ι, λ, and Γ are functions of the lattice details and of the infection rate. The occurrence of a term proportional to ρ and one proportional to ρ^2 in equation (B.4) can be explained heuristically from the mean-field treatment of the transition rates in equation (17). The number of sites becoming inactive at time t is $\sum_i s_t^i \propto \rho_t$. Instead, the number of sites becoming active is given by the number of inactive sites next to an active site that can be thus be infected. This number is given by $\sum_i (s_t^i - s_t^{i+1})^2 = 2\sum_i s_t^i - 2\sum_i s_t^i s_t^{i+1} \propto 2\rho_t - 2\rho_t^2$. In the mean-field treatment, the master equation thus reads $\partial_t\rho_t = (\kappa-1)\rho_t - \kappa\rho_t^2$. The form of the noise proportional to $\sqrt{\rho}$, can be justified by observing that only active sites contribute to the density fluctuations. To see this, let N be the total number of sites, and let n be the number of active sites at time t. The density of active sites at time t is thus $\rho_t = n/N$. Now, let X_i be the number of active sites at time $t' > t$ whose infection can be traced back to the ith active site at time t. Notice that the sequence $\{X_1,\dots,X_n\}$ is formed by independent identically distributed random variables, and $\rho_{t'} = 1/N\sum_i X_i$. Their expectation value and variance will thus be independent of the site i: $\mathbb{E}[X_i] = \nu$, $\mathrm{Var}[X_i] = \zeta^2$ for some real number ν and ζ. The relation between ν, ζ^2 and the sample average $\bar{X}_n = 1/n\sum_i X_i \equiv \rho_{t'}/\rho_t$ is described by the central limit theorem. This theorem states that for large n, the probability distribution p of the random variable $\sqrt{n}(\bar{X}_n - \nu)$ converges to a normal distribution centered around zero and with variance ζ^2, $\mathcal{N}(0,\zeta^2)$:

$$p\left(\sqrt{n}(\bar{X}_n - \nu)\right)\mathcal{N}\left(0,\zeta^2\right). \tag{B.6}$$

Note that for ρ_t to be finite also N must be large. Substituting $\sqrt{n} \to N\rho_t$ and $\bar{X}_n \to \rho_{t'}/\rho_t$, one obtains

$$p\left(\sqrt{N}(\rho_{t'} - \rho_t\nu)\right) \to \sqrt{\rho_t}\mathcal{N}\left(0,\zeta^2\right), \tag{B.7}$$

which means that the expectation value of $\rho_{t'}$ is $\rho_t\nu$, and its variance $\rho_t\zeta^2$.

ORCID iDs

Francesco Carnazza https://orcid.org/0000-0002-1458-8701
Federico Carollo https://orcid.org/0000-0002-6961-7143
Sabine Andergassen https://orcid.org/0000-0002-3128-6350
Georg Martius https://orcid.org/0000-0002-8963-7627
Miriam Klopotek https://orcid.org/0000-0001-9174-1282
Igor Lesanovsky https://orcid.org/0000-0001-9660-9467

References

[1] Van Kampen N 2007 *Stochastic Processes in Physics and Chemistry* 3rd edn (Elsevier)
[2] Coffey W T, Kalmykov Y P and Waldron J T 1996 *The Langevin equation* (World Scientific)
[3] Gillespie D 1992 *Markov Processes* (Academic)
[4] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087–92
[5] Hohenberg P and Kohn W 1964 *Phys. Rev.* **136** B864–71
[6] Rosenbluth M N and Rosenbluth A W 2004 *J. Chem. Phys.* **23** 356–9
[7] Widder C, Koch F and Schilling T 2022 *J. Chem. Phys.* **157** 194107
[8] Schilling T 2022 *Phys. Rep.* **972** 1–45
[9] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[10] Mehta P, Bukov M, Wang C H, Day A G, Richardson C, Fisher C K and Schwab D J 2019 *Phys. Rep.* **810** 1–124
[11] Grogan F, Lei H, Li X and Baker N A 2020 *J. Comput. Phys.* **418** 109633
[12] Yang L, Gao T, Lu Y, Duan J and Liu T 2023 *Appl. Math. Model.* **115** 279–99
[13] Oh Y, Lim D and Kim S 2024 Stable neural stochastic differential equations in analyzing irregular time series data *12th Int. Conf. Learning Representations*
[14] Bai Y, Yan B, Zhou C, Su T and Jin X 2023 *Annu. Rev. Control* **56** 100909
[15] Glauber R J 1963 *J. Math. Phys.* **4** 294–307
[16] Süzen M 2014 *Phys. Rev.* E **90** 032141
[17] Walter J C and Barkema G 2015 *Physica* A **418** 78–87
[18] Majumdar S N 1999 *Curr. Sci.* **77** 370–5
[19] Callen H B 1985 *Thermodynamics and an Introduction to Thermostatistics* 2nd edn (Wiley)
[20] Halpin-Healy T 1998 *Phys. Rev.* E **58** R4096–9
[21] Halpin-Healy T 1991 *Phys. Rev.* A **44** R3415–8
[22] Xu J, Dupont E, Märtens K, Rainforth T and Teh Y W 2024 Deep stochastic processes via functional Markov transition operators *Proc. 37th Int. Conf. Neural Information Processing Systems* (*NIPS '23* Red Hook, NY, USA) (Curran Associates Inc.)
[23] Garnelo M, Rosenbaum D, Maddison C J, Ramalho T, Saxton D, Shanahan W, Teh Y W, Rezende D J and Eslami S M A 2018 Conditional neural processes (arXiv:1807.01613)
[24] Capobianco E 2008 *Physica* A **387** 4077–98
[25] Xu W, Chen R T Q, Li X and Duvenaud D 2022 Infinitely deep bayesian neural networks with stochastic differential equations *Proc. 25th Int. Conf. Artificial Intelligence and Statistics* (*Proc. Machine Learning Research*) vol 151, ed G Camps-Valls, F J R Ruiz and I Valera (PMLR) pp 721–38
[26] García C A, Félix P, Presedo J M R and Otero A 2022 *J. Comput. Phys.* **454** 110970
[27] Fernández-Fernández G, Manzo C, Lewenstein M, Dauphin A and Muñoz Gil G 2024 *Phys. Rev.* E **110** L012102
[28] Morrill J, Salvi C, Kidger P and Foster J 2021 Neural rough differential equations for long time series *Proc. 38th Int. Conf. Machine Learning, ICML (18–24 July 2021) Virtual Event* (*Proc. Machine Learning Research*) vol 139, ed M Meila and T Zhang (PMLR) pp 7829–38
[29] Kidger P, Morrill J, Foster J and Lyons T 2020 Neural controlled differential equations for irregular time series *Advances in Neural Information Processing Systems* vol 33, ed H Larochelle, M Ranzato, R Hadsell, M Balcan and H Lin (Curran Associates, Inc.) pp 6696–707
[30] Chen R T Q, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* vol 31, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.)
[31] Toth P, Rezende D J, Jaegle A, Racanière S, Botev A and Higgins I 2020 Hamiltonian generative networks *Int. Conf. Learning Representations*
[32] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* vol 27, ed Z Ghahramani, M Welling, C Cortes, N Lawrence and K Weinberger (Curran Associates, Inc.)
[33] Kidger P, Foster J, Li X and Lyons T 2021 Efficient and accurate gradients for neural sdes *Advances in Neural Information Processing Systems*
[34] Dietrich F, Makeev A, Kevrekidis G, Evangelou N, Bertalan T, Reich S and Kevrekidis I G 2023 *Chaos* **33** 023121
[35] Clarke J B, Hastie J W, Kihlborg L H E, Metselaar R and Thackeray M M 1994 *Pure Appl. Chem.* **66** 577–94
[36] Pavliotis G A 2016 *Stochastic Processes and Applications* (Springer)
[37] Dynkin E B, Fabius J and Greenberg V 1965 Markov processes *(Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen, Band 121)* vol I (Springer)
[38] Øksendal B 2014 *Stochastic Differential Equations: an Introduction with Applications (Universitext)* 6th edn (Springer)
[39] Protter P 1990 *Stochastic Integration and Differential Equations* (Springer)
[40] Karandikar R L and Rao B V 2014 *Proc. Math. Sci.* **124** 457–69
[41] Onsager L 1944 *Phys. Rev.* **65** 117–49
[42] Hohenberg P C and Halperin B I 1977 *Rev. Mod. Phys.* **49** 435–79
[43] Debye P 1965 *Phys. Rev. Lett.* **14** 783–4
[44] Kawasaki K 1966 *Phys. Rev.* **145** 224–30
[45] Coniglio A and Zannetti M 1989 *Europhys. Lett.* **10** 575
[46] Majumdar S N, Bray A J, Cornell S J and Sire C 1996 *Phys. Rev. Lett.* **77** 3704–7
[47] Harris T E 1974 *Ann. Probab.* **2** 969–88
[48] Durrett R 1984 *Ann. Probab.* **12** 999–1040
[49] Broadbent S R and Hammersley J M 1957 *Math. Proc. Camb. Philos. Soc.* **53** 629–41
[50] Ódor G 2004 *Rev. Mod. Phys.* **76** 663–724
[51] Domany E and Kinzel W 1984 *Phys. Rev. Lett.* **53** 311–4
[52] Hinrichsen H, Jiménez-Dalmaroni A, Rozov Y and Domany E 1999 *Phys. Rev. Lett.* **83** 4999–5002
[53] Liggett T M 1985 *Interacting Particle Systems* (Springer)
[54] Hinrichsen H 2000 *Adv. Phys.* **49** 815–958
[55] Dickman R and Jensen I 1991 *Phys. Rev. Lett.* **67** 2391–4

[56] Dickman R and da Silva J K L 1998 *Phys. Rev.* E **58** 4266–70
[57] Jensen I and Dickman R 1993 *J. Stat. Phys.* **71** 89
[58] Muñoz M A, Grinstein G and Tu Y 1997 *Phys. Rev.* E **56** 5101–5
[59] Jensen I 1999 *J. Phys.* A **32** 5233–49
[60] Janssen H K 1981 *Z. Phys.* B **42** 151–4
[61] Cardy J L and Sugar R L 1980 *J. Phys.* A **13** L423
[62] 2023 (available at: https://github.com/fcarnazza/sde_order_params)
[63] Cybenko G 1989 *Math. Control Signals Syst.* **2** 303–14
[64] Rosenblatt F 1958 *Psychol. Rev.* **65** 386–408
[65] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library *Advances in Neural Information Processing Systems vol 32*, ed H ed Wallach, H Larochelle, A Beygelzimer, B F d'Alché, E Fox and R Garnett (Curran Associates, Inc.) pp 8024–35
[66] Feinman R 2021 Pytorch-minimize: a library for numerical optimization with autograd (available at: https://github.com/rfeinman/pytorch-minimize)
[67] Kidger P, Foster J, Li X and Lyons T J 2021 Neural sdes as infinite-dimensional gans *Proc. 38th Int. Conf. Machine Learning* (*Proc. Machine Learning Research*) vol 139, ed M Meila and T Zhang (PMLR) pp 5453–63

5 Conclusions and outlook

There are many potential directions for future research. One significant avenue would be to incorporate time dependency into the generator we have identified. This generator would still maintain the property of complete positivity, essential for preserving the physical interpretation of a quantum state, while also extending its capabilities. Having addressed the efficiency of the algorithm in handling noisy data in [28, 99], it would now be natural to use it to model actual quantum noise in an experimental setting, as was done with a related model of open quantum systems in [39]. In the case of [20], it would be beneficial to apply the same network to data sets obtained at different values of the control parameter. By providing the value of the control parameter directly to the networks as an external input, we could reduce the amount of training data required. Allowing the networks to account for the control parameter in both the drift term and the diffusion term would enable us to potentially infer the dynamics and properties at unseen values of the control parameters. This approach would enhance the network's generalization capabilities across different control parameters, and making them less prone to overfitting.

A notable example where noise plays a crucial role is the current simulation of quantum systems using so-called Noisy Intermediate-Scale Quantum (NISQ) devices, which are the quantum computers currently available for quantum computations [100, 101]. The method developed in our work can be used to assess the noise in simulations on these quantum devices, evaluate the form of the noise, and determine the deformation of the directed force. This assessment is feasible because the gates applied on quantum devices are known.

In this study, we used only classical data as input for the network. A challenging and interesting extension to stochastic quantum data can be achieved through the simulation method for the time evolution of the quantum state described in [102, 103]. In this method, instead of evolving a single *quantum state*, an ensemble" of *wave functions* is evolved. Rather than applying a deterministic Lindblad operator to the entire density matrix of the system, each quantum wave function evolves according to a "stochastic Schrödinger equation", which includes dissipation effects through non-Hermitian and stochastic terms in an effective Hamiltonian operator, that act linearly on each of the wave functions of the ensemble. The total quantum state is then approximate by the average of the outer products of this “stochastic” wave vectors. The main advantage of such an approach is that while the density matrix scales quadratically with the number of basis elements in the Hilbert space of the system under consideration, due to its intrinsic formulation as on outer product, a much smaller number of wave-vectors may be able to capture the significant information.

In the context of approximating quantum states using ensembles of pure states, the application of machine learning tools like automatic differentiation and optimizers proves to be intriguing [104]. Specifically, when estimating quantum stationary states in dissipative dynamics, one can exploit the fact that probability amplitudes, which are the optimization target, are being directly manipulated. This allows for samples to be drawn directly from the amplitudes themselves, bypassing the need for external simulations to generate training data. Since the steady state of the Lindblad equation is characterized by a vanishing time derivative, a corresponding matrix norm can be employed as a loss function. This approach, first proposed by Weimer in his paper [105], aligns with the ensemble representation of quantum states as described in [102]. Such a method naturally preserves the positivity of the quantum state by expressing it as a sum of outer products of wave functions. These are just a few examples of how the power of readily available frameworks for automatic differentiation and neural networks can be harnessed by the physics community to derive consistent and physically interpretable quantities. We believe that, in the future, such tools will become indispensable to the working physicist.

6 Bibliography

References

[1] A. L. Fetter and J. D. Walecka, *Quantum Theory of Many-Particle Systems.* New York: Dover., 2003.

[2] D. J. Thouless, *The quantum mechanics of many-body systems.* New York: Academic Press., 1972.

[3] P. Nozières, *Theory of Interacting Fermi Systems.* Addison-Wesley., 1997.

[4] P. W. Anderson, "More is different," *Science*, vol. 177, no. 4047, pp. 393–396, 1972.

[5] S. Kivelson and S. Kivelson, "Defining emergence in physics.," *npj Quantum Mater.*, vol. 1, p. 16024, 2016.

[6] E. Fradkin, *Field Theories of Condensed Matter Physics*. Cambridge University Press, 2 ed., 2013.

[7] M. Kardar, *Statistical Physics of Particles.* Cambridge University Press, 2007.

[8] R. K. Pathria and P. D. Beale, *Statistical Mechanics (3rd ed.).* Elsevier, 2011.

[9] J. Sakurai and J. Napolitano, *Modern Quantum Mechanics Pearson New Int. Edition*. Pearson Deutschland, 2013.

[10] D. Ceperley and B. Alder, "Quantum Monte Carlo," *Science*, vol. 231, no. 4738, pp. 555–560, 1986.

[11] E. Y. Loh, J. E. Gubernatis, R. T. Scalettar, S. R. White, D. J. Scalapino, and R. L. Sugar, "Sign problem in the numerical simulation of many-electron systems," *Phys. Rev. B*, vol. 41, pp. 9301–9307, May 1990.

[12] A. Einstein, B. Podolsky, and N. Rosen, "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?," *Phys. Rev.*, vol. 47, pp. 777–780, May 1935.

[13] J. S. Bell, "On the Einstein Podolsky Rosen paradox," *Phys. Phys. Fiz.*, vol. 1, pp. 195–200, Nov 1964.

[14] G. Vidal, J. I. Latorre, E. Rico, and A. Kitaev, "Entanglement in Quantum Critical Phenomena," *Phys. Rev. Lett.*, vol. 90, p. 227902, Jun 2003.

[15] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge Univ. Press, 2007.

[16] K. G. Wilson and J. Kogut, "The renormalization group and the ϵ expansion," *Phys. Rep.*, vol. 12, no. 2, pp. 75–199, 1974.

[17] K. Huang, *Statistical mechanics*. Wiley, 1987.

[18] B. Widom, "Surface Tension and Molecular Correlations near the Critical Point," *J. Chem. Phys.*, vol. 43, pp. 3892–3897, 12 1965.

[19] P. C. Hohenberg and B. I. Halperin, "Theory of dynamic critical phenomena," *Rev. Mod. Phys.*, vol. 49, pp. 435–479, Jul 1977.

[20] F. Carnazza, F. Carollo, S. Andergassen, G. Martius, M. Klopotek, and I. Lesanovsky, "Machine learning stochastic differential equations for the evolution of order parameters of classical many-body systems in and out of equilibrium," *arXiv 2402.03913, preprint*, 2024.

[21] M. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.

[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[23] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, 2017.

[24] A. Nagy and V. Savona, "Variational quantum monte carlo method with a neural-network ansatz for open quantum systems," *Phys. Rev. Lett.*, vol. 122, p. 250501, Jun 2019.

[25] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.

[26] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control Signals Syst.*, vol. 2, pp. 303–314, 1989.

[27] F. Carnazza, F. Carollo, D. Zietlow, S. Andergassen, G. Martius, and I. Lesanovsky, “Inferring markovian quantum master equations of few-body observables in interacting spin chains,” *New J. Phys.*, vol. 24, p. 073033, jul 2022.

[28] G. Cemin, F. Carnazza, S. Andergassen, G. Martius, F. Carollo, and I. Lesanovsky, “Inferring interpretable dynamical generators of local quantum observables from projective measurements through machine learning,” *Phys. Rev. Appl.*, vol. 21, p. L041001, Apr 2024.

[29] I. Bloch, J. Dalibard, and S. Nascimbène, “Quantum simulations with ultracold quantum gases,” *Nat. Phys.*, vol. 8, 2012.

[30] M. Morgado and S. Whitlock, “Quantum simulation and computing with Rydberg-interacting qubits,” *AVS Quantum Sci.*, vol. 3, 05 2021. 023501.

[31] X. Wu, X. Liang, Y. Tian, F. Yang, C. Chen, Y.-C. Liu, M. K. Tey, and L. You, “A concise review of Rydberg atom based quantum computation and quantum simulation*,” *Chin. Phys. B*, vol. 30, p. 020305, feb 2021.

[32] M. Saffman, T. G. Walker, and K. Molmer, “Quantum information with rydberg atoms,” *Rev. Mod. Phys.*, vol. 82, p. 2313, 2010.

[33] N. Yoshioka and R. Hamazaki, “Constructing neural stationary states for open quantum many-body systems,” *Phys. Rev. B*, vol. 99, p. 214306, Jun 2019.

[34] M. J. Hartmann and G. Carleo, “Neural-Network Approach to Dissipative Quantum Many-Body Dynamics,” *Phys. Rev. Lett.*, vol. 122, p. 250502, Jun 2019.

[35] P. P. Mazza, D. Zietlow, F. Carollo, S. Andergassen, G. Martius, and I. Lesanovsky, “Machine learning time-local generators of open quantum dynamics,” *Phys. Rev. Research*, vol. 3, p. 023084, Apr 2021.

[36] S. Krastanov, K. Head-Marsden, S. Zhou, S. T. Flammia, L. Jiang, and P. Narang, “Unboxing quantum black box models: Learning non-markovian dynamics,” *arXiv:2009.03902*, 2020.

[37] C. Miles, A. Bohrdt, R. Wu, C. Chiu, M. Xu, G. Ji, M. Greiner, K. Q. Weinberger, E. Demler, and E.-A. Kim, “Correlator convolutional neural networks as an inter-

pretable architecture for image-like quantum matter data," *Nat. Commun.*, vol. 12, 6 2021.

[38] S. Martina, S. Gherardini, and F. Caruso, "Machine learning classification of non-markovian noise disturbing quantum dynamics," *Phys. Scr.*, vol. 98, p. 035104, Feb. 2023.

[39] G. Cemin, M. Cech, E. Weiss, S. Soltan, D. Braun, I. Lesanovsky, and F. Carollo, "Machine learning of quantum channels on NISQ devices," *arXiv 2405.12598, preprint*, 2024.

[40] G. Vidal, "Efficient Classical Simulation of Slightly Entangled Quantum Computations," *Phys. Rev. Lett.*, vol. 91, p. 147902, Oct 2003.

[41] F. Verstraete, J. J. García-Ripoll, and J. I. Cirac, "Matrix Product Density Operators: Simulation of Finite-Temperature and Dissipative Systems," *Phys. Rev. Lett.*, vol. 93, p. 207204, Nov 2004.

[42] M. B. Hastings, "An area law for one-dimensional quantum systems," *J. Stat. Mech.: Theory Exp.*, vol. 2007, p. P08024, aug 2007.

[43] P. Kidger, J. Foster, X. Li, and T. Lyons, "Efficient and accurate gradients for neural sdes," in *Adv. NeurIPS*, 2021.

[44] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Adv. NeurIPS*, vol. 31, pp. 6571–6583, Curran Associates, Inc., 2018.

[45] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," in *Adv. NeurIPS* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 6696–6707, Curran Associates, Inc., 2020.

[46] J. Morrill, C. Salvi, P. Kidger, and J. Foster, "Neural rough differential equations for long time series," in *Proc. 38th ICLR, ICML* (M. Meila and T. Zhang, eds.), vol. 139 of *PMLR*, pp. 7829–7838, PMLR, 2021.

[47] R. J. Glauber, "Time-Dependent Statistics of the Ising Model," *J. Math. Phys.*, vol. 4, pp. 294–307, 12 1963.

[48] H. Hinrichsen, A. Jiménez-Dalmaroni, Y. Rozov, and E. Domany, "Flowing Sand: A Physical Realization of Directed Percolation," *Phys. Rev. Lett.*, vol. 83, pp. 4999–5002, Dec 1999.

[49] N. Van Kampen, *Stochastic Processes in Physics and Chemistry (Third Edition)*. Elsevier, 2007.

[50] W. T. Coffey, Y. P. Kalmykov, and J. T. Waldron, *The Langevin Equation*. Singapore: WORLD SCIENTIFIC, 1996.

[51] D. Gillespie, *Markov Processes*. Academic Press, San Diego, 1992.

[52] A. Einstein, "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen," *Ann. Phys. (Berl.)*, vol. 322, no. 8, pp. 549–560, 1905.

[53] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, 1953.

[54] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, no. 3B, p. B864–B871, 1964. Cited by: 41845; All Open Access, Bronze Open Access.

[55] M. N. Rosenbluth and A. W. Rosenbluth, "Monte Carlo Calculation of the Average Extension of Molecular Chains," *J. Chem. Phys.*, vol. 23, pp. 356–359, 12 2004.

[56] N. Wiener, "Differential-Space," *J. Math. Phys.*, vol. 2, no. 1-4, pp. 131–174, 1923.

[57] J. L. Doob, "The Brownian Movement and Stochastic Equations," *Ann. Math.*, vol. 43, no. 2, pp. 351–369, 1942.

[58] G. E. Uhlenbeck and L. S. Ornstein, "On the Theory of the Brownian Motion," *Phys. Rev.*, vol. 36, pp. 823–841, Sep 1930.

[59] G. Franceschetti and D. Riccio, "CHAPTER 3 - Surface Fractal Models," in *Scattering, Natural Surfaces, and Fractals* (G. Franceschetti and D. Riccio, eds.), pp. 61–113, Burlington: Academic Press, 2007.

[60] M. Planck, "Über einen satz der statistischen dynamik und seine erweiterung in der quantentheorie"," *Sitzungsber. Kgl. Preuss. Akad. Wiss.*, vol. 24, p. 324–341, 1917.

[61] M. von Smoluchowski, "Zur kinetischen Theorie der Brownschen Molekularbewegung und der Suspensionen," *Ann. Phys. (Berl.)*, vol. 326, no. 14, pp. 756–780, 1906.

[62] P. Langevin, "Sur la théorie du mouvement brownien," *C. R. Acad. Sci.*, vol. 156, no. 530, 1908.

[63] G. A. Pavliotis, *Stochastic processes and applications*. Berlin, Springer, 2016.

[64] E. B. Dynkin, J. Fabius, and V. Greenberg, *Markov Processes. Volume I. (Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen, Band 121)*. Berlin, Springer Verlag, 1965.

[65] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, 6th ed., Jan. 2014.

[66] K. Itô, "Stochastic integral," *Proc. Imp. Acad.*, vol. 20, no. 8, pp. 519 – 524, 1944.

[67] H. M. Wiseman and G. J. Milburn, *Quantum Measurement and Control*. Cambridge University Press, 2009.

[68] H.-P. Breuer, F. Petruccione, *et al.*, *The theory of open quantum systems*. Oxford University Press on Demand, 2002.

[69] H. Carmichael, *Statistical Methods in Quantum Optics 1: Master Equations and Fokker-Planck Equations*. Theoretical and Mathematical Physics, Springer Berlin Heidelberg, 2013.

[70] F. Benatti and R. Floreanini, "Open quantum dynamics: Complete positivity and entanglement," *Int. J. Mod. Phys. B*, vol. 19, no. 19, pp. 3063–3139, 2005.

[71] M.-D. Choi, "Completely positive linear maps on complex matrices," *Linear Algebra Its Appl.*, vol. 10, no. 3, pp. 285–290, 1975.

[72] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, “Completely positive dynamical semigroups of N-level systems,” *J. Math. Phys.*, vol. 17, no. 5, pp. 821–825, 1976.

[73] G. Lindblad, “On the generators of quantum dynamical semigroups,” *Commun. Math. Phys.*, vol. 48, no. 2, pp. 119–130, 1976.

[74] A. Redfield, “The theory of relaxation processes,” in *Advances in Magnetic Resonance* (H. S. Waugh, ed.), vol. 1 of *Advances in Magnetic and Optical Resonance*, pp. 1–32, Academic Press, 1965.

[75] W. Rudin, *Fourier analysis on groups*. Wiley-Interscience, Hoboken, 1990.

[76] M. S. Byrd and N. Khaneja, “Characterization of the positivity of the density matrix in terms of the coherence vector representation,” *Phys. Rev. A*, vol. 68, Dec 2003.

[77] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[78] E. L. F. A. Charnes and P. L. Yu, “The equivalence of generalized least squares and maximum likelihood estimates in the exponential family,” *J. Am. Stat. Assoc.*, vol. 71, no. 353, pp. 169–171, 1976.

[79] M. Motta, C. Genovese, F. Ma, Z.-H. Cui, R. Sawaya, G. K. Chan, N. Chepiga, P. Helms, C. Jiménez-Hoyos, A. J. Millis, U. Ray, E. Ronca, H. Shi, S. Sorella, E. M. Stoudenmire, S. R. White, and S. Zhang, “Ground-state properties of the hydrogen chain: Dimerization, insulator-to-metal transition, and magnetic phases,” *Phys. Rev. X*, vol. 10, p. 031058, Sep 2020.

[80] A. W. Sandvik and G. Vidal, “Variational Quantum Monte Carlo Simulations with Tensor-Network States,” *Phys. Rev. Lett.*, vol. 99, p. 220602, Nov 2007.

[81] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.,” *Psychol. Rev.*, vol. 65 6, pp. 386–408, 1958.

[82] A. Hodgkin and A. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve.,” *J Physiol.*, vol. 117, pp. 500–44, 1952.

[83] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Adv. NeurIPS* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[84] K. Yao, G. Zweig, M.-Y. Hwang, Y. Shi, and D. Yu, "Recurrent neural networks for language understanding," in *INTERSPEECH 2013, 14th Ann. Conf. ISCA, Lyon, France* (F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, L. Lamel, F. Pellegrino, and P. Perrier, eds.), pp. 2524–2528, ISCA, 2013.

[85] A. Kratsios and L. Papon, "Universal approximation theorems for differentiable geometric deep learning," *J. Mach. Learn. Res.*, vol. 23, no. 196, pp. 1–73, 2022.

[86] S. Park, C. Yun, J. Lee, and J. Shin, "Minimum width for universal approximation," in *ICLR*, 2021.

[87] M. Horodynski, D. Bouchet, M. Kühmayer, and S. Rotter, "Invariance property of the fisher information in scattering media," *Phys. Rev. Lett.*, vol. 127, p. 233201, Dec 2021.

[88] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 – 444, 2015.

[89] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, p. 78–87, oct 2012.

[90] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[91] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Proc. 27th Int. NeurIPS - Vol. 2*, NIPS'14, (Cambridge, MA, USA), p. 2933–2941, MIT Press, 2014.

[92] R. Fletcher, *Practical Methods of Optimization (2nd ed.)*. New York: H. Wiley & Sons, 1987.

[93] C. G. Broyden, "The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations," *IMA J. Appl. Math.*, vol. 6, pp. 76–90, 03 1970.

[94] S. Linnainmaa, "Taylor expansion of the accumulated rounding error," *BIT Numer. Math.*, vol. 16, 1976.

[95] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, vol. 22, pp. 400–407, 1950.

[96] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2017.

[97] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[98] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.

[99] G. Cemin, *Machine learning physically consistanet dynamicsl generators of local quantum observables*. Master of science thesis, University of Tübingen, University of Trento, 2024.

[100] H. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[101] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Mod. Phys.*, vol. 95, p. 045005, Dec 2023.

[102] N. Gisin and I. C. Percival, "The quantum-state diffusion model applied to open systems," *J. Phys. A*, vol. 25, p. 5677, nov 1992.

[103] N. Gisin and I. C. Percival, "The quantum state diffusion picture of physical processes," *J. Phys. A*, vol. 26, p. 2245, may 1993.

[104] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, "Variational neural-network ansatz for steady states in open quantum systems," *Phys. Rev. Lett.*, vol. 122, p. 250503, Jun 2019.

[105] H. Weimer, "Variational Principle for Steady States of Dissipative Quantum Many-Body Systems," *Phys. Rev. Lett.*, vol. 114, p. 040402, Jan 2015.