PAPER • OPEN ACCESS

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

To cite this article: Francesco Carnazza *et al* 2022 *New J. Phys.* **24** 073033

View the article online for updates and enhancements.

New Journal of Physics

The open access journal at the forefront of physics

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.

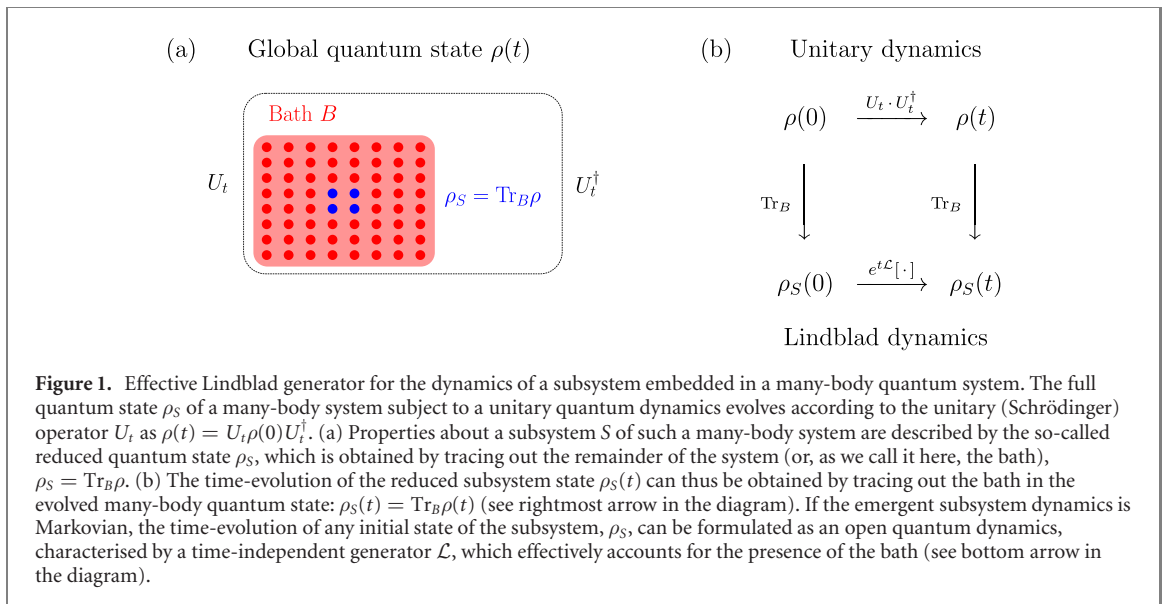E-mail: francesco.carnazza@uni-tuebingen.de

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
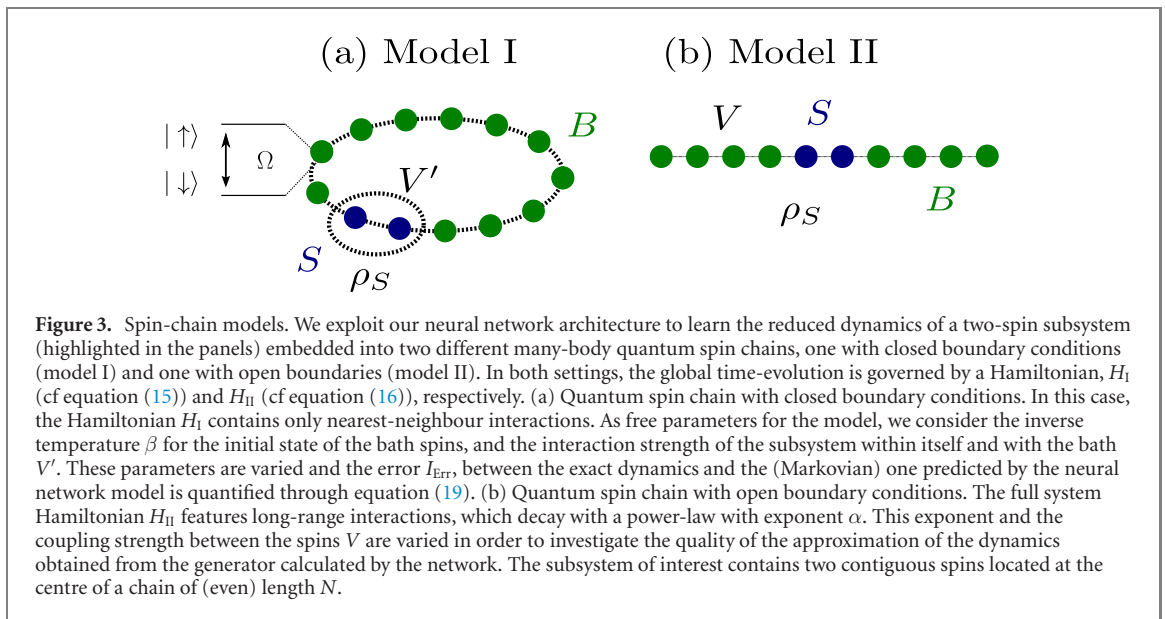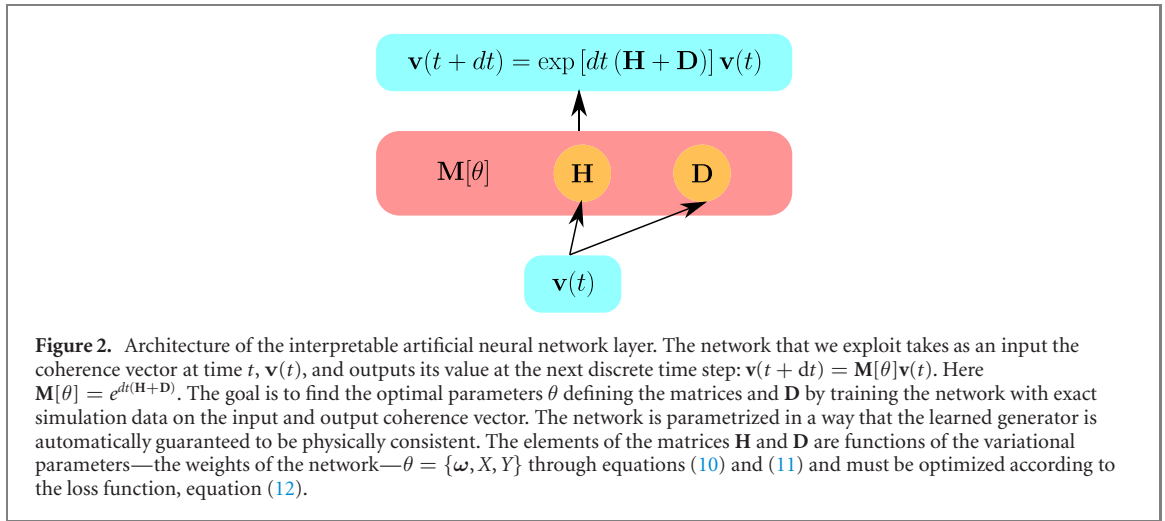
Figure 1. Effective Lindblad generator for the dynamics of a subsystem embedded in a many-body quantum system. The full quantum state ρ_S of a many-body system subject to a unitary quantum dynamics evolves according to the unitary (Schrödinger) operator U_t as $\rho(t) = U_t\rho(0)U_t^\dagger$. (a) Properties 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), $\rho_S = \text{Tr}_B\rho$. (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: $\rho_S(t) = \text{Tr}_B\rho(t)$ (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))

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

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.

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: $\mathbf{v}(t+dt) = \mathbf{M}[\theta]\mathbf{v}(t)$. 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—$\theta = \{\omega, X, Y\}$ 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 $U_t = e^{-itH_{S+B}}$.

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 $i = 1, 2, ..., d^2$ 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 $\mathrm{Tr}(F_i F_j) = \delta_{ij}$, 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 $\mathcal{L}$, $\mathcal{T}_t = e^{\mathcal{L}t}$, 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: $h^\dagger c h = \mathrm{diag}(\gamma_1, \ldots, \gamma_{d^2-1})$, 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

$$\mathbf{H}_{ij} = -4 \sum_{k=1}^{d^2-1} f_{ijk}\omega_k, \quad i,j \in \{1, 2, \ldots, d^2-1\},$$

$$\mathbf{H}_{id^2} = \mathbf{H}_{d^2 i} = 0, \quad i \in \{1, 2, \ldots, d^2\}, \tag{10}$$

where ω defines the expansion of the Hamiltonian matrix H in equation (3) over the orthonormal set $\{F_i\}_{i=1}^{d^2}$: $H = \sum_{i=1}^{d^2-1} \omega_i F_i$. 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)

$$\mathbf{D}_{mn} = -8 \sum_{i,j,k=1}^{d^2-1} \left(f_{mik}f_{njk}\,\mathrm{Re}(c)_{ij} + f_{mik}d_{njk}\,\mathrm{Im}(c)_{ij} \right), \quad m,n \in \{1, 2, \ldots, d^2-1\},$$

$$\mathbf{D}_{md^2} = -4 \sum_{i,j=1}^{d^2-1} f_{imj}\,\mathrm{Im}(c)_{ij}, \qquad \mathbf{D}_{d^2 m} = 0, \quad m \in \{1, 2, \ldots, d^2\}, \tag{11}$$

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 ω 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 $\theta = \{\omega, X, Y\}$, 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 $\theta = \{\omega, X, Y\}$ to be time independent. Moreover, only the symmetric real matrix $\mathrm{Re}(c) = XX^t + YY^t$ and the skew symmetric real matrix $\mathrm{Im}(c) = X^t Y - Y^t X$ 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, $\mathbf{v}_{\text{in}} = \mathbf{v}(t)$, the network $\mathbf{M}$ (see figure 2), which is a function of parameters θ, is trained to output $\mathbf{v}_{\text{out}} = \mathbf{M}[\theta]\mathbf{v}_{\text{in}}$ such that $\mathbf{v}_{\text{out}}$ 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 $(\mathbf{v}_0, \mathbf{v}(\mathrm{d}t), ..., \mathbf{v}(T))$. 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 $\mathbf{v}_{\text{in}}$ as

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

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 $U_t = \mathrm{e}^{-itH_{S+B}}$ 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 $\rho_S(t) = \text{Tr}_B\, U_t\rho_0 U_t^{\dagger}$.

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

$$H_{\text{I}} = \frac{\Omega}{2}\sum_{i=1}^{N}\sigma_i^x + V\sum_{i=3}^{N-1}n_in_{i+1} + V'(n_Ln_1 + n_1n_2 + n_2n_3). \tag{15}$$

---

[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 $\text{Alpha} = 10^{-3}$, $\text{Beta}_1 = 0.9$, $\text{Beta}_2 = 0.999$ and $\text{Epsilon} = 10^{-8}$.

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:

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

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 $H_{II}^{(2)}$, 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, $\{F_1, F_2, \ldots, F_{d^2}\} = \{\mathbb{1}_2 \otimes \sigma^x/2, \mathbb{1}_2 \otimes \sigma^y/2, \ldots, \mathbb{1}_4/2\}$. 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 $\rho_{Bath} \propto e^{-\beta H_{I/II}}$, 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
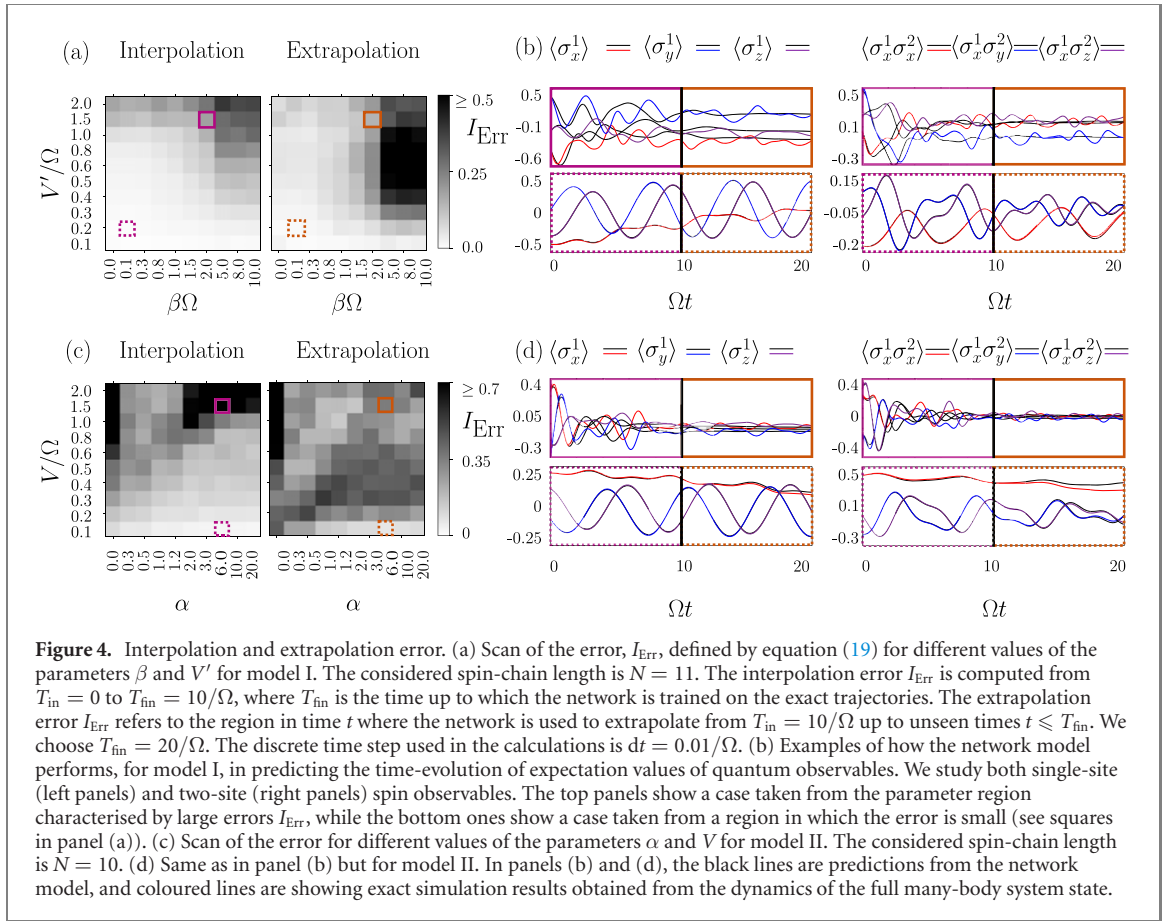
$$\rho_S(0) = \frac{(M + iN)^\dagger(M + iN)}{\mathrm{Tr}\left[(M + iN)^\dagger(M + iN)\right]}. \tag{18}$$

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

$$I_{\mathrm{Err}}(T_{\mathrm{in}}, T_{\mathrm{fin}}) = \frac{1}{T_{\mathrm{fin}} - T_{\mathrm{in}}} \int_{T_{\mathrm{in}}}^{T_{\mathrm{fin}}} \mathrm{d}t \|\rho_{\mathrm{exact}}(t) - \rho_{\mathrm{network}}(t)\|_1. \tag{19}$$
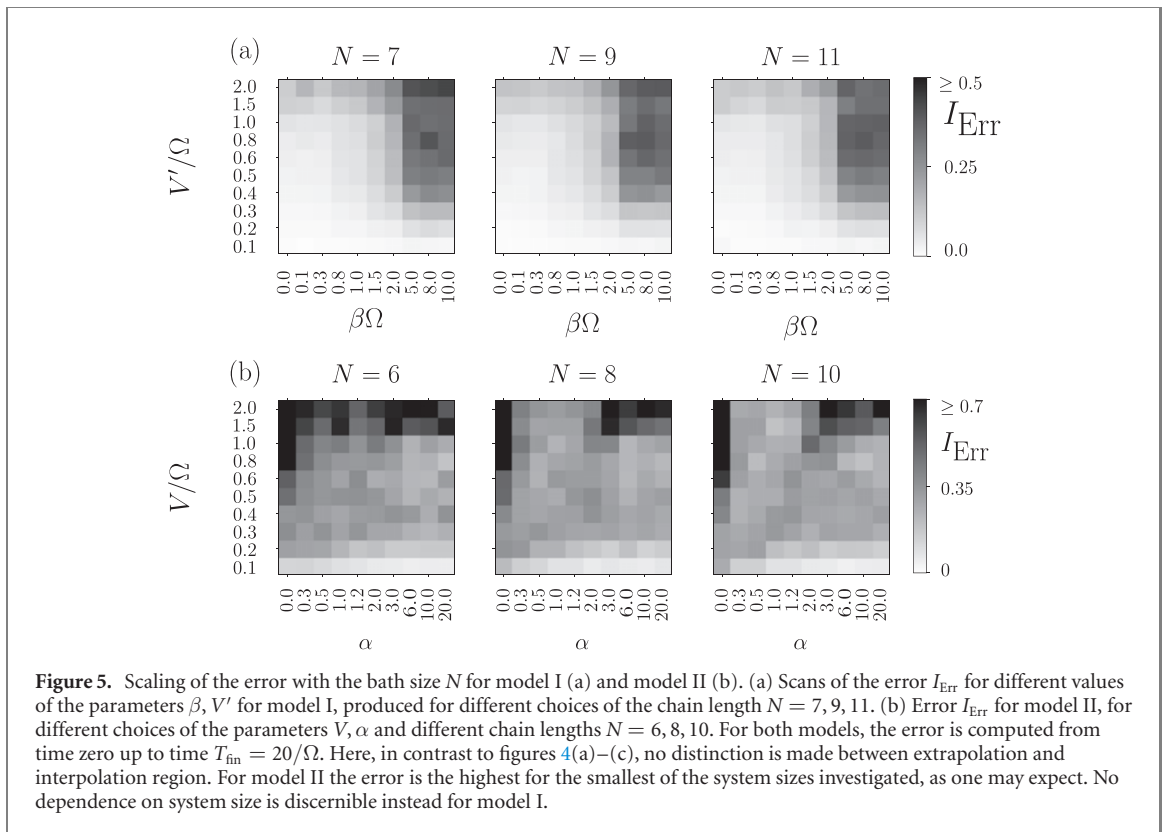
Here, ρ_{exact} is the (time-evolved) quantum state for S obtained from the exact diagonalization of the full many-body problem, while ρ_{network} is the subsystem dynamics as predicted by the network. The trace norm $\|\sigma\|_1$ for a $d \times d$ complex matrix σ, with eigenvalues $\lambda_1, \ldots, \lambda_d$ 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
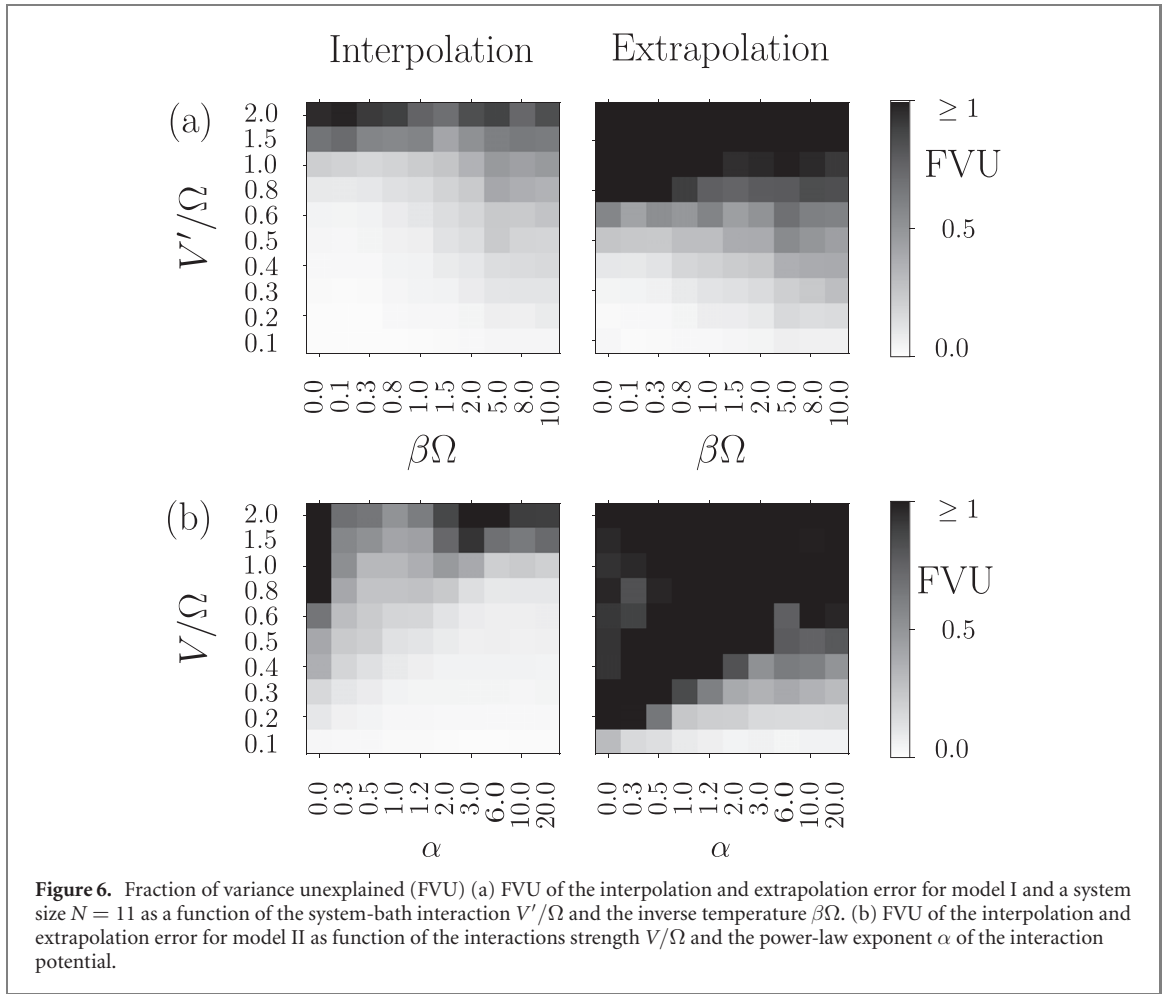
Figure 5. Scaling of the error with the bath size N for model I (a) and model II (b). (a) Scans of the error I_{Err} for different values of the parameters β, V' for model I, produced for different choices of the chain length $N = 7, 9, 11$. (b) Error I_{Err} for model II, for different choices of the parameters V, α and different chain lengths $N = 6, 8, 10$. For both models, the error is computed from time zero up to time $T_{fin} = 20/\Omega$. 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, $v_i = (v_i(t_0), v_i(t_0 + \mathrm{d}t), \ldots)$. 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 **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 **L**: $\mathbf{L}\mathbf{v}_{\text{st}} = 0 = \frac{d\mathbf{v}_{\text{st}}}{dt}$. 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 **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

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

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 dt for the exact integration and for the training of the network is chosen to be $dt = 0.01/\Omega$ as before. Once the training was complete, we could retrieve the Lindblad generator **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 $\mathbf{v}_{st}$, 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.

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
$\mathbf{v} = (v_1, \ldots, v_{d^2})$. 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

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

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 $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_{d^2-1})$ a
$d^2 - 1$ dimensional real vector. Then, the matrix elements of $\mathbf{H}$ are determined by

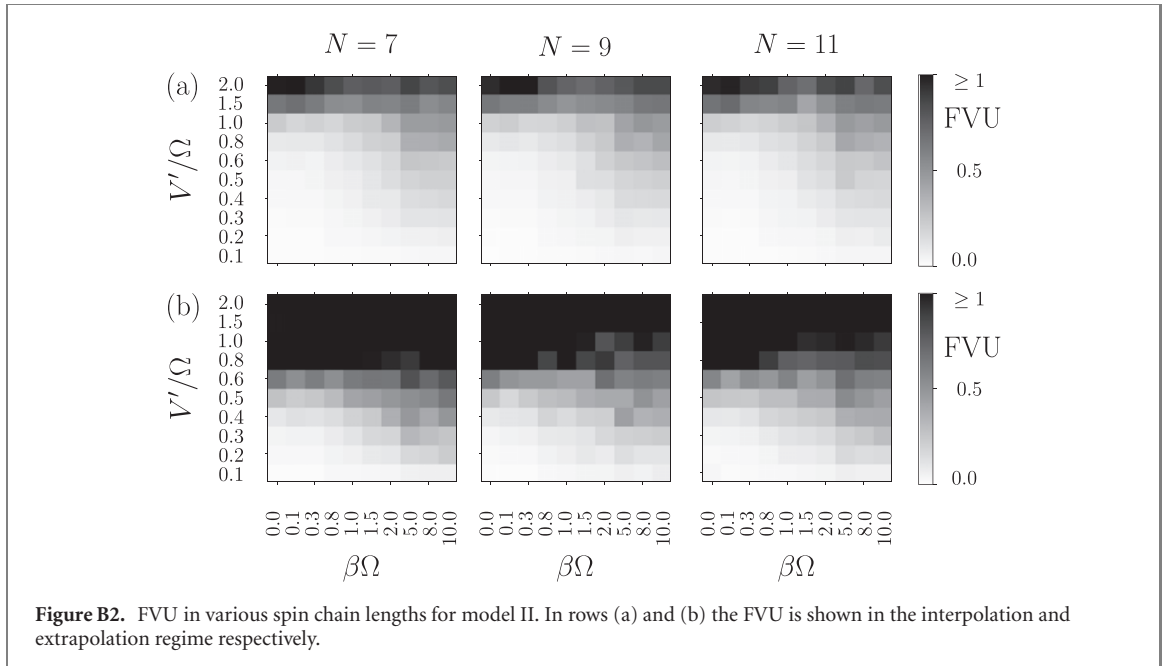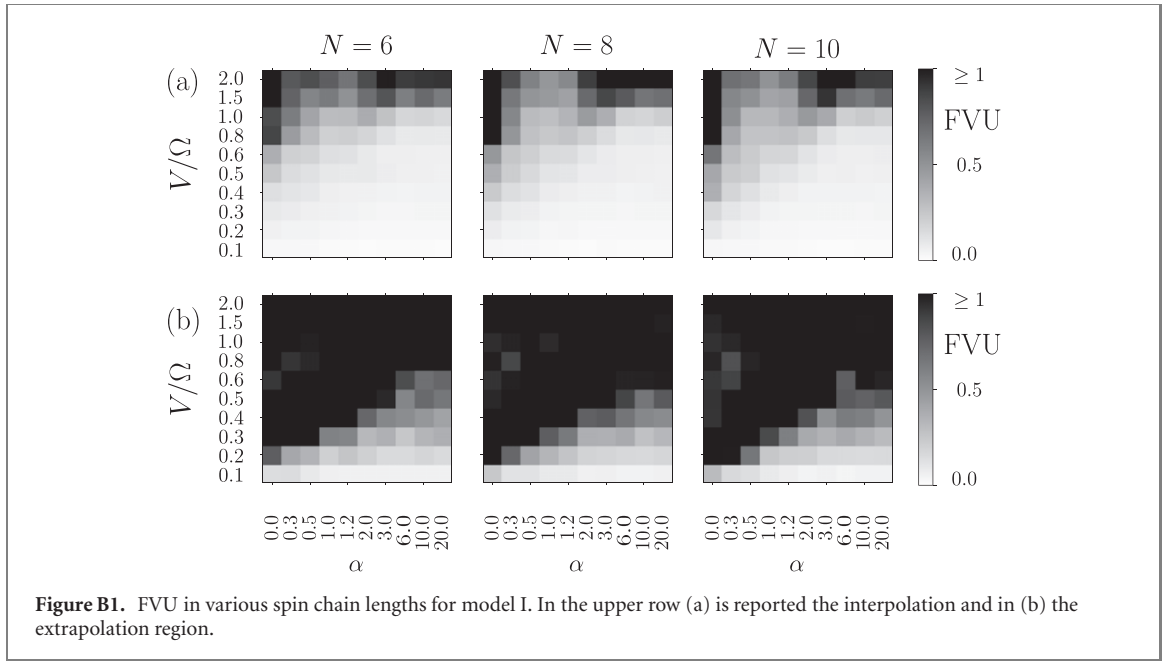$$\mathbf{H}_{mn} = i \sum_{k=1}^{d^2-1} \mathrm{Tr}([F_k, F_m] F_n) \omega_k. \tag{A.7}$$

The Hamiltonian part can thus be rewritten as

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

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

The dissipative part instead takes the form

$$\mathbf{D}_{mn} = \frac{1}{2} \sum_{i,j}^{d^2-1} c_{ij}\, \mathrm{Tr}\left([F_m, F_i] F_n F_j + [F_j, F_m] F_i F_n\right)$$

$$= -8 \sum_{i,j,k=1}^{d^2-1} (f_{mik} f_{njk}\, \mathrm{Re}(c)_{ij} + f_{mik} d_{njk}\, \mathrm{Im}(c)_{ij}) \tag{A.9}$$

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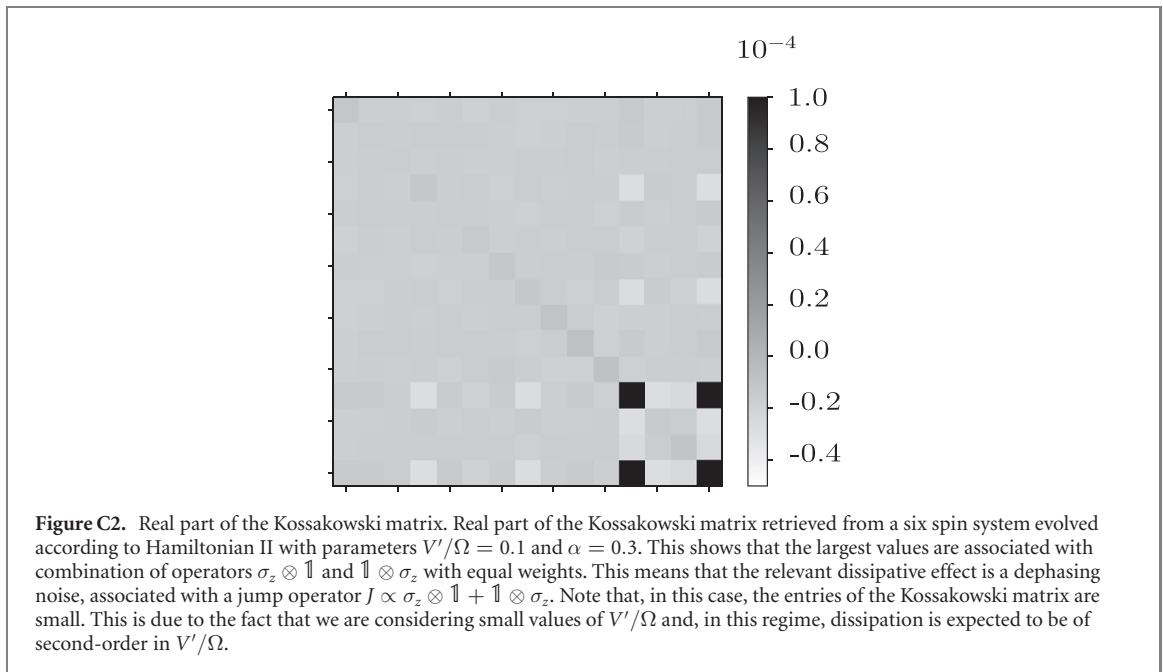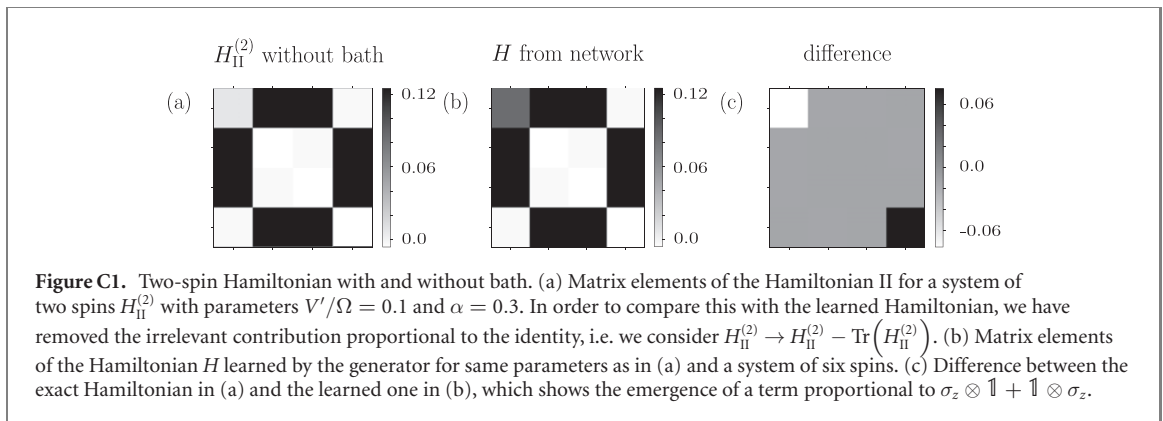
for $1 \leqslant m < d^2, 1 \leqslant n < d^2$. In the last line we used the hermiticity of the Kossakowski matrix c, i.e. that its real (imaginary) part is (skew-) symmetric. For the matrix elements of **D** with $1 \leqslant m < d^2$, $n = d^2$ one has

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

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

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