

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, 72076 Tübingen, Germany*

²*Institute for Solid State Physics and Institute of Information Systems Engineering,
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, Maria-von-Linden-Straße 6 72076 Tübingen*

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*

(Dated: April 9, 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.

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 [4–8]. Furthermore, in the framework of quantum simulation [9–17], it is very 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]. For instance, quantum process tomography with generative adversarial methods [23], neural networks [24], and recurrent neural networks [25, 26] have been developed. These approaches are very 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 Hamiltoni-

ans 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 an open quantum system (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 which is not experimentally accessible, or lack in physical interpretability (which is actually becoming highly desirable [40]).

In this work, we show how to use ML methods to infer the effective dynamical generator of a subsystem from a finite 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 by 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 which may be used to infer noise models on quantum simulators or

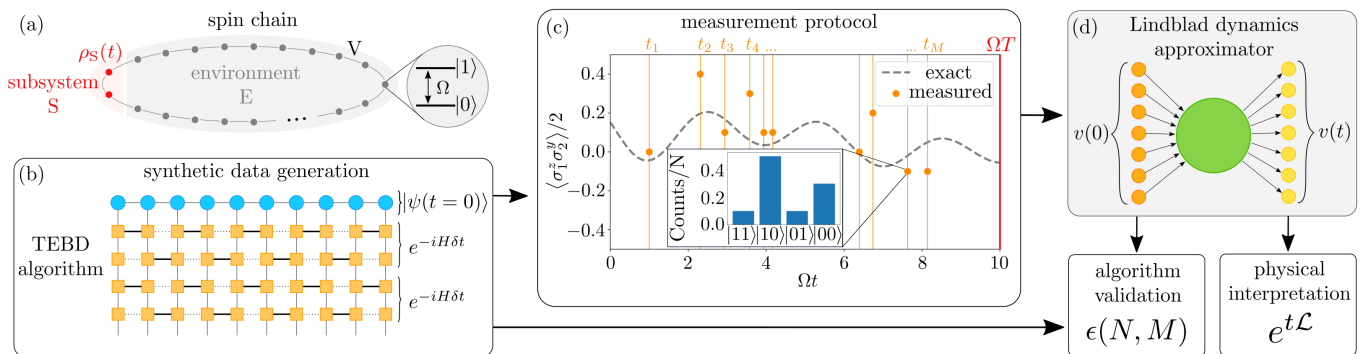


FIG. 1. **System and measurement protocol.** (a) The system is composed of the two-spin *subsystem* S, which is embedded in the *environment* E consisting of $L-2$ spins. The Hamiltonian describing the system [cf. Eq. (1)] features a driving field with Rabi frequency Ω and nearest-neighbor interactions with coupling strength V . (b) Synthetic data generation using the time-evolving block-decimation algorithm. The time evolution operator is approximated using Trotterization: $U_t = e^{-iHt} \approx \prod e^{-iH\delta t}$. The time evolution is then obtained by repeatedly applying $e^{-iH\delta t}$. (c) 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 z/y -basis for the first/second spin. The inset shows an example histogram of measurement outcomes: $|10\rangle$ means the measurement on the first spin along z -basis has outcome 1, and the measurement on the second spin along y -basis has outcome 0. From the histogram, the expectation value depicted as orange dots are obtained. (d) The synthetic data is 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.

to study thermalization dynamics in many-body systems.

Setting.— The system we consider is a 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, that is, the remainder of the spin chain. We assume the whole system to evolve unitarily, through the many-body Hamiltonian

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

The first term in the equation above describes a transverse “laser” field, while the second one accounts for nearest-neighbor interactions. Here, σ_i^α denotes the α Pauli matrix for the i^{th} spin and we have defined the projector $n = \frac{1+\sigma^z}{2}$. 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, \dots, 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, this dynamics is obtained as the partial trace of the evolution of the full many-body state,

i.e., $\rho_S(t) = \text{Tr}_E \left(U_t \rho_{S+E} U_t^\dagger \right)$, where $U_t = e^{-iHt}$, ρ_{S+E} is the initial state of the system and Tr_E denotes the trace over the environment degrees of freedom. In general, such a dynamics is rather involved and may show non-Markovian effects or it may be nonlinear for generic initial states ρ_{S+E} [45]. Here, we restrict ourselves to learning a Markovian dynamics for $\rho_S(t)$, but more general approaches are certainly possible [43]. The goal is then to identify the time-independent generator \mathcal{L} , yielding the Markovian quantum master equation evolution [45–47],

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

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

Data generation.— We simulate the time evolution of the system for times $t \in [0, \Omega T]$ by means of matrix product states and the time-evolving block-decimation 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}_{rand} distributed with the Haar measure. As the system evolves in time, after each time-step $\Omega dt = 0.01$, we calculate the expectation value of all 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. With the reduced state at hand, we can emulate experimental measurements of the above 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 basis in order to produce a noisy estimate of the reduced state. Such a procedure is sketched in Fig. 1(c), in which the histogram depicts the counting of $N = 10$ measurement outcomes for a single time-point, whereas the orange dots represent the experimental expectation value.

ML architecture and training.— The generator \mathcal{L} defining the quantum master equation (2) can be parametrized as $\mathcal{L} = \mathcal{H} + \mathcal{D}$, where

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

Here, we have introduced the Hermitian orthonormal basis $\{F_i\}_{i=1}^{16}$, for the operators of the subsystem S. Note that we choose F_1 to be proportional to the identity operator, specifically $F_1 = \mathbb{1}/\sqrt{d}$. Moreover, we write the Hamiltonian in terms of the “fields” θ_i^H , and the dissipative contribution \mathcal{D} in a non-diagonal form, fully specified by the so-called Kossakowski matrix c_{ij} . The latter must be positive semi-definite 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.

In the same spirit, we decompose the reduced state ρ_S on the basis $\{F_i\}_{i=1}^{16}$ as [51]

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

which defines the *coherence vector* $v_i = \text{Tr}(F_i \rho_S)$. Notice the condition $\text{Tr}(\rho_S) = 1$ implies $v_1 = \frac{1}{\sqrt{d}}$, that we take outside the sum. The coherence vector encodes the full information about the density matrix and provides a representation of it as a vector. This 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. Therefore, in this representation, the quantum master equation (2) becomes

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

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 (see Ref. [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 *Lindblad dynamics approximator* (LDA), as the exponential of the matrices \mathbf{H} and \mathbf{D} ,

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

that 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 $\mathbf{v}_{\text{out}} \simeq \mathbf{v}(t) = \mathbf{M}(\theta, t)\mathbf{v}_{\text{in}}$. 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.

The loss used for the optimization of the LDA parameters is the mean-squared error function (\mathbb{E}_D indicates average over the dataset)

$$\text{MSE}(\theta) = \mathbb{E}_D \left[\|\mathbf{M}(\theta, \tau)\mathbf{v}(0) - \mathbf{v}(\tau)\|^2 \right]. \quad (7)$$

We additionally consider a regularization term on the parameters penalizing nonzero elements of θ . The rationale behind this term is twofold: first, it yields a more stable training procedure, especially for small training datasets; second, it keeps the learned generator as simple as possible for enhanced interpretability. The total loss function is thus

$$\text{Loss}(\theta) = \text{MSE}(\theta) + \alpha_{11}(\|\theta^X\|_1 + \|\theta^Y\|_1) + \alpha_{12}\|\theta^H\|_1, \quad (8)$$

where $\|O\|_1 = \sum_{i,j} |O_{ij}|$. The LDA parameters θ are optimized by means of Adam optimization algorithm [54], with a scheduled learning rate, where it decays by a factor of 0.05 at 100th and 50th epochs from the final one.

After the training is completed, to test the correctness of the learned generator, we produce r new exact trajectories and compare them with our prediction. To have a quantitative measure of the performance of the ML algorithm, we compute the following error function

$$\epsilon(N, M) := \frac{1}{r} \sum_{i=1}^r \frac{1}{T} \int_0^T \frac{\|\rho_{\text{ML}}^i(t) - \rho_S^i(t)\|_2^2}{\|\rho_S^i(t)\|_2^2} dt, \quad (9)$$

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 $\|O\|_2^2 = \text{Tr}(O^\dagger O)$ [55].

Benchmarking the algorithm.— Before training the algorithm on data for the many-body model in Eq. (1),

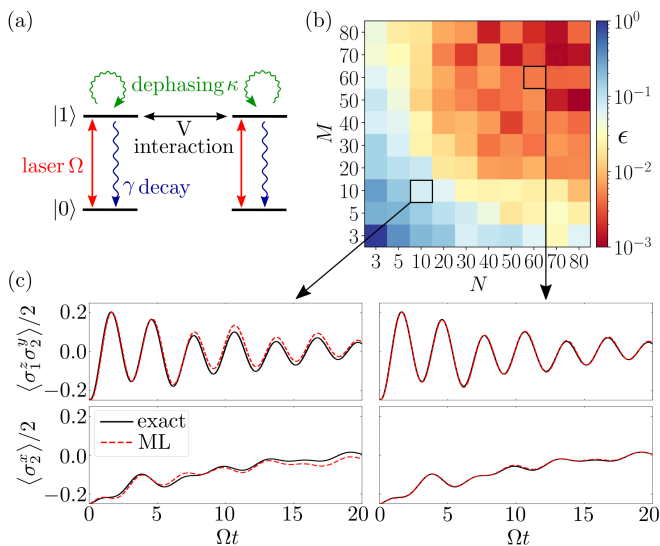


FIG. 2. **Benchmark.** (a) Open system used to benchmark the ML algorithm, described in terms of a Hamiltonian and jump operators [cf. Eq. (10)]. The Hamiltonian consists of a driving (Ω) and an interaction (V). Jump operators describe decay (with rate γ) and dephasing (with rate κ). (b) The error $\epsilon(N, M)$ as in Eq. (9) 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 main 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 is already quite reliable for smaller values of N and M . The parameters are $V = 0.5\Omega$, $\gamma = 0.01\Omega$ and $\kappa = 0.05\Omega$.

we benchmark its ability to learn a Lindblad generator within a well-controlled setting. To this end 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)]

$$\begin{aligned}
 H &= \frac{\Omega}{2}(\sigma_1^x + \sigma_2^x) + V n_1 n_2, \\
 J_1 &= \sqrt{\gamma} \sigma_1^-, & J_2 &= \sqrt{\gamma} \sigma_2^-, \\
 J_3 &= \sqrt{\kappa} n_1, & J_4 &= \sqrt{\kappa} n_2.
 \end{aligned} \tag{10}$$

The jump operators effectively describe the effects of the environment on the subsystem. In particular, 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 Fig. 2 (b-c). The color map [panel (b)] shows the error, $\epsilon(N, M)$, averaged over $r = 10$ test trajectories, for 100 different combinations

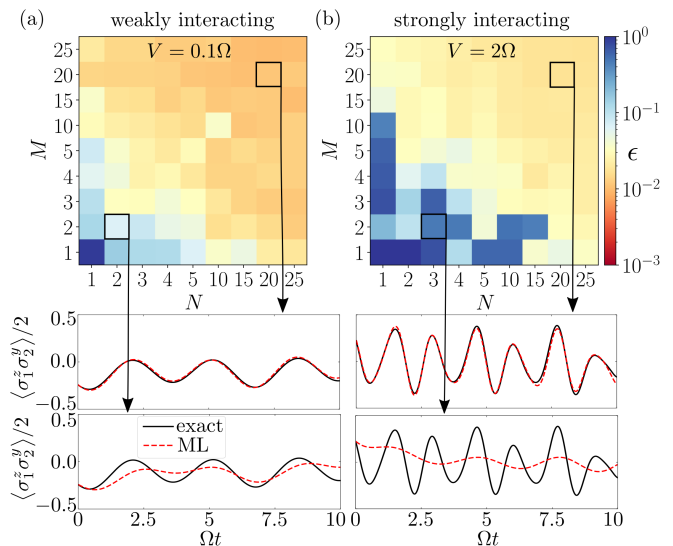


FIG. 3. **Learning the subsystem dynamics.** Error $\epsilon(N, M)$ as in Eq. (9) for different values of N and M , for two different combinations of the parameters. The error is calculated over $r = 10$ trajectories with $t \in [0, \Omega T]$. (a) Data for the case $V = 0.1\Omega$. Here, the dynamics is slow and the ML algorithm learns an effective \mathcal{L} already for small values of N and M . (b) 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 reports the time evolution of an observable for the two different models and different sets of (N, M) .

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 ML algorithm has a more stable training, and hence a better performance. This is due to the fact that the choice of the M time-points is random, and a small value of M has a high probability of yielding data which is not representative of the dynamics. On the other hand, higher values of M yields 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 $N \times M$.

Many-body setting.— Having established the capability of the ML algorithm 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 the state $\rho_S(t)$ can feature non-Markovian effects. The ML algorithm will thus, by construction, learn the “optimal” Markovian description of the system. Whether this description will accurately describe the subsystem or not depends on the relevance of

non-Markovian effects.

In Fig. 3(a), we show results for weak interactions $V = 0.1\Omega$. In this case, the ML algorithm learns a dynamical generator which 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 instead results for strong interactions, $V = 2\Omega$. Quite surprisingly, also in this case, the ML algorithm learns an effective Lindbladian which reproduces very well the subsystem dynamics. Also in this case, 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)-3(b)]. The data for the whole coherence vector are reported in the supplemental material [52], where we also show additional results on 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 [56] in this intermediate regime.

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 restricted the study to 1D systems with NN interactions. However, our ML method yields physically consistent generators regardless of the 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 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 a *bona fide* dynamics can still be learned when leaving out information about certain observables.

Acknowledgments.— We acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany’s Excellence Strategy—EXC-Number 2064/1-Project Number 390727645, 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). FC 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, *Nature Communications* **14**, 10.1038/s41467-023-39024-0 (2023).
- [15] S. Jin, N. Liu, X. Li, and Y. Yu, Quantum simulation for quantum dynamics with artificial boundary conditions (2023), [arXiv:2304.00667](https://arxiv.org/abs/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, K. R. Brown, G. Carleo, L. D. Carr, E. Demler, C. Chin, B. DeMarco, S. E. Economou, M. A. Eriksson, K.-M. C. Fu, M. Greiner, K. R. Hazzard, R. G. Hulet, A. J. Kollár, B. L. Lev, M. D. Lukin, R. Ma, X. Mi, S. Misra, C. Monroe, K. Murch, Z. Nazario, K.-K. Ni, A. C. Potter, P. Roushan, M. Saffman, M. Schleier-Smith, I. Siddiqi, R. Simmonds, M. Singh, I. Spielman, K. Temme, D. S. Weiss, J. Vučković, V. Vuletić, J. Ye, and M. Zwierlein, 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**, 10.1116/5.0036562 (2021), 023501.
- [22] V. Gebhart, R. Santagati, A. A. Gentile, E. M. Gauger, D. Craig, N. Ares, L. Bianchi, F. Marquardt, L. Pezzè, and C. Bonato, Learning quantum systems, *Nat. Rev. Phys.* **5**, 141 (2023).
- [23] P. Braccia, L. Bianchi, 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 (2022), [arXiv:2209.14328](https://arxiv.org/abs/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, 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. Bianchi, 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, Great Clarendon Street, 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, *Journal of 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 Supplemental Material at [URL] 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 (2014), [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [55] The code for the generation of the artificial data and the training of the ML algorithm is made available at https://github.com/giovannicemin/lindblad_dynamics_approximator.
- [56] 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).

SUPPLEMENTAL MATERIAL

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, 72076 Tübingen, Germany*

²*Institute for Solid State Physics and Institute of Information Systems Engineering, 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, Maria-von-Linden-Straße 6 72076 Tübingen*

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*

I. MATRIX REPRESENTATION OF \mathbf{L}

We here report a brief derivation of the explicit matrix elements appearing in eq. (5). The starting point is the coherence vector $v_i = \text{Tr}(F_i \rho_S)$. The time derivative is simply given by

$$\frac{d}{dt} v_i(t) = \text{Tr} \left(F_i \frac{d}{dt} \rho_S(t) \right) = \text{Tr}(F_i \mathcal{L}[\rho_S(t)]) = \text{Tr}(\mathcal{L}^*[F_i] \rho_S(t)) \quad (\text{S1})$$

where in the second equality we used eq. (2). In the last equality \mathcal{L}^* is the dual map of \mathcal{L} , i.e. the one evolving observables instead of states. By expanding the dual map $\mathcal{L}^*[F_i] = \sum_{j=1}^{d^2} \text{Tr}(\mathcal{L}^*[F_i] F_j) F_j$, and substituting into eq. (S1), one obtains

$$\frac{d}{dt} v_i(t) = \sum_{j=1}^{d^2} \text{Tr}(\mathcal{L}^*[F_i] F_j) \text{Tr}(F_j \rho_S) = [\mathbf{L} \mathbf{v}(t)]_i, \quad (\text{S2})$$

where we defined the matrix \mathbf{L} as $\mathbf{L}_{ij} \equiv \text{Tr}(\mathcal{L}^*[F_i] F_j)$. By exploiting the cyclic property of the trace, one can explicitly calculate $\mathcal{L}^*[F_i]$ starting from the known action of $\mathcal{L}[\rho]$. Explicitly we obtain

$$\text{Tr}(\mathcal{L}^*[F_k] F_l) = \text{Tr} \left(i[H, F_k] F_l + \frac{1}{2} \sum_{i,j=2}^{d^2} c_{ij} (F_j^\dagger [F_k, F_i] F_l + [F_j^\dagger, F_k] F_i F_l) \right). \quad (\text{S3})$$

Comparing this result to eq. (5) we obtain

$$\mathbf{H}_{kl} = i \text{Tr}([H, F_k] F_l), \quad \mathbf{D}_{kl} = \frac{1}{2} \sum_{i,j=2}^{d^2} c_{ij} \text{Tr} \left(F_j^\dagger [F_k, F_i] F_l + [F_j^\dagger, F_k] F_i F_l \right). \quad (\text{S4})$$

To simplify these expressions we use the structure constants d_{ijk} and f_{ijk} , defined from the commutation and anticommutation relations as

$$d_{ijk} = \frac{1}{4} \text{Tr}(\{F_i, F_j\} F_k), \quad (\text{S5})$$

$$f_{ijk} = -\frac{i}{4} \text{Tr}([F_i, F_j] F_k). \quad (\text{S6})$$

By substituting those definitions into eq. (S4), we obtain the following expressions for the matrix elements of \mathbf{H}

$$\begin{aligned}\mathbf{H}_{ij} &= -4 \sum_{k=2}^{d^2} f_{ijk} \theta_k^H, & i, j \in \{2, \dots, d^2\} \\ \mathbf{H}_{i1} = \mathbf{H}_{1i} &= 0, & i \in \{1, \dots, d^2\}\end{aligned}\quad (\text{S7})$$

where θ_k^H provides the expansion of the Hamiltonian over the basis F_i , namely $H = \sum_{i=2}^{d^2} F_i \theta_i^H$. The matrix elements of \mathbf{D} reads

$$\begin{aligned}\mathbf{D}_{mn} &= -8 \sum_{i,j,k=2}^{d^2} f_{mik} (f_{njk} \text{Re}(c)_{ij} + d_{njk} \text{Im}(c)_{ij}), & m, n \in \{2, \dots, d^2\} \\ \mathbf{D}_{m1} &= 2 \sum_{i,j=2}^{d^2} f_{imj} \text{Im}(c)_{ij}, & \mathbf{D}_{1m} = 0, & m \in \{1, \dots, d^2\}\end{aligned}\quad (\text{S8})$$

where c_{ij} is the so-called Kossakowski matrix, which we parametrized as $c = Z^\dagger Z$ with $Z = \theta^X + i\theta^Y$ and θ^X, θ^Y real matrices.

Equations (S7) and (S8) are the matrix representation of the Lindblad operator \mathcal{L} in terms of the vector ω_i and the Kossakowski matrix c_{ij} .

II. FURTHER RESULTS

In the main text, we investigate the behavior of the ML framework applied to spin chains having $V = 0.1\Omega, 2\Omega$. Here we report the case of a spin chain having $V = 0.5\Omega$. The results are shown in Figure S1. The graph exhibits a similar behavior to the cases analyzed in the main text. However, there is a notable distinction: the model saturates at a higher error. This discrepancy arises from the stronger coupling between the system and the bath, rendering non-Markovian effects non-negligible.

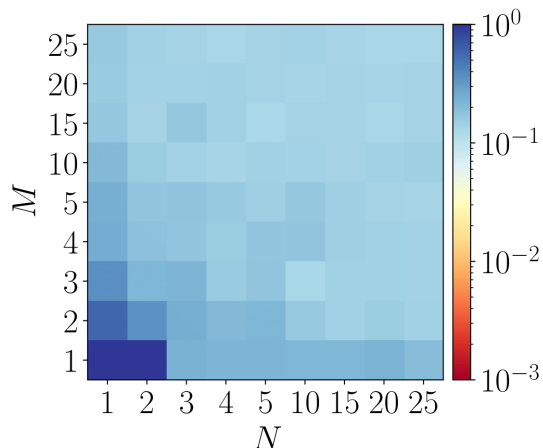


FIG. S1. **ML algorithm’s performance on open system dynamics.** The figure reports the error $\epsilon(N, M)$ as in (9) plotted for different values of N and M , and for the $V = 0.5\Omega$ spin chain.

III. ADDITIONAL PLOTS AND PHYSICAL “READ OUT”

For completeness, we report here the plots of all 15 non-trivial components of the coherence vector v_i . The plots represent the exact time evolution (black solid line) and the dynamics predicted by the ML algorithm (red dashed line). In Figure S2 and S3 we report the performance of the model trained on data that refers to a spin chain

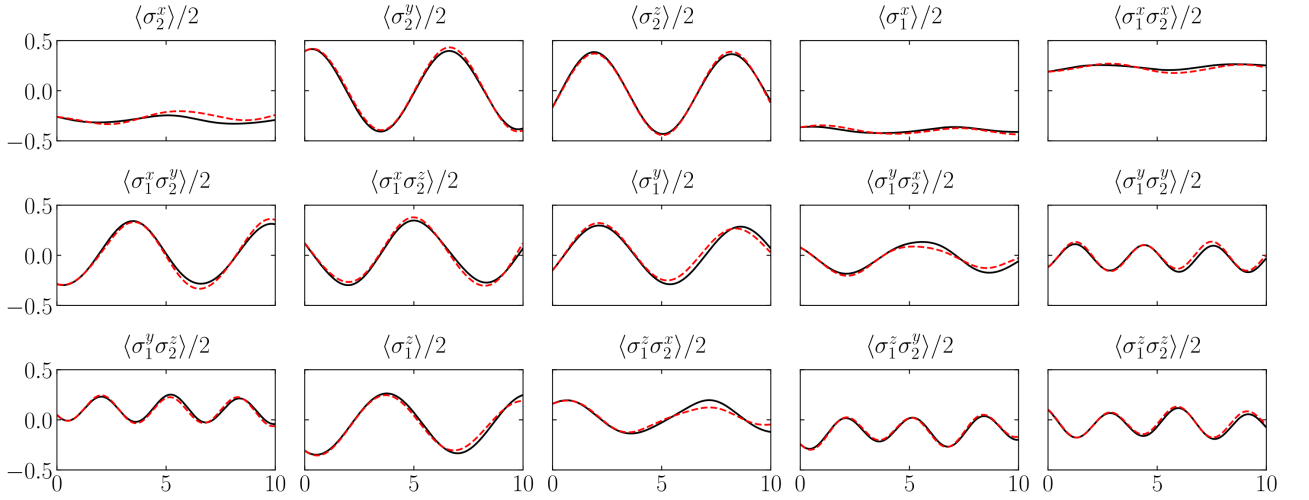


FIG. S2. **Full plot of the ML algorithm's performance.** In figure the 15 non-trivial components of the coherence vector v_i . The data refers to a spin chain with $V = 0.1\Omega$, and $N = 20, M = 20$. The plots represent the exact time evolution (black solid line) and the dynamics predicted by the ML algorithm (red dashed line). In this case, the overall dynamics is roughly captured.

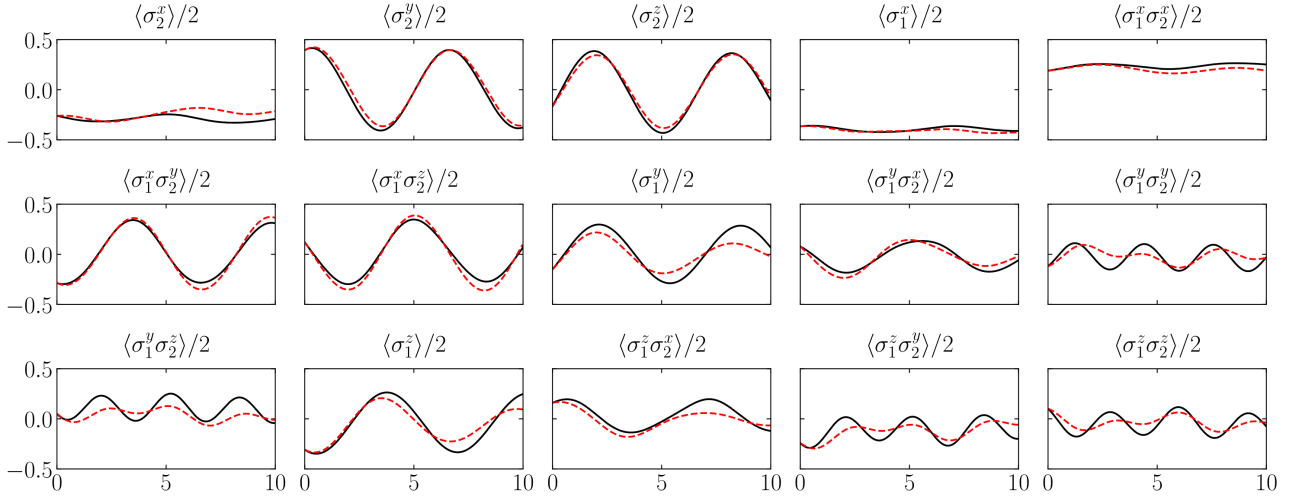


FIG. S3. **Full plot of the ML algorithm's performance.** In figure the 15 non-trivial components of the coherence vector v_i . The data refers to a spin chain with the same parameters as in Fig. S2, but $N = 2, M = 2$. The plots represent the exact time evolution (black solid line) and the dynamics predicted by the ML algorithm (red dashed line). In this case, the overall dynamics is well captured.

with $V = 0.1\Omega$. In Fig. S2 the synthetic experimental data is generated with $N = 20, M = 20$. In this case, the ML prediction is almost perfectly overlapped on the exact line, indicating the correctness of the learned Lindbladian. In Fig. S3 the synthetic experimental data is generated with $N = 2, M = 2$. In this case, the dynamics is roughly captured, made exception for some observables e.g., $\sigma_1^y \sigma_2^y, \sigma_1^y \sigma_2^z, \sigma_1^z \sigma_2^y$ and $\sigma_1^z \sigma_2^z$.

In addition to the complete plots, we here report the learned expressions for the Hamiltonian and jump operators. In this case, we take into consideration the model trained over $N = 20$ and $M = 20$. For the Hamiltonian, we round up to two decimal places to improve readability; it reads

$$H = 0.5(\sigma_1^x + \sigma_2^x) + 0.055(\sigma_1^z + \sigma_2^z) + 0.025 \sigma_1^z \sigma_2^z - 0.03(\sigma_1^y + \sigma_2^y) + 0.005(-\sigma_1^z \sigma_2^x - \sigma_1^y \sigma_2^x + \sigma_1^z \sigma_2^y - \sigma_1^x \sigma_2^z + \sigma_1^x \sigma_2^y).$$

Regarding dissipation part, there is only one non-zero eigenvalue (rounded to three decimal places) of the Kossakowski

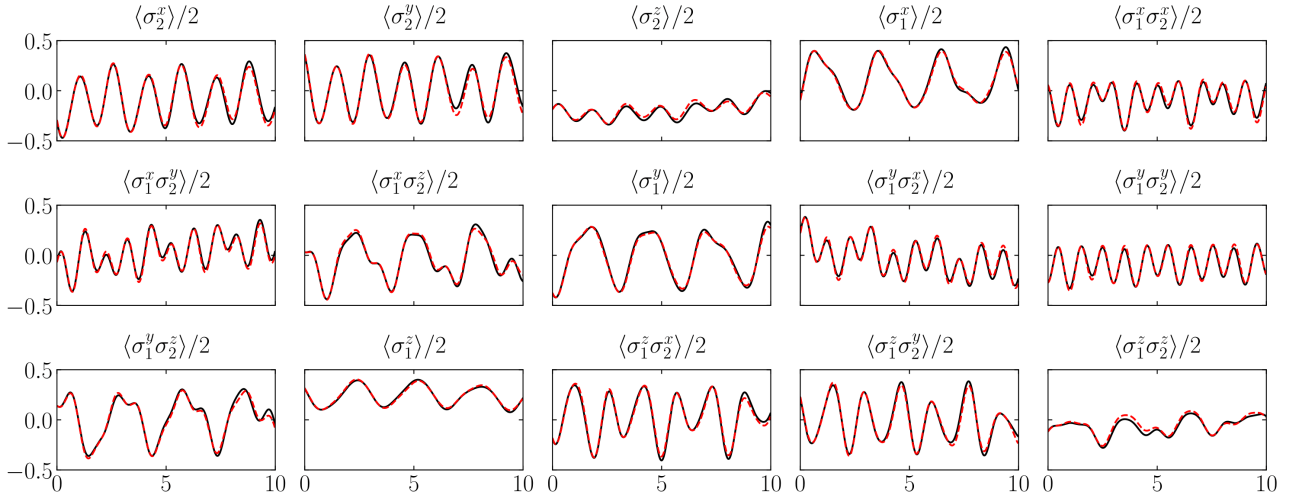


FIG. S4. **Full plot of the ML algorithm's performance.** In figure the 15 non-trivial components of the coherence vector v_i . The data refers to a spin chain with $V = 2\Omega$, and $N = 20, M = 20$. The plots represent the exact time evolution (black solid line) and the dynamics predicted by the ML algorithm (red dashed line). In this case, the overall dynamics is well captured.

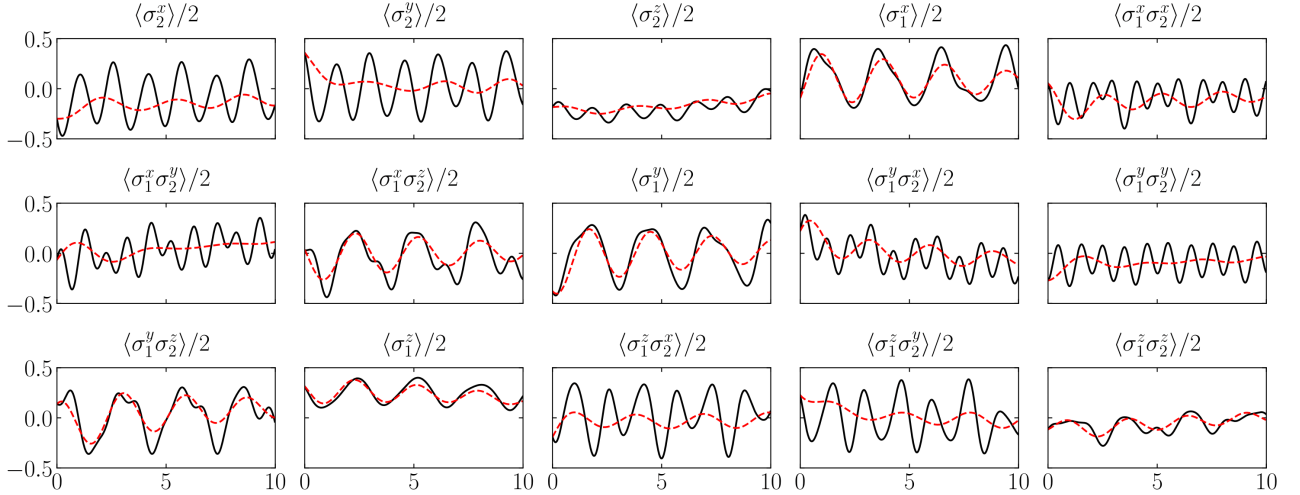


FIG. S5. **Full plot of the ML algorithm's performance.** In figure the 15 non-trivial components of the coherence vector v_i . The data refers to a spin chain with the same parameters as in Fig. S4, but $N = 3, M = 2$. The plots represent the exact time evolution (black solid line) and the dynamics predicted by the ML algorithm (red dashed line). In this case, the overall dynamics is poorly captured.

matrix $\gamma = 0.013$. Regarding the jump operator, rounded up to two decimal places, it reads

$$\begin{aligned}
J = & 0.245 \sigma_1^x \sigma_2^z + (0.19 - 0.1i) \sigma_2^y + (-0.16 + 0.07i) \sigma_1^x \sigma_2^y + (0.055 + 0.145i) \sigma_1^y \sigma_2^y + (0.095 - 0.075i) \sigma_1^y + \\
& + (-0.09 - 0.08i) \sigma_1^y \sigma_2^x - (0.05 + 0.105i) \sigma_1^x + (0.09 - 0.065i) \sigma_1^x \sigma_2^x + (0.03 - 0.1i) \sigma_1^z \sigma_2^x + (-0.09 + 0.02i) \sigma_1^z + \\
& + (0.005 + 0.08i) \sigma_1^z \sigma_2^x + (-0.03 + 0.06i) \sigma_1^z \sigma_2^z + (0.02 + 0.035i) \sigma_1^y \sigma_2^z + 0.005 \sigma_2^z
\end{aligned}$$

In Figure S4 and S5 we report the performance of the model trained on data that refers to a spin chain with $V = 2\Omega$. In Fig. S4 the synthetic experimental data is generated with $N = 20, M = 20$. In this case, the ML prediction is almost perfectly overlapped on the exact line, indicating the correctness of the learned Lindbladian. In Fig. S5 the synthetic experimental data is generated with $N = 3, M = 2$. In this case, the overall dynamics is poorly captured.