Koch, Rouven; Lado, Jose

Neural network enhanced hybrid quantum many-body dynamical distributions

Published in:
Physical Review Research

DOI:
10.1103/PhysRevResearch.3.033102

Published: 29/07/2021

Document Version
Publisher's PDF, also known as Version of record

Neural network enhanced hybrid quantum many-body dynamical distributions

Rouven Koch and Jose L. Lado
Department of Applied Physics, Aalto University, 00076 Aalto, Espoo, Finland

(Received 11 May 2021; accepted 28 June 2021; published 29 July 2021)

Computing dynamical distributions in quantum many-body systems represents one of the paradigmatic open problems in theoretical condensed matter physics. Despite the existence of different techniques both in real-time and frequency space, computational limitations often dramatically constrain the physical regimes in which quantum many-body dynamics can be efficiently solved. Here we show that the combination of machine-learning methods and complementary many-body tensor network techniques substantially decreases the computational cost of quantum many-body dynamics. We demonstrate that combining kernel polynomial techniques and real-time evolution, together with deep neural networks, allows to compute dynamical quantities faithfully. Focusing on many-body dynamical distributions, we show that this hybrid neural-network many-body algorithm, trained with single-particle data only, can efficiently extrapolate dynamics for many-body systems without prior knowledge. Importantly, this algorithm is shown to be substantially resilient to numerical noise, a feature of major importance when using this algorithm together with noisy many-body methods. Ultimately, our results provide a starting point towards neural-network powered algorithms to support a variety of quantum many-body dynamical methods, that could potentially solve computationally expensive many-body systems in a more efficient manner.

DOI: 10.1103/PhysRevResearch.3.033102

I. INTRODUCTION

The dynamical and spectral properties of quantum many-body systems represents one of the central directions of modern quantum many-body physics. In particular, the development of powerful methods, such as auxiliary field quantum Monte Carlo [1–3], tensor networks [4–7], and neural network quantum states [8–15] have shed light onto the properties of paradigmatic quantum many-problems that just a few years ago well drastically beyond conventional methods [7,16,17]. While the previous methods have been shown highly successful for tacking static properties of many-body systems, time-dependent and dynamical properties represent still a critical open problem [18–26]. Spectral properties of many-body systems represent a central issue in several novel directions of quantum-many body physics, including non-equilibrium dynamics [27,28], many-body localization [29], many-body time crystals [30–32], and dynamical phase transitions [33,34]. Interestingly, although a variety of methods to compute dynamical properties exist that are affected in different forms by numerical limitations and noise [18,19,35–44], combining these methods simultaneously is a highly nontrivial problem.

Deep learning [45,46] has risen as a disruptive method for different fields of physics, including condensed matter physics [47,48]. Examples of these successful applications are neural network quantum states [8–15], the detection of phases of matter [49–55], and machine-learning strategies for quantum control [56,57]. Interestingly, the integration of machine-learning methods with conventional many-body formalism has the potential of drastically extending the applicability of typical methods, and even overcoming conventional computational limitations [58–62]. However, up to date, the calculation of specific quantum many-body quantities is usually performed with a single method at a time, and potential synergies of combining them through in a nontrivial way several of them have remained relatively unexplored.

Here we demonstrate that machine-learning methods combined with conventional dynamical-many body techniques allows computing spectral properties with dramatically increased accuracy. In particular, we developed an algorithm combining results of the Kernel polynomial method (KPM) [63] and time evolution (TE) to calculate dynamical spectral distributions of single-particle and many-body systems. This hybrid neural-network algorithm is able to drastically enhance the spectral properties of quantum many-body systems being trained solely in single-particle data. We further demonstrate that our neural-network algorithm is extremely robust, being resilient to up to 10% noise, making it ideal for real applications with state of the art methods showing numerical noise. This paper is organized as follows: in Sec. II we present the details of our time-frequency algorithm, in Sec. III we show its application to a family of single particle problems, in Sec. IV we show its extension to many-body systems without prior knowledge, finally in Sec. V we summarize our conclusions.

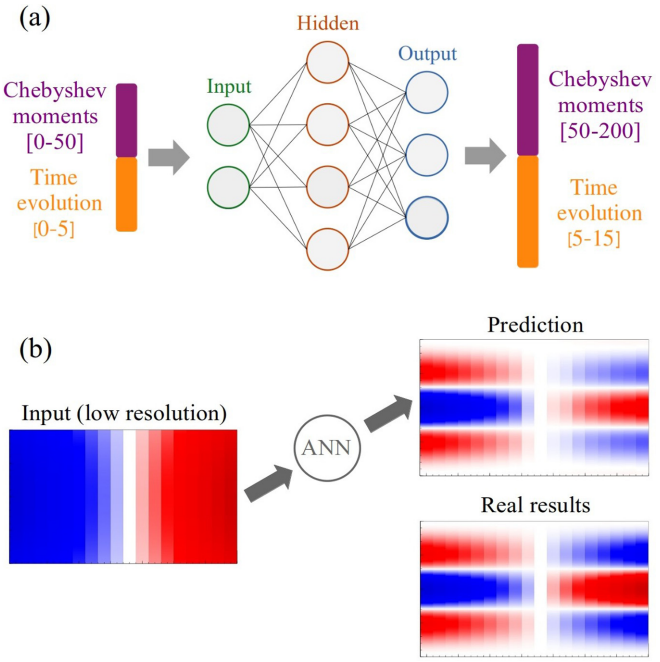

FIG. 1. (a) Sketch of the hybrid neural-network many-body algorithm. The input of the ANN consists of Chebyshev moments and time evolution, predicting higher-order moments and longer time evolution. The network is trained with single-particle, random-generated data and is able to make predictions of many-body systems. (b) Comparison between the input low-quality data taken as input of the algorithm, and the output high-quality prediction together with its comparison with the real high-quality data.

II. METHODS

The method we put forward in this paper combines two well-established techniques, kernel polynomial expansions, and time evolution, to compute the dynamical spectral properties of many-body system powered by a neural-network algorithm. The basic idea of our proposal is shown in Fig. 1. When the machine-learning algorithm is trained, initial many-body data of the distribution is taken as input, and the neural network returns as output a greatly enhanced many-body distribution. The many-body algorithm is solely trained on in single-particle data, much cheaper to generate, and as we will show proves to successfully enhance many-body data. We now elaborate on the different parts of the full method.

A. The kernel polynomial method and Chebyshev expansion

Quantity of critical interest in a variety of single particle and many-body systems can be written in the form

$$f(x) = \langle\psi|A\delta(x-H)B|\psi\rangle = \langle\alpha|\delta(x-H)|\beta\rangle, \quad (1)$$

with A, B, H operators and ψ a certain state. In particular, dynamical spectral functions, such as dynamical spin structure factors in spin systems, electronic many-body density of states, and many-body magnetization distributions can be represented in the previous form. From the computational point of view, a formal way of computing such objects relies on full diagonalization of the operator H. However, the computation of the eigenvalues and eigenvectors of a matrix is for large systems, in particular in a many-body system, an almost unfeasible task. A greatly efficient alternative method is the Kernel polynomial method and expansion in terms of Chebyshev polynomials [63]. Since the Chebyshev polynomials are defined in the interval $I = (-1, 1)$, a general expansion of $f : (-1, 1) \to R$ has the form

$$f(x) = \alpha_0 + 2\sum_{n=1}^{\infty}\alpha_n T_n(x), \quad (2)$$

where $\alpha_n = \langle f|T_n\rangle$ define the coefficients of the expansion and T_n are Chebyshev polynomials of order n. After rescaling the Hamiltonian and corresponding energies into the interval $(-1, 1)$, the Chebyshev moments μ_n are defined as

$$\mu_n = \langle\beta|T_n(H)|\alpha\rangle, \quad (3)$$

where $|\alpha\rangle$ and $|\beta\rangle$ are states of the system. Expansion of Eq. (2) to finite order leads to so-called Gibbs oscillations. To avoid/minimize these oscillations a convolution of $f(x)$ with a kernel is applied. In this paper, we are using the Jackson Kernel to reduce oscillations [64].

We now highlight the data used for training the machine-learning algorithm. We will take of the density of states (DOS) of a single-particle system with eigenenergies ϵ_k. The DOS of a given single-particle Hamiltonian can be written as

$$\rho(\omega) = \sum\delta(\omega-\epsilon_k), \quad (4)$$

which can be calculated with the Chebyshev moments

$$\mu_n = \int_{-1}^{1}\rho(\omega)T_n(\omega)d\omega = \mathrm{Tr}[T_n(H)], \quad (5)$$

that can be computed with the recursion relations mentioned above.

We now elaborate on the data that we will use for the many-body model, for which the machine-learning algorithm trained on single-particle data will be used. While we could use a many-body formulation of the electronic density of states, this choice could be considered trivial to test the generality of the algorithm, and therefore we will take a quantity that has no analogy in a single-particle framework. In a many-body spin system, in order to take the most different case possible with respect to the single-particle training, we will focus on many-body magnetization distributions, that lack an analogy for single particle models and are defined as

$$f(m) = \langle\Omega|\delta\left(m - \frac{1}{L}\sum_n S_n^z\right)|\Omega\rangle, \quad (6)$$

where S_n^z are the local spin operators and $|\Omega\rangle$ is the many-body ground state. These magnetization distributions can be computed within the matrix-product state formalism [4–6], computing the many-body ground state implementing the Chebyshev expansion within the tensor-network algorithm [39,65–70].

B. Time evolution

Another approach to calculate the many-body spectral function or the density of states of single-particle systems is time evolution. For the training of the network, we will

use time evolution of single-particle models under the single-particle Hamiltonian. Focusing first on the case of a single particle system, the time evolution described by the Hamiltonian H is given by

$$f(t) = \langle\alpha|\exp(-iHt)|\alpha\rangle \ , \tag{7}$$

where $|\alpha\rangle$ is for example the state representing the lattice site where an electron is injected. In this case, $f(t)$ is the probability to find the electron at that lattice site at time t. The (local) DOS $\rho(\omega)$ is obtained by Fourier transformation of the function $f(t)$.

For the many-body case, we will use as input data time-evolution under the magnetization operator, defined as

$$f(t) = \langle\Omega|\exp\left(-\frac{i}{L}\sum_n S_n^z\right)|\Omega\rangle, \tag{8}$$

where the Fourier transformation leads to the magnetization distribution of the system.

C. Linear predictions

While both time-evolution and Chebyshev expansion can be arbitrarily accurate, finite truncations either in time or order of moment leads to a finite spectral resolution. In order to reduce these costs, prediction methods as linear predictions have been developed, that formally attempt to extrapolate the behavior of moments or time evolution. We will demonstrate that our neural-network method drastically outperforms these linear-prediction methods, and therefore for the sake of completeness we briefly introduce them here. One of the most commonly used methods for time-series predictions are linear-prediction models, e.g., autoregressive models [71]. In the autoregression model, the quantity of interest (here the next time steps of the time evolution or the Chebyshev moments) is predicted by a linear combination of p previous data points. The model of order p to predict the value X_t at time step t can be written as

$$X_t = c + \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \epsilon_t, \tag{9}$$

where c is a (time-independent) constant, ϵ_t white noise, and ϕ the p parameters that define the model. The considered previous time steps, also called the lags, are used to include autocorrelations among adjacent data points. Equation (9) allows to forecast an arbitrary number of time steps by using the predicted value as input for the next forecast. However, the accuracy (or confidence interval) decreases with every additional time step. Therefore, we are investigating more powerful prediction methods, which leads to the next section about artificial neural networks (ANNs).

D. Artificial neural networks

The field of machine learning, starting from simple linear regressions up to deep neural networks [45], opens new opportunities and insights for the field of physics. In this paper, we illustrate the predictive power of fully-connected deep neural networks and compare them with linear-prediction autoregression models. Deep neural networks consists of an input, output, and several hidden layers. The "neurons" of each layer are fully connected between neighboring layers. The connections, the so-called weights, determine the outcome and predictions of the ANN and are the parameters that have to be tuned during the training process. Using nonlinear activation functions for the neurons in each layer allows the neural network to represent a highly nonlinear function. This function predicts the outcome of the network for an array of input values. In order to train network, we are using the supervised learning based on backpropagation. The neural network is trained with input data, i.e., in our case an input array consisting of a limited amount of Chebyshev moments (and time steps of the TE) and the output is an array of higher-order moments (and more time steps). The update of the weights happens during the training process via minimizing the loss function, in this case the mean squared error, and backpropagation. We use stochastic gradient decent and we are using the Adam optimizer [72].

III. SINGLE-PARTICLE SYSTEMS

We now move on to consider the training data of our algorithm, single-particle data of a Hamiltonian, which in particular is implemented as a wide family of one-dimensional tight binding models.

A. Single-particle model

The physical quantity of interest in case of the tight-binding system is the density of states (DOS), which can either be computed using the Kernel polynomial method and the Chebyshev expansion or with the Fourier transformation of the time evolution. For more complex systems, e.g., many-body systems, the computation of higher-order moments and a long time evolution is very demanding. The higher the order of the moments used to calculate the DOS, the more details and features are captured and to obtain an accurate low-frequency DOS one needs a very long time evolution. A common approach to obtain higher-order moments or a longer time evolution without an explicit calculation is to use linear prediction, e.g., an autoregressive model. Given a small amount of Chebyshev moments (and only the first time steps) the model can extrapolate the following moments (and time evolution). In this paper, we present an alternative approach using an ANN algorithm to predict these quantities. The idea of the algorithm is to train an ANN to predict higher-order moments (and longer time evolution) given a limited amount of input moments (and time-steps), similar to the linear-prediction models. Our algorithm is capable of combining both approaches, i.e., forecasting higher-order moments and a longer time evolution at the same time. We define a parametric family of spinless one-dimensional Hamiltonians of the form

$$H_{TB} = \sum_{\langle i,j\rangle} t_{ij} c_i^\dagger c_j + \sum_i v_i c_i^\dagger c_i + \text{H.c.}, \tag{10}$$

where t_{ij} is the hopping amplitude, v_i the on-site energy, and $c^\dagger$ (c) the electron creation (annihilation) operators.

B. Neural-network versus autoregression

Here we first show how the use of neural-network prediction provides a greatly stable algorithm, overcoming limitations of autoregressive models. For the sake of simplicity,

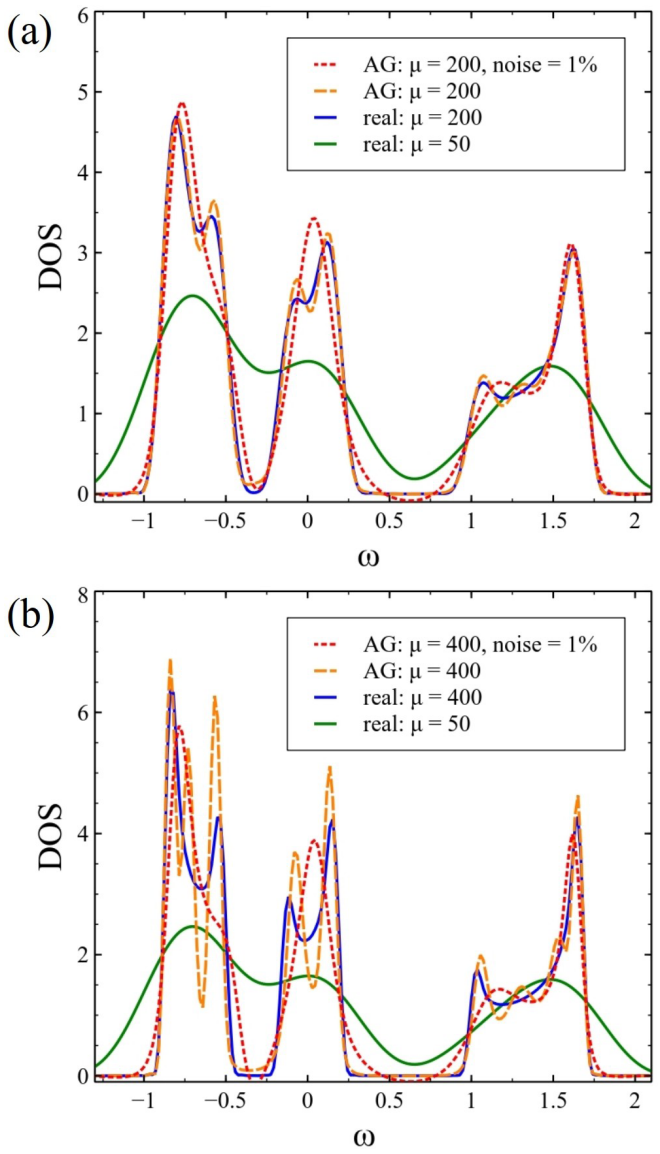

FIG. 2. Linear predictions (autoregressive model) of the DOS of a random-generated tight-binding model with 1% (red) and without noise (orange) in the Chebyshev moments. The predictions up to 200 moments are shown in (a) and up to 400 in (b), respectively. The real distributions for 50 and 200 (400) moments are shown in green and blue.

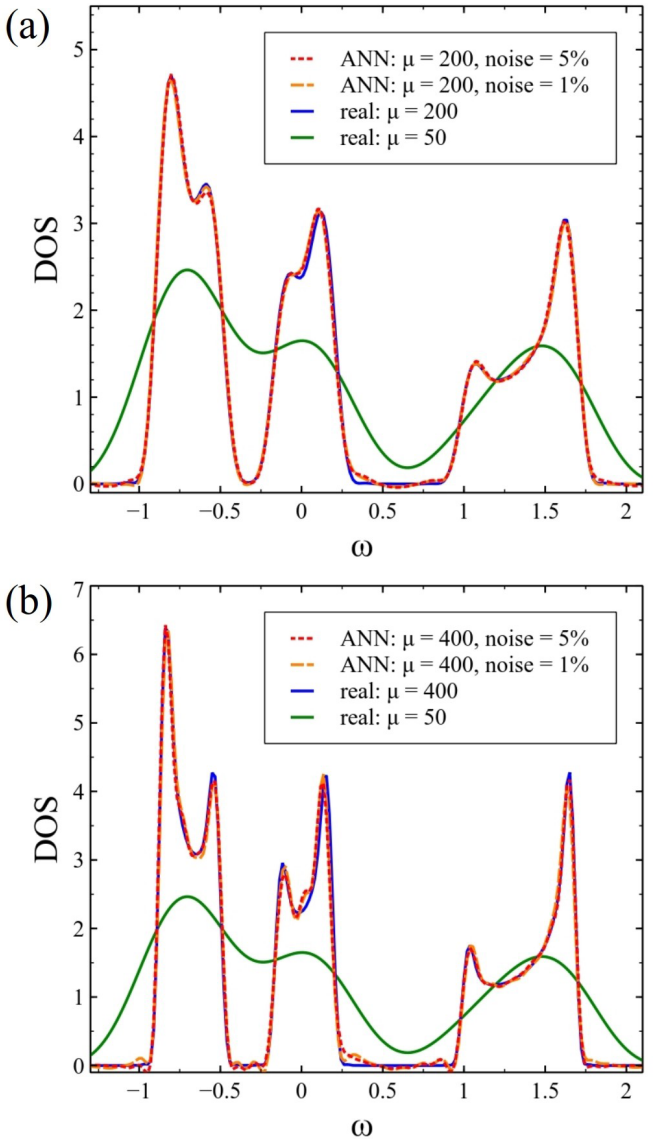

FIG. 3. DOS for a random-generated tight-binding system out of the test set, calculated with 50 (green) and 200/400 (blue) Chebyshev moments. Predictions of the ANN with 1% (orange) and 5% (red) of noise for 200 and 400 moments are shown in (a) and (b).

the neural-network algorithm is trained only with Chebyshev moments, and the combination with time evolution is addressed in the next section.

As a starting point, we first consider the density of states (DOS) of these family of single particle models. For the neural network, the training is performed with single-particle models with translational symmetry and three orbitals per unit cell. We compute the DOS with the Chebyshev algorithm, and we compare with extrapolations performed with the autoregressive model. Figure 2 shows the DOS for a random-generated tight-binding systems. The higher-order moments are predicted with an autoregressive model, including a time window of 10 data points, given 50 moments as input and predictions up to the order of 200 and 400. This model achieves accurate results only in the case without noise up to 200 moments as shown in Fig. 2(a). For higher-order predictions [Fig. 2(b)] and noise values of about 1% the error of the autoregression model drastically increases and fails to predict the DOS (and, therefore, Chebyshev moments) with high accuracy—specific features and peaks of the original DOS are not captured. This shows the high sensitivity of linear predictions to noise, and even in the noiseless severely limits the predictions (a maximum of a factor 4 in the noiseless limit). We now move on to show how neural networks overcome these limitations.

Deep neural networks are in general, more robust when facing noise and can even benefit from it in order to make predictions more stable and reduce the probability of overfitting. Figure 3 shows the predicted DOS of the trained network, showing a great agreement with the true distribution associated to the true moments. The architecture of the ANN is

described in the Appendix. The model is trained with 10 000 random generated tight-binding systems with different values for hopping amplitudes and on-site energies, given the first 50 moments as input vector and the next 350 moments as output during the training process. To show the robustness against noise, we induced 1% and 5% of noise in the Chebyshev moments. The results of Fig. 3 are that the predictions of the ANN algorithm are very accurate up to 200 moments [(a) and (b)] regardless of the induced noise. Even for predictions up to order 400 (factor 8) the ANN is able to predict the real DOS accurately, capturing all main features. In this case, some oscillations start appearing in comparison to the predictions up to 200 moments. Nonetheless, the enhancement of the input DOS (blue) and increased resolution is significant in all cases. This shows that deep neural networks have a great advantage over the linear predictions (see Fig. 2).

C. Chebyshev and time-evolution neural-network architecture

After showing the numerical robustness of the neural network, we now move on to consider a more advanced neural-network architecture in which both Chebyshev moments and time evolution are included. The combination of these complementary methods allows to further test the reliability of the machine-learning algorithm, as the output and input contains both frequency and time information. Overall, we find outstanding agreement for extrapolations of a factor 8 in the Chebyshev moments, and with a strong reliance to numerical noise of up to 10%.

We first comment on the training set. The training models consist of randomly generated 5000 systems, consisting of two sites per unit cell with different hopping amplitudes. The architecture of this model is described in the Appendix and similar to the one for the many-body predictions. The predictions of the trained model are shown in Figs. 4(a), 4(c), and 4(e) and compared with linear predictions of the same system, Figs. 4(b), 4(d), and 4(f). Another difference to the previous model is that we are now predicting the Chebyshev moments [Fig. 4(a)] and time evolution [Fig. 4(c)] at the same time, i.e., including both in the same input array for the ANN. We use the first 25 moments and 50 time steps as input and calculate up to 200 moments and 200 time steps. The predictions shown in Figs. 4(a) and 4(c) are very accurate, especially for the moments and the resulting DOS prediction [Fig. 4(e)] is almost indistinguishable from the original one. In the predictions of the moments, we induced noise of 3% in order to show the robustness of the ANN and also decrease the risk of overfitting. The averaged root mean squared error (RMSE) of the test set (500 random generated systems) for the predictions of the moments is 0.008 and for the time evolution 0.005. The enhancement of the resolution in comparison to the input DOS (blue) is significant and shows a pronounced peak. The combination of the moments with the time evolution is one of the main advantages over the linear-prediction models. Figs. 4(b) and 4(d) show the predictions of the autoregression model (without noise) for the Chebyshev moments and time evolution. The predictions of the moments are accurate up to order 170 and the time evolution without nose up to 115 time steps. The induced noise in the TE leads to instabilities of the linear model and drastically decreases the accuracy. The same

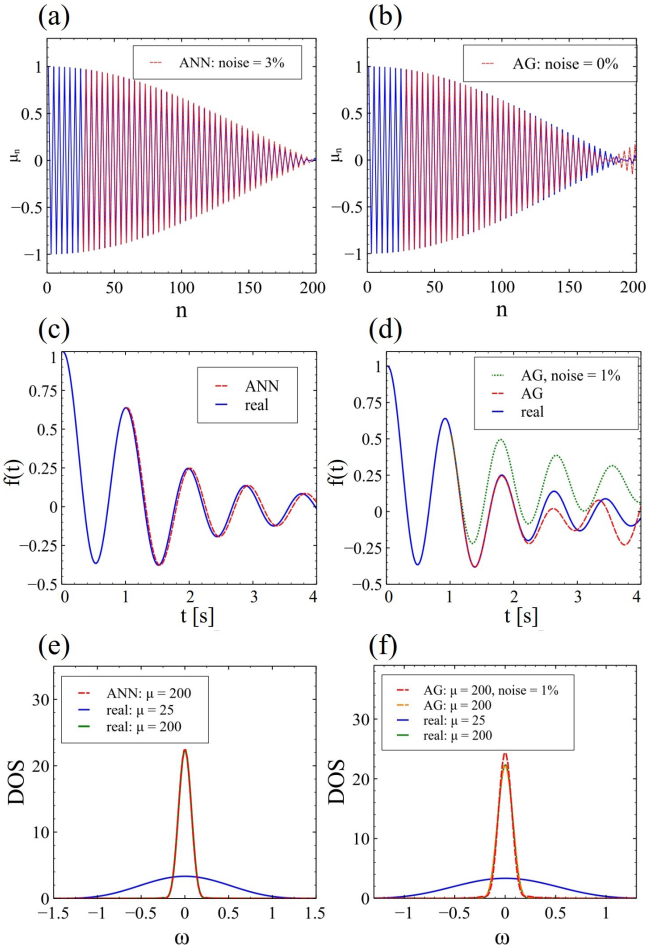

FIG. 4. (Random-generated) tight-binding model similar to the expected many-body systems. The system is chosen out of the test-set of the ANN. The predictions (red) for the Chebyshev moments of the ANN with 3% noise and AG model without noise are shown in (a) and (b). The corresponding time-evolution predictions are shown in (c) and (d) including noise in the AG prediction (green). The corresponding DOS are shown in (e) and (f), comparing the input distribution (green) with the real (blue) and predicted one (red) including 200 instead of 25 moments.

applies for noise in the Chebyshev moments which leads to a decreased accuracy in the prediction of the DOS, shown in Fig. 4(f).

With the previous neural-network architecture, we now move on to extrapolate the many-body distributions of a quantum many-body system.

IV. MANY-BODY SYSTEMS

We now move on to consider how our trained neural network performs when presented with new data, now originating from a many-body Hamiltonian. In particular, in order to demonstrate the generality of the training, we will focus on the many-body magnetization distribution of a spin-chain, a quantity that has no analogy with the single-particle density of states used in the training of the machine-learning algorithm.

A. Many-body model

Here we will consider a many-body spin Hamiltonian, in the form of a $S = 1/2$ one-dimensional many-body system defined by the Hamiltonian

$$H = J \sum_n S_n^x S_{n+1}^x + S_n^y S_{n+1}^y + S_n^z S_{n+1}^z \tag{11}$$

First, it is worth to note that $S = 1/2$ one dimensional models can be generically mapped to interacting spinless fermionic models by means of a Jordan-Wigner transformation [73]. In this context, spin-excitations in the many-body spin chain $\langle\Omega|S_n^+\delta(\omega - H + E_0)S_n^-|\Omega\rangle$ are mapped to quasiparticle spectral functions in the interacting fermionic model $\langle\Omega|c_n^\dagger\delta(\omega - H + E_0)c_n|\Omega\rangle$. In this regard, deploying the trained hybrid neural-network algorithm to compute spin-excitations of the interacting Heisenberg model would be a relatively trivial test of our algorithm. In order to target a more nontrivial problem, here we will focus on the many-body magnetization distribution, which cannot be mapped to a single-particle property, and therefore represents a more advanced test of the algorithm.

The physical quantity of interest is for the many-body systems the magnetization distribution, defined as

$$f(m) = \langle\Omega|\delta(m - \hat{M}_z)|\Omega\rangle, \tag{12}$$

with $\hat{M}_z = \frac{1}{L}\sum_n S_n^z$, with L the number of sites and $|\Omega\rangle$ the many-body ground state of the Hamiltonian H of Eq. (11). This magnetization distribution reflects the spectral composition of the ground state in terms of eigenstates of $\hat{M}_z$, in particular reflecting the many-body entanglement of the system [74–76]. As noted above, this quantity does not map to a meaningful object in the single-particle limit, and therefore represents a nontrivial test for the hybrid neural-network algorithm. The previous quantity is computed both via the Chebyshev expansion and time-evolution methods, and used as input for the neural network. As a reference, we will also compare linear predictions with the results of the ANN algorithm for a many-body system.

B. Autoregressive model

We first address the performance of autoregressive models for extrapolating the many-body magnetization distribution. The results of the autoregressive model are shown in Fig. 5 and lead to similar conclusions as in the single-particle case, where we studied the DOS in Fig. 2. The predictions of the Chebyshev moments with 1% noise are shown in Fig. 5(a). The induced noise in the moments of 1% or even less leads to instabilities in the model and decreases the accuracy enormously. This can also be seen in the corresponding magnetization and the loss in accuracy in Fig. 5(b).

C. Hybrid neural-network many-body algorithm

After showing the sensitivity of autoregression to noise, we now move on to use our trained hybrid neural-network architecture in the many-body distribution problem. The robustness to noise is one of the main advantages and motivations to develop an ANN algorithm to predict higher-order moments (and TE) and enable the calculation of the magnetization

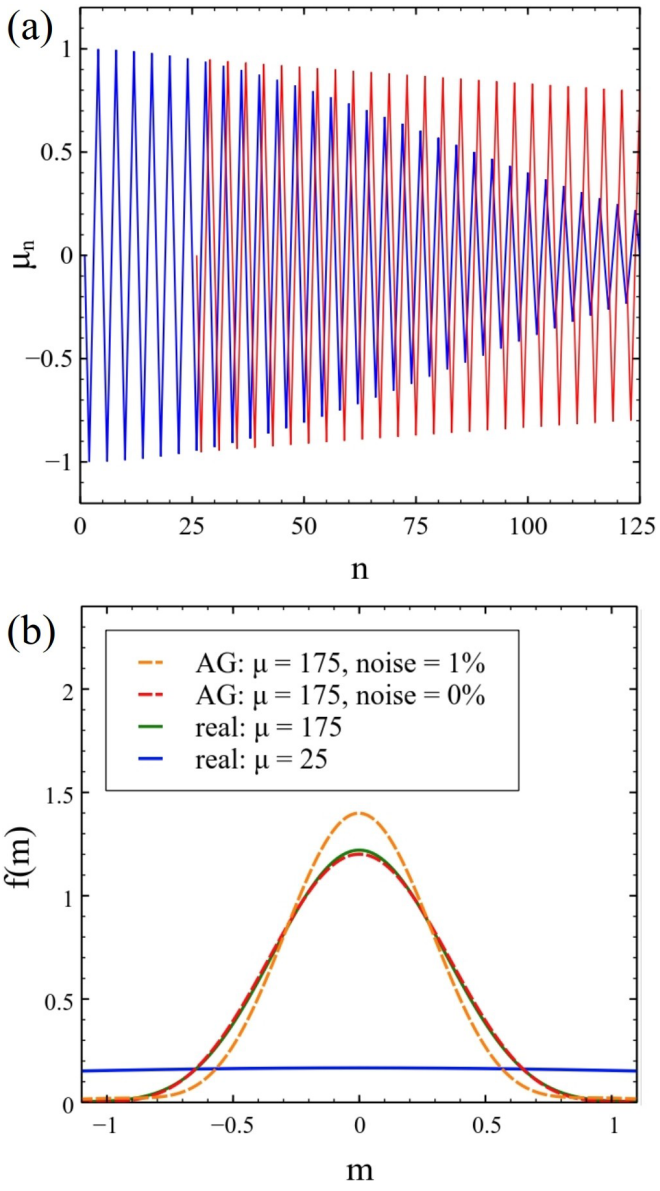

FIG. 5. Linear predictions (red) of the autoregressive model of the Heisenberg S1/2 chain (30 sites) for the Chebyshev moments with 1% noise (a). The input for the model are the first 25 moments and the first 50 time steps for the time evolution. The resulting magnetization distribution without (red) and with 1% (orange) noise is shown in (b).

distribution of complex systems with higher resolution. The training data remains the same as for the single-particle system—random-generated tight-binding systems with a similar DOS as the magnetization distribution of the many-body systems. The ANN is, therefore, trained with purely single-particle physics and predictions are made for many-body systems without prior knowledge.

The predictions of our ANN algorithm for the one-dimensional Heisenberg $S = 1/2$ model are shown in Fig. 6. The algorithm predicts Chebyshev moments and the time evolution simultaneously given an input array of 25 moments and 50 time steps. The predictions of the moments above order 170 become less accurate, achieving an accurate prediction of

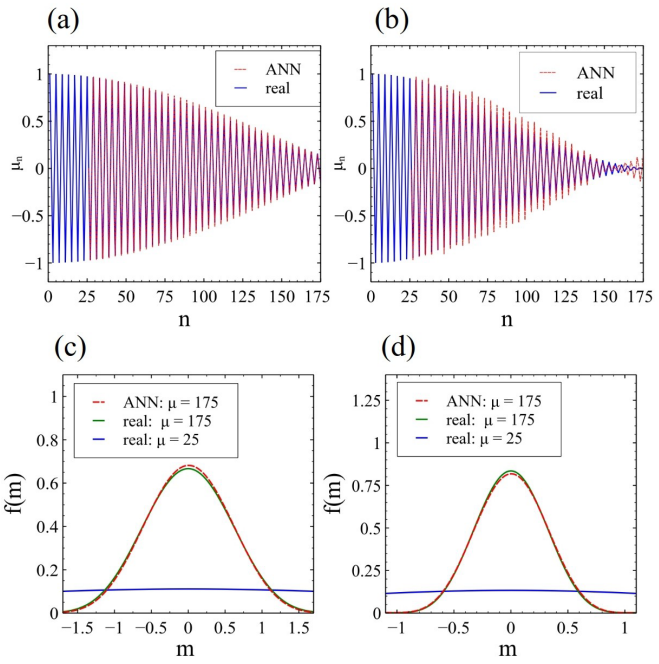

FIG. 6. Predictions of the ANN for the (1d) Heisenberg S1/2 model. The Chebyshev moments and time evolution are predicted simultaneously with an input array consisting of the first 25 moments and 50 time steps. The predictions (red) of the trained ANN of the moments for system size $L = 10$ and $L = 30$ are shown in (a) and (b). The corresponding magnetization distribution in (c) and (d). The magnetization distribution is calculated with the input moments (blue) in comparison with the real (green) and predicted (red) distribution for 175 moments.

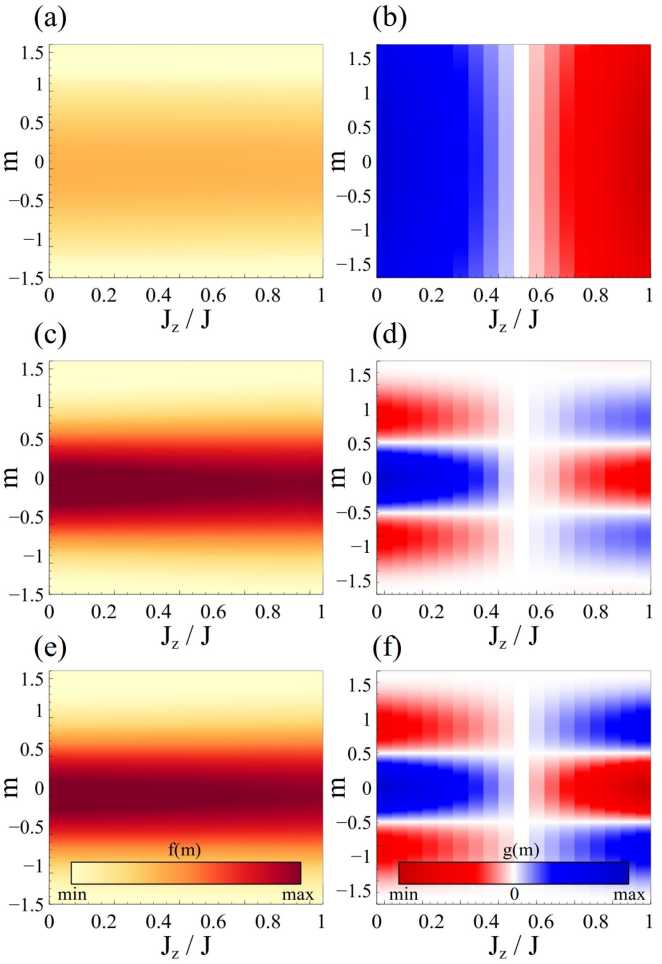

FIG. 7. Magnetization distribution of the quantum many-body model of Eq. (13). Panels [(a),(b)] show the low-quality distribution, [(c),(d)] the predicted distribution by the NN, and [(e),(f)] the true distribution. Panels [(a),(c),(e)] show the original distribution $f(m)$, and [(b),(d),(f)] background-subtracted distribution $g(m)$. It is observed that the predicted distribution [(c),(d)] and the real one [(e),(f)] show a substantial agreement.

factor 7. The magnetization distributions for the Heisenberg $S = 1/2$ chains is shown in Fig. 6(b). Shown is the input distribution using only 25 moments and the real distribution for 175 and the predictions of the ANN algorithm. The predictions show high accuracy and match the expected distributions well for both system sizes. They also show an enormous enhancement to the almost flat input distribution which does not show the pronounced peak. Therefore, the fluctuations in the moments in Fig. 6(b) do not have a significant impact on the distribution. From this follows, that the ANN algorithm is capable of enhancing the many-body distribution significantly. The resolution of the predictions is however limited up to a certain order of Chebyshev moments. In comparison with the AG magnetization distributions [Fig. 5(b)] the ANN algorithm is able to match the real distribution with higher accuracy.

We now use our trained neural-network algorithm to study the magnetization distribution of a generalized many-body model, namely the XXZ $S = 1/2$ model defined as

$$H(J_z) = J \sum_n \left[S_n^x S_{n+1}^x + S_n^y S_{n+1}^y\right] + J_z \sum_n S_n^z S_{n+1}^z. \quad (13)$$

For $J_z = 1$, the previous model corresponds to the Heisenberg model studied previously. The magnetization distribution in the z direction depends on the specific value of J_z. In the following we show how the neural network successfully predicts the evolution of the magnetization distribution in this generalized model. To highlight the differences for the different values of J_z, we will now plot the difference of the distribution with respect to the ground state for $J_z/J = 0.5$, and normalized to the maximum value for the distribution, namely $f(m, J_z) = \langle \Omega_{J_z} | \delta(m - \hat{M}_z) | \Omega_{J_z} \rangle$, where $|\Omega_{J_z}\rangle$ is the many-body ground state of $H(J_z)$. Furthermore, to highlight that the neural network captures even the fine changes in the distribution, we define a background subtracted distribution of the form $g(m, J_z) = f(m, J_z) - f(m, J_z = 0.5J)$, that allows to observe the subtle changes in the distribution with respect to the one for $J_z = 0.5J$. We show in Fig. 7 the distributions $f(m)$ [Figs. 7(a), 7(c), and 7(e)] and $g(m)$ [Figs. 7(b), 7(d), and 7(f)]. In particular, Figs. 7(a) and 7(b) show the low-quality distributions, that represent the input of the trained algorithm. The predicted distribution by our neural network is shown in Figs. 7(c) and 7(d), which shows a great agreement with the true high quality real distributions shown in Figs. 7(e)

and 7(f). The previous results further demonstrate that our trained algorithm successfully predicts distributions of the generalized model of Eq. (13), even capturing fine details of the distribution.

V. SUMMARY

We have shown that neural networks, in combination with standard methods for computing spectral properties, provide a powerful framework to predict dynamics with increased accuracy. In particular, taking as examples time-evolution and Chebyshev methods, we have shown that reliable extrapolations can be performed. This was demonstrated first for a single-particle problem, where our architecture greatly improved the spectral resolution of the density of states. We then showed that the hybrid neural-network architecture trained with single-particle data could be deployed to enhance the many-body magnetization distribution of a purely many-body problem, a problem drastically different from the training set of the algorithm. Importantly, we have shown that our method is highly resilient to noise and provides dramatically more stable predictions than conventional linear regression methods. These results can be further generalized to other methods of computing spectral properties, including correction vector methods and imaginary time evolution. Importantly, while our results have been implemented with a tensor-network formalism, an analogous procedure can be carried out with generic quantum many-body methods. Our results demonstrate that neural network enhanced many-body methods provide a numerically robust formalism to compute many-body spectral properties.

ACKNOWLEDGMENTS

We acknowledge the computational resources provided by the Aalto Science-IT project, and financial support from the Academy of Finland Projects No. 331342 and No. 336243. We thank C. Flindt and T. Kist for fruitful discussions.

APPENDIX: DEEP NEURAL NETWORK ARCHITECTURES

The ANN architecture of the single-particle systems with only Chebyshev moments (see Fig. 3) is defined as follows: an input layer of dimension 50, three hidden layers of dimensions 256, 512, and 512 with 15% of dropout [77], and an output layer of dimension 350. As activation function we are using the nonlinear ReLu function. The test and training data consists of 10 000 random generated tight-binding models (90/10 split) defined with a 3-site unit cell, with different on-site energies and hopping amplitudes. We choose the Adam optimizer and optimized the mean squared error. The training process includes 300 epochs with a batch size of 16.

The ANN architecture of the combined Chebyshev and time-evolution input consists of an input layer of dimension 75, three hidden layers of size 512, 1024, and 512 with 20% dropout to avoid overfitting, and an output layer of dimension 525. As activation function we are using the nonlinear ReLu function. The training data consists of 5000 random generated tight-binding systems; in this case the unit cell consists of two sites with different hopping amplitudes and zero on-site energies. The training process is done with a batch size of 50 for 512 epochs and a train-test-split of 90/10. In addition, we included 5% of noise in the Chebyshev moments and 1% in the time evolution.

[1] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar, Monte Carlo calculations of coupled boson-fermion systems. I, Phys. Rev. D **24**, 2278 (1981).

[2] G. Sugiyama and S. Koonin, Auxiliary field Monte-Carlo for quantum many-body ground states, Ann. Phys. NY **168**, 1 (1986).

[3] ALF Collaboration, F. F. Assaad, M. Bercx, F. Goth, A. Götz, J. S. Hofmann, E. Huffman, Z. Liu, F. Parisen Toldin, J. S. E. Portela, and J. Schwab, The ALF (Algorithms for Lattice Fermions) project release 2.0. Documentation for the auxiliary-field quantum Monte Carlo code, 2012.11914v2.

[4] S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, Phys. Rev. Lett. **69**, 2863 (1992).

[5] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. NY **326**, 96 (2011).

[6] M. Fishman, S. R. White, and E. Miles Stoudenmire, The itensor software library for tensor network calculations, arXiv:2007.14822.

[7] T. Felser, S. Notarnicola, and S. Montangero, Efficient Tensor Network Ansatz for High-Dimensional Quantum Many-Body Problems, Phys. Rev. Lett. **126**, 170603 (2021).

[8] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[9] R. Kaubruegger, L. Pastori, and J. C. Budich, Chiral topological phases from artificial neural networks, Phys. Rev. B **97**, 195136 (2018).

[10] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Symmetries and Many-Body Excitations with Neural-Network Quantum States, Phys. Rev. Lett. **121**, 167204 (2018).

[11] A. Valenti, E. Greplova, N. H. Lindner, and S. D. Huber, Correlation-enhanced neural networks as interpretable variational quantum states, arXiv:2103.05017.

[12] D. Hendry and A. E. Feiguin, Machine learning approach to dynamical properties of quantum many-body systems, Phys. Rev. B **100**, 245123 (2019).

[13] X. Liang, W.-Y. Liu, P.-Z. Lin, G.-C. Guo, Y.-S. Zhang, and L. He, Solving frustrated quantum many-particle models with convolutional neural networks, Phys. Rev. B **98**, 104426 (2018).

[14] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Neural-Network Quantum States, String-Bond States, and Chiral Topological States, Phys. Rev. X **8**, 011006 (2018).

[15] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, Equivalence of restricted Boltzmann machines and tensor network states, Phys. Rev. B **97**, 085104 (2018).

[16] J. P. F. LeBlanc, A. E. Antipov, F. Becca, I. W. Bulik, G. K.-L. Chan, C. M. Chung, Y. Deng, M. Ferrero, T. M. Henderson, C. A. Jimenez-Hoyos, E. Kozik, X. W. Liu, A. J. Millis, N. V. Prokofev, M. Qin, G. E. Scuseria, H. Shi, B. V. Svistunov,

L. F. Tocchio, I. S. Tupitsyn, S. R. White, S. Zhang, B. X. Zheng, Z. Zhu, and E. Gull (Simons Collaboration on the Many-Electron Problem), Solutions of the Two-Dimensional Hubbard Model: Benchmarks and Results from a Wide Range of Numerical Algorithms, Phys. Rev. X **5**, 041041 (2015).

[17] G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin *et al.*, NetKet: A machine learning toolkit for many-body quantum systems, SoftwareX **10**, 100311 (2019).

[18] S. R. White and A. E. Feiguin, Real-Time Evolution Using the Density Matrix Renormalization Group, Phys. Rev. Lett. **93**, 076401 (2004).

[19] G. Vidal, Efficient Simulation of One-Dimensional Quantum Many-Body Systems, Phys. Rev. Lett. **93**, 040502 (2004).

[20] P. Silvi, F. Tschirsich, M. Gerster, J. Jünemann, D. Jaschke, M. Rizzi, and S. Montangero, The tensor networks anthology: Simulation techniques for many-body quantum lattice systems, SciPost Phys. Lect. Notes **8**, 61 (2019).

[21] N. Yoshioka and R. Hamazaki, Constructing neural stationary states for open quantum many-body systems, Phys. Rev. B **99**, 214306 (2019).

[22] L. Banchi, D. Burgarth, and M. J. Kastoryano, Driven Quantum Dynamics: Will it Blend? Phys. Rev. X **7**, 041015 (2017).

[23] M. J. Hartmann and G. Carleo, Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, Phys. Rev. Lett. **122**, 250502 (2019).

[24] A. Nagy and V. Savona, Variational Quantum Monte Carlo Method with A Neural-Network Ansatz for Open Quantum Systems, Phys. Rev. Lett. **122**, 250501 (2019).

[25] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Variational Neural-Network Ansatz for Steady States in Open Quantum Systems, Phys. Rev. Lett. **122**, 250503 (2019).

[26] A. Valenti, G. Jin, J. Léonard, S. D. Huber, and E. Greplova, Scalable Hamiltonian learning for large-scale out-of-equilibrium quantum dynamics, arXiv:2103.01240.

[27] V. Khemani, A. Lazarides, R. Moessner, and S. L. Sondhi, Phase Structure of Driven Quantum Systems, Phys. Rev. Lett. **116**, 250401 (2016).

[28] A. Eckardt, Colloquium: Atomic quantum gases in periodically driven optical lattices, Rev. Mod. Phys. **89**, 011004 (2017).

[29] D. A. Abanin, E. Altman, I. Bloch, and M. Serbyn, Colloquium: Many-body localization, thermalization, and entanglement, Rev. Mod. Phys. **91**, 021001 (2019).

[30] F. Wilczek, Quantum Time Crystals, Phys. Rev. Lett. **109**, 160401 (2012).

[31] D. V. Else, B. Bauer, and C. Nayak, Floquet Time Crystals, Phys. Rev. Lett. **117**, 090402 (2016).

[32] K. Sacha and J. Zakrzewski, Time crystals: A review, Rep. Prog. Phys. **81**, 016401 (2017).

[33] R. Vosk and E. Altman, Dynamical Quantum Phase Transitions in Random Spin Chains, Phys. Rev. Lett. **112**, 217204 (2014).

[34] M. Heyl, Dynamical quantum phase transitions: A review, Rep. Prog. Phys. **81**, 054001 (2018).

[35] 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).

[36] M. P. Zaletel, R. S. K. Mong, C. Karrasch, J. E. Moore, and F. Pollmann, Time-evolving a matrix product state with long-ranged interactions, Phys. Rev. B **91**, 165112 (2015).

[37] A. E. Feiguin and S. R. White, Time-step targeting methods for real-time dynamics using the density matrix renormalization group, Phys. Rev. B **72**, 020404(R) (2005).

[38] J. L. Lado and O. Zilberberg, Topological spin excitations in Harper-Heisenberg spin chains, Phys. Rev. Research **1**, 033009 (2019).

[39] A. Holzner, A. Weichselbaum, I. P. McCulloch, U. Schollwöck, and J. von Delft, Chebyshev matrix product state approach for spectral functions, Phys. Rev. B **83**, 195115 (2011).

[40] D. Hendry, H. Chen, P. Weinberg, and A. E. Feiguin, Chebyshev expansion of spectral functions using restricted Boltzmann machines, arXiv:2103.08804.

[41] I. López Gutiérrez and C. B. Mendl, Real time evolution with neural-network quantum states, arXiv:1912.08831.

[42] D. Hofmann, G. Fabiani, J. H. Mentink, G. Carleo, and M. A. Sentef, Role of stochastic noise and generalization error in the time propagation of neural-network quantum states, arXiv:2105.01054.

[43] M. Schmitt and M. Heyl, Quantum Many-Body Dynamics in Two Dimensions with Artificial Neural Networks, Phys. Rev. Lett. **125**, 100503 (2020).

[44] S. Barison, F. Vicentini, and G. Carleo, An efficient quantum algorithm for the time evolution of parameterized circuits, arXiv:2101.04579.

[45] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, 2016).

[46] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[47] J. Carrasquilla, Machine learning for quantum matter, Adv. Phys.: X **5**, 1797528 (2020).

[48] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[49] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[50] D. Carvalho, N. A. García-Martínez, J. L. Lado, and J. Fernández-Rossier, Real-space mapping of topological invariants using artificial neural networks, Phys. Rev. B **97**, 115453 (2018).

[51] P. M. Vecsei, K. Choo, J. Chang, and T. Neupert, Neural network based classification of crystal symmetries from x-ray diffraction patterns, Phys. Rev. B **99**, 245120 (2019).

[52] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber, Unsupervised identification of topological phase transitions using predictive models, New J. Phys. **22**, 045003 (2020).

[53] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[54] J. F. Rodriguez-Nieva and M. S. Scheurer, Identifying topological order through unsupervised machine learning, Nat. Phys. **15**, 790 (2019).

[55] P. Huembeli, A. Dauphin, P. Wittek, and C. Gogolin, Automated discovery of characteristic features of phase transitions in many-body localization, Phys. Rev. B **99**, 104106 (2019).

[56] A. Valenti, E. van Nieuwenburg, S. Huber, and E. Greplova, Hamiltonian learning for quantum error correction, Phys. Rev. Research **1**, 033092 (2019).

[57] R. Durrer, B. Kratochwil, J. V. Koski, A. J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova, Automated Tuning of

Double Quantum Dots into Specific Charge States using Neural Networks, Phys. Rev. Appl. **13**, 054019 (2020).

[58] L. Huang and L. Wang, Accelerated Monte Carlo simulations with restricted Boltzmann machines, Phys. Rev. B **95**, 035105 (2017).

[59] D. Liu, S.-J. Ran, P. Wittek, C. Peng, R. B. García, G. Su, and M. Lewenstein, Machine learning by unitary tensor network of hierarchical tree structure, New J. Phys. **21**, 073059 (2019).

[60] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems, Phys. Rev. Lett. **124**, 020503 (2020).

[61] R. Fournier, L. Wang, O. V. Yazyev, and Q. S. Wu, Artificial Neural Network Approach to the Analytic Continuation Problem, Phys. Rev. Lett. **124**, 056401 (2020).

[62] H. Yoon, J.-H. Sim, and M. J. Han, Analytic continuation via domain knowledge free machine learning, Phys. Rev. B **98**, 245101 (2018).

[63] A. Weiße, G. Wellein, A. Alvermann, and H. Fehske, The kernel polynomial method, Rev. Mod. Phys. **78**, 275 (2006).

[64] D. Jackson, On approximation by trigonometric sums and polynomials, Trans. Am. Math. Soc. **13**, 491 (1912).

[65] F. A. Wolf, I. P. McCulloch, O. Parcollet, and U. Schollwöck, Chebyshev matrix product state impurity solver for dynamical mean-field theory, Phys. Rev. B **90**, 115124 (2014).

[66] M. Ganahl, P. Thunström, F. Verstraete, K. Held, and H. G. Evertz, Chebyshev expansion for impurity models using matrix product states, Phys. Rev. B **90**, 045144 (2014).

[67] V. Kaskela and J. L. Lado, Dynamical topological excitations in parafermion chains, Phys. Rev. Research **3**, 013095 (2021).

[68] J. L. Lado and M. Sigrist, Solitonic in-gap modes in a superconductor-quantum antiferromagnet interface, Phys. Rev. Research **2**, 023347 (2020).

[69] M. Rösner and J. L. Lado, Inducing a many-body topological state of matter through Coulomb-engineered local interactions, Phys. Rev. Research **3**, 013265 (2021).

[70] V. Vadimov, T. Hyart, J. L. Lado, M. Möttönen, and T. Ala-Nissila, Many-body Majorana-like zero modes without gauge symmetry breaking, Phys. Rev. Research **3**, 023002 (2021).

[71] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. (OTexts, Melbourne, Australia, 2018), OTexts.com/fpp2.

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

[73] T. Giamarchi, *Quantum Physics in One Dimension* (Oxford University Press, Oxford, 2003).

[74] A. Lamacraft and P. Fendley, Order Parameter Statistics in the Critical Quantum Ising Chain, Phys. Rev. Lett. **100**, 165706 (2008).

[75] V. Eisler, Z. Rácz, and F. van Wijland, Magnetization distribution in the transverse Ising chain with energy flux, Phys. Rev. E **67**, 056129 (2003).

[76] V. Gritsev, E. Altman, E. Demler, and A. Polkovnikov, Full quantum distribution of contrast in interference experiments between interacting one-dimensional Bose liquids, Nat. Phys. **2**, 705 (2006).

[77] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. **15**, 1929 (2014).