Liouville-space neural network representation of density matrices

Simon Kothe and Peter Kirton

Department of Physics and SUPA, University of Strathclyde, Glasgow G4 0NG, United Kingdom

Neural network quantum states such as ansatz wave functions have shown a great deal of promise for finding the ground state of spin models. Recently, work has focused on extending this idea to mixed states for simulating the dynamics of open systems. Most approaches so far have used a purification ansatz where a copy of the system Hilbert space is added, which when traced out gives the correct density matrix. Here we instead present an extension of the restricted Boltzmann machine which directly represents the density matrix in Liouville space. This allows the compact representation of states which appear in mean-field theory. We benchmark our approach on two different versions of the dissipative transverse-field Ising model, which show our ansatz is able to compete with other state-of-the-art approaches.

I. INTRODUCTION

Developing techniques to study the many-body physics of open quantum systems presents the possibility to access behavior beyond that possible in equilibrium. This is relevant for understanding the dynamics of a variety of experimental platforms, but it also allows us to address fundamental questions about what kinds of physics is realizable in situations where incoherent driving and dissipation compete with coherent Hamiltonian dynamics. For example, in arrays of superconducting qubits, coherent hopping of excitations can compete with on-site losses [1] or in semiconductor microcavities nonlinearities in the Hamiltonian can compete with photon losses [2].

Performing accurate simulations of the models describing this kind of physics can be very challenging. Even in the simplest case, where the dynamics of the system is well captured by a Markovian master equation [3], the effective size of the space required grows exponentially with the number of degrees of freedom included. This makes it difficult to make concrete statements about what phases are stable for large system sizes, in the thermodynamic limit. Thus, developing new techniques that are able to capture all of the required physical processes is a key challenge to be overcome.

For closed systems a variety of techniques are available, allowing calculation of both ground states and dynamics in a variety of situations. These range from density-functional theory, which relies on approximations of the electronic density [4], to Monte Carlo methods [5–9], which draw a small number of samples from the Hilbert space approximating the correct distribution, and tensor network methods [10–12],

which rely on compression of the full many-body wave function.

Recently, there has been much progress in using neural networks to accurately represent wave functions of many-body systems [13–15]. These neural network quantum states (NNQSs) rely on the ability of neural networks to learn and represent any sufficiently smooth function [16–18], allowing an efficient description without requiring an exponential number of parameters. The simplest architecture used for NNQSs is the restricted Boltzmann machine (RBM). They consist of two fully connected layers, one visible and one hidden. This simple structure of the architecture translates to a function which can be easily implemented numerically, allowing for an efficient Markov sampling of the relevant probability distribution [19,20] and efficient optimization. By increasing the hidden unit density, the expressibility of the ansatz increases, allowing it to represent any state in the limit of infinite parameters. This has already shown to be competitive with other state-of-the-art approaches for finding ground [13] and excited [21] states of one-dimensional (1D) and 2D spin systems. This ansatz has been applied to a wide range of problems in this area, from simulating topologically complex states [22] to investigating how the eigenvalue spectrum of the quantum Fisher matrix gives information about how networks learn ground states [23]. Restricted Boltzmann machines are able to represent some states with a volume-law entropy scaling [24].

Architectures beyond the RBM have also been used to study related problems. These range from recurrent neural networks [25,26] to convolutional neural networks [27–29] and deep autoregressive models, which forgo the need for Monte Carlo sampling altogether [30].

These approaches have also begun to be applied to the dynamics and steady states of open quantum systems. Neural density machines (NDMs) [31–34] use a pair of copies of a pure-state NNQS coupled together by an extra layer of hidden neurons. The density matrix of interest is then obtained by tracing out the degrees of freedom associated with one of

the copies. Such purification schemes have been used also in tensor network approaches [11,35,36]. Other developments use a positive-operator-valued-measure representation of the density matrix, achieving accurate results [37–39] as well as autoregressive Gram-Hadamard density operators which extend the NDM by adding additional layers which allows the capture of more complex correlations [40]. Another ansatz was introduced by Yoshioka and Hamazaki [41] which uses a binary encoding to vectorize their density matrix. So far, the literature has focused mostly on purifying the density matrix in Hilbert space, with the exception of Ref. [41]. Here we propose a different approach, which writes the density matrix directly in Liouville space. The Liouville density machine ansatz which we propose does not require additional visible units and maintains a correspondence of the visible layer and the concrete physical system. We show that this is able to efficiently represent a larger range of physically relevant states than the NDM, improving learning and accuracy while retaining the simplicity of RBMs.

This paper is organized as follows. In Sec. II we give a brief description of the Lindblad master equation which describes Markovian open-system dynamics. Section III details how finding the steady state of this equation can be recast in terms minimization of a cost function and how it can be estimated using Monte Carlo sampling. In Sec. IV we give a brief introduction to NNQSs, using the original ansatz of Ref. [13], going on to show how this can be generalized to open systems. Section V gives a detailed comparison of the different approaches to solve this problem, using two versions of the dissipative transverse-field Ising model as a benchmark. We gain insights into which kinds of states are easy and difficult to represent efficiently with the architectures described. In Sec. VI we give our conclusions and discuss possible ways to build upon these results. The Appendixes contain a detailed derivation of the stochastic reconfiguration algorithm and an explanation of how we produce the Markov chains for estimating the required expectation values from the neural network.

II. OPEN-SYSTEM DYNAMICS

The presence of an external environment fundamentally changes the nature of the dynamics of a quantum system. The state is no longer captured by a wave function and the nonunitary dynamics cannot be described by the Schrödinger equation. Instead, the natural description is in terms of a reduced density operator for the system, ρ, and a master equation which defines its time evolution. This then allows for energy dissipation, decoherence, and, most importantly for the present paper, the relaxation into a nonequilibrium steady state in the long-time limit.

If the coupling to the environment is weak and structureless, one can assume that system-induced correlations within the bath decay faster than the effects of the bath on the system. This leads to the particularly simple Lindblad master equation [3], which takes the form

$$\frac{d\rho}{dt} = \mathcal{L}\rho = -i[H, \rho] + \sum_i \gamma_i \mathcal{D}[A_i]\rho. \quad (1)$$

Here the sum runs over the dissipation channels i, which are defined by the jump operators A_i and rates γ_i. The dissipation superoperators are given by the Lindblad form

$$\mathcal{D}[A]\rho = A\rho A^\dagger - \frac{1}{2}(A^\dagger A\rho + \rho A^\dagger A). \quad (2)$$

We only consider time-independent master equations.

The time evolution of the density operator is then governed by the formal expression

$$\rho(t) = e^{\mathcal{L}t}\rho(0) \quad (3)$$

such that in the long-time limit the stationary state satisfies

$$\lim_{t\to\infty} \mathcal{L}\rho(t) = 0. \quad (4)$$

To simplify what follows both mathematically and numerically we recast Eq. (1) in Liouville space (sometimes also referred to as Choi's space). In this space the density matrix is reshaped into a vector,

$$\rho = \sum_m \sum_n \rho_{m,n} |m\rangle \langle n| \to |\rho\rangle\rangle = \sum_{m,n} \rho_{m,n} |m, n\rangle\rangle. \quad (5)$$

Here the density matrix is expanded in the basis $\{|m\rangle\}$ with expansion coefficients $\rho_{m,n}$. The double-ket notation $|\cdot\rangle\rangle$ used above denotes a vectorized operator which exists in a Hilbert space consisting of two copies of the original, $|\rho\rangle\rangle \in \mathcal{H} \otimes \mathcal{H}$. Then superoperators which act on these operator kets are elements of Liouville space $L \in (\mathcal{H} \otimes \mathcal{H})^* \otimes (\mathcal{H} \otimes \mathcal{H})$ [41,42]. Here we often abbreviate the double index (m, n) with a single index s which runs over the ket and bra indices of the density matrix $\sum_{m,n} \rho_{m,n} |m, n\rangle\rangle \equiv \sum_s \rho(s) |s\rangle\rangle$. With this Eq. (1) now reads

$$\frac{d}{dt} |\rho\rangle\rangle = L |\rho\rangle\rangle, \quad (6)$$

where

$$L = -i(H \otimes \mathbb{1} - \mathbb{1} \otimes H^T) + \sum_k \left\{ A_k \otimes A_k^* - \frac{1}{2}[\mathbb{1} \otimes (A_k^\dagger A_k)^T + (A_k^\dagger A_k) \otimes \mathbb{1}] \right\}$$

is the matrix form of the superoperator $\mathcal{L}$ that acts from the left on a density ket $|\rho\rangle\rangle$.

We can then use standard linear algebra results to find the eigenvalues and eigenkets of the matrix L such that

$$L |\rho_i\rangle\rangle = \lambda_i |\rho_i\rangle\rangle. \quad (7)$$

A stationary state has $\lambda_i = 0$, while all other states have $\text{Re}\lambda_i < 0$.

As we have seen above, the size of the required Liouville space scales much more quickly than the already exponentially growing Hilbert space. For example, for spin-1/2 lattice problems, the state vector of a closed system grows as $O(2^N)$, with the system size N. The density matrix, on the other hand, grows as $O(4^N)$ and hence the Liouvillian has up to $O(16^N)$ elements which limits the size of accessible systems even further. In the following sections we will detail our approach to tackling this problem.

III. FINDING THE STEADY STATE

Many routes to accessing larger system sizes with numerical simulations are variational methods [43,44]. For example, tensor network methods [45] can be seen as variationally finding the steady state by optimizing over the set of states captured by a given tensor network architecture, while the corner space renormalization-group method [46] optimizes a particular (small) set of basis states which can be used to build up to larger system sizes. As we will see later, neural network techniques make use of the same underlying mathematical structures. Crucial to these methods is the use of an ansatz function which specifies the full state with a finite number of parameters, which we denote by α. We may then write $\rho(s;t) \rightarrow \rho_\alpha(s;t)$ and optimize over α to find the closest representation of the steady state within this class. For this approach to be an efficient solution, the number of parameters must grow at most polynomially with the system size.

To find the steady state $\lim_{t\to\infty} \rho(t) \equiv \rho_{SS}$, we require a cost function which measures how close the ansatz is to the true stationary density operator. To do this we make use of the fact that $L|\rho_{SS}\rangle\rangle = 0$ and that for all other states this quantity is finite. This gives us a cost function to optimize as well as a metric for how well the variational state parametrizes the true steady state. The cost function we wish to optimize is given by the quantity

$$|C_\alpha|^2 = \left| \frac{\langle\langle \rho_\alpha | L | \rho_\alpha \rangle\rangle}{\langle\langle \rho_\alpha | \rho_\alpha \rangle\rangle} \right|^2, \tag{8}$$

which uniquely vanishes in the steady state. Note that the vectors appearing above are in Liouville space; hence the denominator involves the sum over all elements of the density matrix $\langle\langle \rho_\alpha | \rho_\alpha \rangle\rangle = \sum_s |\rho_\alpha(s)|^2$. Expanding this cost function yields a form that can be easily evaluated via the Markov chain Monte Carlo

$$\begin{aligned} C_\alpha &= \frac{\langle\langle \rho_\alpha | L | \rho_\alpha \rangle\rangle}{\langle\langle \rho_\alpha | \rho_\alpha \rangle\rangle} \\ &= \frac{1}{\sum_{s'} |\rho_\alpha(s')|^2} \sum_{s,s'} \rho_\alpha^*(s)\rho_\alpha(s') \langle\langle s|L|s'\rangle\rangle \\ &= \frac{1}{\sum_{s'} |\rho_\alpha(s')|^2} \sum_{s,s'} \frac{|\rho_\alpha(s)|^2 \rho_\alpha(s')}{\rho_\alpha(s)} \langle\langle s|L|s'\rangle\rangle \\ &= \sum_s p(s;\alpha) C_{loc}(s;\alpha), \end{aligned}$$

where

$$p(s;\alpha) = \frac{|\rho_\alpha(s)|^2}{\sum_{s'} |\rho_\alpha(s')|^2} \tag{9}$$

defines a probability distribution over the entire density matrix which we can draw samples from. The local cost associated with each element of this distribution is then

$$C_{loc}(s;\alpha) = \sum_{s'} \langle\langle s|L|s'\rangle\rangle \frac{\rho_\alpha(s')}{\rho_\alpha(s)}. \tag{10}$$

All parts of this local cost can be efficiently calculated. The states s' which connect, via the matrix element of the Liouvillian, to the original state s are generated during the

Metropolis-Hastings step and can be accessed at any later stage of the process (see Appendix B for details on the algorithm used). Since, for a local Liouvillian, the number of nonzero elements grows only linearly in system size, the evaluation of the matrix elements of L can also be done efficiently. The samples we draw follow the distribution $p(s;\alpha)$ and so the cost can be calculated as a simple mean over the local costs [20]

$$C_\alpha \approx \frac{1}{N_s} \sum_s C_{loc}(s;\alpha), \tag{11}$$

where N_s is the number of samples and the sum runs over the Monte Carlo samples.

Along with an estimate for the cost function we also need a way of updating the parameter α such that the state we find is optimized. The simplest way to do this is via stochastic gradient ascent (SGA), where the gradients of C_α are also estimated with the Monte Carlo samples and at each step in the simulation the parameters are updated as

$$\alpha \rightarrow \alpha' = \alpha + \eta \nabla_\alpha C_\alpha, \tag{12}$$

with the learning rate η. There are several problems with this approach, e.g., it has been shown [23] that SGA has severe problems with steep energy surfaces. The main problem for the present case is that L also has a left eigenstate with eigenvalue 0,

$$\langle\langle \mathbb{T} | L = 0. \tag{13}$$

This is the trace state, which is defined as

$$\langle\langle \mathbb{T} | \rho \rangle\rangle = \text{Tr}(\rho), \tag{14}$$

and since the dynamics of any physical master equation is necessarily trace preserving we find the result above. This means that there is another state which optimizes the cost function. A solution to both of these problems is to instead use stochastic reconfiguration (SR) [20,23,47,48] to update the parameters. Stochastic reconfiguration can be derived by asking which parameter update γ,

$$\alpha \rightarrow \alpha' = \alpha + \eta\gamma, \tag{15}$$

best approximates a step in real time (see Appendix A for a derivation). Since the trace state cannot be found by a real-time evolution generated by L, this guarantees that we will find the correct steady state when optimizing the cost [Eq. (8)]. Furthermore, SR takes into account the curvature of the energy landscape, speeding up the optimization on flat areas and slowing down in the presence of strong curvature. The result is that the updates are calculated as

$$\gamma = S^{-1} f, \tag{16}$$

where $S_{i,j} = \langle \mathcal{O}_i^* \mathcal{O}_j \rangle - \langle \mathcal{O}_i^* \rangle \langle \mathcal{O}_j \rangle$ is the quantum Fisher matrix and $f_i = \langle \mathcal{O}_i^* L \rangle - \langle \mathcal{O}_i^* \rangle \langle L \rangle$ is the gradient of the cost function in Eq. (8). The angular brackets denote the expectation value over the Monte Carlo samples and the operator $\mathcal{O}_i(s)$ is the logarithmic derivative of $\rho_\alpha(s)$ with respect to the ith parameter

$$\mathcal{O}_i(s) = \frac{1}{\rho_\alpha(s)} \frac{\partial}{\partial \alpha_i} \rho_\alpha(s). \tag{17}$$
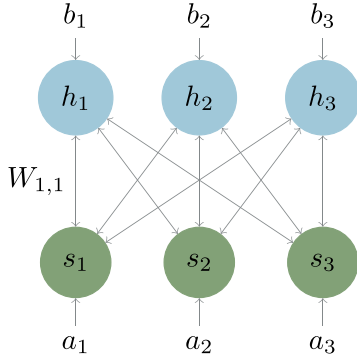
FIG. 1. Schematic representation of an RBM, showing the visible units (lower row, green) connected to the hidden units (upper row, blue) via the weight matrix (double-headed arrows). The single-headed arrows labeled by a_i and b_i show the biases applied to the neurons.

The matrix S is defined to be positive; however, when estimating it via Monte Carlo sampling it can happen that some eigenvalues vanish and S becomes singular. One can work around this problem by either calculating the pseudoinverse or adding a small regularization $\lambda \approx 10^{-3}$ to the diagonal. We employ the latter method here.

We next need to specify the form of our ansatz function for the density operator ρ_α. To do this we first give a brief overview of how neural networks can be used to represent many-body wave functions.

IV. NEURAL NETWORK QUANTUM STATES

Neural network quantum states are a variational ansatz whose parameters and functional form are defined by an underlying neural network architecture. This approach has been recently developed to describe the many-body wave function of interacting spin systems and was introduced in Ref. [13]. The key insight here is that the heart of the many-body problem is to find the relevant parts of Hilbert space in which, e.g., the ground state of a system exists. This is essentially a problem of dimensionality reduction and feature extraction, which are the two strongest points of neural networks.

Neural network quantum states use this ability of neural networks to efficiently represent any continuous function [16–18], e.g., a wave function [13,21,22] or density matrix [32–34], by learning the most important features which define that function. The simplest NNQS architecture is the RBM. These are bilayer neural networks with one visible and one hidden layer of neurons. The network is parametrized by a bias attached to each neuron and a set of weights which connects the two layers. This setup is shown schematically in Fig. 1. In the same spirit as the discussion in the preceding section, it is possible to then write a variational ansatz for a wave function for a spin-1/2 lattice model. This takes the form

$$\Psi_\alpha(s) = \sum_{\{h_i\}} \exp\left(\sum_j a_j s_j + \sum_i b_i h_i + \sum_{ij} W_{ij} h_i s_j \right) \quad (18)$$

$$= \exp\left(\sum_j a_j s_j \right) \prod_{i=1}^M 2\cosh\left(b_i + \sum_j W_{ij} s_j \right), \quad (19)$$

where in Eq. (19) we use the fact that the sum over configurations can be analytically calculated. Here a_i (b_j) denotes the visible (hidden) bias of the ith site (jth hidden unit) and W_{ij} the weights connecting the jth visible unit to the ith hidden unit. In addition, M (N) denotes the number of hidden (visible) units. The ratio between hidden and visible units is $\beta = M/N$. For $\beta = 0$ the only states which can be described are those that would arise in a mean-field description, i.e., those which are a product of single-site wave functions. Increasing the value of β allows a systematic increase in the amount of entanglement which can be described [13]. A further strength of the RBM wave function is the simplicity of its derivatives with respect to the parameters. This allows for efficient evaluation of all of the derivatives required to update the state.

In its simplest form the RBM ansatz described above is only able to discriminate between two different visible states $s = [-1, 1]$, which allows at most for the representation of pure states of spin-1/2 systems. However, this is not sufficient to describe a vectorized density operator in Liouville space, which, even for a spin-1/2 system, has four elements. To overcome this issue we use an approach based on that introduced in Ref. [19] for the study of pure states of spin-1 lattices. This is achieved by adding an additional set of biases and weights which allows discrimination between the $s = [-1, 0, 1]$ states.

We can adapt this idea to provide a basis for a NNQS ansatz for density kets $|\rho\rangle\rangle$. Such an ansatz needs to be able to discriminate between the four states of the local density operator while maintaining a one-to-one correspondence between the visible layer and the physical system. To label the density matrix elements on each site we choose to use the mapping

$$s = \begin{cases} 2 \to |\uparrow\uparrow\rangle\rangle \\ 1 \to |\uparrow\downarrow\rangle\rangle \\ -1 \to |\downarrow\uparrow\rangle\rangle \\ -2 \to |\downarrow\downarrow\rangle\rangle, \end{cases} \quad (20)$$

where $s = \pm 2$ denote the diagonal density matrix elements, while $s = \pm 1$ are the off-diagonal elements. These values are chosen as they give a simple mapping from integers to density matrix elements and allow our neural network ansatz to easily distinguish between all of the states. To discriminate these four different local states, we add one more set of visible biases and weights, allowing us to represent a density ket. Furthermore, we do not require any additional visible nodes to achieve this, as in Ref. [41]. The total ansatz function now reads

$$\rho_\alpha(s) = \exp\left(\sum_j^N a_j^{(1)} s_j + a_j^{(2)} s_j^2 + a_j^{(3)} s_j^3 \right)$$

$$\times \prod_{i=1}^M 2\cosh\left(b_i + \sum_j^N U_{i,j} s_j + V_{i,j} s_j^2 + W_{i,j} s_j^3 \right), \quad (21)$$
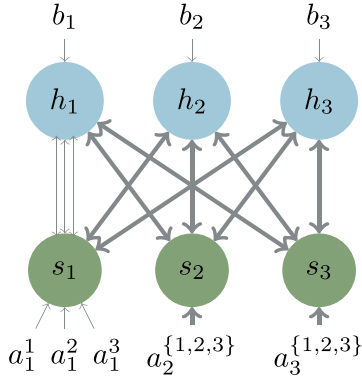
FIG. 2. Schematic representation of an LDM. Similar to the RBM, it has visible (lower row, green) and hidden (upper row, blue) units. The LDM, however has three visible biases $a_i^{\{1,2,3\}}$ as well as three weight matrices U, V, and W (double arrows). The weight matrices couple to different powers of the configuration vector s and encode correlations between different local states.

where N and M are the numbers of visible and hidden nodes, respectively. We have introduced a set of complex biases $a^{(n)}$ to the visible units, which allow us to distinguish different visible biases; b again gives the bias on each hidden neuron and U, V, and W are the complex weight matrices that connect the two layers. Theses biases and connections are shown schematically in Fig. 2. Similar to the pure-state case of Eq. (18), this ansatz has labeling freedom and very simple derivatives which can be found analytically and efficiently implemented. Calculating the derivatives of a complex-valued and complex parametrized function is done using Wirtinger calculus [49].

If the hidden unit density $\beta = 0$, this ansatz is able to exactly capture those states described by mean-field theory, i.e., where the full density matrix can be factorized onto individual sites. This then allows the parameter β to systematically increase the number of correlations which can be represented in a similar way to the bond dimension of a matrix product state (MPS).

This ansatz, which we refer to as a Liouville density machine (LDM), bears some similarities to that proposed in Ref. [41]. In both cases the problem is transformed into Liouville space, but in Ref. [41] this is achieved by adding an additional visible unit on each site, allowing them to represent the four required states. A detailed comparison between the expressibility of these approaches is left as an open question. The LDM ansatz should be seen in contrast to those in Refs. [31–34]. In these papers a purification ansatz is used and an extra layer of hidden units is added to represent the state of an auxiliary system which, when traced over, leaves the appropriate density matrix for the original system. Similar techniques have been applied to MPS simulations [35,36]. In what follows we refer to this purification ansatz as a neural density machine (NDM).

Compared to the purification approaches of Refs. [31–34], the LDM is not guaranteed by construction to be either Hermitian or positive definite. However, we find that this is not a significant issue in our simulations because of our choice of SR as the descent algorithm. Note that any matrix can be represented in the eigenbasis of the Liouvillian [42] as

$$|A\rangle\rangle = c_0 |\rho_0\rangle\rangle + \sum_{i\neq 0} c_i |\rho_i\rangle\rangle, \tag{22}$$

where c_i are expansion coefficients and $|\rho_i\rangle\rangle$ are the vectorized eigenmatrices of L. Under real-time evolution, all eigenvalues of eigenstates except for the steady state $|\rho_0\rangle\rangle$ have a real part $\mathrm{Re}(\lambda_{i\neq 0}) < 0$ and hence exponentially decay over time. Hence, *any* initial matrix, independent of its physicality, must necessarily converge to the steady state of the Liouvillian. We stress now that, up to first order, the optimization algorithm we employ was derived as an approximation to real-time evolution under a Liouvillian and hence gives a good approximation to this kind of behavior.

V. RESULTS

To benchmark and understand the strengths and limitations of the LDM ansatz, we study the stationary state of the 1D dissipative transverse-field Ising (TFI) model [34,39,41,50–52]. The system consists of a chain of spin-1/2 particles. The Hamiltonian part of the evolution is governed by

$$H = \frac{J}{4} \sum_i^{N-1} \sigma_i^z \sigma_{i+1}^z + \frac{h}{2} \sum_i^N \sigma_i^x. \tag{23}$$

Here N is the number of sites, σ are the usual Pauli matrices, J is the interaction strength, and h is the strength of the transverse field. The dissipation is governed by excitation loss on each site, so the jump operators which appear in the Liouvillian are $A_i = \sqrt{\gamma}\sigma_i^-$. We choose the units of the interaction and field such that the dissipation strength is $\gamma = 1$. In all calculations that follow we set the interaction strength to $J/\gamma = 2$.

The steady state of this model then has simple solutions in two limiting cases. When $h \to 0$ the dissipation dominates the dynamics and the stationary state is a pure product state with all the spins pointing down

$$\lim_{h\to 0} \rho_{\mathrm{SS}} = \bigotimes^N |\downarrow\rangle \langle\downarrow|. \tag{24}$$

In the opposite limit where $h \to \infty$ there is only competition between the local on-site field and the dissipation and so the steady state ends up again as a product state, but this time the state on each site is mixed

$$\lim_{h\to\infty} \rho_{\mathrm{SS}} = \bigotimes^N \frac{1}{2}(|\uparrow\rangle \langle\uparrow| + |\downarrow\rangle \langle\downarrow|). \tag{25}$$

At intermediate values of h the steady state interpolates between these two, building up complex long-range classical and quantum correlations.

We will compare the results of the LDM ansatz to those obtained using the NDM approach as implemented in the NETKET library [53]. For small system sizes we will also be able to compare to results obtained from exact diagonalization, which are calculated using QUTIP [54].

As the first example we look at the case of a small system with $N = 6$. Our findings are summarized in Fig. 3. In each case we randomly initialize the parameter values and at each step we take Monte Carlo samples to approximate the cost
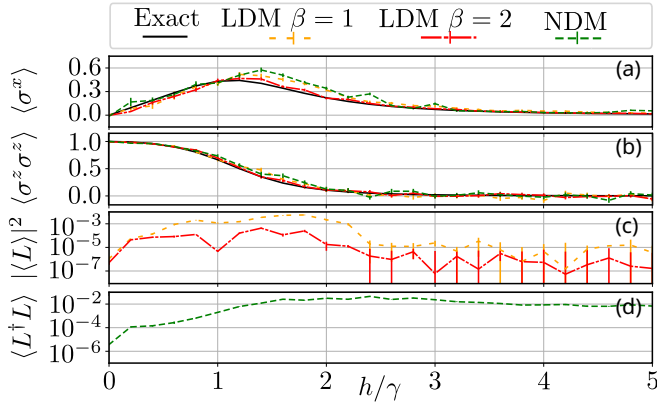
FIG. 3. Comparison of the steady state of Eq. (23) using both the NDM and LDM approaches. The system size $N = 6$ is small enough that exact diagonalization is possible. (a) Expectation value of σ_i^x as a function of h. (b) Expectation value of the $\sigma_i^z \sigma_{i+1}^z$ correlation function. In both cases the expectation value is taken on the central site(s). The black solid lines show the exact result, the NDM with $\beta = 1$ is shown by the green dashed lines, and the LDM with $\beta = 1$ is shown by the orange dotted lines and with $\beta = 2$ by the red dot-dashed lines. (c) Absolute value squared of the cost function in Eq. (8) for the two LDM results. (d) Expectation value $\langle L^\dagger L \rangle$ used as a cost function for the NDM ansatz employed by NETKET. For both cases with $\beta = 1$ we used 4500 samples and optimized for 1000 steps with a learning rate of $\eta = 10^{-2}$ and a regularization of $\lambda = 10^{-2}$; for the expectation values we used 500 diagonal samples. In the $\beta = 2$ case we used 6000 samples, 2000 steps, and 800 diagonal samples. The other meta parameters were the same in both cases.

function and the best updates to the parameters. At the end of each run we produce a new Markov chain, but this time sample from a probability distribution which follows the diagonal of the density matrix. This allows us to estimate the expectation value of observables of interest. Since there are fewer diagonal states than entries in the full density matrix, we usually only need about 500–800 diagonal samples.

In Figs. 3(a) and 3(b) we show the expectation value of σ_i^x on the central site and the $\sigma_i^z \sigma_{i+1}^z$ correlation function on the central pair of sites, respectively, as a function of the field strength h. This gives a good indication of how well the various approaches are able to produce single-site observables. We see that, in general, there is good agreement between the exact results and those obtained using both the LDM and NDM approaches. In all cases the agreement is worst in the central region where $1 \leqslant h/\gamma \leqslant 2.5$, which is in agreement with the results of Ref. [34]. For the LDM a hidden unit density of $\beta = 1$ corresponds to 132 parameters; for the NDM this is 174 parameters. We see that even with only 3/4 of the parameters the LDM generally gives results as good as or better than the NDM. By increasing the number of hidden units in the LDM we also increase the number of parameters so that, at $\beta = 2$, there are 246 parameters. We see that for the LDM ansatz, increasing the value of β and hence the number of parameters used is able to significantly decrease the deviation from the exact result; thus we may use the hidden unit density as a way of checking for convergence when exact results are no longer possible. We can also see this effect more clearly in Figs. 3(c) and 3(d), where we show the Monte
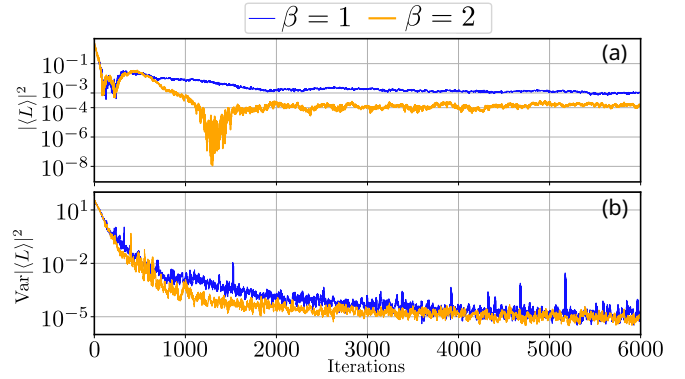

FIG. 4. Convergence of the LDM ansatz for two different hidden unit densities $\beta = 1$ (thin blue line) and $\beta = 2$ (thick orange line): (a) running estimate for the cost function and (b) variance in the same quantity. Increasing the hidden unit density improves the accuracy of the results. Both calculations were done at $h/\gamma = 1$ using the same parameters as in Fig. 3.

Carlo estimated cost function for each ansatz. The value of the cost function is significantly decreased at all values of h when β is increased. We also see that for the LDM ansatz the cost function is at a maximum in the regions where the convergence to the steady state is worst, which allows us to use this estimate to again judge the accuracy of our results for system sizes where exact methods are unavailable. This is not true of the NDM approach, where the cost function reaches a maximum at intermediate values of h and does not significantly decrease as h increases further. This is because the mixed product state, described in Eq. (25), is not so easy to represent in a purification ansatz; this mixed state requires a large amount of entanglement between the real and auxiliary spins. This is not the case for the LDM approach, which can represent this state exactly without using hidden units.

A further comparison is shown in Fig. 4. This figure shows how the cost function of the LDM evolves for two different values of β over 6000 steps at one of the most difficult points $h = \gamma$. By increasing the number of variational parameters from 132 to 246 we were able to reduce the cost function by an order of magnitude. We also see that simply checking the value of the cost function does not give an accurate stopping condition for the algorithm. After around 1400 steps the cost function for $\beta = 2$ is very small but the variance is quite large. This means that the LDM has not found an eigenstate of the Liouvillian but is still giving a small value for the cost function. We propose that a condition based on a combination of both of these quantities can give a good way to automatically stop the learning process when a good approximation to the steady state has been reached.

To test that the state we obtain is physical, we construct the full density matrix from the LDM. In Fig. 5(a) we show how the real part of the smallest eigenvalue of this constructed density matrix evolves. For a randomly initiated state the minimum eigenvalue is negative, which indicates a nonphysical density matrix. However, as the optimization goes on the minimum eigenvalues become closer to 0 or becomes positive, indicating that the final density matrix is positive semidefinite. Figure 5(b) shows that the imaginary parts of all of the
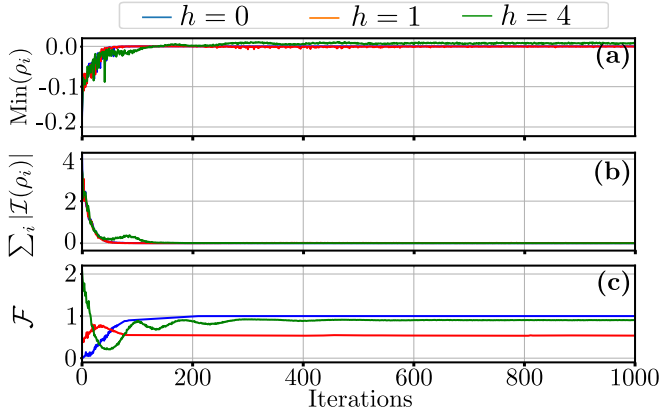
FIG. 5. Properties of the state obtained by using SR at three different values of h and all other parameters the same as the $\beta = 1$ results in Fig. 3. (a) Real part of the minimum eigenvalue of the full density matrix obtained from the LDM. (b) Sum of the absolute value of the imaginary parts of the eigenvalues of the density matrix. (c) Fidelity of the ansatz with the exact density matrix.

eigenvalues are quickly suppressed, which indicates a Hermitian matrix, and hence we can produce a physical density matrix. In Fig. 5(c) we show the fidelity of the state with one obtained from the exact solution of the master equation. We see that, as expected from the results of Fig. 3, the fidelity is highest in the cases where the transverse-field strength is either very large or very small and the fidelity is lowest where the steady state has large entanglement around $h = 1\gamma$.

We now go on to examine how the accuracy of these approaches scales to larger system sizes. At $N = 16$ it becomes difficult to use exact methods to compare against; however, this model is straightforward to solve with MPS simulations, which we found to be fully converged for a bond dimension of $\chi = 7$. Results of these calculations are shown in Fig. 6. The number of parameters for the NDM is 1104, while the LDM has 1126. For reference, a bond dimension of $\chi = 7$ corresponds to 1952 matrix elements in the MPS. We see behavior very similar to that for the $N = 6$ case; both approaches are more difficult to converge in the region of intermediate h/γ and the cost function for the LDM has a peak in this region. In Fig. 7 we show how the convergence can again be improved by increasing the hidden unit density. Here we choose $h = 2\gamma$, as this is the point where the convergence is worst. We see that as β is increased the cost function decreases towards zero and the expectation value moves towards that found in the MPS simulation. The expectation value here is a two-point correlation function, which in general is harder to converge than single-site operators.

A. TFI model with rotated Hamiltonian

By making a simple change to the model discussed above, it is possible to make the convergence of both neural network approaches considerably worse. To do this we change the Hamiltonian to a rotated basis [50,52,55]

$$H = \frac{J}{4} \sum_{i=0}^{4} \sigma_i^x \sigma_{i+1}^x + \frac{h}{2} \sum_{i=0}^{5} \sigma_i^z \qquad (26)$$
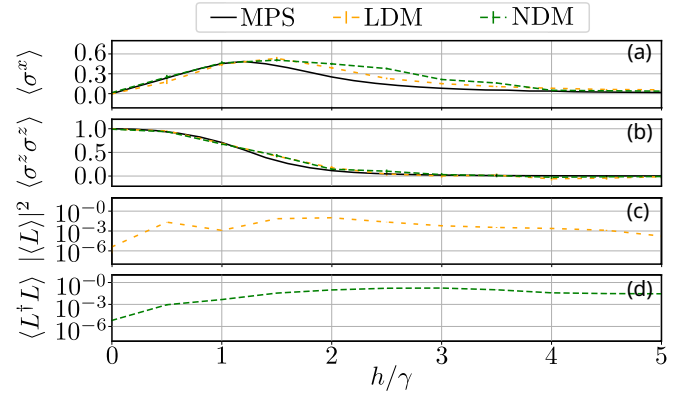

FIG. 6. Steady state of Eq. (23) as a function of field strength, similar to Fig. 3, for a larger system size $N = 16$. Shown are the expectation values of (a) σ_i^x and (b) $\sigma_i^z \sigma_{i+1}^z$ on the central site(s) and (c) and (d) the relevant cost functions. The LDM (orange dotted lines) and NDM (green dashed lines) results are compared to those obtained from MPS simulations (black solid lines). We used $\beta = 1.4$ in the case of the LDM and $\beta = 1$ for the NDM to ensure that both approaches use a similar number of parameters. The NDM has 1104 parameters, while the LDM has 1126. In both cases we evolved for 7000 steps with a learning rate of $\eta = 10^{-3}$, we took 9000 Monte Carlo samples at each step and a regularization of $\lambda = 3 \times 10^{-3}$. We used 800 diagonal samples to estimate the expectation values.

but keep the dissipation processes the same as for the previous model. In this case the dissipation does not explicitly break the $\mathbb{Z}_2$ symmetry of the model as the interaction term is perpendicular to the dissipation. Therefore, the competition between the coherent and dissipative dynamics gives rise to complex correlations in the steady state. This leads to a very rich mean-field phase diagram in high dimensions with possibilities for both first- and second-order phase transitions between different magnetic orderings [51,55,56]. In one dimension these phase transitions turn into continuous crossovers, but complex correlations still build up when $h \sim J \sim \gamma$. (For a detailed
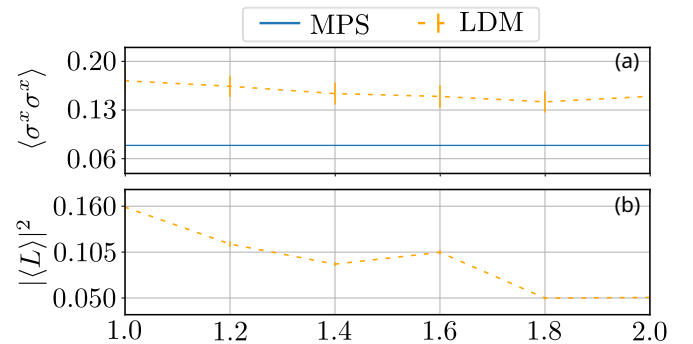

FIG. 7. Improvement of convergence as a function of β for the $N = 16$ TFI model at $h = 2\gamma$ (orange dotted line) compared to the results of an MPS calculation (blue solid line). (a) The $\sigma_i^x \sigma_{i+1}^x$ correlation function and (b) the estimate for the cost function. All calculations ran for 7000 steps. To accommodate for higher parameter counts we increased the number of samples with β from 9000 at $\beta = 1$ to 17 000 at $\beta = 2$. The other parameters are the same as in Fig. 6.
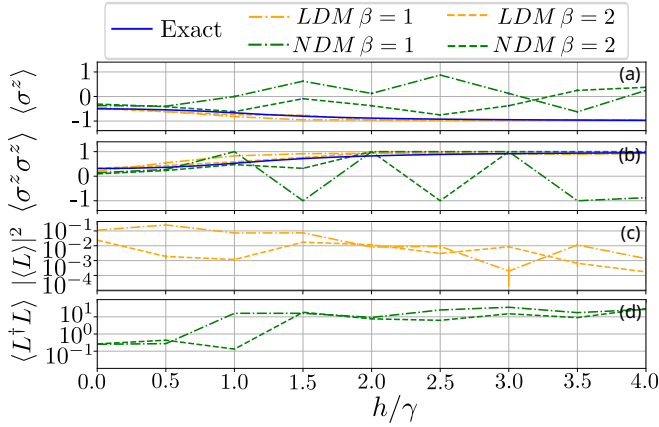
FIG. 8. Optimizing the rotated TFI model as in Eq. (26) for $N = 6$. The exact results are in blue, those obtained with the LDM are orange, and those for the NDM are green. Different hidden unit densities are shown by different line styles. In all cases the optimization was run for 4000 steps, with a learning rate of $\eta = 10^{-3}$ and 2000 diagonal samples to estimate the expectation values. For $\beta = 1$ we used 4500 samples and a regularization of $\lambda = 10^{-3}$. For $\beta = 2$ the number of samples was increased to 18 000 and the regularization was 10^{-2}. Shown are (a) the steady-state expectation value of σ_i^z, (b) the two-point correlation function $\langle \sigma_i^z \sigma_{i+1}^z \rangle$, and (c) and (d) the relevant cost functions for each ansatz.

review of the behavior of this model in one dimension see Ref. [52].)

We again study the convergence of both the LDM and NDM approaches for finding the steady state of this model for a system size of $N = 6$. In Figs. 8(a) and 8(b) we show how both a single-site and a two-site observable vary with the applied field h. We see that, even when using a large number of samples and parameters, the NDM ansatz is not able to find a good approximation to the exact result, while the LDM is able to get much closer to the expected result, especially at small values of h. We see that for both approaches the cost function estimate is much larger than it was for the simpler model described by Eq. (23). This is because the steady state in this case has many more complex correlations than in the previous model; simple expressions like those in Eqs. (24) and (25) are not available, except at very large $h \to \infty$, where the steady state is the same as given in Eq. (24). We next go on to show how using measures of the entanglement found in the steady state can give good intuition for when these kinds of difficulties arise.

B. Entanglement properties

To further understand the convergence of these approaches we investigate the entanglement present in the steady states of both models over a range of parameters. Contrary to pure states, quantifying the number of correlations that are present in a mixed state is not as simple as just calculating the entanglement entropy between two halves of the system [57]. For our purposes, we find that the negativity [58,59]

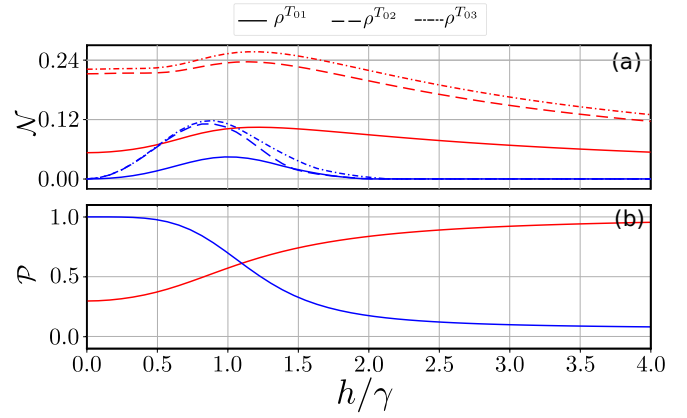$$\mathcal{N} = \frac{\|\rho^{T_A}\| - 1}{2} \tag{27}$$

FIG. 9. (a) Steady-state entanglement negativity, defined in Eq. (27), and (b) purity $\mathcal{P} = \text{Tr}(\rho^2)$ for the $\sigma^x \sigma^x$ model (upper, red curves) and the $\sigma^z \sigma^z$ model (lower, blue curves). Comparing with the results of Figs. 3 and 8, we observe a correlation between a large negativity and poor accuracy of the neural network.

provides a useful measure of the correlations which are difficult to represent using the LDM approach described above. Here $\|\rho^{T_A}\|$ denotes the trace norm of the partially transposed density matrix with the transpose taken over the degrees of freedom labeled by A. This quantity gives a measure for the separability of a state. If two subsystems are entangled, the partial transpose can lead to negative eigenvalues, which leads to a trace norm greater than one and hence a nonzero negativity.

Figure 9(a) shows the negativity for both models considered in this paper for the three different possible bipartitions of a four-site system. In the case of the simpler model in Eq. (23) with the $\sigma^z \sigma^z$ interaction, we see a clear peak in the negativity at around $h \sim 0.9\gamma$ for all partitions and a fast decay to zero at values of h above and below this point. This is because of the two limiting cases described in Eqs. (24) and (25), which both have zero negativity. The peak corresponds well to the range of h values which were the most difficult to find convergence with the neural networks.

In the case of the more difficult model described in Eq. (26) with $\sigma^x \sigma^x$ interactions we see a much higher negativity across the whole range of values of h. This ties in well with our experience that this model is much harder to represent using both the LDM and NDM approaches across the board, with generally slight improvements around $h = 0$ and $h \gtrsim 4\gamma$. We see that there is no region where the negativity reaches zero. This model does not have a simple product-state steady state anywhere in the observed parameter range.

VI. CONCLUSION

In summary, we have proposed a NNQS ansatz which compactly represents density matrices in Liouville space, allowing us to find the steady state of lattice models described by a Markovian master equation. This LDM approach was shown to be able to calculate the steady state of a 1D open transverse-field Ising model with 6 and 16 sites. The results were compared to the powerful NDM ansatz as implemented in NETKET. We found that our approach is always able to

reach a comparable accuracy and in many cases is better able to find the steady state, especially when it contains many correlations. We were able to show that the accuracy of this approach is able to be systematically improved by increasing the number of hidden units and hence free parameters in the ansatz.

This permits a clear understanding of the class of states accessible to the LDM ansatz. As we show in Figs. 3 and 6, the most difficult regions to find convergence are very strongly correlated with parameters where the true steady state has high negativity. This is in contrast with the NDM approach, where there is difficulty representing mixed states with no correlations leading to a plateau in the cost function which do not significantly decrease as the steady state becomes more separable.

The results shown here are just a starting point for examining the usefulness of neural network approaches to finding the steady state of open quantum systems. The RBM ansatz we used is the simplest possible architecture and extending the approach here to deep networks such as deep RBMs [60], recurrent neural networks [61], or transformers [62], which have already proven successful in closed system, provides a route to improving both the accuracy and numerical efficiency. The models studied here are also very simple and are accessible by other methods such as tensor-network-based techniques. However, the lack of an underlying lattice geometry for these neural networks can be exploited to study models with long-range interactions and in higher dimensions which are much more difficult to simulate using other approaches.

ACKNOWLEDGMENTS

APPENDIX A: STOCHASTIC RECONFIGURATION

The stochastic reconfiguration update method was originally proposed in the context of attempting to develop variational quantum Monte Carlo algorithms which avoid the sign problem [47,48]. The derivation of the update in this Appendix is based on the ones found in Refs. [20,22,23]. We recommend especially [20] for various additional ways of deriving SR and Ref. [23] for an in-depth discussion of the properties of the quantum Fisher matrix, which plays an important role in SR.

Consider an ansatz function $|\rho_\alpha\rangle$ with variational parameters α. For the derivation, we assume that the ansatz is normalized: $\|\rho_\alpha\|^2 = \langle\rho_\alpha|\rho_\alpha\rangle = 1$. We can use this to define a semiorthogonal basis consisting of $|\rho_{i=0,\alpha}\rangle = |\rho_\alpha\rangle$, the ansatz function, as well as its derivatives

$$|\rho_{i,\alpha}\rangle = (\mathcal{O}_i - \langle\mathcal{O}_i\rangle)|\rho_{0,\alpha}\rangle, \qquad (A1)$$

where $\langle\mathcal{O}_i\rangle = \langle\rho_{0,\alpha}|\mathcal{O}_i|\rho_{0,\alpha}\rangle$ and $\mathcal{O}_i$ is the logarithmic derivative operator $\mathcal{O}_i(s) = \frac{1}{\rho_\alpha(s)}\frac{\partial}{\partial\alpha_i}\rho_\alpha(s)$. A variation by a small parameter shift γ then yields

$$|\rho_{\alpha+\gamma}\rangle \approx \sum_{i=0}^{N_p} \gamma_i |\rho_{i,\alpha}\rangle. \qquad (A2)$$

Let us now suppose we are given some Liouvillian L. Then the real-time evolution over some small time step δt is given by $e^{-\delta t L}|\rho_\alpha\rangle$. Expanding this around a small time step yields

$$e^{-\delta t L}|\rho_{0,\alpha}\rangle \approx (1 - \delta t L)|\rho_{0,\alpha}\rangle. \qquad (A3)$$

We now have

$$|\rho_{\alpha+\gamma}\rangle \approx \sum_{i=0}^{N_p} \gamma_i |\rho_{i,\alpha}\rangle,$$

$$|\tilde{\rho}_{0,\alpha}\rangle \approx (1 - \delta t L)|\rho_{0,\alpha}\rangle.$$

The idea is now to project both sides into the nonorthogonal basis and ask under which conditions they become equal [22]:

$$\langle\rho_{i,\alpha}|\tilde{\rho}_{0,\alpha}\rangle = \langle\rho_{i,\alpha}|\rho_{\alpha+\gamma}\rangle$$

$$\Rightarrow \langle\rho_{i,\alpha}|(1 - \delta t L)|\rho_{0,\alpha}\rangle = \langle\rho_{i,\alpha}|\sum_{j=0}^{N_p} \gamma_j |\rho_{j,\alpha}\rangle$$

$$\Rightarrow -\delta t\langle\rho_{i,\alpha}|L|\rho_{0,\alpha}\rangle = \sum_{j=1}^{N_p} \gamma_j\langle\rho_{i,\alpha}|\rho_{j,\alpha}\rangle.$$

The left-hand side now reads

$$-\delta t\langle\rho_{0,\alpha}|(\mathcal{O}_i^* - \langle\mathcal{O}_i^*\rangle)L|\rho_{0,\alpha}\rangle = -\delta t(\langle\mathcal{O}_i^* L\rangle - \langle\mathcal{O}_i^*\rangle\langle L\rangle), \qquad (A4)$$

while the right-hand side reads

$$\sum_{j=1}^{N_p} \gamma_j\langle\rho_{0,\alpha}|(\mathcal{O}_i^* - \langle\mathcal{O}_i^*\rangle)(\mathcal{O}_j - \langle\mathcal{O}_j\rangle)\rho_{0,\alpha}\rangle,$$

$$\times \sum_{j=1}^{N_p} \gamma_j(\langle\mathcal{O}_i^*\mathcal{O}_j\rangle - \langle\mathcal{O}_i^*\rangle\langle\mathcal{O}_j\rangle).$$

Equating both sides yields

$$-\delta t(\langle\mathcal{O}_i^* L\rangle - \langle\mathcal{O}_i^*\rangle\langle L\rangle) = \sum_{j=1}^{N_p} \gamma_j(\langle\mathcal{O}_i^* O_j\rangle - \langle\mathcal{O}_i^*\rangle\langle\mathcal{O}_j\rangle). \qquad (A5)$$

Identifying the forces $f_i = \langle\mathcal{O}_i^* L\rangle - \langle\mathcal{O}_i^*\rangle\langle L\rangle$ and the quantum Fisher matrix $S_{i,j} = \langle\mathcal{O}_i^*\mathcal{O}_j\rangle - \langle\mathcal{O}_i^*\rangle\langle\mathcal{O}_j\rangle$, we can write this more succinctly

$$-\delta t f = S\gamma. \qquad (A6)$$

The solution of this system of linear equations γ is the parameter updates which most closely resemble a step in real time of size δt.

APPENDIX B: MARKOV CHAIN MONTE CARLO SAMPLING

Since the size of the required Liouville space grows exponentially fast with the system size, it quickly becomes impossible to calculate expectation values and gradients exactly. This requires us to draw sample states from the total

space in a way that follows the true probability distribution parametrized by the variational ansatz.

The algorithm we employ to generate the samples is the Metropolis-Hastings algorithm. It can be used to draw samples from an unknown distribution, as long as we have a function which is at least proportional to that distribution. We achieve this by realizing that the relative probability of two samples σ, $\frac{p(\sigma_0)}{p(\sigma_1)}$, is independent of the exact normalization of the distribution.

Each sample is saved and generated iteratively in such a way that the next sample depends only on the current sample. In this way a Markov chain of samples is created that can be used to estimate expectations and gradients.

The full algorithm is as follows. Let $\rho_\alpha(x)$ be a function that is proportional to the desired distribution $|\rho|^2$. Further, let $Q(x|y) = Q(y|x)$ be a symmetric, conditional probability distribution that determines the jumps. Note that we never explicitly calculate $Q(x|y)$. In the case of flipping one random spin, the probability to flip any spin is always $1/N$, independent of the current configuration. Hence, $Q(x|y) = Q(y|x)$ is trivially fulfilled.

(i) Choose a random spin configuration σ_0.

(ii) Use $Q(x|y)$ to suggest a new configuration σ_1.

(iii) Calculate the relative probability $p = |\frac{\rho_\alpha(\sigma_1)}{\rho_\alpha(\sigma_0)}|^2$.

(iv) Generate a random number r between 0 and 1 and compare with the relative probability. If $r < p$, accept the new configuration and add it to the list. Otherwise add the old sample to the list again.

(v) Repeat the process until a sufficient number of samples has been generated.

(vi) Calculate the average of the operators and gradients from this set as required.

There are a few pitfalls to look out for. First, since each sample is generated as a small change from the previous sample, neighboring samples are strongly correlated and the full chain may not faithfully represent the true distribution. Second, the first sample of a chain is randomly chosen from all possible states with a uniform probability, while it might have an infinitesimally small weight in the true distribution. Hence, the first sample and a number of samples following it can drastically skew the distribution represented by the chain. Third, it is not at all obvious when enough samples have been generated to represent the true distribution.

These problems have been known for a long time and have been discussed in the literature [9,20,64]. The first problem is best addressed by only keeping one in every n of the samples. This ensures that each sample in the resulting chain is independent of the previous one (see [9,20]). The second problem can be addressed in a similar fashion. Instead of taking a randomly selected sample, one generates an entire chain again and uses the last entry as the starting point for the final Markov chain. This ensures that the starting point of the simulation lies in a corner of the state space with finite weight. The third problem is addressed in, e.g., Refs. [64–66]. They proposed to estimate the variance between several Markov chains and within these chains, arguing that their ratio should approach unity in the limit of perfect convergence or infinite samples. This gives a good estimate of how close the samples are to representing the true probability distribution. This measure is known in the literature as the Gelman-Rubin R value.

---

[1] M. Fitzpatrick, N. M. Sundaresan, A. C. Y. Li, J. Koch, and A. A. Houck, Observation of a dissipative phase transition in a one-dimensional circuit QED lattice, Phys. Rev. X **7**, 011016 (2017).

[2] S. R. K. Rodriguez, W. Casteels, F. Storme, N. Carlon Zambon, I. Sagnes, L. Le Gratiet, E. Galopin, A. Lemaître, A. Amo, C. Ciuti, and J. Bloch, Probing a dissipative phase transition via dynamical optical hysteresis, Phys. Rev. Lett. **118**, 247402 (2017).

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

[4] R. O. Jones, Density functional theory: Its origins, rise to prominence, and future, Rev. Mod. Phys. **87**, 897 (2015).

[5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. **21**, 1087 (1953).

[6] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika **57**, 97 (1970).

[7] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[8] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer, Feedback-optimized parallel tempering Monte Carlo, J. Stat. Mech. (2006) P03018.

[9] H. G. Katzgraber, Introduction to Monte Carlo methods, arXiv:0905.1629.

[10] G. Vidal, Efficient classical simulation of slightly entangled quantum computations, Phys. Rev. Lett. **91**, 147902 (2003).

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

[12] R. Orús, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Ann. Phys. (NY) **349**, 117 (2014).

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

[14] 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).

[15] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted Boltzmann machines in quantum physics, Nat. Phys. **15**, 887 (2019).

[16] A. N. Kolmogorov, On the representation of continuous functions of several variables by superpositions of continuous functions of fewer variables, Dokl. Akad. Nauk SSSR **108**, 179 (1956).

[17] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks **4**, 251 (1991).

[18] N. L. Roux and Y. Bengio, Representational power of restricted Boltzmann machines and deep belief networks, Neural Comput. **20**, 1631 (2008).

[19] M. Y. Pei and S. R. Clark, Neural-network quantum states for spin-1 systems: Spin-basis and parameterization effects on compactness of representations, Entropy **23**, 879 (2021).

[20] F. Becca and S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, Cambridge, 2017).

[21] 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).

[22] 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).

[23] C.-Y. Park and M. J. Kastoryano, Geometry of learning neural quantum states, Phys. Rev. Res. **2**, 023232 (2020).

[24] D.-L. Deng, X. Li, and S. Das Sarma, Quantum entanglement in neural network states, Phys. Rev. X **7**, 021021 (2017).

[25] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Recurrent neural network wave functions, Phys. Rev. Res. **2**, 023358 (2020).

[26] S. A. Khandoker, J. M. Abedin, and M. Hibat-Allah, Supplementing recurrent neural networks with annealing to solve combinatorial optimization problems, Mach. Learn.: Sci. Technol. **4**, 015026 (2023).

[27] K. Choo, T. Neupert, and G. Carleo, Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states, Phys. Rev. B **100**, 125124 (2019).

[28] S. Efthymiou, M. J. S. Beach, and R. G. Melko, Super-resolving the Ising model with convolutional neural networks, Phys. Rev. B **99**, 075113 (2019).

[29] C. Roth and A. H. MacDonald, Group convolutional neural networks improve quantum state accuracy, arXiv:2104.05085.

[30] 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).

[31] G. Torlai and R. G. Melko, Latent space purification via neural density operators, Phys. Rev. Lett. **120**, 240503 (2018).

[32] M. J. Hartmann and G. Carleo, Neural-network approach to dissipative quantum many-body dynamics, Phys. Rev. Lett. **122**, 250502 (2019).

[33] 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).

[34] 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).

[35] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[36] T. Barthel, U. Schollwöck, and S. R. White, Spectral functions in one-dimensional quantum systems at finite temperature using the density matrix renormalization group, Phys. Rev. B **79**, 245101 (2009).

[37] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with geenrative models, Nat. Mach. Intell. **1**, 155 (2019).

[38] M. Reh, M. Schmitt, and M. Gärttner, Time-dependent variational principle for open quantum systems with artificial neural networks, Phys. Rev. Lett. **127**, 230501 (2021).

[39] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Autoregressive neural network for simulating open quantum systems via a probabilistic formulation, Phys. Rev. Lett. **128**, 090501 (2022).

[40] F. Vicentini, R. Rossi, and G. Carleo, Positive-definite parametrization of mixed quantum states with deep neural networks, arXiv:2206.13488.

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

[42] F. Minganti, A. Biella, N. Bartolo, and C. Ciuti, Spectral theory of Liouvillians for dissipative phase transitions, Phys. Rev. A **98**, 042118 (2018).

[43] H. Weimer, Variational principle for steady states of dissipative quantum many-body systems, Phys. Rev. Lett. **114**, 040402 (2015).

[44] H. Weimer, A. Kshetrimayum, and R. Orús, Simulation methods for open quantum many-body systems, Rev. Mod. Phys. **93**, 015008 (2021).

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

[46] S. Finazzi, A. Le Boité, F. Storme, A. Baksic, and C. Ciuti, Corner-space renormalization method for driven-dissipative two-dimensional correlated systems, Phys. Rev. Lett. **115**, 080604 (2015).

[47] S. Sorella, Green function Monte Carlo with stochastic reconfiguration, Phys. Rev. Lett. **80**, 4558 (1998).

[48] S. Sorella, Generalized Lanczos algorithm for variational quantum Monte Carlo, Phys. Rev. B **64**, 024512 (2001).

[49] F. Amin, Complex-valued neural networks: Learning algorithms and applications, Ph.D. thesis, University of Fukui, 2012.

[50] C.-E. Bardyn and A. İmamoğlu, Majorana-like modes of light in a one-dimensional array of nonlinear cavities, Phys. Rev. Lett. **109**, 253606 (2012).

[51] T. E. Lee, S. Gopalakrishnan, and M. D. Lukin, Unconventional magnetism via optical pumping of interacting spin systems, Phys. Rev. Lett. **110**, 257204 (2013).

[52] C. Joshi, F. Nissen, and J. Keeling, Quantum correlations in the one-dimensional driven dissipative *XY* model, Phys. Rev. A **88**, 063835 (2013).

[53] G. Carleo, K. Choo, D. Hofmann, J. E. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, M. Mauri, G. Mazzola, C. B. Mendl, E. van Nieuwenburg, O. O'Reilly, H. Théveniaut, G. Torlai, F. Vicentini, and A. Wietek, NetKet: A machine learning toolkit for many-body quantum systems, SoftwareX **10**, 100311 (2019).

[54] J. Johansson, P. Nation, and F. Nori, QuTiP 2: A Python framework for the dynamics of open quantum systems, Comput. Phys. Commun. **184**, 1234 (2013).

[55] J. Jin, A. Biella, O. Viyuela, L. Mazza, J. Keeling, R. Fazio, and D. Rossini, Cluster mean-field approach to the steady-state phase diagram of dissipative spin systems, Phys. Rev. X **6**, 031011 (2016).

[56] E. Mascarenhas, H. Flayac, and V. Savona, Matrix-product-operator approach to the nonequilibrium steady state of

driven-dissipative quantum arrays, Phys. Rev. A **92**, 022116 (2015).

[57] L. Henderson and V. Vedral, Classical, quantum and total correlations, J. Phys. A: Math. Gen. **34**, 6899 (2001).

[58] G. Vidal and R. F. Werner, Computable measure of entanglement, Phys. Rev. A **65**, 032314 (2002).

[59] K. Modi, A. Brodutch, H. Cable, T. Paterek, and V. Vedral, The classical-quantum boundary for correlations: Discord and related measures, Rev. Mod. Phys. **84**, 1655 (2012).

[60] H. Hu, L. Gao, and Q. Ma, Deep restricted Boltzmann networks, arXiv:1611.07917.

[61] A. Sherstinsky, Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network, Physica D **404**, 132306 (2020).

[62] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, arXiv:1706.03762.

[63] S. Kothe and P. Kirton, Data for: "Liouville-space neural network representation of density matrices", https://doi.org/10.15129/8abe808c-d1a2-43ed-bbe0-708d4f01c8ea.

[64] A. Gelman and D. B. Rubin, Inference from iterative simulation using multiple sequences, Stat. Sci. **7**, 457 (1992).

[65] A. Gelman, W. R. Gilks, and G. O. Roberts, Weak convergence and optimal scaling of random walk Metropolis algorithms, Ann. Appl. Probab. **7**, 110 (1997).

[66] D. Vats and C. Knudson, Revisiting the Gelman-Rubin diagnostic, Statist. Sci. **36**, 518 (2021).