Variational neural network ansatz for steady states in open quantum systems

Filippo Vicentini,[1] Alberto Biella,[1] Nicolas Regnault,[2] and Cristiano Ciuti[1]

[1] *Laboratoire Matériaux et Phénomènes Quantiques,
Université Paris Diderot, CNRS-UMR7162, 75013 Paris, France*
[2] *Laboratoire de Physique de l'Ecole Normale Supérieure,
ENS, Université PSL, CNRS, Sorbonne Université,
Université Paris-Diderot, Sorbonne Paris Cité, Paris, France*

(Dated: May 29, 2019)

We present a general variational approach to determine the steady state of open quantum lattice systems via a neural network approach. The steady-state density matrix of the lattice system is constructed via a purified neural network ansatz in an extended Hilbert space with ancillary degrees of freedom. The variational minimization of cost functions associated to the master equation can be performed using a Markov chain Monte Carlo sampling. As a first application and proof-of-principle, we apply the method to the dissipative quantum transverse Ising model.

In spite of the tremendous experimental progress in the isolation of quantum systems, a finite coupling to the environment [1] is unavoidable and certainly plays a crucial role in the practical implementation of quantum information and quantum simulation protocols [2]. Moreover, through an active control of the environment via the so-called reservoir engineering, an open quantum manybody system can be prepared in non-trivial phases [3–5] with also possible quantum applications [6, 7]. The theoretical description of open quantum manybody systems is in general out-of-the equilibrium and much less developed than for equilibrium systems. A mixed state with a finite entropy can be described by a density matrix, whose evolution is described by a master equation. Recently, a few theoretical methods have been developed to solve the master equation of open quantum manybody systems, including analytical approaches based on the Keldysh formalism [8, 9], numerical algorithms based on matrix product operator and tensor-network techniques [10–14], cluster mean-field methods [15, 16], corner-space renormalization [17–19], Gutzwiller mean-field [20], full configuration-interaction Monte Carlo [21], permutation-invariant solvers [22] or efficient stochastic unravelings for disordered systems [23]. The research in the field is very active, since the different methods are optimal for different specific regimes. For example, the corner-space renormalization method is best suited for systems with moderate entropy, while matrix product operator techniques to systems with short-range quantum correlations.

In the last decade, the field of artificial neural networks has enjoyed a dramatic expansion and success thanks to remarkable applications in the recognition of complex patterns such as visual images or human speech (for a recent review see, e.g., [24]). The optimization (supervised learning) of the network is obtained by tuning the weights quantifying the connections between neural units via a variational minimization of a properly defined cost function. The wavefunction of a manybody system is in general a complex quantity, which is hard to be recognized. Recent works have proposed to exploit artificial neural networks to construct trial wavefunctions, where the connection weights in the network play the role of variational parameters [25, 26]. Neural network approaches have already been successfuly applied to a wide number (see e.g. [27–31]) of close Hamiltonian systems. However, they have not yet been generalized to the important quantum manybody problem of open systems.

In this Letter, we present a theoretical approach based on a variational neural network ansatz in order to determine the steady state of the master equation of open quantum lattice systems. We construct the ansatz for the mixed density matrix starting from a Restricted Boltzman Machine ansatz for a pure many-body wavefunction in an extended Hilbert space. We determine the optimal variational parameters by minimizing a cost function which involves the Liouvillian superoperator associated to the master equation for the density matrix. As a first application, we have considered the dissipative tranverse field quantum Ising model. We present a proof-of-principle demonstration by benchmarking the neural network calculations of the steady state against numerically exact simulations performed by quantum trajectories in the full Hilbert space [32]. Our minimization of the cost function is performed by Markov chain Monte Carlo sampling of the gradient and is thus scalable to a large number of lattice sites. Perspectives of the present approach are discussed in the conclusions.

The general task that we wish to solve is the determination of the steady state of an open quantum system described by the Lindblad master equation [1] for the system reduced density matrix $\hat{\rho}$, which reads (setting $\hbar = 1$):

$$\dot{\hat{\rho}} = \mathcal{L}\hat{\rho} = -i\left[\hat{H}, \hat{\rho}\right] + \sum_j \frac{\gamma_j}{2}\left[2\hat{L}_j\hat{\rho}\hat{L}_j^\dagger - \{\hat{L}_j^\dagger\hat{L}_j, \hat{\rho}\}\right], \quad (1)$$

where $\mathcal{L}$ is the so-called Liouvillian superoperator depending on the system Hamiltonian operator $\hat{H}$. The coupling to the environment is represented by interaction channels with the reservoir characterized by dissipation rates γ_j and jump operators $\hat{L}_j$ acting on the system.

Here we will focus on situations where the steady state ($\partial_t \hat{\rho}_{ss} = 0$) is unique. In this case, the steady-state density matrix can be obtained as $\hat{\rho}_{ss} = \lim_{t\to\infty} \hat{\rho}(t)$ regardless of the initial condition. Although it is possible to engineer peculiar Liouvillians with more than one steady state [33], typical physical systems with a finite Hilbert space dimension have a unique steady-state [34**?** , 35].

For the many-body problem an analytical expression for $\hat{\rho}_{ss}$ can be found in very few cases [36, 37]. In general, because of the exponential growth of the Hilbert space with the number of lattice sites, describing the full density matrix requires exponentially many complex numbers, which in practice can be done exactly only for a small number of sites. If one wants to attack the problem within a variational framework, the density matrix can be represented by an ansatz $\hat{\rho}_{\boldsymbol{v}}$ depending on a set of variational parameters $\boldsymbol{v}$. If $\{|\boldsymbol{\sigma}\rangle = |\sigma_1, \sigma_2, ..., \sigma_N\rangle\}$ denotes a basis of states for the system Hilbert space, the density matrix can be expressed in the form

$$\hat{\rho}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\boldsymbol{\sigma'}} \rho_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) \, |\boldsymbol{\sigma}\rangle \langle \boldsymbol{\sigma'}| \, . \tag{2}$$

In order to construct our neural network ansatz for the density matrix, we consider an extended Hilbert space $\mathcal{H} = \mathcal{H}_S \otimes \mathcal{H}_A$ where $\mathcal{H}_{S,A}$ represents respectively the system and ancillary Hilbert spaces. Such extended space is spanned by the basis set $\{|\boldsymbol{\sigma}, \boldsymbol{a}\rangle\}$ where $\boldsymbol{a} = (a_1, a_2, ..., a_{N_a})$ labels the ancillary degrees of freedom. We start by considering a pure state in the extended Hilbert space, represented by the wavefunction $\psi_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{a})$. In this framework the reduced density matrix of the system S is obtained by tracing out the ancillary degrees freedom [38], namely

$$\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) = \sum_{\boldsymbol{a}} \psi_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{a}) \psi^{\star}_{\boldsymbol{v}}(\boldsymbol{\sigma'}, \boldsymbol{a}). \tag{3}$$

The next step is to represent $\psi_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{a})$ via a neural network ansatz. This purified procedure automatically ensures that $\hat{\rho}_{\boldsymbol{v}}$ is Hermitian and positive semi-definite, as required for a density matrix. In a recent paper, Torlai and Melko [38] proposed to describe purified wavefunctions as

$$\psi_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{a}) = \sqrt{\mathcal{P}_{\boldsymbol{v}_A}(\boldsymbol{\sigma}, \boldsymbol{a})} \exp[-1/2 \log(\mathcal{P}_{\boldsymbol{v}_\theta}(\boldsymbol{\sigma}, \boldsymbol{a}))]. \tag{4}$$

Both the amplitude $\mathcal{P}_{\boldsymbol{v}_A}(\boldsymbol{\sigma}, \boldsymbol{a})$ and phase-related function $\mathcal{P}_{\boldsymbol{v}_\theta}(\boldsymbol{\sigma}, \boldsymbol{a})$ of the purified wavefunction are given by the Boltzmann-like expression $\mathcal{P}_\nu(\boldsymbol{\sigma}, \boldsymbol{a}) = \sum_{\boldsymbol{h}} \exp[-E_\nu(\boldsymbol{\sigma}, \boldsymbol{a}, \boldsymbol{h})]$ (with $\nu \in \{\boldsymbol{v}_A, \boldsymbol{v}_\theta\}$), where the associated dimensionless energy reads

$$E_\nu(\boldsymbol{\sigma}, \boldsymbol{a}, \boldsymbol{h}) = \boldsymbol{\sigma} \cdot \mathbf{b}^{(\sigma)}_\nu + \boldsymbol{a} \cdot \boldsymbol{b}^{(a)}_\nu + \boldsymbol{h} \cdot \boldsymbol{b}^{(h)}_\nu + \boldsymbol{\sigma}^T \boldsymbol{W}_\nu \boldsymbol{h} + \boldsymbol{\sigma}^T \boldsymbol{U}_\nu \boldsymbol{a}. \tag{5}$$

Note that the ansatz parameters are $\boldsymbol{v} = (\boldsymbol{v}_A, \boldsymbol{v}_\theta)$ where $\boldsymbol{v}_\nu = (\mathbf{b}^{(\sigma)}_\nu, \boldsymbol{b}^{(a)}_\nu, \boldsymbol{b}^{(h)}_\nu, \boldsymbol{W}_\nu, \boldsymbol{U}_\nu)$. The rectangular matrix $\boldsymbol{W}_\nu$ weighs the connections between the system variables (visible layer) to the auxiliary variables (hidden layer), while the weight matrix $\boldsymbol{U}_\nu$ quantifies the connection between the system variables and the ancillary ones (ancillary layer). Such neural network ansatz is represented by a tri-partite Restricted Boltzmann Machine depicted in fig. 1. In other words, there are two independent artificial neural networks, one for the amplitude ($\nu = A$) and one for the phase ($\nu = \theta$). By substituting those formulas into Eq. (3) and carrying out the sum over the ancillary degrees of freedom one obtains a closed formula for the entries of the density matrix:

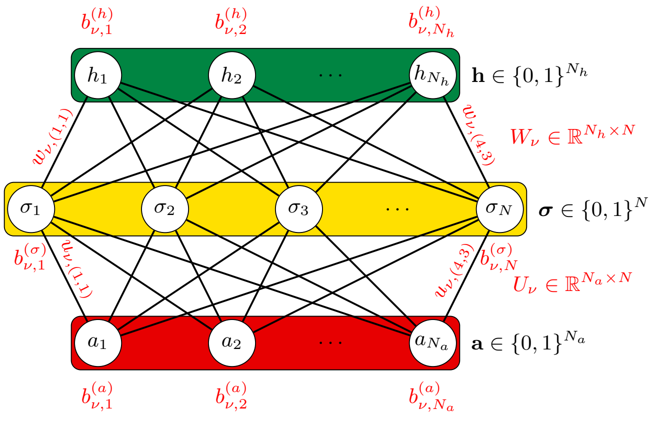

FIG. 1. Graph representation of the artificial neural network used for the density matrix ansatz. The vector $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, ..., \sigma_N)$ contains the variables of the physical system (visible layer). The vector $\boldsymbol{a} = (a_1, a_2, ..., \sigma_{N_a})$ describes the ancillary degrees of freedom of the extended Hilbert space (ancilla layer), where a purified density matrix is considered (see Eq. (3)). The vector $\boldsymbol{h} = (h_1, h_2, ..., h_{N_h})$ contains variables of auxiliary nodes (hidden layer). The network parameters are $\boldsymbol{v}_\nu = (\mathbf{b}^{(\sigma)}_\nu, \boldsymbol{b}^{(a)}_\nu, \boldsymbol{b}^{(h)}_\nu, \boldsymbol{W}_\nu, \boldsymbol{U}_\nu)$. One network is used for the representation of the amplitude ($\nu = \boldsymbol{v}_A$) of the purified wavefunction in Eq. (3), while another independent network with the same topology is used to represent the phase ($\nu = \boldsymbol{v}_\theta$).

$$\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) = \exp\left[\Gamma^-_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) + \Gamma^+_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) + \Pi_{\boldsymbol{v}}(\boldsymbol{\sigma}, \boldsymbol{\sigma'})\right] \tag{6}$$

where the expression of $\Gamma^{+/-}$ and Π can be found in the Supplemental Material [39]. The representation power [40–42] of this ansatz can be systematically improved by increasing the density of the hidden ($\alpha = N_h/N$) and ancillary layer ($\beta = N_a/N$). It is worth pointing out that this scheme is not specific to this network topology, but relies only on the general fact that if two visible layers are connected by a shallow ancillary layer, the ancilla can be traced out analitically and an efficient neural-network description of the density matrix can be obtained.

Having defined a variational ansatz $\hat{\rho}(\boldsymbol{v})$, we now wish to define a variational principle to determine the optimal parameters. In particular, we have to recast the search for the steady state into a minimization problem for a real, positive cost function $\mathcal{C}(\boldsymbol{v})$ which has a global minimum when the master equation $\mathcal{L}\hat{\rho}_{\boldsymbol{v}} = 0$ is satisfied [43]. Moreover, in order to be able to deal with large Hilbert spaces, we need a quantity which can be sampled and computed efficiently. These requirements are met by the following cost function expressed in terms of the 2-norm of the time derivative of the density-matrix:

$$\mathcal{C}(\boldsymbol{v}) = \frac{\|d\hat{\rho}_{\boldsymbol{v}}/dt\|_2^2}{\|\hat{\rho}_{\boldsymbol{v}}\|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \mathcal{L}^\dagger \mathcal{L}\hat{\rho}_{\boldsymbol{v}}\right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \hat{\rho}_{\boldsymbol{v}}\right]}, \tag{7}$$

as (i) $\mathcal{C}(\boldsymbol{v}_{ss}) = 0 \iff \hat{\rho}(\boldsymbol{v}_{ss}) = \hat{\rho}_{ss}$ and (ii) $\mathcal{C}(\boldsymbol{v}) \geq 0$.

It is useful to rewrite Eq. (7) as a sum over the whole space of bounded operators on the Hilbert space:

$$\mathcal{C}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) |\mathcal{C}^{\mathrm{Loc}}(\boldsymbol{v},\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})|^2, \tag{8}$$

where $p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = |\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})|^2/Z$ corresponds to a probability distribution as $Z = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} |\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})|^2$ [44]. The local contribution reads [45] :

$$\mathcal{C}^{\mathrm{Loc}}(\boldsymbol{v},\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \sum_{\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}'} \mathcal{L}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}};\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}') \frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})}. \tag{9}$$

In order to find the global minimum of the cost function (7) by means of gradient-based iterative schemes we need to compute its gradient

$$\nabla_{\boldsymbol{v}}\, \mathcal{C}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \mathcal{C}^{\mathrm{Loc}}(\boldsymbol{v},\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})^\star \Bigg[\sum_{\sigma',\tilde{\sigma}'} \mathcal{L}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}};\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}') \frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})} \mathcal{O}_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}') \Bigg] - \mathcal{C}(\boldsymbol{v})\mathcal{O}_{\boldsymbol{v}}, \tag{10}$$

where we have defined the log-derivatives of the density matrix $\mathcal{O}_{\boldsymbol{v}} = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} \mathcal{O}_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$ and $\mathcal{O}_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \nabla_{\boldsymbol{v}} \log \rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$, which can be efficiently computed for the considered neural network.

The computational complexity of evaluating $\nabla\mathcal{C}(\boldsymbol{v})$ exactly grows exponentially with the size of the system. This cost can be considerably reduced if one only uses an estimate of $\nabla\mathcal{C}(\boldsymbol{v})$ obtained by sampling the values $(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$ according to the probability $p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$. Because the normalisation factor Z is not fixed, we cannot sample the distribution directly and have to resort to a Markov Chain Monte Carlo [46] with Metropolis update rules [47]. At every sampling step, we propose to update the configuration $(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \to (\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')$ by switching a random number of spins and accept the new configuration with probability $\min(\exp[p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})/p_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')], 1)$.

Finally, in order to find the global minimum of the cost function we employ a standard Stochastic Gradient Descent algorithm [48]. In order to improve the performance of the Stochastic Gradient Descent (i.e. to reduce the number of iterations needed to converge to the global minima of the cost function) we update the variational parameters according to to the metric of the space of

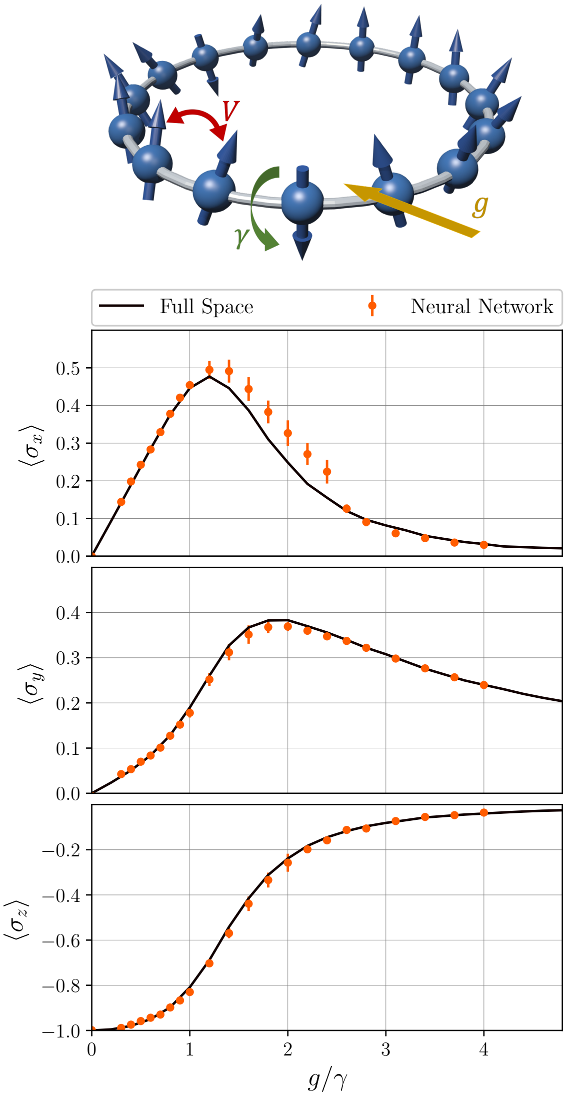

FIG. 2. Top panel: sketch of the considered physical system described by the dissipative quantum transverse Ising 1D model with periodic boundary conditions. The quantity g denotes an applied magnetic field, V the spin-spin coupling and γ the spin flip rate. Bottom panel: the prediction of the neural-network variational calculations (circles) are compared to the results obtained by quantum trajectory simulations of the master equation by considering the whole Hilbert space (solid lines). The top, middle and bottom panels depict the expectation values of the three components of the averaged magnetization as a function of the applied magnetic field g (in units of γ). Model parameters: $V/\gamma = 2$ (spin-spin coupling), $N = 16$ (number of lattice sites). Neural-network parameters: $\alpha = \beta = 1$ for $g \leq \gamma$ and $g \geq 2.5\gamma$ while $\alpha = 1$ and $\beta = 4$ for the remaining points. The parameters required for the convergence of the Monte Carlo calculations depend on the value of g/γ, with the intermediate region being the most demanding. The maximum number of accepted Monte Carlo samples is 8640 and the maximum number of steps for the stochastic gradient descent is 10^4. For points outside the intermediate region 3000 accepted Monte Carlo samples and 10^3 iteration steps have been performed.

density matrices exploiting the Stochastic Reconfiguration Approach [49]. During the optimization procedure we sample the physical observables of interest through another Markov chain as

$$\langle\hat{\Theta}\rangle = \frac{\mathrm{Tr}\left[\hat{\rho}\ \hat{\Theta}\right]}{\mathrm{Tr}[\hat{\rho}]} = \sum_{\boldsymbol{\sigma}} p^{\mathrm{obs}}_{\boldsymbol{v}}(\boldsymbol{\sigma}) \sum_{\tilde{\boldsymbol{\sigma}}} \frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\Theta(\tilde{\boldsymbol{\sigma}},\boldsymbol{\sigma})}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\boldsymbol{\sigma})}, \tag{11}$$

where $p^{\mathrm{obs}}_{\boldsymbol{v}}(\boldsymbol{\sigma}) = \rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\boldsymbol{\sigma})/\,\mathrm{Tr}[\hat{\rho}]$.

In order to benchmark our neural-network approach for open quantum systems, we consider here the dissipative quantum transverse Ising model, whose Hamiltonian is

$$H = \frac{V}{4}\sum_{\langle j,l\rangle}\hat{\sigma}^z_j\hat{\sigma}^z_l + \frac{g}{2}\sum_j \hat{\sigma}^x_j, \tag{12}$$

being $\hat{\sigma}^\alpha_j$ the Pauli matrices ($\alpha \in \{x,y,z\}$) acting on the j-th site. The first term represents the nearest-neighbor spin-spin interaction depending only on the z-components, being V the coupling strength. The second term accounts for a local and uniform magnetic field along the transverse direction x. We consider local dissipative spin-flip processes described by the site-dependent jump operator $\hat{L}^{(z)}_j = \hat{\sigma}^-_j = \frac{1}{2}(\hat{\sigma}^x_j - i\hat{\sigma}^y_j)$, which fully determine the master equation in Eq. (1).

Numerical results for steady-state observables of the dissipative quantum transverse Ising model on a 1D periodic chain are reported in Fig. 2. In particular, we report the spatial components of the averaged magnetization as a function of the magnetic field g (in units of the dissipation rate γ) for $V/\gamma = 2$. For $N = 16$ lattice sites the predictions of the neural-network variational method (circles) are compared to the results obtained with a brute-force exact integration of the master equation in the whole Hilbert space, showing a good agreement over all the parameter range. For $g \lesssim \gamma$ and $g \gtrsim 2.5\gamma$ a remarkable precision is reached for all the local observables with a low density of the hidden and ancillary layer $\alpha = \beta = 1$ and $\mathcal{O}(10^2)$ minimization steps. For $1 \lesssim g/\gamma \lesssim 2.5$ an higher number of variational parameters is required. In particular, as shown in Fig. 3, a systematic improvement of the relative error $\epsilon_{\mathrm{rel}}\left[\,\langle\sigma_x\rangle\,\right]$ with respect to the exact solution can be obtained by increasing β. Interestingly, for $1 \lesssim g/\gamma \lesssim 2.5$, we note that the gradient-descent procedure requires more iterations. This region corresponds to the range of g/γ where the smallest nonzero eigenvalue of $\mathcal{L}^\dagger\mathcal{L}$ decreases significantly [50]. In this range the steady-state density matrix also displays nontrivial correlations and non-thermal mixness properties [50]. Remarkably, the fidelity of the reconstructed local density matrix with respect to the exact one is alway larger then 0.998 for all the values of g/γ considered. Finally, as an example of convergence, the top panel of Fig. 4 depicts a typical evolution of the cost function in the iterative minimization procedure for a fixed set of parameters ($g/\gamma = 1$), showing a good convergence towards the global minimum. In the bottom panel of Fig. 4, the convergence of the x-component of the averaged magnetization is also reported.

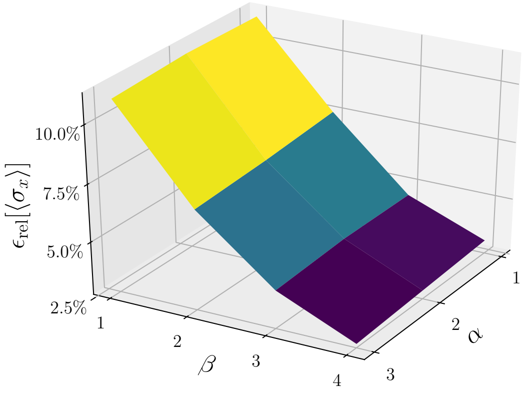

FIG. 3. Relative error with respect to the exact result for the observable $\langle\sigma_x\rangle$ as a function of α and β. Parameters are set as in Fig.2 but for a fixed value $g/\gamma = 1.2$.

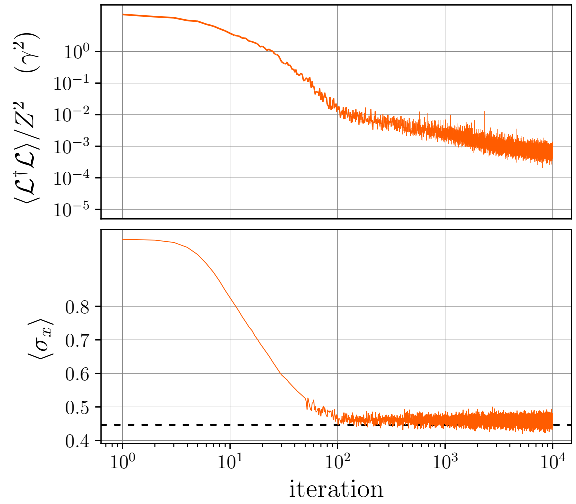

FIG. 4. Same parameters as in Fig. 2 but for a fixed value $g/\gamma = 1$. Top panel: the cost function is shown as a function of the iteration steps. Bottom panel: the corresponding evolution of the x-component of the average magnetization during the stochastic minimization is shown.

In conclusion, we have presented a general variational approach for the steady-state density matrix of open quantum manybody systems based on an artificial neural network scheme. Our method is scalable since the cost function associated to the Liouvillian of the master equation can be calculated via Monte Carlo sampling. We have demonstrated a proof-of-principle of the theoretical scheme by a successful benchmarking to brute-force finite-size simulations in the full Hilbert space for arrays of spins described by the dissipative quantum transverse Ising model. We would like to point out that the present approach does not depend on the specific network topology. Indeed, the variational procedure presented in this Letter is general and can be applied to many other neural-networks or physically-inspired variational ansätze. There are many future developments at the horizon, including the study of dynamical properties, the use of deep neural networks and/or alternative cost functions, comparison with other existing techniques as well as the study of disordered systems without translational invariance. The neural network approach has the potential to pave the way to the theoretical study of a wide spectrum of open quantum manybody systems.

We thank G. Carleo, V. Savona and G. Orso for fruitful discussions. Numerical code for this paper has been written in *Julia* [51]. Full space simulations have been made with *QuantumOptics.jl* [52] and with *QuTiP* [53, 54]. We acknowledge support from ERC (via Consolidator Grant CORPHO No. 616233). This work was granted access to the HPC resources of TGCC under the allocation 2018-A0050510601 attributed by GENCI (Grand Equipement National de Calcul Intensif).

Note: while completing this work, we became aware of related independent theoretical works that have been carried on in parallel [55–57].

[1] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2007).

[2] M. H. Devoret and R. J. Schoelkopf, Science **339**, 1169 (2013).

[3] I. Carusotto and C. Ciuti, Rev. Mod. Phys. **85**, 299 (2013).

[4] C. Noh and D. G. Angelakis, Reports on Progress in Physics **80**, 016401 (2016).

[5] M. J. Hartmann, Journal of Optics **18**, 104005 (2016).

[6] F. Verstraete, M. M. Wolf, and J. Ignacio Cirac, Nature Physics **5**, 633 EP (2009).

[7] J. T. Barreiro, M. Müller, P. Schindler, D. Nigg, T. Monz, M. Chwalla, M. Hennrich, C. F. Roos, P. Zoller, and R. Blatt, Nature **470**, 486 EP (2011).

[8] L. M. Sieberer, M. Buchhold, and S. Diehl, Reports on Progress in Physics **79**, 096001 (2016).

[9] M. F. Maghrebi and A. V. Gorshkov, Phys. Rev. B **93**, 014307 (2016).

[10] E. Mascarenhas, H. Flayac, and V. Savona, Phys. Rev. A **92**, 022116 (2015).

[11] J. Cui, J. I. Cirac, and M. C. Bañuls, Phys. Rev. Lett. **114**, 220601 (2015).

[12] D. Jaschke, S. Montangero, and L. D. Carr, Quantum Science and Technology **4**, 013001 (2019).

[13] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero, Phys. Rev. Lett. **116**, 237201 (2016).

[14] A. Kshetrimayum, H. Weimer, and R. Orús, Nature Communications **8**, 1291 (2017).

[15] A. Biella, J. Jin, O. Viyuela, C. Ciuti, R. Fazio, and D. Rossini, Phys. Rev. B **97**, 035103 (2018).

[16] J. Jin, A. Biella, O. Viyuela, L. Mazza, J. Keeling, R. Fazio, and D. Rossini, Phys. Rev. X **6**, 031011 (2016).

[17] S. Finazzi, A. Le Boité, F. Storme, A. Baksic, and C. Ciuti, Phys. Rev. Lett. **115**, 080604 (2015).

[18] R. Rota, F. Storme, N. Bartolo, R. Fazio, and C. Ciuti, Phys. Rev. B **95**, 134431 (2017).

[19] R. Rota, F. Minganti, C. Ciuti, and V. Savona, (2018), arXiv:1809.10138 [quant-ph].

[20] W. Casteels, R. M. Wilson, and M. Wouters, Phys. Rev. A **97**, 062107 (2018).

[21] A. Nagy and V. Savona, Phys. Rev. A **97**, 052129 (2018).

[22] N. Shammah, S. Ahmed, N. Lambert, S. De Liberato, and F. Nori, Phys. Rev. A **98**, 063815 (2018).

[23] F. Vicentini, F. Minganti, A. Biella, G. Orso, and C. Ciuti, (2018), arXiv:1812.08582 [quant-ph].

[24] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).

[25] G. Carleo and M. Troyer, Science **355**, 602 (2017).

[26] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Phys. Rev. X **8**, 011006 (2018).

[27] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nature Physics **13**, 435 EP (2017).

[28] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).

[29] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Phys. Rev. Lett. **121**, 167204 (2018).

[30] S. Czischek, M. Gärttner, and T. Gasenzer, Phys. Rev. B **98**, 024311 (2018).

[31] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, (2019), arXiv:1902.04057 [cond-mat.dis-nn].

[32] A. J. Daley, Advances in Physics **63**, 77 (2014).

[33] V. V. Albert and L. Jiang, Phys. Rev. A **89**, 022118 (2014).

[34] H. Spohn, Letters in Mathematical Physics **2**, 33 (1977).

[35] F. Minganti, A. Biella, N. Bartolo, and C. Ciuti, Phys. Rev. A **98**, 042118 (2018).

[36] T. Prosen, Phys. Rev. Lett. **112**, 030603 (2014).

[37] T. Prosen, Phys. Rev. Lett. **107**, 137201 (2011).

[38] G. Torlai and R. G. Melko, Phys. Rev. Lett. **120**, 240503 (2018).

[39] For more details, see Supplemental Material at the end of this PDF document.

[40] N. L. Roux and Y. Bengio, Neural Computation **20**, 1631 (2008).

[41] L. Younes, Applied Mathematics Letters **9**, 109 (1996).

[42] G. Montufar and N. Ay, Neural Computation **23**, 1306 (2011).

[43] H. Weimer, Phys. Rev. Lett. **114**, 040402 (2015).

[44] Being able to rewrite the cost function (7) into this form allows us to sample it efficiently. This is the main advantage with respect to using the trace norm of $\mathcal{L}\hat{\rho}$ [43] as a cost function, which requires computing the singular-value-decomposition of $\mathcal{L}\rho$ at each iteration.

[45] This is not the only possible expression that allows to sample the cost function $\mathcal{C}(\boldsymbol{v})$. However, this choice of the local cost function $\mathcal{C}^{\text{Loc}}$ is particularly convenient since it respects the zero-variance property, it is more numerically stable and cheaper to compute. For further details see Sec. 3 of the Supplemental Material.

[46] F. Becca and S. Sorella, *Quantum Monte Carlo Ap-*

proaches for Correlated Systems (Cambridge University Press, 2017).

[47] We point out that a promising direction of research would be to devise particular trial wave functions where Z is fixed or cheaper to compute, so that a direct sampling of the distribution $p_{\sigma,\tilde{\sigma}}$ without a Markov Chain would lead to easier convergence properties. Indeed, it has been recently shown [31] that a direct sampling is possible in some types of networks.

[48] L. Bottou, in *Proceedings of COMPSTAT'2010*, edited by Y. Lechevallier and G. Saporta (Physica-Verlag HD, Heidelberg, 2010) pp. 177–186.

[49] F. Becca and S. Sorella, in *Quantum Monte Carlo Approaches Correl. Syst.* (Cambridge University Press, 2017) pp. 131–155.

[50] J. Jin, A. Biella, O. Viyuela, C. Ciuti, R. Fazio, and D. Rossini, Phys. Rev. B **98**, 241108 (2018).

[51] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, SIAM Review **59**, 65 (2017).

[52] S. Krämer, D. Plankensteiner, L. Ostermann, and H. Ritsch, Computer Physics Communications **227**, 109 (2018).

[53] J. Johansson, P. Nation, and F. Nori, Computer Physics Communications **183**, 1760 (2012).

[54] J. Johansson, P. Nation, and F. Nori, Computer Physics Communications **184**, 1234 (2013).

[55] M. J. Hartmann and G. Carleo, (2019), arXiv:1902.05131 [quant-ph].

[56] N. Yoshioka and R. Hamazaki, (2019), arXiv:1902.07006 [cond-mat.dis-nn].

[57] A. Nagy and V. Savona, (2019), arXiv:1902.09483 [quant-ph].

Supplemental online material for the paper *Variational neural network ansatz for steady states in open quantum systems*

Filippo Vicentini,[1] Alberto Biella,[1] Nicolas Regnault,[2] and Cristiano Ciuti[1]

[1] *Laboratoire Matériaux et Phénomènes Quantiques,*
Université Paris Diderot, CNRS-UMR7162, 75013 Paris, France

[2] *Laboratoire de Physique de l'Ecole Normale Supérieure,*
ENS, Université PSL, CNRS, Sorbonne Université,
Université Paris-Diderot, Sorbonne Paris Cité, Paris, France

(Dated: May 27, 2019)

I. NEURAL NETWORK ANSATZ

We provide here the detailed expressions of the quantities $\Gamma^+_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$, $\Gamma^-_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$ and $\Pi_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})$ entering the neural network ansatz $\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \exp[\Gamma^+_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) + \Gamma^-_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) + \Pi_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]$:

$$\Gamma^+_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \frac{1}{2}\left[\boldsymbol{b}^{(\boldsymbol{v})}_{\boldsymbol{A}}\cdot(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}}) + \sum_j \log\left[\mathcal{G}(\theta^{[j]}_A(\boldsymbol{\sigma}))\right] + \sum_j \log\left[\mathcal{G}(\theta^{[j]}_A(\tilde{\boldsymbol{\sigma}}))\right]\right], \tag{1}$$

$$\Gamma^-_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \frac{i}{2}\left\{\boldsymbol{b}^{(\boldsymbol{v})}_{\boldsymbol{\theta}}\cdot(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}}) + \sum_j \log\left[\mathcal{G}(\theta^{[j]}_\theta(\boldsymbol{\sigma}))\right] - \sum_j \log\left[\mathcal{G}(\theta^{[j]}_\theta(\tilde{\boldsymbol{\sigma}}))\right]\right\}, \tag{2}$$

$$\Pi_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) = \sum_k \log\left\{\mathcal{G}\left[1/2(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}})^T\boldsymbol{U}^{[\boldsymbol{k}]}_{\boldsymbol{A}} + i/2(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}})^T\boldsymbol{U}^{[\boldsymbol{k}]}_{\boldsymbol{\theta}} + b^{(a)}_{A,k}\right]\right\}, \tag{3}$$

where $U^{[k]}_\nu$ is the k-th column of the U_ν matrix, and $\theta^{[j]}_\nu$ is the j-th component of the θ_ν vector

$$\boldsymbol{\theta_\nu}(\boldsymbol{\sigma}) = \boldsymbol{b^{(h)}_\nu} + \boldsymbol{\sigma}^T W_\nu. \tag{4}$$

$\mathcal{G}(x)$ is a non-linear activation function dependent on the values that hidden spins can take. In the manuscript we consider binary spins ($h_i = \{0,1\}$), so $\mathcal{G}(x) = 1 + e^{-x}$.

The total number of real variational parameters of the neural network ansatz (the length of the vector $\boldsymbol{v}$) is given by the following formula:

$$N_{\text{par}} = dim(\boldsymbol{v}) = 2(N + \alpha N + \beta N + \alpha N^2 + \beta N^2), \tag{5}$$

where $\alpha = N_h/N$ and $\beta = N_a/N$ are the hidden and ancillary layer density. For $N \gg 1$ and assuming $\beta = 1$ as in the main text, we get $N_{\text{par}} \sim \mathcal{O}\left((\alpha+1)N^2\right)$. The number of parameters can be compared to a Matrix Product Operator representation of the density matrix where $N^{\text{MPO}}_{\text{par}} \sim \mathcal{O}(N\chi^2)$ where χ is the bond-link dimension [1], playing a role similar to α. For comparison, an exact description in the full Hilbert space requires $\mathcal{O}(2^{2N})$ parameters.

II. LOG-DERIVATIVES

The logarithmic derivatives of the variational density matrix $\rho_{\boldsymbol{v}}(\sigma,\tilde{\boldsymbol{\sigma}})$ appearing in Eq. (10) of the main text are computed with respect to the various components of $\boldsymbol{v}$ as follows:

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{b}_A^{(v)}} = \frac{1}{2}(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}}), \tag{6}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{b}_\theta^{(v)}} = \frac{i}{2}(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}}), \tag{7}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial b_A^{(h)[j]}} = \frac{1}{2}\left\{\frac{\mathcal{G}'}{\mathcal{G}}[\theta_A^{[j]}(\boldsymbol{\sigma})] + \frac{\mathcal{G}'}{\mathcal{G}}[\theta_A^{[j]}(\tilde{\boldsymbol{\sigma}})]\right\}, \tag{8}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial b_\theta^{(h)[j]}} = \frac{i}{2}\left\{\frac{\mathcal{G}'}{\mathcal{G}}[\theta_\theta^{[j]}(\boldsymbol{\sigma})] - \frac{\mathcal{G}'}{\mathcal{G}}[\theta_\theta^{[j]}(\tilde{\boldsymbol{\sigma}})]\right\}, \tag{9}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial b_A^{(a)[j]}} = \frac{\mathcal{G}'}{\mathcal{G}}\left[\frac{1}{2}(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_A^{[j]} + \frac{i}{2}(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_\theta^{[j]} + b_{A,j}^{(a)}\right], \tag{10}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{W}_A^{[j]}} = \frac{1}{2}\left\{\frac{\mathcal{G}'}{\mathcal{G}}[\theta_A^{[j]}(\boldsymbol{\sigma})]\;\boldsymbol{\sigma} + \frac{\mathcal{G}'}{\mathcal{G}}[\theta_A^{[j]}(\tilde{\boldsymbol{\sigma}})]\;\tilde{\boldsymbol{\sigma}}\right\}, \tag{11}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{W}_\theta^{[j]}} = \frac{i}{2}\left\{\frac{\mathcal{G}'}{\mathcal{G}}[\theta_\theta^{[j]}(\boldsymbol{\sigma})]\;\boldsymbol{\sigma} - \frac{\mathcal{G}'}{\mathcal{G}}[\theta_\theta^{[j]}(\tilde{\boldsymbol{\sigma}})]\;\tilde{\boldsymbol{\sigma}}\right\}, \tag{12}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{U}_A^{[j]}} = \frac{1}{2}\;(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}})\;\frac{\mathcal{G}'}{\mathcal{G}}\left[\frac{1}{2}(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_A^{[j]} + \frac{i}{2}(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_\theta^{[j]} + b_{A,j}^{(a)}\right], \tag{13}$$

$$\frac{\partial \log[\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})]}{\partial \mathbf{U}_\theta^{[j]}} = \frac{i}{2}\;(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}})\;\frac{\mathcal{G}'}{\mathcal{G}}\left[\frac{1}{2}(\boldsymbol{\sigma}+\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_A^{[j]} + \frac{i}{2}(\boldsymbol{\sigma}-\tilde{\boldsymbol{\sigma}})\cdot\mathbf{U}_\theta^{[j]} + b_{A,j}^{(a)}\right], \tag{14}$$

$$\tag{15}$$

where $\mathcal{G}'(x) = -e^{-x}$ and $\frac{\mathcal{G}'}{\mathcal{G}}[x] = (e^x - 1)^{-1}$.

III. SAMPLING OF THE COST FUNCTION

In the main text we show that the cost function $\mathcal{C}(\boldsymbol{v})$ defined in Eq. (7)

$$\mathcal{C}(\boldsymbol{v}) = \frac{\|d\hat{\rho}_{\boldsymbol{v}}/dt\|_2^2}{\|\hat{\rho}_{\boldsymbol{v}}\|_2^2} = \frac{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \mathcal{L}^\dagger \mathcal{L} \hat{\rho}_{\boldsymbol{v}}\right]}{\mathrm{Tr}\left[\hat{\rho}_{\boldsymbol{v}}^\dagger \hat{\rho}_{\boldsymbol{v}}\right]}, \tag{16}$$

can be rewritten in such a way that it is possible to sample it from a distribution $p_{\boldsymbol{v}}(\sigma,\tilde{\sigma})$. In particoular, it can be rewritten in two different forms:

$$\mathcal{C}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \left(\sum_{\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}'} (\mathcal{L}^\dagger\mathcal{L})(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}};\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}') \frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})}\right), \tag{17}$$

or

$$\mathcal{C}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \left|\sum_{\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}'} \mathcal{L}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}};\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}') \frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})}\right|^2. \tag{18}$$

In the main text we only considered the latter, but Eq. (17) and Eq. (18) are algebraically equivalent. This equivalence is only valid as long as the sum $\sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}}$ is performed exactly over the whole space $\mathcal{H}\otimes\mathcal{H}$. However, when the sum is evaluated via a Monte Carlo procedure, (17) and Eq. (18) have different convergence properties. In particular, Eq. (17) can be interpreted as the energy of the state $\rho_{\boldsymbol{v}}$ according to the *Hamiltonian* $\mathcal{L}^\dagger\mathcal{L}$, where every term in the sum can have either positive or negative real part. Equation (18), instead, is a sum over positive real terms which are respect the 0-variance property for the steady-state. This leads to a faster convergence of Eq. 18 and improved numerical stability when $\mathcal{C}(\boldsymbol{v})$ is evaluated through a Markov Chain. This is similar to what reported in several works that compare the performance of straightforward energy minimisation schemes to that of variance minimisation [2, 3].

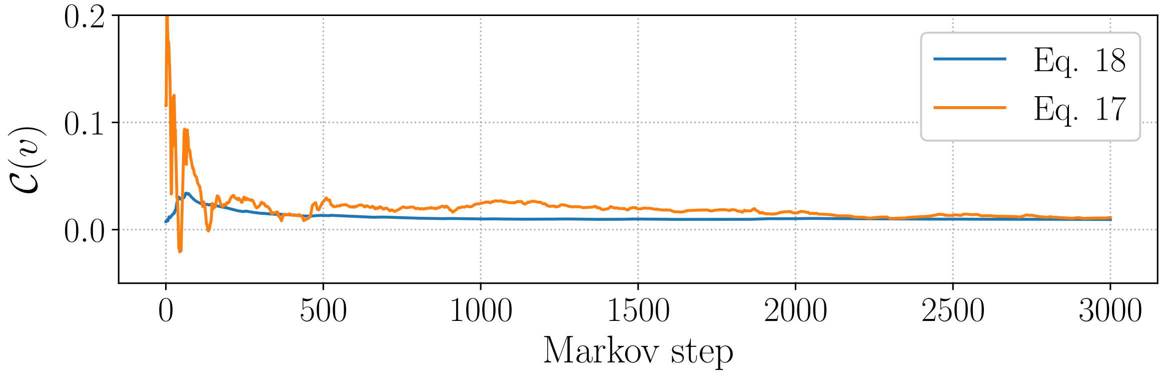

FIG. 1: The cost function Eq. 16 computed iteratively with the two schemes Eq. 17 and Eq. 18 along the same chain. It is clear that the latter converges much faster to the expected results.

To give an explicit example, in Fig. 1 the cost function for a set of weights far from the steady-state has been estimated both with Eq.(17) and Eq.(18) along a sample chain of length $M = 3000$. From the point of view of the computational cost, when computing the local terms $\mathcal{C}^{\mathrm{Loc}}(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}})$ one must evaluate the neural network once for every pair $(\boldsymbol{\sigma}', \tilde{\boldsymbol{\sigma}}')$ for which the tensors $(\mathcal{L}^\dagger\mathcal{L})(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}; \boldsymbol{\sigma}', \tilde{\boldsymbol{\sigma}}')$ or $\mathcal{L}(\boldsymbol{\sigma}, \tilde{\boldsymbol{\sigma}}; \boldsymbol{\sigma}', \tilde{\boldsymbol{\sigma}}')$ are non-zero. Clearly, $\mathcal{L}^\dagger\mathcal{L}$ is less sparse than $\mathcal{L}$, as it has approximately squared the number of non-zero entries, therefore it is computationally more expensive than estimating Eq. 18.

IV. STOCHASTIC RECONFIGURATION

To improve convergence toward the global minimum of the cost function, we adopt the stochastic reconfiguration method, also known as natural gradient by the machine learning community [4, 5]. The stochastic reconfiguration corrects the gradient descent so that small variations $\boldsymbol{\delta v}$ in the parameters do not cause large variations, according to a certain norm, for the density matrices. The generalization of this method, previously developed for closed systems [6], can be straightforwardly generalized to our case. In what follows we give a sketch of the derivation.

Consider the first-order Taylor expansion of the variational density matrix with respect to the variational parameters:

$$\hat{\rho}_{\boldsymbol{v}+\boldsymbol{\delta v}} = \hat{\rho}_{\boldsymbol{v}} + \sum_i \delta v_i \mathcal{O}_{v_i} \hat{\rho}_{\boldsymbol{v}}, \tag{19}$$

where v_i, with $i \in [1, N_{\mathrm{par}}]$ spans all the variational parameters. The distance between $\hat{\rho}_{\boldsymbol{v}}$ and $\hat{\rho}_{\boldsymbol{v}+\boldsymbol{\delta v}}$ is defined to be:

$$\delta s^2_{\boldsymbol{v}} = \left\| \frac{\hat{\rho}_{\boldsymbol{v}+\boldsymbol{\delta v}}}{\|\hat{\rho}_{\boldsymbol{v}+\boldsymbol{\delta v}}\|} - \frac{\hat{\rho}_{\boldsymbol{v}}}{\|\hat{\rho}_{\boldsymbol{v}}\|} \right\|^2, \tag{20}$$

where in the following we have chosen the 2-norm, defined by $\|\hat{A}\|_2^2 = \mathrm{Tr}\left[\hat{A}^\dagger \hat{A}\right]$. It is also known that the distance between two vectors in a space is given by the metric tensor S as

$$\delta s^2_{\boldsymbol{v}} = \sum_{i,j} \delta v_i S^{i,j}_{\boldsymbol{v}} \delta v_j. \tag{21}$$

After some algebra and keeping only the leading terms of the expansion, it is possible to write the (local) metric tensor as a quantity that can be easily sampled, namely:

$$S^{i,j}_{\boldsymbol{v}} = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \mathcal{O}^i_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \mathcal{O}^j_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) - \left(\sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \mathcal{O}^i_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \right) \left(\sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \mathcal{O}^j_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}) \right). \tag{22}$$

Here, we also provide the explicit expressions of the components of the gradient of the cost function defined in Eq. (10) of the main text:

$$\nabla^i_{\boldsymbol{v}}\,\mathcal{C}(\boldsymbol{v}) = \sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\mathcal{C}^{\mathrm{Loc}}(\boldsymbol{v},\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})^{\star}\Big(\sum_{\sigma',\tilde{\sigma}'}\mathcal{L}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}};\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')\frac{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')}{\rho_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})}\mathcal{O}_{\boldsymbol{v}}(\boldsymbol{\sigma}',\tilde{\boldsymbol{\sigma}}')\Big) -$$

$$- \left(\sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\left|\mathcal{C}^{\mathrm{Loc}}(\boldsymbol{v},\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\right|^2\right)\left(\sum_{\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}}} p(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\mathcal{O}^i_{\boldsymbol{v}}(\boldsymbol{\sigma},\tilde{\boldsymbol{\sigma}})\right). \quad (23)$$

In a standard gradient descent scheme one would update the variational parameters at each iteration according to the steepest gradient:

$$v_i \to v_i + \delta v_i \quad \text{where} \quad \delta v_i = -\nabla^i_{\boldsymbol{v}}\mathcal{C}(\boldsymbol{v}). \quad (24)$$

Instead, if we also require that δs^2 be small, we obtain that $\boldsymbol{\delta v}$ is determined by the linear system of equations

$$S^{i,j}_{\boldsymbol{v}}\delta v_j = \nabla^i_{\boldsymbol{v}}\mathcal{C}(\boldsymbol{v}) \quad (25)$$

where $S_{\boldsymbol{v}}$ is a real, dense, symmetric matrix of size $N_{\mathrm{par}} \times N_{\mathrm{par}}$. S has often many degenerate eigenvalues. As in this work $N_{\mathrm{par}} > 1000$, Krylov-space iterative solvers such as MINRES-QLP are particularly suitable [7]. To improve the stability of the solution we solved the equivalent system

$$(S^{i,j}_{\boldsymbol{v}} + \ b\ \delta_{i,j})\delta v_j = \nabla^i_{\boldsymbol{v}}\mathcal{C}(\boldsymbol{v}) \quad (26)$$

with $b \in [10^{-4}, 10^{-3}]$.

V. NOTES ON THE NUMERICS

The core of the algorithm consists in sampling with a Markov-Chain Monte Carlo the two quantities Eq.(20) and Eq.(21), and then solving the linear system defined by Eq.(23). The Monte Carlo step of the algorithm can be easily and efficiently distributed across many cores. Unfortunately, to the best of our knowledge there do not exist efficient distributed solvers for dense systems of linear equations, so the second part of the algorithm cannot be parallelized as efficiently (although the speedup from performing matrix-vector multiplications on several threads is still significant). Therefore the bottleneck of the algorithm is, as of now, the numerical solution of Eq. 23.

We wrote our code in Julia, using MPI.jl to distribute the Monte Carlo Chains across several processors and/or nodes. The linear system of equations was solved either by inversion with standard library routines or by linking to the FORTRAN routines of MINRES-QLP developed by the autors of [7].

VI. CPU TIME

In Fig. 2 we report the scaling of the runtime performance of the neural network algorithm, compared to traditional brute-force Runge-Kutta integration of the master equation and to a quantum trajectory approach.

VII. PROOF OF PRINCIPLE FOR 2D SYSTEMS

In this section we show how the neural ansatz presented in the main text is also suitable to describe the steady-state of 2D lattices. We consider again the dissipative quantum Ising model [see Eq. (1) and Eq. (12) in the main text] for a 3 × 3 lattice with periodic boundary conditions. For $V/\gamma = 2$ and different values of the transverse magnetic field we find again a good agreement with the exact results (obtained via brute-force integration of the master equation). A summary of the results is shown in Table I.

[1] J. Cui, J. I. Cirac, and M. C. Bañuls, Phys. Rev. Lett. **114**, 220601 (2015).

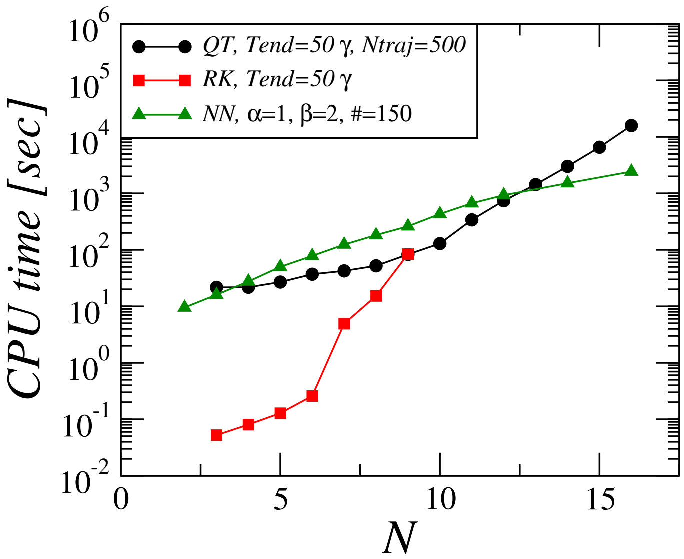

FIG. 2: Runtime cost (in CPU-core-seconds) of various algorithms as a function of the system size N. Red squared: the brute-force RK45 integration of the master equation up to $T_{\mathrm{end}} = 50\gamma$. Green triangles: the integration of 500 trajectories up to $T_{\mathrm{end}} = 50\gamma$ with the TSIT5 ode-integrator. Black circles: the CPU-time of the Neural Network algorithm presented in this paper, with $\alpha = 1$, $\beta = 2$, $M = 10^3 * (1 + log(N))$ Markov-chain samples and 150 iterations.

g/γ	$\langle\sigma_x\rangle$	$\epsilon_{\mathrm{rel}}[\langle\sigma_x\rangle]$	$\langle\sigma_y\rangle$	$\epsilon_{\mathrm{rel}}[\langle\sigma_y\rangle]$	$\langle\sigma_z\rangle$	$\epsilon_{\mathrm{rel}}[\langle\sigma_z\rangle]$
0.5	0.123	0.007 %	0.0158	0.979 %	-0.992	0.051 %
0.8	0.198	0.104 %	0.0258	0.598 %	-0.978	0.103 %
1.1	0.274	0.134 %	0.0370	0.529 %	-0.959	0.061 %
1.4	0.350	0.117 %	0.0506	0.495 %	-0.929	0.007 %
1.7	0.427	0.101 %	0.0672	2.154 %	-0.888	0.522 %

TABLE I: Steady-state values of local magnetizations $\langle\sigma_\alpha\rangle$ with $\{\alpha = x, y, z\}$ for a 3×3 lattice with periodic boundary conditions for $V/\gamma = 2$. The relative error with respect to the exact solution is also shown.

[2] C. J. Umrigar and C. Filippi, Physical Review Letters **94** (2005), 10.1103/physrevlett.94.150201.
[3] P. R. C. Kent, R. J. Needs, and G. Rajagopal, Physical Review B **59**, 12344 (1999).
[4] R. Pascanu and Y. Bengio, (2013), arXiv:arXiv:1301.3584 [cs.LG].
[5] J. Martens, (2014), arXiv:1412.1193 [cs.LG].
[6] F. Becca and S. Sorella, in *Quantum Monte Carlo Approaches Correl. Syst.* (Cambridge University Press, 2017) pp. 131–155.
[7] S.-C. T. Choi and M. A. Saunders, ACM Transactions on Mathematical Software **40**, 1 (2014).