Variational quantum dynamics of two-dimensional rotor models

Matija Medvidović [1 2] **Dries Sels** [1 3]

Abstract

We present a simulation method for the dynamics of continuous-variable quantum many-body systems based on neural-network quantum states. The focus is put on dynamics of experimentally relevant two-dimensional quantum rotors. We simulate previously unreachable system sizes and simulation times using a neural-network trial wavefunction in a continuous basis and using modern sampling approaches based on Hamiltonian Monte Carlo. The method is demonstrated to be able to access quantities like the return probability and vorticity oscillations after a quantum quench in two-dimensional systems of up to 64 (8×8) coupled rotors. Our approach can be used for accurate non-equilibrium simulations of continuous systems at previously unexplored system sizes and evolution times, bridging the gap between simulation and experiment.

1. Introduction

Non-equilibrium quantum many-body physics has been at the forefront of condensed matter, atomic physics and chemistry research for over a decade (Warren et al., 1993; Polkovnikov et al., 2011). The field is driven by remarkable progress in our ability to coherently control matter at the atomic scale. This control has resulted in the creation of novel phases of matter, including observations of light-induced superconductivity (Budden et al., 2021), cavity-enhanced chemical reactions (Thomas et al., 2019) and dynamical phase transitions (Zhang et al., 2017).

The capacity to precisely control (Sivak et al., 2022; Porotti et al., 2022; Metz & Bukov, 2022; Bukov et al., 2018) modern quantum experiments and hardware is roadblocked by the scale and quality of numerical simulations of the real-time evolution of quantum systems. At its core, the problem is related to fast entanglement growth in systems out of equilibrium, which forces one to keep track of all the intricate correlations that build up in the system. While there has been considerable progress (Zaletel et al., 2015; Wurtz et al., 2018; Czarnik et al., 2019; Hubig et al., 2020; Zhou et al., 2020), challenges remain, in particular if one moves away from simple one-dimensional spin models.

Recently, it has been proposed that classical AI methods might alleviate some of the problems (Carleo et al., 2012; 2017; Schmitt & Heyl, 2020; Hofmann et al., 2022; Barison et al., 2021) with efficient quantum state representation. These proposals rely on encoding relevant information in an exponentially many quantum amplitudes into polynomially many parameters of a classical model such as a neural network. In practice, however, it has been difficult to achieve reliable results due numerical instabilities resulting from a combination of Monte Carlo noise and flatness of the quantum geometry of modern neural-network wave functions (Czischek et al., 2018; Schmitt & Heyl, 2020; Amari, 1998; Yuan et al., 2019; Stokes et al., 2020).

In this work, we present an *ab-initio* approach for reliably capturing long-time dynamics of 2D lattice models with continuous degrees of freedom, without an underlying dataset, by projecting the full Schrödinger dynamics into the space of neural-network parameters. A combination of methods that were previously unexplored in the field of variational simulations are used - the Hamiltonian Monte Carlo sampler, a tailored parameterized wavefunction based on convolutional neural networks and a modification of natural gradient descent optimization. We focus on the quantum rotor model with direct applications to arrays of coupled Josephson junctions and explore previously unreachable system sizes and evolution times, up to 8×8 square lattices.

The paper is organized as follows. First, we introduce the quantum rotor model and the variational wavefunction. Then, we outline the Hamiltonian Monte Carlo sampler and its connection to the time-dependent variational Monte Carlo algorithm. Finally, we present results for the two-dimensional model. Our simulations are substantiated by self-consistency checks when key hyperparameters are

[1]Center for Computational Quantum Physics, Flatiron Institute, 162 5th Avenue, New York, NY 10010, USA [2]Department of Physics, Columbia University, New York 10027, USA [3]Department of Physics, New York University, New York 10003, USA. Correspondence to: Matija Medvidović <matija.medvidovic@columbia.edu>.

Workshop on New Frontiers in Learning, Control, and Dynamical Systems at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023. Copyright 2023 by the author(s).

changed and by comparing our approach to independent tensor-network calculations in one and two spatial dimensions.

2. Model and Methods

Consider a system of continuous planar rotors with angles θ_k with respect to an arbitrary axis, on a lattice Λ with N sites. We use the basis $|\boldsymbol{\theta}\rangle \equiv |\theta_1, \dots, \theta_N\rangle$ for the underlying Hilbert space $\mathcal{H}$ and examine an effective Hamiltonian that captures the relevant physics such experiments:

$$H = \frac{gJ}{2} \sum_k L_k^2 - J \sum_{\langle k,l \rangle} \hat{\mathbf{n}}_k \cdot \hat{\mathbf{n}}_l \ , \tag{1}$$

where $L_k = -i\, \partial_{\theta_k}$ and $\hat{\mathbf{n}}_k = (\cos\theta_k, \sin\theta_k)$ in the continuous basis $|\boldsymbol{\theta}\rangle$ of choice. The Hamiltonian in Eq. 1 is often called the Quantum Rotor Model (QRM). Its equilibrium (time-independent) properties (José et al., 1977) have been studied using Variational Monte Carlo (VMC) (Stokes et al., 2021) and other Quantum Monte Carlo (QMC) (Jiang et al., 2019) methods.

However, as noted in the introduction, real-time evolution properties of the QRM have barely been explored. This is mainly due to the lack of suitable methods that can access experimentally relevant evolution times $t \gg J^{-1}$, at large system sizes in two dimensions. Angles θ_k could, for example, represent superconducting phases of adjacent Josephson junctions (Josephson, 1962; 1965; Vogt et al., 2015; Martinoli & Leemann, 2000; Kockum & Nori, 2019) set up in a two dimensional arrays. Therefore, the ability to simulate relatively large system sizes is not only of theoretical interests but has technological applications in the study of dynamics of such experiments (Berke et al., 2022).

The evolution equation for a state $\Psi = \Psi(\boldsymbol{\theta}, t)$, in the continuous basis $|\boldsymbol{\theta}\rangle$, reads:

$$i\frac{\partial \Psi}{\partial t} = -\frac{gJ}{2} \sum_k \frac{\partial^2 \Psi}{\partial \theta_k^2} - J \sum_{\langle k,l \rangle} \cos(\theta_k - \theta_l)\Psi \tag{2}$$

with appropriate periodic boundary conditions

$$\Psi(\theta_1, \dots, \theta_k + 2\pi, \dots, \theta_N) = \Psi(\theta_1, \dots, \theta_k, \dots, \theta_N) \tag{3}$$

for each rotor k. Eq. 2 is prohibitively expensive to solve exactly even for a handful of interacting rotors. The continuous nature of the $|\boldsymbol{\theta}\rangle$ basis exacerbates the problem.

2.1. Variational simulation

We represent a quantum state using a wavefunction $\psi_\alpha(\boldsymbol{\theta})$ where $\alpha \in \mathbb{C}^P$ is a set of P complex variational parameters. Since any $|\psi\rangle \in \mathcal{H}$ admits an expansion in terms of $|\boldsymbol{\theta}\rangle$, we define the following un-normalized variational quantum state (VQS):

$$|\psi_\alpha\rangle = \int \mathrm{d}\boldsymbol{\theta}\ \psi_\alpha(\boldsymbol{\theta})\, |\boldsymbol{\theta}\rangle \tag{4}$$

where $\mathrm{d}\boldsymbol{\theta} \equiv \mathrm{d}\theta_1 \cdots \mathrm{d}\theta_N$. The integral is performed over the cube $[-\pi, \pi]^N$.

Our simulation of the real-time dynamics of the state given in Eq. 4 is based on the time-dependent Variational Monte Carlo (t-VMC) method (Becca & Sorella, 2017; Carleo et al., 2012). The core assumption that allows us to approximately solve Eq. 2 is that of time dependence of parameters $\alpha = \alpha(t)$. In simple words, the method relies on using the neural network trial state as a change of variables, allowing one to approximate the linear Eq. 2 in an *exponentially* large space as a nonlinear equation in the *polynomially* large parameter ($\alpha(t)$) space.

Optimal trajectories $\alpha(t)$ induced by unitary hamiltonian evolution $e^{-iHt}\,|\psi_\alpha\rangle$ can conveniently be found by extremizing the *time-dependent variational principle* (TDVP) (Yuan et al., 2019) action

$$\mathcal{C}[\alpha] = \int \mathrm{d}t\, \langle \Psi_{\alpha(t)}| \left(i\frac{\mathrm{d}}{\mathrm{d}t} - H \right) |\Psi_{\alpha(t)}\rangle \ . \tag{5}$$

where $|\Psi_\alpha\rangle$ is a normalized version of state $|\psi_\alpha\rangle$. Optimal evolution equations (after the change of variables is performed) read $i\, S\dot{\alpha} = g$, where

$$\begin{aligned} S_{\mu\nu} &= \langle \mathcal{O}_\mu^\dagger \mathcal{O}_\nu \rangle - \langle \mathcal{O}_\mu^\dagger \rangle \langle \mathcal{O}_\nu \rangle \\ g_\mu &= \langle \mathcal{O}_\mu^\dagger H \rangle - \langle \mathcal{O}_\mu^\dagger \rangle \langle H \rangle \end{aligned} \tag{6}$$

with averages $\langle \cdot \rangle \equiv {}^{\langle \psi_\alpha | \cdot | \psi_\alpha \rangle}/_{\langle \psi_\alpha | \psi_\alpha \rangle}$ being performed at time t (i.e. for $\alpha = \alpha(t)$). Operators $\mathcal{O}_\mu$ are defined by $\partial_{\alpha_\mu}\, |\psi_\alpha\rangle = \mathcal{O}_\mu\, |\psi_\alpha\rangle$ or

$$\mathcal{O}_\mu(\boldsymbol{\theta}, t) = \frac{\partial}{\partial \alpha_\mu} \ln \psi_{\alpha(t)}(\boldsymbol{\theta}) \tag{7}$$

and can be evaluated using automatic differentiation.

We note that the matrix S is commonly called the *quantum geometric tensor* (QGT) or *quantum Fisher information matrix* (Sorella, 1998; Amari, 1998; Stokes et al., 2020) and corresponds to the metric tensor of the parameter manifold induced by the distance in $\mathcal{H}$ between states defined in Eq. 4. In Eqs. 6, we have chosen our ansatz ψ_α such that it is a holomorphic function of complex parameters α.

Since quantum averages over an exponentially large Hilbert space $\mathcal{H}$ in Eq. 6 cannot be computed exactly, Markov chain Monte Carlo (MCMC) sampling methods are employed

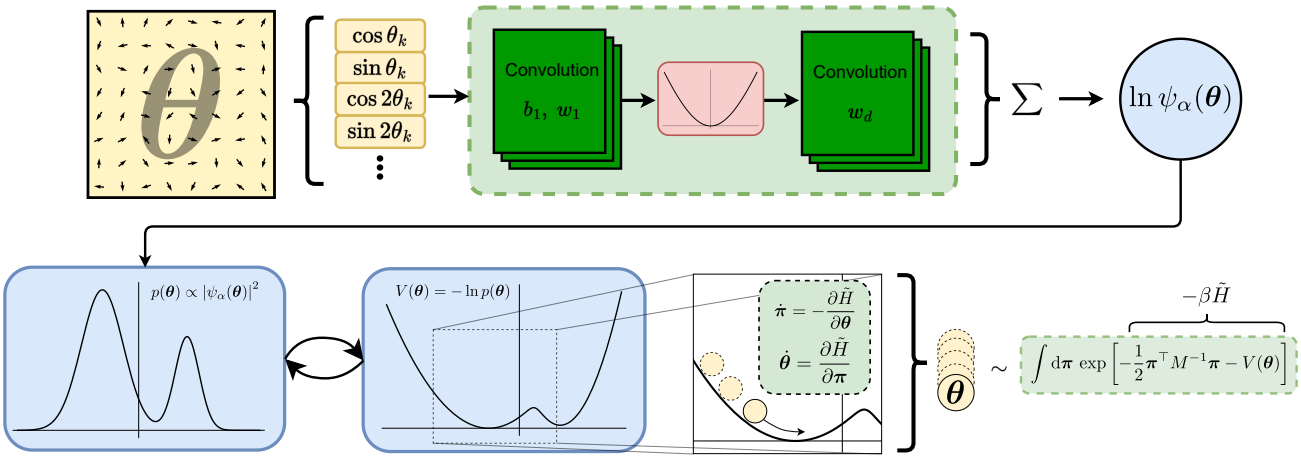

Figure 1. **Top**: The ansatz $\psi_\alpha(\boldsymbol{\theta})$ architecture used for simulations of two-dimensional QRM systems. It amounts to a two-layer convolutional neural network with an activation function given by Eq. 14. To enforce periodicity and improve expressivity, we precalculate sines and cosines of input angles which are treated as different input channels by the CNN. The final layer outputs a single channel and all of its components are summed into a single complex number (because of complex parameters $\alpha \in \mathbb{C}^P$) we then interpret as $\ln\psi_\alpha(\boldsymbol{\theta})$. **Bottom**: An illustration of the Hamiltonian Monte Carlo algorithm. Dummy momentum variables are introduced and sampling the given N-dimensional probability distribution is rewritten in $2N$-dimensional phase space with an artificial effective Hamiltonian $\tilde{H}$. Samples are collected as snapshots of solutions of Hamilton's equations of motion.

(Metropolis et al., 1953; Hastings, 1970). In VMC calculations, it is common to rewrite quantum averages, such as those in Eq. 6, as expressions amenable to estimation through sampling. For example, in the case of the Hamiltonian H, we obtain the *local energy* E_L:

$$E = \langle H \rangle = \frac{\langle \psi_\alpha | H | \psi_\alpha \rangle}{\langle \psi_\alpha | \psi_\alpha \rangle} = \int \mathrm{d}\boldsymbol{\theta}\, p_\alpha(\boldsymbol{\theta})\, E_L(\boldsymbol{\theta}) \tag{8}$$

where

$$p_\alpha(\boldsymbol{\theta}) \propto |\psi_\alpha(\boldsymbol{\theta})|^2 \quad \text{and} \quad E_L(\boldsymbol{\theta}) = \frac{\langle \boldsymbol{\theta} | H | \psi_\alpha \rangle}{\langle \boldsymbol{\theta} | \psi_\alpha \rangle}\,. \tag{9}$$

For more details about the specific sampling algorithm employed in this work, we refer the reader to Sec 2.2 and Appendix A.1.

After computing the matrix S and the vector g at time t, one can formally define $\dot{\alpha} = -i\, S^{-1} g$ and use any ordinary differential equation (ODE) integrator (see Appendix A.3) to obtain the next set of parameters, at time $t + \delta t$. For example, if a simple Euler integrator is chosen, the update reads

$$\alpha' = \alpha - i\, \delta t\, S^{-1} \nabla_\alpha E \tag{10}$$

which can be interpreted as natural gradient descent (NGD) (Amari, 1998) with imaginary learning rate. Indeed, if one is interested in finding the ground state of the same hamiltonian, the update rule in Eq. 10 can be show to lose the factor of i. We refer the interested reader to an excellent overview of related methods in Ref. (Yuan et al., 2019).

However, the inverse S^{-1} is often ill-defined. One reason is that Monte Carlo estimates of matrix elements are noisy. Noise accumulates to render the matrix singular by making a small eigenvalues vanish. Therefore, quickly and efficiently obtaining many uncorrelated samples from $p(\boldsymbol{\theta}, t) \propto |\psi_{\alpha(t)}(\boldsymbol{\theta})|^2$ is crucial. The other reason is that the specific choice of ψ_α introduces redundancy between different parameters, producing linearly dependent or vanishing rows/columns in S. Therefore, choosing an efficiently parameterized trial wavefunction is equally important. In practice, adding more parameters to the wavefunction can sometimes unexpectedly reduce accuracy by making S ill-conditioned. In this work, we use a custom pseudoinverse S^{-1} by diagonalizing S each time step. For more details on regularization, see Appendix A.2.

After calculating averages in Eq. 6 and appropriately regularizing the QGT inverse S^{-1}, one can use any external ODE integrator to perform time-stepping in the top-level equation $\dot{\alpha} = -i\, S^{-1} g$. In this work, we use the embedded Bogacki-Shampine adaptive solver RK3(2) from the Runge-Kutta family. (Bogacki & Shampine, 1989; Butcher, 2008; Press et al., 1992)

2.2. Hamiltonian Monte Carlo

Hilbert space averages defined in Eq. 6 cannot be evaluated analytically for an arbitrary ψ_α. To perform this task in an

efficient and scalable way, we employ Hamiltonian Monte Carlo (HMC) (Neal, 2011; Betancourt, 2017) to obtain samples from the distribution $p_\alpha(\boldsymbol{\theta}) \propto |\psi_\alpha(\boldsymbol{\theta})|^2$ at each time step t. We make this choice because HMC offers a systematic way of making large steps in MCMC proposals while still keeping acceptance probabilities high, unlike more conventional approaches like Random Walk Metropolis (RWM). This results in a Markov chain with considerably lower autocorrelation times, allowing for treatments of larger systems with less overall runtime spent on sampling.

For a generic probability distribution $p(\boldsymbol{\theta})$, HMC augments the configuration space with artificial momentum variables $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N) \sim \mathcal{N}(0, M)$:

$$p(\boldsymbol{\theta}) \propto \int \mathrm{d}\boldsymbol{\pi}\, \exp\left\{-\frac{1}{2}\boldsymbol{\pi}^\top M^{-1}\boldsymbol{\pi} + \ln p(\boldsymbol{\theta})\right\} \tag{11}$$

for some choice of a positive-definite *mass matrix* M. Interpreting the exponent in Eq. 11 as an effective classical hamiltonian $\beta\tilde{H}(\boldsymbol{\theta}, \boldsymbol{\pi})$ inducing a Boltzmann weight $e^{-\beta\tilde{H}}$, Monte Carlo updates can be defined through numerical integration of relevant Hamilton's equations. Owing to insights from statistical physics, we know that a large number of particles in equilibrium following classical equations of motion have precisely this desired Boltzmann distribution.

Given $\boldsymbol{\theta}(0)$, $\boldsymbol{\pi}(0)$ and a small step size ε, the leapfrog integrator (Neal, 2011) is chosen to propagate Hamilton's equations $\dot{\boldsymbol{\theta}} = \frac{\partial\tilde{H}}{\partial\boldsymbol{\pi}}$ and $\dot{\boldsymbol{\pi}} = -\frac{\partial\tilde{H}}{\partial\boldsymbol{\theta}}$. This specific integrator is chosen because of its symplectic (Neal, 2011; Press et al., 1992) property – it conserves energy/probability exactly, allowing for large jumps in the $\boldsymbol{\theta}$-space while keeping high acceptance probabilities.

After integrating for L steps, the new configuration $(\boldsymbol{\theta}(L\varepsilon), \boldsymbol{\pi}(L\varepsilon))$ is proposed as the next sample in the Markov chain. It is common to apply the Metropolis-Hastings accept-reject step (Metropolis et al., 1953; Hastings, 1970) despite the fact that the new configuration has the same energy (probability) as the initial one. This is done to offset the effects of unwanted numerical errors in the leapfrog scheme, usually improving overall performance for many samples. (Neal, 2011; Betancourt, 2017)

These Hamilton's equations simulate a swarm of effective classical particles whose positions and momenta follow the desired joint Boltzmann distribution in Eq. 11. Discarding all $\boldsymbol{\pi}$ samples is equivalent to marginalizing the distribution in Eq. 11. In practice, randomness is injected by sampling the normal distribution $\boldsymbol{\pi}(0) \sim \mathcal{N}(0, M)$ each time initial conditions are required for numerical integration.

Choosing the mass matrix M, the time step ε and the integration length L carefully is crucial for efficient exploration of the configuration space. In this work, we chose to set M and ε automatically, by using heuristically proven (Nesterov, 2009; Hoffman & Gelman, 2011; Betancourt, 2017) algorithms operating samples from an extended warmup phase for each Markov chain individually. Integration length L was treated as a hyperparameter. For more details and specific values, see Appendix A.1.

2.3. The trial wavefunction

In this work, we use a variant of the standard Convolutional Neural Network (CNN) architecture (LeCun et al., 2015; Carleo et al., 2019) to model $\psi_\alpha(\boldsymbol{\theta})$. Our approach is built on those of Refs. (Schmitt & Heyl, 2020; Pescia et al., 2022). Specifically, we set

$$\ln\psi_\alpha(\boldsymbol{\theta}) = \frac{1}{\sqrt{2KN}}\sum_{c=1}^{2K}\sum_k \left[w_D^c * h_{D-1}^c(\boldsymbol{\theta})\right]_k\,, \tag{12}$$

where $*$ denotes a convolution over lattice indices k and $c = 1, \ldots, 2K$ is the channel index. Features $h_{D-1}^c(\boldsymbol{\theta})$ are the output of $D-1$-layer CNN.

We include all weights and biases into the set of trainable parameters α and use automatic differentiation (AD) techniques to obtain all derivatives $\mathcal{O}_\mu$ required for evaluation of Eqs. 6. For CNN inputs h_0, we concatenate the following features:

$$h_0 = \left\{(\cos n\theta_k, \sin n\theta_k)\;\middle|\; n = 1, \ldots, K\right\} \tag{13}$$

along the channel axis, as illustrated on Fig. 1. This construction allows us to include a limited number of higher Fourier modes *a priori*, improving ansatz expressivity in a controlled way. In this work, we set $D = 2$, $K = 4$ for larger two-dimensional (8×8) experiments and $K = 1$ for smaller systems.

To maintain analytic dependence on parameters α, we restrict the CNN nonlinearities f_d to polynomial functions. The Taylor expansion of the logarithm of the zeroth-order modified Bessel function of the first kind is used:

$$\ln I_0(z) = \frac{z^2}{4} - \frac{z^4}{64} + \frac{z^6}{576} + \mathcal{O}(z^8)\,. \tag{14}$$

This particular activation function choice is motivated by the appearance of I_0 in the version of the Restricted Boltzmann Machine (RBM) adapted to the QRM in Ref. (Stokes et al., 2021). This approach has the advantage of maintaining the holomorphic dependence of ψ_α on α and preserving the form of Eqs. 6.

In this work, we focus on a simple 2-layer CNN ansatz to control the number of parameters P. In addition nontrivially affecting the QGT inverse (see subsection 2.1), the diagonalization cost in order to regularize the inverse grows as $\mathcal{O}(P^3)$. Heuristically, we also find that introducing more

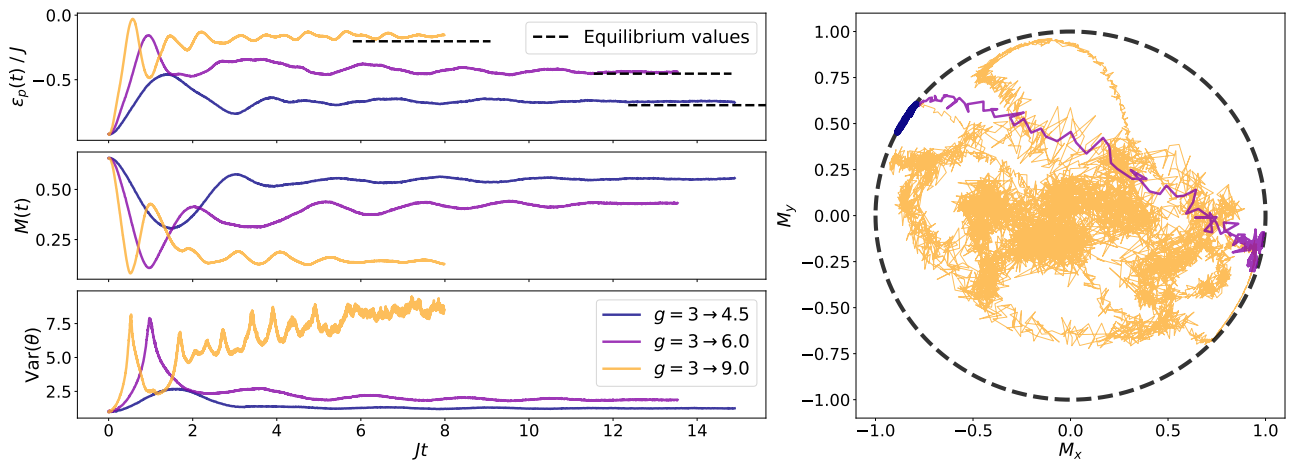

Figure 2. Results for different quenches from initial value $g_i = 3$ on a two-dimensional 8×8 square lattice. **Left**: Potential energy, magnetization and angular variance as functions of real time. For the small quench to $g_f = 4.5$, we observe the expected behavior with slower approach to the new ordered equilibrium state. Convergence is similar to adiabatic change. The moderate quench to $g_f = 6.0$ exhibits a sharp increase in rotor angle variance is accompanied by a single flip (right panel) in the average magnetization at $t \approx J^{-1}$. For the large quench to $g_f = 9.0$, many rotor flips occur after the first one, indicating much more detailed exploration of the underlying Hilbert space. Convergence to the new equilibrium starts taking place only for $t \gtrsim 5J^{-1}$. **Right**: A parametric plot of the mean rotor direction. We observe a more thorough exploration of the magnetization sphere for larger quenches.

parameters α requires more Monte Carlo samples to correctly resolve the relevant averages in Eq. 6 and does not significantly contribute to simulation accuracy in our case. A systematic investigation of larger neural-network architecture details is left for future work.

3. Results

In this section, we study dynamical properties of several observables of the QRM, focusing on the two-dimensional model. A series of benchmarks in one and two dimensions can be found in section 3.1.

We simulate the effects instantaneous *quenches* of the coupling constant g in Eq. 1. Specifically, we initialize parameters α of the ansatz ψ_α illustrated on Fig. 1 to the ground state of the QRM hamiltonian with $g = g_i$ using imaginary-time Variational Monte Carlo (VMC) (Carleo et al., 2019; Becca & Sorella, 2017) methods. We then simulate real-time dynamics under $g = g_f$. In this work, we focus on quenches from the ordered phase to the disordered: $g_i < g_c < g_f$.

In Fig. 2, we choose a square 8×8 lattice, tracking the dynamics of the potential energy density

$$\epsilon_{\mathrm{p}}(t) = -\frac{J}{N}\left\langle \sum_{\langle k,l\rangle} \hat{\mathbf{n}}_k \cdot \hat{\mathbf{n}}_l \right\rangle_t \tag{15}$$

and the average magnetization magnitude M

$$M(t) = \frac{1}{N}\left\langle \left|\sum\nolimits_k \hat{\mathbf{n}}_k\right| \right\rangle_t , \tag{16}$$

along with its x, y components defined by $\mathbf{M} = N^{-1}\sum_k \langle \hat{\mathbf{n}}_k \rangle_t$. Averages $\langle \cdot \rangle_t$ are performed with respect to the ansatz state at time t. In addition, corresponding circular variances were defined as $\mathrm{Var}(\theta_k) = -2\ln|\langle \hat{\mathbf{n}}_k \rangle_t|$ and averaged over the lattice index k.

These observables were chosen as a proxy for thermalization. Across a wide range of quenches we observe convergence to their respective equilibrium values at $g = g_f$, see Fig. 2. We observe two distinct dynamical regimes in relation to the quantum critical point $g_c \approx 4.25$, when $g_i < g_c$. For small quenches (left column of Fig. 2) we see the expected outcome – slower equilibriation with only small fluctuations in the direction of the magnetization. However, for moderate to large quenches in Fig. 2, we observe a (transient) demagnetization of the sample and convergence to a new equilibrium state.

In addition, we define a measure of average vorticity

$$v(A) = \frac{1}{|A|}\int_A \mathrm{d}\mathbf{a} \cdot \nabla \times \hat{\mathbf{n}} = \frac{1}{|A|}\oint_{\partial A} \mathrm{d}\boldsymbol{\ell} \cdot \hat{\mathbf{n}} \tag{17}$$

over a surface A with edge ∂A on the lattice. Using Stokes' theorem, we rewrite the expression as a contour integral over ∂A in the positive direction. On Fig. 3 (right panel),

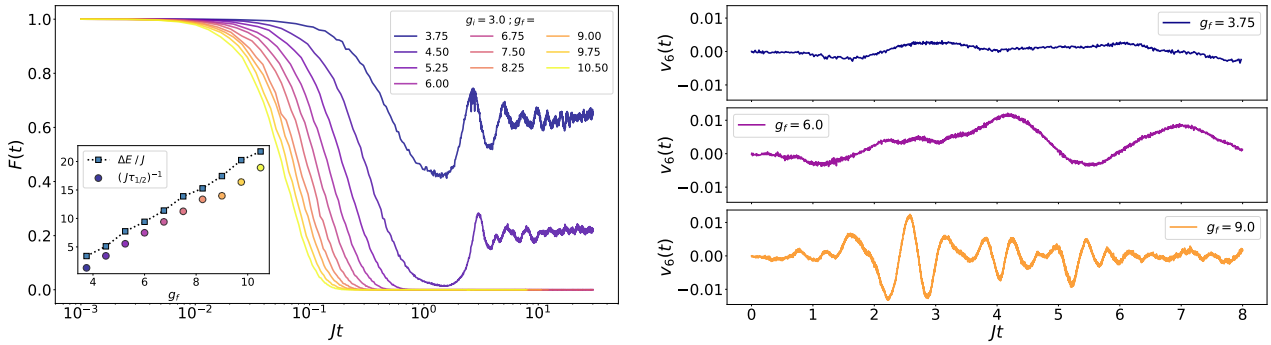

Figure 3. Fidelity and vorticity as functions of time. **Left**: Time-dependent many-body fidelity $F(t)$ defined in Eq. 18, for a number of quenches. For trajectories quenching to values of g_f in the same equilibrium phase, we see convergence to nonzero values at late times. Conversely, trajectories with $g_f > g_c$ converge to $F(t \to \infty) = 0$. Additionally, $\tau_{1/2}$ (the time it takes for fidelity to decrease by 50%) is shown to scale linearly with g in agreement with the appropriate uncertainty relation $\Delta E \Delta t \geq 1/2$. **Right**: The onset of vorticity (defined in Eq. 17) for three quenches of increasing magnitude.

we plot $v(A)$ averaged over all n_ℓ square $\ell \times \ell$ surfaces: $v_\ell = n_\ell^{-1} \sum_{|A|=\ell^2} v(A)$. As expected, we find almost zero vorticity for quenches in the ordered phase, while larger fluctuations are generated for quenches across the critical point. We postpone a detailed analysis for future work.

Aside from local observables, such as energy and magnetization, one also has access to global observables such as the Loschmidt echo. The latter has some interesting properties in the context of dynamical phase transitions (Heyl et al., 2013) and quantum chaos (Schmitt et al., 2019). The Loschmidt echo expresses the quantum state overlap between the initial state and some time-evolved state. In general, the fidelity $F(\Psi, \Phi)$ between two generic normalized quantum states Ψ and Φ is defined as $F(\Psi, \Phi) = |\langle \Psi | \Phi \rangle|^2$. For real-time evolution, we expect the fidelity $F(\Psi(t = 0), \Psi(t))$ to decay as a function of time t, for any given initial state $|\Psi(t = 0)\rangle$.

To evaluate this quantity using Monte Carlo sampling of unnormalized ansatz wavefunctions $\psi(\boldsymbol{\theta}, t) = \psi_{\alpha(t)}(\boldsymbol{\theta})$, we rewrite the fidelity definition as:

$$F(t) = \left\langle \frac{\psi(\boldsymbol{\theta}, t)}{\psi(\boldsymbol{\theta}, 0)} \right\rangle_{\boldsymbol{\theta} \sim |\psi(\cdot,0)|^2} \left\langle \frac{\psi(\boldsymbol{\theta}, 0)}{\psi(\boldsymbol{\theta}, t)} \right\rangle_{\boldsymbol{\theta} \sim |\psi(\cdot,t)|^2}, \tag{18}$$

which is manifestly independent of the normalization factor. In practice, we take the real part of Eq. 18 to discard the small nonzero imaginary part coming from finite-sample estimates of the two factors. In addition, we calculate and store both factors in log-space to preserve accuracy and maintain numerical stability.

As expected, we find that that the return probability (or fidelity in short) decays quickly with time, as illustrated in Fig. 3 (left panel). For smaller quenches, the fidelity shoots back up to a nonzero value suggesting a finite overlap between the initial state the long time "equilibrium" state after the quench. The latter may be interpreted as a signature of quenching between two Hamiltonians in the ordered phase.

As a measure of the fidelity decay, we introduce another time scale $\tau_{1/2}$ defined as the time needed for the fidelity to decrease by 50%. We observe that $\tau_{1/2}$ has increases linearly with the quench g_f. This result matches basic estimates given by the second-order short-time expansion of $F(t)$ and uncertainty relation $\Delta E \Delta t \geq 1/2$. Therefore, fidelity decay time can be lower bounded by ΔE^{-1}, estimated using samples from the initial state $\psi_{\alpha(0)}$ (Mandelstam & Tamm, 1991). Reference points from this calculation are presented in Fig. 3 (left, inset). This comparison demonstrates that the t-VMC method can be used to estimate quantities of experimental interest for system sizes unreachable by other wavefunction-based methods.

3.1. Benchmarks

To substantiate our results, we perform a series of benchmarks and compare results to tensor-network simulations for a one- and two-dimensional versions of the model. In particular we benchmark the results with the time-evolving block decimation (TEBD) (Vidal, 2003; 2004) algorithm. For all benchmarks, states were initialized to the coherent superposition of all basis states $|\psi(0)\rangle \propto \int \mathrm{d}\boldsymbol{\theta} \, |\boldsymbol{\theta}\rangle$ by explicitly setting the final convolution kernel w_D^c (Eq. 12) to zero. All presented tensor-network simulations have been performed with a fixed singular value cutoff. Convergence within the matrix product state (MPS) variational manifold has been confirmed by repeating simulations with larger cutoff values.

We organize numerical benchmarks as follows. First, we compare t-VMC results with TEBD for an extended one-dimensional and a smaller two-dimensional system. Practi-

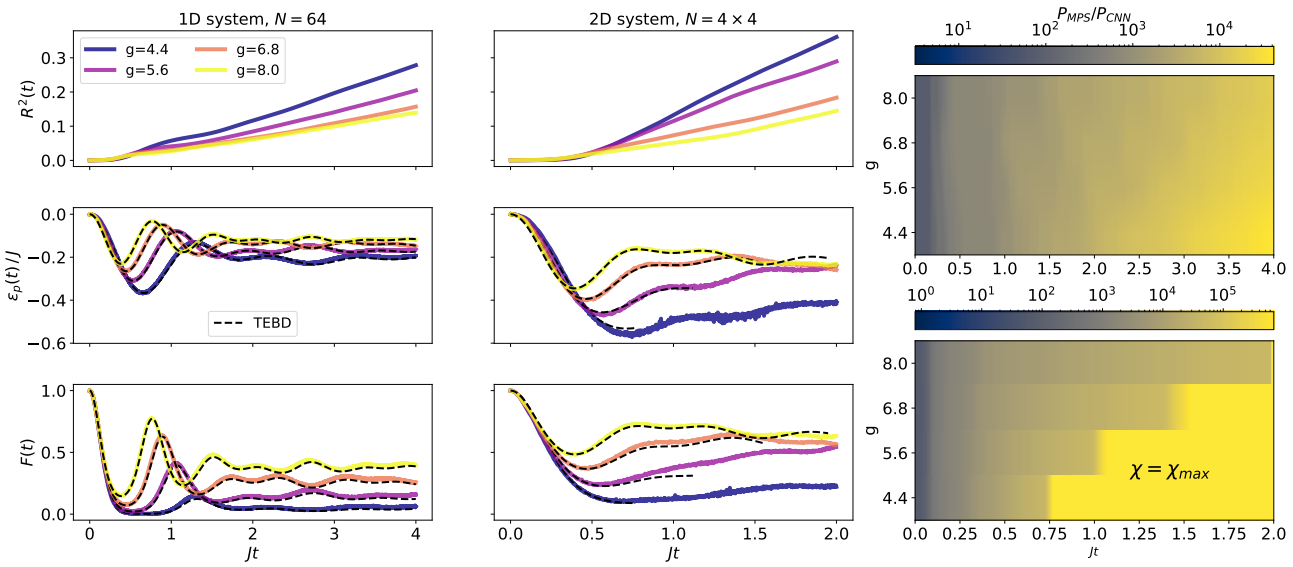

Figure 4. One- and two-dimensional benchmarks and comparison with tensor-network data. Evolution was performed starting from a coherent superposition state $|\psi(0)\rangle \propto \int \mathrm{d}\boldsymbol{\theta}\, |\boldsymbol{\theta}\rangle$. Results are compared with the TEBD tensor-network algorithm evolving a Matrix Product State (MPS) in the conjugate angular-momentum eigenbasis (see Sec. 3.1 and Appendix B). **Left**: A one-dimensional benchmark on a chain with $N = 64$ rotors and open boundary conditions. **Center**: A two-dimensional benchmark of the t-VMC method on a 4×4 lattice and open boundary conditions. We note that disagreement between t-VMC and TEBD results appears as the maximum bond dimension χ_{max} is reached. Singular value cutoff of 10^{-12} was used. **Right**: The growing number of MPS parameters P_{MPS} associated with the increasing bond dimension χ is plotted in units of the number of the CNN parameter count P_{CNN} as a function of time. One- and two-dimensional cases are compared. A cutoff of $\chi_{\text{max}} = 1000$ was reached in the 2D system for the singular value cutoff of 10^{-9}.

cal error estimates are defined. Then, we turn to examining effects of key hyperparameters in the t-VMC approach and show evidence of self-consistent convergence.

Following Refs. (Schmitt & Heyl, 2020; Carleo & Troyer, 2017), we use the following figure of merit:

$$r(t) = \frac{\mathcal{D}\left(\psi(t+\delta t), e^{-iH\delta t}\,\psi(t)\right)}{\mathcal{D}\left(\psi(t), e^{-iH\delta t}\,\psi(t)\right)} \tag{19}$$

where $|\psi(t)\rangle = \left|\psi_{\alpha(t)}\right\rangle$. In Eq. 19, $\mathcal{D}(\cdot,\cdot)$ represents the Fubini-Study distance on the Hilbert space $\mathcal{H}$. We estimate $r^2(t)$ at each time t using HMC samples from the ansatz (see Ref. (Schmitt & Heyl, 2020) and Appendix D). Intuitively, $r^2(t)$ measures an appropriately normalized measure of deviation between the full state $e^{-iH\delta t}\,|\psi(t)\rangle$ after one time step δt and its projection onto the variational manifold $\left|\psi_{\alpha(t+\delta t)}\right\rangle$. We plot the integrated square error $R^2(t) = \int_0^t r^2(s)\,\mathrm{d}s$ to reflect error propagation through time as accurately as possible. We remark that $R^2(t)$ should be interpreted an upper bound on the square of the integrated error $R(t) = \int_0^t r(s)\,\mathrm{d}s$ due to the triangle inequality.

In Fig. 4 (left), we show that this algorithm performs well on a one-dimensional system of $N = 64$ rotors where the growth of the so-called *bond dimension* χ is limited. Convergence to appropriate equilibrium values is reached for both methods with good agreement at intermediate times for the dynamics of potential energy density $\epsilon_{\text{p}}(t)$ and the Loschmidt echo $F(t)$. The *integrated residual* $R^2(t)$ grows more rapidly for lower values of g. This is expected because the initial state $\psi(0)$ has lower energy for larger values of g in the QRM hamiltonian, Eq. 1, representing a more typical state in the disordered phase.

In contrast to the 1D case, in Fig. 4 (center), we observe that the TEBD method exponentially grows the MPS bond dimension χ past the cutoff $\chi_{\text{max}} = 1000$ at relatively short times. We plot the number of parameters P_{MPS} in the MPS as a function of time in the right panel of Fig. 4, in units of the number of parameters P_{CNN} in the CNN ansatz presented in this work. We see qualitative agreement between the two methods for early times, before χ grows to the point where further simulation is numerically prohibitively expensive.

In Fig. 5 we show evidence that the variance of observables is controllable through the most important Monte Carlo (HMC) hyperparameters while the bias is mostly controlled by different regularizations of the S-matrix inverse (Eq. 6). In the top panel of Fig. 5, we see that the standard deviation of the estimator for total magnetization $M(t)$ scales with the number of HMC samples N_{s} in an expected way: $\sigma_M \propto$

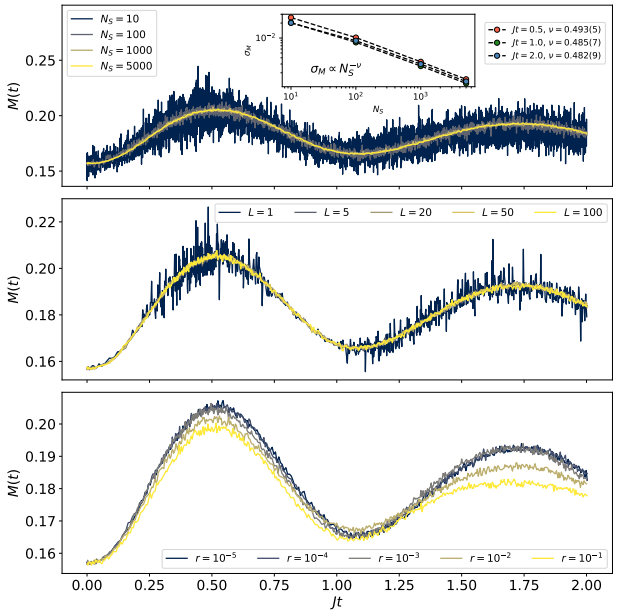

Figure 5. Effects of key hyperparameters on magnetization measurements. All experiments were performed on a one-dimensional chain with $N = 32$. **Top**: Effects on magnetization estimates by varying the number of HMC samples N_s. Errors were estimated using bootstrap resampling independently at different times show expected scaling $\sigma_M \propto N_s^{-1/2}$ in all cases. **Middle**: Variance change in magnetization estimates by varying the number of leapfrog integrator steps L between HMC proposals. In the $L \to 1$ limit, HMC approaches the random-walk Metropolis sampler. **Bottom**: Bias increase associated with changing the pseudoinverse cutoff parameter r_c (see Appendix A.2).

$N_s^{-1/2}$ for three different times during the evolution.

In addition, we report that heuristically varying the number of leapfrog integration steps L increases estimator variances the most around segments of trajectories with higher curvature, as evidenced by the middle panel of Fig. 5. Intuitively, in the limit of $L \to 1$ and small leapfrog step sizes ε, HMC approaches random-walk Metropolis sampling (see Ref. (Betancourt, 2017) and Appendix A.1) which suffers from lower acceptance rates and longer mixing times in cases of sharply-peaked target distributions. We observe that even a moderate increase to $L \approx 10$ accompanied by automatic hyperparameter tuning described in Sec. 2.2 considerably reduces variance.

Finally, we explore the effects of S-matrix regularization (see Appendix A.2 for details). In practice, when calculating the pseudoinverse, we discard all eigenvalues of S smaller than r_c times the maximum eigenvalue. In the bottom panel of Fig. 5, we see that, increasing r_c leads to increased estimator bias – excluding relevant eigenvalues from participating in time evolution can lead to a failure to capture relevant physics.

Overall, both t-VMC and TEBD algorithms predict similar dynamical behavior of the potential energy density (Eq. 15) and the fidelity (Eq. 18), as shown on Fig. 4. However, the number of parameters in the MPS grows exponentially due to entropy build-up during time-evolution. Tensor-network real-time evolution algorithms (Haegeman et al., 2011; 2016) based on MPS or two-dimensional architectures such as projected entangled pair states (PEPS) (Verstraete & Cirac, 2004; Czarnik et al., 2019) face several challenges to extend to late times and higher dimensions. Incorporating continuous degrees of freedom exacerbates the problem – tensor network algorithms are limited to using the locally truncated eigenbasis of the angular momentum operator L_k in the QRM hamiltonian in Eq. 1, in contrast to the t-VMC method (see Appendix B).

4. Conclusion

We present a method to approximate unitary dynamics of continuous-variable quantum many-body systems, based on custom neural-network quantum states. The approach employs Hamiltonian Monte Carlo sampling and custom regularization of the quantum geometric tensor. The method was benchmarked on quench dynamics of two-dimensional quantum rotors. We indicated that our calculations are able to access non-local quantities like the return probability. Good agreement was found with tensor-network-based TEBD simulations for the case of one-dimensional systems of comparable size. Finally, we showed evidence that the method is controlled by a handful of key hyperparameters. Our approach paves the way for accurate non-equilibrium simulations of continuous systems at previously unexplored system sizes and evolution times, bridging the gap between simulation and experiment.

5. Acknowledgements

MM acknowledges insightful discussions with Filippo Vicentini about t-VMC regularization, Bob Carpenter about the role of circular geometry in Monte Carlo sampling and Hamiltonian Monte Carlo details. In addition, discussions with Sandro Sorella about the infinite variance problem and James Stokes about different ansatze were very helpful for fine-tuning simulations. MM also acknowledges support from the CCQ graduate fellowship in computational quantum physics. The Flatiron Institute is a division of the Simons Foundation. DS was supported by AFOSR: Grant FA9550-21-1-0236 and NSF: Grant OAC-2118310.

Software libraries

The code used in this work has been packaged into an installable library and is publicly available to reproduce any results in this work or explore new ones: github.com/Matematija/continuous-vmc.

It was built on JAX (Bradbury et al., 2018) for array manipulations, automatic differentiation for sampling and optimization and GPU support, Flax (Heek et al., 2020) for neural-network manipulation and NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020) for CPU array manipulations. Matplotlib (Hunter, 2007) was used to produce figures.

References

Amari, S. I. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 2 1998. ISSN 08997667. doi: 10.1162/089976698300017746. URL http://www.mitpressjournals.org/doi/10.1162/089976698300017746.

Barison, S., Vicentini, F., and Carleo, G. An efficient quantum algorithm for the time evolution of parameterized circuits. *Quantum*, 5:512, 7 2021. ISSN 2521-327X. doi: 10.22331/q-2021-07-28-512.

Becca, F. and Sorella, S. *Quantum Monte Carlo Approaches for Correlated Systems*. Cambridge University Press, 11 2017. ISBN 9781107129931. doi: 10.1017/9781316417041.

Berke, C., Varvelis, E., Trebst, S., Altland, A., and DiVincenzo, D. P. Transmon platform for quantum computing challenged by chaotic fluctuations. *Nature Communications*, 13(1):2495, 2022. doi: 10.1038/s41467-022-29940-y. URL https://doi.org/10.1038/s41467-022-29940-y.

Betancourt, M. A conceptual introduction to hamiltonian monte carlo. 1 2017. URL http://arxiv.org/abs/1701.02434.

Bogacki, P. and Shampine, L. A 3(2) pair of runge - kutta formulas. *Applied Mathematics Letters*, 2:321–325, 1 1989. ISSN 08939659. doi: 10.1016/0893-9659(89)90079-7.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Budden, M., Gebert, T., Buzzi, M., Jotzu, G., Wang, E., Matsuyama, T., Meier, G., Laplace, Y., Pontiroli, D., Riccò, M., Schlawin, F., Jaksch, D., and Cavalleri, A. Evidence for metastable photo-induced superconductivity in k3c60. *Nature Physics*, 17(5):611–618, 2021. doi: 10.1038/s41567-020-01148-1. URL https://doi.org/10.1038/s41567-020-01148-1.

Bukov, M., Day, A. G. R., Sels, D., Weinberg, P., Polkovnikov, A., and Mehta, P. Reinforcement learning in different phases of quantum control. *Physical Review X*, 8:031086, 9 2018. ISSN 2160-3308. doi: 10.1103/PhysRevX.8.031086. URL https://link.aps.org/doi/10.1103/PhysRevX.8.031086.

Butcher, J. C. Coefficients for the study of runge-kutta integration processes. *Journal of the Australian Mathematical Society*, 3:185–201, 5 1963. ISSN 0004-9735. doi: 10.1017/S1446788700027932.

Butcher, J. C. *Numerical Methods for Ordinary Differential Equations*. John Wiley & Sons, Ltd, 3 2008. ISBN 9780470753767. doi: 10.1002/9780470753767.

Carleo, G. and Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science*, 355:602–606, 2017. ISSN 10959203. doi: 10.1126/science.aag2302.

Carleo, G., Becca, F., Schiró, M., and Fabrizio, M. Localization and glassy dynamics of many-body quantum systems. *Scientific Reports*, 2:243, 12 2012. ISSN 2045-2322. doi: 10.1038/srep00243. URL http://www.nature.com/articles/srep00243.

Carleo, G., Cevolani, L., Sanchez-Palencia, L., and Holzmann, M. Unitary dynamics of strongly interacting bose gases with the time-dependent variational monte carlo method in continuous space. *Physical Review X*, 7:031026, 8 2017. ISSN 21603308. doi: 10.1103/PHYSREVX.7.031026/FIGURES/5/MEDIUM. URL https://journals.aps.org/prx/abstract/10.1103/PhysRevX.7.031026.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., Vogt-Maranto, L., and Zdeborová, L. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91:045002, 12 2019. ISSN 0034-6861. doi: 10.1103/RevModPhys.91.045002. URL https://link.aps.org/doi/10.1103/RevModPhys.91.045002.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76, 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01. URL http://www.jstatsoft.org/v76/i01/.

Czarnik, P., Dziarmaga, J., and Corboz, P. Time evolution of an infinite projected entangled pair state: An

efficient algorithm. *Physical Review B*, 99:035115, 1 2019. ISSN 2469-9950. doi: 10.1103/PhysRevB.99.035115. URL https://link.aps.org/doi/10.1103/PhysRevB.99.035115.

Czischek, S., Gärttner, M., and Gasenzer, T. Quenches near ising quantum criticality as a challenge for artificial neural networks. *Phys. Rev. B*, 98:024311, Jul 2018. doi: 10.1103/PhysRevB.98.024311. URL https://link.aps.org/doi/10.1103/PhysRevB.98.024311.

Haegeman, J., Cirac, J. I., Osborne, T. J., Pižorn, I., Verschelde, H., and Verstraete, F. Time-dependent variational principle for quantum lattices. *Physical Review Letters*, 107:070601, 8 2011. ISSN 0031-9007. doi: 10.1103/PhysRevLett.107.070601.

Haegeman, J., Lubich, C., Oseledets, I., Vandereycken, B., and Verstraete, F. Unifying time evolution and optimization with matrix product states. *Physical Review B*, 94:165116, 10 2016. ISSN 2469-9950. doi: 10.1103/PhysRevB.94.165116.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, sep 2020. ISSN 14764687. doi: 10.1038/s41586-020-2649-2.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109, 1970. ISSN 00063444. doi: 10.1093/biomet/57.1.97.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2020. URL http://github.com/google/flax.

Heyl, M., Polkovnikov, A., and Kehrein, S. Dynamical quantum phase transitions in the transverse-field ising model. *Phys. Rev. Lett.*, 110:135704, Mar 2013. doi: 10.1103/PhysRevLett.110.135704. URL https://link.aps.org/doi/10.1103/PhysRevLett.110.135704.

Hoffman, M. D. and Gelman, A. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15:1593–1623, 11 2011. ISSN 15337928. doi: 10.48550/arxiv.1111.4246. URL https://arxiv.org/abs/1111.4246v1.

Hofmann, D., Fabiani, G., Mentink, J., Carleo, G., and Sentef, M. Role of stochastic noise and generalization error in the time propagation of neural-network quantum states. *SciPost Physics*, 12:165, 5 2022. ISSN 2542-4653. doi: 10.21468/SciPostPhys.12.5.165. URL https://scipost.org/10.21468/SciPostPhys.12.5.165.

Hubig, C., Bohrdt, A., Knap, M., Grusdt, F., and Cirac, I. Evaluation of time-dependent correlators after a local quench in ipeps: hole motion in the t-j model. *SciPost Physics*, 8:021, 2 2020. ISSN 2542-4653. doi: 10.21468/SciPostPhys.8.2.021. URL https://scipost.org/10.21468/SciPostPhys.8.2.021.

Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.*, 9(3):99–104, 2007. ISSN 15219615. doi: 10.1109/MCSE.2007.55.

Jiang, W., Pan, G., Liu, Y., and Meng, Z. Y. Solving quantum rotor model with different monte carlo techniques. *Chinese Physics B*, 31, 12 2019. doi: 10.1088/1674-1056/ac4f52. URL http://arxiv.org/abs/1912.08229.

Josephson, B. Possible new effects in superconductive tunnelling. *Physics Letters*, 1(7):251–253, jul 1962. ISSN 00319163. doi: 10.1016/0031-9163(62)91369-0.

Josephson, B. Supercurrents through barriers. *Advances in Physics*, 14(56):419–451, oct 1965. ISSN 0001-8732. doi: 10.1080/00018736500101091.

José, J. V., Kadanoff, L. P., Kirkpatrick, S., and Nelson, D. R. Renormalization, vortices, and symmetry-breaking perturbations in the two-dimensional planar model. *Physical Review B*, 16:1217–1241, 8 1977. ISSN 0556-2805. doi: 10.1103/PhysRevB.16.1217. URL https://link.aps.org/doi/10.1103/PhysRevB.16.1217.

Kockum, A. F. and Nori, F. Quantum bits with josephson junctions. *Springer Series in Materials Science*, 286:703–741, 8 2019. ISSN 21962812. doi: 10.1007/978-3-030-20726-7_17. URL http://dx.doi.org/10.1007/978-3-030-20726-7_17.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *Nature*, 521:436–44, 5 2015. ISSN 1476-4687. doi: 10.1038/nature14539. URL http://www.ncbi.nlm.nih.gov/pubmed/26017442.

Mandelstam, L. and Tamm, I. *The Uncertainty Relation Between Energy and Time in Non-relativistic Quantum Mechanics*, pp. 115–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. ISBN 978-3-642-74626-0. doi: 10.1007/978-3-642-74626-0_8. URL https://doi.org/10.1007/978-3-642-74626-0_8.

Martinoli, P. and Leemann, C. Two Dimensional Josephson Junction Arrays. *Journal of Low Temperature Physics*, 118(5-6):699–731, 2000. ISSN 00222291. doi: 10.1023/a:1004651730459. URL https://link.springer.com/article/10.1023/A:1004651730459.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21:1087–1092, 1953. ISSN 00219606. doi: 10.1063/1.1699114.

Metz, F. and Bukov, M. Self-correcting quantum many-body control using reinforcement learning with tensor networks. 1 2022. URL http://arxiv.org/abs/2201.11790.

Neal, R. M. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 5 2011. ISBN 9780429138508. doi: 10.1201/b10905. URL https://www.taylorfrancis.com/books/9781420079425.

Nesterov, Y. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 8 2009. ISSN 0025-5610. doi: 10.1007/s10107-007-0149-x. URL http://link.springer.com/10.1007/s10107-007-0149-x.

Pescia, G., Han, J., Lovato, A., Lu, J., and Carleo, G. Neural-network quantum states for periodic systems in continuous space. *Physical Review Research*, 4:023138, 5 2022. ISSN 2643-1564. doi: 10.1103/PhysRevResearch.4.023138. URL https://link.aps.org/doi/10.1103/PhysRevResearch.4.023138.

Polkovnikov, A., Sengupta, K., Silva, A., and Vengalattore, M. Colloquium: Nonequilibrium dynamics of closed interacting quantum systems. *Rev. Mod. Phys.*, 83:863–883, Aug 2011. doi: 10.1103/RevModPhys.83.863. URL https://link.aps.org/doi/10.1103/RevModPhys.83.863.

Porotti, R., Essig, A., Huard, B., and Marquardt, F. Deep reinforcement learning for quantum state preparation with weak nonlinear measurements. *Quantum*, 6:747, 6 2022. ISSN 2521-327X. doi: 10.22331/q-2022-06-28-747.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C*. Cambridge University Press, Cambridge, USA, second edition, 1992.

Schmitt, M. and Heyl, M. Quantum many-body dynamics in two dimensions with artificial neural networks. *Physical Review Letters*, 125:100503, 9 2020. ISSN 0031-9007. doi: 10.1103/PhysRevLett.125.100503. URL https://link.aps.org/doi/10.1103/PhysRevLett.125.100503.

Schmitt, M., Sels, D., Kehrein, S., and Polkovnikov, A. Semiclassical echo dynamics in the sachdev-ye-kitaev model. *Phys. Rev. B*, 99:134301, Apr 2019. doi: 10.1103/PhysRevB.99.134301. URL https://link.aps.org/doi/10.1103/PhysRevB.99.134301.

Sivak, V. V., Eickbusch, A., Liu, H., Royer, B., Tsioutsios, I., and Devoret, M. H. Model-free quantum control with reinforcement learning. *Physical Review X*, 12:011059, 3 2022. ISSN 2160-3308. doi: 10.1103/PhysRevX.12.011059. URL https://link.aps.org/doi/10.1103/PhysRevX.12.011059.

Sorella, S. Green function monte carlo with stochastic reconfiguration. *Physical Review Letters*, 80:4558–4561, 5 1998. ISSN 0031-9007. doi: 10.1103/PhysRevLett.80.4558. URL https://link.aps.org/doi/10.1103/PhysRevLett.80.4558.

Stokes, J., Izaac, J., Killoran, N., and Carleo, G. Quantum natural gradient. *Quantum*, 4:269, 5 2020. ISSN 2521327X. doi: 10.22331/Q-2020-05-25-269. URL https://quantum-journal.org/papers/q-2020-05-25-269/.

Stokes, J., De, S., Veerapaneni, S., and Carleo, G. Continuous-variable neural-network quantum states and the quantum rotor model. 7 2021. URL http://arxiv.org/abs/2107.07105.

Thomas, A., Lethuillier-Karl, L., Nagarajan, K., Vergauwe, R. M. A., George, J., Chervy, T., Shalabney, A., Devaux, E., Genet, C., Moran, J., and Ebbesen, T. W. Tilting a ground-state reactivity landscape by vibrational strong coupling. *Science*, 363(6427):615–619, 2019. doi: 10.1126/science.aau7742. URL https://www.science.org/doi/abs/10.1126/science.aau7742.

Verstraete, F. and Cirac, J. I. Renormalization algorithms for quantum-many body systems in two and higher dimensions. 7 2004. doi: 10.48550/arxiv.cond-mat/0407066. URL http://arxiv.org/abs/cond-mat/0407066.

Vidal, G. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91(14):147902, oct 2003. ISSN 10797114. doi: 10.1103/PhysRevLett.91.147902. URL https://link.aps.org/doi/10.1103/PhysRevLett.91.147902.

Vidal, G. Efficient simulation of one-dimensional quantum many-body systems. *Physical Review Letters*, 93(4):040502, jul 2004. ISSN 00319007.

doi: 10.1103/PhysRevLett.93.040502. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.93.040502.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., İlhan Polat, Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., de Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., and Vázquez-Baeza, Y. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17:261–272, 3 2020. ISSN 1548-7091. doi: 10.1038/s41592-019-0686-2. URL http://www.nature.com/articles/s41592-019-0686-2.

Vogt, N., Schäfer, R., Rotzinger, H., Cui, W., Fiebig, A., Shnirman, A., and Ustinov, A. V. One-dimensional Josephson junction arrays: Lifting the Coulomb blockade by depinning. *Physical Review B*, 92(4):045435, jul 2015. ISSN 1098-0121. doi: 10.1103/PhysRevB.92.045435. URL http://dx.doi.org/10.1103/PhysRevB.92.045435.

Warren, W. S., Rabitz, H., and Dahleh, M. Coherent control of quantum dynamics: The dream is alive. *Science*, 259 (5101):1581–1589, 1993. doi: 10.1126/science.259.5101.1581.

White, S. R. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863–2866, nov 1992. ISSN 00319007. doi: 10.1103/PhysRevLett.69.2863. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.69.2863.

Wurtz, J., Polkovnikov, A., and Sels, D. Cluster truncated wigner approximation in strongly interacting systems. *Annals of Physics*, 395:341–365, 2018. ISSN 0003-4916. doi: https://doi.org/10.1016/j.aop.2018.06.001. URL https://www.sciencedirect.com/science/article/pii/S0003491618301647.

Yuan, X., Endo, S., Zhao, Q., Li, Y., and Benjamin, S. C. Theory of variational quantum simulation. *Quantum*, 3:191, 10 2019. ISSN 2521327X. doi: 10.22331/q-2019-10-07-191. URL https://quantum-journal.org/papers/q-2019-10-07-191/.

Zaletel, M. P., Mong, R. S. K., Karrasch, C., Moore, J. E., and Pollmann, F. Time-evolving a matrix product state with long-ranged interactions. *Phys. Rev. B*, 91:165112, Apr 2015. doi: 10.1103/PhysRevB.91.165112. URL https://link.aps.org/doi/10.1103/PhysRevB.91.165112.

Zhang, J., Pagano, G., Hess, P. W., Kyprianidis, A., Becker, P., Kaplan, H., Gorshkov, A. V., Gong, Z. X., and Monroe, C. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature*, 551 (7682):601–604, 2017. doi: 10.1038/nature24654. URL https://doi.org/10.1038/nature24654.

Zhou, Y., Stoudenmire, E. M., and Waintal, X. What limits the simulation of quantum computers? *Phys. Rev. X*, 10:041038, Nov 2020. doi: 10.1103/PhysRevX.10.041038. URL https://link.aps.org/doi/10.1103/PhysRevX.10.041038.

A. Simulation details

In this appendix, we mention some of the details of numerical simulations performed in this work that have not been discussed in the main text. We also clearly state different hyperparameters and their observed effect on performance and numerical stability.

A.1. Hamiltonian Monte Carlo details

As noted in the main text, the Hamiltonian Monte Carlo (HMC) algorithm used in this work has many important hyperparameters. To define the proposal, we must specify: the leapfrog integration length L, leapfrog step size ε the mass matrix M. We assume that the mass matrix is diagonal $M = \mathrm{diag}(m_1, \ldots, m_N)$. Samples are proposed by numerically integrating Hamilton's equations corresponding to the hamiltonian $\tilde{H} = \frac{1}{2}\boldsymbol{\pi}^\top M^{-1}\boldsymbol{\pi} - \ln V(\boldsymbol{\theta})$ along an fictitious time axis τ, unrelated to t in Eq. 5:

$$\begin{aligned} \boldsymbol{\pi}(\tau + \varepsilon/2) &= \boldsymbol{\pi}(\tau) - \frac{\varepsilon}{2}\frac{\partial V}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}(\tau)) \\ \boldsymbol{\theta}(\tau + \varepsilon) &= \boldsymbol{\theta}(\tau) + \varepsilon\, M^{-1}\, \boldsymbol{\pi}(\tau + \varepsilon/2) \\ \boldsymbol{\pi}(\tau + \varepsilon) &= \boldsymbol{\pi}(\tau + \varepsilon/2) - \frac{\varepsilon}{2}\frac{\partial V}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}(\tau + \varepsilon)) \end{aligned} \tag{20}$$

We fix L heuristically and adaptively set M and ε during an extended *warm-up* phase for each Markov chain independently. In other words, before any samples are collected for evaluation of Eq. 6, each chain is run for N_w steps. Following the popular software package *Stan* (Carpenter et al., 2017), we subdivide the warmup period into $N_\mathrm{p} + 2$ phases (*windows*), each of which is one of two types:

- **Fast**: Samples are collected and only step size ε is adapted using the online optimization algorithm in Ref. (Nesterov, 2009). Mass matrix remains unchanged. Fast windows are used to efficiently *initialize* the chain by moving it towards a typical set of highly-probable samples.
- **Slow**: Samples are collected and both step size ε and the mass matrix M are estimated. Step size is estimated the same way as in the fast window. Mass matrix elements are estimated as the variance of corresponding variables: $m_k = \mathrm{Var}(\theta_k)$ using the appropriate formula for the variance of periodic random variables presented in the main text, Sec. 3.

After initializing each $\theta_k \sim \mathrm{Uniform}(-\pi, \pi)$, we begin the warm-up phase with a single fast window of length $N_\mathrm{w}/12$, followed by 5 fast windows. The first fast window is $N_\mathrm{w}/36$ steps long with each subsequent slow window doubling in size. Finally, we end the warm-up by running an additional fast window for the remaining $N_\mathrm{w}/18$ steps. After each window, the HMC transition kernel (the leapfrog ODE solver) is updated with adapted values for ε and M (for slow windows). After the final fast window, all hyperparameters are locked in and actual collection of the N_s for Eq. 6 begins. The full list of relevant hyperparameters can be found in Table A.1.

We use automatic differentiation (using JAX (Bradbury et al., 2018)) to obtain numerically exact gradients $\nabla_{\boldsymbol{\theta}} \ln p(\boldsymbol{\theta}, t)$ of needed to run the leapfrog integrator. To avoid loss of accuracy or numerical instabilities through exponentiation, we employ the following identity:

$$\ln p(\boldsymbol{\theta}, t) = \ln \left|\psi_{\alpha(t)}(\boldsymbol{\theta})\right|^2 = 2\,\mathrm{Re}\left\{\ln \psi_{\alpha(t)}(\boldsymbol{\theta})\right\} \ , \tag{21}$$

when the logarithm of the wavefunction is parameterized instead of the wavefunction itself.

For completeness, we note that a common precaution against leapfrog integration getting stuck in regions of high curvature used in this work. Instead of fixing the integration length to a specific value $L = L_0$, it is randomly chosen between $(1-\gamma)L_0$ and $(1+\gamma)L_0$ each time the integrator is called, with a new hyperparameter $0 \leq \gamma < 1$. This *jittering* of trajectory lengths can help HMC walkers move away from regions of high curvature if they get stuck (Betancourt, 2017; Neal, 2011; Carpenter et al., 2017). To collect more independent samples by utilizing modern massively-parallel GPU hardware, we run N_c such chains in parallel, each one warmed up independently.

Finally, we note that the HMC proposal generated by the leapfrog integrator approaches the random-walk Metropolis (RWM) update:

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \sqrt{\Sigma}\,\mathbf{z}\ ; \quad \mathbf{z} \sim \mathcal{N}(0, \mathbb{1})\ , \tag{22}$$

Symbol	Name	Value	Domain	Description
ε	Step size	Dynamically adapted	$\mathbb{R}$	The leapfrog integrator step size.
M	Mass matrix	Dynamically adapted	$\mathbb{R}^{N^2}$	The covariance (metric) tensor of the dummy momentum variables $\boldsymbol{\pi}$.
L	(Average) integration length	20	$\mathbb{N}$	The number of leapfrog steps taken before proposing a sample. (If $\gamma > 0$, we relabel $L \to L_0$.)
γ	Jitter	0.2	$[0, 1\rangle$	Randomness for L during sampling – it is drawn uniformly between $(1-\gamma)L_0$ and $(1+\gamma)L_0$.
ε_0	Initial step size	0.1	$\mathbb{R}$	A guess for the value of ε to refine during the warmup phase.
δ	Target acceptance rate	0.8	$[0, 1]$	Target acceptance rate used for optimization of ε by algorithm in Ref. (Nesterov, 2009).
N_w	Length of warmup phase	800	$\mathbb{N}$	Total number of MC samples used for extended warmup.
N_p	Number of slow windows	5	$\mathbb{N}$	Total number of slow adaptation windows during warmup.
N_s	Number of samples	2000	$\mathbb{N}$	Total number of samples (per chain).
N_c	Number of chains	20	$\mathbb{N}$	Total number of independent Markov chains.

Table 1. The list of relevant hyperparameters for the Hamiltonian Monte Carlo algorithm with their values used in this work.

in the limit of few leapfrog integrator steps: $L \to 1$. Indeed, for $L = 1$ and small step sizes ε, Eq. 20 becomes

$$\boldsymbol{\theta}' = \boldsymbol{\theta}(\epsilon) = \boldsymbol{\theta}(0) + \varepsilon M^{-1}\boldsymbol{\pi}(\varepsilon/2) = \boldsymbol{\theta}(0) + \varepsilon M^{-1}\boldsymbol{\pi}(0) - \frac{\varepsilon^2}{2} M^{-1}\frac{\partial V}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}(0)) = \boldsymbol{\theta}(0) + \varepsilon M^{-1}\boldsymbol{\pi}(0) + \mathcal{O}(\varepsilon^2)\ , \quad (23)$$

where M^{-1} is equivalent in effect to the $\sqrt{\Sigma}$ matrix and $\boldsymbol{\pi}(0) \sim \mathcal{N}(0, M^{-1})$ by construction in Eq. 11.

A.2. Numerical regularization schemes

After evaluating the averages in Eq. 6 at time t, one needs to solve the linear system $i\,S\dot{\alpha} = g$ to obtain $\dot{\alpha}$ needed to progress to time $t + \delta t$. Since the S matrix is singular in most cases of interest, a robust regularization scheme is needed. As pointed out in the main text, replacing $S \to S + \epsilon\mathbb{1}$ is often enough in the case of ground-state searches (imaginary-time evolution). We remark that this is equivalent to the L2-regularized least-squares solution of $i\,L\dot{\alpha} = h$.

$$\dot{\alpha} = \underset{\dot{\alpha}\in\mathbb{C}^P}{\operatorname{argmin}} \left\{ \|i\,L\,\dot{\alpha} - h\|_2^2 + \epsilon\|\dot{\alpha}\|_2^2 \right\} \quad (24)$$

where $L^\dagger L = S$ is the Cholesky decomposition of the S-matrix (assuming S is positive-definite), $L^\dagger h = g$, and $\|\cdot\|_2$ is the standard euclidean 2-norm on $\mathbb{C}^P$.

We instead adopt a regularization scheme based on the spectrum of the S-matrix, $S = U\Sigma U^\dagger$, where $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_P^2)$. Our definition of the pseudoinverse is $S^{-1} \approx U\tilde{\Sigma}^{-1}U^\dagger$ with:

$$\tilde{\Sigma}^{-1}_{\mu\nu} = f(\sigma_\mu^2)\,\frac{\delta_{\mu\nu}}{\sigma_\mu^2} \qquad \text{and} \qquad f(\sigma^2) = \frac{1}{1 + \left(\lambda^2/\sigma^2\right)^6}\ . \quad (25)$$

In the limit of $\lambda^2 \to 0$, we recover the actual matrix inverse. As opposed to the more traditional choice of the step-function $f(\sigma^2) = \theta(\sigma^2 - \lambda^2)$, we find that choosing a smooth functional form for $f(\sigma^2)$ in Eq. 25 makes the adaptive time-stepping in the top-level integration routine (see Appendix A.3) more stable.

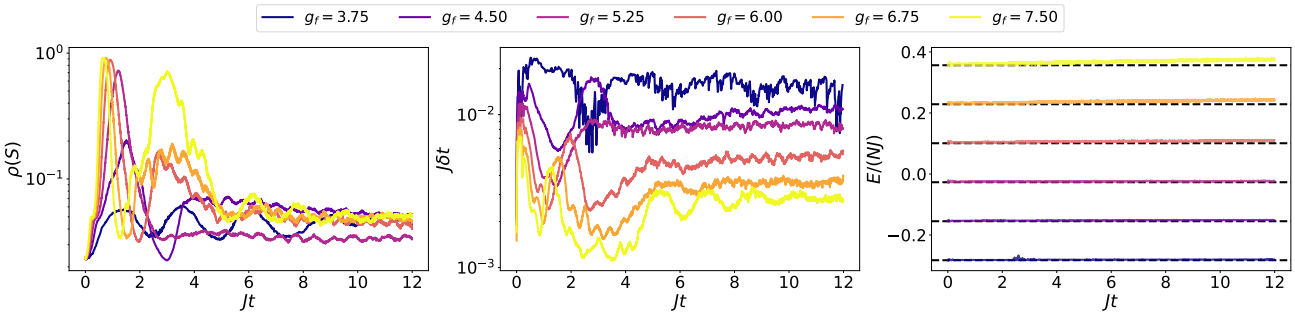

Figure 6. Some ODE integrator and regularization details. **Left**: The *effective rank* of the S-matrix $\rho(S)$ as a function of time, reflecting the internal dimensionality of the parameter manifold α, as discussed in Sec. A.2. **Center**: Time steps δt taken by the adaptive ODE integrator of choice. **Right**: Approximate conservation of energy as a function of time, for different quenches.

We set λ^2 to:

$$\lambda^2 = \lambda^2(\sigma_1^2, \ldots, \sigma_P^2) = \max\left(a_c,\ r_c \times \max_\mu(\sigma_\mu^2)\right) , \tag{26}$$

each iteration, with $a_c = 10^{-4}$ and $r_c = 10^{-2}$ chosen for 2D calculations and $a_c = 10^{-5}$ and $r_c = 10^{-4}$ for 1D benchmarks. To track potential over-regularization and as a measure of ansatz expressivity, we define the *effective rank* $\rho(S) = \sum_\mu f(\sigma_\mu^2)$. Intuitively, since $0 < f(\sigma^2) < 1$ for all eigenvalues σ^2, $\rho(S)$ can be interpreted as the effective number of eigenvalues that have not been set to zero by the regularization function f. In other words, it corresponds to the number of parameters in α that get updated at time t.

We plot $\rho(S)$ as a function of time on Fig. 6 for some simulated quenches. In all cases, we see that the effective rank increases rapidly to $\rho \sim 1$ at intermediate times that, for larger quenches, correspond to rapid oscillations and onset of vorticity. In those cases, it is natural to interpret this regime as almost all parameters α being important to capture the relevant physics. At later times, ρ converges to values below 10%, indicating equilibration and less oscillatory behavior.

For completeness, we mention that alternative regularization techniques have been explored as well. For example, the method of Schmitt and Heyl in Ref. (Schmitt & Heyl, 2020), based on the signal-to-noise ratio (SNR) for each eigenvalue in σ^2, represents a computationally and physically well-motivated approach. However, it did not bring any measurable performance improvement in our case.

A.3. The time-dependent variational principle and ODE integrators

To make use of the TDVP action in Eq. 5 in the main text, to propagate the variational parameters forward in time, one must construct the corresponding Euler-Lagrange equations. To this end, we first manipulate the action into a more transparent form:

$$\delta\mathcal{C}[\alpha] = \delta \int \mathrm{d}t \left\langle \Psi_{\alpha(t)} \right| \left(i\frac{\mathrm{d}}{\mathrm{d}t} - H \right) \left| \Psi_{\alpha(t)} \right\rangle = \tag{27}$$

$$= \delta \int \mathrm{d}t \left\{ \frac{i}{2} \frac{\left\langle \psi_{\alpha(t)} \middle| \dot{\psi}_{\alpha(t)} \right\rangle - \left\langle \dot{\psi}_{\alpha(t)} \middle| \psi_{\alpha(t)} \right\rangle}{\left\langle \psi_{\alpha(t)} \middle| \psi_{\alpha(t)} \right\rangle} - \frac{\left\langle \psi_{\alpha(t)} \right| H \left| \psi_{\alpha(t)} \right\rangle}{\left\langle \psi_{\alpha(t)} \middle| \psi_{\alpha(t)} \right\rangle} \right\} = \tag{28}$$

$$= -\delta \int \mathrm{d}t \int \mathrm{d}\boldsymbol{\theta}\ \frac{\left| \left\langle \boldsymbol{\theta} \middle| \psi_{\alpha(t)} \right\rangle \right|^2}{\left\langle \psi_{\alpha(t)} \middle| \psi_{\alpha(t)} \right\rangle} \left\{ \mathrm{Im} \sum_\mu \mathcal{O}_\mu(\boldsymbol{\theta}, t)\dot{\alpha}_\mu(t) + E_L(\boldsymbol{\theta}, t) \right\} \propto \tag{29}$$

$$\propto \int \mathrm{d}t \sum_\mu \left\{ i \sum_\nu S_{\mu\nu}(t)\, \dot{\alpha}_\nu(t) - g_\mu(t) \right\} \delta\alpha_\mu(t) - \text{c.c.} , \tag{30}$$

where we used definitions of $S_{\mu\nu}$, g_μ and E_L from Eqs. 8 and 6, respectively as well as $\dot{\alpha} = {}^{\mathrm{d}\alpha}/_{\mathrm{d}t}$. The explicit form of

log-derivative operators $\mathcal{O}_\mu$ introduced in the main text is

$$\mathcal{O}_\mu(\boldsymbol{\theta}, t) = \frac{\partial}{\partial \alpha_\mu} \ln \psi_{\alpha(t)}(\boldsymbol{\theta}) \tag{31}$$

in the $|\boldsymbol{\theta}\rangle$ basis. We note that the expression in Eq. 29 and the definitions of S and g in Eq. 30 change form if $\psi_\alpha(\boldsymbol{\theta})$ cannot be interpreted as a holomorphic function of α. The reader is referred to Ref. (Yuan et al., 2019) for detailed derivations.

We implement and experiment with a number of different Runge-Kutta (Press et al., 1992) (RK) ODE solvers. Heuristically, we notice that higher-order adaptive embedded solvers do help offset the effects of imperfect Monte Carlo estimates of S and g in two ways:

- Adaptive solvers are naturally higher-order because of an embedded lower-order method. Using an adaptive solver can locally adjust the time step δt, usually drastically reducing the overall number of time steps required.
- The solution at $t + \delta t$ is constructed as a linear combination of solutions estimated on a fixed set of points within the interval $[t, t + \delta t]$. Any leftover errors in these intermediate estimates have a higher probability of canceling out.

In this work, we choose the adaptive third-order method with an embedded second order method using the Bogacki-Shampine (Bogacki & Shampine, 1989; Butcher, 1963; 2008; Press et al., 1992) pair of formulas. It balances being low-enough order to avoid wasting computational resources with still being high-enough order to allow for adaptive time-stepping.

For quenches shown in Fig. 2, we show variations in δt in Fig. 6 (center panel). We note that the time stepping varies more in regions of higher curvature (kinetic energy, shorter times) and for larger quenches, successfully adjusting to conserve energy in all cases (Fig. 6 right). For longer times, on the order of thermalization in observed quenches, variations are reduced and δt approximately converges to a constant value.

B. Angular momentum basis and tensor network calculations

In this appendix we focus on some of the conventions and formalism involved with treating the QRM in the discrete eigenbasis of the angular momentum operator L_k. This basis is useful for attacking the model with tensor-network methods or perturbation theory.

In the $|\boldsymbol{\theta}\rangle$ basis, we have $L_k = -i\,\partial_k$ (where we adopt the convention $\partial_k \equiv \partial_{\theta_k}$). Therefore:

$$-\,i\frac{\partial}{\partial\theta}\langle\theta|m\rangle = m\,\langle\theta|m\rangle \qquad \Longrightarrow \qquad \langle\theta|m\rangle = \frac{e^{-im\theta}}{\sqrt{2\pi}}\ ; \quad m \in \mathbb{Z}\ , \tag{32}$$

which is identical to eigenfunctions for a particle on a circle at each lattice site – we have a product basis basis $|\mathbf{m}\rangle = |m_1, \ldots, m_N\rangle$. The hamiltonian given in Eq. 1 then reads:

$$\langle\mathbf{m}'|\,H\,|\mathbf{m}\rangle = \frac{gJ}{2}\sum_k m_k^2 - J\sum_{\langle k,l\rangle}\langle\mathbf{m}'|\,\hat{\mathbf{n}}_k\cdot\hat{\mathbf{n}}_l\,|\mathbf{m}\rangle\ . \tag{33}$$

After inserting the identity $\mathbb{1} = \int \mathrm{d}\boldsymbol{\theta}\,|\boldsymbol{\theta}\rangle\langle\boldsymbol{\theta}|$ into the second term and simple integration, we obtain

$$\langle m_k', m_l'|\,\hat{\mathbf{n}}_k\cdot\hat{\mathbf{n}}_l\,|m_k, m_l\rangle = \frac{1}{2}\left(\delta_{m_k',m_k+1}\delta_{m_l',m_l-1} + \delta_{m_k',m_k-1}\delta_{m_l',m_l+1}\right)\ . \tag{34}$$

where $\delta_{..}$ is the Kronecker delta symbol. The structure of Eq. 34 suggests rewriting the original Hamiltonian as:

$$H = \frac{gJ}{2}\sum_k L_k^2 - \frac{J}{2}\sum_{\langle k,l\rangle}\left(L_k^+L_l^- + L_l^+L_k^-\right) \tag{35}$$

where

$$L_k^+ \equiv \sum_{m_k \in \mathbb{Z}} |m_k + 1\rangle\langle m_k| \qquad \text{and} \qquad L_k^- \equiv \sum_{m_k \in \mathbb{Z}} |m_k - 1\rangle\langle m_k| \qquad \text{so that} \qquad \left[L_k^+, L_k^-\right] = 0\,, \tag{36}$$

for all $k \in \Lambda$. To perform tensor-network calculations, we truncate local basis states to $\{|-M\rangle, \ldots, |M\rangle\}$ so that $L^+ |M\rangle = L^- |-M\rangle = 0$. Namely, we set:

$$L^+ \to \begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{pmatrix}, \qquad L^- \to \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ & 1 & & & \\ & & \ddots & 0 & \\ & & & 1 & 0 \end{pmatrix} \tag{37}$$

to build a Matrix Product Operator (MPO) representation of the Hamiltonian. For use in real-time evolution through the TEBD algorithm in Sec. 3.1 in the main state, we initialize the trial wavefunction as a Matrix Product State (MPS) (White, 1992) (see Appendix C.2). We exploit

$$|\psi(t=0)\rangle \propto \int \mathrm{d}\boldsymbol{\theta}\, |\boldsymbol{\theta}\rangle = \prod_{i \in \Lambda} |m_i = 0\rangle \tag{38}$$

to match the initial state given in the main text in Sec. 3.1 by initializing the corresponding MPS to have bond dimension $\chi = 1$.

C. Variational Quantum States on a circle

In order to perform numerically efficient t-VMC iterations described in Sec. 2.1, we need to keep the parameter count in ψ_α relatively low (order 10^2-10^4 on modern GPU hardware.). The reason for this constraint is the regularization scheme we employ to stabilize the QGT inverse in Eq. 6 – each iteration requires us to diagonalize the $P \times P$ hermitian matrix S which becomes prohibitively expensive and memory-consuming for large parameter counts P. We note that iterative solvers such as the *conjugate gradient* can formally help push the limit of tractable P by several orders of magnitude. However, that speedup comes at the cost of having to rely on weaker regularization schemes that do not require the full QGT spectrum. In this appendix, we describe different ansatzes (trial wavefunctions) considered in our simulations.

C.1. The Jastrow wavefunction

The Jastrow wavefunction is defined as:

$$\ln \psi_\alpha(\boldsymbol{\theta}) = \frac{1}{2} \sum_{ij} w_{ij}\, \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j = \frac{1}{2} \sum_{ij} w_{ij} \cos(\theta_i - \theta_j) \tag{39}$$

where $\alpha = \{w_{ij}\}$ is a symmetric matrix of variational parameters. The advantage of a simple Jastrow ansatz is the fact that the QGT given in Eq. 6 is never ill-conditioned. However, Jastrow expressivity is limited compared to deeper neural-network quantum state parametrizations.

C.2. Matrix Product States

Using notation and conventions laid out in Appendix B, one can write down a traditional MPS ansatz in the discrete angular momentum basis $|\mathbf{m}\rangle$:

$$\psi_\alpha(\theta) = \sum_{m_1=-M}^{M} \cdots \sum_{m_N=-M}^{M} c_{m_1 \cdots m_N}\, e^{-i \sum_i m_i \theta_i}\,; \quad c_{m_1 \cdots m_N} = \sum_{l_1=1}^{\chi_1} \cdots \sum_{l_N=1}^{\chi_N} A_{l_1}^{m_1} A_{l_1 l_2}^{m_2} \cdots A_{l_{N-2} l_{N-1}}^{m_{N-1}} A_{l_{N-1}}^{m_N}\,. \tag{40}$$

In one spatial dimension and for ground state searches (imaginary time evolution), this trial wavefunction form is the most accurate due to the Density Matrix Renormalization Group algorithm (DMRG) (White, 1992) algorithm. Expressivity is controlled by *bond dimensions* χ_i and the basis truncation parameter M.

C.3. The (circular) Restricted Boltzmann Machine

The circular Restricted Boltzman Machine (RBM) (Stokes et al., 2021) is defined as:

$$\psi_\alpha(\boldsymbol{\theta}) \propto \int \mathrm{d}\mu(\hat{\mathbf{h}})\ \exp\left\{\sum_j \mathbf{a}_j \cdot \hat{\mathbf{n}}_j + \sum_k \mathbf{b}_k \cdot \hat{\mathbf{h}}_k + \sum_{jk} w_{jk}\, \hat{\mathbf{n}}_j \cdot \hat{\mathbf{h}}_k\right\} \tag{41}$$

where $\alpha = \{\mathbf{a}_j, \mathbf{b}_k, w_{jk}\}$ are variational parameters and $\mathrm{d}\mu(\hat{\mathbf{h}})$ is the relevant measure for *hidden units* $\hat{\mathbf{h}}_k$. It is natural to choose hidden units to have the same intrinsic Hilbert space as visible rotors $\hat{\mathbf{n}}_j$. Therefore, for the $O(2)$ Quantum Rotor Model, we choose

$$\hat{\mathbf{h}}_k = (\cos\phi_k, \sin\phi_k) \qquad \text{so that} \qquad \mathrm{d}\mu(\hat{\mathbf{h}}) = \mathrm{d}\boldsymbol{\phi} = \mathrm{d}\phi_i\, \mathrm{d}\phi_2 \cdots \mathrm{d}\phi_{N_\mathrm{h}}\ , \tag{42}$$

up to an overall multiplicative constant. We note that the number hidden units N_h is a hyperparameter and can be increased to control ansatz expressivity.

After performing the integrals in Eq. 41, one obtains the following closed-form expression:

$$\ln\psi_\alpha(\boldsymbol{\theta}) = \sum_{j=1}^{N} \mathbf{a}_j \cdot \hat{\mathbf{n}}_j + \sum_{k=1}^{N_\mathrm{h}} \ln I_0\left(\sqrt{\sum\nolimits_l (\mathbf{x}_k)_l^2}\right) \tag{43}$$

where $(\mathbf{x}_k)_l$ stands for the l-th component of vector $\mathbf{x}_k = \mathbf{b}_k + \sum_j w_{jk}\hat{\mathbf{n}}_j$ and I_0 is the zeroth-order modified Bessel function of the first kind.

Instead of a full dense matrix, we can restrict the general linear map $\hat{\mathbf{n}}_j \mapsto \sum_j w_{jk}\hat{\mathbf{n}}_j$ to a convolution, assuming that underlying rotors are arranged in a square lattice. This restriction cuts P down by approximately an order of magnitude while not sacrificing any measurable accuracy in ground state optimization tasks.

C.4. Activation functions

In order to define an analytic ansatz ψ_α with no hidden sigularities, care must be taken when choosing activation functions for complex-valued inputs. Informally, singularities often appear in one of the following two ways, when using holomorphic activations:

- A well-behaved function (or its derivatives) on the real axis has singularities on the imaginary axis. This is the case for $\tanh$ and $\ln I_0$ from Eq. 41, for example.
- An otherwise well-behaved function has a branch cut that is crossed during time evolution. Side-effects include sudden jumps in conserved quantities during real-time evolution. This is the case for $\ln I_0$ from Eq. 41 and similar functions involving logarithms and/or roots.

There are two solutions to this problem. As noted in the main text, one can restrict themselves to (higher-order) polynomial activations which are analytic everywhere and have no branch cuts. Inspired by Eq. 41, we use Taylor expansions of $\ln I_0$ and its gradient:

$$\ln I_0(z) = \frac{z^2}{4} - \frac{z^4}{64} + \frac{z^6}{576} + \mathcal{O}(z^8) \quad \text{and} \quad \frac{I_1(z)}{I_0(z)} = \frac{z}{2} - \frac{z^3}{16} + \frac{z^5}{96} + \mathcal{O}(z^7)\ . \tag{44}$$

This approach has the advantage of maintaining the holomorphic dependence of ψ_α on α and preserving the form of Eq. 6. We note that if the effect of isolated singularities is not as important, Padé approximants often provide better approximations of target functions (and better ground state energies) while still eliminating branch cuts.

The second option is abandoning holomorphicity in parameters α and applying well-behaved real activations to real and imaginary parts of the input separately. In that case, Eq. 6 must be corrected. We refer interested readers to the excellent overview of subtleties associated with complex parameters in Ref. (Yuan et al., 2019).

D. The R^2 performance metric

In the main text, we define a measure of time-dependent integration error in Eq 19, following Refs. (Schmitt & Heyl, 2020; Carleo & Troyer, 2017). The Fubini-Study distance $\mathcal{D}(\cdot,\cdot)$ on the Hilbert space $\mathcal{H}$ defined as:

$$\mathcal{D}(\psi,\phi) = \cos^{-1}\left(\sqrt{F(\psi,\phi)}\right) = \cos^{-1}\left(\sqrt{\frac{|\langle\psi|\phi\rangle|^2}{\langle\psi|\psi\rangle\,\langle\phi|\phi\rangle}}\right) . \tag{45}$$

Using a consistent Taylor expansion in the limit of $\delta t \ll J^{-1}$,

$$e^{-iH\delta t} = \mathbb{1} - iH\delta t + \mathcal{O}(\delta t^2) \tag{46}$$

$$|\psi(t+\delta t)\rangle = \left(1 + \delta t \sum_{\mu} \dot{\alpha}_\mu \mathcal{O}_\mu\right) \left|\psi_{\alpha(t)}\right\rangle + \mathcal{O}(\delta t^2) , \tag{47}$$

authors in Ref. (Schmitt & Heyl, 2020) rewrite Eq. 19 as:

$$r^2(t) = 1 - \frac{1}{\mathrm{Var}_t H} \sum_{\mu\nu} S^{-1}_{\mu\nu} g^*_\mu g_\nu + \mathcal{O}(\delta t^2) . \tag{48}$$

In Eq. 48, we used notation from Eq. 6 in the main text with

$$\mathrm{Var}_t H = \left\langle H^2 \right\rangle_t - \langle H\rangle_t^2 \approx \left\langle |E_L - \langle E_L\rangle|^2 \right\rangle_t \tag{49}$$

Finally, the R^2 figure of merit is constructed as a time-integral of r^2: $R^2(t) = \int_0^t r^2(s)\mathrm{d}s$. We note that $r^2(t)$ and $R^2(t)$ are readily available for estimation through Monte Carlo sampling.