Coarse-grained spectral projection: A deep learning assisted approach to quantum unitary dynamics

Pinchen Xie
Program in Applied and Computational Mathematics, Princeton University, New Jersey 08544, USA

Weinan E
Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University, New Jersey 08544, USA

(Received 8 November 2020; revised 16 November 2020; accepted 13 January 2021; published 28 January 2021)

We propose the coarse-grained spectral projection method (CGSP), a deep learning assisted approach for tackling quantum unitary dynamic problems with an emphasis on quench dynamics. We show that CGSP can extract spectral components of many-body quantum states systematically with a sophisticated neural network quantum ansatz. CGSP fully exploits the linear unitary nature of the quantum dynamics and is potentially superior to other quantum Monte Carlo methods for ergodic dynamics. Preliminary numerical results on one-dimensional *XXZ* models with periodic boundary conditions are carried out to demonstrate the practicality of CGSP.

DOI: 10.1103/PhysRevB.103.024304

I. INTRODUCTION

The past several decades have witnessed the rapid growth of research interest in dynamical quantum many-body systems [1–5], leading to the observation of novel quantum phenomena [6–10] outside the scope of equilibrium physics. The possibility of realistic implementations of quantum computations has also been introduced [11–15].

At the same time, there has also been a great deal of activity on scalable algorithms for numerical simulations of quantum dynamics [16–26]. The main challenge in this pursuit is modeling highly entangled high-dimensional quantum states present during evolution, a task that usually requires exponential complexity in classical computing. Examples that fall in this category include most tensor network ansatzes, such as matrix product states (MPSs), projected entangled pair states, and multiscale entanglement renormalization ansatzes [27–29], originating from the density matrix renormalization group method [30]. As a consequence, the application of these ansatzes is usually limited to one-dimensional (1D) or two-dimensional (2D) systems featuring area-law entanglement with or without logarithmic correction [31,32]. Hence more versatile ansatzes are desired in the face of quantum dynamics.

In recent years, the most promising candidate turned out to be artificial neural networks, which are believed to have a huge entanglement capacity [33]. An early practice along this line of research was the application of the restricted Boltzmann machine in solving the ground state and the dynamics of quantum spin models [34]. Later, symmetry-preserving deep fully connected neural networks and convolutional neural networks were also shown to be efficient quantum state ansatzes [35–39]. In particular, the convolutional neural network is believed to support volume-law entanglement scaling while being polynomially more efficient in resources compared to the restricted Boltzmann machine–like ansatz in two dimensions, due to an inherent reuse of information [40].

So far, the main algorithm accompanying black-box-like neural networks for simulating quantum dynamics is the time-dependent variational principle (TDVP) method [34]. In plain words, TDVP projects a real-time quantum evolution trajectory into a tiny useful subset, parameterized by a neural network, of the Hilbert space. The projected dynamics is then described by a low-dimensional time-dependent differential equation of neural network parameters. In spite of the numerical instability and the limited expressive power of the neural networks, TDVP methods are potentially capable of simulating quench dynamics of very large quantum spin systems with strong entanglement [41], ultrafast dynamics [42], and evolution of open quantum systems as well as stationary states [43,44]. However, TDVP methods do not give special treatment to dynamics driven by a static Hamiltonian where the quantum evolution has a certain spectral structure and multiple intrinsic time scales. The ignorance of both may lead to prohibitive numerical instability in integrating the TDVP-induced low-dimensional dynamics step by step.

In this work, we show how to poke into the spectral structure of unitary dynamics and extract limited but useful high-dimensional information directly from the initial condition. This is done through a coarse-grained representation of the spectral projection with deep learning, a procedure we have dubbed coarse-grained spectral projection (CGSP). The results of CGSP can be used to simulate a unitary dynamics driven by a static Hamiltonian directly without step-by-step integration.

II. COARSE-GRAINED SPECTRAL PROJECTION

Considering a pure quantum state $|\Psi_o\rangle$ (in the following the brackets for a ket are dropped in the absence of an inner product) in a closed system as the initial condition of a unitary evolution driven by the static Hamiltonian H, a complete

eigendecomposition of Ψ_o can be expressed as

$$\Psi_o = \sum_{i=1}^{N_h} b_i \psi_i, \tag{1}$$

where $\{b_i\}_{i\in[1,N_h]}$ are real constants. The eigenstates $\{\psi_i\}_{i\in[1,N_h]}$ satisfy $H\psi_i = E_i\psi_i$. They are orthonormal and increasingly ordered with respect to the energy level E_i. N_h is the dimension of the Hilbert space $\mathcal{H}$. There exists trivial disjoint cover of the entire energy spectrum on the real axis,

$$\left[E_1, E_{N_h}\right] \subset [x_0, x_1) \cup [x_1, x_2) \cup \ldots \cup [x_{N-1}, x_N], \tag{2}$$

such that $\{x_i\}_{i\in[1,N]}$ is an arithmetic sequence satisfying $x_0 < E_1 < E_{N_h} < x_N$. Then for each interval $[x_i, x_{i+1}]$ we associate the direct sum of the eigensubspaces whose eigenvalues lie in $[x_i, x_{i+1}]$. We obtain N subspace $\{\mathcal{Q}_i \in \mathcal{H}\}_{i\in[0,N-1]}$ and N corresponding projection operators $\{\mathcal{P}_i\}_{i\in[0,N-1]}$. Since $\mathcal{H} = \bigoplus_i \mathcal{Q}_i$, it is obvious that $\Psi_o \in \mathrm{span}(\mathcal{P}_0\Psi_o, \ldots, \mathcal{P}_{N-1}\Psi_o)$. Let θ_i denote the normalized $\mathcal{P}_i\Psi_o$. Ψ_o can be expressed as $\Psi_o = \sum_{i=0}^{N-1} c_i\theta_i$, where c_i are real constants.

Let $\epsilon = x_{i+1} - x_i$ and $\lambda_i = (x_i + x_{i-1})/2$ be the center of the ith interval. The unitary evolution of $\Psi_o(t) = e^{-iHt}\Psi_o$ driven by time-independent Hamiltonian H can be approximated by

$$\varphi_o(t) = \sum_{i=0}^{N-1} c_i e^{-i\lambda_i t}\theta_i. \tag{3}$$

The error has an evident upper bound given as $\|\Psi_o(t) - \varphi_o(t)\| \leqslant \frac{\epsilon t}{2}$. To achieve $\|\Psi_o(t) - \varphi_o(t)\| < \delta$ for small enough δ, N should satisfy

$$N > \frac{\left(E_{N_h} - E_1\right)t}{2\delta}. \tag{4}$$

Note that the energy spectrum range $(E_{N_h} - E_1)$ usually increases linearly with system size for quantum models defined on (nearly) regular graphs with bounded short-range interaction and disorder. Recently, a similar bound has also been derived in the context of quantum state compression via principal component analysis [45].

It is both unnecessary and difficult, if not impossible, to solve the N projected state θ_i exactly for simulating dynamics. Compromises should be made for practical reasons. Especially, the uniqueness of θ_i should be loosened by allowing polluted projection, which implies a nonorthogonal decomposition of $\Psi_o(0)$. Assuming N normalized states Θ_i parameterized by N classical ansatzes, respectively, a nonorthogonal decomposition of $\Psi_o(0)$ can be achieved through the objective function as a constrained optimization problem,

$$\begin{aligned} &\underset{\{c_i\},\{\Theta_i\}}{\text{minimize}} \quad \sum_{i=0}^{N-1} c_i^2 \langle\Theta_i|(H-\Lambda_i)^2|\Theta_i\rangle, \\ &\text{subject to} \quad d\left(\Psi_o(0), \sum c_i\Theta_i\right) = 0. \end{aligned} \tag{5}$$

d can be any legal distance function in the Hilbert space including the Fubini-Study metric and L_2 norm, regardless of U(1) symmetry. The fixed constant $\{\Lambda_i\}$ in the objective function is an arithmetic sequence satisfying $\Lambda_0 \leqslant E_{\min}$ and $\Lambda_{N-1} \geqslant E_{\max}$, with a common increment of $\epsilon > 0$. The ground-state energy $E_{\min}$ of the system Hamiltonian and the maximum energy $E_{\max}$ can be easily estimated using the usual variational Monte Carlo (VMC) techniques with the same classical ansatz.

With a slight abuse of notation, let $\lambda_i = \frac{\langle\Theta_i|H|\Theta_i\rangle}{\langle\Theta_i|\Theta_i\rangle}$. $\Psi_o(t)$ can be approximated by $\varphi_o(t) = \sum_{i=0}^{N-1} c_i e^{-i\lambda_i t}\Theta_i$ up to a constant phase difference ϕ_{ph}. Without loss of generality, we assume $\phi_{ph} = 0$ for all that follows. The quality of this approximation is reflected by the "covariance" matrix

$$(K)_{ij} = \langle\Theta_i|(H-\lambda_i)(H-\lambda_j)|\Theta_j\rangle. \tag{6}$$

For small t, the error of the approximation is

$$\|\Psi_o(t) - \varphi_o(t)\|^2 \approx t^2\mathbf{c}^\dagger K\mathbf{c}. \tag{7}$$

Consequently, $|K_{ij}|/\epsilon^2 \gg 1$ is allowed for a successful decomposition as long as $|c_ic_j|$ is small enough. Instead, the weighted "covariance" $|c_iK_{ij}c_j|(i \neq j)$ is always expected to be much smaller than ϵ^2. In practice, it is easier to calculate only the diagonal element of K, i.e., the variance $\sigma_i^2 = \langle\Theta_i|(H-\lambda_i)^2|\Theta_i\rangle$ of each Θ_i. Let $|\sigma|^2 = \sum_i c_i^2\sigma_i^2 / \sum_i c_i^2$. We have a very rough but inexpensive estimation of the error:

$$\|\Psi_o(t) - \varphi_o(t)\|^2 \approx |\sigma|^2 t^2. \tag{8}$$

Notably, Eq. (5) minimizes a weighted sum of individual variances. We show in Appendix A that this particular choice of objective function leads to an N^{-1} scaling of each σ_i in the ideal case.

When $N \to \infty$, Eq. (5) converges to its continuous form:

$$\begin{aligned} &\underset{c,\Theta}{\text{minimize}} \quad \int_{w_a}^{w_b} c^2(w)\,\langle\Theta(w)|(H-w)^2|\Theta(w)\rangle\,dw, \\ &\text{subject to} \quad d\left(\Psi_o(0), \int_{w_a}^{w_b} c(w)\Theta(w)dw\right) = 0. \end{aligned} \tag{9}$$

For many disordered systems, eigenstates with very close energy levels can have completely different local observables [46]. Hence a global minimizer $\Theta_m(w)$ of Eq. (9) is not expected to be continuous with respect to w. Therefore we adopt the discrete form, Eq. (5), as the starting point for extracting spectral information and call it "coarse-grained spectral projection."

III. NUMERICAL FRAMEWORK AND RESULTS

To show that CGSP is applicable to real quantum dynamic problems, we propose a feasible numerical framework for using Eq. (5) in the quench dynamics of quantum lattice models. When deep neural networks serve as ansatzes, it is desirable to convert Eq. (5) into an unconstrained loss function for practical training. A naive treatment is to handle the constraint in Eq. (5) with the penalty method. We found this approach very problematic because a noisy estimation of the emphasized penalty will greatly slow down the minimization of the original objective function. A more considerate approach is to construct the ansatzes in a way such that the constraint is automatically satisfied.

Suppose the initial state $\Psi_o(0) \in \mathcal{H}$ can be parameterized exactly by the classical ansatz Υ_0 with fixed parameters. In addition, we have M classical ansatzes $\{\Upsilon_j\}_{j\in[1,M]}$

$\subset \mathcal{H}$ with free parameters. Let $A = (A_{ij})_{i\in[0,N-1],j\in[0,M]}$ be a real matrix. Then $c_i\Theta_i$ as a whole is constructed to satisfy $d(\Psi_o(0), \sum c_i\Theta_i) = 0$, given as

$$c_i\Theta_i = \sum_{j=0}^{M}\left(\frac{\delta_{j0}}{N} + A_{ij} - \frac{\sum_{i=0}^{N-1} A_{ij}}{N}\right)\Upsilon_j. \tag{10}$$

In principle, M should be larger than N to ensure the linear independence of $\{\Theta_i\}_{i\in[0,N-1]}$. Also, a larger M will provide stronger variational freedom for $\{\Theta_i\}_{i\in[0,N-1]}$. But in practice M is flexible because some c_i vanishes. Next, we define a new objective function without explicit constraint:

$$L = \left(\frac{2(N-1)}{\Lambda_{N-1} - \Lambda_0}\right)^2 \sum_{i=0}^{N-1} c_i^2 \langle\Theta_i|(H - \Lambda_i)^2|\Theta_i\rangle\,. \tag{11}$$

Equation (11) is used for training $\{\Upsilon_j\}_{j\in[1,M]}$ and A. The constant in front of the summation in Eq. (11) ensures that the minimum of L is of the order of $O(1)$ rather than $O(N^{-2})$.

Due to the exponentially large Hilbert space, L should be estimated with Monte Carlo methods. Compared to traditional sequential Monte Carlo sampling methods such as the Markov chain Monte Carlo (MCMC), we find that recently developed neural autoregressive quantum states (NAQSs) [47] can achieve a higher efficiency and better sampling quality at the same time, if employed on graphics processing units (GPUs). Moreover, the NAQS allows exact normalization. Therefore, we develop a CGSP-adapted NAQS that supports parallel evaluation of $\{\Upsilon_j\}_{j\in[1,M]}$ and also parallel sampling for practical CGSP application. Because the CGSP-adapted NAQS is not an essential part of the CGSP method and can be substituted by any proper classical ansatz, we report the detailed information of its construction in Appendix B.

In the following, with CGSP-adapted NAQSs as classical ansatzes and the direct sampling algorithm associated with the NAQS for Monte Carlo sampling, we demonstrate the practicality of CGSP by simulating the unitary quench dynamics of a 1D spin-1/2 *XXZ* model. The Hamiltonian is given by

$$H(J,\Delta,h) = \sum_{k=1}^{l} J\left(S_k^x S_{k+1}^x + S_k^y S_{k+1}^y + \Delta S_k^z S_{k+1}^z\right) + hS_k^z. \tag{12}$$

We assume periodic boundary conditions and work strictly within the zero total S^z sector so h is irrelevant. For the numerical results presented here, the *XXZ* chain contains $l = 32$ spins suddenly quenched from $\Delta \to -\infty$ to $\Delta = -1$ with initial condition $\Psi_0(0) = |\downarrow\downarrow\ldots\downarrow\uparrow\uparrow\ldots\uparrow\rangle$. We compare our results to the converged TDVP calculation with the MPS (TDVP-MPS). In Fig. 1(a), we plot the σ^z local magnetization of several representative spins computed by CGSP with $(M,N) = (32,32)$ and $(M,N) = (32,64)$, respectively. In Fig. 1(b), we show the amplitude of nonvanishing projected states. It is evident in Fig. 1(b) that the initial state $\Psi_0(0)$ contains mainly low-lying eigenmodes of $H(J,-1,h)$. This justifies using CGSP with $M < N$ for $\Psi_0(0)$. Based on this observation, $M = 32$ should be enough for CGSP with $N = 32$ and $N = 64$. In Fig. 1(a), we find that with $N = 64$ CGSP can simulate longer dynamics than with $N = 32$. If we increase

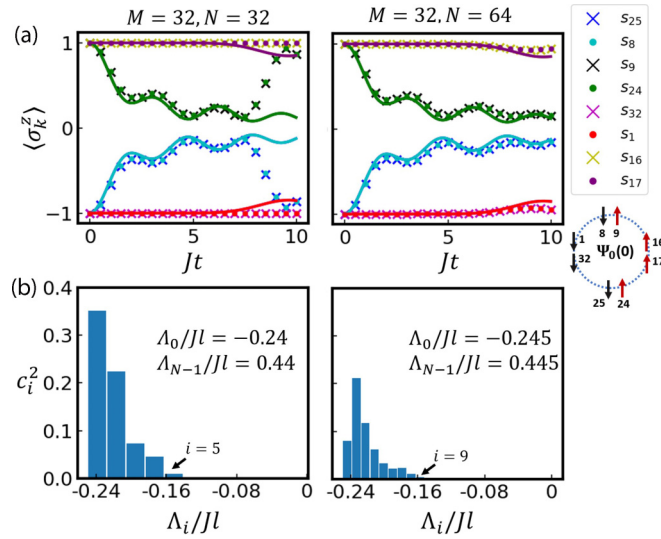

FIG. 1. (a) Dynamics of local magnetization $\langle\sigma_k^z(t)\rangle$ for spins adjacent to the domain wall and spins in the middle of the ferromagnetic domain (the scheme below the legend specifies the labeling). The scattered data are calculated by CGSP with $(M,N) = (32,32)$ (left column) and $(M,N) = (32,64)$ (right column). Solid lines are results from the converged TDVP-MPS. The color of the solid line matches the color of scattered data for the same spin. Due to the mirror symmetry of the initial state, the solid lines associated with $k = 9, 16, 25, 32$ are covered by the others. (b) The amplitude c_i^2 of projected states for $(M,N) = (32,32)$ (left column) and $(M,N) = (32,64)$ (right column). Because c_i^2 vanishes for all $\Lambda_i > 0$, we plot only the lower section of the energy spectrum.

N further, CGSP should be more accurate until the expressive power limited by $M = 32$ becomes the main bottleneck.

In Fig. 2, we show the dynamics of $\langle\sigma_k^z(t)\rangle$ for all the spins [Fig. 2(a)], compared to the TDVP-MPS benchmarks [Fig. 2(b)]. For CGSP simulation with $(M,N) = (32,64)$, the evolution of the local magnetization shows the light-cone structure predicted by the Lieb-Robinson bounds. In Figs. 2(c) and 2(d), we plot the correlation function, calculated by CGSP and TDVP-MPS, between pairs of spins the same distance from the domain wall but on opposite sides according to the initial state. A long-range correlation emerges during evolution.

Based on Eq. (8), the numerical coherence time T_c with respect to $\|\Psi_o(T_c) - \varphi_o(T_c)\|^2 \approx 0.5$ can be estimated from the training results for predicting the valid region of simulated dynamics without benchmarking. We obtain $JT_c \approx 3.5$ for $(M,N) = (32,32)$ and $JT_c \approx 5.6$ for $(M,N) = (32,64)$. Observing Figs. 1 and 2, we see that T_c may slightly underestimate the region of validity of the simulated dynamics.

For a successful CGSP like the ones shown here, we find that the weighted "covariance" $|c_i^* K_{ij} c_j|/\epsilon^2 (i \neq j)$ should be at most of the order of 10^{-3} to maintain approximate orthogonality among projected states. Meanwhile, $|c_i|^2 K_{ii}/\epsilon^2$ is expected to be at most of the order of 10^{-2} for all i.

Technical details of numerical experiments can be found in Appendix C. Besides, in Appendix D, we propose a simple parallel framework for breaking down CGSP into hierarchically organized subtasks. Note that the numerical experiments

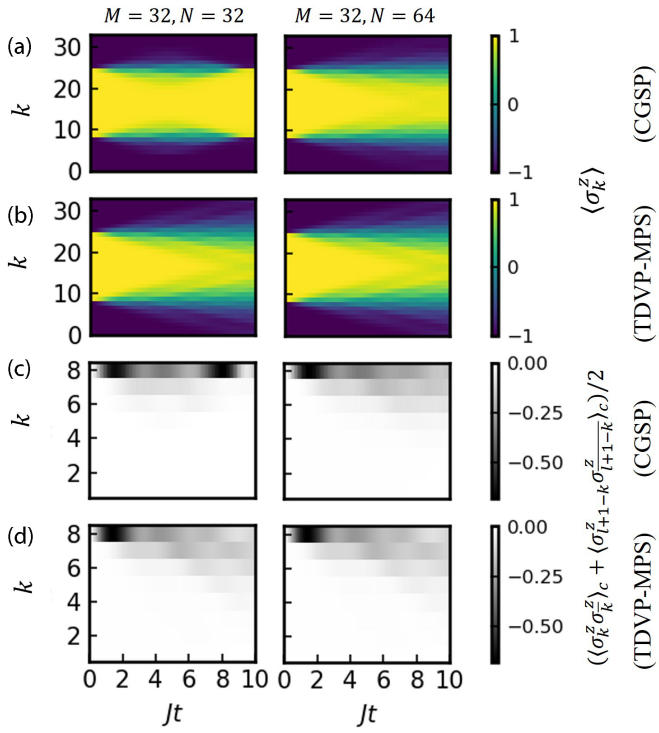

FIG. 2. (a) Dynamics of local magnetization $\langle\sigma_k^z(t)\rangle$ after a sudden quench calculated by CGSP for the 32-spin XXZ model. (b) Exact dynamics of $\langle\sigma_k^z(t)\rangle$ obtained with the TDVP-MPS. The right column is a duplicate of the left column for easier comparison. (c) The correlation function $\langle\sigma_k^z\sigma_{\bar{k}}^z\rangle_c$ between spin k and its counterpart spin $\bar{k}$ ($\bar{k} = \frac{l}{2} + 1 - k$), averaged with $\langle\sigma_{l+1-k}^z\sigma_{\overline{l+1-k}}^z\rangle_c$. (d) Exact dynamics of $(\langle\sigma_k^z\sigma_{\bar{k}}^z\rangle_c + \langle\sigma_{l+1-k}^z\sigma_{\overline{l+1-k}}^z\rangle_c)/2$ obtained with the TDVP-MPS. The right column is a duplicate of the left column.

presented here do not utilize this parallel framework. The codes for implementation of CGSP-adapted NAQSs and the numerical experiments are available in [48].

IV. DISCUSSION

Our numerical experiments mainly showcase the practicality of CGSP without further analyzing its scalability and other issues such as entanglement, symmetry, nonlocality, and thermalization. Nor do we show how different types of classical ansatzes can be fitted into the framework of CGSP. But this does not prevent us from estimating the complexity of CGSP in terms of ansatz complexity and quantum system specifications. Suppose that the spectrum range of an l-spin Hamiltonian with only short-range interaction is E and the time scale we want to simulate is represented by T. The number of necessary projected states obeys $O(ET)$. The number of stochastic samples needed to control the noise level is $O(T^4)$. So the computational cost of estimating the loss function (including the sampling process) is $O(lT^4) \times C(l, ET)$, where $C(l, ET)$ denotes the computational complexity of one forward propagation of the classical ansatz in terms of l and ET. For optimization methods based on first-order gradient descent, the total number of iterations required for convergence is unknown.

Unfortunately, even though we guess $C(l, ET)$ to be polynomial for some specific tasks, their is no conclusive complexity theory yet to predict $C(l, ET)$ or the neural network complexity in other numerical algorithms. This renders the comparison between deep learning algorithms rather difficult, especially when there is no general-purpose neural network structure for different kinds of quantum problems. But we are still able to qualitatively compare CGSP with the TDVP. First, one realizes that a classical ansatz like a neural network can only represent a low-dimensional section of a high-dimensional Hilbert space regardless of its entanglement capacity. There should not exist a generic polynomial algorithm for simulating quantum dynamics in the long term as long as a classical ansatz is used to represent the whole or the decomposition of the evolving quantum state. So both the TDVP and CGSP will lose polynomial complexity for generic problems, but in different scenarios. For TDVP methods, polynomial complexity is not possible when the actual quantum state trajectory travels away from the low-dimensional section parameterized by the neural network. This failure is inevitable for ergodic dynamics harnessed by few symmetries and may be more easily detected in quench dynamics over criticality [49]. For CGSP, the neural network is expected to parametrize only $O(T)$ quantum many-body states rather than a differentiable subset containing the real-time evolution trajectory. This statement holds true, regardless of ergodicity, for finite-time dynamics driven by a time-independent Hamiltonian. However, when T is large or the target quantum state is featureless, even a countable finite subset of the Hilbert space is too difficult for neural networks to represent fully. This is when CGSP also encounters exponential complexity.

Based on the discussion above, CGSP seems to be less demanding on the expressive power of classical ansatzes. However, from the optimization perspective, CGSP requires more training efforts compared to TDVP methods, which propagate in a deterministic way when Monte Carlo sampling is nearly exact. Because the optimization of a CGSP task is nonconvex towards the objective Eq. (5), CGSP may suffer from the local optimum and ill conditioning like almost every deep learning task. Since gradient-based optimization methods are almost the only practical choice for deep neural networks, these issues could be the major obstruction to the scalability of CGSP.

V. OUTLOOK

So far, we find CGSP to be potentially a good candidate for studying the unitary dynamics of quantum systems, for it not only provides access to almost all observables but also unfolds the spectral structure of a unitary evolution. More meaningful physics are encoded in the CGSP results than conventional VMC simulations. Being fundamentally different from previous methods utilizing the TDVP or Krylov subspace, CGSP is expected to solve specific problems that have been inaccessible in the past. In Appendix E, we also discuss the possibility that CGSP can improve TDVP simulations driven by a slow-varying time-dependent Hamiltonian.

Future development of CGSP may focus on more efficient utilization of the spectral structure of an initial state or the design of a more sophisticated loss function for enhancing

the orthogonality between projected states. Moreover, a lot of effort should be devoted to further developing a neural network ansatz that can model quantum states in different scenarios, for example, states near thermalization.

ACKNOWLEDGMENTS

This work was supported in part by a gift to Princeton University from iFlytek.

APPENDIX A: IDEAL MINIMIZER OF THE OBJECTIVE FUNCTION

We derive the global minimizer of Eq. (5) in the ideal case where the classical ansatz $\{\Theta_i\}_{i\in[0,N-1]}$ can represent any quantum state faithfully. We use the same notation $\{\psi_i\}_{i\in[1,N_h]}$ to denote an increasingly ordered orthonormal eigenbasis associated with Hamiltonian H as in Eq. (1).

For each Θ_i, its unique eigendecomposition can be expressed as

$$\Theta_i = \sum_{k=1}^{N_h} a_k^{(i)} \psi_k. \tag{A1}$$

In the same way, the initial condition can be decomposed into

$$\Psi_o(0) = \sum_{k=1}^{N_h} b_k \psi_k. \tag{A2}$$

Using L_2 norm as the distance measure, the original optimization problem, Eq. (5), can be written as

$$\begin{aligned} &\underset{\{c_i\},\{a_k^{(i)}\}}{\text{minimize}} \quad \sum_{i=0}^{N-1}\sum_{k=1}^{N_h} \left|c_i a_k^{(i)}\right|^2 (E_k-\Lambda_i)^2, \\ &\text{subject to} \quad \sum_{k=1}^{N_h} \left| b_k - \sum_{i=0}^{N-1} c_i a_k^{(i)} \right|^2 = 0. \end{aligned} \tag{A3}$$

Let $g_{i,k} = c_i a_k^{(i)}$; the necessary conditions for the minimizer ($\overline{g}_{i,k} = \overline{c}_i \overline{a}_k^{(i)}$) can be written as

$$\overline{g}_{i,k}(E_k-\Lambda_i)^2 = \mu_k \tag{A4}$$

and

$$b_k - \sum_{i=0}^{N-1} \overline{g}_{i,k} = 0. \tag{A5}$$

μ_k is an undetermined multiplier in Eq. (A4). Combining Eqs. (A4) and (A5) yields

$$\overline{g}_{i,k} = \frac{b_i}{\sum_{j=0}^{N-1} \frac{(E_k-\Lambda_i)^2}{(E_k-\Lambda_j)^2}}. \tag{A6}$$

To understand Eq. (A6), recall that by definition $\{\Lambda_i\}$ is evenly spaced with energy gap ϵ. We call $\overline{d}_{i,k} = |E_k - \Lambda_i|/\epsilon$ the relative spectral distance between the kth eigenmode and $\overline{\Theta}_i = \sum \overline{a}_k^{(i)} \psi_k$. In addition, we interpret $|\overline{g}_{i,k}/b_i|^2$ as the dispersion of the kth eigenmode in the minimizer. When $(\Lambda_{N-1} - \Lambda_0)$ is fixed and N is large enough, Eq. (A6) suggests that $|\overline{g}_{i,k}/b_i|^2$ scales with $1/\overline{d}_{i,k}^2$, which leads to the suggestion that $\frac{\langle\overline{\Theta}_i|(H-\Lambda_i)^2|\overline{\Theta}_i\rangle}{\langle\overline{\Theta}_i|\overline{\Theta}_i\rangle}$ scales with ϵ^2. Hence we can conclude that the particular choice of objective function decided by Eq. (5) can systematically improve the monochromaticity of each Θ_i by squeezing the dispersion of every eigenmode ψ_k.

APPENDIX B: CGSP-ADAPTED NEURAL AUTOREGRESSIVE QUANTUM STATES

In addition to serving as eligible VMC ansatzes, neural autoregressive quantum states can greatly boost the efficiency of stochastic importance sampling. Before the NAQS, the sampling tool accompanying neural quantum states was the Markov chain Monte Carlo by default. Though parallelizable to some extent, the MCMC is essentially a sequential algorithm. The fact that the MCMC requires a long mixing time is not desirable for large-scale deep learning applications using graphics processing units. In contrast, NAQSs realize importance sampling in a parallel manner suitable for GPUs.

A NAQS is a normalized wave function that can be expressed as a product of the conditional wave function. For a general introduction to NAQSs, readers are referred to [47]. Here we give only the example of NAQSs in the context of spin-1/2 models. With the S_z basis of a 1D XXZ model, a NAQS can be expressed as

$$\Upsilon(s_1,\ldots,s_l) = \prod_{i=1}^{l} \phi_i\left(s_{\mu_i}\middle|s_{\mu_{i-1}},\ldots,s_{\mu_1}\right), \tag{B1}$$

where $(\mu_1,\ldots,\mu_l)$ is a permutation of the natural spin order $(1,\ldots,l)$ in a 1D chain. The conditional wave function ϕ_i should satisfy a local normalization condition,

$$\sum_{s'_{\mu_i}\in\{\downarrow,\uparrow\}} \left\|\phi_i\left(s'_{\mu_i}\middle|s_{\mu_{i-1}},\ldots,s_{\mu_1}\right)\right\|^2 = 1, \tag{B2}$$

for any legal configuration $(s_1,\ldots,s_l)$ that does not break any conservation law. With Eqs. (B1) and (B2), the wave function Υ is automatically normalized. When Υ is within a specific S_z sector, any ϕ_i should vanish in illegal configurations. In the realization of an NAQS, any spin-1/2 configuration $(s_{\mu_l}, s_{\mu_{i-1}},\ldots,s_{\mu_1})$ can be encoded by an l-digit binary number where 0 denotes $\downarrow$ and 1 denotes $\uparrow$. Then ϕ_i can be parameterized by neural networks with the input $(s'_{\mu_i}, s_{\mu_{i-1}},\ldots,s_{\mu_1})$ and the output $\phi_i(s'_{\mu_i}|s_{\mu_{i-1}},\ldots,s_{\mu_1})$. In practice, we find that having l different conditional wave functions for a long chain ($l > 20$) is quite clumsy and hard to optimize. So it is helpful to group consecutive spins together. Supposing that l can be divided by 4, a convenient strategy is to convert the l-digit binary number associated with a spin configuration to its hexadecimal equivalent. For example, the spin configuration (0,1,1,0,1,1,0,0) is converted to ((0110),(1100)). In this way the number of conditional wave functions is reduced to one-fourth of the original number. Let $(h_{v_{l/4}}, h_{v_{l/4-1}},\ldots,h_{v_1})$ be the hexadecimal equivalent of $(s_{\mu_l}, s_{\mu_{i-1}},\ldots,s_{\mu_1})$. The total wave function is given as

$$\Upsilon(s_1,\ldots,s_l) = \prod_{i=1}^{l/4} \tilde{\phi}_i\left(h_{v_i}\middle|h_{v_{i-1}},\ldots,h_{v_1}\right), \tag{B3}$$

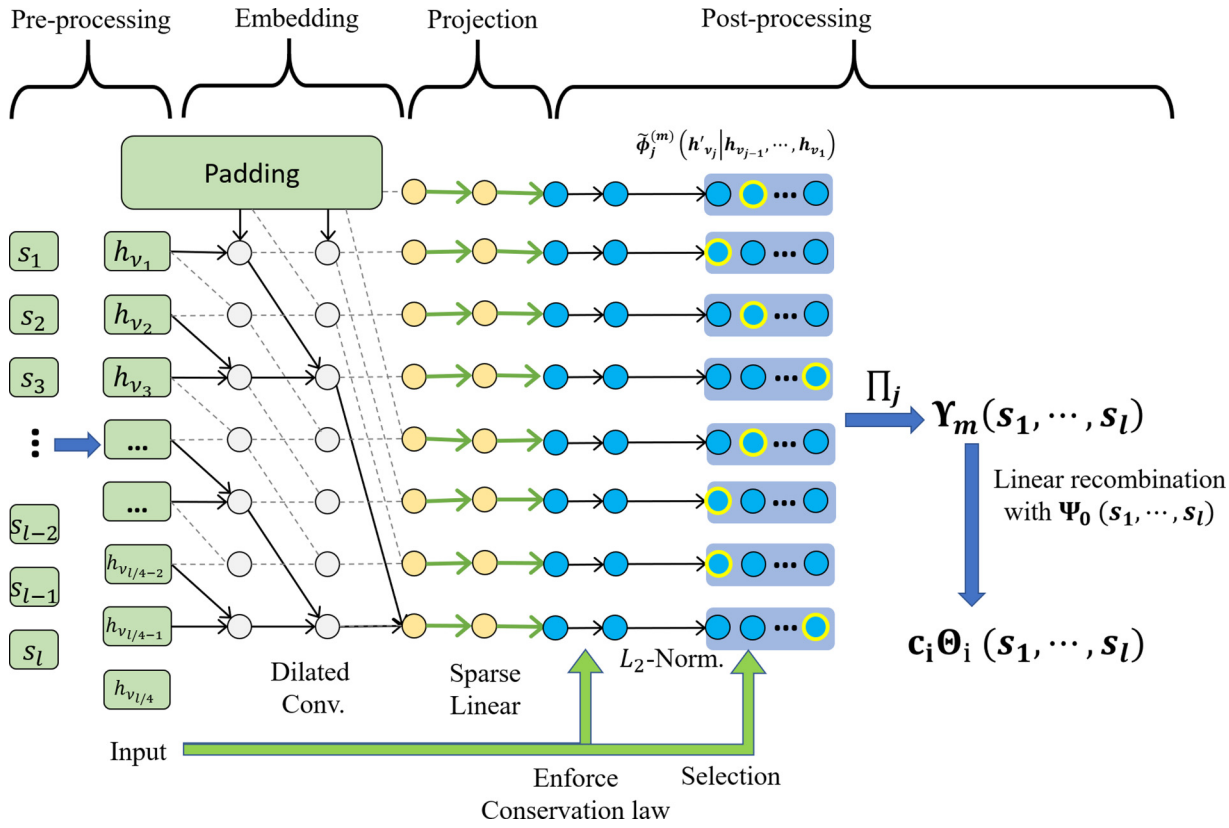

FIG. 3. Structure of a CGSP-adapted NAQS and the workflow of forward propagation in the evaluation mode. The original input is a batch of 1D spin configurations. The output is a batch of complex N vectors, i.e., $\{\Theta_k(s_1,\ldots,s_l)\}_{k\in[0,N-1]}$. The forward propagation consists of four steps. (i) Preprocessing: Reordering of the original spin configuration and regrouping of the outcome into its hexadecimal equivalent. (ii) Embedding: Dilated convolution layers. Only one end of the convolution input is padded with 0 such that the convolution output obeys the conditional dependence required by the NAQS. (iii) Projection: Fully connected layers applied to each input node, respectively. The connection is sparse for the entire input tensor. Let N_B denote the batch size. The output is a 4D (5D) real tensor of size (M, N_B, $l/4$, 16, (2)). The fourth dimension is associated with $[0, 1]^4$, i.e., the possible outcomes of h'_{v_j}. The optional fifth dimension is devoted to representing the real part and the imaginary part of a complex wave function separately. (iv) Postprocessing: Conservation law enforced by multiplication of the tensor element corresponding to illegal spin configurations with 0. L_2 normalization $x \to x/\sqrt{||x||_2}$ is imposed for evaluating the conditional wave function $\tilde{\phi}_j^{(m)}$ in possible configurations. Then with the input spin configuration, one can select the corresponding outcomes from the conditional wave function and compute the total wave function $\Upsilon_m(s_1,\ldots,s_l)$ ($m \in [1, M]$). Finally, with Eq. (10), $\{\Upsilon_m(s_1,\ldots,s_l)\}_{m\in[1,M]}$ is recombined together with the initial condition to produce the final results.

satisfying

$$\sum_{h'_{v_i}\in\{0,1\}^4} \left|\tilde{\phi}_i\left(h'_{v_i}\middle|h_{v_{i-1}},\ldots,h_{v_1}\right)\right|^2 = 1. \tag{B4}$$

In CGSP, we need multiple linearly independent wave functions $\{\Upsilon_m(s_1,\ldots,s_l)\}_{m\in[1,M]}$. It will be unnecessarily expensive if each of them is represented by a totally independent NAQS. It is wiser to allow them to share some of the parameters. Because nonlinearity is applied in each hidden layer of a deep neural network, the sharing of some parameters will not violate the linear independence of the obtained M wave functions. In practice, we let the sharing of parameters occur in the first several hidden layers, which can be understood as a global embedding process.

Figure 3 is a schematic of the NAQS satisfying these requirements. We call it the CGSP-adapted NAQS for ease of reference. A detailed explanation of the forward propagation of the CGSP-adapted NAQS is given in the figure caption. It is noteworthy that our design of the CGSP-adapted NAQS was inspired by WaveNet [50], where dilated convolution with an exponentially increasing dilation size is used to limit the depth of the neural network. We use the same technique in the CGSP-adapted NAQS. So the number of convolution layers required by the conditional dependence of the NAQS is only $O(\log l)$.

The carefully designed structure of the CGSP-adapted NAQS enables the direct sampling of spin configurations in an efficient parallel manner as illustrated in Fig. 4. In the sampling mode of the CGSP-adapted NAQS, we use an auxiliary NAQS, $\Upsilon_0(s_1,\ldots,s_l) = \prod_{i=1}^{l/4} \tilde{\phi}_i^{(0)}(h_{v_i}|h_{v_{i-1}},\ldots,h_{v_1})$, also satisfying the local normalization condition for approximating the initial state $\Psi_0(0)$. When $\Psi_0(0)$ is a simple product state, $\Upsilon_0(s_1,\ldots,s_l)$ can be easily constructed as an exact representation of $\Psi_0(0)$ and used in both the evaluation and the sampling mode of the CGSP-adapted NAQS. Otherwise, $\Upsilon_0(s_1,\ldots,s_l)$ will be utilized only in the sampling mode.

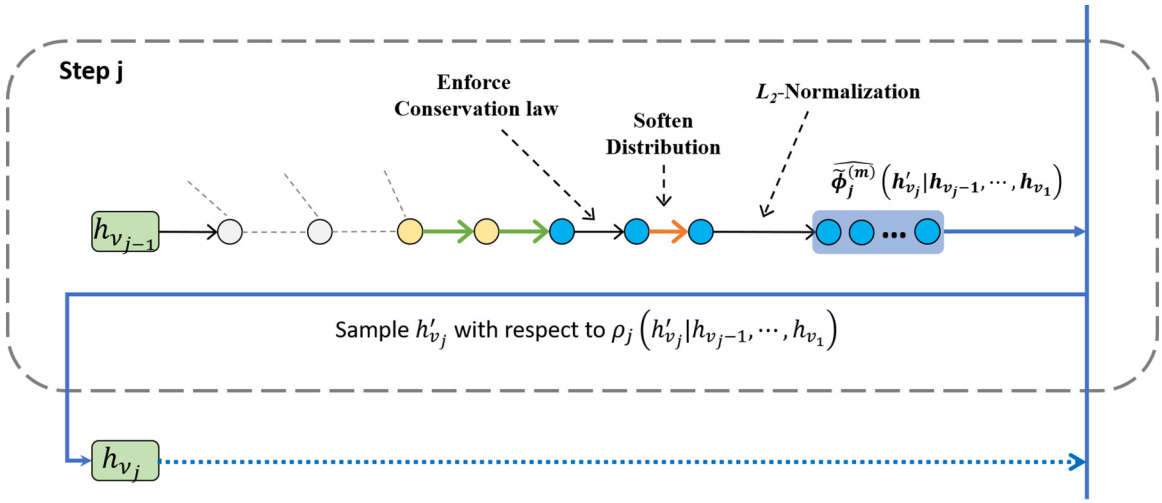

FIG. 4. Schematic of the direct sampling process (sampling mode) of the CGSP-adapted NAQS. Forward propagation is different in the sampling mode in two aspects. First, between enforcing the conservation law and enforcing L_2 normalization, there is an extra operation devoted to softening the distribution to be sampled. The softening acts only on the (real) amplitude part of the unnormalized wave function through $x \to x|x|^{\gamma-1}$. The purpose is to prevent mode collapse, i.e., the neural network continues to generate a small set of samples. The resultant normalized conditional wave function is denoted $\widehat{\tilde{\phi}}_j^m$, to distinguished it from the original $\tilde{\phi}_j^m$. The second difference in the sampling mode is that the postprocessing procedure is terminated immediately after obtaining $\widehat{\tilde{\phi}}_j^m$.

The whole sampling process consists of $l/4$ steps. In the initial step, N_B empty (all-0) spin configurations are generated to be placeholders and fed into the neural network. The softened conditional wave function (softening operation explained in the caption to Fig. 4) $\{\widehat{\tilde{\phi}}_1^m(h'_{v_1})\}_{m\in[1,M]}$ is obtained to sample h'_{v_1} with respect to the distribution

$$\rho_1\left(h'_{v_1}\right) = \frac{\sum_{m=0}^{M} w_m \left\|\widehat{\tilde{\phi}}_1^m\left(h'_{v_1}\right)\right\|^2}{\sum_{m=0}^{M} w_m}, \tag{B5}$$

where the importance weight w_m is suggested by matrix A in Eq. (10). Then the first positions of the N_B placeholders are updated accordingly.

The jth $(1 < j \leqslant l/4)$ step in the sampling process is feeding the N_B placeholders back into the neural network and obtaining $\{\widehat{\tilde{\phi}}_j^m(h'_{v_j}|h_{v_{j-1}},\ldots,h_{v_1})\}_{m\in[1,M]}$. Then h'_{v_j} is sampled with respect to the distribution

$$\rho_j\left(h'_{v_j}|h_{v_{j-1}},\ldots,h_{v_1}\right) = \frac{\sum_{m=0}^{M} w_m \left\|\widehat{\tilde{\phi}}_j^m\left(h'_{v_j}|h_{v_{j-1}},\ldots,h_{v_1}\right)\right\|^2 \prod_{p=1}^{j-1}\left\|\widehat{\tilde{\phi}}_p^m\left(h_{v_p}|h_{v_{p-1}},\ldots,h_{v_1}\right)\right\|^2}{\sum_{m=0}^{M} w_m \prod_{p=1}^{j-1}\left\|\widehat{\tilde{\phi}}_p^m\left(h_{v_p}|h_{v_{p-1}},\ldots,h_{v_1}\right)\right\|^2}. \tag{B6}$$

It is straightforward to verify that $\rho_j(h'_{v_j}|h_{v_{j-1}},\ldots,h_{v_1})$ also satisfies the local normalization condition

$$\sum_{h'_{v_j}\in[0,1]^4} \rho_j\left(h'_{v_j}|h_{v_{j-1}},\ldots,h_{v_1}\right) = 1. \tag{B7}$$

Therefore the target probability distribution of the whole sampling process can be expressed as

$$P\left(h'_{v_1},\ldots,h'_{v_{l/4}}\right) = \prod_{j=1}^{l/4} \rho_j\left(h'_{v_j}|h'_{v_{j-1}},\ldots,h'_{v_1}\right). \tag{B8}$$

It is easy to see that the normalization condition is automatically satisfied.

There are several *ad hoc* parameters to be determined in the sampling mode of the CGSP-adapted NAQS. The first is the real number $0 < \gamma \leqslant 1$ in the softening operation. We find its empirical optimum to be near 0.5. If $\gamma = 1$, this operation is an identity and we find the training of neural network inefficient and suffering from a large local optimum. The second one is the importance weight w_m $(m \in [0, M])$. In our experiments, we let

$$w_m = \sum_i \left|\frac{\delta_{m0}}{N} + A_{im} - \frac{\sum_{k=0}^{N-1} A_{km}}{N}\right|. \tag{B9}$$

In addition, there is the permutation $(\mu_1,\ldots,\mu_l)$ of the natural spin order $(1,\ldots,l)$ to be determined. An adequate permutation μ should minimize "long-range correlation" in the CGSP-adapted NAQS to control the model complexity. In straightforward terms, $\langle s_{\mu_i} s_{\mu_j}\rangle_t - \langle s_{\mu_i}\rangle_t \langle s_{\mu_j}\rangle_t$ should be small for large $|i-j|$ and the time scale with which we are concerned. The design of μ should also take the symmetry of the initial condition, Hamiltonian, and topology of the lattice into consideration. Empirically, we find that the natural spin order is already satisfactory for a 1D chain with open boundary conditions. For periodic boundary conditions, the design of μ relevant to the initial state will require more strategies.

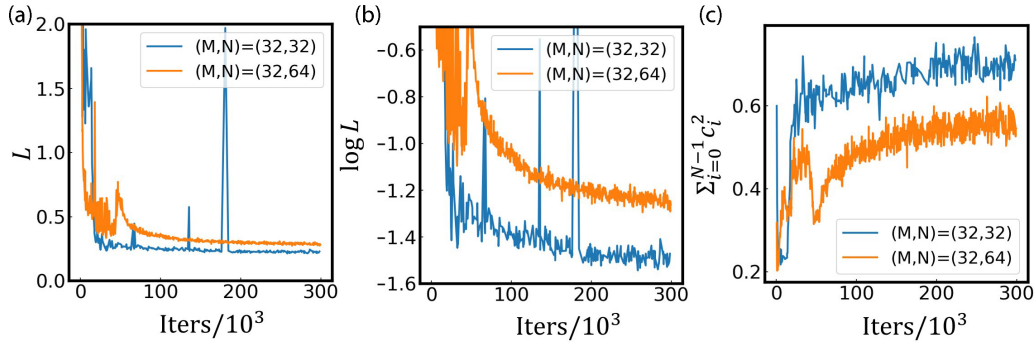

FIG. 5. (a) Estimated loss L versus iterations. (b) Estimated $\log L$ versus iterations. (c) Estimated $\sum_{i=0}^{N-1} c_i^2$ versus iterations.

In summary, the direct sampling algorithm of the CGSP-adapted NAQS allows the generation of N_B samples simultaneously through l sequential tailored forward propagation. This is extremely GPU-friendly compared to MCMC algorithms, which usually require $O(10^3$–$10^6)$ many sequential forward propagations.

APPENDIX C: TECHNICAL DETAILS OF NUMERICAL EXPERIMENTS

The CGSP-adapted NAQSs used for our numerical experiments were implemented in PyTorch [51] as introduced in Appendix B. For the 32-spin 1D XXZ model and our initial state, the projected states could all be real functions. So we restricted our CGSP-adapted NAQS to represent real wave functions only. Our numerical experiments covered two cases: $(M, N) = (32, 32)$ and $(M, N) = (32, 64)$. Because M was identical in these two simulations, the neural network structure was also identical except that the matrix $A = (A_{ij})_{i\in[0,N-1],j\in[0,M]}$ associated with $(M, N) = (32, 64)$ had more parameters than the one associated with $(M, N) = (32, 32)$. For the forward propagation process in these experiments, the neural network contained four dilated convolution layers [dilation $= (1, 1, 2, 4)$, kernel size $= (2, 2, 2, 2)$, out channel $= (128, 128, 128, 128)$]. The outputs of the last three convolution layers were concatenated through the channel dimension and rescaled by a 1×1 convolution layer. This completed the "embedding" stage in Fig. 3 and yielded a 3D tensor of size $(l/4, N_B, 384)$, where $l/4$ corresponded to the number of vertical nodes in Fig. 3. The next stage, "projection," had two sparsely connected linear layers. The first linear layer consisted of $l/4$ small fully connected layers operating on each node independently, yielding a 3D tensor of size $(l/4, N_B, 64M)$. The result was reshaped into a 4D tensor of size $(M, l/4, N_B, 64)$ and fed into the second linear layer consisting of $Ml/4$ small fully connected layers assigned to the first two dimensions, yielding a 4D tensor of size $(M, l/4, N_B, 16)$. The output was reshaped into size $(M, N_B, l/4, 16)$, which completed the projection stage. The last stage, "postprocessing," has been described in Appendix B as well as in the text.

For the 32-spin 1D XXZ model, the dimension of the underlying Hilbert space is about 6×10^8 within the zero total S^z sector. The number of parameters used in TDVP-MPS simulations is about 2×10^5. The number of trainable parameters in both CGSP experiments is about 7×10^6. For about 1% of the Hilbert space complexity, our CGSP-adapted NAQS demonstrated its parameter sharing strategy to be very efficient. The training part of the two CGSP experiments was accomplished by ADAM [52], a first-order gradient descent optimizer with an adaptive learning rate for each parameters. We did not rule out the possibility that second-order optimization methods could be more efficient for CGSP tasks. In both experiments, the total number of stochastic samples for each update (iteration) was 4000 and the learning rate was fixed at 1×10^{-3}. We did not find a learning rate decay improving the convergence. We plot the training curve in Fig. 5. For $(M, N) = (32, 32)$, the wall-clock time for 10^5 iterations trained with two NVIDIA Tesla V100 GPUs was about 6 h. For $(M, N) = (32, 64)$, the wall-clock time for 10^5 iterations trained with four NVIDIA Tesla V100 GPUs was about 5 h.

APPENDIX D: A PARALLEL FRAMEWORK FOR CGSP BREAKDOWN

Training a large neural network with a complicated loss function can be numerically unstable and troubled by the local optimum. So we propose a simple parallel framework for breaking down CGSP into hierarchically organized subtasks. A flowchart of its realization is shown in Fig. 6, where the whole CGSP process is divided into several layers. An initial

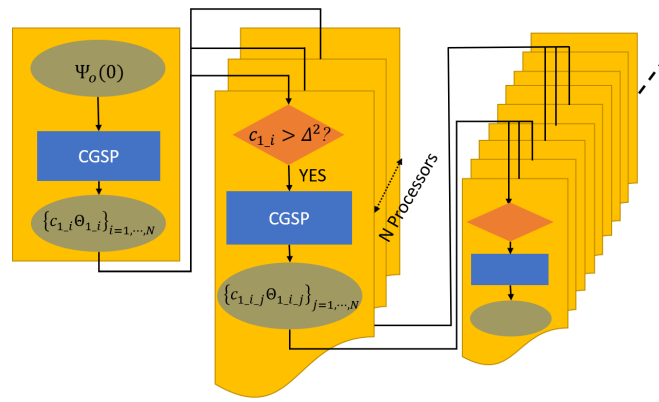

FIG. 6. A parallel scheme for CGSP breakdown. The label of each state implies a tree structure. The label $\iota = o$ denotes the root node, $\iota =$ 1_i denotes the ith child of the root node, and $\iota =$ 1_i_j denotes the jth child of $\iota =$ 1_i.

CGSP of the initial state $\Psi_o(0)$ is carried out in one processor with an affordable M and N. Then the N projected states whose amplitude is above a certain threshold Δ are sent to different processors for the next-layer CGSP. Note that the second-step CGSPs are independent and naturally parallel. This procedure can be repeated for a higher resolution of the spectrum if satisfactory convergence is achieved at each step. At the end, one obtains a family of neural network quantum states organized in a tree structure.

To recover the unitary quantum dynamics, the energy expectation λ_ι of the leaf state labeled ι should be computed for all leaf nodes. The approximation to $\Psi_o(t)$ thus becomes

$$\varphi_o(t) = \sum_{\iota \in \text{leaf nodes}} c_\iota e^{-i\lambda_\iota t}\Theta_\iota. \tag{D1}$$

APPENDIX E: CGSP-INITIALIZED TDVP SIMULATION

Let $H(t)$ be a slowly varying Hamiltonian that the spectral norm $\|dH(t)/dt\|_s$ is bounded by $B > 0$. Let $\Psi_o(t)$ be the pure state evolving with $H(t)$.

Suppose the initial state $\Psi_o(0)$ has already found its CGSP representation, $\Psi_o(0) = \sum_{i=0}^{N-1} c_i\Theta_i$, with $\lambda_i = \frac{\langle\Theta_i|H(0)|\Theta_i\rangle}{\langle\Theta_i|\Theta_i\rangle}$. By allowing the parameters of the neural networks to be time dependent and identifying Θ_i as $\Theta_i(t=0)$, a new ansatz can be defined with $\eta_i(t) = \Theta_i(t)e^{-i\lambda_i t}$. If $\eta_i(t)$ evolves under the Schrödinger equation exactly, then $\sum_{i=0}^{N-1} c_i\eta_i(t)$ is identical to $\Psi_o(t)$. The time-dependent variational principle for $\eta_i(t)$ is written

$$\delta\int_0^t \left\| i\frac{d\eta_i(s)}{ds} - H(s)\eta_i(s)\right\|^2 ds = 0, \tag{E1}$$

which can be translated into

$$\text{minimize} \quad \left\| i\frac{d\Theta_i(t)}{dt} - (H(t)-\lambda_i)\Theta_i(t)\right\|. \tag{E2}$$

Considering that the Hamiltonian is slowly varying, we have

$$\left\|\frac{d\Theta_i(t)}{dt}\right\| < Bt + \|(H(0)-\lambda_i)\Theta_i(t)\|. \tag{E3}$$

If CGSP is successful, one expects $\|(H(0)-\lambda_i)\Theta_i(t)\| \ll 1$ for t small enough. Equation (E3) suggests how CGSP may help with TDVP-based simulation. For the plain TDVP approach, the variation of the ansatz $||\frac{d\Psi(t)}{dt}||$ is bounded by $||H(t)||_s$, which usually increases linearly with the system size for lattice models. This means that the differentiable manifold that the neural network should parametrize increases rapidly for ergodic dynamics. However, with CGSP-initialized TDVP simulation, the desired expressive power of the neural network increases much more slowly due to the constraint, Eq. (E3).

[1] A. M. Kaufman, B. J. Lester, and C. A. Regal, Phys. Rev. X **2**, 041014 (2012).
[2] W. D. Phillips, Rev. Mod. Phys. **70**, 721 (1998).
[3] D. S. Weiss and M. Saffman, Phys. Today **70**(7), 44 (2017).
[4] J. B. Spring, B. J. Metcalf, P. C. Humphreys, W. S. Kolthammer, X.-M. Jin, M. Barbieri, A. Datta, N. Thomas-Peter, N. K. Langford, D. Kundys *et al.*, Science **339**, 798 (2013).
[5] L. M. K. Vandersypen and I. L. Chuang, Rev. Mod. Phys. **76**, 1037 (2005).
[6] H. Schmitz, R. Matjeschk, C. Schneider, J. Glueckert, M. Enderlein, T. Huber, and T. Schaetz, Phys. Rev. Lett. **103**, 090504 (2009).
[7] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z.-X. Gong, and C. Monroe, Nature **551**, 601 (2017).
[8] J. Zhang, P. Hess, A. Kyprianidis, P. Becker, A. Lee, J. Smith, G. Pagano, I.-D. Potirniche, A. C. Potter, A. Vishwanath *et al.*, Nature **543**, 217 (2017).
[9] J. Smith, A. Lee, P. Richerme, B. Neyenhuis, P. W. Hess, P. Hauke, M. Heyl, D. A. Huse, and C. Monroe, Nat. Phys. **12**, 907 (2016).
[10] M. Gring, M. Kuhnert, T. Langen, T. Kitagawa, B. Rauer, M. Schreitl, I. Mazets, D. A. Smith, E. Demler, and J. Schmiedmayer, Science **337**, 1318 (2012).
[11] X.-D. Cai, C. Weedbrook, Z.-E. Su, M.-C. Chen, M. Gu, M.-J. Zhu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, Phys. Rev. Lett. **110**, 230501 (2013).
[12] A. A. Houck, H. E. Türeci, and J. Koch, Nat. Phys. **8**, 292 (2012).
[13] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, Phys. Rev. Lett. **113**, 220501 (2014).
[14] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, Nature **574**, 505 (2019).
[15] J.-W. Pan, Z.-B. Chen, C.-Y. Lu, H. Weinfurter, A. Zeilinger, and M. Żukowski, Rev. Mod. Phys. **84**, 777 (2012).
[16] T. B. Wahl, A. Pal, and S. H. Simon, Phys. Rev. X **7**, 021018 (2017).
[17] T. Devakul and R. R. P. Singh, Phys. Rev. Lett. **115**, 187201 (2015).
[18] F. A. Schröder, D. H. Turban, A. J. Musser, N. D. Hine, and A. W. Chin, Nat. Commun. **10**, 1 (2019).
[19] R. Khasseh, A. Russomanno, M. Schmitt, M. Heyl, and R. Fazio, Phys. Rev. B **102**, 014303 (2020).
[20] J. del Pino, F. A. Y. N. Schröder, A. W. Chin, J. Feist, and F. J. Garcia-Vidal, Phys. Rev. Lett. **121**, 227401 (2018).
[21] A. H. Werner, D. Jaschke, P. Silvi, M. Kliesch, T. Calarco, J. Eisert, and S. Montangero, Phys. Rev. Lett. **116**, 237201 (2016).
[22] P. Doria, T. Calarco, and S. Montangero, Phys. Rev. Lett. **106**, 190501 (2011).
[23] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Phys. Rev. X **8**, 031086 (2018).
[24] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, npj Quantum Info. **5**, 33 (2019).
[25] K. Wang, X. Qiu, L. Xiao, X. Zhan, Z. Bian, W. Yi, and P. Xue, Phys. Rev. Lett. **122**, 020501 (2019).
[26] G. A. Worth, H.-D. Meyer, H. Köppel, L. S. Cederbaum, and I. Burghardt, Int. Rev. Phys. Chem. **27**, 569 (2008).
[27] U. Schollwöck, Ann. Phys. (NY) **326**, 96 (2011).
[28] F. Verstraete, V. Murg, and J. Cirac, Adv. Phys. **57**, 143 (2008).
[29] G. Vidal, Phys. Rev. Lett. **99**, 220405 (2007).
[30] S. R. White, Phys. Rev. Lett. **69**, 2863 (1992).

[31] M. B. Hastings, J. Stat. Mech.: Theory Exp. (2007) P08024.
[32] S. Bravyi, M. B. Hastings, and F. Verstraete, Phys. Rev. Lett. **97**, 050401 (2006).
[33] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. X **7**, 021021 (2017).
[34] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[35] J. Han, L. Zhang, and Weinan E, J. Comput. Phys. **399**, 108929 (2019).
[36] K. Choo, T. Neupert, and G. Carleo, Phys. Rev. B **100**, 125124 (2019).
[37] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, Phys. Rev. Res. **2**, 033429 (2020).
[38] D. Luo and B. K. Clark, Phys. Rev. Lett. **122**, 226401 (2019).
[39] J. Hermann, Z. Schätzle, and F. Noé, Nat. Chem. **12**, 891 (2020).
[40] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, Phys. Rev. Lett. **122**, 065301 (2019).
[41] M. Schmitt and M. Heyl, Phys. Rev. Lett. **125**, 100503 (2020).
[42] G. Fabiani and J. Mentink, arXiv:1912.10845.
[43] M. J. Hartmann and G. Carleo, Phys. Rev. Lett. **122**, 250502 (2019).
[44] N. Yoshioka and R. Hamazaki, Phys. Rev. B **99**, 214306 (2019).
[45] R. L. Kosut, T.-S. Ho, and H. Rabitz, Phys. Rev. A **103**, 012406 (2021).
[46] F. Alet and N. Laflorencie, C.R. Phys. **19**, 498 (2018).
[47] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, Phys. Rev. Lett. **124**, 020503 (2020).
[48] https://github.com/salinelake/cgsp.
[49] S. Czischek, M. Gärttner, and T. Gasenzer, Phys. Rev. B **98**, 024311 (2018).
[50] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, *9th ISCA Speech Synthesis Workshop* (ISCA, Sunnyvale, 2016), p. 125.
[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Red Hook, NY, 2019), pp. 8026–8037.
[52] D. P. Kingma and J. Ba, *ICLR 2015: International Conference on Learning Representations 2015* (ICLR, San Diego, 2015).