Understanding and eliminating spurious modes in variational Monte Carlo using collective variables

Huan Zhang,[1] Robert J. Webber,[2] Michael Lindsey,[3] Timothy C. Berkelbach,[4,5,*] and Jonathan Weare[1,†]

[1]*Courant Institute of Mathematical Sciences, New York University, New York 10012, USA*

[2]*Division of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California 91125, USA*

[3]*Department of Mathematics, University of California, Berkeley, California 94720, USA*

[4]*Department of Chemistry, Columbia University, New York, New York 10027, USA*

[5]*Center for Computational Quantum Physics, Flatiron Institute, New York, New York 10010, USA*

(Received 10 November 2022; accepted 24 March 2023; published 15 May 2023)

The use of neural network parametrizations to represent the ground state in variational Monte Carlo (VMC) calculations has generated intense interest in recent years. However, as we demonstrate in the context of the periodic Heisenberg spin chain, this approach can produce unreliable wave function approximations. One of the most obvious signs of failure is the occurrence of random, persistent spikes in the energy estimate during training. These energy spikes are caused by regions of configuration space that are over-represented by the wave function density, which are called "spurious modes" in the machine learning literature. After exploring these spurious modes in detail, we demonstrate that a collective-variable-based penalization yields a substantially more robust training procedure, preventing the formation of spurious modes and improving the accuracy of energy estimates. Because the penalization scheme is cheap to implement and is not specific to the particular model studied here, it can be extended to other applications of VMC where a reasonable choice of collective variable is available.

DOI: 10.1103/PhysRevResearch.5.023101

I. INTRODUCTION

Variational Monte Carlo (VMC) is an algorithm for approximating the ground-state energy and wave function of a quantum many-body system [1,2]. As a variational method, VMC seeks the lowest-energy wave function $\psi_{\boldsymbol{\theta}}(\cdot)$ by minimizing the energy with respect to a set of variational parameters $\boldsymbol{\theta}$. Building on a history of successful VMC applications, researchers have recently introduced neural-network-based families of wave functions that can be evaluated and differentiated efficiently [3–6]. Although neural networks are sufficiently flexible to represent difficult wave functions [7], neural network parameter optimization can be slow or unstable, and parameters can converge to local minima [8,9], limiting the accuracy that can be practically attained.

Here, we identify the formation of *spurious modes* as a problem that degrades the accuracy and robustness of neural VMC wave functions. A spurious mode is defined in the machine learning literature as a high-probability region that is absent in the data distribution but present in the model distribution (see, for example, Ref. [10], Secs. 18.1 and 18.2). Like machine learning for probability density estimation, VMC needs to sample from the wave function probability density $\rho_{\boldsymbol{\theta}} \propto |\psi_{\boldsymbol{\theta}}(\cdot)|^2$ in order to estimate the energy and energy gradient. In this work, we show that spurious modes can occur in VMC, i.e., parameter updates yield a wave function probability density that is artificially large in regions far away from the samples, as illustrated schematically in Fig. 1. As can be seen in this figure and documented in detail in this paper, the formation of spurious modes is only possible because the variational wave function is unconstrained in undersampled regions of configuration space. As a symptom, the VMC energy estimator typically exhibits a large energy spike when the sampler first encounters a spurious mode. The energy spike can persist over thousands of optimization steps, making it difficult to extract a usable energy estimate from VMC.

This work can be viewed as a constructive approach for diagnosing and mitigating generalization error from limited Monte Carlo sampling, which is already recognized as a challenge for VMC [11]. Indeed, it is important to address the problem of spurious modes as neural VMC rapidly grows in popularity and finds applications to systems of ever increasing complexity [1–10]. We might hope that enhanced sampling techniques such as parallel tempering and umbrella sampling [12–14] could remedy the issue of spurious modes, as has been suggested in the machine learning literature [15]. However, in the context of VMC, we demonstrate that enhanced sampling methods do not solve the problem.

We introduce collective-variable-informed VMC (CV-VMC) as a new, effective strategy for addressing spurious

*tim.berkelbach@gmail.com

†weare@nyu.edu

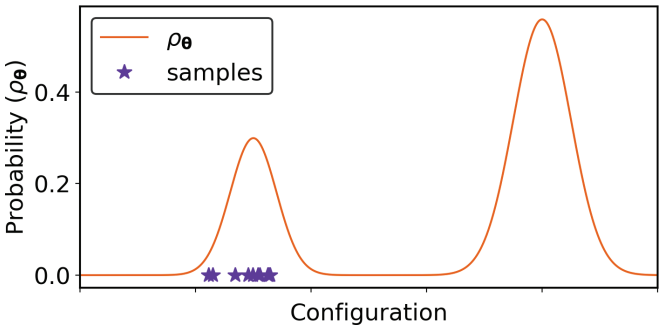

FIG. 1. A spurious mode is a high-probability region that is absent in the empirical sample distribution but present in the model distribution.

modes in VMC. To our knowledge we are the first to link the spurious modes in VMC to the appearance of energy spikes and also the first to offer a satisfactory resolution. The main idea of our CV-VMC framework is to exploit a well-chosen one-dimensional collective variable (CV) that distinguishes physically reasonable configurations from physically unreasonable configurations. More specifically, we generate a pool of samples across a range of CV values, and we use the samples to penalize large wave function densities in physically unreasonable regions of configuration space.

CV-VMC is simple and cheap to implement, and it leverages physical intuition to accelerate and improve VMC optimization. This strategy is inspired by the use of CVs in guiding free energy calculations [16,17]. While the design of appropriate CVs must be calibrated based on the type of problem, this work advances a general and modular framework for incorporating *a priori* intuitions about the wave function into the optimization procedure itself.

The paper is structured as follows. Section II presents the model problem of interest as well, as well as the traditional VMC optimization approach. In Sec. III, we study the appearance of energy spikes in VMC, connect the spikes to the existence of spurious modes, and show the inadequacy of generic enhanced sampling techniques for addressing this problem. Section IV introduces and tests the CV-VMC approach for eradicating the spurious modes. Section V concludes.

II. PRELIMINARIES

Throughout this work, we apply VMC to the antiferromagnetic Heisenberg model for N spin-1/2 particles in a one-dimensional periodic chain, defined by the Hamiltonian

$$\hat{H} = \sum_{i=1}^{N} \left(\hat{\sigma}_i^x \hat{\sigma}_{i+1}^x + \hat{\sigma}_i^y \hat{\sigma}_{i+1}^y + \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z\right), \tag{1}$$

where $\hat{\sigma}_i^x$, $\hat{\sigma}_i^y$, and $\hat{\sigma}_i^z$ are the Pauli operators for the ith spin. Here and throughout, periodic boundary conditions are implied via the identification $\hat{\sigma}_{N+1} = \hat{\sigma}_1$. For our variational wave function, we use the neural quantum state ansatz [3] inspired by the restricted Boltzmann machine (RBM) [18]. This ansatz is a two-layer (or one-hidden-layer) neural network that has been widely used in VMC in recent years [19]. We refer to our ansatz as the RBM throughout. Working in the many-body basis of spin configurations $\boldsymbol{\sigma}$ defined as simultaneous eigenstates of the operators $\{\hat{\sigma}_i^z\}$ with eigenvalues $\{\sigma_i\}$ (dropping the z indicator for notational simplicity), an RBM wave function can be written as

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \sum_{\{h_k\}} \exp\left(\sum_{i=1}^{N} a_i \sigma_i + \sum_{k=1}^{M} b_k h_k + \sum_{ik} W_{ki} h_k \sigma_i\right), \tag{2}$$

where $\sigma_i \in \{-1, +1\}$ are the spin variables, $h_k \in \{-1, +1\}$ are an additional set of M hidden spin variables, and $\boldsymbol{\theta} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{W}\}$ are the variational parameters. Summing over the hidden spins h_k and enforcing the translational symmetry that is exhibited by the exact ground state gives the modified RBM ansatz,

$$\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) = \prod_{k=1}^{M} \prod_{j=1}^{N} \cosh\left(b_k + \sum_{i=1}^{N} W_{ki} \sigma_{i+j}\right), \tag{3}$$

which reduces the set of variational parameters to $\boldsymbol{\theta} = \{\boldsymbol{b}, \boldsymbol{W}\}$, and, again, periodic boundary conditions are implied.

For fixed parameters $\boldsymbol{\theta}$, the energy can be calculated as

$$E = \frac{\langle \psi_{\boldsymbol{\theta}}, H\psi_{\boldsymbol{\theta}} \rangle}{\langle \psi_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}} \rangle} = \sum_{\{\sigma_i\}} E_{\mathrm{loc}}(\boldsymbol{\sigma}) \rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma}), \tag{4}$$

where $E_{\mathrm{loc}}(\boldsymbol{\sigma}) = (H\psi_{\boldsymbol{\theta}})(\boldsymbol{\sigma})/\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is the local energy, $\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \propto |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2$ is the normalized probability density, and we have adopted the inner product notation

$$\langle \psi, \phi \rangle = \sum_{\{\sigma_i\}} \overline{\psi(\boldsymbol{\sigma})} \phi(\boldsymbol{\sigma}). \tag{5}$$

It remains to optimize the parameters $\boldsymbol{\theta}$ in the RBM ansatz to minimize the energy functional E. Here, we use the stochastic reconfiguration (SR) [2] algorithm. In the SR method, the parameter update $\boldsymbol{\delta}$ can be derived by minimizing a cost function

$$\frac{\langle \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}, H\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}} \rangle}{\langle \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}, \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}} \rangle} - \frac{1}{\epsilon} \left(\frac{|\langle \psi_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}} \rangle|}{\|\psi_{\boldsymbol{\theta}}\| \|\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}\|} \right)^2, \tag{6}$$

which contains the usual energy expression (4) and an additional penalization term that prevents large wave function updates. After differentiating the cost function (6) and performing algebraic manipulations (see Ref. [19]), this approach leads to the following algorithmic approach to VMC.

Algorithm 1 (VMC via SR). Choose the parameter update $\boldsymbol{\delta}$ to solve

$$(\boldsymbol{S} + \eta \boldsymbol{I})\boldsymbol{\delta} = -\epsilon \boldsymbol{g}. \tag{7}$$

Here, $\eta \geqslant 0$ is a nonnegative parameter chosen to make $\boldsymbol{S} + \eta \boldsymbol{I}$ positive definite. The energy E, gradient vector $\boldsymbol{g}$, and overlap matrix $\boldsymbol{S}$ are defined by

$$E = \mathbb{E}_{\rho_{\boldsymbol{\theta}}}[E_{\mathrm{loc}}(\boldsymbol{\sigma})], \tag{8a}$$

$$g_i = \mathrm{Cov}_{\rho_{\boldsymbol{\theta}}}\left[\frac{\partial_{\theta_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \frac{H\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}\right], \tag{8b}$$

$$S_{ij} = \mathrm{Cov}_{\rho_{\boldsymbol{\theta}}}\left[\frac{\partial_{\theta_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \frac{\partial_{\theta_j}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}\right], \tag{8c}$$

where $\mathbb{E}_{\rho_\theta}$ and $\mathrm{Cov}_{\rho_\theta}$ indicate the expectation value and covariance with respect to the current probability distribution $\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$.

Because of the high dimensionality, the averages appearing above are estimated stochastically. With SR, this leads to the following iterative VMC strategy.

(1) Draw samples from the probability distribution

$$\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \propto |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2. \tag{9}$$

(2) Use the samples to provide an energy estimate $\hat{E}$, gradient estimate $\hat{\boldsymbol{g}}$, and overlap estimate $\hat{\boldsymbol{S}}$ by Eqs. (8).

(3) Update $\boldsymbol{\theta}$ by solving the regularized linear system

$$(\hat{\boldsymbol{S}} + \eta \boldsymbol{I})\boldsymbol{\delta} = -\epsilon \hat{\boldsymbol{g}} \tag{10}$$

and setting $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \boldsymbol{\delta}$.

In this paper, we generate samples from the wave function density $\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma}) \propto |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2$ using a Metropolis-type Markov chain Monte Carlo (MCMC) sampler. Because the Hamiltonian (1) conserves the total magnetization, we focus on the sector with $\sum_i \sigma_i = 0$. Our sampler starts from a uniformly distributed configuration within the subspace and attempts to swap a randomly chosen $+1$ spin with a randomly chosen -1 spin. During each iteration (defined as one parameter update), we generate data by running each MCMC chain for 2000 Metropolis steps, and we subsample the data once every 100 steps to reduce the storage and computation costs. We typically run $n = 100$ independent MCMC walkers of this type and pool together the resulting data to calculate parameter updates. We initialize the MCMC chains at each new parameter value using the final configurations from the previous iteration. However, as enhanced sampling alternatives, we also experiment with the parallel tempering [12] and umbrella sampling [13] methods described in Sec. III C.

Throughout this work, we use $M = 5N$ hidden spins. The results are similar when we use $M = 3N$ hidden spins or $M = 4N$ hidden spins; however, with fewer than $M = 3N$ hidden spins, the ansatz is less flexible and the energy estimates are less accurate. We choose optimization parameter that ensure numerical stability, while also allowing for large updates whenever possible. Following the procedure in [19], we initialize our neural network wave function parameters as independent complex-valued $\mathcal{N}(0, 0.001)$ random variables. We increase the step size parameter ϵ from $\epsilon = 0.001$ to $\epsilon = 0.01$ at a geometric rate over first 500 iterations, after which it is held constant, and we use $\eta = 0.001$ for all iterations. When a large parameter update occurs, we restore ϵ to its initial value and restart the geometric progression.

III. A STUDY OF SPURIOUS MODE FORMATION

A. Energy spikes and spurious modes

When we optimize our VMC wave function for 5000 iterations, we obtain the results depicted in Fig. 2. The VMC energy error decreases over the initial 1800 iterations and begins to exhibit high-frequency fluctuations on the scale of 10^{-4}. Then, during iterations 1800–5000, large sustained energy spikes occur in 9 out of 20 independent VMC training runs. In several training runs, the energy spikes appear more than once.

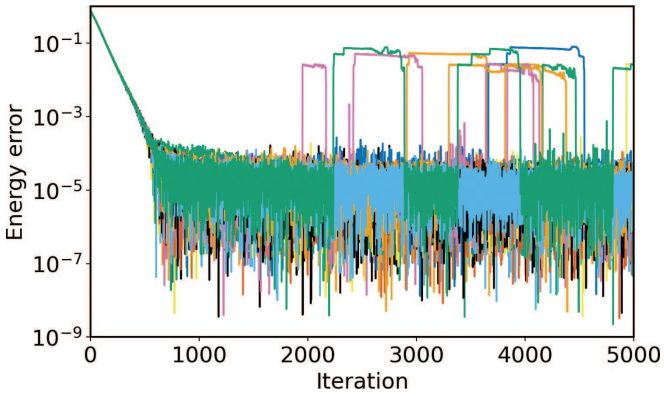

FIG. 2. Per-site error in VMC energy estimates obtained over 20 independent runs for a chain of $N = 100$ spins. The exact energy is computed using the Bethe ansatz [20].

The occurrence of the energy spikes is concerning for two reasons. First, the repeated spikes make it difficult to decide when the VMC statistics converge and when to stop training. Second, spikes may arise unexpectedly when an apparently converged VMC wave function is used for downstream investigations, including the computation of other observables besides the energy [21] and the refinement of the energy estimate by diffusion Monte Carlo [22]. We explore this possibility in more detail in Sec. III B.

The frequency of the spikes depends on the number N of sites in the chain and the amount of sampling performed at each training iteration. In Table I, we report the number of energy spikes observed in 20 training runs for various chain lengths N and numbers n of parallel MCMC walkers. When N is small ($N \leqslant 50$), there are fewer spikes. When the number of walkers is small ($n \leqslant 20$), the energy spikes rarely if ever occur, but in this case the VMC energy estimates are inaccurate, with variance 10 times higher than in the $n = 100$ case. If infinitely many MCMC steps were performed between parameter updates and ϵ were sufficiently small, energy spikes could not occur [19]. In practice, however, VMC is carried out far from this limit. With $n = 500$ parallel walkers, we still observe many training runs with energy spikes.

We can zoom in on an energy spike to understand the phenomenon better. Figure 3 presents a typical VMC training run exhibiting energy spikes. The first spike occurs at iteration 2236, which is marked by the orange dot in the upper panel. The lower panel shows the cause of the energy spike. During

TABLE I. The number of training runs (out of 20) exhibiting energy spikes. Energy spikes are identified by per-site energy estimates with error $\geqslant 10^{-3}$.

	$N = 50$	$N = 100$	$N = 200$
$n = 20$	0	0	0
$n = 50$	1	8	6
$n = 100$	0	9	9
$n = 200$	1	8	14
$n = 500$	0	5	12

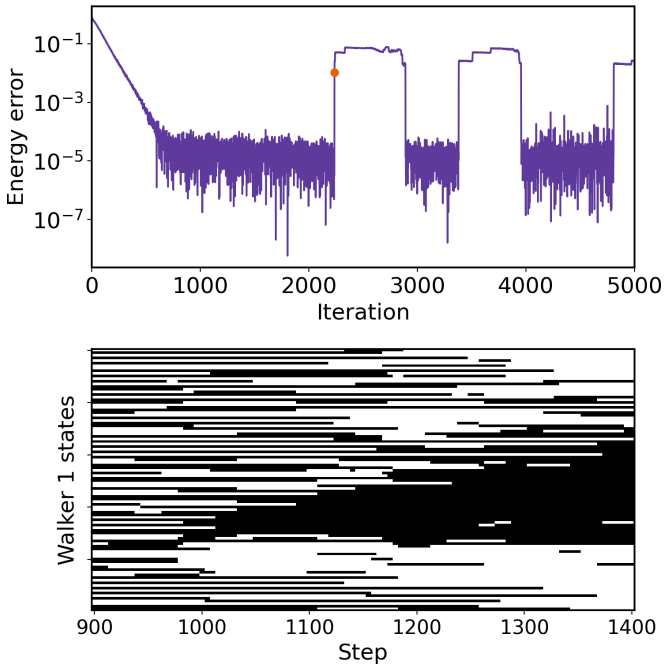

FIG. 3. (Top) Per-site energy error, with an orange dot marking iteration 2236 in which the first energy spike occurs. (Bottom) States of walker 1 during steps 900–1400 of iteration 2236, with black indicating up spins and white indicating down spins.

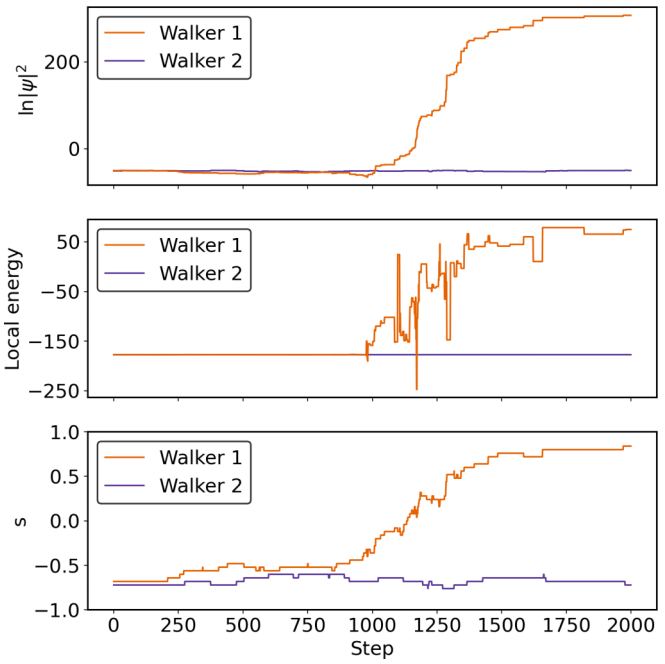

FIG. 4. Values of $\ln|\psi|^2$ (top), local energy (middle), and s (bottom) for walker 1 and walker 2.

the sampling stage of iteration 2236, a single MCMC chain ("walker 1") transitions suddenly from an antiferromagnetic state (characteristic of the ground state of the antiferromagnetic Heisenberg model studied here) to a ferromagnetic state with two domain walls.

The statistics of walker 1 are presented in more detail in Fig. 4. We notice that walker 1 experiences an abrupt increase in its estimated probability density $\rho_{\boldsymbol{\theta}}$ (consistent with a spurious mode), as well as a large increase in its local energy, yielding the spike in the energy estimate. Motivated by the trajectory in Fig. 3, we characterize this transition with a CV that captures the local magnetic ordering,

$$s(\boldsymbol{\sigma}) \equiv \frac{1}{N}\sum_{i=1}^{N}\sigma_i\sigma_{i+1}. \tag{11}$$

The collective variable s ranges from -1 (antiferromagnetic), to 0 (nonmagnetic), to $+1$ (ferromagnetic). For the antiferromagnetic Heisenberg Hamiltonian (1), we expect the ground state to be predominantly supported by configurations with $s < 0$. We see in Fig. 4 that the value of s for walker 1 sharply increases exactly when the energy spike occurs.

For comparison, we also plot data for one of the MCMC chains ("walker 2") that shows no abrupt change in either the wave function magnitude or local energy. Most of the $n = 100$ MCMC walkers have profiles similar to walker 2. Apart from walker 1, only two other MCMC chains ever enter the $s > 0$ region and contribute to the energy spike, starting at iterations 2241 and 2332.

We loosely define a *spurious mode* as a collection of configurations $\boldsymbol{\sigma}$ for which the wave function probability density $\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ is large and $s(\boldsymbol{\sigma}) > 0$, the latter of which implies that the local energy will be large for this Hamiltonian. As seen in Fig. 4, the energy spike begins when walker 1 suddenly encounters a configuration in a spurious mode.

Figure 5 charts the emergence of a spurious mode over thousands of optimization steps. The orange line indicates the marginal probability density for s,

$$P(s') = \sum_{\boldsymbol{\sigma}} \delta_{s',s(\boldsymbol{\sigma})}\,\rho_{\boldsymbol{\theta}}(\boldsymbol{\sigma}), \tag{12}$$

where δ_{s_1,s_2} indicates the Kronecker delta, and $P(s)$ is estimated using the procedure in Appendix B. At time $t = 0$, the MCMC walkers are randomly initialized with a symmetric distribution of s values. However, by time $t = 400$, all the walkers have moved into the $s < 0$ region that is physically relevant for the antiferromagneic Heisenberg model. Next, the marginal probability density $P(s)$ in the undersampled $s > 0$ region starts to increase, forming a spurious mode. Starting at iteration $t = 2236$, several of the walkers find their way across the energy barrier to the spurious mode, and they become stuck due the high wave function density there. The local energies in the $s > 0$ region are far higher than the ground state energy, leading to a dramatic energy spike. The bottom row of Fig. 5 demonstrates the spurious mode disappearing. Eventually, the spurious mode disappears and walkers return to the $s < 0$ region. In the final panel, the spurious mode has reappeared, which will lead to another energy spike.

As has been well established, the single-layer RBM ansatz is flexible enough to accurately approximate the physical wavefunction. What we see in Fig. 5 is that it is also flexible enough to form a spurious mode in regions where it is not constrained by samples, and it will typically do so. Restricting samples to the region of high probability would only exacerbate the problem of spurious mode formation. In contrast,

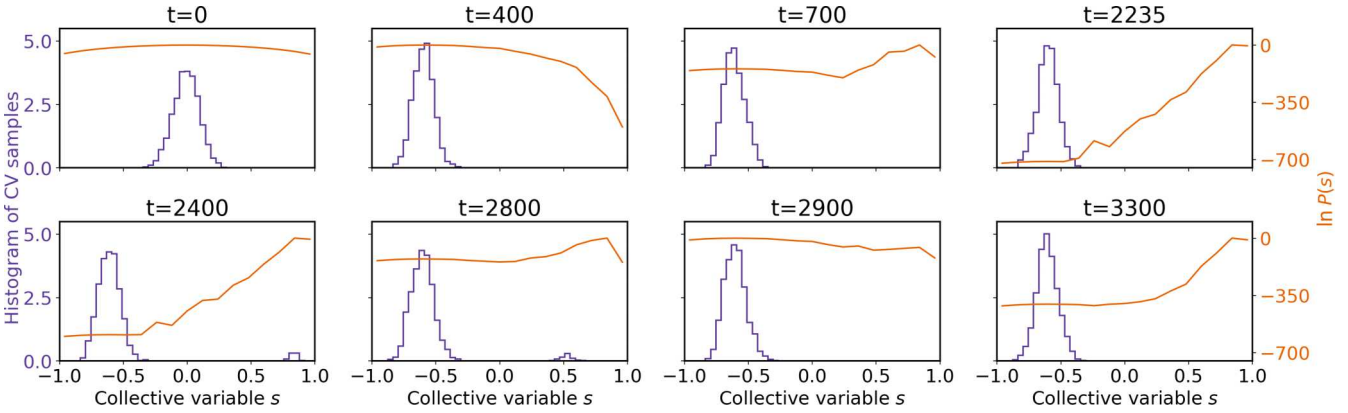

FIG. 5. Empirical histogram of CV samples (purple) and the marginal probability density $P(s)$ of the CV (orange).

the mitigation strategies explored in this paper all involve introducing additional samples in the low probability region.

B. Robustness testing identifies spurious modes

With a view toward downstream tasks, we assess the reliability of the optimized wave function (i.e., the possibility of a spurious mode) at a given VMC iteration using the following robustness test.

(i) With the parameters $\boldsymbol{\theta}$ held fixed to their value at the current iteration, we choose 100 configurations from the final states of the MCMC chains used to train the optimized wave function. Starting from these states, we generate an additional 2×10^6 Metropolis steps, saving samples every 10^5 steps.

(ii) We compute an energy estimate using the resulting 2×10^3 sample configurations.

(iii) We repeat steps (i) and (ii) 10 times and compare the resulting 10 energy estimates.

The mean and variance calculated from steps (ii) and (iii) are indicative of the global quality of the current wave function and the existence of spurious modes. Figure 6 presents results from three independent VMC optimizations, showing the instantaneous energy error (red) and the statistics of the energy errors calculated as described above (box-and-whisker plots, black).

There are several possible outcomes: the VMC training shows no spikes and the robustness testing shows no spikes (top panel), the VMC training shows no spikes but the robustness testing shows spikes (middle panel), or both training and robustness testing show spikes (bottom panel). We infer that the optimized wave function in the top panel has no spurious modes, the optimized wave function in the middle panel develops a spurious mode around iteration 2500 that is *unobserved* during training even after 5000 iterations, and the optimized wave function in the third panel develops multiple spurious modes that are also evident in training.

These results highlight the fact that accurate energies during training do not guarantee accurate energies during testing. Indeed, we tested the robustness of the 20 VMC wave functions obtained via the 20 independent VMC runs shown in Fig. 2. Although 11 of the 20 VMC runs did not show any energy spikes during training, 5 of these 11 VMC runs exhibited energy spikes in testing, with an energy spike here defined as a per-site energy estimate with error $\geqslant 10^{-3}$.

C. Enhanced sampling does not prevent spurious mode formation

We have demonstrated that undersampled regions are prone to the formation of spurious modes in the VMC wave

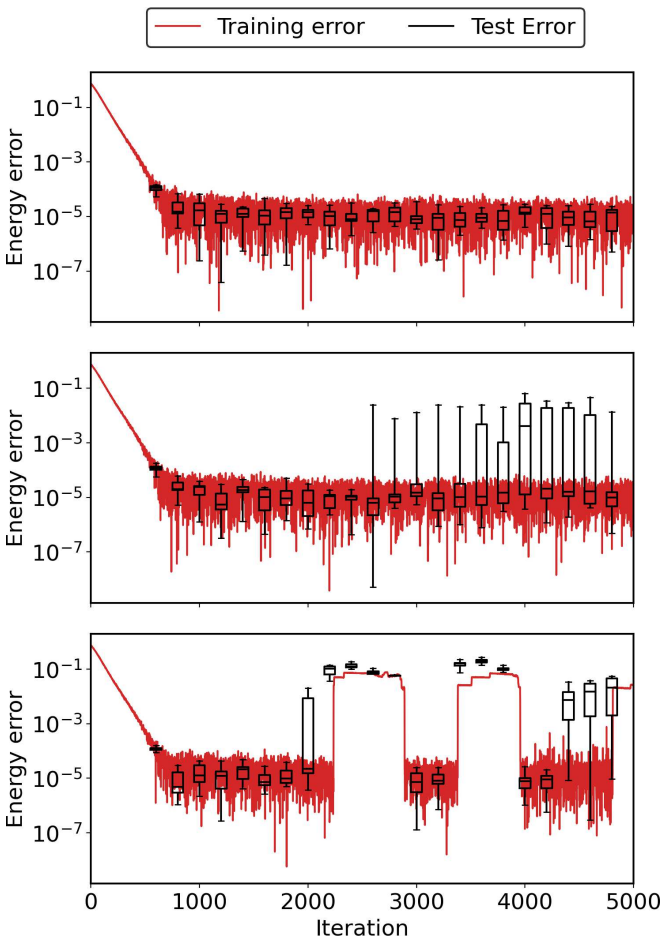

FIG. 6. Per-site energy errors during training (red) and during robustness testing (box-and-whisker plots, black) for three independent VMC training runs.

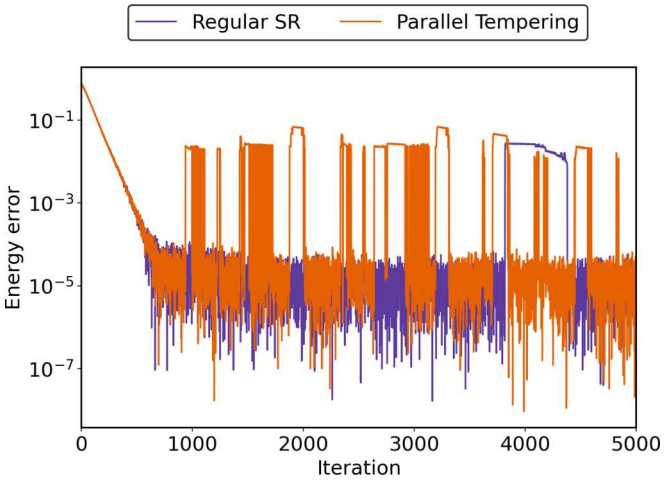

FIG. 7. VMC with parallel tempering (orange) leads to shorter, more frequent spikes than VMC with direct MCMC sampling (purple).

function. Enhanced sampling methods, which have been developed specifically to increase sampling in low-probability regions without sacrificing the in-principle exactness of Monte Carlo estimation, seem to offer a straightforward remedy. We explore two of the most commonly used enhanced sampling methods, parallel tempering and umbrella sampling, but find that in fact they do not prevent the formation of spurious modes.

Parallel tempering is a popular enhanced sampling method across a wide range of applications [23], including in RBM training [15]. The previous paper [19] applied parallel tempering to improve sampling in VMC. In parallel tempering, multiple MCMC chains indexed by k are simulated, each sampling from a density proportional to $\rho_{\boldsymbol{\theta}}^{\beta_k}$, with the values of the β_k spaced to cover the interval [0,1]. The states of these chains are exchanged when appropriate, ideally allowing the chain sampling $\rho_{\boldsymbol{\theta}}$ itself to escape local maxima of the density [23]. Only data from the chain sampling $\rho_{\boldsymbol{\theta}}$ are used to compute averages.

In our tests, parallel tempering causes the spikes to occur more frequently but subside more rapidly, as shown in Fig. 7. (See Appendix A for more details.) Once the spurious mode is fully formed, parallel tempering can transport the chain there promptly, and subsequent parameter updates can partially correct the wave function within the spurious mode. However, before the spurious mode is fully formed, samples from $\rho_{\boldsymbol{\theta}}$ are concentrated in the $s < 0$ region, and the emerging spurious mode has no effect on training. In short, parallel tempering does nothing to prevent the emergence of the spurious modes in the first place.

We also consider the umbrella sampling method, which has been used in free energy calcuations for decades [24,25]. Umbrella sampling in the form used here is a stratified sampling method for general averages, as suggested and explored in Ref. [13]. In umbrella sampling, each MCMC sampler is restricted to remain within a range, or "window," of possible s values. By covering the range of all possible s values with such windows (16 windows in our tests), sampling resolution is increased in the $s > 0$ region relative to unbiased MCMC

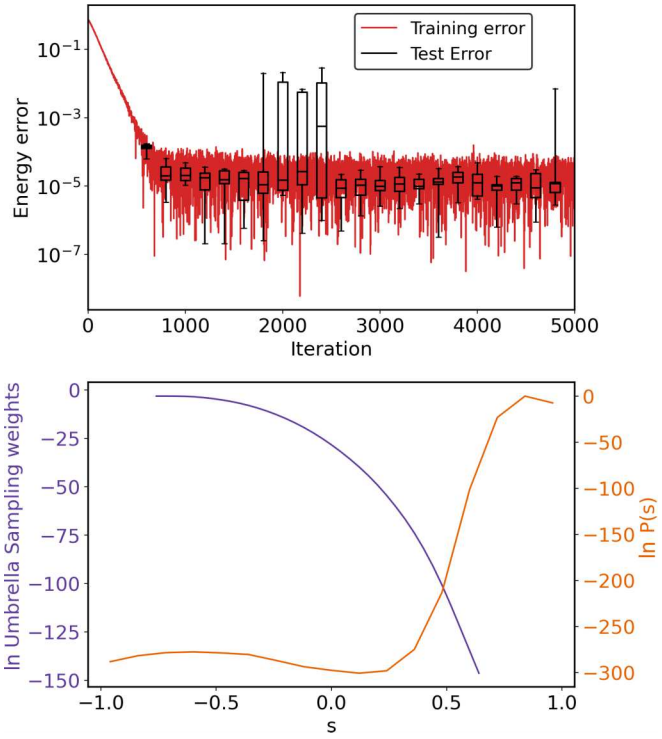

FIG. 8. (Top) Per-site energy errors during the training (orange) and during the testing (box-and-whisker plots, black). (Bottom) Weights assigned to each of the restrained umbrella sampling MCMC samplers (purple), along with the marginal probability density $P(s)$ of the CV (orange).

sampling. Statistical weights are then assigned to samples to correct for the biased sampling distribution. See Appendix B or Ref. [13] for further details of the method.

While umbrella sampling does eliminate visible energy spikes during training, we find that, like parallel tempering, it does not prevent the formation of spurious modes. A typical training run and the results of our robustness test are shown Fig. 8 (top panel). Because the proposal distribution used in our Metropolis sampling scheme favors moves toward $s = 0$, the MCMC samplers restrained to regions of higher s suffer from very low acceptance rates (0.01–0.05) and tend to discover the spurious mode well after it has formed. Having not discovered the region of artificially high probability, umbrella sampling assigns very small statistical weight to samples in the higher s region, and the emerging spurious mode has no impact on parameter updates. In Fig. 8 (bottom panel), for a representative choice of parameters, we validate this claim by comparing the average statistical weight assigned to samples at each value of s to the marginal density $P(s)$ of $\rho_{\boldsymbol{\theta}}$. Details of this calculation can be found in Appendix B.

Robustness tests for the parallel tempering trained wave function approximation are carried out exactly as described in Sec. III B. Robustness tests for the umbrella sampling-trained wave function approximation require an additional resampling of the final states of the MCMC chains used in training because of the statistical weights assigned to samples in umbrella sampling. For more details see Appendix D.

IV. COLLECTIVE-VARIABLE-INFORMED VMC

In this section, we introduce a new term into the objective function of VMC meant to penalize the formation of spurious modes. The new term, which we call the "spurious mode functional," eliminates spurious modes by confining the wave function to the physically reasonable region of s values. Moreover it is quick to evaluate and differentiate. In the next section, we mathematically motivate our method, which we call collective-variable-informed VMC (CV-VMC). The new objective that defines CV-VMC is optimized via SR. Results from numerical experiments are presented in the following section.

A. Mathematical motivation

Define the spurious mode functional L_{CV} by

$$L_{\mathrm{CV}}(\boldsymbol{\theta}) = -\Delta\mathrm{Cov}_{\boldsymbol{\sigma}\sim p}[\ln|\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2, \mathbb{1}_{s(\boldsymbol{\sigma})<c}], \tag{13}$$

where we have used the symbol $\mathbb{1}_I$ to indicate the function that is 1 when $\boldsymbol{\sigma} \in I$ and 0 otherwise. This expression involves a reference density p, which is fixed in advance. Unlike the wave function density $|\psi_{\boldsymbol{\theta}}|^2$, the reference density p is not adjusted during training. Samples can therefore be drawn from p once, prior to training.

The choice of p is crucial and, perhaps, counterintuitive. Instead of increasing sampling in the physical $\{\boldsymbol{\sigma} : s(\boldsymbol{\sigma}) < c\}$ region, p ideally generates a large number of samples in the *unphysical* $\{\boldsymbol{\sigma} : s(\boldsymbol{\sigma}) \geqslant c\}$ region in which $|\psi_{\boldsymbol{\theta}}|^2$ is meant to be small. An appropriate density p allows the spurious mode functional L_{CV} (13) to detect and penalize an emerging spurious mode. We describe the particular choice of p used in our experiments in Appendix C.

The spurious mode functional (13) can be viewed as rewarding the concentration of wave function mass in the physical region $\{\boldsymbol{\sigma} : s(\boldsymbol{\sigma}) < c\}$, or equivalently as penalizing the accumulation of mass outside of this region. The CV-VMC objective function is the sum of the ordinary energy functional and the new spurious mode functional, and the optimization problem becomes

$$\arg\min_{\boldsymbol{\theta}} \left\{ \frac{\langle\psi_{\boldsymbol{\theta}}, H\psi_{\boldsymbol{\theta}}\rangle}{\langle\psi_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}}\rangle} + L_{\mathrm{CV}}(\boldsymbol{\theta}) \right\}. \tag{14}$$

We optimize this new objective via SR in which the parameter update $\boldsymbol{\delta}$ is derived by minimizing the cost function

$$\frac{\langle\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}, H\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}\rangle}{\langle\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}, \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}\rangle} + L_{\mathrm{CV}}(\boldsymbol{\theta}+\boldsymbol{\delta}) - \frac{1}{\epsilon}\left(\frac{|\langle\psi_{\boldsymbol{\theta}}, \psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}\rangle|}{\|\psi_{\boldsymbol{\theta}}\|\,\|\psi_{\boldsymbol{\theta}+\boldsymbol{\delta}}\|}\right)^2. \tag{15}$$

The Wirtinger derivative [26] of $L_{\mathrm{CV}}(\boldsymbol{\theta})$ is

$$\frac{\partial}{\partial\bar{\boldsymbol{\theta}}} L_{\mathrm{CV}}(\boldsymbol{\theta}) = -\Delta\mathrm{Cov}_{\boldsymbol{\sigma}\sim p}\left[\frac{\partial_{\boldsymbol{\theta}_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \mathbb{1}_{s(\boldsymbol{\sigma})<c}\right], \tag{16}$$

which yields the following algorithm.

Algorithm 2 (CV-VMC via SR). Choose the parameter update $\boldsymbol{\delta}$ to solve

$$(\boldsymbol{S} + \eta\boldsymbol{I})\boldsymbol{\delta} = \epsilon(\Delta\tilde{\boldsymbol{g}} - \boldsymbol{g}). \tag{17}$$

Here, $\eta \geqslant 0$ is a nonnegative parameter chosen to make $\boldsymbol{S} + \eta\boldsymbol{I}$ positive definite. The gradients $\tilde{\boldsymbol{g}}$ and $\boldsymbol{g}$ and overlap matrix

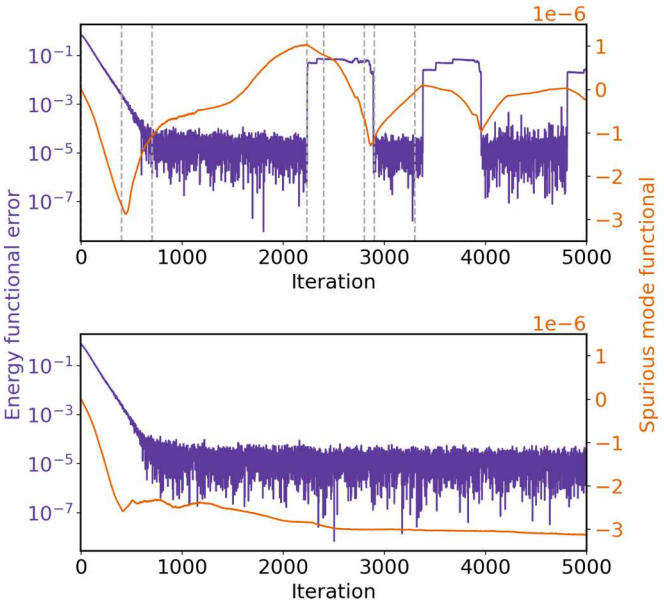

FIG. 9. (Top) Per-site energy functional error (purple) and per-site spurious mode functional (orange) during the same VMC training run depicted in Fig. 5. The iterations associated with panels in Fig. 5 are indicated by vertical dashed lines. (Bottom) Same quantities during CV-VMC training with the same random seed.

$\boldsymbol{S}$ are defined by

$$\begin{aligned}
\tilde{\boldsymbol{g}}_i &= \mathrm{Cov}_{\boldsymbol{\sigma}\sim p}\left[\frac{\partial_{\boldsymbol{\theta}_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \mathbb{1}_{s(\boldsymbol{\sigma})<c}\right], \\
\boldsymbol{g}_i &= \mathrm{Cov}_{\boldsymbol{\sigma}\sim|\psi|^2}\left[\frac{\partial_{\boldsymbol{\theta}_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \frac{H\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}\right], \\
\boldsymbol{S}_{ij} &= \mathrm{Cov}_{\boldsymbol{\sigma}\sim|\psi|^2}\left[\frac{\partial_{\boldsymbol{\theta}_i}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}, \frac{\partial_{\boldsymbol{\theta}_j}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}{\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})}\right].
\end{aligned} \tag{18}$$

Relative to the SR implementation of ordinary VMC, the only difference is the replacement of the gradient $\boldsymbol{g} \leftarrow \boldsymbol{g} - \Delta\tilde{\boldsymbol{g}}$.

To validate the success of the spurious mode functional in detecting spurious modes, in the top panel of Fig. 9, we plot both the energy functional and the spurious mode functional over the course of the *same* VMC (*not* CV-VMC) training run depicted in Fig. 5. Evidently the spurious mode functional tracks the formation of a spurious mode even before the energy spikes appear in the energy estimate. In the bottom panel of Fig. 9, we plot the same quantities over the course of a CV-VMC run, noting that CV-VMC controls the spurious mode functional throughout the training.

B. Numerical results

Until commented otherwise, the results of this section concern Heisenberg spin chains with $N = 100$ spins, and estimates are computed with $n = 100$ independent walkers. Figure 10 presents results comparing VMC and CV-VMC. The energy errors are nearly the same for the two optimization approaches except that VMC exhibits a large spike during iterations 4100–4500 (top panel). Additionally, even before VMC exhibits an energy spike, there is a spurious mode in

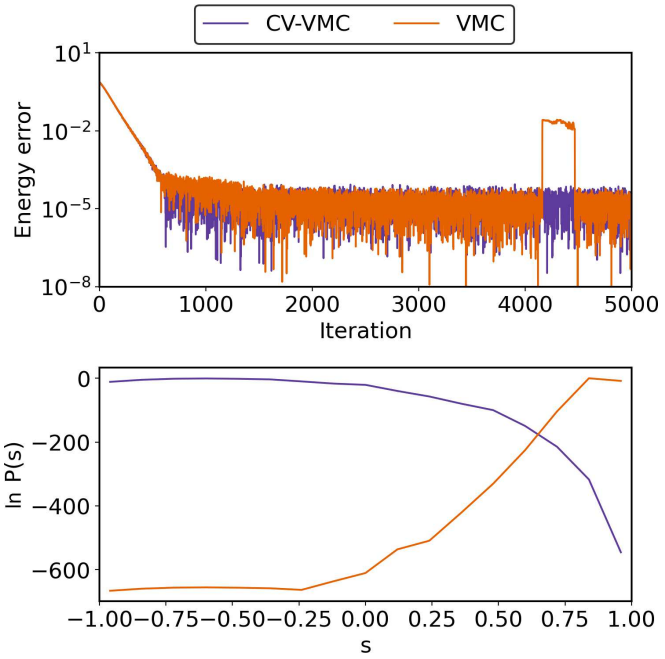

FIG. 10. (Top) Per-site energy errors for VMC and CV-VMC. (Bottom) Marginal density $P(s)$ of CV for the VMC and CV-VMC wave functions at iteration 3000.

the VMC wave function (bottom panel). In contrast, CV-VMC does not lead to any energy spikes, and the CV-VMC wave function is physically reasonable, quickly losing probability mass as s increases.

To produce the data in Fig. 10, we have chosen the reference density p to be stratified across a wide range of s values (see Appendix C) so that the spurious modes can be readily identified from data. To speed up the computations, we have prepared a large pool of 5×10^5 samples before beginning the VMC optimization and then subsampled 2000 randomly chosen configurations to determine each parameter update.

Although in principle VMC with perfect sampling converges to a local minimum of the energy, CV-VMC converges to a different fixed point, due to the additional gradient term $\Delta\tilde{\boldsymbol{g}}$. To evaluate the impact of this change, we show energy errors for different choices of Δ and c in Table II. For comparison, the average energy error for 11 cherry-picked standard VMC runs that happen to avoid energy spikes is 6.7×10^{-6}.

Let us discuss the impact of varying c. We first observe that the penalization becomes ineffective for the extreme parameter choices $c=-1$ and $c=1$. Indeed, we can rewrite $\Delta\tilde{\boldsymbol{g}}$ as

$$\Delta\tilde{\boldsymbol{g}} = \Delta\mathbb{E}_{\boldsymbol{\sigma}\sim p}[\mathbb{1}_{s(\boldsymbol{\sigma})<c}](\mathbb{E}_{\boldsymbol{\sigma}\sim p}[\overline{\partial_{\boldsymbol{\theta}}\ln\psi_{\boldsymbol{\theta}}}\,|\,s(\boldsymbol{\sigma})<c] - \mathbb{E}_{\boldsymbol{\sigma}\sim p}[\overline{\partial_{\boldsymbol{\theta}}\ln\psi_{\boldsymbol{\theta}}}]). \tag{19}$$

When $c\to-1$, $\mathbb{E}_{\boldsymbol{\sigma}\sim p}[\mathbb{1}_{s(\boldsymbol{\sigma})<c}]\to0$ because the CV cannot take values less than -1. Meanwhile, when $c\to1$, we have

$$\mathbb{E}_{\boldsymbol{\sigma}\sim p}[\overline{\partial_{\boldsymbol{\theta}}\ln\psi_{\boldsymbol{\theta}}}\,|\,s(\boldsymbol{\sigma})<c] - \mathbb{E}_{\boldsymbol{\sigma}\sim p}[\overline{\partial_{\boldsymbol{\theta}}\ln\psi_{\boldsymbol{\theta}}}]\to0 \tag{20}$$

because the CV cannot take values greater than 1. In both cases, Eq. (19) implies that $\tilde{\boldsymbol{g}}\to0$.

TABLE II. Mean per-site energy error of 20 independent runs for different Δ and c. For each run, the energy is estimated by taking the average of sampled local energies of all walkers in iterations 1000–5000. In the left table, $\Delta\equiv5.0\times10^{-6}$. In the right table, $c\equiv0$.

c	Energy error
-0.2	8.1×10^{-6}
0.0	7.9×10^{-6}
0.2	1.0×10^{-5}
0.4	8.8×10^{-6}
Δ	Energy error
2.5×10^{-6}	7.5×10^{-6}
5.0×10^{-6}	7.9×10^{-6}
1.0×10^{-5}	1.2×10^{-5}
2.0×10^{-5}	3.8×10^{-5}

In our numerical experiments, we tested the values $c=-0.8,-0.6,\ldots,0.8$, fixing $\Delta=5\times10^{-6}$ throughout. For values $c\leqslant-0.4$ and $c\geqslant0.8$, we observe the formation of spurious modes. When $c=-0.6$, for example, we find that one in 20 independent runs develops a spurious mode and exhibits energy spikes. In contrast, when $-0.2\leqslant c\leqslant0.4$, we plot the marginal density $P(s)$ of the CV in Fig. 11, and there is no evidence of a spurious mode developing. Meanwhile, Table II confirms that the energy error of CV-VMC is robust to the choice of c for $c=-0.2, 0.0, 0.2, 0.4$, i.e., in the range where spurious modes are avoided. From this comparison, we find that CV-VMC is effective for a range of c values, and we can set c as an approximate threshold beyond which the true ground state assigns a negligible amount of probability mass.

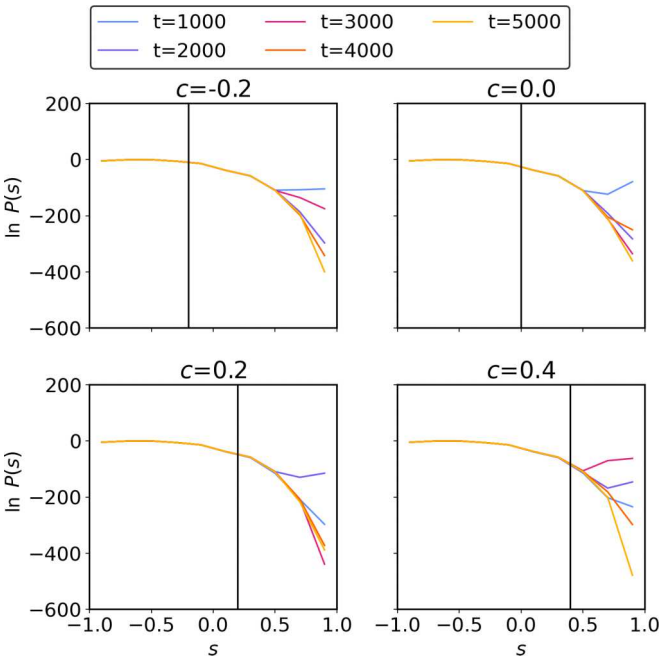

FIG. 11. CV-VMC results with $\Delta=5\times10^{-6}$ and different cutoff values c as indicated by black vertical lines.

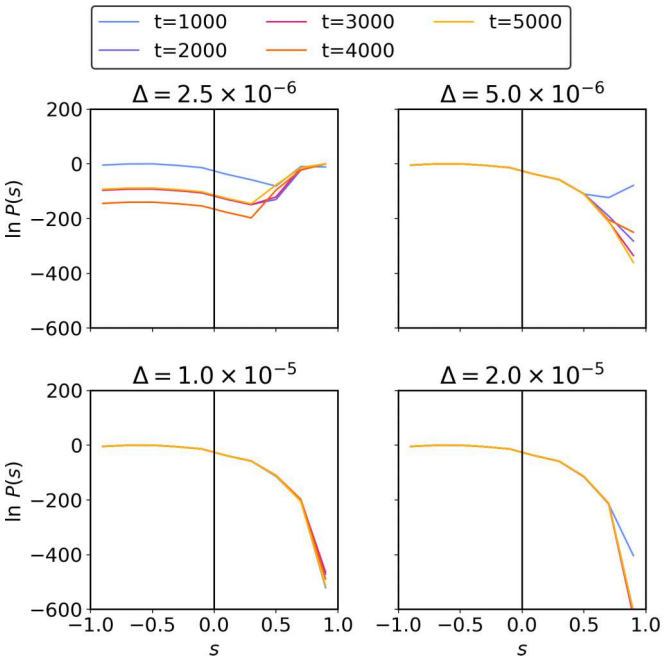

FIG. 12. The CV-VMC scheme with $c = 0$ and increasing penalization strengths Δ. The colored lines shows the marginal density $P(s)$ at different times during the training.

Next, we discuss the impact of varying Δ for fixed $c = 0$. In Table II, none of the parameter choices $\Delta = 2.5\times10^{-6}$, 5.0×10^{-6}, 1.0×10^{-5}, and 2.0×10^{-5} happens to lead to energy spikes, though we find that for the smallest value $\Delta = 2.5\times10^{-6}$, an unexplored spurious mode does develop. For this smallest value of Δ, the energy estimate is close to that of the reference VMC energy, but the penalty is too weak to eliminate the spurious mode, as shown in Fig. 12. As the penalization parameter Δ increases, the energy estimate error increases (see Table II) and the spurious mode is eliminated. In our tests, we find that $\Delta = 5\times10^{-6}$ is large enough to reliably eliminate spurious modes but still small enough to preserve the quality of the energy estimate. In general we recommend choosing Δ sufficiently large to eliminate energy spikes in both training and testing, but no larger to avoid degrading the energy estimate. A general strategy is to select Δ so that the update terms $\|(\boldsymbol{S}+\eta\boldsymbol{I})^{-1}\Delta\tilde{\boldsymbol{g}}\|$ and $\|(\boldsymbol{S}+\eta\boldsymbol{I})^{-1}\boldsymbol{g}\|$ in (17) are balanced.

In Table III, we demonstrate that CV-VMC outperforms VMC if we do not allow for cherry-picking of spike-free runs. This conclusion holds even when we consider a larger system, consisting of $N = 200$ spins. The table shows energy estimate errors for both methods in the cases $N = 100$ and 200, averaged over several independent runs. Energy spikes contaminate the results for standard VMC, yielding results with per-site energy error on the order $\sim10^{-3}$.

TABLE III. VMC and CV-VMC per-site energy errors. Each number is obtained by averaged over 20 independent runs.

	$N = 100$	$N = 200$
VMC	4.970×10^{-3}	3.282×10^{-3}
CV-VMC	7.950×10^{-6}	1.390×10^{-5}

Last, we comment on the computational cost. In VMC, there are three main contributions to the computational cost: querying $\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ at each *Metropolis-Hastings step* and querying $(H\psi_{\boldsymbol{\theta}})(\boldsymbol{\sigma})/\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and $\partial_{\boldsymbol{\theta}}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})/\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ at each *subsampled data point*. In CV-VMC, there are two additional costs: querying $\partial_{\boldsymbol{\theta}}\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})/\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ and $s(\boldsymbol{\sigma})$ at each *recorded sample*. Considering the complexity of the Hamiltonian (which often requires many wavefunction queries), we expect that in most cases the additional cost to evaluate $\tilde{\boldsymbol{g}}$ will be small compared to the computation of $\boldsymbol{S}$ and $\boldsymbol{g}$. In our training, CV-VMC increases the computational cost relative to VMC by less than 2%.

V. CONCLUSION

Variational Monte Carlo is a powerful framework for calculating ground state wave functions of many-body systems, but it is limited by the potentially long autocorrelation time of Markov chain Monte Carlo sampling. Incomplete sampling of the configuration space can result in incorrect gradient estimation, local overfitting, and large, random energy spikes. These issues cannot always be addressed by simply increasing the length of the Markov chains used for estimation [8,9,19], or even by applying enhanced sampling approaches.

Modifying the objective function is a more promising strategy for improving VMC robustness. Here, we have proposed collective-variable-informed VMC (CV-VMC) as a specific approach for modifying the objective function to incoporate *a priori* intuitions about the wave function, particularly knowledge of order parameters. The CV-VMC approach has a negligible added cost relative to standard VMC and is applicable to any ground-state estimation problem in which an appropriate choice of collective variable is available.

The single-layer RBM ansatz has been widely applied and has enjoyed remarkable success when modeling complex wave functions in many-body problems [3,27,28]. Careful attention to its training robustness is therefore warranted. The success of CV-VMC in our tests indicates that the RBM is flexible enough to accurately approximate the wave function of the periodic Heisenberg spin chain with 100–200 spins. However, in the absence of any mitigation strategy, the flexibility of the RBM leads to the formation of a spurious mode in regions that are not constrained by data.

In the future, there should be further studies of the training robustness of other, potentially more flexible ansatzes that have appeared recently in VMC applications, such as the multi-layer RBM [29] and convolutional neural networks [30]. Any of these ansatzes can potentially lead to the formation of spurious modes, since they are targeting a physical wave function which is concentrated in a narrow region of configuration space. Outside of that region, the ansatzes will not be contrained by samples. When evaluating different ansatzes, we emphasize that a decrease in overfitting can either result from slow, incomplete optimization of the wave function or from the genuine elimination of spurious modes, and these phenomena need to be carefully distinguished. For help with eliminating the spurious modes, CV-VMC is a

general approach which can be built into any VMC optimization of any ansatz.

ACKNOWLEDGMENTS

This work was supported by NYU IT High Performance Computing resources, services, and staff expertise. H.Z. and J.W. acknowledge support from the Advanced Scientific Computing Research Program within the DOE Office of Science through Award No. DE-SC0020427. H.Z. was also supported by the National Science Foundation through Award No. DMS-1913129. R.J.W. was supported by the Office of Naval Research through BRC Award No. N00014-18-1-2363 and the National Science Foundation through FRG Award No. 1952777, under the aegis of Joel A. Tropp. M.L. acknowledges the support of the National Science Foundation under Award No. 1903031 as well as his host institution for this fellowship, the Courant Institute of Mathematical Sciences, New York University. J.W. was also supported by the National Science Foundation through Award No. DMS-2054306.

APPENDIX A: PARALLEL TEMPERING

In parallel tempering (or replica exchange), one generates samples using several MCMC simulations, each targeting a distribution of the form $\pi_i \propto |\psi_{\boldsymbol{\theta}}|^{2/T_i}$. The constants T_i are commonly referred to as temperatures and the chain targeting $T_i = 1$ samples from ρ_θ. Periodically an exchange of states is proposed between samplers targeting neighboring temperatures and accepted or rejected according to the METROPOLIS criterion [23]. In Sec. III C, we use 9 temperatures 1, 1.4, 2, 3, 5, 10, 30, 500, and 20000, with 100 samplers targeting each of these temperatures. For a more complete discussion of parallel tempering in the current context, see Ref. [19]. The temperatures are set to maintain the swapping rate between neighboring temperatures between 30%–40% [31] (but the swapping rate between the last two ones are above 50%). Our conclusions are robust to the choice and number of temperatures used.

APPENDIX B: UMBRELLA SAMPLING

The eigenvector method for umbrella sampling (EMUS, [13]) is an enhanced sampling approach using a sequence of biasing functions $U_1, \ldots, U_L$ to produce low variance estimates of averages with respect to a given probability density $\pi(\boldsymbol{\sigma})$, i.e., estimate $\pi[g] := \Sigma_{\{\sigma_i\}} g(\boldsymbol{\sigma})\pi(\boldsymbol{\sigma})$. The steps of EMUS are as follows.

Algorithm 3 (EMUS). (1) Sample from the distribution

$$\pi_i(\boldsymbol{\sigma}) \propto \pi(\boldsymbol{\sigma}) U_i(\boldsymbol{\sigma}), \tag{B1}$$

for $i = 1, \ldots, L$.

(2) Initialize $\boldsymbol{u} = (\frac{1}{L} \quad \cdots \quad \frac{1}{L})$ and repeat the following steps until the vector $\boldsymbol{u}$ converges.

(a) Form the matrix $\mathbf{F} = \mathbf{F}(\boldsymbol{u})$ *via*

$$F_{ij} = \pi_i\left[\frac{U_j(\boldsymbol{\sigma})/u_j}{\sum_{k=1}^L U_k(\boldsymbol{\sigma})/u_k}\right]. \tag{B2}$$

(b) Solve the eigenvalue problem

$$\boldsymbol{w}^T = \boldsymbol{w}^T \mathbf{F}(\boldsymbol{u}), \quad \sum_{i=1}^L w_i = 1. \tag{B3}$$

(c) Set

$$u_i = \frac{w_i u_i}{\sum_{j=1}^L w_j u_j} \tag{B4}$$

for $i = 1, \ldots, L$.

(3) Compute averages $\pi[g]$ *as*

$$\pi[g] = \frac{\sum_{i=1}^L w_i \pi_i\left[\frac{g(\boldsymbol{\sigma})}{\sum U_k(\boldsymbol{\sigma})/u_k}\right]}{\sum_{i=1}^L w_i \pi_i\left[\frac{1}{\sum U_k(\boldsymbol{\sigma})/u_k}\right]}. \tag{B5}$$

We have used the symbol $\pi_i[\cdot]$ to denote averages with respect to the biased distributions π_i. In this paper, we apply the EMUS algorithm both (1) as one of the enhanced sampling methods of Sec. III C and (2) as a tool for analyzing the VMC wave function, specifically the marginal density $P(s)$.

We now discuss the details of applications (1) and (2).

(1) *Enhancing sampling.* In Sec. III C, we implement umbrella sampling with target distribution $\pi(\boldsymbol{\sigma}) \propto |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2$. We introduce $L = 16$ Gaussian biasing functions, defined by

$$U_i(\boldsymbol{\sigma}) = \exp\left(-\frac{1}{2}\left(\frac{s(\boldsymbol{\sigma}) - m_i}{\kappa}\right)^2\right) \tag{B6}$$

for $i = 2, \ldots, 15$,

$$U_1(\boldsymbol{\sigma}) = \begin{cases} 1, & s(\boldsymbol{\sigma}) \leqslant m_1 \\ \exp\left(-\frac{1}{2}\left(\frac{s(\boldsymbol{\sigma})-m_1}{\kappa}\right)^2\right), & s(\boldsymbol{\sigma}) > m_1 \end{cases}$$

and

$$U_{16}(\boldsymbol{\sigma}) = \begin{cases} \exp\left(-\frac{1}{2}\left(\frac{s(\boldsymbol{\sigma})-m_{16}}{\kappa}\right)^2\right), & s(\boldsymbol{\sigma}) < m_{16} \\ 1, & s(\boldsymbol{\sigma}) \geqslant m_{16} \end{cases}$$

with width parameter $\kappa = 0.1$ and centers m_i given by -0.5, -0.35, -0.15, 0.05, 0.25, 0.45, 0.65, 0.85, 1.05, 1.3, 1.6, 1.9, 2.3, 2.7, 3.1, and 3.5 for $i = 1, \ldots, 16$, respectively. The centers are chosen to spread the MCMC samples across a broad range of s values, while ensuring overlap between the MCMC samples in neighboring windows. Replica exchange moves between samples in neighboring windows are proposed every 2×10^3 Metropolis steps [32]. The swapping rates for these moves are 30%–40%, indicating sufficient overlap between windows.

(2) *Computing the marginal density* $P(s)$. Taking

$$g_i(\boldsymbol{\sigma}) = |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2 \mathbb{1}_{s(\boldsymbol{\sigma})\in I_i} \tag{B7}$$

for an interval of CV values I_i and π to be the uniform distribution on spin configurations $\boldsymbol{\sigma}$ with an equal number of $+1$ and -1 spins, we use EMUS to estimate

$$\pi[g_i] = \sum_{\{\sigma_j\}}^{s(\boldsymbol{\sigma})\in I_i} |\psi_{\boldsymbol{\theta}}(\boldsymbol{\sigma})|^2 \tag{B8}$$

The CV marginal probability densities plotted in Fig. 5 in Sec. III A, Fig. 8 in Sec. III C, and Figs. 10–12 in Sec. IV B are estimated using this approach, where we define the intervals I_i for $i = 1, \ldots, 17$ via

$$I_i = (-1.14 + 0.12i, -1.02 + 0.12i]. \tag{B9}$$

For these experiments, we use $L = 26$ Gaussian biasing functions of the same form as Eq. (B6) but with width $\kappa = 0.04$ and centers

$$m_i = -1.08 + 0.08i, \quad i = 1, \ldots, 26. \tag{B10}$$

APPENDIX C: CV-VMC SAMPLING

We use Gaussian biasing functions of the same form as Eq. (B6) when preparing the pool of samples for the CV-VMC scheme in Sec. IV B. To spread the samples to roughly evenly cover the range of possible CV values, we use a greater number ($L = 52$) of Gaussian biasing functions, defined by width $\kappa = 0.04$ and centers

$$m_i = -1.06 + 0.04i, \quad i = 1, \ldots, 52. \tag{C1}$$

These closely spaced biasing functions ensure that MCMC samples are evenly distributed across all values of s, as verified in Fig. 13. The reference probability density is chosen to be

$$p(\boldsymbol{\sigma}) = \frac{1}{\sum_{i,\boldsymbol{\sigma}'} U_i(\boldsymbol{\sigma}')} \sum_i U_i(\boldsymbol{\sigma}). \tag{C2}$$

In each window, 10 samplers start from uniformly distributed configurations within the subspace. We run 2×10^6 Metropolis steps and collect one sample every 2×10^3 steps. In this way, we build a pool consisting of 5.2×10^5 samples from p.

APPENDIX D: ROBUSTNESS TEST INITIALIZATION

When training is carried out using standard METROPOLIS sampling or parallel tempering, the samples generated are distributed (approximately) according to $\rho_{\boldsymbol{\theta}} \propto |\psi_{\boldsymbol{\theta}}|^2$. However, when umbrella sampling is used to estimate averages with respect to $\pi = \rho_{\boldsymbol{\theta}}$, Eq. (B5) in Appendix B shows that a sample at state $\boldsymbol{\sigma}$ drawn from the restrained distribution π_i weighted by

$$\frac{w_i}{\sum_k U_k(\boldsymbol{\sigma})/u_k}.$$

Because of these weights, the samples used in training must be resampled to generate an unweighted set of samples for robustness test initialization (Sec. III B). Specifically, we select 100 of the final states of the MCMC samplers with replacement with probabilities proportional to $w_i/\sum_k (U_k(\boldsymbol{\sigma})/u_k)$, where $\boldsymbol{\sigma}$ is a state from the ith umbrella sampling window.

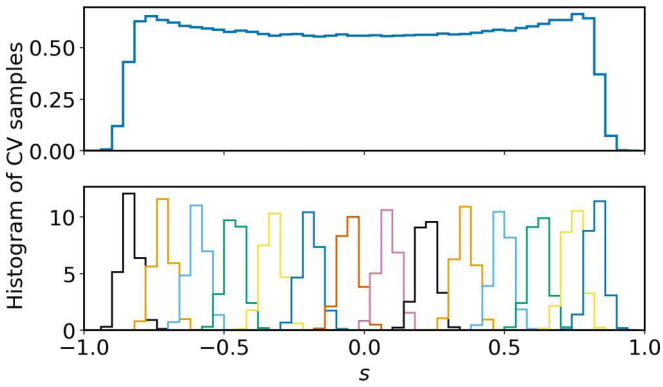

FIG. 13. (Top) Histogram of samples used in the CV-VMC scheme. (Bottom) Histograms of MCMC samples in 14 representative windows (out of 52 total).

[1] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods: Algorithms for Lattice Models* (Cambridge University Press, 2016).

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

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

[4] D. Luo and B. K. Clark, Backflow Transformations via Neural Networks for Quantum Many-Body Wave Functions, Phys. Rev. Lett. **122**, 226401 (2019).

[5] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, and W. M. C. Foulkes, *Ab initio* solution of the many-electron Schrödinger equation with deep neural networks, Phys. Rev. Res. **2**, 033429 (2020).

[6] J. Hermann, Z. Schätzle, and F. Noé, Deep-neural-network solution of the electronic Schrödinger equation, Nat. Chem. **12**, 891 (2020).

[7] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signal Systems **2**, 303 (1989).

[8] L. Yang, W. Hu, and L. Li, Scalable variational Monte Carlo with graph neural ansatz (2020).

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

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) http://www.deeplearningbook.org.

[11] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, Generalization properties of neural network approximations to frustrated magnet ground states, Nat. Commun. **11**, 1593 (2020).

[12] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo Simulation of Spin-Glasses, Phys. Rev. Lett. **57**, 2607 (1986).

[13] A. R. Dinner, E. H. Thiede, B. V. Koten, and J. Weare, Stratification as a general variance reduction method for Markov chain Monte Carlo, SIAM/ASA J. Uncert. Quantif. **8**, 1139 (2020).

[14] E. M. Boczko and C. L. Brooks, First-principles calculation of the folding free energy of a three-helix bundle protein, Science **269**, 393 (1995).

[15] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, Tempered Markov chain Monte Carlo for training of restricted Boltzmann machines, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 9, edited by Y. W. Teh and M. Titterington (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010), pp. 145–152.

[16] C. Pangali, M. Rao, and B. J. Berne, A Monte Carlo simulation of the hydrophobic interaction, J. Chem. Phys. **71**, 2975 (1979).

[17] S. Bernèche and B. Roux, Energetics of ion conduction through the k+ channel, Nature (London) **414**, 73 (2001).

[18] T. Tieleman, Training restricted Boltzmann machines using approximations to the likelihood gradient, in *Proceedings of the 25th International Conference on Machine Learning*, ICML '08 (ACM Press, New York, NY, 2008), pp. 1064–1071.

[19] R. J. Webber and M. Lindsey, Rayleigh-Gauss-Newton optimization with enhanced sampling for variational Monte Carlo, Phys. Rev. Res. **4**, 033099 (2022).

[20] H. Bethe, Zur theorie der metalle, Z. Phys. **71**, 205 (1931).

[21] T. Misawa, S. Morita, K. Yoshimi, M. Kawamura, Y. Motoyama, K. Ido, T. Ohgoe, M. Imada, and T. Kato, mVMCOpen-source software for many-variable variational Monte Carlo method, Comput. Phys. Commun. **235**, 447 (2019).

[22] R. J. Needs, M. D. Towler, N. D. Drummond, and P. L. Ríos, Continuum variational and diffusion quantum Monte Carlo calculations, J. Phys.: Condens. Matter **22**, 023201 (2010).

[23] D. J. Earl and M. W. Deem, Parallel tempering: Theory, applications, and new perspectives, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[24] J. Kästner, Umbrella sampling, WIREs Comput. Mole. Sci. **1**, 932 (2011).

[25] E. H. Thiede, B. Van Koten, J. Weare, and A. R. Dinner, Eigenvector method for umbrella sampling enables error analysis, J. Chem. Phys. **145**, 084115 (2016).

[26] W. Wirtinger, Zur formalen theorie der funktionen von mehr komplexen veränderlichen, Math. Ann. **97**, 357 (1927).

[27] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann Machine, Phys. Rev. X **8**, 021050 (2018).

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

[29] H. He, Y. Zheng, B. A. Bernevig, and G. Sierra, Multi-layer restricted Boltzmann machine representation of 1D quantum many-body wave functions, arXiv:1910.13454.

[30] L. Yang, Z. Leng, G. Yu, A. Patel, W.-J. Hu, and H. Pu, Deep learning-enhanced variational monte carlo method for quantum many-body physics, Phys. Rev. Res. **2**, 012039(R) (2020).

[31] C. Predescu, M. Predescu, and C. V. Ciobanu, On the efficiency of exchange in parallel tempering monte carlo simulations, J. Phys. Chem. B **109**, 4189 (2005).

[32] C. Matthews, J. Weare, A. Kravtsov, and E. Jennings, Umbrella sampling: A powerful method to sample tails of distributions, Mon. Not. R. Astron. Soc. **480**, 4069 (2018).