Kinetic samplers for neural quantum states

Andrey A. Bagrov,[1,2,*] Askar A. Iliasov,[3,4,†] and Tom Westerhout[3,‡]

[1]*Department of Physics and Astronomy, Uppsala University, Box 516, SE-75120 Uppsala, Sweden*

[2]*Theoretical Physics and Applied Mathematics Department, Ural Federal University, 620002 Yekaterinburg, Russia*

[3]*Institute for Molecules and Materials, Radboud University, Heyendaalseweg 135, 6525AJ Nijmegen, Netherlands*

[4]*Space Research Institute of the Russian Academy of Science, Moscow, 117997, Russia*

(Received 25 November 2020; revised 16 August 2021; accepted 16 August 2021; published 7 September 2021)

Neural quantum states are a recently introduced class of variational many-body wave functions that are very flexible in approximating diverse quantum states. Optimization of an NQS ansatz requires sampling from the corresponding probability distribution defined by squared wave function amplitude. For this purpose, we propose to use kinetic sampling protocols and demonstrate that in many important cases such methods lead to much smaller autocorrelation times than the Metropolis-Hastings sampling algorithm while still allowing to easily implement lattice symmetries (unlike autoregressive models). We also use uniform manifold approximation and projection algorithm to construct two-dimensional isometric embedding of Markov chains and show that kinetic sampling helps attain a more homogeneous and ergodic coverage of the Hilbert space basis.

DOI: 10.1103/PhysRevB.104.104407

The concept of neural quantum states (NQS) emerged several years ago, when it was suggested that variational wave functions possessing a structure of simple neural networks—restricted Boltzmann machines—can be efficiently optimized to approximate ground states of some many-body quantum systems [1]. The idea of using an ansatz of that type turned out to be very appealing because of neural networks' flexibility in representing data: Instead of constructing a very specific trial function that accounts for physical properties of the concrete model of interest [2], one could hope to get away with a universal neural approximator [3] that can automatically adjust itself over the course of learning and approach the ground state of *any* local Hamiltonian. However, soon it became clear that fermionic systems [4] (away from the neutrality point) and frustrated quantum magnets are challenging for the NQS approach [5,6], just as they are for the more traditional and established methods [7]. This posed a natural quest for improving upon this approach and bringing it closer to the point when it can be successfully applied to studying such models. Since then, the method of NQS has evolved into a solid framework embracing a number of optimization schemes and variational ansätze (going far beyond the originally proposed shallow Boltzmann machines) [8–10], and considerable progress has been made in understanding both strong points and shortcomings of neural network wave functions [11]. On the positive side, it was realized that even the simplest NQS could host volume law entanglement [12,13] and, in fact, have great capacity to neatly express a vast variety of many-body states, including ground states of frustrated spin Hamiltonians [6]. For instance, choosing a suitable NQS architecture combining the flexibility of a neural network with some prior knowledge about the model allowed to attain high accuracy in solving the $J_1 - J_2$ Heisenberg antiferromagnet on a square lattice and reveal the Dirac nodal nature [14,15] of its spin liquid phase [16,17]. Some of the reasons why NQS could not be blindly applied to highly frustrated systems have been identified as well. The progress made in the field encourages further improvement of the method to make it suitable for studying many-body systems that are currently beyond its scope of applicability.

An important aspect of all the NQS optimization algorithms is Monte Carlo sampling. Since neural network architectures are not amenable to full contraction, computing loss functions (energies, fidelities) require sampling from the probability distribution defined on the Hilbert space basis by the wave function amplitudes. At this point, exceptionally high expressibility of NQS, while being a clear advantage of the method, turns out to hold a hidden danger. During the learning procedure, NQS undergo a sequence of weight updates, and the corresponding probability distribution evolves in a highly nontrivial way. It could easily happen that the distribution acquires a form which is problematic to sampling by means of Monte Carlo techniques. For example, if the distribution constitutes a number of well-separated narrow peaks on the set of basis vectors, inaccurate sampling could lead to ergodicity problems, incorrect estimates of the distribution, and, as a result, the NQS following a wrong direction on the optimization landscape.

*andrey.bagrov@physics.uu.se

†a.iliasov@science.ru.nl

‡tom.westerhout@ru.nl

Perhaps, the most promising way to overcome the nonergodicity issue is to employ a certain class of neural architectures called generative models [18], as was recently suggested. The most well-known example of generative models are autoregressive models [19]. These models are constructed to represent probability distributions as products of conditional probabilities. In the context of finite-dimensional lattice quantum models, the conditional probabilities have the meaning of probabilities for a subset of degrees of freedom, e.g., spins, to be in a certain classical state given the state of the complementing degrees of freedom fixed. Such representation allows to sample from the distribution *exactly* without resorting to Markov chain Monte Carlo (MCMC) techniques [20]. The downside is that implementation of symmetries becomes problematic. So far, only the basic constraints such as the fixed total magnetization [21,22] and translation invariance [23] have been formulated within the framework of generative models.

Since using all the accessible symmetries, such as lattice symmetries, provides an essential advantage in studying many-body quantum systems [15], it is natural to ask whether there is an alternative way to bypass the problem of correlated samples generated with MCMC. In this paper, we propose an approach to sampling from NQS probability distributions based on the concept of continuous-time kinetic Monte Carlo. The idea behind it is to substitute the discrete chain of proposition-acceptance steps with a rejection-free process evolving in continuous time [24]. Although well-appreciated in many other domains of computational physics [25], it has not been used within the domain of machine learning for quantum simulations, and here we make a step in this direction. In particular, we focus on the minimal continuous-in-time sampling algorithm which we shall call Zanella process following [26]. We consider several classes of many-body quantum states such as exact ground states of frustrated systems of up to 36 spins ($J_1 - J_2$ Heisenberg antiferromagnet on a square lattice and the nearest-neighbor Heisenberg antiferromagnet on a kagome lattice) and neural quantum representations obtained during ground-state optimization for the same models. For each state, we assess the quality of sampling protocols. The two gauges we use are autocorrelation time and the visualized coverage of configuration space constructed with the uniform manifold approximation and projection (UMAP) dimension reduction algorithm [27].

The paper is organized as follows. In Sec. I, we outline the implementation of lattice symmetries. In Sec. II, we provide a pedagogical introduction to Zanella process closely following Ref. [26]. Section III contains the main results regarding the use of sampling protocols for different quantum states. We conclude with Sec. IV.

I. IMPLEMENTATION OF LATTICE SYMMETRIES

In the context of NQS, the conventional way to take into account lattice symmetries is to impose the corresponding constraints on the architecture of neural networks [1,5]. Although it is rather straightforward to implement translation invariance of a chain or a square lattice in this way, for more general crystal symmetries, this approach becomes problematic. To go beyond translation invariance, one can average the output of the neural network: For a symmetry group $\mathcal{G}$, replace $\psi(\sigma)$ by $\sum_{g\in\mathcal{G}} \psi(\sigma)$ [28]. The main drawback of this approach is the increase in computational resources: One now has to propagate many more spin configurations through the neural network. Recently, group convolutional neural networks have been applied to quantum systems [29]. They allow to also encode nontrivial symmetries (such as rotation) directly into the architecture. However, the implicit assumption in all these methods is that no extra phase factors accumulate from application of symmetry operators; in other words, cases such as nonzero momentum cannot be treated easily.

If arbitrary quantum numbers are desired, one can resort to operating within a symmetry-adapted Hilbert space basis, which is often used in exact diagonalization [30,31] (see also Refs. [32,33]). This approach was employed in Ref. [34], and in contrast to averaging, it does not increase the computational complexity. In our paper, we also adopt this approach and for completeness outline it here. We work under the assumption that the symmetry group has at least one one-dimensional irreducible representation, and the state of interest which one is sampling from can be expressed as a combination of basis vectors from one of these representations.

Let $\mathcal{H}$ be the Hamiltonian and A be the finite symmetry group generated by the lattice *symmetry operators* $\{T_k\}$, i.e., operators which commute with the Hamiltonian: $[\mathcal{H}, T_k] = 0$. In σ^z product state basis, spin configurations are represented as binary sequences $|\sigma\rangle = |\sigma_1\sigma_2\ldots\sigma_n\rangle$, $\sigma_i = 0, 1$. To define the symmetry-adapted basis, we first introduce equivalence classes of basis spin configurations under the group action as orbits $\texttt{orbit}(|\sigma\rangle) = \{g|\sigma\rangle | g \in A\}$. Every orbit is then represented by the basis state $|\tilde{\sigma}\rangle \in \texttt{orbit}(|\sigma\rangle)$ which has the minimal value when viewed as a binary representation of an integer number: $\texttt{representative}(|\sigma\rangle) = |\tilde{\sigma}\rangle = \min_{\text{int}} \texttt{orbit}(\sigma)$. For example, orbit of basis vector $|\sigma\rangle = |{\downarrow}{\downarrow}{\uparrow}{\uparrow}\rangle \simeq \{0011\} = 2^2 + 2^3 = 12$ in a periodic four-spin chain would be represented by $|\tilde{\sigma}\rangle = |{\uparrow}{\uparrow}{\downarrow}{\downarrow}\rangle \simeq \{1100\} = 2^0 + 2^1 = 3$.

To build the one-dimensional representation, we note that, since A is finite, for every symmetry generator T_k, there is a n_k such that $T_k^{n_k} = \mathbb{1}$, which is typically quite small (at most of the order the of system size). Hence, eigenvalues of each symmetry generator are roots of 1, and $T_k|0\rangle = \lambda_k|0\rangle$, $|\lambda_k| = 1$, where $|0\rangle$ is the ground state of $\mathcal{H}$. Thus for any $g \in A$, one can write $g|0\rangle = \lambda_g|0\rangle$, and $\lambda_{gh} = \lambda_g\lambda_h$ for $g, h \in A$, which determines the one-dimensional irreducible representation of the symmetry group.

Importantly, even if A itself is a non-Abelian group, this construction is valid as long as its representation is Abelian. Although generally $[T_i, T_j] \neq 0$, on the ground state (as well as any other state $|\sigma\rangle$ belonging to this representation): $[T_i, T_j]|\sigma\rangle = 0$. For example, for square lattice, translation by one lattice vector T_x and rotation by 90 degrees, $R_{\frac{\pi}{4}}$ do not commute, but $T_x|0\rangle = R_{\frac{\pi}{4}}|0\rangle = |0\rangle$.

For each $|\sigma\rangle$, there exists a $g \in A$ such that $\langle\sigma|0\rangle = \langle\tilde{\sigma}|g^\dagger|0\rangle = \lambda_g^*\langle\tilde{\sigma}|0\rangle$. Thus,

$$\langle\sigma|0\rangle \cdot |\sigma\rangle = \lambda_g^*\langle\tilde{\sigma}|0\rangle \cdot g|\tilde{\sigma}\rangle = \langle\tilde{\sigma}|0\rangle \cdot \lambda_g^* g|\tilde{\sigma}\rangle. \quad (1)$$

This means that the standard basis expansion of $|0\rangle$ can be rewritten as

$$|0\rangle = \sum_{\sigma} \langle\sigma|0\rangle \cdot |\sigma\rangle = \sum_{\tilde{\sigma}} \sum_{g\in A} \frac{N_{\tilde{\sigma}}}{|A|} \langle\tilde{\sigma}|0\rangle \cdot \lambda_g^* g|\tilde{\sigma}\rangle$$
$$= \frac{1}{|A|} \sum_{\tilde{\sigma}} N_{\tilde{\sigma}} \langle\tilde{\sigma}|0\rangle \cdot \sum_{g\in A} \lambda_g^* g|\tilde{\sigma}\rangle, \quad (2)$$

where $N_{\tilde{\sigma}} \in \mathbb{N}$ is the number of original basis elements in the orbit of $|\tilde{\sigma}\rangle$; $|A|$ denotes the number of elements in the symmetry group A; the sum over σ runs over all basis vectors of the Hilbert space, and the sum over $\tilde{\sigma}$ runs over representatives of all orbits. Using Eq. (2), we define a new basis,

$$|\mathcal{S}_{\tilde{\sigma}}\rangle = \frac{1}{\sqrt{N_{\tilde{\sigma}}}} \sum_{g\in A} \lambda_g^* \cdot g|\tilde{\sigma}\rangle, \quad (3)$$

where $1/\sqrt{N_{\tilde{\sigma}}}$ coefficient is introduced to ensure proper normalization.

We can redefine the Hamiltonian $\mathcal{H}$ in the new basis. Suppose that originally we had $\mathcal{H}|\sigma\rangle = \sum_i c_i|\sigma_i\rangle$. Then, in the symmetry-adapted basis, we get

$$\mathcal{H}|\mathcal{S}_{\tilde{\sigma}}\rangle = \sum_i c_i \frac{\sqrt{N_{\tilde{\sigma}_i}}}{\sqrt{N_{\tilde{\sigma}}}} \lambda_{h_i} \cdot |\mathcal{S}_{\tilde{\sigma}_i}\rangle. \quad (4)$$

One should keep in mind that when constructing the symmetry-adapted basis, we used the ground state $|0\rangle$ for illustrative purposes only to make sure that the representation we are dealing with includes $|0\rangle$. In a real-world scenario, one does not know the ground state beforehand, and in fact it is not required to find characters λ of the representation. If the group has several one-dimensional irreducible representations and it is unknown which one the ground state belongs to, the optimization problem should be solved separately in each of the corresponding symmetry-adapted bases.

II. KINETIC MONTE CARLO

The Metropolis-Hastings algorithm [35,36] is the most popular choice for Markov chain sampling from a probability distribution $\pi(x)$ defined on a discrete set of elements $\mathcal{X}$, such as the Hilbert space basis of a finite-dimensional quantum system. At every iteration, the state of the chain is given by some $x \in \mathcal{X}$. An element y from the vicinity ∂x of x is then suggested, and the sampling process either transitions to y or remains at x. The transition happens with probability $p = \min(1, \frac{\pi(y)}{\pi(x)})$. Which elements should be considered as belonging to ∂x depends on the problem, but for closed quantum spin systems with fixed magnetization, where every element is a product state $|\uparrow\downarrow\uparrow\ldots\downarrow\downarrow\rangle$, ∂x is often chosen to include elements that differ from x by a binary spin flip that preserves total magnetization.

If for some x it turns out that $\pi(x) \gg \pi(y)$ for the majority of $y \in \partial x$, the acceptance rate becomes very low and the sampling process gets stuck at x for many iterations. This negatively affects the quality of the sampled sequence, making it too correlated and not accurately representing the desired $\pi(x)$ distribution. If $\pi(x)$ has a number of far-separated peaks of this kind, or if elements of $\mathcal{X}$ tend to form clusters such that the Markov chain cannot leave them once entered, the sampling could lead to severely wrong results.

The Zanella process [24,26] is a natural way to bypass this problem by using a rejection-free scheme instead of the acceptance-rejection protocol. As before, at every step the sampling process is located at some $x_i \in \mathcal{X}$. Now, however, even if this point is a local maximum of the probability distribution and the acceptance rate in the Metropolis-Hastings algorithm would be very low, the process still jumps to an x_{i+1} from ∂x_i. To preserve the information about probability distribution, we introduce a waiting time. In other words, before jumping, the process sits at x_i for some time $\tau_i \in \mathbb{R}$ which depends on the probability ratio $\pi(x_{i+1})/\pi(x_i)$. Hence, instead of getting stuck at x_i for many steps, τ_i is set to a high value and the process moves on. To be more precise, the algorithm can be outlined as follows:

(1) At step i, compute normalization for the probability of jumping away from x_i (i.e., a decay rate),

$$\lambda_i = \sum_{y\in\partial x_i} g\left(\frac{\pi(y)}{\pi(x_i)}\right),$$

where g is a function obeying $g(l) = l \cdot g(1/l)$, usually called a *balancing function* [26].

(2) Estimate waiting time τ_i by sampling it from the exponential distribution

$$\tau_i \sim \mathrm{Exp}(\lambda_i),$$

where the probability density function of $\mathrm{Exp}(\lambda)$ is $f(x;\lambda) = \lambda e^{-\lambda x}$.

(3) Increase the overall running time:

$$t_{i+1} = t_i + \tau_i.$$

(4) Choose a state $y \in \partial x_i$ with probability

$$p(y) = \frac{1}{\lambda_i} \cdot g\left(\frac{\pi(y)}{\pi(x_i)}\right).$$

(5) Jump to $x_{i+1} = y$ and repeat the scheme.

In this formulation, one can think of the sequence $\{x_i\}$ of samples as if it were a piecewise constant function of time $x(t)$. Expectation value of a function defined on $\mathcal{X}$ can then be computed as follows:

$$E_\pi[f(x)] = \lim_{T\to\infty} \frac{1}{T} \int_0^T f(x(t))dt$$
$$= \lim_{N\to\infty} \sum_{k=0}^{N-1} \frac{t_{k+1} - t_k}{t_N} f(x_k)$$
$$= \lim_{N\to\infty} \frac{1}{N} \sum_{k=0}^{N-1} f(x(k\Delta t)), \quad (5)$$

where $\Delta t = t_N/N$. Balancing function g plays a role similar to the importance sampling in MCMC, but in this paper we consider the simplest case of $g = 1$.

As we will see, this simple algorithm already allows to drastically improve the ergodicity of sampling from problematic distributions.

III. KINETIC SAMPLING VERSUS METROPOLIS-HASTING

A. Assessment criteria

The standard way to estimate the quality of Monte Carlo sampling is to compute autocorrelation time τ_{corr} for some relevant quantity O. The autocorrelation time is the characteristic decay time (number of steps in the Markov chain) of the corresponding two-point correlation function [37]:

$$C_O(t) = \langle O(t)O(0)\rangle - \langle O(t)\rangle^2,$$

$$C_O(t)/C_O(0) \simeq e^{-t/\tau_{\text{corr}}}, \text{ for } t \gg 1.$$

In the context of sampling from probability distributions given by many-body wave functions, two natural choices of O are logarithmic probabilityln $|\psi(\mathcal{S})|^2$ and local energy estimator $E_{\text{loc}}(\mathcal{S})$ defined by the following equation:

$$E = \langle\psi|H|\psi\rangle = \sum_{\mathcal{S}} \frac{\langle\mathcal{S}|H|\psi\rangle}{\langle\mathcal{S}|\psi\rangle} \cdot |\langle\mathcal{S}|\psi\rangle|^2$$

$$\equiv \sum_{\mathcal{S}} E_{\text{loc}}(\mathcal{S}) \cdot |\langle\mathcal{S}|\psi\rangle|^2 \approx \sum_{\mathcal{S}\sim|\psi|^2} E_{\text{loc}}(\mathcal{S}).$$

Logarithmic probability is chosen instead of $|\psi(\mathcal{S})|^2$ to avoid numerical issues. In the following, we will use $C_\psi(t)$ to denote autocorrelation function computed for $\ln|\psi(\mathcal{S}(t))|^2$, and $C_E(t)$ — for $E_{\text{loc}}(\mathcal{S}(t))$.

Although autocorrelation time is a well-established criterion, it is not the only way to judge the quality of drawn samples. In Ref. [20], authors used the principle component analysis (PCA) algorithm [38] to reduce the dimension of basis vectors. By visualizing the resulting 2D vectors, they were able to compare the Metropolis-Hastings algorithm to the exact sampling procedure of autoregressive neural network architectures and demonstrate that the latter leads to much more homogeneous coverage of the Hilbert space basis. For systems considered in this work, PCA does not seem to reveal much additional information about the quality of samplers. Thus we suggest using a more involved but arguably also superior dimension reduction algorithm called UMAP [27]. We refer the reader to the original paper for a detailed motivation and description of the algorithm (especially Sec. 3), but would still like to briefly outline the procedure here.

UMAP algorithm operates on a discrete data set $\{x_i\}$ equipped with some metric $d(x_i, x_j)$ which measures dissimilarity between data points. In our case, the data set is a subset of the Hilbert space basis sampled using either the Metropolis-Hastings or Zanella algorithm (excluding all duplicates). The metric is simply the Hamming distance between the corresponding spin sequences $|\uparrow\downarrow\ldots\uparrow\uparrow\rangle$. The first stage of the algorithm is to equip the data set with a structure of a weighted undirected graph. One fixes the number of nearest neighbors k every vertex (basis vector) should be connected with. For our purpose, it is natural to choose $k \lesssim N$, where N is the number of spins in the system. Concretely, for 36-spin systems we take $k = 20$. In the resulting graph, each edge is assigned some weight w_{ij} which is a function of d and k. Once the graph is constructed, the UMAP algorithm projects it on a low-dimensional space (usually the 2D plane) in a way that approximately preserves the distances between the vertices such that the resulting visualization maximally accurately represents the actual metric structure of the data set. Embedding into a 2D space allows to directly compare the quality of different samplers by contrasting sampled sequences visually for different types of many-body quantum states.

To represent Monte Carlo samples, we adopt the following protocol:

(1) Using each of the two samplers (Metropolis-Hastings and Zanella), generate a sequence of 10^5 vectors.

(2) Merge these two sequences and discard all repetitions to obtain a set of unique basis vectors visited by either of the samplers.

(3) Equip this set with Hamming distance and build its UMAP embedding into the two-dimensional plane.

(4) Visualize the sequences within this embedding.

B. Benchmarks

First, we compare the Monte Carlo algorithms by sampling from the exact ground states of three quantum spin models: the $J_1 - J_2$ Heisenberg antiferromagnet on 6×6 square lattice with periodic boundary conditions at $J_2 = 0.0$

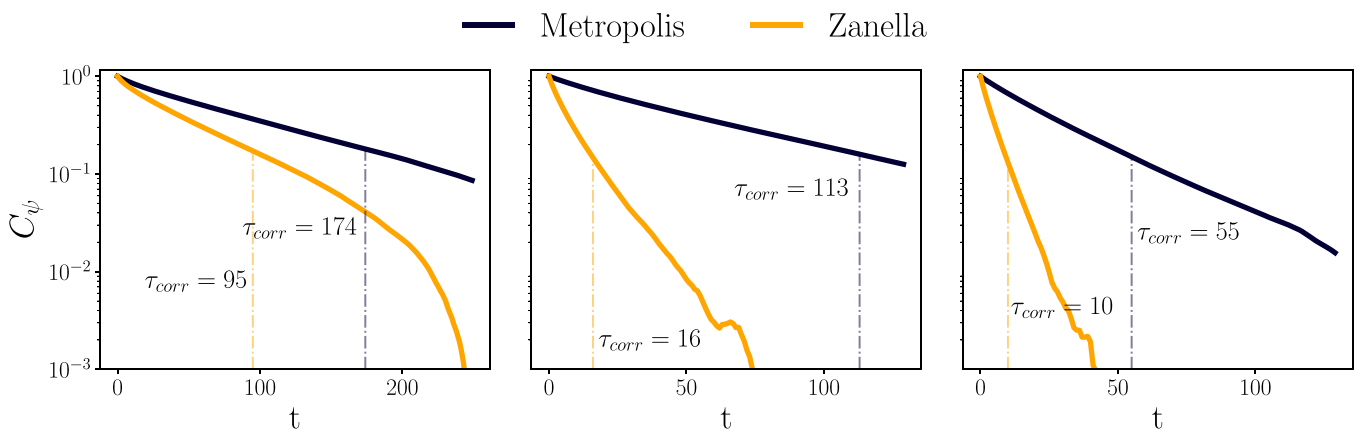

FIG. 1. Autocorrelation function C_ψ of Metropolis-Hastings and Zanella processes computed for the cases of sampling from probability distributions corresponding to ground states of Heisenberg antiferromagnet on square lattice at $J_2 = 0$ (left) and $J_2 = 0.55$ (middle) and kagome lattice (right). Autocorrelation function was computed by averaging 300 chains of length 8000

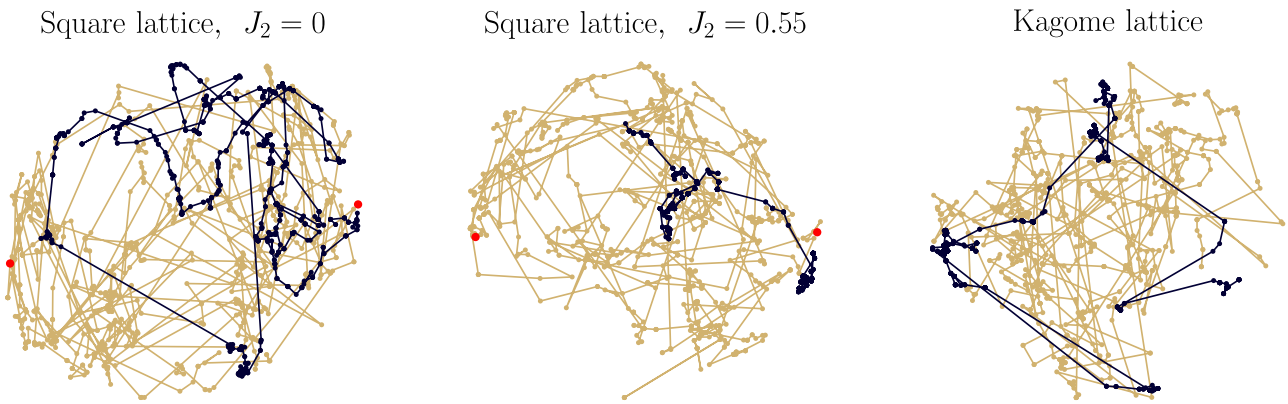

FIG. 2. UMAP visualization of sequences of basis elements produced by Metropolis-Hastings (dark blue) and Zanella (tan) samplers. The sampled probability distributions correspond to ground states of Heisenberg antiferromagnets on square lattice at $J_2 = 0$ and $J_2 = 0.55$ and kagome lattice. Every point represents a vector from the nonsymmetrized basis. Red points on the first two plots represent Neel states with checkerboard spin ordering (so the Hamming distance between the two Neel states is 36). For both samplers, 800 elements are shown.

(nonfrustrated) and $J_2 = 0.55$ (maximally frustrated), and Heisenberg antiferromagnet on 36-site kagome cluster with periodic boundary conditions obtained with exact diagonalization [39]. For exact ground states, energy autocorrelation functions are not well-defined because $E_{\mathrm{loc}}(\mathcal{S})$ are identical for all basis spin configurations $\mathcal{S}$. We thus only compute the probability autocorrelation functions C_ψ which are shown in Fig. 1. For all three systems, the Zanella process shows superior performance. In the ground states, it allows to reduce autocorrelation time by a factor of 2–7, and, as will be shown below; for more generic states encountered during NQS optimization the gain could be up to a factor of 10–50. We expect this effect to increase even more for larger systems.

Second, we analyze Zanella and Metropolis-Hastings algorithms using UMAP dimension reduction. Defining a metric on the symmetry-adapted basis is a nontrivial task and is left for future studies. Instead, we do the sampling in a nonsymmetrized basis. Dimension of the Hilbert space is thus $9075135300 = O(10^{10})$ for all three systems. Hamming distance acquires a very concrete physical interpretation—it counts the minimal number of steps an algorithm needs to move from one basis state to another. In Fig. 2, we show relatively short (800 steps) parts of Markov chains. They are taken from the middle of the chain to ensure that thermalization effects do not disrupt the picture. Visual distance in the figure approximately reflects the Hamming distance between points (of course, some aberrations caused by embedding into a lower-dimensional space are unavoidable). One can see that for all three considered wave functions, the kinetic sampler explores the Hilbert space in a much more ergodic and swift manner than the Metropolis-Hastings algorithm. Note also that for square lattice Metropolis-Hastings algorithm samples, there are more unique states and it covers a bigger part of the basis for $J_2 = 0$ than it does for $J_2 = 0.55$, even though its autocorrelation time is much larger in the former case. Without the UMAP algorithm, one would have ended up under the impression that the frustrated case was easier to sample.

Since we are mainly motivated by improving sampling in the context of NQS, it is instructive to compare quality of the methods in a realistic learning scenario. To perform this test, we run stochastic reconfiguration optimization [1,40]

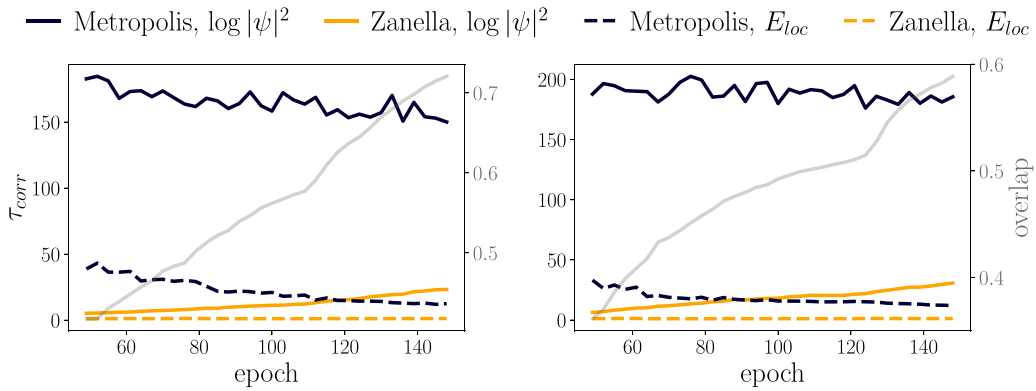

FIG. 3. Evolution of autocorrelation time during the NQS training procedure for Heisenberg antiferromagnet on 6×6 square lattice with $J_2 = 0$ (left) and $J_2 = 0.55$ (right). Grey curves represent overlap with the exact ground state, which is computed exactly at every learning epoch by evaluating sum over the complete Hilbert space basis. Autocorrelation times were estimated from 200 chains of length 7000. Upon approaching the ground state, the advantage of Zanella process becomes less significant, but over the course of learning it outperforms Metropolis-Hastings sampling by at least an order of magnitude.

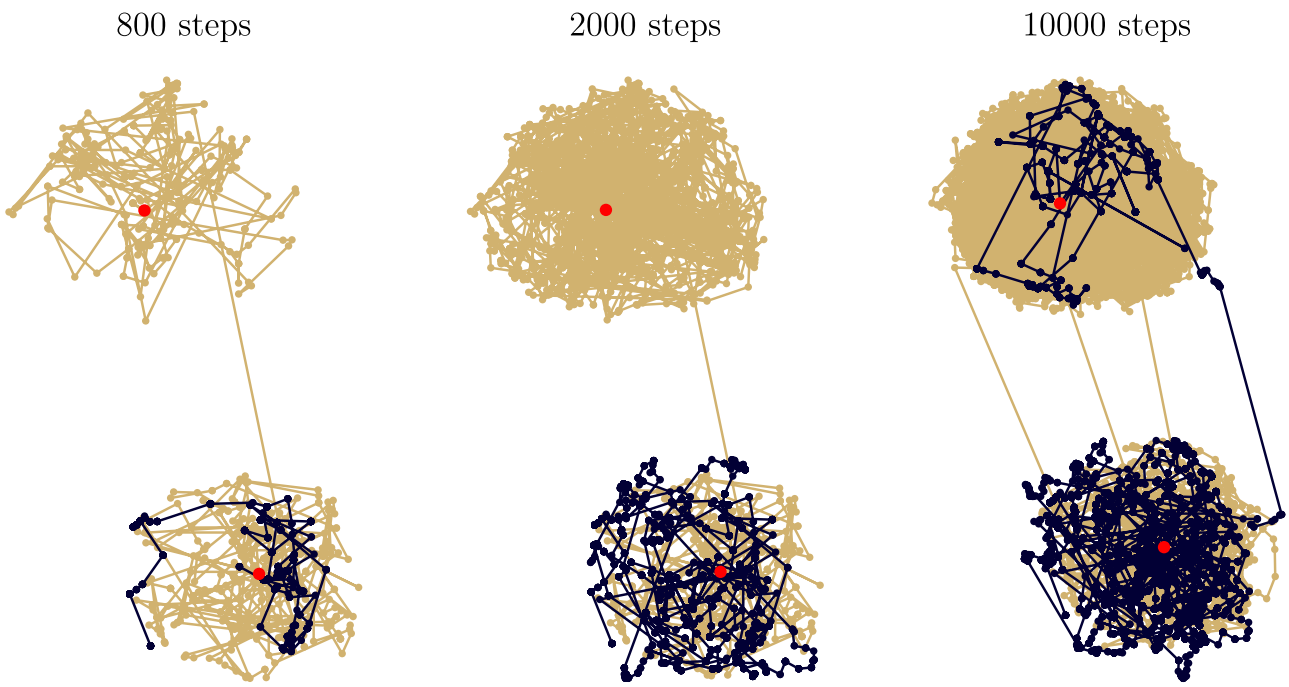

FIG. 4. UMAP visualization of sequences of basis elements produced by Metropolis-Hastings (dark blue) and Zanella (tan) samplers. The sampled probability distribution corresponds to a typical undertrained NQS (Heisenberg antiferromagnet on 6×6 square lattice with $J_2 = 0$, epoch №200). Every point represents a vector from the nonsymmetrized basis. Red points represent Neel states with checkerboard spin ordering (so the Hamming distance between the two Neel states is 36). As before, the parts of Markov chains are taken from the middle of the chain to ensure proper thermalization.

of simple neural networks in the symmetry-adapted basis, aiming at finding good approximations to ground states of the $J_1 - J_2$ model on a 6×6 square lattice with periodic boundary conditions at $J_2 = 0$ and $J_2 = 0.55$. Since in the more complicated case of the kagome lattice, the NQS method leads to quite a large relative energy error and low fidelity of the variational approximation, we do not consider it here. We represent absolute values and signs of the wave function coefficients with two independent one-hidden-layer dense networks with $4 \times 36 = 144$ hidden neurons. To make the comparison unbiased, to optimize the NQS, we use neither of the Monte Carlo samplers but rather compute energies of the variational states and the weight updates at every epoch by means of exact sampling. We view squared amplitudes of the wave as a discrete probability distribution and sample from it directly using standard textbook algorithms [41]. After the optimization, we go through obtained NQS at every epoch of training and sample from them using Metropolis-Hastings and Zanella algorithms. Corresponding autocorrelation times are shown in Fig. 3. One can see that autocorrelation times computed from both energy and probability correlation functions C_E and C_ψ are significantly smaller for the Zanella process. Upon approaching the ground state, the probabilistic autocorrelation time of the Zanella process tends to increase, but using kinetic sampling remains highly advantageous at all stages of optimization. In Fig. 4, we show UMAP visualization of Metropolis-Hastings and Zanella Markov chains in the case of sampling from a typical NQS encountered during the learning process. The advantage of kinetic sampling over the Metropolis-Hastings algorithm is evident: while the latter tends to stick to the Hilbert space basis sector around one of the two Neel states, the former explores the basis in a much more ergodic manner. The transition between two Neel states is problematic for two reasons: the states are far from each other (Hamming distance between them is maximal) and happen to have very high amplitudes. Although both samplers struggle to make this transition, the Zanella algorithm seems to handle this issue much better than the Metropolis-Hastings algorithm. Note, however, that if the variational ansatz gets trapped in a local metastable minimum over the course of learning (which often happens for highly frustrated quantum spin models) even perfect sampling method cannot help escape it, and the whole optimization procedure needs to be modified [42].

IV. CONCLUSIONS

In most of the NQS optimization algorithms, unless one is using generative models, Monte Carlo sampling is required to compute observables and gradients, which makes it a crucial part of the learning scheme. In this paper, we have analyzed how the quality of sampling from probability distributions defined by many-body wave functions can be improved by using a kinetic Monte Carlo algorithm—continuous-in-time Zanella process—instead of the conventional Metropolis-Hastings algorithm. Being extremely easy to implement, the Zanella process gives a substantial improvement in autocorrelation times. To further assess the quality of sampling, we proposed to employ an UMAP embedding algorithm which constructs visualizations of high-dimensional data sets approximately

preserving distances between elements. It thus serves as a much better source of geometric intuition about the data set structure than, for example, principal component analysis. As follows from UMAP analysis, on top of having smaller autocorrelation times, the Zanella process gives a more uniform coverage of the Hilbert space basis.

Possibly, the main research domain where the advantage provided by kinetic sampling could be of high importance is NQS application to real-time dynamics of nonequilibrium quantum many-body systems. In settings of that kind, not only does the resulting quality of approximation matter, but every single step of the simulation should conform with energy-preserving Hamiltonian evolution. Slight nonergodicity of the sampler could introduce deviations from the proper evolution trajectory that would eventually lead to accumulation of large errors. In this context, employing a sampling algorithm that generates high-quality uncorrelated sequences could be as important as using neural network architectures with good expressibility and generalization properties.

Although even the simplest Zanella algorithm appears to be superior to the Metropolis-Hastings algorithm, further improvements are possible. Using a rejection-free continuous-in-time process allows to avoid getting trapped at the same point for many iterations. However, another possible danger is localization of a Markov chain within a small subset of the space $\mathcal{X}$. If the process enters a region of high probabilities, it could start wandering along short closed trajectories within this region such as $x \to y \to z \to x \to \ldots$, which would negatively affect ergodicity. For probability distributions of this kind, an algorithm that forbids back-tracking might be desirable. Recently, an extension of Zanella process has been suggested which approximately avoids back-tracking on short-to-medium time scales. This is done by promoting Zanella process to a non-Markovian metaheuristic which combines ideas of kinetic Monte Carlo and self-avoiding walks. The algorithm was named Tabu sampler [26]. When applied to sampling from probability distributions on large graphs, Tabu sampler was shown to decrease autocorrelation times by one or two orders of magnitude compared to Zanella process. Implementing it for sampling from many-body wave functions is straightforward if lattice symmetries of the quantum system are not taken into account. However, the algorithm requires non-trivial modifications to be applied in the symmetry-adapted basis, which is a direction for future research.

ACKNOWLEDGMENTS

The authors thank Olle Eriksson, Mikhail Katsnelson, and Danny Thonig for useful discussions. The work of T.W. was supported by European Research Council via Synergy Grant 854843—FASTCORR. A.A.I. acknowledges financial support from Dutch Science Foundation NWO/FOM under Grant No. 16PR1024. A.A.B. acknowledges support from the Russian Science Foundation, Grant No. 18-12-00185. This work was partially supported by Knut and Alice Wallenberg Foundation through Grant No. 2018.0060. This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

APPENDIX A: CORRECTNESS OF SAMPLING ALGORITHMS IMPLEMENTATION

When a sampling algorithm is suggested, it is important to make sure whether it actually samples the correct probability distribution. To assess that, we have conducted two types of tests. First, we took ground states of a few short antiferromagnetic isotropic Heisenberg spin chains (so the Markov chain can visit each basis vector a large number of times), sampled from the corresponding probability distributions, and performed χ^2 [43] and ℓ_1 closeness [44] tests comparing frequencies of basis vectors appearing in the Markov chain with the expected exact probabilities. For both Zanella and Metropolis-Hastings algorithms, we run 32 Markov chains

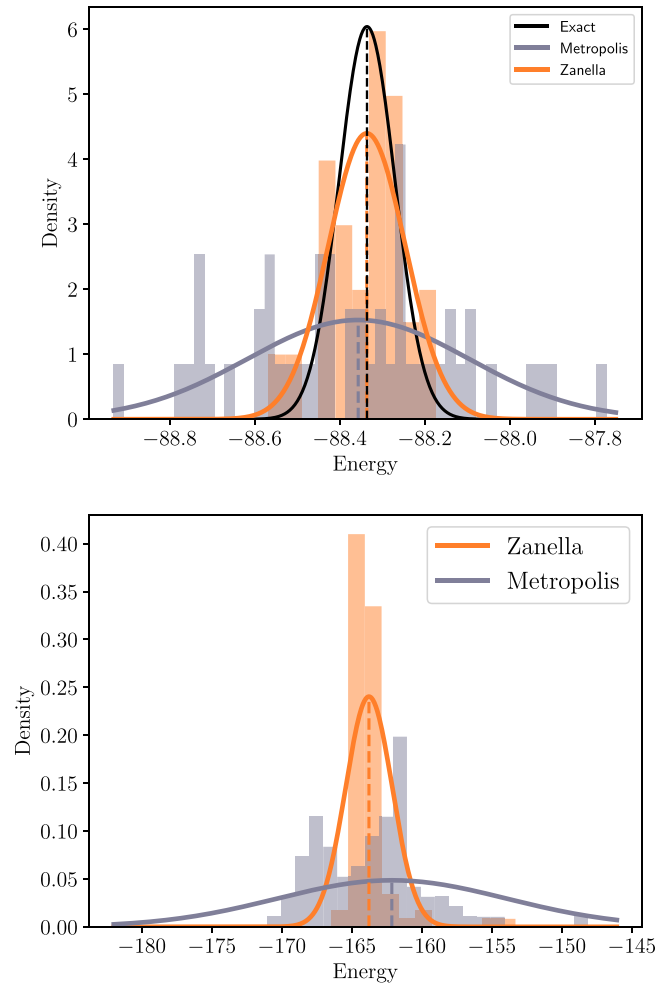

FIG. 5. Comparison of quality of different samplers. Zanella (orange) and Metropolis-Hastings (lavender) algorithms are used to compute energy of a partially converged NQS of 6×6 (top) and 10×10 (bottom) square lattices. In both cases, we run 100 Markov chains of 10 000 samples, and the resulting energies are binned in histograms. Sweep size was taken 1 for Zanella Markov chains and 5 for Metropolis-Hastings Markov chains. Gauss enveloping functions represent the mean and the standard deviation within the ensemble of Markov chains. For the 6×6 lattice, black Gaussian curve represents the distribution of energies computed with exact sampling algorithms with the same number of runs/samples (without showing the histogram).

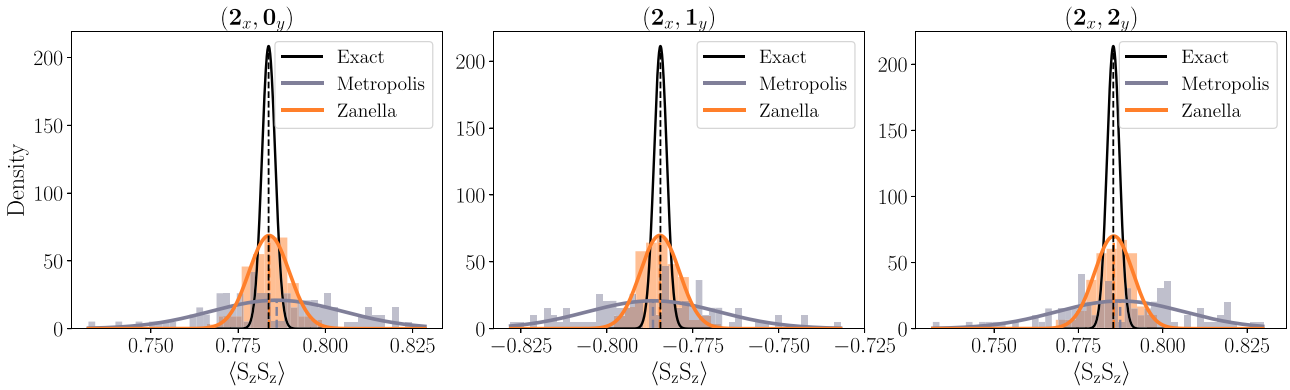

FIG. 6. Comparison of quality of different samplers. Zanella (orange) and Metropolis-Hastings (lavender) algorithms are used to compute $\langle S_z S_z \rangle$ correlation function of a partially converged NQS for different distances between sites (bold numbers represent coordinates on the square lattice). In both cases, we run 100 Markov chains of 10 000 samples, and the resulting energies are binned in histograms. Sweep size was taken 1 for Zanella Markov chains and 5 for Metropolis-Hastings Markov chains. Gauss enveloping functions represent the mean and the standard deviation within the ensemble of Markov chains. Black Gaussian curve represents the distribution of $\langle S_z S_z \rangle$ value computed with exact sampling algorithms with the same number of runs/samples (without showing the histogram).

of length 10^5 for χ^2 test (where we took chains of four and six spins, both in symmetric and nonsymmetric bases), and 10^4 — for ℓ_1 closeness tests (chains of four, six, and eight spins, both in symmetric and nonsymmetric bases). In the χ^2 test, we test that the combined p-value of all Markov chains exceeds 10^{-4}; in the ℓ_1-closeness test we assert that the sampled and target distributions are ε-close in ℓ_1 norm for $\varepsilon = 10^{-4}$.

To demonstrate correctness of sampling in a real-life NQS learning scenario, we picked a generic partially converged neural quantum state of a system of 36 spins (Heisenberg antiferromagnet on a square lattice without frustration, 36 spins, epoch No. 86) and employed Zanella and Metropolis-Hastings algorithms, as well as exact sampling, to estimate its energy, Fig. 5, and spin correlation function $\langle S_z S_z \rangle$, Fig. 6. We have found that for the same length of Markov chain (or the number of samples in the case of exact sampling) and the same number of runs, both sampling schemes give very similar mean values of the observables, while the standard deviation is much lower for the kinetic Zanella algorithm. Even with sweep size 1, it is nearly as good as that of the exact sampling algorithm when used to estimate energy. To attain a similar quality with the Metropolis-Hastings algorithm, sweep size should be taken $\simeq$100.

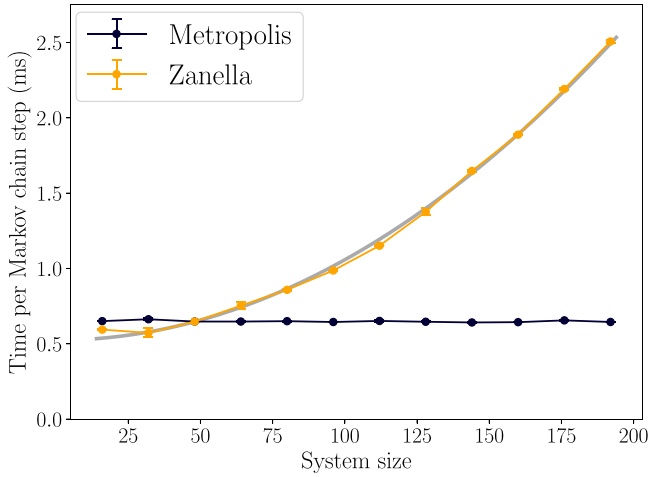

FIG. 7. Performance of different samplers. Time per one Markov chain step for Zanella (orange) and Metropolis-Hastings (blue) algorithms is shown as function of the system size. For Zanella algorithm, this dependence can be fit with $t \simeq 0.524 + 5.34 \cdot 10^{-5} N^2$ ms. Sampling was performed on NVIDIA Tesla V100 GPU for a two-hidden-layer dense network with 512 neurons in each hidden layer and ReLU activation function.

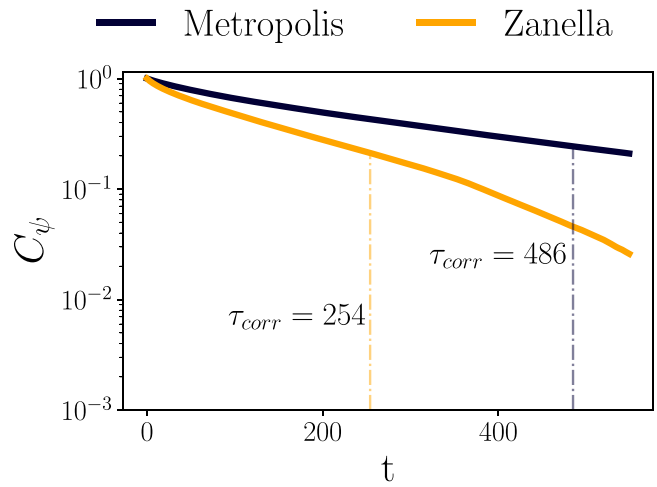

FIG. 8. Autocorrelation function C_ψ of Metropolis-Hastings and Zanella processes computed for the cases of sampling from probability distributions corresponding to approximate ground states of Heisenberg antiferromagnet at $J_2 = 0$ on square lattice with 100 spins. Autocorrelation function was computed by averaging 300 chains of length 8000.

APPENDIX B: PERFORMANCE OF DIFFERENT SAMPLERS

To compare performance of samplers, we use them to sample from probability distributions set by randomly initialized NQS of one-dimensional spin chains of various sizes with global spin inversion and parity symmetries taken into account. We see that time per elementary step of Zanella sampler increases quadratically as the system size is increased, but even for 200 spins it is still less than four times larger than that of the Metropolis-Hastings algorithm, Fig. 7. Taking into account that, as outlined in the previous section, Zanella sampler requires much smaller sweep sizes than Metropolis-Hastings sampler, it is still highly beneficial to use Zanella even on large systems.

The quadratic scaling is due to the computation of ∂x_i at every step of sampling rather than forward propagation through the neural network [i.e., computation of $\pi(y)$] and can be reduced by optimizing the code. In our particular implementation, we allowed arbitrary magnetization-preserving binary spin flips, meaning that the number of configurations in ∂x_i is $\sim N^2$. However, it could easily be the case that if the Markov chain goes over a sparser graph on the Hilbert space basis (e.g., only jumps between basis vectors coupled via Hamiltonian matrix element are allowed), quality of sampling would not be significantly affected with the performance drastically improved.

APPENDIX C: AUTOCORRELATION TIME FOR LARGE SYSTEMS

Finally, we would like to check whether the gain in autocorrelation time provided by the kinetic sampler is still considerable for larger systems. For that, we used Zanella and Metropolis-Hastings algorithms to sample from the ground state of the Heisenberg antiferromagnet ($J_2 = 0$) on a 10×10 lattice. Since it is impossible to have an exact ground-state wave function of that large number of spins, we need to work with its approximation. To make our analysis unbiased, we borrowed a well-trained restricted Boltzmann machine from Ref. [1] with parameters corresponding to relative energy error $\simeq 8 \times 10^{-4}$. Similarly to the case of a 6×6 lattice, the autocorrelation time of Zanella Markov chain is two times smaller than that of Metropolis-Hastings, Fig. 8, indicating that the advantage factor does not depend on the system size.

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

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

[3] K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators, Neural Networks **2**, 359 (1989).

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

[6] 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**, 1 (2020).

[7] *Introduction to Frustrated Magnetism: Materials, Experiments, Theory*, edited by C. Lacroix, P. Mendels, and F. Mila (Springer Science and Business Media, Springer-Verlag Berlin Heidelberg, 2011), Vol. 164.

[8] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Restricted Boltzmann machine learning for solving strongly correlated quantum systems, Phys. Rev. B **96**, 205152 (2017).

[9] Y. Rath, A. Glielmo, and G. H. Booth, A Bayesian inference framework for compression and prediction of quantum states, J. Chem. Phys. **153**, 124108 (2020).

[10] X. Liang, S.-J. Dong, and L. He, Hybrid convolutional neural network and projected entangled pair states wave functions for quantum many-particle states, Phys. Rev. B **103**, 035138 (2021).

[11] A. Szabó and C. Castelnovo, Neural network wave functions and the sign problem, Phys. Rev. Research **2**, 033075 (2020).

[12] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).

[13] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, Quantum Entanglement in Deep Learning Architectures, Phys. Rev. Lett. **122**, 065301 (2019).

[14] Y. Nomura and M. Imada, Dirac-Type Nodal Spin Liquid Revealed by Refined Quantum Many-Body Solver Using Neural-Network Wave Function, Correlation Ratio, and Level Spectroscopy, Phys. Rev. X **11**, 031034 (2021).

[15] Y. Nomura, Helping restricted Boltzmann machines with quantum-state representation by restoring symmetry, J. Phys.: Condens. Matter **33**, 174003 (2021).

[16] H.-C. Jiang, H. Yao, and L. Balents, Spin liquid ground state of the spin-1/2 square $J_1 - J_2$ Heisenberg model, Phys. Rev. B **86**, 024424 (2012).

[17] L. Wang and A. W. Sandvik, Critical Level Crossings and Gapless Spin Liquid in the Square-Lattice Spin-1/2$J_1 - J_2$ Heisenberg Antiferromagnet, Phys. Rev. Lett. **121**, 107202 (2018).

[18] R. Salakhutdinov, Learning deep generative models, Annu. Rev. Stat. Appl. **2**, 361 (2015).

[19] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, Deep autoregressive networks, in *Proceedings of the 31st International Conference on Machine Learning*, edited by E. P. Xing and T. Jebara (PMLR, 2014), Vol. 32, pp. 1242–1250.

[20] O. Sharir, Y. Levine, N. Wies, G. Carleo, A. Shashua, Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems, Phys. Rev. Lett. **124**, 020503 (2020).

[21] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Probabilistic formulation of open quantum many-body systems dynamics with autoregressive models, arXiv:2009.05580

[22] S. Morawetz, I. J. De Vlugt, J. Carrasquilla, and R. G. Melko, U(1)-symmetric recurrent neural networks for quantum state reconstruction, Phys. Rev. A **104**, 012401 (2021).

[23] C. Roth, Iterative retraining of quantum spin models using recurrent neural networks, arXiv:2003.06228

[24] G. Zanella, Informed proposals for local MCMC in discrete spaces, J. Am. Stat. Assoc. **115**, 852 (2020).

[25] A. F. Voter, Introduction to the kinetic Monte Carlo method, *Radiation Effects in Solids*, edited by K. E. Sickafus, E. A. Kotomin, B. P. Uberuaga (Springer, Dordrecht, 2007), pp. 1–23.

[26] S. Power, J. V. Goldman, Accelerated sampling on discrete spaces with non-reversible markov processes, arXiv:1912.04681

[27] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform manifold approximation and projection, J. Open Source Softw. **3**(29), 861 (2018).

[28] F. Ferrari, F. Becca, and J. Carrasquilla, Neural Gutzwiller-projected variational wave functions, Phys. Rev. B **100**, 125131 (2019).

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

[30] A. W. Sandvik, Computational studies of quantum spin systems, AIP Conf. Proc. **1297**, 135 (2010).

[31] A. Wietek and A. M. Läuchli, Sublattice coding algorithm and distributed memory parallelization for large-scale exact diagonalizations of quantum many-body systems, Phys. Rev. E **98**, 033309 (2018).

[32] A. Weiße and H. Fehske, *Exact diagonalization techniques*, Computational many-particle physics (Springer, Berlin, 2008), pp. 529–544.

[33] J. F. Cornwell, *Group theory in physics: An introduction* (Academic Press, San Diego, 1997), Chap. 7.

[34] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Symmetries and Many-Body Excitations with Neural-Network Quantum States, Phys. Rev. Lett. **121**, 167204 (2018).

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

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

[37] A. Sokal, *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms, Functional Integration* (Springer, Boston, MA, 1997), pp. 131–192.

[38] K. Pearson, LIII. On lines and planes of closest fit to systems of points in space, London, Edinburgh, Dublin Philos. Mag. J. Sci. **2**(11), 559 (1901).

[39] T. Westerhout, Lattice-symmetries: A package for working with quantum many-body bases, J. Open Source Softw. **6**(64), 3537 (2021).

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

[41] A. J. Walker, New fast method for generating discrete random numbers with arbitrary frequency distributions, Electron. Lett. **10**, 127 (1974).

[42] C.-Y. Park and M. J. Kastoryano, Are neural quantum states good at solving non-stoquastic spin Hamiltonians? arXiv:2012.08889

[43] S. Raychaudhuri, Introduction to Monte Carlo simulation, *2008 Winter Simulation Conference* (IEEE, 2008), pp. 91–100.

[44] G. Valiant and P. Valiant, An automatic inequality prover and instance optimal identity testing, SIAM J. Comput. **46**, 429 (2017).