Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks

B. McNaughton,[1,2] M. V. Milošević,[2,3] A. Perali,[4] and S. Pilati[1]
[1]*School of Science and Technology, Physics Division, Università di Camerino, 62032 Camerino (MC), Italy*
[2]*Department of Physics, University of Antwerp, Groenenborgerlaan 171, B-2020 Antwerp, Belgium*
[3]*NANOlab Center of Excellence, University of Antwerp, Belgium*
[4]*School of Pharmacy, Physics Unit, Università di Camerino, 62032 Camerino (MC), Italy*

(Received 16 February 2020; accepted 12 May 2020; published 28 May 2020)

The autoregressive neural networks are emerging as a powerful computational tool to solve relevant problems in classical and quantum mechanics. One of their appealing functionalities is that, after they have learned a probability distribution from a dataset, they allow exact and efficient sampling of typical system configurations. Here we employ a neural autoregressive distribution estimator (NADE) to boost Markov chain Monte Carlo (MCMC) simulations of a paradigmatic classical model of spin-glass theory, namely, the two-dimensional Edwards-Anderson Hamiltonian. We show that a NADE can be trained to accurately mimic the Boltzmann distribution using unsupervised learning from system configurations generated using standard MCMC algorithms. The trained NADE is then employed as smart proposal distribution for the Metropolis-Hastings algorithm. This allows us to perform efficient MCMC simulations, which provide unbiased results even if the expectation value corresponding to the probability distribution learned by the NADE is not exact. Notably, we implement a sequential tempering procedure, whereby a NADE trained at a higher temperature is iteratively employed as proposal distribution in a MCMC simulation run at a slightly lower temperature. This allows one to efficiently simulate the spin-glass model even in the low-temperature regime, avoiding the divergent correlation times that plague MCMC simulations driven by local-update algorithms. Furthermore, we show that the NADE-driven simulations quickly sample ground-state configurations, paving the way to their future utilization to tackle binary optimization problems.

DOI: 10.1103/PhysRevE.101.053312

I. INTODUCTION

Artificial neural networks are finding increasing applicability in various fields of classical and quantum physics [1], where the generative neural networks are turning out to be particularly useful. They can be trained to mimic complex probability distributions, either using unsupervised learning protocols from unlabeled datasets, or using reinforcement learning schemes whereby a reward function is optimized. Among other generative models (e.g., the variational autoencoders [2,3]), the restricted Boltzmann machines [4] were already proven successful in solving several computational tasks, including: learning classical thermodynamics from Monte Carlo samples [5], accelerating classical Monte Carlo simulations [6] (see also Ref. [7] for a related method), building accurate variational wave functions [8], performing quantum state tomography [9], simulating open quantum systems [10–13], decoding topological codes [14], guiding projective quantum Monte Carlo simulations [15], and reconstructing density matrices [16]. The autoregressive neural networks provide additional distinctive functionalities compared to the restricted Boltzmann machines. In particular, owing to a specific connectivity structure, they allow efficient and exact sampling of system configurations according to the learned probability distribution. This is achieved via so-called ancestral sampling (see, e.g., Ref. [17]), whereby the system variables are sampled in a pre-determined order according to a chain of conditional probability distributions. This avoids resorting to Markov chain Monte Carlo (MCMC) algorithms, which are often plagued by long correlations times, leading to an excessive computational cost or even to biased results due to the lack of ergodicity. Ancestral sampling has already been exploited in quantum physics to accelerate the optimization of variational wave functions [18,19]. Recently, a variational framework to solve rather general classical statistical-mechanics problems using autoregressive networks has also been presented [20]. It has been applied to clean systems and also to a mean-field disordered model, namely, the Sherrington-Kirkpatrick spin Hamiltonian [21]. This model has infinite-range interactions, and its properties are exactly predicted by Parisi's mean-field theory based on the replica method [22]. While the variational framework of Ref. [20] extends well beyond the standard mean-field theories commonly employed to study spin glasses, it might still provide biased results. This bias is due to the inevitable difference between the probability distribution learned by the neural network and the Boltzmann distribution.

This bias can be eliminated with two approaches. In the first, the autoregressive model is used for importance sampling in a reweighting scheme. In the second, it is used as a smart proposal distribution for the Metropolis-Hastings algorithm. In the field of machine learning, the first approach has been employed to compute otherwise intractable normalization integrals [23]. An application of both approaches to classical statistical mechanics has appeared only very

recently [24].[1] However, the study of Ref. [24] focused only on clean ferromagnetic models, and it considered only the reinforcement learning scheme to train the neural network, as in Ref. [20].

Random spin systems with frustrated interactions display many intriguing phenomena related to glass physics [25,26], including magnetic correlations, replica symmetry breaking, hysteresis, and aging. In fact, spin-glass models with short-range interactions have challenged theoretical physicists for decades. It is unclear whether the mean-field replica theory, which exactly describes infinite-range models, applies also to short-range systems, even at the qualitative level (see, e.g., Refs. [27–30]). The computational problems originate from the exceedingly long autocorrelation times that plague standard MCMC simulations driven by local-update algorithms. This problem arises also when addressing binary optimization problems—which are ubiquitous in scientific research and in industry—via stochastic optimization methods such as simulated annealing [31]. Indeed, identifying the optimal solution is equivalent to finding the lowest-energy configuration of a disordered Ising Hamiltonian. In recent decades, relevant algorithmic developments have occurred in the field of spin glasses. In particular, one should mention the global-update methods such as the parallel tempering technique [32] and the isoenergetic cluster updates [33,34]. Still, novel and possibly more flexible MCMC methods would be extremely useful.

In this article we investigate the use of autoregressive neural networks to increase the efficiency of MCMC simulations of spin glasses. The model we focus on is a paradigmatic short-range spin model, namely, the two-dimensional Edwards-Anderson Hamiltonian. The nearest-neighbor couplings are sampled from a gaussian distribution. The neural network we employ is a standard autoregressive generative model, namely, the so-called neural autoregressive distribution estimator (NADE) [35]. In this article, the network is trained in an unsupervised learning scheme, which consists of minimizing the Kullback-Leibler divergence with respect to a set of spin configurations sampled using MCMC simulations driven by a standard local-update algorithm. Our analysis shows that the NADE can learn to accurately mimic the Boltzmann distribution of the spin-glass model. We quantify this accuracy, and how it varies with the number of hidden neurons and the physical system size, by comparing the energy expectation-values corresponding to the NADE with unbiased results corresponding to the Boltzmann distribution. The trained NADE is then employed to generate spin configurations for the proposal step of Metropolis-Hastings algorithm. This allows us to essentially eliminate the correlations that affect standard MCMC simulations driven by local-update algorithms. An important contribution of this article is the implementation of a sequential tempering procedure. This starts from a moderately high temperature, where uncorrelated configurations are easily sampled also via the local-update MCMC algorithm. Then, it performs a sequence of MCMC simulations at successively lower temperatures, whereby the NADE trained at the previous temperature is used to drive the Metropolis-Hastings algorithm. This allows us to perform efficient MCMC simulations even in the low temperature regime, where local-update algorithms become impractical due to the diverging autocorrelation times. Finally, we analyze how efficiently the NADE-driven simulations performed at low temperatures sample the ground-state configurations. The obtained encouraging results lead us to advocate the use of autoregressive models to boost stochastic optimization methods such as the simulated annealing.

The article is organized as follows: in Sec. II we describe the Ising glass Hamiltonian, the NADE training method, the local-update as well as the NADE-driven MCMC algorithms. Section III presents the results obtained by training the NADE, by running local-update and NADE-driven MCMC simulations, and by performing the sequential tempering procedure. The possible use of NADEs to boost stochastic optimization algorithms is also discussed. Our conclusions and some future perspectives are given in Sec. IV.

II. MODEL AND METHODS

The spin-glass model addressed in this article is the two-dimensional Edwards-Anderson Hamiltonian:

$$H(\boldsymbol{\sigma}) = -\sum_{\langle ij \rangle} J_{ij}\sigma_i\sigma_j, \tag{1}$$

where $\sigma_i \in \{-1, 1\}$ are binary spin variables at the sites labeled by the indices $i, j = 1, \ldots, N$, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_N)$ indicates the spin configuration, and J_{ij} is the coupling strength between the spins i and j. We consider random couplings sampled from a gaussian distribution with zero mean and unit variance. The sum in the above equation runs over nearest-neighbor sites on a square lattice. Periodic boundary conditions are adopted. The thermodynamic properties, e.g., the average energy E, are computed as $E = \langle H(\boldsymbol{\sigma})\rangle$, where the angular brackets indicate expectation values over the Boltzmann distribution $P(\boldsymbol{\sigma}) = \exp[-\beta H(\boldsymbol{\sigma})]/Z$. Here, $\beta = 1/k_BT$ is the inverse temperature, $k_B = 1$ is the Boltzmann constant, T is the temperature, and $Z = \sum_{\boldsymbol{\sigma}} \exp[-\beta H(\boldsymbol{\sigma})]$ is the partition function. Expectation values of this kind can be computed by implementing a stochastic Markov chain in the configuration space driven by a transition matrix. The entries of this matrix will be denoted as $T_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$. They must satisfy the conditions $T_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} \geqslant 0$ and $\sum_{\boldsymbol{\sigma}'} T_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = 1$, for any $\boldsymbol{\sigma}$, meaning that the transition matrix is stochastic. $T_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$ represents the probability to move to the configuration $\boldsymbol{\sigma}'$ from $\boldsymbol{\sigma}$. A common procedure is to decompose the transition matrix in the proposal-acceptance form: $T_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = \omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$ for $\boldsymbol{\sigma}' \neq \boldsymbol{\sigma}$, and $T_{\boldsymbol{\sigma}\boldsymbol{\sigma}} = \sum_{\boldsymbol{\sigma}' \neq \boldsymbol{\sigma}} w_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}(1 - A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}) + w_{\boldsymbol{\sigma}\boldsymbol{\sigma}}$. The entries of the proposal distribution matrix $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$ represent the probability to propose moving to $\boldsymbol{\sigma}'$ from $\boldsymbol{\sigma}$. This matrix must be stochastic, and satisfy an ergodic condition, meaning that it is possible to reach any configuration from any other in a finite number of steps. $A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$ is the probability to accept the proposed update; the probability to reject it, thus iterating the old configuration

[1] A generative neural network has been used to propose updates in a MCMC simulation also in Ref. [6] but using a restricted Boltzmann machine instead of an autoregressive model. This required to run a parallel MCMC simulation, which was implemented via alternated Gibbs sampling.

$\boldsymbol{\sigma}$ in the Markov chain, is $1 - A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$. One way to satisfy the detailed balance condition—which is sufficient (although not necessary) to ensure that the Boltzmann distribution is the stationary distribution of the Markov chain—is to define the acceptance probability as

$$A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = \min\left(1, \frac{P(\boldsymbol{\sigma}')\omega_{\boldsymbol{\sigma}\boldsymbol{\sigma}'}}{P(\boldsymbol{\sigma})\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}}\right). \tag{2}$$

This formula corresponds to the famous Metropolis-Hastings algorithm [36]. Notice that, since one needs only ratios of Boltzmann-distribution values, the normalization factor Z is not needed. A common choice is to consider a symmetric proposal distribution, i.e., $\omega_{\boldsymbol{\sigma}\boldsymbol{\sigma}'} = \omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$. In this case the acceptance probability simplifies to: $A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = \min(1, \frac{P(\boldsymbol{\sigma}')}{P(\boldsymbol{\sigma})})$. For example, one can randomly choose a single spin i and propose to flip it, setting $\sigma'_i = -\sigma_i$. This corresponds to the matrix entries: $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = 1/N$ if $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}'$ differ by one spin-flip only, and $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = 0$ otherwise. In what follows, this local method will be referred to as single spin-flip algorithm. It is efficient enough for rather generic models. However, in the vicinity of phase transitions (e.g., ferromagnetic transitions in ordered Ising models) or in the glassy phases of disordered systems, the dynamics of MCMC simulations driven by the single spin-flip algorithm suffer a pathological slowing down, possibly leading to the breakdown of ergodicity. This slowing down is associated to strong statistical correlations between configurations subsequently sampled along the Markov chain. In particular, in the case of low-temperature spin-glass models, the correlation time diverges and the Markov chain is not ergodic, meaning that not all physically relevant regions of the configuration space are explored in the feasible computational times. For certain relevant systems, these correlations can be suppressed adopting more sophisticated (in general, nonsymmetric) proposal distributions $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$. This approach is often referred to as smart Monte Carlo method [37]. For example, for ferromagnetic Ising models one can adopt the Swendsen-Wang or the Wolff algorithms [38,39]. These perform cluster moves instead of single spin-flip updates. The worm algorithm is a relevant alternative [40,41]. MCMC methods that perform significantly better than the single spin-flip updates have been developed also for spin glasses. Relevant examples are the parallel tempering method [32] and the iso-energetic cluster-update algorithms [33,34]. However, spin glasses still represent a computational challenge. The long correlation times plague also most heuristic methods commonly employed to solve binary optimization problems. In fact, identifying the optimal solution is equivalent to finding one of the ground-state configurations of a spin-glass model. This task constitutes a nondeterministic polynomial hard problem when implemented on a nonplanar graph [42]. The archetypal heuristic optimization method is simulated annealing [31]. This method exploits MCMC algorithms to explore possible solutions, but the lack of ergodicity might prevent the dynamics from reaching the lowest-energy configuration. Below we report how to exploit autoregressive neural networks to implement efficient MCMC algorithms for Ising Hamiltonians of the type defined in Eq. (1). This the central objective of this work.

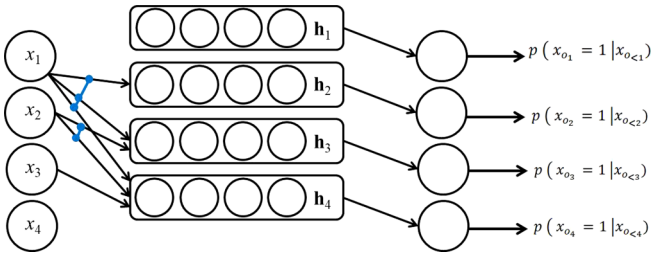

FIG. 1. Illustration of a neural autoregressive distribution estimator. Arrows connected by blue segments correspond to connections with shared parameters.

The neural autoregressive distribution estimator (NADE)

An efficient proposal distribution can be constructed using autoregressive neural networks. In this article, we consider the use of NADEs. Like other generative neural networks, NADEs can be trained to model complex probability distribution from sampled data. Then, they allow direct sampling of system instances from the learned probability distribution. The system instances are represented by vectors of binary variables $\boldsymbol{x} = (x_1, \ldots, x_D)$. Here, following the convention of the machine-learning literature, we consider the binary values $x_d \in \{0, 1\}$, for $d = 1, \ldots, D$. The joint distribution of the binary variables is decomposed as a product of chained conditional probabilities:

$$p(\mathbf{x}) = \prod_{d=1}^{D} p\big(x_{o_d} \big| \mathbf{x}_{o_{<d}}\big). \tag{3}$$

In the above equation, o denotes the chosen ordering of the binary variables, o_d indicates the dth variable in the ordering o, and the slice subscript $o_{<d}$ indicates the first $d-1$ dimensions in o; therefore, $\mathbf{x}_{o_{<d}}$ represents the subvector including the indicated dimensions only. Each conditional probability is modeled using a feed-forward neural network, defined as

$$p\big(x_{o_d} = 1 \big| \mathbf{x}_{o_{<d}}\big) = \mathrm{sigm}\big(\mathbf{V}_{o_d,\cdot}\mathbf{h}_d + b_{o_d}\big), \tag{4}$$

where $\mathbf{V}$ is a weight matrix, b_{o_d} is the bias, and one has D vectors of hidden-neuron activations, computed as

$$\mathbf{h}_d = \mathrm{sigm}\big(\mathbf{W}_{\cdot,o_{<d}}\mathbf{x}_{o_{<d}} + \mathbf{c}\big), \tag{5}$$

with the weight matrix $\mathbf{W}$ and the bias vector $\mathbf{c}$. The activations are computed with the logistic function $\mathrm{sigm}(x) = 1/[1 + \exp(-x)]$. We used the notation of vectorized functions and that of slice indices also for submatrices. The NADE has D hidden layers, each including N_H neurons. An important property is that the matrix $\mathbf{W}$ and the bias vector $\mathbf{c}$ are shared by all hidden layers. The parameters to be optimized in the training process are $\mathbf{V} \in \mathbb{R}^{D\times N_H}$, $\mathbf{b} \in \mathbb{R}^{D}$, $\mathbf{W} \in \mathbb{R}^{N_H\times D}$, and $\mathbf{c} \in \mathbb{R}^{N_H}$. The structure of the whole neural network is represented in Fig. 1. By construction, the conditional probability distribution $p(x_{o_d}|\mathbf{x}_{o_{<d}})$ for the variable x_{o_d} does not depend on the subsequent variables $\mathbf{x}_{o_{>d}}$ in the ordering o. Therefore, after training has been performed, one can sample configurations $\mathbf{x}$ from the learned distribution $p(\mathbf{x})$ via ancestral sampling. Following the ordering o, each variable x_{o_d} is sampled from the binary distribution $p(x_{o_d}|\mathbf{x}_{o_{<d}})$, computed using the previously sampled variables. In practice, for each

spin x_{o_d}, a uniform random variable $r_d \in (0, 1)$ is sampled with a pseudo-random number generator, and one sets $x_{o_d} = 1$ if $r_d < p(x_{o_d} = 1|\mathbf{x}_{o_{<d}})$, and $x_{o_d} = -1$ otherwise. This would not be possible with a closely related neural-network model such as the restricted Boltzmann machine. In fact, in that case one has to resort to MCMC algorithms, usually implemented via alternated Gibbs sampling of hidden neurons and visible neurons [43]. This might cause problems associated to long correlation times, leading to an excessive computational cost in practical application of generative sampling. Furthermore, with the NADE the (normalized) likelihood of a configuration $\mathbf{x}$ can be efficiently computed via Eq. (3). Instead, with the restricted Boltzmann machine one has to determine the normalization integral, namely, the partition function, which is an intractable computation already for moderately large systems.

The NADE can be trained in an unsupervised learning scheme from a (typically large) dataset $\{\mathbf{x}^{(n)}\}$, where $n = 1, \ldots, N_t$, and N_t is the training-dataset size. The cost function to be minimized is the average negative log-likelihood, given by

$$\mathrm{nl}(\mathbf{V}, \mathbf{b}, \mathbf{W}, \mathbf{c}) = -\frac{1}{N_t} \sum_{n=1}^{N_t} \log p(\mathbf{x}^{(n)}). \tag{6}$$

It can be shown that this criterion corresponds to the minimization of the so-called Kullback-Leibler divergence (see, e.g., Ref. [44]), which is defined as

$$\mathrm{KL}(q||p) = \sum_{\mathbf{x}} q(\mathbf{x}) \ln[q(\mathbf{x})/p(\mathbf{x})], \tag{7}$$

where $q(\mathbf{x})$ is the underlying probability distribution of the samples $\{\mathbf{x}^{(n)}\}$, which is in general unknown. The optimization of the network parameters $\mathbf{\Theta} \equiv \{\mathbf{V}, \mathbf{b}, \mathbf{W}, \mathbf{c}\}$ is performed via the stochastic gradient descent algorithm. Starting from reasonably chosen initial values $\mathbf{\Theta}_0$, at each step $s = 0, 1, \ldots$ one performs the update $\mathbf{\Theta}_{s+1} = \mathbf{\Theta}_s - \eta \nabla_{\mathbf{\Theta}} \mathrm{nl}(\mathbf{\Theta})|_{\mathbf{\Theta}_s}$, where the scalar η is the learning rate. The log-likelihood gradient is computed via the back-propagation algorithm. In the standard implementation, at each step only a small mini-batch of N_b system instances, randomly sampled from the training set, is used to compute the gradient. One learning epoch includes $\lfloor N_t/N_b \rfloor$ steps ($\lfloor\ \rfloor$ is the floor function), and the optimization is iterated for a number of epochs N_E, till convergence is reached. In our implementation, a large training set is considered, and in each epoch the mini-batches are sampled from a random subset of the total training set, as detailed in the next section. For the specific problems addressed in this article, it appears that regularization procedures are not strictly necessary. All details and the pseudocode of the algorithm to compute the log-likelihood and its gradient can be found in Refs. [35,45]. It is worth mentioning that this algorithm exploits the NADE's specific structure, in particular the sharing of $\mathbf{W}$ and $\mathbf{c}$, to enhance efficiency. Specifically, the computational cost to compute the likelihood of a system instance scales as $N_H D$, instead of the $N_H D^2$ scaling corresponding to the naïve implementation that does not exploit the NADE's structure [35,45].

In this article, a NADE is used to learn the probability distribution corresponding to a large dataset of equilibrium configurations of the spin-glass model Eq. (1). These configurations are generated using a single spin-flip MCMC simulation. In regimes where this local algorithm performs an ergodic dynamics, this simple procedure is sufficient to allow the NADE to learn the Boltzmann distribution. To reach also the regimes where the dynamics given by the local algorithm are not ergodic, we implement a sequential tempering procedure. This is described in the next section. For our purpose, the size of the network input-layer has to be $D = N$. The spins are ordered line-by-line. To switch between the two binary-value conventions, namely, $\sigma_i \in \{-1, 1\}$ and $x_i \in \{0, 1\}$, the following mapping is used: $x_i = 1$ if $\sigma_i = 1$, and $x_i = 0$ if $\sigma_i = -1$. Once the training has been performed, the NADE probability distribution $p(\boldsymbol{\sigma})$ approximates the Boltzmann distribution $P(\boldsymbol{\sigma})$. Therefore, the thermodynamic properties as, e.g., the average energy E, can be approximated by expectation values over the NADE probability distribution:

$$E_{\mathrm{NADE}} \simeq \langle H(\boldsymbol{\sigma}) \rangle_{p(\boldsymbol{\sigma})} = \lim_{N_s \to \infty} \frac{1}{N_s} \sum_{n=1}^{N_s} H(\boldsymbol{\sigma}^{(n)}), \tag{8}$$

where the N_s configurations $\boldsymbol{\sigma}^{(n)} \sim p(\boldsymbol{\sigma}^{(n)})$ are efficiently sampled from the NADE distribution using ancestral sampling. In practice, one uses a large but finite number of configurations N_s. This procedure is expected to be very efficient, since the sampled configurations are perfectly uncorrelated. As shown in the next section, this calculation allows us to quantify how accurately the NADE approximates the Boltzmann distribution. In general, a NADE with a finite number of hidden neurons N_H will not exactly match the Boltzmann distribution, so the expectation values over the NADE distribution will be biased. Note that, even if N_H is very large, a bias might also originate from an imperfect training due to a too small training set or to a failure of the optimization algorithm. Two strategies can be adopted to remove this bias. The first one, which we employ in this article, is to combine the trained NADE with a standard MCMC method, as anticipated above. Specifically, we use the NADE as proposal distribution for the Metropolis-Hastings algorithm; that is, we set $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = p(\boldsymbol{\sigma}')$. This procedure is correct if $p(\boldsymbol{\sigma})$ and $P(\boldsymbol{\sigma})$ have the same support. Formally, this condition is always satisfied since the conditional probabilities are computed using the logistic function, which has values $\sigma(x) \in (0, 1)$ for any finite x. In the next section we analyze if and when the weight of the NADE distribution is sufficiently large, in any physically relevant configuration, to produce an efficient simulation. It is worth stressing that, with this choice, the proposal distribution $\omega_{\boldsymbol{\sigma}'\boldsymbol{\sigma}}$ is independent of the starting configuration $\boldsymbol{\sigma}$. Therefore, when the proposal is accepted, the next configuration in the Markov process is uncorrelated with the previous one. If $p(\boldsymbol{\sigma}) = P(\boldsymbol{\sigma})$ the acceptance probability is $A_{\boldsymbol{\sigma}'\boldsymbol{\sigma}} = 1$. Therefore, statistical correlations along the Markov chain only originate from the approximation in the NADE distribution, since this leads to some rejections. Our results, shown in the next section, indicate that NADEs can be accurate enough to have a high acceptance rate and, therefore, enable efficient MCMC simulations of the spin-glass model. One should also note that, since the computational cost of computing the configuration likelihood and the conditional probabilities (required for ancestral sampling) is linear in the system size, the NADE allows

one to implement an efficient global-update algorithm with a linear computational cost. The MCMC simulations and the NADE trainings discussed in this article are performed with a custom Python code. It is worth emphasizing that in this strategy the proposal configurations are generated directly via ancestral sampling. This is in stark contrast with a previous related approach [46], in which configurations are selected from a database of previously visited configurations according to the running estimate of the density of states.

The second strategy to remove the bias in the NADE expectation values is based on a re-weighting scheme. In fact, an unbiased expectation value can be computed as [24]

$$E = \lim_{N_s\to\infty} \sum_{n=1}^{N_s} z_n H(\boldsymbol{\sigma}^{(n)}), \tag{9}$$

where $\boldsymbol{\sigma}^{(n)} \sim p(\boldsymbol{\sigma}^{(n)})$, the weights are $z_n = \zeta_n / \sum_{n=1}^{N_s} \zeta_n$, with $\zeta_n = \exp[-H(\boldsymbol{\sigma}^{(n)})]/p(\boldsymbol{\sigma}^{(n)})$. An analogous re-weighting strategy has been used in Ref. [23] to compute the partition function of a restricted Boltzmann machine. This computation would otherwise be intractable. Very recently, in Ref. [24] both strategies described above have been adopted, but focusing only on the thermodynamic properties of a clean ferromagnetic Ising model. Interestingly, Ref. [24] discusses also how to determine expectation values that explicitly depend on the partition function, as, e.g, the entropy, including the correct formula for the corresponding statistical uncertainty. In the study of Ref. [24], the neural network has been trained via a reinforcement learning procedure, as in Ref. [20]. This reinforcement procedure consists of minimizing a variational ansatz for the free energy based on the generative neural network. The training batches are sampled from the neural network distribution, rather than from a MCMC simulation. Rather than a NADE, Ref. [24] employed an autoregressive neural network named PixelCNN [47], as in Ref. [20].

III. RESULTS

The spin-glass model defined in Eq. (1) can be simulated using the single spin-flip Metropolis-Hastings algorithm, as explained in the previous section. However, it is well known that such simulations can be affected by long autocorrelation times along the Markov chain [25], in particular in the low-temperature regime. To illustrate this effect and to quantify these statistical correlations, we compute the autocorrelation function $c(\tau)$ of the configuration energy. $c(\tau)$ is defined as

$$c(\tau) = \frac{\langle H(t+\tau)H(t)\rangle - \langle H(t)\rangle^2}{\langle H(t)H(t)\rangle - \langle H(t)\rangle^2}, \tag{10}$$

where integers τ and t count MCMC steps, $H(t)$ is the energy in the spin configuration sampled at the tth step of the Markov chain, and the angular brackets indicate the average over the whole Markov chain. Following the standard procedure, we disregard an initial segment to account for equilibration effects. The results corresponding to a typical realization of the gaussian random couplings are shown in Fig. 2. Different realizations provide qualitatively similar results. The correlation function appears to be reasonably well described by an empirical stretched-exponential fitting function: $c(\tau) = a\exp[-(\tau/\tau^*)^\alpha]$. Here, a, τ^*, and α are fitting parameters.

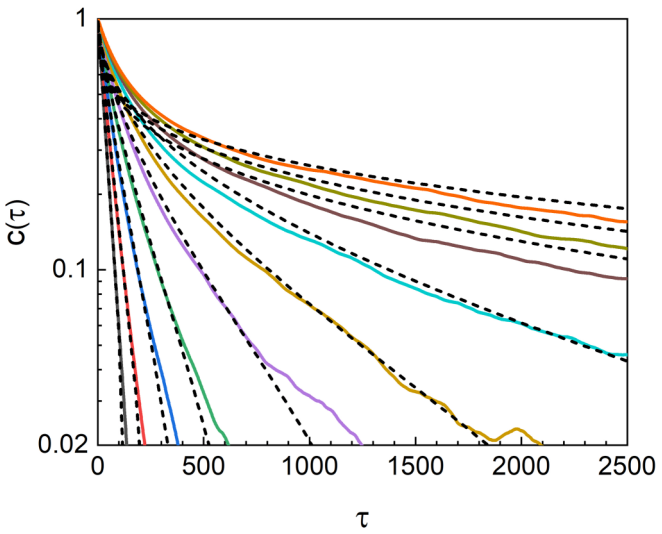

FIG. 2. Autocorrelation function $c(\tau)$ of the configuration energies H obtained from single spin-flip MCMC simulations at inverse temperatures $\beta = 0.1, 0.2, \ldots, 1$ (bottom to top). Dashed lines represent stretched-exponential fitting functions.

At relatively low inverse temperatures $\beta \lesssim 0.5$, we obtain $\alpha \simeq 1$, corresponding to the common exponential decay, and $\tau^* \approx 50$, indicating that the correlation time is sufficiently short. Indeed, MCMC simulations much longer than $\tau^* = 50$ are feasible, even with modest computational resources. However, already at $\beta \simeq 1$, we obtain $\alpha \simeq 0.3$ and $\tau^* \simeq 400$. For inverse temperatures $\beta > 1$, computing the thermodynamic properties with the single spin-flip updates becomes hard or absolutely unfeasible due to the breakdown of ergodicity for the feasible simulation times in the low-temperature regime. In the early literature on spin glasses it had been conjectured that this freezing temperature $\beta \simeq 1$ was associated to a spin-glass transition (see, e.g., Ref. [48], and also Ref. [25] for a review). However, later studies based on faster computers and on more efficient algorithms have clarified that for two-dimensional Ising Hamiltonian with nearest-neighbor interactions, a proper spin-glass phase—identified, among other criteria, by the Edwards-Anderson order parameters or from the distribution of overlaps among system replicas—occurs only in the zero-temperature limit [33,49–52].

Our main goal is to use a NADE to implement efficient MCMC simulations of the spin-glass model Eq. (1), even in the low-temperature regime. As a preliminary step, we show that the NADE can be trained to mimic the Boltzmann distribution in an unsupervised learning scheme. As an illustrative example, the case of a spin glass of size $N = 100$ at $\beta = 1$ is analyzed in Fig. 3. The total training set includes 10^5 configurations, generated using single spin-flip updates. Every 800th configuration sampled by the Markov chain is included in the training set (corresponding to a total of 8×10^7 single spin-flip MCMC steps). This allows suppressing the statistical correlations among the training configurations. In each training epoch, 2×10^4 configurations are randomly selected. The total training set is considered to be representative of the Boltzmann distribution, since at $\beta = 1$ the single spin-flip algorithm is still sufficiently efficient to explore all physically

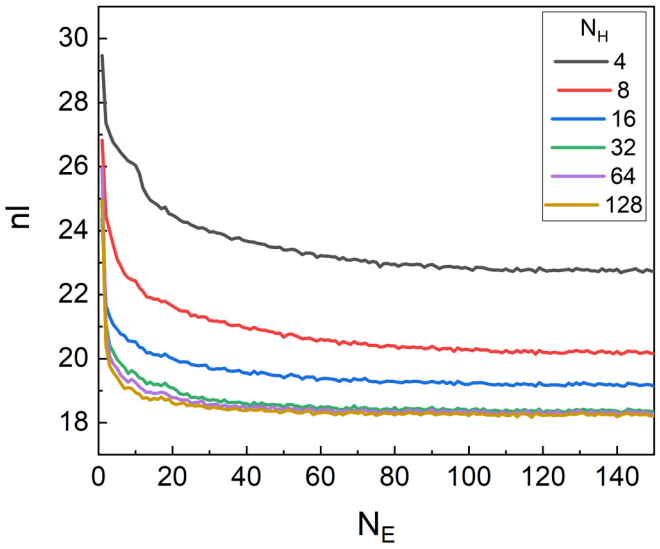

FIG. 3. Average of the negative log-likelihood (nl) as a function of the number of epochs (N_E). Different curves correspond to different numbers of hidden units N_H, increasing from top to bottom. The size of the two-dimensional spin glass is $N = 100$ and the inverse temperature is $\beta = 1$.

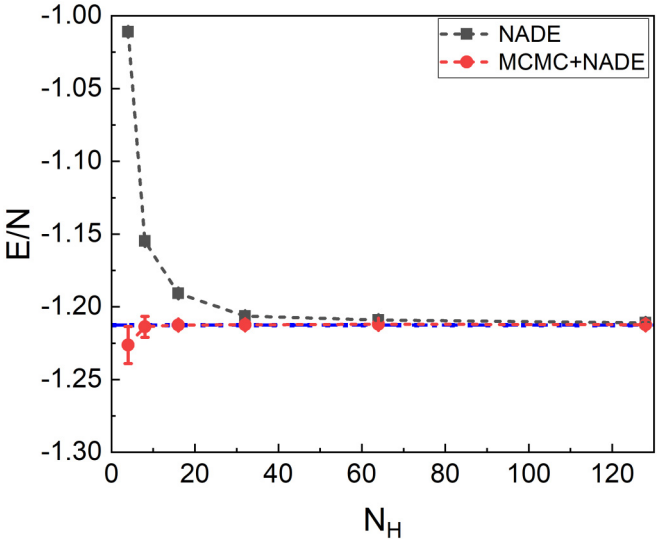

FIG. 4. Average energy per spin E/N as a function of the number of hidden units N_H. The system size is $N = 100$ and the inverse temperature is $\beta = 1$. The black squares correspond to the expectation value over the probability distribution learned by the NADE [see Eq. (8)]. The red dots correspond to the results of the NADE-driven MCMC simulation. The horizontal blue line indicates the statistically unbiased result corresponding to the Boltzmann distribution.

relevant regions of the configuration space within the feasible simulation times. The training is performed by minimizing the negative log-likelihood of the training set via stochastic gradient descent, as explained in the previous section. The learning rate is $\eta = 10^{-3}$ and the mini-batch size is $N_b = 16$. The results obtained with reasonable variations around these values are comparable. After $N_e \simeq 100$ learning epochs the training appears to have converged, slightly depending on the number of hidden units N_H. To verify that the NADE is not overfitting the training data, and to quantify its generalization accuracy, we compare the energy expectation value E_{NADE} over the probability distribution learned by the NADE [see Eq. (8)], with the statistically unbiased result E, corresponding to the exact Boltzmann distribution. This comparison is shown in Fig. 4. For small N_H a sizable bias occurs, indicating that a small NADE is not flexible enough to accurately reproduce the Boltzmann distribution. However, with $N_H \approx 100$ hidden neurons the bias is smaller than the statistical error-bars. To perform a more stringent energy-resolved benchmarking, we compare the histograms of the energies sampled by the NADE probability distribution with the values sampled during the single spin-flip MCMC simulation (see Fig. 5). Excellent agreement is found in all relevant energy regimes. This is a remarkable finding of our work, in view of the complex structure of the spin-glass configuration-space in this intermediate temperature regime. As shown if Fig. 6, the accuracy seems to slowly decrease with the system size N, if N_H is fixed.

By combining the trained NADE with the Metropolis-Hastings algorithm one can, on the one hand, remove the residual bias in the NADE expectation value, and, on the other hand, boost the efficiency of the MCMC simulation. The trained NADE is used to define the proposal distribution for the Metropolis-Hastings algorithm. As explained in the previous section, this leads to an efficient algorithm if the acceptance rate A_R, namely, the percentage of accepted updates, is high. We recall that if the probability distribution learned by NADE exactly matches the Boltzmann distribution, the acceptance rate is $A_R = 100\%$. As shown in Fig. 7, for $N_H = 10$ one has $R \simeq 20\%$, but for $N_H \simeq 100$ the acceptance rate is as high as $R \simeq 80\%$. A_R appears to saturate for large N_H. This might indicate that, to train large NADEs, one needs a more efficient optimization algorithm and/or a larger training

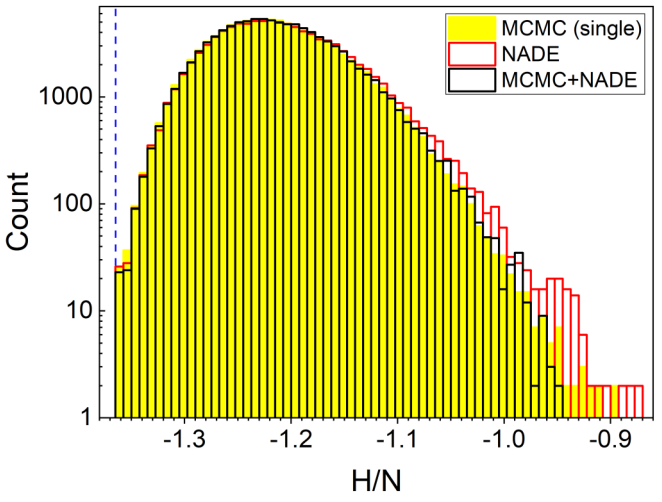

FIG. 5. Histogram of 10^5 sampled configuration energies per spin H/N. The full yellow columns correspond to a single spin-flip MCMC simulation (8×10^7 steps, every 800th configuration is counted), the empty black columns to a NADE-driven MCMC simulation (every 10th configuration is counted), and the empty red columns to the energies sampled via ancestral sampling from the probability distribution learned by the NADE. $N = 100$, $\beta = 1$, and $N_H = 64$.

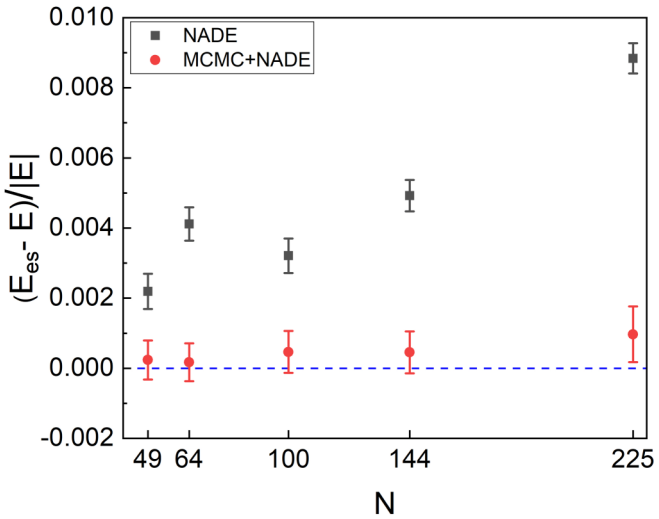

FIG. 6. Relative error $(E_{es} - E)/|E|$ of the estimated energy E_{es} with respect to the exact value E, as a function of the system size N. For the black squares E_{es} is the expectation value over the probability distribution learned by NADE, while for the red dots E_{es} is the result of a NADE-driven MCMC simulation. $N_H = 64$ and $\beta = 1$.

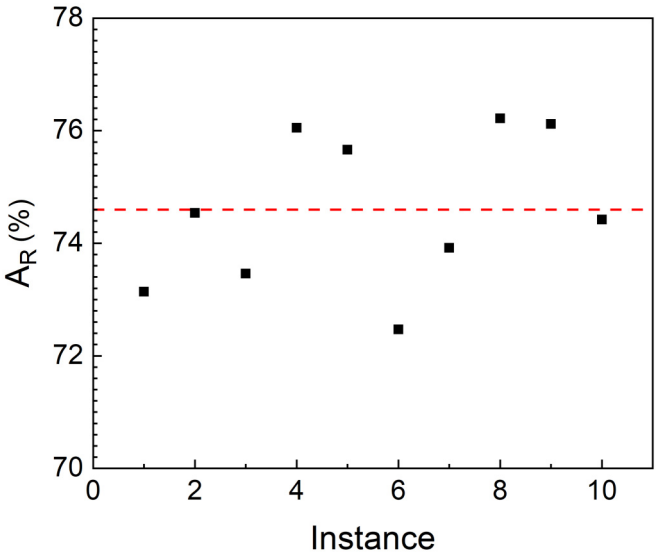

FIG. 8. Percentage acceptance ratio A_R in NADE-driven MCMC simulations for 10 instances of the spin-glass Hamiltonian with different realizations of the gaussian random couplings. The red dashed line indicates the average value. $N = 100$, $\beta = 1$, and $N_H = 64$.

dataset. This would help the NADE to accurately generalize, avoiding overfitting. Regularization methods might also improve the results. To verify that for different realizations of the gaussian couplings one obtains comparable results, we show in Fig. 8 the acceptance ratios for 10 instances of the spin-glass model Eq. (1). The relative variations are smaller than 10%, indicating that the performance of NADE does not strongly depend on the specific instance considered. Instead, the acceptance rate slightly decreases with the system size N if the number of hidden neurons $N_H = 64$ is kept fixed (see Fig. 9), reaching $R \simeq 60\%$ for $N = 225$. This suggests that more hidden neurons or larger training sets might be required when the system size is substantially increased. However, we emphasize that the simulation of $N = 225$ spins is still very efficient, meaning that moderately larger systems could be addressed even with the NADE considered here. Importantly, the predictions provided by the NADE-driven MCMC simulations always agree with the Boltzmann-distribution results for the corresponding systems size. The latter value is computed via long single spin-flip MCMC simulations, including 8×10^7 MCMC steps, verifying that simulations started from different configurations agree within the statistical uncertainties. The latter are determined by the standard blocking method [53]. This agreement confirms that the NADE probability distribution has a sizable weight in all physically relevant regions of the configuration

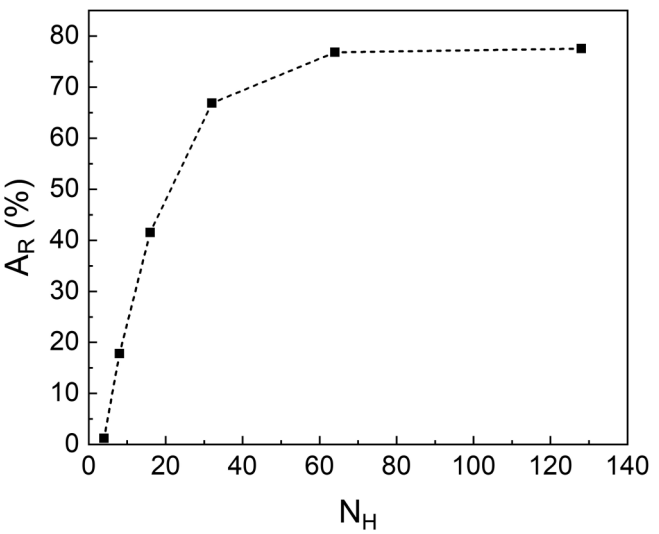

FIG. 7. Percentage acceptance ratio A_R in NADE-driven MCMC simulations, as a function of hidden-neuron number N_H. $N = 100$ and $\beta = 1$.

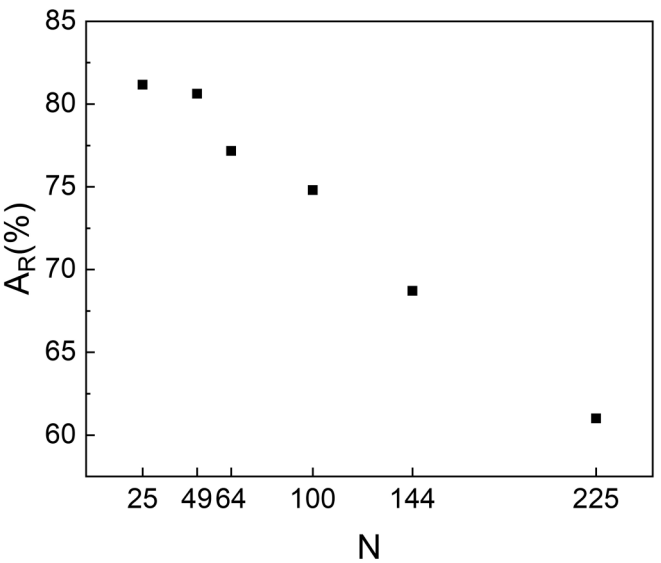

FIG. 9. Percentage acceptance ratio A_R in NADE-driven MCMC simulations as a function of the system size N. $\beta = 1$ and $N_H = 64$.

space, allowing an ergodic MCMC simulation. For small N_H one obtains larger error-bars due to the lower acceptance rate (see Fig. 4), which leads to longer correlation times. The NADE-driven MCMC simulation is unbiased even for the largest system size addressed in this work (see Fig. 6), for which the NADE less accurately mimics the Boltzmann distribution. The histogram of the configuration energies sampled during the NADE-driven MCMC simulation agrees with the one corresponding to a long single-spin flip simulation (see Fig. 5), confirming that the probability distribution learned by NADE has essentially the same support as the Boltzmann distribution.

The unsupervised learning scheme described in the previous paragraph assumes that a training dataset representative of the Boltzmann distribution can be generated. As explained above, with the single spin-flip algorithm this becomes unfeasible for inverse temperatures $\beta \gg 1$ due to the diverging correlation times. Clearly, one could adopt more sophisticated global-update MCMC algorithms as, e.g., the isoenergetic cluster updates [33,34], and then use the NADE only to accelerate the computation of physical properties in a second MCMC simulation. As we discuss in the following, a simple procedure can be implemented to efficiently simulate the low temperature regime, even without employing global-update methods to generate the training dataset. We dub this procedure sequential tempering, in analogy with the popular parallel tempering method [32]. This procedure begins by training a NADE at a relatively low inverse temperature β_0, where ergodic MCMC simulations are feasible even with single spin-flip algorithms. Next, various NADE-driven MCMC simulations are run at the inverse temperatures $\beta_s = \beta_{s-1} + \delta\beta$, where $\delta\beta$ is a small increment and $s = 1, 2, \ldots, s_{\max}$, using the NADE trained at β_{s-1} as proposal distribution. Notably, each subsequent training stage can be significantly accelerated by initializing the NADE parameters to the final values of the previous learning stage. In our numerical experiment with an Ising glass of size $N = 100$ and a NADE with $N_H = 64$ hidden neurons, we choose $\beta_0 = 0.5$, $\delta\beta = 0.1$, and $s_{\max} = 15$, allowing us to reach a low temperature corresponding to $\beta_{\max} = 2$. We emphasize that at this final temperature even 8×10^7 single spin-flip MCMC steps might not be sufficient to guarantee ergodicity. In each NADE-driven MCMC simulation, every 10th configuration is stored to train the next NADE. As discussed below, this is sufficient to suppress the residual statistical correlations. The total training-set size is again $N_t = 10^5$. Since each NADE is employed as proposal distributions at a different temperature compared to the training temperature, one should expect a decreased acceptance rate A_R. As shown in Fig. 10, in the first few steps of the sequence the acceptance rate is indeed moderately low, $R \approx 40\%$. However, it increases up to $R \approx 75\%$ in the low-temperature regime. We attribute this increase to the slower changes the Boltzmann distribution undergoes in the low-temperature regime, and to the fact that the pre-trained NADEs initialized with the parameters of the previous temperature in the sequence, more accurately learn the Boltzmann distribution at the new temperature. It is worth emphasizing again that with different disorder realizations the NADEs provide comparable results. Indeed, the data shown in Fig. 10 correspond to the average of five instances of the random

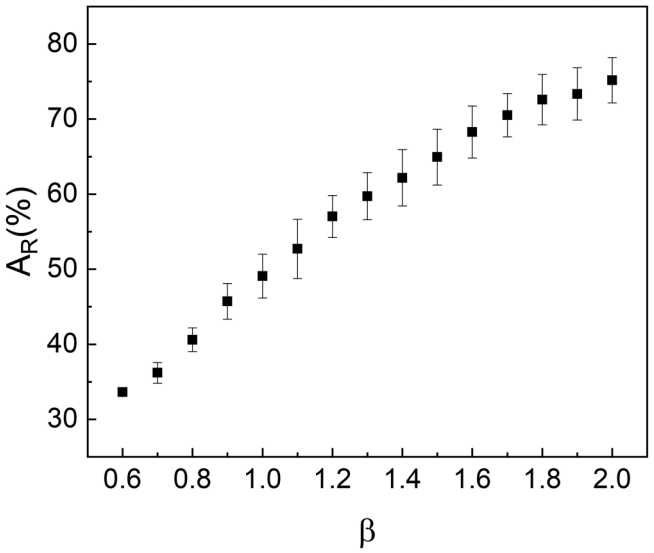

FIG. 10. Percentage acceptance ratio A_R in the NADE-driven MCMC simulations of the sequential tempering procedure, as a function of the inverse temperature β. $N = 100$ and $N_H = 64$.

couplings. The NADE-driven MCMC simulations are ergodic and efficient even in the low-temperature regime. The correlation function at $\beta = 2$ displays a sharp drop (see Fig. 11), confirming that the correlation times are minimal, as anticipated above. Instead, with the single spin-flip algorithm even after $t \sim 10^6$ MCMC steps the statistical correlations are sizable. This important finding indicates that the sequential tempering allows us to efficiently sample the low-temperature Boltzmann distribution. This statement is confirmed by the precise agreement of the five histograms shown in Fig. 12, which correspond to just as many sequences for the same instance

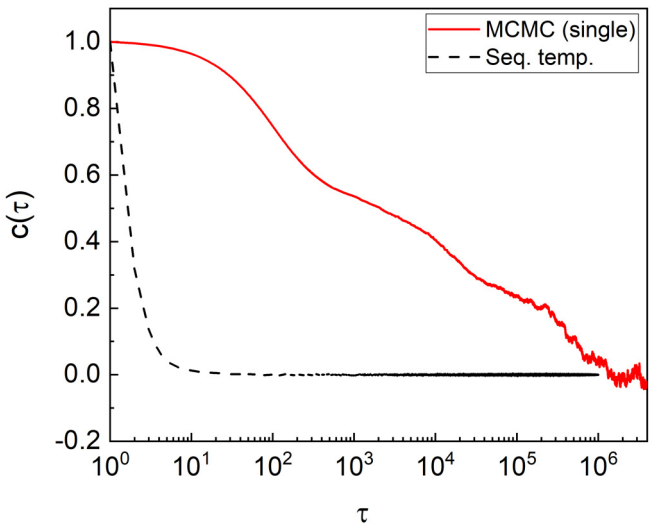

FIG. 11. Autocorrelation function $c(\tau)$ of the configuration energies at $\beta = 2$. The red line corresponds to the single spin-flip MCMC algorithm (data averaged of 25 simulations started from different initial configurations), while the black dashed line corresponds to the last NADE-driven MCMC simulation of the sequential tempering procedure. $N = 100$ and $N_H = 64$.

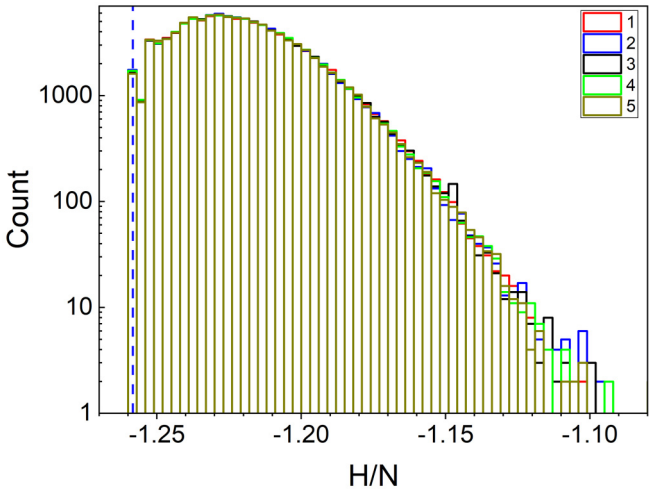

FIG. 12. Histogram of 10^5 configuration energies per spin H/N generated in the last NADE-driven MCMC simulation of the sequential tempering procedure at $\beta_{\max} = 2$. Every 10th configuration is counted. The 5 datasets correspond to just as many sequences started from different initial configurations at $\beta_0 = 0.5$, for the same realization of the spin-glass model. $N = 100$ and $N_H = 64$. The blue vertical dashed line indicates the ground-state energy for this instance of the spin-glass model, obtained from the spin-glass server [54].

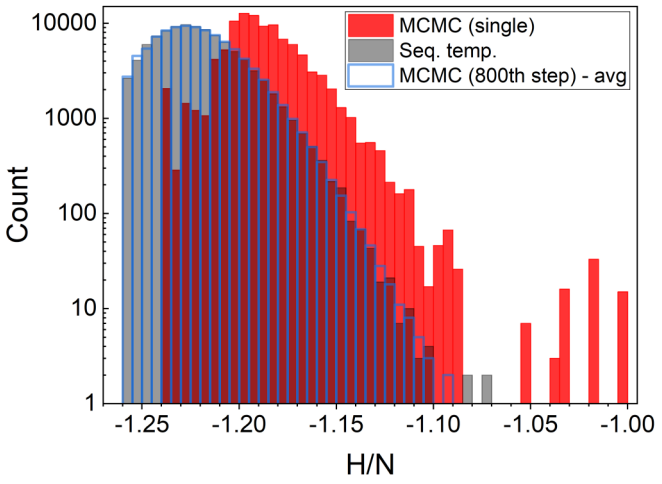

FIG. 13. Histogram of 10^5 configuration energies per spin H/N sampled at $\beta = 2$. The empty blue columns correspond to the single spin-flip MCMC algorithm (data averaged over 25 simulations started from different configurations, ran for 8×10^7 steps, every 800th configuration is counted). The gray columns correspond to the last NADE-driven MCMC simulation of the sequential tempering procedure (every 10th configuration is counted). For comparison, the full red columns indicate the first 10^5 configurations sampled via the single spin-flip MCMC algorithm. The vertical dashed line indicates the ground-state energy, obtained from the spin-glass server [54]. $N = 100$ and $N_H = 64$.

of the random couplings started from different configurations at $\beta_0 = 0.5$. To obtain a comparable histogram with the single spin-flip algorithm, we have to perform as many as 8×10^7 steps, and to average over 25 runs started from different configurations (every 800th sampled energy is counted). The comparison with the sequential-tempering histogram is shown in Fig. 13. For illustrative purposes, Fig. 13 displays also the histogram of the first 10^5 energies sampled in the initial portion of a single spin-flip MCMC run. Clearly, this dataset is highly biased, in particular in the low-energy regime where the weight is negligible. This indicates that the single spin-flip algorithm requires many more steps to reach low-energy configurations. To shed more light on the equilibration dynamics, it is useful to visualize how the configuration energy $H(t)$ evolves along the last NADE-driven MCMC simulation of the sequential tempering (see Fig. 14). The equilibration time appears to be negligible. In particular, this simulation touches a ground-state configuration after as few as $t = 31$ MCMC steps. This ground-state energy is obtained from the spin-glass server [54] at the University of Cologne, which implements an exact polynomial-time algorithm for two-dimensional lattices. This phenomenology appears to be general: repeating five sequential tempering procedures for the same instance of the random couplings but different initial configurations at β_0, or for five different instances of the random couplings, we always find that the ground-state energy is sampled within $t \lesssim 10^2$ MCMC steps. For comparison, the configuration energies obtained from a single spin-flip MCMC simulation are also shown in Fig. 14. In this specific case, a ground-state configuration is reached only after $t \sim 10^6$ MCMC steps. In fact, an (admittedly incomplete) analysis shows that with $t \sim 10^7$ single spin-flip steps only ∼50% of the times the ground-state energy is sampled. This illustrates the well-known difficulty in identifying ground-state configurations via single spin-flip Metropolis-Hastings updates.

The findings discussed above indicate that the sequential tempering procedure allows one to efficiently simulate the low-temperature regime of a short-range spin-glass model.

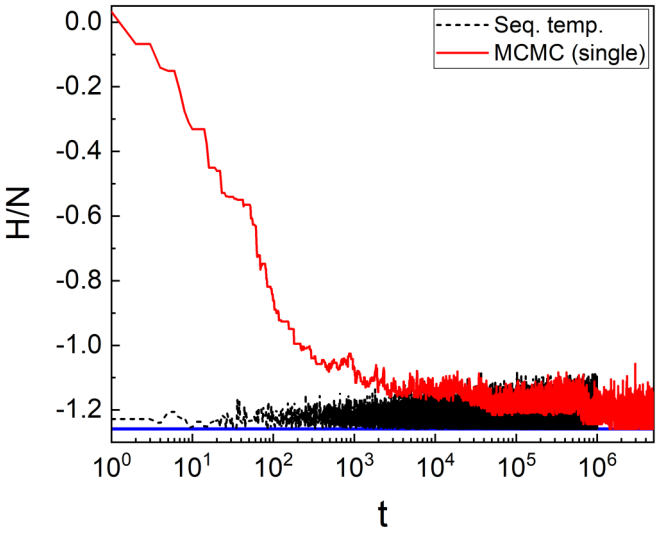

FIG. 14. Configuration energy per spin H/N as a function of the number of MCMC steps t, at $\beta = 2$. Dashed black line corresponds to the last NADE-driven MCMC simulation of the sequential tempering procedure. Solid red line corresponds to the single spin-flip MCMC simulation. The horizontal blue line indicates the ground-state energy, obtained from the spin-glass server [54]. $N = 100$ and $N_H = 64$.

Furthermore, they suggest that NADEs could be employed to boost the efficiency of stochastic optimization methods for binary optimization problems. Indeed, heuristic optimization methods like simulated annealing [31] exploit MCMC algorithms to explore the solution space. Generic binary optimization problems can be mapped to disordered Ising Hamiltonians analogous to Eq. (1). The long correlation times discussed above also occur when tackling the optimization problem, often preventing the optimal solution from being found. Our results point to the use of NADEs trained with a sequential tempering procedure as the engine for simulated-annealing optimizations. It is also worth mentioning that NADEs could also be trained from data produced using physical quantum annealers, such as the devices commercialized by D-Wave Systems (see, e.g, Ref. [55]). These devices are special-purpose adiabatic quantum computers designed to solve quadratic unconstrained optimization problems. They allow sampling low-energy configurations of programmable Ising Hamiltonians, with the goal to identify the optimal solution. The trained NADE could be employed to drive simulated-annealing optimizations. This might help eliminating the effect of the inevitable noise present in the values of the model parameters programed on the physical device.

IV. CONCLUSIONS

We have shown how to use an autoregressive generative neural network, namely, a NADE [35], to boost the efficiency of Markov chain Monte Carlo (MCMC) simulations of a two-dimensional Ising Hamiltonian with nearest-neighbor gaussian couplings. This model Hamiltonian is an archetype of spin-glass theory. The NADEs have been trained within an unsupervised learning scheme, which consists of minimizing the Kullback-Leibler divergence with respect to a dataset of configurations generated via standard MCMC simulations. Our analysis quantified how accurately a NADE can mimic the Boltzmann distribution of a spin-glass model, depending on the number of hidden neurons and the number of visible spins. The trained NADEs have then been used as proposal distributions in smart Monte Carlo simulations based on the Metropolis-Hastings algorithm. This allowed us to implement efficient global updates, whose computational cost is linear in the system size. In particular, we have implemented a sequential tempering procedure. Starting from a higher temperature, the procedure reaches the low temperature regime via a sequence of MCMC simulations and training stages performed at successively lower temperatures. This allowed us to run efficient MCMC simulations with very short autocorrelation times, even in regimes where computing thermodynamic properties with standard local algorithms is difficult, if not totally unfeasible. Furthermore, it has been verified that at low temperatures the NADE-driven MCMC simulations quickly sample ground-state configurations. This result suggests to employ autoregressive neural networks in combination with simulated annealing or other stochastic methods to solve binary optimization problems.

Our work complements other recent investigations on the use of autoregressive models for classical mechanics problems [20,24]. We described the use of an unsupervised learning scheme, instead of reinforcement learning, and we tackled a short-range spin-glass Hamiltonian instead of clean systems or mean-field infinite-range disordered models. It is worth stressing that the unsupervised learning scheme could be combined with any of the sophisticated MCMC techniques that have been developed over the years to simulate spin glasses. Specifically, the global-update methods could be employed to efficiently generate training datasets. The NADE could then be used to speed up the computation of physical properties, including observables that explicitly depend on the partition function [24]. The results we have presented in this article are encouraging, and they raise ambition to further investigations on different spin-glass models and on observables other than the average energy and the energy distribution as, e.g., the spin-spin correlations or the Edwards-Anderson order parameter. These investigations could clarify if in certain circumstances a NADE might fail to accurately mimic the Boltzmann distribution, possibly leading to very low acceptance ratios or nonergodic simulations. In such cases, one could resort to deeper generative neural networks such as PixelCNN [47], variational autoencoder [56], and generative adversarial networks [57]. Moreover, further studies should be devoted to the use of NADEs for binary optimization problems. Interesting candidates are the hard instances of spin-glass problems with planted solutions generated with the algorithms of Refs. [58,59]. We leave these endeavors to future studies.

ACKNOWLEDGMENTS

The authors thank I. Murray, G. Carleo, and F. Ricci-Tersenghi for useful discussions. Financial support from the FAR2018 project titled "Supervised machine learning for quantum matter and computational docking" of the University of Camerino and from the Italian MIUR under Project No. PRIN2017 CEnTraL 20172H2SC4 is gratefully acknowledged. S.P. also acknowledges the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support. M.V.M. gratefully acknowledges the Visiting Professorship program at the University of Camerino that facilitated the collaboration in this work.

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[2] A. Rocchetto, E. Grant, S. Strelchuk, G. Carleo, and S. Severini, Learning hard quantum distributions with variational autoencoders, Npj Quantum Inf. **4**, 28 (2018).

[3] I. A. Luchnikov, A. Ryzhov, P.-J. Stas, S. N. Filippov, and H. Ouerdane, Variational autoencoder reconstruction of complex many-body physics, Entropy **21**, 1091 (2019).

[4] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, Cogn. Sci. **9**, 147 (1985).

[5] G. Torlai and R. G. Melko, Learning thermodynamics with Boltzmann machines, Phys. Rev. B **94**, 165134 (2016).

[6] L. Huang and L. Wang, Accelerated Monte Carlo simulations with restricted Boltzmann machines, Phys. Rev. B **95**, 035105 (2017).

[7] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Self-learning monte carlo method, Phys. Rev. B **95**, 041101(R) (2017).

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

[9] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[10] A. Nagy and V. Savona, Variational quantum Monte Carlo Method with a Neural-Network Ansatz for Open Quantum Systems, Phys. Rev. Lett. **122**, 250501 (2019).

[11] M. J. Hartmann and G. Carleo, Neural-Network Approach to Dissipative Quantum Many-Body Dynamics, Phys. Rev. Lett. **122**, 250502 (2019).

[12] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Variational neuraL-Network Ansatz for Steady States in Open Quantum Systems, Phys. Rev. Lett. **122**, 250503 (2019).

[13] N. Yoshioka and R. Hamazaki, Constructing neural stationary states for open quantum many-body systems, Phys. Rev. B **99**, 214306 (2019).

[14] G. Torlai and R. G. Melko, Neural Decoder for Topological Codes, Phys. Rev. Lett. **119**, 030501 (2017).

[15] S. Pilati, E. M. Inack, and P. Pieri, Self-learning projective quantum Monte Carlo simulations guided by restricted Boltzmann machines, Phys. Rev. E **100**, 043301 (2019).

[16] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, Nat. Mach. Intell. **1**, 155 (2019).

[17] K. Gregor, I. Danihelka, A. Mnih, C. Blundell, and D. Wierstra, Deep autoregressive networks, in *Proceedings of the International Conference on Machine Learning* (ACM, New York, NY, 2014), pp. 1242–1250.

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

[19] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Recurrent neural network wave functions, arXiv:2002.02973 (2020).

[20] D. Wu, L. Wang, and P. Zhang, Solving Statistical Mechanics Using Variational Autoregressive Networks, Phys. Rev. Lett. **122**, 080602 (2019).

[21] D. Sherrington and S. Kirkpatrick, Solvable Model of a Spin-Glass, Phys. Rev. Lett. **35**, 1792 (1975).

[22] M. Mézard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*, Vol. 9 (World Scientific Publishing Company, Singapore, 1987).

[23] G. Papamakarios and I. Murray, Distilling intractable generative models, in *Proceedings of the Probabilistic Integration Workshop at Neural Information Processing Systems* (2015), https://homepages.inf.ed.ac.uk/imurray2/pub/15distill/distill.pdf.

[24] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel, Asymptotically unbiased estimation of physical observables with neural samplers, Phys. Rev. E **101**, 023304 (2020).

[25] K. Binder and A. P. Young, Spin glasses: Experimental facts, theoretical concepts, and open questions, Rev. Mod. Phys. **58**, 801 (1986).

[26] D. Chowdhury, *Spin Glasses and Other Frustrated Systems* (Princeton University Press, Princeton, NJ, 2014).

[27] E. Marinari, G. Parisi, F. Ricci-Tersenghi, J. J. Ruiz-Lorenzo, and F. Zuliani, Replica symmetry breaking in short-range spin glasses: Theoretical foundations and numerical evidences, J. Stat. Phys. **98**, 973 (2000).

[28] S. Franz and F. Ricci-Tersenghi, Ultrametricity in three-dimensional Edwards-Anderson spin glasses, Phys. Rev. E **61**, 1121 (2000).

[29] W. Wang, J. Machta, H. Munoz-Bauza, and H. G. Katzgraber, Number of thermodynamic states in the three-dimensional Edwards-Anderson spin glass, Phys. Rev. B **96**, 184417 (2017).

[30] W. Wang, J. Machta, and H. G. Katzgraber, Evidence against a mean-field description of short-range spin glasses revealed through thermal boundary conditions, Phys. Rev. B **90**, 184412 (2014).

[31] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science **220**, 671 (1983).

[32] K. Hukushima and K. Nemoto, Exchange Monte Carlo method and application to spin glass simulations, J. Phys. Soc. Jpn. **65**, 1604 (1996).

[33] J. Houdayer, A cluster Monte Carlo algorithm for 2-dimensional spin glasses, Eur. Phys. J. B **22**, 479 (2001).

[34] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, Efficient Cluster Algorithm for Spin Glasses in Any Space Dimension, Phys. Rev. Lett. **115**, 077201 (2015).

[35] H. Larochelle and I. Murray, The neural autoregressive distribution estimator, in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 15, edited by G. Gordon, D. Dunson, and M. Dudí (PMLR, Fort Lauderdale, FL, 2011), pp. 29–37.

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

[37] J. Thijssen, *Computational Physics* (Cambridge University Press, UK, 2007).

[38] R. H. Swendsen and J.-S. Wang, Nonuniversal Critical Dynamics in Monte Carlo Simulations, Phys. Rev. Lett. **58**, 86 (1987).

[39] U. Wolff, Collective Monte Carlo Updating for Spin Systems, Phys. Rev. Lett. **62**, 361 (1989).

[40] N. Prokof'ev and B. Svistunov, Worm Algorithms for Classical Statistical Models, Phys. Rev. Lett. **87**, 160601 (2001).

[41] Y. Deng, T. M. Garoni, and A. D. Sokal, Dynamic Critical Behavior of the Worm Algorithm for the Ising Model, Phys. Rev. Lett. **99**, 110601 (2007).

[42] F. Barahona, On the computational complexity of Ising spin glass models, J. Phys. A **15**, 3241 (1982).

[43] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. **14**, 1771 (2002).

[44] A. Fischer and C. Igel, An introduction to restricted Boltzmann machines, in *Proceedings of the Iberoamerican Congress on Pattern Recognition* (Springer, Berlin, 2012), pp. 14–36.

[45] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, Neural autoregressive distribution estimation, J. Mach. Learn. Res. **17**, 1 (2016).

[46] T. Vogel and D. Perez, Towards an Optimal Flow: Density-of-States-Informed Replica-Exchange Simulations, Phys. Rev. Lett. **115**, 190602 (2015).

[47] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu, Pixel recurrent neural networks, arXiv:1601.06759 (2016).

[48] K. Binder and K. Schröder, Phase transitions of a nearest-neighbor Ising-model spin glass, Phys. Rev. B **14**, 2142 (1976).

[49] H. Rieger, L. Santen, U. Blasum, M. Diehl, M. Jünger, and G. Rinaldi, The critical exponents of the two-dimensional Ising spin glass revisited: Exact ground-state calculations and Monte Carlo simulations, J. Phys. A **29**, 3939 (1996).

[50] N. Kawashima and H. Rieger, Finite-size scaling analysis of exact ground states for ±j spin glass models in two dimensions, Europhys. Lett. **39**, 85 (1997).

[51] A. K. Hartmann and A. P. Young, Lower critical dimension of Ising spin glasses, Phys. Rev. B **64**, 180404(R) (2001).

[52] A. C. Carter, A. J. Bray, and M. A. Moore, Aspect-Ratio Scaling and the Stiffness Exponent θ for Ising Spin Glasses, Phys. Rev. Lett. **88**, 077201 (2002).

[53] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, New York, NY, 1999).

[54] The spin glass ground state server, https://cs.unikoeln.de/ls-juenger/research/sgs/sgs.html.

[55] S. Boixo, T. F. Rønnow, S. V. Isakov, Z. Wang, D. Wecker, D. A. Lidar, J. M. Martinis, and M. Troyer, Evidence for quantum annealing with more than one hundred qubits, Nat. Phys. **10**, 218 (2014).

[56] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv:1312.6114 (2013).

[57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, 2014), pp. 2672–2680.

[58] W. Wang, S. Mandrà, and H. G. Katzgraber, Patch-planting spin-glass solution for benchmarking, Phys. Rev. E **96**, 023312 (2017).

[59] D. Perera, F. Hamze, J. Raymond, M. Weigel, and H. G. Katzgraber, Computational hardness of spin-glass problems with tile-planted solutions, Phys. Rev. E **101**, 023316 (2020).