Quantum neural networks to simulate many-body quantum systems

Bartłomiej Gardas,[1,2,3,*] Marek M. Rams,[3] and Jacek Dziarmaga[3]
[1]*Theoretical Division, LANL, Los Alamos, New Mexico 87545, USA*
[2]*Institute of Physics, University of Silesia, 40-007 Katowice, Poland*
[3]*Marian Smoluchowski Institute of Physics, Jagiellonian University, Łojasiewicza 11, 30-348 Kraków, Poland*

(Received 26 May 2018; revised manuscript received 30 August 2018; published 26 November 2018)

We conduct experimental simulations of many-body quantum systems using a *hybrid* classical-quantum algorithm. In our setup, the wave function of the transverse field quantum Ising model is represented by a restricted Boltzmann machine. This neural network is then trained using variational Monte Carlo assisted by a D-wave quantum sampler to find the ground-state energy. Our results clearly demonstrate that already the first generation of quantum computers can be harnessed to tackle nontrivial problems concerning physics of many-body quantum systems.

DOI: 10.1103/PhysRevB.98.184304

I. INTRODUCTION

Building a universal quantum computer is a holy grail of modern sciences [1]. Such a machine offers *necessary* capabilities allowing one to simulate *highly* entangled quantum systems [2,3]. In contrast, a universal Turing machine [4] realizing classical computation can only simulate *slightly* entangled quantum states [5].

Although quantum supremacy is yet to be demonstrated [6], many important properties of quantum systems can be captured by artificial intelligence [7] and neural networks, in particular [8,9]. The so-called quantum neural states provide a novel ansatz to represent the wave function of a many-body quantum system [10,11]. Such neural networks can be *taught* using various techniques—most notably the variational Monte Carlo [12–14]. In general, however, sampling the state space, which is the key ingredient of all Monte Carlo methods, cannot be executed efficiently by any classical algorithm [15,16]. Hence, there exist natural limitations to any classical algorithm that aims to teach the network about quantum systems.

It is well known that these limitations can be broken by harnessing the power of a quantum sampler [17,18]. It is needless to say that the existing annealers are far from perfect [19–21]. Nevertheless, they can be turned into quantum samplers rather easily. This provides an ideal playground for testing a new generation of *hybrid* classical-quantum algorithms [22].

In this paper, we investigate to what extent such algorithms can run on the existing hardware [23]. Our purpose is to demonstrate that already the first generation of quantum computers can, in fact, *assist* in simulations of simple yet nontrivial many-body quantum systems. A similar conceptual idea has been applied *very* recently to investigate quantum phase transition in many-body systems [24,25], risk analysis [26], and quantum circuits diagonalizing *small* quantum Hamiltonians [27,28]. In our setup, however, the wave function of the transverse quantum Ising model is represented by a restricted Boltzmann machine [29]. This neural network is then trained in an *unsupervised* manner to find the ground-state energy. The learning process is assisted by a D-wave chip as explained below.

II. PRELIMINARIES

A. Quantum neural states

We begin by writing a many-body quantum state $|\psi\rangle$ using restricted Boltzmann machine (RBM) as a wave-function ansatz [29,30],

$$|\Phi\rangle = \sum_{\boldsymbol{v}} \Phi(\boldsymbol{v})|\boldsymbol{v}\rangle, \quad \Phi(\boldsymbol{v}) = \sum_{\boldsymbol{h}} e^{-\phi(\boldsymbol{v},\boldsymbol{h})}. \tag{1}$$

Here $\boldsymbol{v} = (v_1, \ldots, v_N)$ is a collection of physical degrees of freedom called *visible* neurons,

$$\phi(\boldsymbol{v}, \boldsymbol{h}) = \boldsymbol{a} \cdot \boldsymbol{v} + \boldsymbol{b} \cdot \boldsymbol{h} + \boldsymbol{h} \cdot \boldsymbol{W} \cdot \boldsymbol{v}, \tag{2}$$

and $\boldsymbol{h} = (h_1, \ldots, h_M)$ are *hidden* neurons, see Fig. 2. This network is fully specified by the weights $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{W}$ which are determined during the learning stage. Surprisingly, $M = \alpha N$ for moderate α, say <4, is often sufficient to accurately calculate the ground-state properties of many important physical systems [29].

Our objective here is to train the quantum neural state using a D-wave annealer to find the ground-state energy E of the transverse field quantum Ising model [31–33],

$$H = -h \sum_i \hat{\tau}_i^x - \sum_{\langle i,j \rangle} \hat{\tau}_i^z \hat{\tau}_j^z. \tag{3}$$

Above, $\langle i, j \rangle$ denotes nearest neighbors and τ_i^x, τ_i^x are the standard Pauli spin operators [34]. Periodic boundary conditions are assumed. We consider both one-dimensional (1D) and two-dimensional (2D) lattices. In the former case, the ground-state energy can be calculated exactly [35]. In the latter, we use density-matrix renormalization-group algorithm [36,37] to obtain its sufficient approximation that will

*bartek.gardas@gmail.com

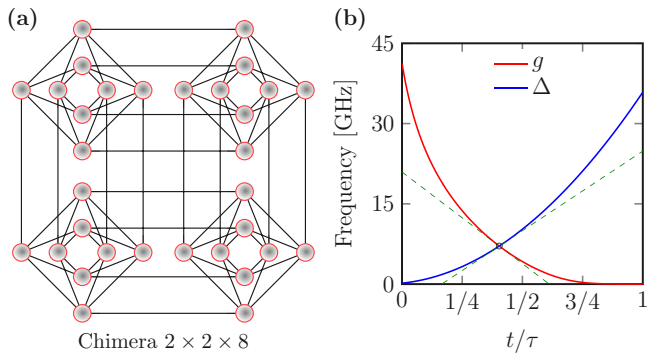

FIG. 1. *D-wave processor specification*. (a) An example of the chimera architecture comprising a 2×2 grid consisting of clusters (units cells) of eight qubits each. (b) A typical annealing schedule where the annealing time reads τ.

serve as a reference value. This allows us to assess the robustness of our method and at the same time benchmark the annealer [38]. To perform a quantum sampling we use both the newest $2000Q$ chip and its predecessor DW2X [39].

Henceforward, we also assume that all weights $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{W}$ are *real*. Thus, instead of Eq. (1), one can use the following ansatz to represent the ground state of the Ising model (3):

$$\Psi(\boldsymbol{v}) = \sqrt{\sum_{\boldsymbol{h}} e^{-\phi(\boldsymbol{v},\boldsymbol{h})}} = \sqrt{\Phi(\boldsymbol{v})}. \tag{4}$$

As before, $\phi(\boldsymbol{v}, \boldsymbol{h})$ is given by Eq. (2). This way, the *quantum* probability distribution,

$$\rho(\boldsymbol{v}) = \frac{|\Psi(\boldsymbol{v})|^2}{\sum_{\boldsymbol{v}'} |\Psi(\boldsymbol{v}')|^2} \tag{5}$$

can be represented by a RBM and as such can be sampled using a quantum annealer. This is the key insight into all conceptual ideas we outline in this article.

B. Adiabatic quantum computing

During quantum annealing a many-body system is to be evolved from the ground state of a problem Hamiltonian $\mathcal{H}_0$ to the ground state of a final Hamiltonian $\mathcal{H}(\tau)$ that encodes the solution for the problem of interest [40–42]. The dynamics of a D-wave chip is governed by the time-dependent Hamiltonian [17,43],

$$\begin{aligned} \mathcal{H}(t)/h &= -g(t) \sum_{i \in \mathcal{V}} \hat{\sigma}_i^x - \Delta(t)\mathcal{H}_0, \\ \mathcal{H}_0 &= \sum_{i \in \mathcal{V}} B_i \hat{\sigma}_i^z + \sum_{(i,j) \in \mathcal{E}} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z, \end{aligned} \tag{6}$$

where $\hat{\sigma}_i^x$ and $\hat{\sigma}_i^z$ are the Pauli spin operators. Here, $\mathcal{H}_0$ is defined on a chimera graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [44], see Fig. 1(a). *Dimensionless* couplers J_{ij} and biases B_i can be controlled by the users but only within a predefined range. For example, on the $2000Q$ device we have $|J_{ij}| \leqslant 1$ and $|B_i| \leqslant 2$.

Presumably, the Hamiltonian $\mathcal{H}(t)$ varies slowly whereas $\Delta(t)$ is changed from initial $\Delta(0) \approx 0$ to large final $\Delta(\tau)$, whereas $g(t)$ varies from large initial $g(0)$ to final $g(\tau) \approx 0$ [cf. Fig. 1(b)]. As a result, under optimal conditions, the system remains in its ground state. Thus, the desired solution—encoded in eigenvalues σ_i of $\hat{\sigma}_i^z$—can be extracted through a measurement of the final state. It is worth emphasizing that with D-wave annealers one can only carry a suitable measurement in the computational z basis ($|\uparrow\rangle, |\downarrow\rangle$). This essentially excludes any possibility to measure the ground-state energy of the transverse Ising model (3) *directly*, even if measurements for intermediate values of Δ and g were available.

Furthermore, since no real hardware is completely isolated from its environment, the outcome of such experiment will be distributed according to some *temperature*-dependent probability distribution $p(\boldsymbol{\sigma})$ [45]. Relaxing the annealer to the equilibrium, one can approximate $p(\boldsymbol{\sigma})$ by the *classical* Boltzmann distribution [46],

$$p(\boldsymbol{\sigma}) \sim e^{-\beta E_\tau(\boldsymbol{\sigma})}, \tag{7}$$

where the energy function reads

$$E_\tau(\boldsymbol{\sigma}) = -\sum_{i \in \mathcal{V}} B_i \sigma_i - \sum_{(i,j) \in \mathcal{E}} J_{ij} \sigma_i \sigma_j. \tag{8}$$

The time to complete the annealing cycle is denoted by τ. The *effective* inverse temperature $\beta = h\,\Delta(\tau^*)\beta_{\text{chip}}/k_B$ is affected by many factors, including the specific values of the control parameters J_{ij} and B_i [47]. Here, $\tau^* \leqslant \tau$ is the so-called freeze-out time, and $1/\beta_{\text{chip}}$ denotes the chip's operational temperature [48]. Note, β is *a priori* unknown, and it can only be determined on a case-by-case basis [49]. In this paper, however, we do not attempt to estimate the function $\beta(J_{ij}, B_i, \tau^*)$. We rather try to modify the sampling algorithm to account for its possible variation with the values of the parameters.

In an annealer with the sufficient connectivity between qubits, there would be a one-to-one mapping between the set of σ_i and the two sets of visible and hidden neurons $\boldsymbol{\sigma} = [\boldsymbol{v}, \boldsymbol{h}]$. Accordingly, every nonzero J_{ij} would be identified with W_{ij}/β between the visible and the hidden neuron and the biases $\boldsymbol{B} = [\boldsymbol{a}, \boldsymbol{b}]/\beta$. In practice,

$$J_{ij} = W_{ij}/\beta_{\text{x}}, \tag{9}$$

where β_{x} is an *estimation* of the inverse temperature β.

III. UNSUPERVISED LEARNING

A. Variational Monte Carlo

Training neural networks can be tedious. Moreover, due to their topology, RBMs may be highly susceptible to small changes in the variational parameters. Their adjustments can further propagate throughout the network causing even larger changes in the wave function. To mitigate these problems we use a stochastic reconfiguration, a method that is widely used in the variational Monte Carlo [12]. At each iteration, the network weights $\boldsymbol{w} = [\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{W}]$ are refined according to

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \gamma_k \boldsymbol{x}_k, \quad \boldsymbol{S}^{(k)} \boldsymbol{x}_k = \boldsymbol{F}^{(k)}, \tag{10}$$

where a non-negative definite covariance matrix reads

$$\boldsymbol{S}_{ij} = \langle\langle D_i^* D_j \rangle\rangle_\rho - \langle\langle D_i^* \rangle\rangle \langle\langle D_j \rangle\rangle_\rho, \tag{11}$$

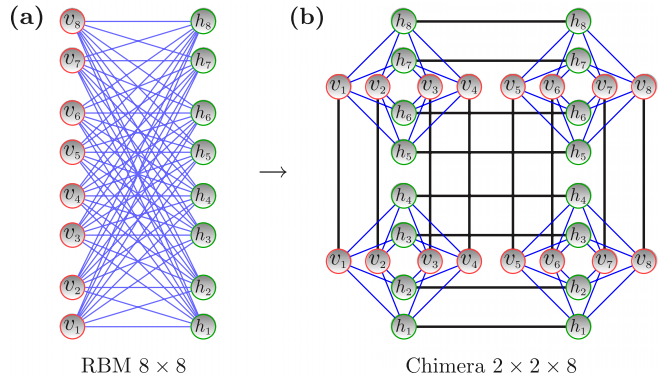

FIG. 2. *Restricted Boltzmann machine embedded on a D wave.* (a) Graphical representation of the neural network (2) with eight visible and eight hidden neurons. (b) The same network embedded on the chimera graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ of size $2 \times 2 \times 8$. Strong ferromagnetic couplings (thick lines) "glue" qubits in different unit cells to represent single neurons.

and the so-called *forces* are given by

$$\boldsymbol{F}_j = \langle\langle E D_j^* \rangle\rangle_\rho - \langle\langle E \rangle\rangle_\rho \langle\langle D_j^* \rangle\rangle_\rho. \tag{12}$$

Double brackets $\langle\langle \cdot \rangle\rangle_\rho$ indicate averages with respect to the distribution in Eq. (5), γ_k is the learning rate. Finally,

$$D_i = \frac{1}{\Psi(\boldsymbol{v})} \frac{\partial}{\partial w_i} \Psi(\boldsymbol{v}), \quad E_{\text{loc}} = \frac{\langle \boldsymbol{v}|H|\Psi\rangle}{\Psi(\boldsymbol{v})} \tag{13}$$

denote the gradients and local energy, respectively [29].

B. Sampling

Usually, the importance sampling is performed using the Metropolis-Hastings algorithm [50]. In this paper, we employ the newest generations of D-wave samplers to calculate covariance matrix (11) and forces (12) at each iteration [51]. The remaining part of the algorithm is executed on a classical processing unit: CPU or GPU [52].

To this end, we first rewrite the ansatz (4) as

$$\Psi(\boldsymbol{v}) = e^{\boldsymbol{a}\cdot\boldsymbol{v}/2} \left[\prod_{j=1}^{M} 2 \cosh(b_j + \boldsymbol{W}_j \cdot \boldsymbol{v}) \right]^{1/2}, \tag{14}$$

where we explicitly traced out hidden variables. This is possible due to the lack of intralayer interactions between hidden neurons [cf. Fig. 2(b)]. Now, all derivatives in Eq. (13) can be expressed using visible neurons only [29],

$$\frac{1}{\Psi(\boldsymbol{v})} \frac{\partial}{\partial p} \Psi(\boldsymbol{v}) = \frac{1}{2} \times \begin{cases} v_i, & p = a_i, \\ \tanh(\theta_j), & p = b_j, \\ v_i \tanh(\theta_j), & p = W_{ij}, \end{cases} \tag{15}$$

where we introduced $\theta_j = b_j + \boldsymbol{W}_j \cdot \boldsymbol{v}$. Similarly, the local energy can be simplified to take the form

$$E_{\text{loc}} = -h \sum_i \frac{\Psi(\bar{\boldsymbol{v}}_i)}{\Psi(\boldsymbol{v})} - \sum_{\langle i,j \rangle} v_i v_j, \tag{16}$$

where $\bar{\boldsymbol{v}}_i$ denotes a vector $\boldsymbol{v}$ with ith spin flipped and Ψ is given by Eq. (14). Finally, to compute $\langle\langle \cdot \rangle\rangle_\rho$ using samples gathered from a quantum annealer, we note that

$$\begin{aligned} \langle\langle f \rangle\rangle_\rho &= \sum_{\boldsymbol{v}} \rho(\boldsymbol{v}) f(\boldsymbol{v}) \approx \sum_{\boldsymbol{v},\boldsymbol{h}} p(\boldsymbol{v},\boldsymbol{h}) f(\boldsymbol{v}) \\ &= \sum_{\boldsymbol{\sigma}} p(\boldsymbol{\sigma}) f(\boldsymbol{\sigma}) \approx \frac{1}{N_s} \sum_{i=1}^{N_s} f(\boldsymbol{\sigma}_i), \end{aligned} \tag{17}$$

where the *bare* output of the D-wave annealer $\boldsymbol{\sigma} = [\boldsymbol{v}, \boldsymbol{h}]$ encodes both hidden and visible neurons. Here, N_s is the number of samples. The first approximation is true under a proper embedding as long as $p(\boldsymbol{v}, \boldsymbol{h})$ is *close* to the Boltzmann distribution. The second one holds for sufficiently large N_s (in practice $N_s \sim 10^4$) provided that all samples $\boldsymbol{\sigma}_i$ are distributed according to $p(\boldsymbol{\sigma}_i)$. Note, one quantum annealing corresponds to one sample. This means that gathering samples requires running the annealer over and over again. However, in practice this is executed in "chunks," and it is very fast (microsecond to millisecond depending on the annealing time τ) and efficient.

A possible advantage of using a D-wave computer is that it can sample both visible and hidden neurons simultaneously. In principle, this allows to calculate the gradients directly using the ansatz in Eq. (4) even try to extend it to *deep* Boltzmann machine [53]. We leave this for future investigations.

C. Embedding RBM on a D-wave

Unfortunately, a RBM *cannot* be directly placed on the D-wave chip due to limited (sparse) connectivity between qubits [46]. However, this problem can be circumvented using suitable embedding [54]. The idea is to emulate a single neuron using available (local) connections between physical qubits on the chip. To this end, a strong ferromagnetic couplings is set between the latter qubits. We stress that even a proper embedding can break during the annealing. However, for small enough RBM's weights the frequency at which they do break should not be too high (in practice ∼0.2). In that case, the *majority vote* or a similar method can be invoked to correct the sample [55].

Figure 2(b) shows a chimera graph with four unit cells. Each unit cell has eight qubits—with full connectivity between the horizontal and the vertical ones and can represent as many neurons. In order to construct, for instance, a RBM with eight visible and eight hidden neurons, four unit cells with suitable qubits glued together are necessary. That amounts to 32 physical qubits. In this embedding, all qubits connected vertically (horizontally) represent a visible (hidden) neurons [54]. Consequently, the maximum number of neurons that the chimera graph C_n, consisting of $n \times n \times 8$ qubits, can represent is $L_{\max} = 8n$. For example, all 2048 qubits on the $2000Q$ chips can be utilized to build, e.g., a RBM with 64 visible and 64 hidden neurons.

IV. RESULTS

For the sake of simplicity and without loss of generality, we only consider RBMs with the same number of hidden and visible neurons, i.e., $M = N$ ($\alpha = 1$). A classical Metropolis-Hastings sampling technique has no problems finding the ground state. The relative error of the solution $\delta E = |(E - E_{\text{exact}})/E_{\text{exact}}|$ is on the order of 10^{-4}, see Fig. 3(a). The same

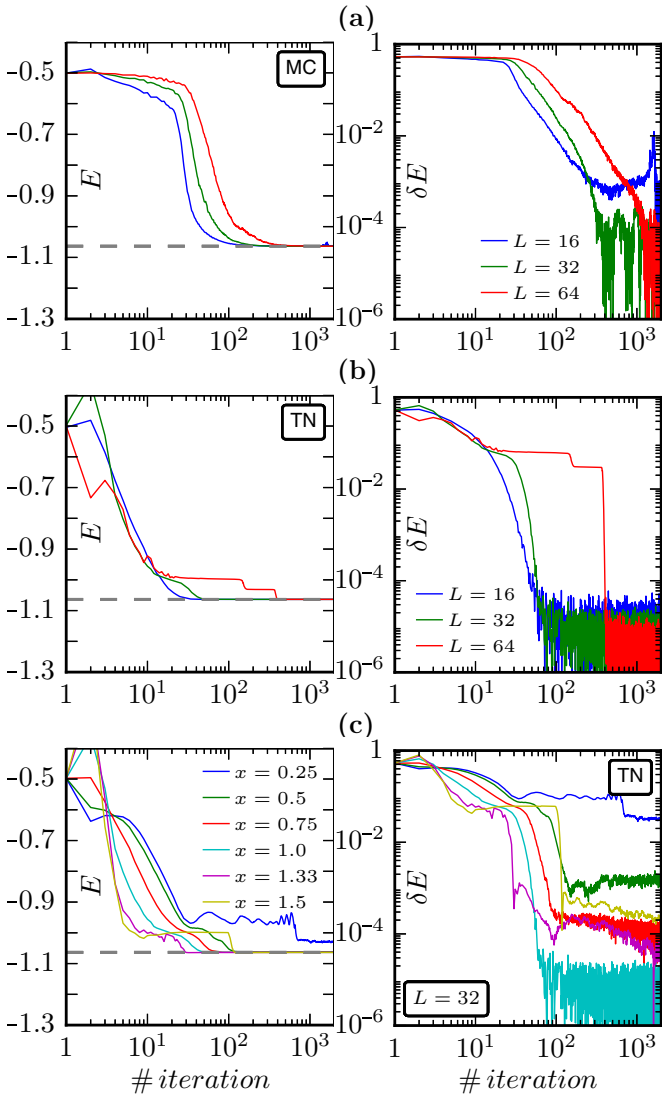

FIG. 3. The ground-state energy per spin E for the transverse field quantum Ising model (3) in 1D obtained for different system sizes. The dashed lines indicate the exact values. Sampling was carried out using Monte Carlo with $\gamma = 0.2$ in panels (a) and tensor networks algorithm with $\gamma = 0.05$ in (b). They serve as a reference point for the results in Fig. 4. In panels (c), we show the influence of incorrect inverse temperature estimation $x = \beta/\beta_x$ on the results.

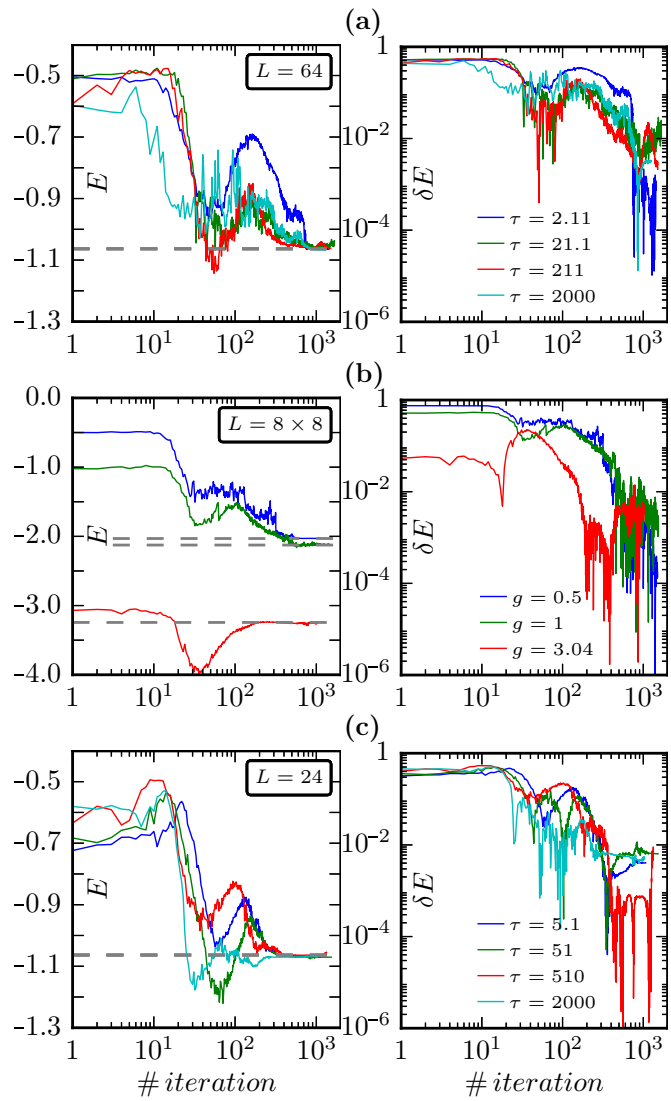

FIG. 4. The ground-state energy per spin E for the transverse field quantum Ising models (3) in 1D and 2D. The dashed lines indicate the exact values. Sampling was executed using D-wave annealers: $2000Q$ in (a) and (b) and DW2X in (c). δE shows the relative energy error reached. We set $h = 0.5$ for the 1D system in panels (a) and (c), and $h = 0.5$, 1, and $3.044 \simeq h_c$ for the 2D system in (b), where $\tau = 20$. The annealing time τ is measured in microseconds. Besides, $\alpha = 1$, $\gamma = 0.2$, and $N_s = 10^4$.

conclusion is reached using a more sophisticated sampling technique based on tensor networks algorithms, see Fig. 3(b). To that end we used a simpler—matrix product state-based—variant of the procedures described in Ref. [56].

We collect the results which were obtained running the *hybrid* algorithm in Fig. 4. For sampling, we used two generations of D-wave annealers: $2000Q$ and DW2X. As one can see in Fig. 4, both of them were capable of finding the correct ground-state energy. The solutions reached are, nonetheless, less accurate with δE on the order of 10^{-3}–10^{-2}. This can be expected from the real physical device which is prone to errors [39]. One can also expect those results to improve with each new generation of quantum computers.

There are many factors that can contribute to the errors and limited precision [39]. To mitigate some of them, the D-wave solver offers postprocessing optimization options. The idea is to bring $p(\boldsymbol{\sigma})$ to the Boltzmann distribution (8) as close as possible, ideally at some predefined inverse temperature β. However, in our minimalistic approach we did not use any of those options. Instead, we allowed the algorithm to change the initial inverse temperature so that it could converge to the correct solution, see Fig. 5. To that end we randomly increased or decreased the effective temperature β_x when the energy between subsequent iterations was growing. Given the lack of any comprehensive theory explaining how D-wave annealers work, this approach seems optimal for the current purpose. The idea can be further motivated by numerical simulations. Figure 3(c) shows the robustness of the algorithm against the variability of β_x. Surprisingly, the correct solution can

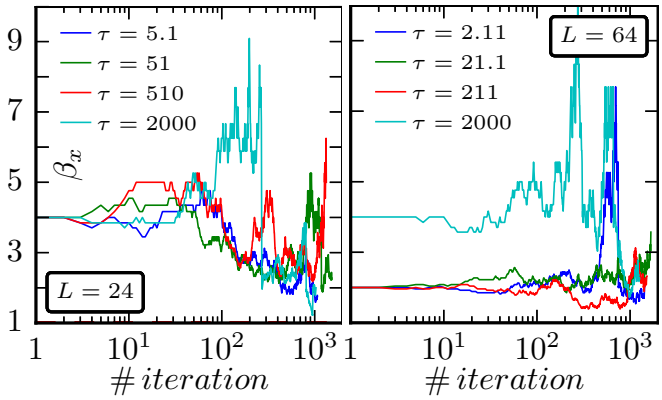

FIG. 5. The inverse temperature β_x *estimated* during the learning stage. See Fig. 4 for comparison and the text for a discussion.

still be reached despite incorrect estimations of the inverse temperature.

V. CONCLUDING REMARKS

In this article, we argued that, despite their limited capabilities, the existing annealers can be harnessed to simulate many-body quantum systems. In our simple model a restricted Boltzmann machine was used to represent the wave function of the transverse field quantum Ising model. Next, we show how this neural network can be trained with the help of a D-wave annealer to find the ground-state energy. The maximum system sizes that we were able to embed were restricted to $L = 64$ (requiring 2048 qubits) for the $2000Q$ chip and $L = 24$ (requiring ∼800 qubits) for its predecessor DW2X. This approach is nonetheless fully scalable.

As a final note, we stress that a neural network trained with an imperfect quantum annealer should, to some extent, reflect on the errors that are generated during the annealing [39]. This means that $\boldsymbol{w}$ found by a faulty quantum sampler will *not* produce correct results with a different sampler. This, on the other hand, allows one to test and possibly calibrate quantum annealers against errors.

ACKNOWLEDGMENTS

We appreciate discussions with A. Mason, E. Dahl, and S. Yarkoni of D-Wave Systems. This work was supported by National Science Centre (NCN), Poland under Projects No. 2016/20/S/ST2/00152 (B.G.) and No. 2016/23/B/ST3/00830 (J.D.) and NCN together with European Union through QuantERA ERA NET Program No. 2017/25/Z/ST2/03028 (M.M.R.). M.M.R. acknowledges receiving a Google Faculty Research Award 2017. This research was supported, in part, by PL-Grid Infrastructure.

[1] I. Cong, M. Cheng, and Z. Wang, Universal Quantum Computation with Gapped Boundaries, Phys. Rev. Lett. **119**, 170504 (2017).
[2] R. Sweke, M. Sanz, I. Sinayskiy, F. Petruccione, and E. Solano, Digital quantum simulation of many-body non-markovian dynamics, Phys. Rev. A **94**, 022317 (2016).
[3] A. Chenu, M. Beau, J. Cao, and A. del Campo, Quantum Simulation of Generic Many-Body Open System Dynamics using Classical Noise, Phys. Rev. Lett. **118**, 140403 (2017).
[4] S. Perrard, E. Fort, and Y. Couder, Wave-Based Turing Machine: Time Reversal and Information Erasing, Phys. Rev. Lett. **117**, 094502 (2016).
[5] G. Vidal, Efficient Classical Simulation of Slightly Entangled Quantum Computations, Phys. Rev. Lett. **91**, 147902 (2003).
[6] B. M. Terhal, Quantum supremacy, here we come, Nat. Phys. **14**, 530 (2018).
[7] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quantum Inf. Process. **13**, 2567 (2014).
[8] D.-L. Deng, X. Li, and S. Das Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).
[9] X. Gao, and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, Nat. Commun. **8**, 2041 (2017).
[10] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).
[11] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Neural-Network Quantum States, String-Bond States, and Chiral Topological States, Phys. Rev. X **8**, 011006 (2018).
[12] S. Sorella, Green Function Monte Carlo with Stochastic Reconfiguration, Phys. Rev. Lett. **80**, 4558 (1998).
[13] J. Carlson, S. Gandolfi, F. Pederiva, S. C. Pieper, R. Schiavilla, K. E. Schmidt, and R. B. Wiringa, Quantum Monte Carlo methods for nuclear physics, Rev. Mod. Phys. **87**, 1067 (2015).
[14] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Quantum Monte Carlo simulations of solids, Rev. Mod. Phys. **73**, 33 (2001).
[15] M. Troyer and U.-J. Wiese, Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations, Phys. Rev. Lett. **94**, 170201 (2005).
[16] M. A. Nielsen, and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, U.K., 2010).
[17] F. Li, V. Y. Chernyak, and N. A. Sinitsyn, Quantum Annealing and Thermalization: Insights from Integrability, Phys. Rev. Lett. **121**, 190601 (2018).
[18] V. S. Denchev, S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, What is the Computational Value of Finite-Range Tunneling?, Phys. Rev. X **6**, 031015 (2016).
[19] S. W. Shin, G. Smith, J. A. Smolin, and U. Vazirani, How quantum" is the D-Wave machine? arXiv:1401.7087.
[20] T. Albash, T. Rønnow, M. Troyer, and D. Lidar, Reexamining classical and quantum models for the d-wave one processor, Eur. Phys. J.: Spec. Top. **224**, 111 (2015).
[21] M. S. Könz, G. Mazzola, A. J. Ochoa, H. G. Katzgraber, and M. Troyer, Uncertain fate of fair sampling in quantum annealing, arXiv:1806.06081.

[22] A. Monràs, G. Sentís, and P. Wittek, Inductive Supervised Quantum Learning, Phys. Rev. Lett. **118**, 190503 (2017).

[23] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[24] A. King *et al.*, Observation of topological phenomena in a programmable lattice of 1,800 qubits, Nature (London) **560**, 456 (2018).

[25] R. Harris *et al.*, Phase transitions in a programmable quantum spin glass simulator, Science **361**, 162 (2018).

[26] S. Woerner and D. J. Egger, Quantum risk analysis, arXiv:1806.06893.

[27] K. Abhinav *et al.*, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[28] A. Cervera-Lierta, Exact Ising model simulation on a quantum computer, arXiv:1807.07112.

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

[30] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, Equivalence of restricted Boltzmann machines and tensor network states, Phys. Rev. B **97**, 085104 (2018).

[31] J. Dziarmaga, Dynamics of a Quantum Phase Transition: Exact Solution of the Quantum Ising Model, Phys. Rev. Lett. **95**, 245701 (2005).

[32] W. H. Zurek, U. Dorner, and P. Zoller, Dynamics of a Quantum Phase Transition, Phys. Rev. Lett. **95**, 105701 (2005).

[33] S. Czischek, M. Gärttner, and T. Gasenzer, Quenches near Ising quantum criticality as a challenge for artificial neural networks, Phys. Rev. B **98**, 024311 (2018).

[34] We reserve symbols σ_i^x, σ_i^z for the annealer Hamiltonian.

[35] E. Lieb, T. Schultz, and D. Mattis, Two soluble models of an antiferromagnetic chain, Ann. Phys. (N.Y.) **16**, 407 (1961).

[36] U. Schollwöck, The density-matrix renormalization group, Rev. Mod. Phys. **77**, 259 (2005).

[37] D. Jaschke, M. L. Wall, and L. D. Carr, Open source matrix product states: Opening ways to simulate entangled many-body quantum systems in one dimension, Comput. Phys. Commun. **225**, 59 (2018).

[38] B. Gardas and S. Deffner, Quantum fluctuation theorem to benchmark quantum annealers, arXiv:1801.06925.

[39] B. Gardas, J. Dziarmaga, W. H. Zurek, and M. Zwolak, Defects in quantum computers, Sci. Rep. **8**, 4539 (2018).

[40] T. Kadowaki and H. Nishimori, Quantum annealing in the transverse Ising model, Phys. Rev. E **58**, 5355 (1998).

[41] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, arXiv:quant-ph/0001106.

[42] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, Adiabatic quantum computation is equivalent to standard quantum computation, SIAM J. Comput. **37**, 166 (2007).

[43] T. Lanting, A. J. Przybysz, A. Y. Smirnov, F. M. Spedalieri, M. H. Amin, A. J. Berkley, R. Harris, F. Altomare, S. Boixo, P. Bunyk, N. Dickson, C. Enderud, J. P. Hilton, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, R. Neufeld, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, S. Uchaikin, A. B. Wilson, G. Rose, Entanglement in a Quantum Annealing Processor, Phys. Rev. X **4**, 021041 (2014).

[44] H. Ushijima-Mwesigwa, C. F. A. Negre, and S. M. Mniszewski, Graph partitioning using quantum annealing on the d-wave system, *Proceedings of the Second International Workshop on Post Moores Era Supercomputing, Denver, 2017, PMES'17* (ACM, New York, 2017), pp. 22–29.

[45] J. Raymond, S. Yarkoni, and E. Andriyash, Global warming: Temperature estimation in annealers, Front. ICT **3**, 23 (2016).

[46] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models, Phys. Rev. X **7**, 041052 (2017).

[47] M. H. Amin, Searching for quantum speedup in quasistatic quantum annealers, Phys. Rev. A **92**, 052323 (2015).

[48] To sample form a *classical* distribution $g(\tau^*) \ll \Delta(\tau^*)$.

[49] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning, Phys. Rev. A **94**, 022308 (2016).

[50] B. Calderhead, A general construction for parallelizing metropolis-hastings algorithms, Proc. Natl. Acad. Sci. U.S.A. **111**, 17408 (2014).

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

[52] B. Gardas and A. Ptok, Counting defects in quantum computers with graphics processing units, J. Comput. Phys. **366**, 320 (2018).

[53] E. W. Tramel, M. Gabrié, A. Manoel, F. Caltagirone, and F. Krzakala, Deterministic and Generalized Framework for Unsupervised Learning with Restricted Boltzmann Machines, Phys. Rev. X **8**, 041006 (2018).

[54] S. H. Adachi, and M. P. Henderson, Application of quantum annealing to training of deep neural networks, arXiv:1510.06356.

[55] K. L. Pudenz, T. Albash, and D. A. Lidar, Error-corrected quantum annealing with hundreds of qubits, Nat. Commun. **5**, 3243 (2014).

[56] M. M. Rams, M. Mohseni, and B. Gardas, Heuristic optimization and sampling with tensor networks for quasi-2D spin glass problems, arXiv:1811.06518v1 (2018).