Quantum artificial intelligence 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]*Institute of Physics, Jagiellonian University, 30-348 Kraków, Poland*

(Dated: November 9, 2019)

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 non-trivial problems concerning physics of many body quantum systems.

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

While 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, 20]. 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 [21].

In this work, we investigate to what extent such algorithms can run on the existing hardware [22]. Our purpose is to demonstrate that already the first generation of quantum computers can in fact *assist* in simulations of simple yet non-trivial many body quantum systems. In our setup, the wave function of the quantum Ising model is represented by a restricted Boltzmann machine [23]. 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.

Quantum neural states. We begin by writing a many body quantum state $|\psi\rangle$ using restricted Boltzmann machine (RBM) as a wave function ansatz [23, 24]:

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

Here $\boldsymbol{v} = (v_1, \dots, v_N)$ 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 $\boldsymbol{h} = (h_1, \dots, h_M)$ are *hidden* neurons, see Fig. 1(a). 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 [23].

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 [25–27],

$$H_{\text{sys}} = -g\sum_i \hat{\sigma}_i^x - \sum_{\langle i,j\rangle} \hat{\sigma}_i^z\hat{\sigma}_j^z. \tag{3}$$

Above, $\langle i,j\rangle$ denotes nearest neighbors and periodic boundary conditions are assumed. We consider both 1D and 2D lattices. In the former case, the ground state energy can be calculated exactly [28]. In the latter, for the system sizes considered here, we used density matrix renormalization group algorithm [29, 30] to obtain its sufficient approximation. This allows us to assess the robustness of our method and at the same time benchmark the annealer [31]. To perform a quantum sampling we use both the newest 2000Q chip and its predecessor DW2X [32].

Henceforward, we also assume that all weights $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{W}$ are *real*. This allows us to use the following ansatz to represent the ground state of the Ising model (3):

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

where, 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 [33]. This is the key insight into all conceptual ideas we outline in this Letter.

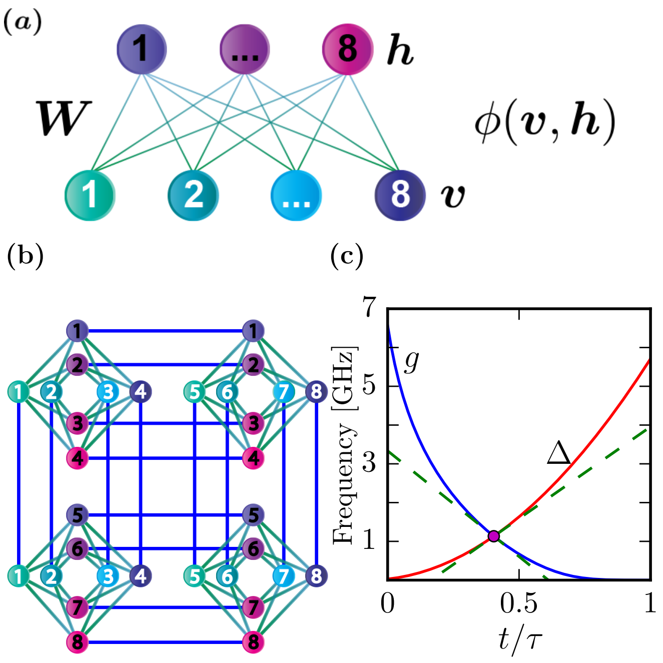

FIG. 1. **(a)** Graphical representation of the neural network (2) with 8 visible and 8 hidden neurons. **(b)** The corresponding network embedded on the actual hardware where allowed connections form so called chimera graph. Strong ferromagnetic couplings (blue lines) "glue" qubits in different unit cells to represent single neurons. **(c)** A typical annealing schedule for the D-Wave chip, τ is the annealing time.

Quantum computing. During quantum annealing a many-body system is to be evolved from the ground state of a problem Hamiltonian H_0 to the ground state of a final Hamiltonian $H(\tau)$ that encodes the solution for the problem of interest [34–36]. The dynamics of a D-Wave annealer is governed by the time-dependent Hamiltonain [17, 37]

$$
\begin{aligned}
H(t)/h &= -g(t)\sum_{i\in\mathcal{V}}\hat{\sigma}_i^x - \Delta(t)H_0, \\
H_0 &= \sum_{i\in\mathcal{V}} B_i\hat{\sigma}_i^z + \sum_{(i,j)\in\mathcal{E}} J_{ij}\hat{\sigma}_i^z\hat{\sigma}_j^z,
\end{aligned}
\tag{6}
$$

where $\hat{\sigma}_i^x$ and $\hat{\sigma}_i^z$ denote the standard Pauli spin operators. H_0 is defined on a chimera graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ [38], see Fig. 1(b). Couplers J_{ij} and biases B_i can be controlled by the users, but only within a predefined range. For example, on the 2000Q platform we have $|J_{ij}| \le 1$ and $|B_i| \le 2$.

The Hamiltonian $H(t)$ varies slowly while $\Delta(t)$ is changed from $\Delta(0) \approx 0$ to $\Delta(\tau) \gg 0$, and $g(t)$ from $g(0) \gg 0$ to $g(\tau) \approx 0$, see Fig. 1(c). As a result, under ideal conditions, the system remains in its ground state and the answer – encoded in eigenvalues, σ_i, of $\hat{\sigma}_i^z$ – can be extracted through a measurement of the final state. However, since no real hardware is completely isolated from its environment, the final solution is distributed according to some temperature-dependent probability distribution $p(\boldsymbol{\sigma})$ [39]. Relaxing the annealer to the equilibrium, one can approximate $p(\boldsymbol{\sigma})$ by the *classical* Boltzmann distribution [40],

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

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, whereas $\beta = \hbar\Delta(\tau)\beta_{\text{eff}}/k_B$. The *unknown*, effective inverse temperature β_{eff} is affected by many factors, including the specific values of the control parameters J_{ij} and B_i [41]. It can only be determined on a case-by-case basis [42]. In this work, 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 a W_{ij}/β between the visible and 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.

Variational Monte Carlo. Training neural networks can be tedious. Moreover, due to its topology, RBMs may be highly susceptible to small changes of the variational parameters. Their adjustments can further propagate throughout the network causing even larger changes of the wave function. To mitigate these problems we use 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

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

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,

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

denote the gradients and local energy, respectively [23].

Usually, the importance sampling is performed using Metropolis-Hastings algorithm [43]. In this work we employ the newest generations of D-Wave samplers to calculate covariance matrix (11) and forces (12) at each iteration [44]. The remaining part of the algorithm is executed on a classical processing unit: CPU or GPU [45].

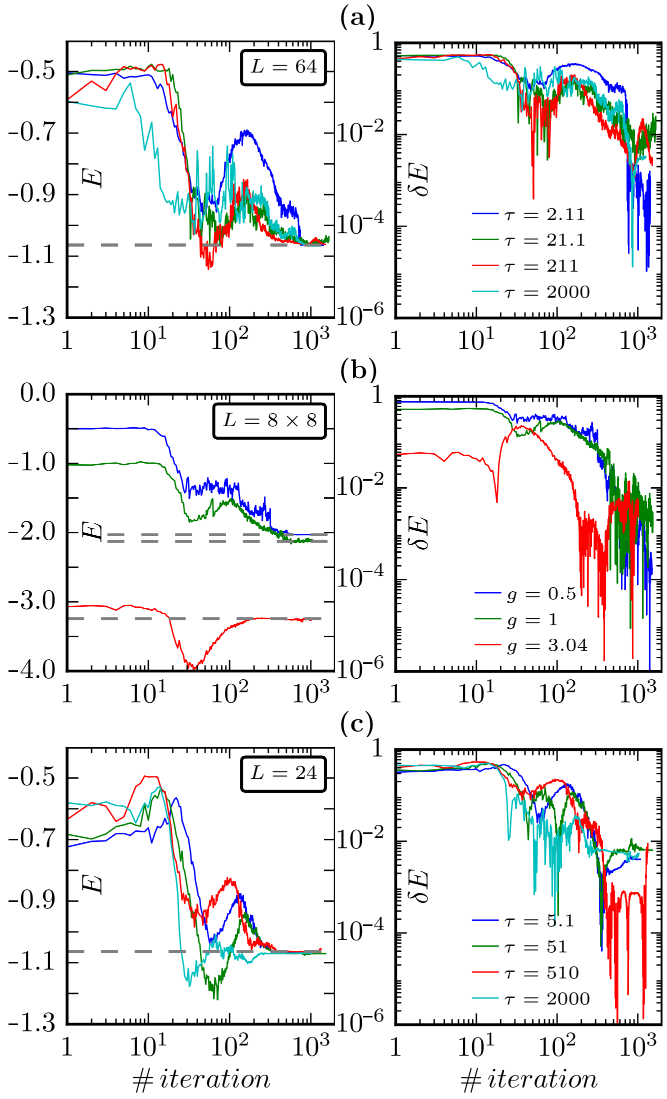

FIG. 2. The ground state energy per spin E for the transverse field quantum Ising models (3) in 1D and 2D. Dashed lines indicate the exact values. Sampling was executed using D-Wave annealers: 2000Q in (a) and (b) and DW2X in (c). δE shows the relative energy error reached. We set $g = 0.5$ for 1D system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

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

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

where we explicitly traced out hidden variables. This is possible due to the lack of intra-layer interactions between hidden neurons, cf. Fig 1(a). Now, all derivatives in

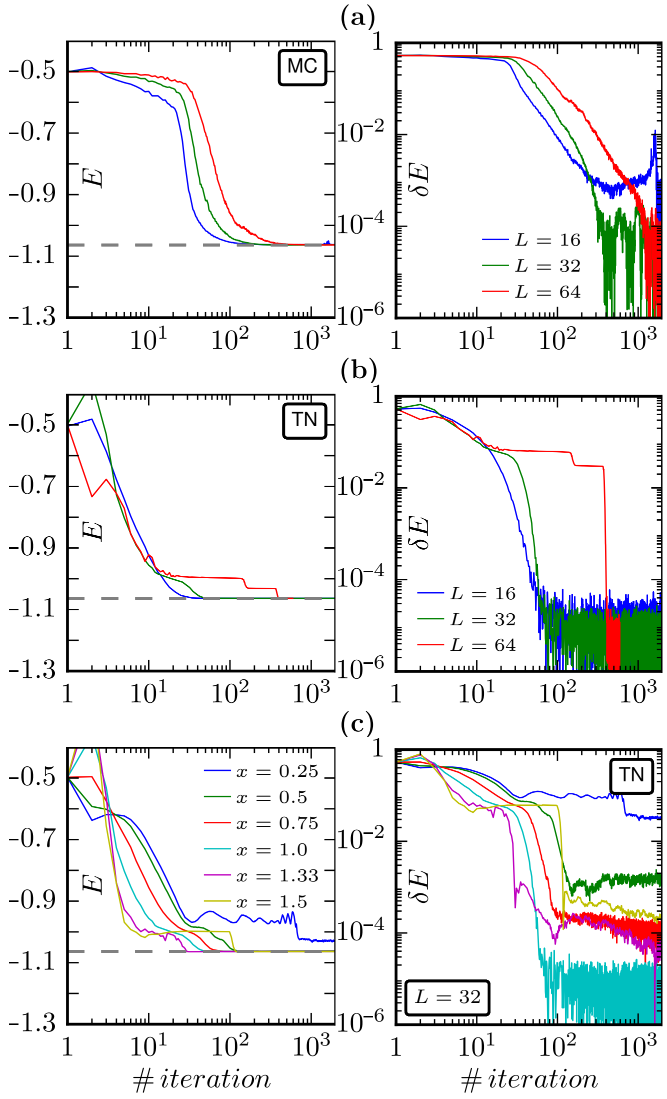

FIG. 3. Similar results as in Fig. 2 obtained for different system sizes in 1D. Sampling was carried out using Monte Carlo with $\gamma = 0.2$ in panels (a) and tensor networks algorithm with $\gamma = 0.05$ in (b). They serve as a reference point for the results in Fig. 2. In panels (c) we show the influence of incorrect inverse temperature estimation, $x = \beta/\beta_x$, on the results.

Eq. (13) can be expressed using visible neurons only [23],

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

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

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

where $\bar{\boldsymbol{v}}_i$ denotes a vector $\boldsymbol{v}$ with i-th 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

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

where $\boldsymbol{\sigma} = [\boldsymbol{v}, \boldsymbol{h}]$ encodes both hidden and visible neurons and N 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 (in practice $N \sim 10^4$) provided that all samples $\boldsymbol{\sigma}_i$ are distributed according to $p(\boldsymbol{\sigma}_i)$. A possible advantage of using a D-Wave computer is that it can sample both visible and hidden neurons simultaneously [46].

RBM on D-Wave. Unfortunately, a RBM *cannot* be directly placed on the D-Wave chip due to limited (sparse) connectivity between qubits [40]. However, this problem can be circumvented using suitable embedding [47]. 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 [48].

Figure 1(b) shows a chimera graph with 4 unit cells. Each unit cell has 8 qubits – with full connectivity between the horizontal and vertical ones and can represent as many neurons. In order to construct, for instance, a RBM with 8 visible and 8 hidden neurons, 4 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 [47]. Provided that the chimera graph C_n has no missing qubits, the maximum number of neurons that it 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.

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., $\alpha = 1$. 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 of the order of 10^{-4}, see Fig. 3(a). The same conclusion is reached using a more sophisticated sampling technique based on tensor networks algorithms, see Fig. 3(b) [49].

We collect the best results which were obtained running the *hybrid* algorithm in Fig. 2. For sampling, we used two generations of D-Wave annealers: 2000Q and DW2X. As one can see in Fig. 2, both of them were capable of finding the correct ground state energy. The solutions reached are, nonetheless, less accurate with δE of the order of 10^{-3}—10^{-2}. This can be expected from the real physical device which is prone to errors [32]. One can also expect those results to improve with each new generation of quantum computers.

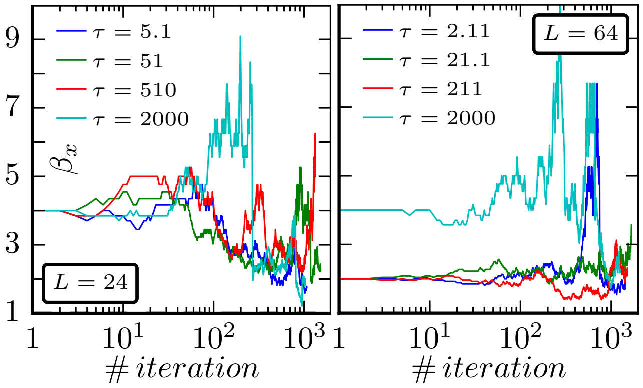

FIG. 4. The inverse temperature β_x *estimated* during the learning stage. See Fig. 2 for comparison and text for discussion.

There are many factors that can contribute to the errors and limited precision [32]. To mitigate some of them the D-Wave solver offers post-processing 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. 4. To that end we randomly increased or decreased the effective temperature β_{eff} 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. Fig. 3(c) shows the robustness of the algorithm against variability of β_{eff}. Surprisingly, the correct solution can still be reached despite incorrect estimations of the inverse temperature.

Concluding remarks. In this Letter 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 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 [32]. 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.

Acknowledgements. We appreciate discussions with Andy Mason, Edward Dahl and Sheir Yarkoni of D-Wave Systems. We gratefully acknowledge Marta Paczyńska who helped us create Figs. 1(a), 1(b). This work was supported by Narodowe Centrum Nauki under projects 2016/20/S/ST2/00152 (BG), 2016/23/B/ST3/00830 (JD) and funded by the National Science Centre, Poland under QuantERA programme 2017/25/Z/ST2/03028 (MMR). MMR acknowledges receiving Google Faculty Research Award 2017. This research was supported in part by PL-Grid Infrastructure

[*] bartek.gardas@gmail.com

[1] Cong, I., Cheng, M. & Wang, Z. Universal quantum computation with gapped boundaries. *Phys. Rev. Lett.* **119**, 170504 (2017).

[2] Sweke, R., Sanz, M., Sinayskiy, I., Petruccione, F. & Solano, E. Digital quantum simulation of many-body non-markovian dynamics. *Phys. Rev. A* **94**, 022317 (2016).

[3] Chenu, A., Beau, M., Cao, J. & del Campo, A. Quantum simulation of generic many-body open system dynamics using classical noise. *Phys. Rev. Lett.* **118**, 140403 (2017).

[4] Perrard, S., Fort, E. & Couder, Y. Wave-based turing machine: Time reversal and information erasing. *Phys. Rev. Lett.* **117**, 094502 (2016).

[5] Vidal, G. Efficient classical simulation of slightly entangled quantum computations. *Phys. Rev. Lett.* **91**, 147902 (2003).

[6] Terhal, B. M. Quantum supremacy, here we come. *Nat. Phys.* (2018) 10.1038/s41567-018-0131-y.

[7] Schuld, M., Sinayskiy, I. & Petruccione, F. The quest for a quantum neural network. *Quantum Inf. Process.* **13**, 2567 (2014).

[8] Deng, D.-L., Li, X. & Das Sarma, S. Quantum entanglement in neural network states. *Phys. Rev. X* **7**, 021021 (2017).

[9] Gao, X. & Duan, L.-M. Efficient representation of quantum many-body states with deep neural networks. *Nat. Comm.* **8**, 2041 (2017).

[10] Torlai, G., Mazzola, G., Carrasquilla, J., Troyer, M., Melko, R. & Carleo, G. Neural-network quantum state tomography. *Nat. Phys.* **5**, 1745 (2018).

[11] Glasser, I., Pancotti, N., August, M., Rodriguez, I. D. & Cirac, J. I. Neural-network quantum states, string-bond states, and chiral topological states. *Phys. Rev. X* **8**, 011006 (2018).

[12] Sorella, S. Green function Monte Carlo with stochastic reconfiguration. *Phys. Rev. Lett.* **80**, 4558 (1998).

[13] Carlson, J., Gandolfi, S., Pederiva, F., Pieper, S. C., Schiavilla, R., Schmidt, K. E. & Wiringa, R. B. Quantum Monte Carlo methods for nuclear physics. *Rev. Mod. Phys.* **87**, 1067 (2015).

[14] Foulkes, W. M. C., Mitas, L., Needs, R. J. & Rajagopal, G. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.* **73**, 33 (2001).

[15] Troyer, M. & Wiese, U.-J. Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations. *Phys. Rev. Lett.* **94**, 170201 (2005).

[16] Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).

[17] Li, F., Chernyak, V. Y. & Sinitsyn, N. A. Quantum annealing and thermalization: Insights from integrability. Preprint at arXiv:1804.00371v1 (2018).

[18] Denchev, V. S., Boixo, S., Isakov, S. V., Ding, N., Babbush, R., Smelyanskiy, V., Martinis, J. & Neven, H. What is the computational value of finite-range tunneling?. *Phys. Rev. X* **6**, 031015 (2016).

[19] Shin, S. W., Smith, G., Smolin, J. A. & Vazirani, U. How quantum" is the D-Wave machine?. Preprint at arXiv:1304.4595 (2014).

[20] Albash, T., Rønnow, T., Troyer, M. & Lidar, D. Reexamining classical and quantum models for the d-wave one processor. *Eur. Phys J-Spec. Top.* **224**, 111 (2015).

[21] Monràs, A., Sentís, G. & Wittek, P. Inductive supervised quantum learning. *Phys. Rev. Lett.* **118**, 190503 (2017).

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

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

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

[25] Dziarmaga, J. Dynamics of a quantum phase transition: Exact solution of the quantum Ising model. *Phys. Rev. Lett.* **95**, 245701 (2005).

[26] Zurek, W. H., Dorner, U. & Zoller, P. Dynamics of a quantum phase transition. *Phys. Rev. Lett.* **95**, 105701 (2005).

[27] Czischek, S., Gärttner, M. & Gasenzer, T. Quenches near Ising quantum criticality as a challenge for artificial neural networks. Preprint at arXiv:1803.08321v1 (2018).

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

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

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

[31] Gardas, B. & Deffner, S. Quantum fluctuation theorem to benchmark quantum annealers. Preprint at arXiv:1801.06925 (2018).

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

[33] Employing the ansatz in Eq. (1) is also possible, but would require to double the number of hidden neurons simulated by the annealer.

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

[35] Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. Preprint at arXiv:0001106v1 (2000).

[36] Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S. & Regev, O. Adiabatic quantum computation is equiva-

lent to standard quantum computation. *SIAM J. Comput.* **37**, 166 (2007).

[37] T. Lanting, *at al.*, Entanglement in a quantum annealing processor. *Phys. Rev. X* **4**, 021041 (2014).

[38] Ushijima-Mwesigwa, H., Negre, C. F. A. & Mniszewski, S. M. Graph partitioning using quantum annealing on the D-Wave system. Preprint at arXiv:1705.03082 (2017).

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

[40] Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Phys. Rev. X* **7**, 041052 (2017).

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

[42] Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. 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).

[43] Calderhead, B. A general construction for parallelizing metropolis-hastings algorithms. *PNAS* **111**, 17408 (2014).

[44] Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum Boltzmann Machine. Preprint at arXiv:1601.02036 (2016).

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

[46] In principle this allows to calculate the gradients directly using the ansatz in Eq. (4), even extending it to deep Boltzmann machine. We leave this for future investigations.

[47] Adachi, S. H. & Henderson, M. P. Application of quantum annealing to training of deep neural networks. Preprint at arXiv:1510.06356 (2015).

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

[49] In preparation. The details will be published elsewhere.