Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning

Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz

Phys. Rev. A **94**, 022308 — Published 9 August 2016

DOI: 10.1103/PhysRevA.94.022308

Estimation of effective temperatures in quantum annealers for sampling applications: A case study towards deep learning

Marcello Benedetti
Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA 94035, USA
SGT Inc., 7701 Greenbelt Rd., Suite 400, Greenbelt, MD 20770, USA and
Department of Computer Science, University College London, WC1E 6BT London, United Kingdom

John Realpe-Gómez
Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA 94035, USA
SGT Inc., 7701 Greenbelt Rd., Suite 400, Greenbelt, MD 20770, USA and
Instituto de Matemáticas Aplicadas, Universidad de Cartagena, Bolívar 130001, Colombia

Rupak Biswas
Exploration Technology Directorate, NASA Ames Research Center, Moffett Field, CA 94035, USA

Alejandro Perdomo-Ortiz*
Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA 94035, USA and
University of California Santa Cruz @ NASA Ames Research Center, Moffett Field, CA 94035, USA

An increase in the efficiency of sampling from Boltzmann distributions would have a significant impact in deep learning and other machine learning applications. Recently, quantum annealers have been proposed as a potential candidate to speed up this task, but several limitations still bar these state-of-the-art technologies from being used effectively. One of the main limitations is that, while the device may indeed sample from a Boltzmann-like distribution, quantum dynamical arguments suggests it will do so with an *instance-dependent* effective temperature, different from its physical temperature. Unless this unknown temperature can be unveiled, it might not be possible to effectively use a quantum annealer for Boltzmann sampling. In this work, we propose a strategy to overcome this challenge with a simple effective-temperature estimation algorithm. We provide a systematic study assessing the impact of the effective temperatures in the learning of a special class of restricted Boltzmann machine embedded on quantum hardware, which can serve as a building block for deep learning architectures. We also provide a comparison to k-step contrastive divergence (CD-k) with k up to 100. Although assuming a suitable fixed effective temperature also allows to outperform one step contrastive divergence (CD-1), only when using an instance-dependent effective temperature we find a performance close to that of CD-100 for the case studied here.

I. INTRODUCTION

The use of quantum computing technologies for sampling and machine learning applications is attracting increasing interest from the research community in recent years [1–16]. Although the main focus of the quantum annealing computational paradigm [17–19] has been in solving discrete optimization problems in a wide variety of application domains [20–27], it has been also introduced as a potential candidate to speed up computations in sampling applications. Indeed, it is an important open research question whether or not quantum annealers can sample from Boltzmann distributions more efficiently than traditional techniques [4, 5, 9].

There are challenges that need to be overcome before uncovering the potential of quantum annealing hardware for sampling problems. One of the main difficulties is that the device does not necessarily sample from the Boltzmann distribution associated with the physical temperature and the user-specified control parameters of the device. Instead, there might be instance-dependent corrections leading, in principle, to instance-dependent effective temperature [4, 28, 29]. Bian *et al.* [4] has used the maximum likelihood method to estimate such an instance-dependent temperature and introduced additional shifts in the control parameters of the quantum device; this was done for several realizations of small eight-qubit instances on an early generation of quantum annealers produced by D-Wave Systems. The authors showed that, with these additional estimated shifts in place, the empirical probability distribution obtained from the D-Wave appears to correlate very well with the corresponding Boltzmann distribution. Further experimental evidence of this effective temperature can be found in Ref. [29] where its proper estimation is needed to determine residual bias in the programmable parameters of the device.

Recent works have explored the use of quantum annealing hardware for the learning of Boltzmann machines and deep neural networks [4, 5, 9, 14, 30]. Learning a Boltzmann machine or a deep neural network is in general intractable due to long equilibration times of sampling

*Electronic address: alejandro.perdomoortiz@nasa.gov

techniques like Markov Chain Monte Carlo (MCMC) [31–33]. One of the strategies that have made possible the recent spectacular success [34] of these techniques is to deal with less general architectures that allow for substantial algorithmic speedups. Restricted Boltzmann Machines (RBMs) [35, 36] are an important example of this kind that moreover serve as a suitable building block for deeper architectures. Still, quantum annealers have the potential to allow for learning more complex architectures.

When applying quantum annealing hardware to the learning of Boltzmann machines, the interest is in finding the optimal control parameters that best represent the empirical distribution of a dataset. However, estimating additional shifts for the control parameters, as done by Bian *et al.* [4], would not be practical since it is in a sense similar to the very kind of problem that a Boltzmann machine attempts to solve. One could then ask what is the meaning of using a quantum annealer for learning the parameters of a distribution, if to do so we need to use standard techniques to learn the corrections to the control parameters.

Here we explore a different approach by taking into account only the possibility of an instance-dependent effective temperature without the need of considering further instance-dependent shifts in the control parameters. We devise a technique to estimate the effective temperature associated to a given instance by generating only two sets of samples from the machine and performing a linear regression. The samples used in our effective-temperature estimation algorithm are the same ones used towards achieving the final goal of the sampling application. This is in contrast with the approach taken in Ref. [4] which needs many evaluations of the gradient of the log-likelihood of a set of samples from the device, making it impractical for large problem instances.

We test our ideas in the learning of a special class of restricted Boltzmann machines. In the next section we shall present a brief overview of Boltzmann machines and discuss how quantum annealing hardware can be used to assist their learning. Afterwards, we discuss related work. In the section that follows we introduce our technique to estimate the effective temperature associated to a given instance. We then show an implementation of these ideas for our Quantum-Assisted Learning (QuALe) of a Chimera-RBM on the Bars And Stripes (BAS) dataset [37–39], implemented in the D-Wave 2X device (DW2X) located at NASA Ames Research Center. Finally, we present the conclusions of the work and some perspectives of the future work we shall be exploring.

II. GENERAL CONSIDERATIONS

A. Boltzmann machines

Consider a binary data set $\mathcal{D} = \{\mathbf{v}^1, \ldots, \mathbf{v}^D\}$ whose empiric distribution is $Q(\mathbf{v})$; here each datapoint can be represented as an array of Ising variables, i.e. $\mathbf{v}^d = (v_1^d, \ldots, v_N^d)$ with $v_i^d \in \{-1, +1\}$, for $i = 1, \ldots, N$. A Boltzmann machine models the data via a probability distribution $P(\mathbf{v}) = \sum_{\mathbf{u}} P_B(\mathbf{u}, \mathbf{v})$, where $P_B(\mathbf{u}, \mathbf{v})$ is a Boltzmann distribution on a possibly extended sample space $\{\mathbf{u}, \mathbf{v}\}$. Here $\mathbf{u} = (u_1, \ldots, u_M)$ are the 'unobservable' or 'hidden' variables, that help capture higher level structure in the data [40], and $\mathbf{v} = (v_1, \ldots, v_N)$ are the 'visible' variables, that correspond to the data themselves. More precisely, denoting these variables collectively by $\mathbf{s} = (\mathbf{u}, \mathbf{v})$ we can write

$$P_B(\mathbf{s}) = \frac{e^{-E(\mathbf{s})}}{Z}, \tag{1}$$

where

$$E(\mathbf{s}) = -\sum_{ij \in \mathcal{E}} W_{ij} s_i s_j - \sum_{i \in \mathcal{V}} b_i s_i, \tag{2}$$

is the corresponding energy function, and Z is the normalization constant or partition function. Notice that in this case we do not need a temperature parameter, since it only amounts at a rescaling of the *model parameters* W_{ij} and b_i that we want to learn. Here $\mathcal{V}$ and $\mathcal{E}$ are the set of vertices and edges, respectively, that make up the interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The task is then to find the model parameters that make the model distribution P as close as possible to the data distribution Q. This can be accomplished by minimizing the Kullback-Leibler (KL) divergence [39]

$$D_{KL}(Q||P) = \sum_{\mathbf{v}} Q(\mathbf{v}) \log \frac{Q(\mathbf{v})}{P(\mathbf{v})}, \tag{3}$$

between Q and P or, equivalently, by maximizing the average log-likelihood

$$\mathcal{L}_{\text{av}} = \frac{1}{D} \sum_{d=1}^{D} \log P(\mathbf{v}^d), \tag{4}$$

with respect to the model parameters W_{ij} and b_i.

Gradient ascent is a standard method to carry out this optimization via the rule

$$W_{ij}^{(t+1)} = W_{ij}^{(t)} + \eta \frac{\partial \mathcal{L}_{\text{av}}}{\partial W_{ij}}, \tag{5}$$

$$b_i^{(t+1)} = b_i^{(t)} + \eta \frac{\partial \mathcal{L}_{\text{av}}}{\partial b_i}, \tag{6}$$

where $\eta > 0$ is the learning rate, and the gradient of the average log-likelihood function is given by [39]

$$\frac{\partial \mathcal{L}_{\text{av}}}{\partial W_{ij}} = \langle s_i s_j \rangle_{\mathcal{D}} - \langle s_i s_j \rangle_{\mathcal{M}}, \tag{7}$$

$$\frac{\partial \mathcal{L}_{\text{av}}}{\partial b_i} = \langle s_i \rangle_{\mathcal{D}} - \langle s_i \rangle_{\mathcal{M}}. \tag{8}$$

Here $\langle \cdot \rangle_{\mathcal{D}}$ denotes the ensemble average with respect to the distribution $P(\mathbf{u}|\mathbf{v})Q(\mathbf{v})$ that involves the data. Similarly, $\langle \cdot \rangle_{\mathcal{M}}$ denotes the ensemble average with respect to

the distribution $P(\mathbf{u}|\mathbf{v})P(\mathbf{v}) = P_B(\mathbf{u},\mathbf{v})$ that involves exclusively the model. Such averages can be estimated by standard sampling techniques, such as MCMC. Another possibility, explored in this work, is to rely on a physical process that naturally generates samples from a Boltzmann distribution.

B. Quantum annealing

Quantum annealing is an algorithm that attempts to exploit quantum effects to find the configurations with the lowest cost of a function describing a problem of interest [17–19]. It relies on finding a mapping of such a function into the energy function of an equivalent physical system. The latter is suitably modified to incorporate quantum fluctuations whose purpose is to maintain the system in its lowest-energy solution space.

In short, the algorithm consists in slowly transforming the ground state of an initial quantum system, that is relatively easy to prepare, into the ground state of a final Hamiltonian that encodes the problem to be solved. The device produced by D-Wave Systems [41, 42] is a realization of this idea for solving quadratic unconstrained optimization problems on binary variables. It implements the Hamiltonian

$$H(\tau) = A(\tau)H_D + B(\tau)H_P, \tag{9}$$

$$H_D = -\sum_{i\in\mathcal{V}_C}\sigma_i^x, \tag{10}$$

$$H_P = \sum_{ij\in\mathcal{E}_C} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i\in\mathcal{V}_C} h_i\sigma_i^z, \tag{11}$$

where $\sigma_i^{x,z}$ are Pauli matrices that operate on spin or qubit i. The *control parameters* of the D-Wave machine are composed of a field h_i for each qubit i and a coupling J_{ij} for each pair of interacting qubits i and j. The topology of the interactions between qubits in the D-Wave is given by a so-called Chimera graph $\mathcal{C} = (\mathcal{V}_C, \mathcal{E}_C)$. This is made up of elementary cells of 4×4 complete bipartite graphs that are coupled as shown in Fig. 1 (a). The transformation from the simple Hamiltonian H_D to the problem Hamiltonian H_P is controlled by time-dependent monotonic functions $A(\tau)$ and $B(\tau)$, such that $A(0) \gg B(0)$ and $A(1) \ll B(1)$. Here $\tau = t/t_a$, where t is the physical time and t_a is the annealing time, i.e. the time that it takes to transform Hamiltonian H_D into Hamiltonian H_P.

Although quantum annealers were designed with the purpose of reaching a ground state of the problem Hamiltonian H_P, there are theoretical arguments [28] and experimental evidence [4, 29] suggesting that under certain conditions the device can sample from an approximately Boltzmann distribution at a given effective temperature, as described in more detail in the next section.

C. Quantum annealing for sampling applications

There are many classical computations that are intrinsically hard and that might benefit from quantum technologies. Common tasks include the factoring of large numbers into its basic primes, as is the case with Shor's algorithm [43] in the gate model of quantum computation. Another one described above consists of finding the global minimum of a hard-to-optimize cost function, where quantum annealing is the most natural paradigm. As described at the end of Sec. II A, another computationally hard problem, key for the successful training of Boltzmann machines and related machine learning tasks, is for example the estimation of averages $\langle\cdot\rangle_{\mathcal{M}}$ over probability distribution functions $P_B(\mathbf{s})$. In the case of models with a slow mixing rate, the standard MCMC approaches would have a hard time obtaining reliable samples from the probability distribution $P_B(\mathbf{s})$ [44, 45]. As long as the quantum annealer can sample more reliably or more efficiently from this Boltzmann distribution, then we can find value in using it to solve a problem where MCMC might become intractable. It has been pointed out in the literature [34, 46] by several experts in the field that to a large extent the key to success of unsupervised learning relies on breakthroughs towards efficient sampling algorithms.

Several key questions arise when considering quantum annealers as potential technologies for providing an algorithmic speed up in sampling applications. Why is a quantum annealer expected to sample from a classical Boltzmann distribution $P_B(\mathbf{s})$, given that it is a quantum device? Shouldn't we expect the quantum annealer to sample from a quantum distribution instead? When and why could we expect the quantum annealer to do better than classical MCMC approaches?

There are several competing dynamical processes happening at different time scales, with the time per annealing cycle being one, while decoherence and relaxation processes having their intrinsic timescale as well. For example, if the annealing time is much larger that the thermal equilibration timescale, the system will remain in its thermal equilibrium until the end of the annealing schedule. On the contrary, if it is too short, diabatic transitions promoting undesirable population flux from the ground state to excited states, would become relevant, leading it to be in a non-equilibrium state.

For quantum annealers that have a strong interaction with the environment leading to relatively fast thermalization and decoherence, theory suggests that the relevant quantum dynamics during an annealing essentially *freezes* somewhere between the critical point associated with the minimum gap and the end of the annealing schedule [28, 47, 48]. In a quasistatic regime [28, 48], the system happens to be close to a Boltzmann distribution but at a certain effective temperature that is in general different from the physical temperature of the device. Such a *freezing point* τ_{freeze} tends to coincide with the coefficients in Eq. (9) satisfying $A(\tau_{\text{freeze}}) \ll B(\tau_{\text{freeze}})$,

which suggests that the system being quantum annealed might end up in a Boltzmann distribution of the classical cost function encoded in H_P.

The intuition behind this phenomenon is that the dominant coupling of the qubits to the environment/bath degrees of freedom is via the σ^z operator (for details, see supplementary material of Refs. [41, 49]). Since at the freezing point we have $A(\tau_{\text{freeze}}) \ll B(\tau_{\text{freeze}})$, and the interaction with the bath lacks a strong σ^x component capable of causing relaxation between the states of the computational basis (i.e. eigenstates of σ^z), then the system cannot relax its population anymore; in other words, its population dynamics freezes. Since around τ_{freeze} the full Hamiltonian driving the dynamics is $H(\tau_{\text{freeze}}) \approx B(\tau_{\text{freeze}})H_P$, if a Boltzmann distribution is indeed reached, it would correspond to an effective temperature T_{eff} different from the physical temperature of the device. Here we will follow the convention that the units of temperature are given in a dimensionless energy scale where 1.0 is the maximum programmable value for the J couplers. According to Eq. (9) the total Hamiltonian at the end of the annealing ($\tau = 1$) is given by $H(1) = B(1)H_P$, so $J = 1.0$ would correspond to an energy value given by $B(1)$. For the DW2X at NASA, $J = 1.0$ corresponds to $B(1) = 7.9$ GHz. For example, the physical fridge temperature of this quantum annealer, $T_{\text{DW2X}} = 12.5$ mK, corresponds to $T_{\text{DW2X}} = 0.033$ in the dimensionless units we follow in this paper. The effective temperature would be $T_{\text{eff}} \equiv T_{\text{DW2X}} B(1)/B(\tau_{\text{freeze}})$; since $B(\tau_{\text{freeze}}) < B(1)$, then $T_{\text{eff}} > T_{\text{DW2X}}$. Such an effective temperature is expected to depend on the specific instance being studied and on the details of its energy landscape. Some recent unpublished work in our research team indicates that the effective temperature could also be influenced by the noise in the programmable parameters and by its interplay with the specific instance studied, making an *a priori* estimation a daunting task. The approach we take in this work is to estimate this effective temperature from the same samples that would be eventually used for the subsequent training process.

We could wonder why a quantum annealer is expected to help in this computational task? It has been shown that quantum tunneling [49] might be a powerful computational resource for keeping the system close to the ground state and to the proper thermal distribution. It is these quantum resources, available during the quantum dynamics before the freezing point, that might assist and speed up the thermalization process, making sampling more efficient than other classical approaches, such as MCMC. It is important to mention that such a quantum advantage is not expected for all energy landscapes; there will be instances that will be hard for both classical annealers and for quantum annealers. The answer to this question will be highly dependent on the quantum resources available and on the complexity of the energy landscape itself. This is an important and interesting question in its own that we will addressing in future work. In this work we focus in unveiling the effective temperature that properly defines the distribution we are sampling from, and test our method in the context of a machine learning problem related to the training of Boltzmann machines.

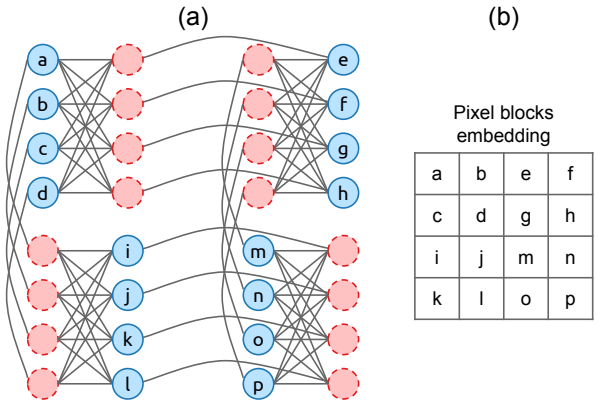

FIG. 1: *Chimera-RBM and data representation:* (a) D-Wave hardware embedding of a Chimera-RBM with 16 visible and 16 hidden variables. (b) Mapping of the pixels in the pictures to the visible units in the Chimera-RBM that has been used in this work (cf. [9]).

D. Chimera restricted Boltzmann machines

Learning a Boltzmann machine is in general intractable due to long equilibration time of sampling techniques like MCMC. One way to escape this issue is to use less general architectures. One of the most investigated architectures is the *Restricted Boltzmann Machine* (RBM). The interaction graph $\mathcal{G}$ of an RBM is a complete bipartite graph in which visible and hidden units interact with each other, but not among themselves. This implies that the conditional distributions $P(\mathbf{v}|\mathbf{u})$ and $P(\mathbf{u}|\mathbf{v})$ factorize in terms of single variable marginals, which substantially simplifies the problem. One the one hand, data averages $\langle\cdot\rangle_{\mathcal{D}}$ can be computed exactly in one shot. On the other hand, model averages $\langle\cdot\rangle_{\mathcal{M}}$ can be approximated by k-step contrastive divergence (CD-k): first, we start with a datapoint $\mathbf{v}^{(0)}$; then we sample $\mathbf{u}^{(0)}$ from $p(\mathbf{u}|\mathbf{v}^{(0)})$, and subsequently sample $\mathbf{v}^{(1)}$ from $p(\mathbf{v}|\mathbf{u}^{(0)})$ and so on for k steps. At the end of this process we obtain samples $\mathbf{v}^{(k)}$ and $\mathbf{u}^{(k)}$ from which it is possible to estimate model averages [39]. CD-k is not guarantee to give correct results [33, 44] nor does it actually follow the log-likelihood gradient nor the gradient of any function indeed. Better sampling methods can have therefore a positive impact in the kind of models learned.

It is in principle possible to embed an RBM in quantum annealing hardware [14]. However, due to limited connectivity of the device, the resulting physical representation would involve a number of qubits and couplings between them much larger than the number of logical variables and weights in the original RBM being repre-

sented. It would be preferable to use an alternative model that can be naturally represented in the device. For this reason we will focus on the kind of models that are obtained after removing from a given RBM all the links that are not present in the D-Wave machine [9]. We will call this type of model a *Chimera Restricted Boltzmann Machine* (Chimera-RBM). Fig. 1 (a) shows an example of a Chimera-RBM and Fig. 1 (b) shows a possible embedding of the pixels of an image into its visible units (cf. Ref. [9]).

III. RELATED WORK

Dumoulin *et al.* [9] have studied the impact of different limitations of quantum annealing hardware for learning restricted Boltzmann machines. The authors have focused on three kinds of limitations: noisy parameters, limited parameter range, and restricted architecture. The learning method used was persistent contrastive divergence where the model ensemble averages were estimated with samples from simulated quantum hardware while the data ensemble averages were estimated by exact mean field.

For assessing the impact of limited connectivity, Dumoulin *et al.* investigated a Chimera-RBM. They found that limited connectivity is the most relevant limitation in this context. In a sense this is understandable as RBMs are based on complete bipartite graphs while Chimera-RBMs are sparse. Roughly speaking, this means that if the number of variables is of order N, the number of parameters present in a Chimera-RBM is a vanishing fraction (of order $1/N$) of the number of parameters in the corresponding RBM. Furthermore, connections in a Chimera-RBM are rather localized. This feature may make more difficult to capture higher level correlations.

The authors also found that noise in the parameters of an RBM is the next relevant limitation and that noise in the weights W_{ij} is more relevant than noise in the biases b_i. This could happen because the number of biases is a vanishing fraction of the number of weights in an RBM. This argument is not valid anymore in a Chimera-RBM, though. Now, the authors also mention that noise in the weights changes only when these change, while noise in the biases changes in every sample generated. If this is indeed the case, this could be another reason for the higher relevance of noise in the weights than noise in the biases.

Finally, an upper bound in the magnitude of the model parameters, similar to the one present in the D-Wave device, does not seem to have much impact. In this respect, we should notice that current D-Wave devices are designed with the sole aim of consistently reaching the ground state. In contrast, typical applications of Boltzmann machines deals with heterogeneous real data which contains a relatively high level of uncertainty, and are expected to exploit a wider range of configurations. This suggests that in sampling applications control parameters are typically smaller than those explored in combinatorial optimization applications. If this is indeed the case, potential lower bounds in the magnitude of the control parameters can turn out to be more relevant for sampling applications. In this respect, it is important to notice that noise in the control parameters can lead to an effective lower bound.

While Dumoulin *et al.* modeled the instance-dependent corrections as independent Gaussian noise around the user defined parameter values, Denil and Freitas [5] devised a way to by-pass this problem altogether. For doing this, the authors have optimized the one-step reconstruction error as a black-box function and approximate its gradient empirically. They do this by a technique called simultaneous perturbation stochastic approximation. However, with this approach, it is not possible to decouple the model from the machine. Furthermore, it is not clear what is the efficiency of this technique nor how to extend it to deal with the more robust log-likelihood function instead of the reconstruction error. In their approach only the hidden layer is embedded in the D-Wave, and qubit interactions are exploited to build a semi-restricted Boltzmann machine. Although they report encouraging results, the authors acknowledge that these are still not conclusive.

More recently, Adachi and Henderson [14] have devised a way to embed an RBM on a D-Wave chip with Chimera topology. They do this by representing each logical variable by a string of qubits with strong ferromagnetic interactions. Furthermore, they implement a simple strategy to average out the effects of the noise in the D-Wave control parameters. They use the quantum annealer to estimate model averages as in Ref. [9] for pre-training a two-layer neural network. However, the authors do not evaluate the performance of the quantum device at this stage; they rather post-train the model with (classical) discriminative techniques for learning the labels of a coarse-grained version of the MNIST dataset and compute the classification error. They report that this approach outperforms the standard approach where CD-1, instead of quantum annealing, is used for pre-training the generative model.

IV. QUANTUM-ASSISTED LEARNING OF BOLTZMANN MACHINES

In this work we assume that quantum annealers, like those produced by D-Wave Systems, sample from a Boltzmann distribution defined by an energy function as in Eq. (2), with $W_{ij} = J_{ij}/T_{\text{eff}}$ and $b_i = h_i/T_{\text{eff}}$, where T_{eff} can be instance-dependent. While the control parameters for the D-Wave are couplings and fields, i.e. J_{ij} and h_i, the learning takes place on the ratio of the control parameters to the temperature, i.e. W_{ij} and b_i. Inferring temperature is therefore a fundamental step to be able to use samples from a device like D-Wave for

learning, since it provides a translation from $\{W_{ij}\}$ to the $\{J_{ij}\}$ and from the $\{b_i\}$ to the $\{h_i\}$. We propose a quantum-assisted learning (QuALe) technique that includes an efficient estimation of the effective temperature. It is initialized as follows:

- Pick small initial control parameters $J_{ij}^{(0)}$ and $h_i^{(0)}$, and sample from the device.
- Using the samples obtained in the previous item, estimate the initial temperature $T_{\text{eff}}^{(0)}$ to compute the initial model parameters $W_{ij}^{(0)} = J_{ij}^{(0)}/T_{\text{eff}}^{(0)}$ and $b_i^{(0)} = h_i^{(0)}/T_{\text{eff}}^{(0)}$.

Then it iterates as follows:

- Using the samples and model parameters obtained at step t, estimate the corresponding temperature $T_{\text{eff}}^{(t)}$ and update the model parameters according to Eqs. (5) and (6) to obtain $W_{ij}^{(t+1)}$ and $b_i^{(t+1)}$.
- Obtain new control parameters by doing $J_{ij}^{(t+1)} \approx T_{\text{eff}}^{(t)} W_{ij}^{(t+1)}$ and $h_i^{(t+1)} \approx T_{\text{eff}}^{(t)} b_i^{(t+1)}$ and sample from the device.

A few comments are in order. First, for each sample step we need to generate samples for estimating model and data ensemble averages. For the former we just need to run the device with the specified control parameters. For the latter we need to generate samples with the visible units clamped to the data points, which can be done by applying suitable fields to the corresponding qubits. However, in the case of restricted Boltzmann machines we can avoid this last step as it is possible to compute exactly the data ensemble averages. Second, notice that to compute the new control parameters at step $t+1$ it would have been ideal to estimate the temperature $T_{\text{eff}}^{(t+1)}$ at the same step. However, to estimate such a temperature we would need to know which are the parameters at time $t+1$. To escape this vicious cycle we have done $T_{\text{eff}}^{(t+1)} \approx T_{\text{eff}}^{(t)}$. Finally, notice that if we think of the learning process in terms of the control parameters J_{ij} and h_i, we may get the impression that the learning rate is temperature-dependent. We would like to emphasize that the learning operates on the model parameters W_{ij} and b_i, which are those that actually shape the Boltzmann distribution, through the update rules given by Eqs. (5) and (6). So, the actual learning rate is given by η in the update equations above; if we fix η to a constant, it would remain so. We need T_{eff} only to estimate the required control parameters. Still, the approximation $T_{\text{eff}}^{(t+1)} \approx T_{\text{eff}}^{(t)}$ and the error in their estimation can introduce noise that may deviate the learning process from the actual update rules given by Eqs. (5) and (6). It would be interesting to investigate what is the impact of this noise in contrast to that due to the estimation of the log-likelihood gradient with a finite number of samples. In the next section we discuss a method for estimating this instance-dependent temperature.

V. TEMPERATURE ESTIMATION

A. Extracting temperature from two sample sets

At a generic inverse temperature β, the probability of observing a configuration of energy E is given by $P_\beta(E) = g(E)e^{-\beta E}/Z(\beta)$. Here $g(E)$ is the degeneracy of the energy level E and the normalization factor, $Z(\beta)$, is the partition function. We want to devise an efficient method for estimating the effective temperature associated with a given instance. To do this, consider the log-ratio of probabilities associated with two different energy levels, E_1 and E_2, given by

$$\ell(\beta) \equiv \log \frac{P_\beta(E_1)}{P_\beta(E_2)} = \log \frac{g(E_1)}{g(E_2)} - \beta \Delta E, \qquad (12)$$

where $\Delta E = E_1 - E_2$. We can estimate this log-ratio by estimating the frequencies of the two energy levels involved; in practice, we may have to do a suitable binning to have more robust statistics. Although we cannot control the physical temperature, we could in principle do this for different values of the parameter β by rescaling the control parameters of the device. Indeed, rescaling the control parameters by a factor x. This is equivalent to setting a parameter $\beta = x\beta_{\text{eff}}$, where $\beta_{\text{eff}} = 1/T_{\text{eff}}$ is the inverse of the effective temperature T_{eff} associated to the instance of interest. Notice that this is only true under the assumption that T_{eff}, despite being generally dependent on arbitrary variations of the control parameters, does not change appreciably under these small rescalings. By plotting the log-ratio $\ell(x\beta_{\text{eff}})$ against the scaling parameter x, we should obtain a straight line whose slope and intercept are given by $-\beta_{\text{eff}}\Delta E$ and $\log[g(E_1)/g(E_2)]$, respectively. Since we know the energy levels we can in principle infer β_{eff}. However, the performance of this method was rather poor in all experiments we carried out (not shown). A reason could be that to perform the linear regression and extract the corresponding effective temperature, several values of x need to be explored in a relatively wide range. Next we present a proposal that mitigates this limitation, which also happens to be much more efficient.

The previous approach relied on several values of the scaling parameter x but only two energy levels. We were not exploiting all the information available in the other energy levels sampled from the quantum annealer. We can exploit such an information to obtain a more robust estimate of the temperature by sampling only for the original control parameters and a single rescaling of them. The idea is to take the difference $\Delta\ell \equiv \ell(\beta) - \ell(\beta')$, with $\beta = \beta_{\text{eff}}$ and $\beta' = x\beta_{\text{eff}}$, to eliminate the unknown degeneracies altogether, yielding

$$\Delta\ell = \log \frac{P_\beta(E_1)P_{\beta'}(E_2)}{P_\beta(E_2)P_{\beta'}(E_1)} = \Delta\beta\Delta E, \qquad (13)$$

where $\Delta\beta = \beta' - \beta = (x-1)\beta_{\text{eff}}$. In this way, by generating a second set of samples at a suitable value of x and

then taking the differences of all pairs of populated levels, we can plot $\Delta\ell$ against ΔE. According to Eq. (13) this is expected to be a straight line with slope given by $(x-1)\beta_{\rm eff}$. In practice, one has to choose a binning strategy and use the same bin intervals in both histograms so that the *overlap* makes sense. For example, by setting the number of bins to $K = \left\lceil \sqrt{2R} \right\rceil$, where R is the number of samples per set, one obtains $\mathcal{O}(K^2) = \mathcal{O}(R)$ data points for linear regression. Notice that the raw energies computed before binning refer to the *original* values of the control parameters in *both* cases, *not* the rescaled ones. This is because we have already counted the effect of the rescaling in a different inverse effective temperature $\beta' = x\beta_{\rm eff}$. Finally, the energies levels obtained after binning correspond to the midpoint of each bin.

The choice of x matters: if it is too small no informative changes would be detected, other than noise due to finite sampling and uncontrolled physical processes in the device. If it is too large, several levels would become unpopulated and we would not be able to compare them at both the original and rescaled control parameters; moreover, the assumption of the invariance of $T_{\rm eff}$ under small perturbations around the original control parameters would be less likely to be valid. Next we discuss how to choose the value of x.

B. A rule of thumb for the scaling factor

We can rely on concepts of information theory to guide the choice of the scaling factor x. The idea here is to choose the value of x as close as possible to one that still allows us to distinguish between the two sets of samples of a given size. Via Sanov's theorem, the KL divergence provides a natural way to characterize the notion of distinguishability in this case [50–52]. Here we will briefly discuss the main ideas in a rather informal way; the interested reader can refer to Ref. [52] for details. We want to know whether we can distinguish between two Boltzmann distributions at different inverse temperatures β and β' from a set of R samples. For doing this, it is useful to consider that we compute the maximum likelihood estimate of the inverse temperature $\beta_{\rm ML}$ from the sample set corresponding to inverse temperature β. We can consider that we repeat this procedure many times so we can compute the probability distribution of $\beta_{\rm ML}$. The two Boltzmann distributions are said to be *distinguishable* from a set of R samples if the probability of $\beta_{\rm ML}$ being close to β' is smaller than a given tolerance P_0, i.e. if

$$\text{Prob}\left[|\beta_{\rm ML} - \beta'| < \delta\right] < P_0, \tag{14}$$

where δ is a suitably small constant. From Sanov's theorem it follows that when R is large enough

$$\text{Prob}\left[|\beta_{\rm ML} - \beta'| < \delta\right] \approx C e^{-R D_{\rm KL}(P_{\beta'}||P_\beta)}, \tag{15}$$

where the factor C gathers sub-dominant terms in R. So, if $D_{\rm KL}(P_{\beta'}||P_\beta) > \log(C/P_0)/R$ the two Boltzmann distributions are distinguishable in the sense defined above.

Assuming that β and β' are close enough, the KL divergence can be expanded up to second order to yield

$$D_{\rm KL}(P_{\beta'}||P_\beta) \approx \frac{1}{2}\chi(\beta)\Delta\beta^2, \tag{16}$$

where

$$\chi(\beta) = \frac{\partial^2 \log Z(\beta)}{\partial \beta^2} = \left\langle E^2 \right\rangle - \left\langle E \right\rangle^2 \equiv \sigma_E^2, \tag{17}$$

is known in information theory as the Fisher information, or generalized succeptibility; in this case, it is essentially the specific heat. When R is large enough, the right hand side in Eq. (15) becomes appreciable only for β and β' very close. So, for large R we can replace the KL divergence by the Fisher information in Eq. (15).

Following these ideas, we propose to choose the scaling factor x such that $\frac{1}{2}\chi(\beta)(1-x)^2\beta_{\rm eff}^2 = d_{\rm KL}/R$, where $d_{\rm KL}$ is a given constant (cf. Ref. [53]). Eqs. (16) and (17) yield

$$x = 1 \pm \sqrt{\frac{2\, d_{\rm KL}}{R\, \beta_{\rm eff}^2\, \sigma_E^2}}. \tag{18}$$

Some remarks are in order: (i) Eq. (18) gives a rule of thumb to choose a suitable value of x for estimating $\beta_{\rm eff}$; however, the latter also appears in this expression. We can initiate $\beta_{\rm eff}$ by either making a reasonable guess or using the pseudo-likelihood estimate (see Appendix A). (ii) The sign in Eq. (18) could be chosen positive during the first iterations to avoid the rescaled control parameters to be below the noise level of the device, and negative afterwards to avoid the rescaled control parameters to be above the allowed range. (iii) Eq. (18) has been derived assuming that values of the KL divergence about $d_{\rm KL}/R$ can be well approximated with the Fisher information. These assumption may fail in practice when R is relatively small or when x is far from the reference value at $x = 1.0$. (iv) In principle, as long as the samples generated by the quantum annealer follow a Boltzmann distribution and the effective temperature remains constant under re-scalings of the control parameters, our temperature estimation technique is exact if there are enough samples. Still, the number of samples needed could grow exponentially with problem size due to the bias and variance associated to our estimator, whose study we leave for future work. (v) Finally, the linear regression to compute our estimator may be affected by noise due to energy bands with very low frequency; in principle, this could be mitigated by relying on a weighted linear regression giving more weight to points associated with higher frequencies.

VI. A FEW GADGETS TO IMPROVE PERFORMANCE

In this section we discuss three techniques that help improve the performance of our quantum-assisted learning algorithm. First of all, it is known that the performance of quantum annealers can be significantly impaired by the presence of both persistent and random biases between the actual values of the control parameters and the user-specified values. Perdomo-Ortiz *et al.* [29] have developed a technique for determining and correcting the persistent biases and have shown evidence that this recalibration procedure can enhance the performance of the device for solving combinatorial optimization problems. In the next section we will show evidence that correcting for persistent biases can also enhance the performance of quantum annealers for sampling applications.

Second, noise in the control parameters can hinder the initial stage of learning, when these are typically small. In order to avoid this situation we can run CD-1 for a few iterations until we find meaningful initial values for the control parameters that are above the noise level of the device and then restart with QuALe. This is exclusively due to the current state of quantum annealing technologies and it is expected to be further mitigated in new generations of these devices. We emphasize that the number of iterations with CD-1 has to be small to keep the weights within the dynamical range of the device.

Finally, for estimating the effective temperature associated to a given instance we need to generate two sets of samples: one corresponding to the actual values of the control parameters that we are interested in, and another corresponding to these values rescaled by a factor x. According to the discussion in the previous section, the scaling factor is chosen in such a way that the two probability distributions are as close as possible, yet distinguishable. So, we expect that the samples obtained at $\beta' = x\beta_{\text{eff}}$ can also be used for the estimation of the log-likelihood gradient, given by Eqs. (7) and (8), at $\beta = \beta_{\text{eff}}$ via the technique of importance sampling [54]. In short, we can use a set of samples $\{\mathbf{s}^1, \ldots, \mathbf{s}^R\}$ extracted from a Boltzmann distribution at inverse temperature β' to estimate ensamble averages of an arbitrary observable A with a Boltzmann distribution at inverse temperature β as

$$\langle A \rangle_\beta \approx \frac{\sum_{r=1}^{R} \rho(\mathbf{s}^r) A(\mathbf{s}^r)}{\sum_{r=1}^{R} \rho(\mathbf{s}^r)}, \tag{19}$$

where $\rho(\mathbf{s}) = e^{-(\beta-\beta')E(\mathbf{s})}$ is the ratio between the unnormalized probabilities. The approximation is expected to be good as long as the two distributions are close enough [54]. In the next section we will show evidence that including the set of samples corresponding to the rescaled control parameters indeed improves the performance of QuALe.

From now on, when refering to the QuALe algorithm we imply that these three gadgets are also included, unless otherwise specified.

VII. LEARNING A BOLTZMANN MACHINE ASSISTED BY THE D-WAVE 2X

Now that we have at our disposal a robust temperature estimation technique, we can use it for learning Boltzmann machines. We decided to focus on the learning of a Chimera-RBM for two reasons. On the one hand, although an RBM can be embedded into quantum hardware [14], it requires to represent single variables with chains of qubits coupled via ferromagnetic interactions of a given strength. Instead of forcing couplings to take a specific value to meet a preconceived design, it might be better to allow the learning algorithm itself to find the parameter values that work best for a particular application. On the other hand, the focus of our work is in better understanding the challenges that need to be overcome for using quantum annealers for sampling applications, and taking the necessary steps towards an effective implementation of deep learning applications on quantum annealers.

To the best of our knowledge, this is the first systematic study providing both, an assessment of the use of the D-Wave in learning Boltzmann machines and studying the impact of the effective temperature in the learning performance. We consider that it is important to assess the performance of the different methods by computing the exact log-likelihood during the learning process. Otherwise, we could not be sure whether a difference in performance is due to the new learning method or due to errors in the approximation of the log-likelihood. For this reason we tested the method on a small synthetic dataset called Bars and Stripes (BAS) and computed exhaustively the corresponding log-likelihood for evaluation. The BAS dataset consists of 4×4 pictures generated by setting the four pixels of each row (or column) to either black (-1) or white (+1), at random [37–39]. Another reason to focus on this small synthetic dataset is that while generating, e.g., 2000 samples in the DW2X for a given instance can take about 40 ms, the waiting time for accessing the machine to generate a new set of samples for a different instance can vary widely depending on the amount of jobs that are scheduled. So, while running QuALe with 2000 samples per iteration on the whole chip (1097 qubits) for 10^4 iterations could take in principle about 7 minutes had we exclusive access to the device, the waiting times of the different jobs can increase this time by several orders of magnitude.

We modeled the BAS dataset with a Chimera-RBM of 16 visible and 16 hidden units with the topology shown in Fig. 1 (a). The mapping of pixels to visible units is shown in Fig. 1 (b) (cf. [9]). We run all algorithms with learning rate $\eta = 0.03$, which is the best value we found among five values in the range $[0.01, 0.1]$. To begin with, Fig. 2 shows an instance of temperature estimation using $R = 1000$ samples from the DW2X and $d_{\text{KL}} = 500$, for generic control parameters found during the learning process (cf. Fig. 4). This value of d_{KL} is the one that worked best out of a few trial values. Fig. 2a shows the histograms

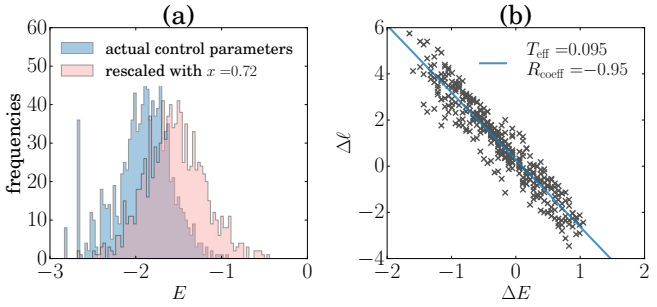

FIG. 2: *Temperature estimation:* (a) Energy histograms obtained from $R = 1000$ samples generated by the DW2X for two different sets of control parameters and $K = \left\lceil \sqrt{2R} \right\rceil$ bins. The (blue) histogram that is shifted to the left corresponds to a typical set of control parameters found during the learning of a Chimera-RBM on the BAS dataset (cf. Fig. 4) using $d_{KL}/R = 1/2$. The (pink) histogram shifted to the right corresponds to these control parameters scaled by a factor $x = 0.72$ obtained using Eq. (18). (b) Log-likelihood ratio differences $\Delta\ell$ in Eq. (13) plotted against the corresponding energy differences ΔE. These are computed using all energy bins in the overlap of the two histograms in (a). The straight line is obtained by a linear regression using least squares and predicts, according to Eq. (13), an effective temperature of $T_{\rm eff} \approx 0.095$ and a regression coefficient of $R_{\rm coeff} \approx -0.95$. The units of temperature are given in a dimensionless energy scale where 1.0 is the maximum programmable value for the J couplers. For the DW2X at NASA, $J = 1.0$ corresponds to 7.9 GHz. For example, the physical fridge temperature of this quantum annealer, $T_{\rm DW2X} = 12.5$ mK, corresponds to $T_{\rm DW2X} = 0.033$ in the dimensionless units we follow in this paper. The instance plotted in this figure has a $T_{\rm eff} \approx 3T_{\rm DW2X}$

corresponding to $K = \left\lceil \sqrt{2R} \right\rceil$ bins of samples obtained at the actual control parameters (blue, shifted to the left) and the rescaled ones (pink, shifted to the right). Fig. 2b shows a plot of $\Delta\ell$ against ΔE for all energy values that appear in the overlap of the two histograms. We can observe a rather clear linear trend as predicted by Eq. (13), which is confirmed by a relatively high regression coefficient, $R_{\rm coeff} \approx -0.95$. From the slope m of the regression line we can obtain the effective temperature by solving $m = \Delta\beta = (x-1)\beta_{\rm eff}$.

Fig. 3a shows the impact of bias correction on the performance of the QuALe algorithm. The performance is measured in terms of the average log-likelihood $\mathcal{L}_{\rm av}$, which has been evaluated exhaustively every fifty iterations. These results are obtained by implementing the Chimera-RBM on five different locations of the DW2X chip and running the QuALe algorithm three times on each location, for a total of fifteen runs. The points correspond to the average of $\mathcal{L}_{\rm av}$ over those fifteen runs and the error bars to one standard deviation. We can see that QuALe with persistent bias correction (blue crosses) outperforms QuALe without it (pink triangles). Fig 3b, on the other hand, shows the QuALe algorithm with (blue crosses) and without (pink triangles) taking into account the samples obtained at $x \neq 1$ for the estimation of the log-likelihood gradient, via importance sampling. The points correspond to the average of $\mathcal{L}_{\rm av}$ over five runs of QuALe on a single location of the DW2X chip. Finally, Fig. 3c shows the positive impact of carrying out a few iterations of CD-1 to generate suitable initial conditions for QuALe.

Fig. 4 shows the evolution of $\mathcal{L}_{\rm av}$ during the learning of a Chimera-RBM on the BAS dataset under different learning algorithms, all of them with learning rate $\eta = 0.03$. We can observe that the quantum assisted learning algorithm with effective-temperature estimation at each iteration (QuALe@$T_{\rm eff}$, blue diagonal crosses) outperforms CD-1 (blue solid squares) after about 300 iterations and CD-10 (green solid circles) after about 1500 iterations. However, within the 5000 iterations shown in the figure, QuALe@$T_{\rm eff}$ has not yet been able to outperform CD-100, although there is a clear trend in that direction. As we did not observe any significant improvement when using larger values of k, we expect that CD-100 is close to an exact computation (cf. Theorem 5.1 in [55]). Interestingly, all CD-k reach their best average performance after a relatively small number of iterations while QuALe@$T_{\rm eff}$, in contrast, increases slowly and steadily. One may be inclined to think this is because CD-k estimates the model averages from samples generated by a k-step Markov chain initialized at each data point. In this way CD-k is using information contained in the data from the very beginning for the estimation of the model ensemble averages, while QuALe@$T_{\rm eff}$ ignores them altogether. However, if this were indeed the case one should expect such a trend to diminish for increasing values of k, something that is not observed in the figure. A better understanding of this point has the potential to considerably improve the performance of QuALe@$T_{\rm eff}$.

To assess the relevance of temperature estimation for QuALe@$T_{\rm eff}$, we also show in Fig. 4 the average performance under quantum assisted learning at a fixed temperature. First, it is worth mentioning that using the physical temperature of the device, $T_{\rm DW2X} = 0.033$ (corresponding to $T_{\rm DW2X} = 12.5$ mK as explained in the caption of Fig. 3), leads to a very poor performance, reaching values $\mathcal{L}_{\rm av} < -14$ (not shown). Fixing the temperature to the average QuALe@$T_{\rm av} \approx 0.1$ over all temperatures found during the run of QuALe@$T_{\rm eff}$ leads to a better performance (red empty circles), but still well below that displayed by QuALe@$T_{\rm eff}$ itself. Fixing the temperature to $T_0 = 0.08 < T_{\rm av}$ (QuALe@$T = 0.08$) and to $T_0 = 0.16 > T_{\rm av}$ (QuALe@$T = 0.16$) leads to a decrease in performance with respect to that displayed with $T_{\rm av}$.

In Fig. 5 we can observe the variation of the effective temperature estimated during a window of 80 iterations of QuALe@$T_{\rm eff}$ (green line). To evaluate whether such a variation is within the finite sampling error, we estimated the effective temperature 15 times at each iteration. The (blue) circles show the median of $T_{\rm eff}$ and the error bars represent the corresponding first and third

quartiles. Clearly, this variation cannot be explained as due to finite sampling error. We emphasize that during the execution of QuALe the effective temperature is estimated only once.

VIII. CONCLUSIONS AND FUTURE WORK

Applications that rely on sampling, such as learning Boltzmann machines, are in general intractable due to long equilibration times of sampling techniques like MCMC [31, 32]. Some authors have conjectured quantum annealing could have an advantage in sampling applications. In this work we proposed a strategy to overcome one of the main limitations when intending to use a quantum annealer to sample from Boltzmann distributions: the determination of effective temperatures. The simple technique proposed in this work uses samples obtained from a quantum annealer (the DW2X at NASA is our experimental implementation) to estimate both the effective temperature and the hard-to-compute term in the log-likelihood gradient, i.e., the averages over the model distribution; these are needed to determine the next step in the learning process. We present a systematic study of the impact of the effective-temperature in the learning of a Chimera-RBM model with 16 visible and 16 hidden units. For doing so, we compared the QuALe algorithm with both instance-dependent effective temperature and different constant effective temperatures to the performance of a CD-k implementation, with k equal to 1, 10, and 100.

The Chimera-RBM model itself is much less powerful than the RBM model. While the former is sparse with a number of parameters increasing linearly with the number of variables, the latter is dense with a number of parameters increasing quadratically. For instance, the Chimera-RBM that we have studied here, with 16 hidden and 16 visible variables, has only about 31% of the weight parameters that a corresponding RBM of the same size has. This is reflected in that a Chimera-RBM, learned either with QuALe or with standard classical techniques, struggles to generate samples faithfully resembling the 4×4 BAS dataset on which it was trained (not shown). In this first study, we have decided to omit any regularization of the learning process. We have done this to keep the focus as clear as possible on the potential gains obtained by using QuALe and to avoid the search of optimal regularization parameters that could be very expensive due to the accessing time to the DW2X. While this may lead to drops in likelihood [45], we expect that the substantial reduction in the number of parameters mentioned above may act as an implicit regularizing sparsity constraint. Since we have neglected regularization altogether in all the learning algorithms, we expect the comparison to be fair. Moreover, as the work by Dumoulin *et al.* [9] suggests, the Chimera-RBM model we have investigated has a limited expressive power. So we have decided to delay the investigation of the role of regular-

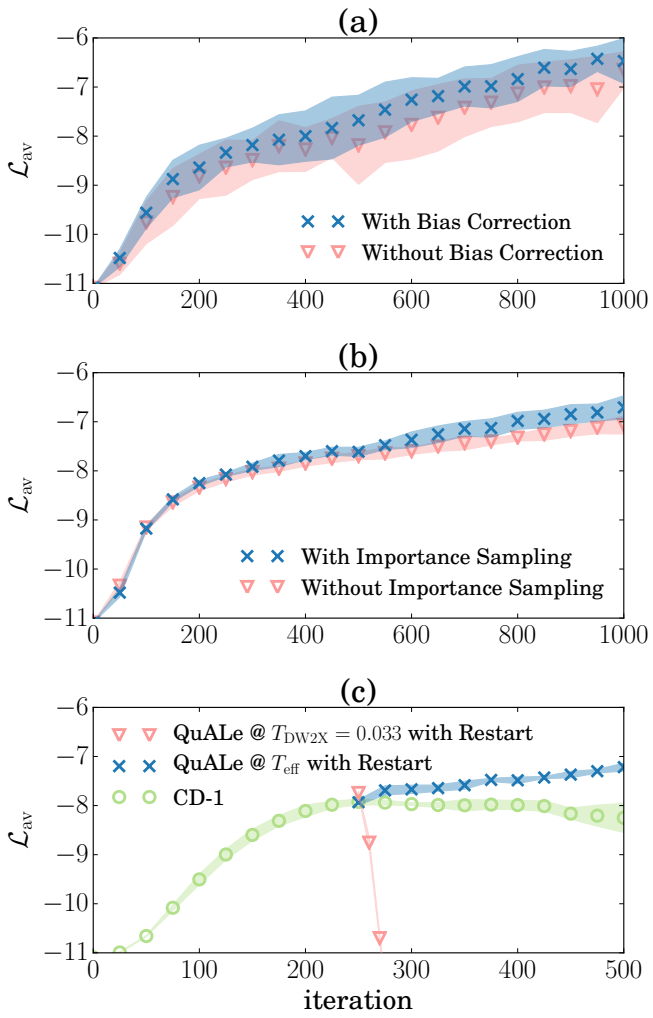

FIG. 3: *Impact of added gadgets:* Average performance of the quantum-assisted learning of a Chimera-RBM on the 4×4 BAS dataset. The performance is measured in terms of the average log-likelihood $\mathcal{L}_{\rm av}$, which has been evaluated exhaustively every fifty iterations. (a) QuALe@$T_{\rm eff}$ with (blue crosses) and without (pink triangles) persistent bias correction. These results are obtained by implementing a Chimera-RBM on five different locations of the DW2X chip and running the QuALe algorithm three times on each location, for a total of fifteen runs. The points correspond to the average of $\mathcal{L}_{\rm av}$ over those fifteen runs and the bands to one standard deviation. (b) QuALe@$T_{\rm eff}$ with (blue crosses) and without (pink triangles) taking into account the samples obtained at $x \neq 1$ for the estimation of the log-likelihood gradient, via importance sampling. The points correspond to the average of $\mathcal{L}_{\rm av}$ over five runs of QuALe on a single location of the DW2X chip. (c) QuALe@$T_{\rm eff}$ (blue crosses) starting after a given number of iterations of CD-1 to escape the noise level of the DW2X. Each point represents the average of $\mathcal{L}_{\rm av}$ over five runs of each algorithm and the error bars correspond to one standard deviation. Notice the dramatic drop in performance of a naive suboptimal version of QuALe@$T_{\rm DW2X}$ that uses the physical temperature instead of estimating $T_{\rm eff}$ as suggested in this work. The value for QuALe@$T_{\rm DW2X}$ is out of the range of the plot and oscillates around $\mathcal{L}_{\rm av} = -14$.

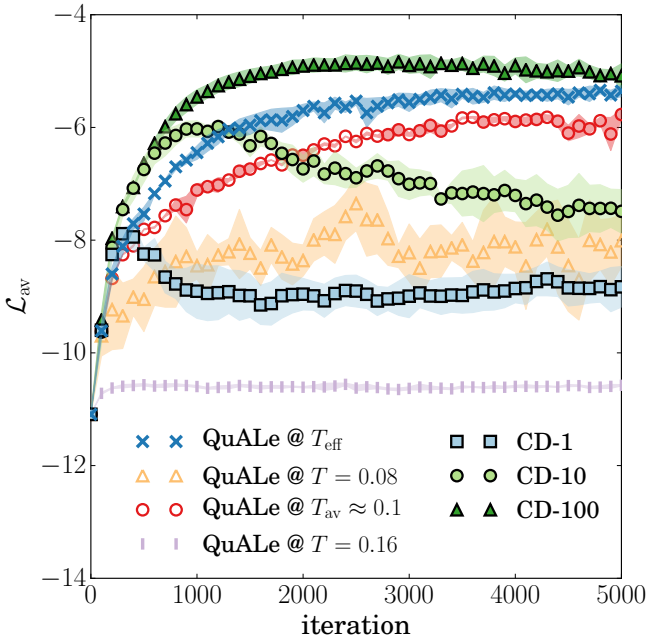

FIG. 4: *Comparison of learning algorithms:* Average performance of different algorithms for the learning of a Chimera-RBM on the 4×4 BAS dataset. The performance is measured in terms of the average log-likelihood $\mathcal{L}_{\rm av}$, which has been evaluated exhaustively every fifty iterations. All points correspond to average of $\mathcal{L}_{\rm av}$ over five different runs on the same location in the DW2X chip, and the bands correspond to one standard deviation. The (blue) diagonal crosses correspond to quantum-assisted learning estimating effective temperatures (QuALe@$T_{\rm eff}$) with the DW2X using $R = 1000$ samples in each iteration for the estimation of both log-likelihood gradient and temperature for actual and rescaled control parameters. The (red) empty circles correspond to fixed-temperature quantum-assisted learning algorithm (QuALe@$T_{\rm av} \approx 0.1$), using the average temperature $T_{\rm av} \approx 0.1$ found during the run of QuALe@$T_{\rm eff}$. The vertical lines and empty triangles correspond to fixed-temperature quantum-assisted learning algorithm using temperatures above and below the average temperature $T_{\rm av}$, namely $T = 0.16$ (QuALe@$T = 0.16$) and $T = 0.08$ (QuALe@$T = 0.08$). The filled squares, circles, and triangles correspond to learning using CD-k for $k = 1$, 10, 100, respectively.

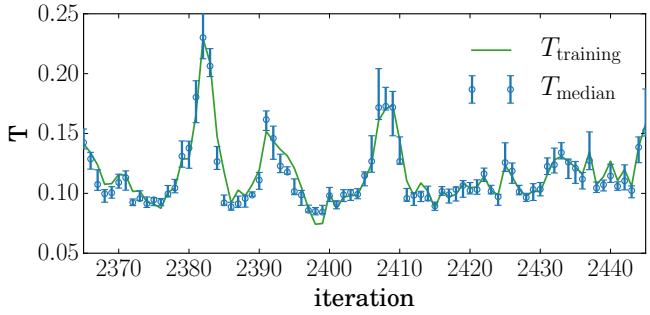

FIG. 5: *Variation of effective temperatures during learning:* The (green) line shows the values of the effective temperature during 80 iterations of QuALe@$T_{\rm eff}$ on the BAS dataset, starting from iteration 2365. The (blue) circles correspond to the median of the effective temperature estimated fifteen times for each instance of the control parameters found during a learning session. The error bars represent the first and third quartiles.

ization for when we deal with more expressive models that can be naturally represented in a Chimera topology.

RBMs have the nice feature that sampling in one layer conditioned to a configuration in the other layer can be done in parallel and in one step; this is one of the main reasons for their wide adoption. This feature does not hold true anymore once we have non-trivial lateral connections in one of the layers, which is the concept behind more powerful Boltzmann machines [56, 57]. We think this is one of the most promising directions to explore with the quantum-assisted learning (QuALe) algorithm. By restricting QuALe to study RBM or Chimera-RBM models, we are paying the price of using a device that is in principle more powerful, but we are not taking advantage of having a more general model. It is important to investigate how to take full advantage of the DW2X by designing more suitable models based on the Chimera graph. An interesting possibility is the one explored in Ref. [5] where the Chimera graph of the DW2X is used as a hidden layer to build a semi-restricted Boltzmann machine, which therefore has lateral connections in the hidden layer. When dealing with more general Boltzmann machines it would be interesting to compare the performance of QuALe against mean field methods. Recently, there has been interest in applying mean field techniques for learning restricted Boltzmann machines too [58, 59]. Future work should explore how the performance of mean field techniques compares with QuALe's.

However, the goal of this first QuALe implementation on small Chimera-RBMs serves several purposes. When dealing with large datasets the log-likelihood cannot be exhaustively computed due to the intractability of computing the partition function. Log-likelihood is the gold standard metric, but it becomes intractable for large systems. In these cases, other performance metrics such as reconstruction error or cross-entropy error turn out to be more convenient, but although widely used, they are rough approximations to the log-likelihood [60]. If we were to use these proxies we could not be sure that we would be drawing the right conclusions. This justifies why we used a moderately small dataset with 16 visible and 16 hidden units, and even though computing the log-likelihood was computationally expensive for the study performed here, having 32 units in total was still a manageable size. Through the computation of the exact likelihood we were able to examine in more detail some of the goals proposed here: being able to assess the best effective temperature fit to the desired Boltzmann distribution and to show that using a constant temperature different from the one estimated with our approach might lead to severe suboptimal performance.

Another aspect we explored in this study was to go beyond the conventional CD-1, with the purpose of having a fairer comparison to the results that might be expected from the entirely classical algorithm counterpart. Previous results from our research group [61], as well as others reported by other researchers [14, 62], are limited to comparing the performance of quantum annealers to the quick but suboptimal CD-1. As shown in those studies, even with a suboptimal constant temperature one might be drawn to conclude that QuALe is outperforming conventional CD. Similar conclusions might be drawn from the curves for constant but suboptimal $T = 0.08$ and $T_{\rm av} \approx 0.1$ vs. CD-1 in Fig. 4. As shown in Fig. 4, this conclusion does not hold anymore for higher values of k, while the method using the effective-temperature estimation proposed here is the only one showing a steady increase in performance, close to matching the largest value of k tried here, i.e. $k = 100$.

Another important point to investigate in the future is whether the differences observed in performance remain for larger and more complex datasets. We would expect that the performance of CD-k degrades with larger instances as equilibration times are expected to grow fast with the number of variables once the probability distribution starts having non-trivial structure. From this perspective, it is important to notice that QuALe is expected to display a more uniform exploration of configuration space.

A related important question has to do with the scalability of our temperature estimation technique, i.e. how should the number of samples grow with problem size? In principle, as long as the quantum annealer converges to an approximately Boltzmann distribution and the effective temperature remains constant under rescalings of its control parameters, our method is exact given enough samples. We have left this question for future work as we consider that there are more pressing issues, i.e. limited connectivity and noise, that need to be addressed before we can say something conclusive about scalability. Needless to say, the validity of the assumptions on which our work relies should also be investigated in more detail. It is also important to devise more controlled experiments that allow us to isolate the different phenomena involved. Two months after submission of this manuscript, we learned of ongoing work addressing some of these issues and putting forward other temperature estimation techniques [63]. Finally, an investigation on the bias and variance of our effective temperature estimator is an interesting theoretical question that we expect to address in future work.

There are other ways in which the ideas explored here could be extended. For instance, we can go beyond restricted Boltzmann machines to build deep learning architectures or beyond unsupervised learning to build discriminative models. In principle the speed of learning could be increased by adding a 'momentum' term to the gradient-ascent learning rule [39]. Indeed, Adachi and Henderson have started exploring these ideas in a contemporary work [14]. Instead, we have focused on first trying to better understand the basics before adding more (classical) complexity to the learning algorithms that we feel have the risk to obscure the actual contributions from the new approach.

Appendix A: Comparison to alternative temperature estimation techniques

Here we discuss alternative techniques to approximately estimate the instance-dependent effective temperature $T_{\rm eff}$, which are in principle efficient too, and show evidence that our method produces superior results.

One of the mainstream approaches in statistical physics to estimate parameters of an Ising model goes under the name of *inverse Ising model* [64–69]. One of the most investigated techniques for solving the inverse Ising model relies on mean field approximations [65–67], due to its relative simplicity. These techniques fail, though, for low temperatures where low-energy configurations are arranged in a non-trivial clustered phase [69]. On the other hand, the so-called pseudo-likelihood method [68] is recognized as the state-of-the art in solving this problem. Recently, it has been suggested that by suitably introducing information about the clustered phase into mean field methods, these can yield comparable results to the pseudo-likelihood method [69].

We first devised a simple strategy to test the feasibility of a mean field approach before attempting to develop a technique specifically targeted to the estimation of $T_{\rm eff}$ alone. Indeed, since we know the control parameters J_{ij} and h_i, we can in principle estimate $T_{\rm eff}$ by first determining W_{ij} and b_i using the Bethe approximation [66], and then finding the value of $T_{\rm eff}$ that minimizes some distance between the control parameters and the estimated ones. However, the estimation of W_{ij} and b_i using the samples from the DW2X along the learning path only produces real values up to about the first hundred iterations (not shown). This suggests the Bethe approximation is not suitable for the parameter regime traversed when learning the BAS dataset studied here.

Since, as we mentioned above, the pseudo-likelihood method [68] is considered the state of the art technique for estimating the parameters of an Ising model we will focus from now on in such an approach. We will see that our method displays a much better performance on the BAS dataset studied here.

Given a set of samples $\mathcal{D} = \{\mathbf{s}^1, \ldots, \mathbf{s}^D\}$, where $\mathbf{s}^d = (s_1^d, \ldots, s_N^d)$ with $d = 1, \ldots, D$, generated by a quantum annealer with control parameters J_{ij} and h_i, we can estimate the effective temperature $T_{\rm eff}$ by maximizing the *average pseudo-likelihood* [68]

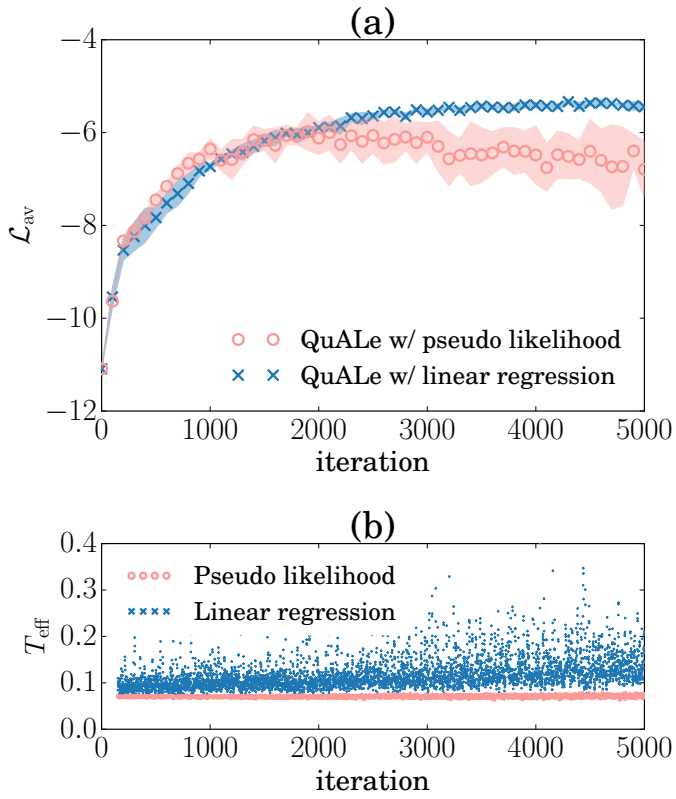

FIG. 6: *Comparison with pseudo-likelihood estimation:* (a) Performance of the quantum-assisted learning of a Chimera-RBM on the 4×4 BAS dataset using the two different temperature estimation techniques described in Sec. V: one (blue crosses) based on linear regression (Eq. (13)) and the other (pink circles) on pseudo-likelihood maximization (Eq. (A1)). The performance is measured in terms of the average log-likelihood $\mathcal{L}_{\rm av}$, which has been evaluated exhaustively every fifty iterations. The points correspond to the average of $\mathcal{L}_{\rm av}$ over five runs and the bands to one standard deviation. (b) Variation of the effective temperatures during one of the five runs using linear regression (blue crosses) and pseudo-likelihood maximization (pink circles). Temperature estimation begins at iteration 100 after restarting from CD-1.

$$\Lambda(T_{\rm eff}) = -\frac{1}{N\,D}\sum_{i=1}^{N}\sum_{d=1}^{D}\ln\left\{1+\exp\left[-\frac{2\,s_i^d}{T_{\rm eff}}\left(h_i+\sum_{j\in\partial i}J_{ij}\,s_j^d\right)\right]\right\}, \tag{A1}$$

where ∂i denotes the set of neighbors of i.

In contrast to the approach in Ref. [68], here the only unknown is $T_{\rm eff}$. We can find a maximum average pseudo-likelihood estimator for the effective temperature $T_{\rm eff}^{\rm PL} = \arg\max_{T_{\rm eff}} \Lambda(T_{\rm eff})$ via second order Newton's method [68]. In our experiments, we start from $T_{\rm eff} = 1$ and iterate until the update is smaller than a tolerance level of 10^{-5}. Fig. 6a shows a comparison of the performance of our quantum-assisted learning algorithm QuALe@$T_{\rm eff}$ with $T_{\rm eff}$ estimated with the pseudo-likelihood method (pink circles) as described here and estimated with the method introduced in Sec. V (blue crosses) of the present work. We can observe that while QuALe@$T_{\rm eff}$ with the pseudo-likelihood method performs better on the first about 1000 iterations, QuALe@$T_{\rm eff}$ with linear regression performs better afterwards, reaching higher values for the likelihood function. Fig. 6b shows the values of effective temperatures estimated by the two techniques along the learning path; interestingly, the effective temperatures estimated by the pseudo-likelihood (pink points on the bottom) are consistently smaller and have less variability than those estimated with our linear regression technique (blue points on the top).

Acknowledgements

This work was supported by NASA Ames Research Center. The authors would like to thank V. M. Janakiraman, Z. Jiang, T. Lanting, E. Rieffel, N. Wiebe, and B. Jacobs for useful discussions.

[1] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, arXiv:0811.0416 (2008).
[2] H. Neven, V. S. Denchev, M. Drew-Brook, J. Zhang, W. G. Macready, and G. Rose, in *Demonstrations at NIPS-09, 24th Annual Conference on Neural Information Processing Systems* (2009), pp. 1–17.
[3] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, arXiv:0912.0779 (2009).
[4] Z. Bian, F. Chudak, W. G. Macready, and G. Rose, Tech. Rep., D-Wave Systems (2010).
[5] M. Denil and N. De Freitas, NIPS Deep Learning and Unsupervised Feature Learning Workshop (2011).
[6] V. S. Denchev, N. Ding, S. Vishwanathan, and H. Neven, arXiv:1205.1148 (2012).
[7] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv:1307.0411 (2013).
[8] K. Pudenz and D. Lidar, Quantum Information Processing **12**, 2027 (2013).
[9] V. Dumoulin, I. J. Goodfellow, A. C. Courville, and Y. Bengio, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.* (2014), pp. 1199–1205.
[10] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics (2014).
[11] P. Rebentrost, M. Mohseni, and S. Lloyd, Phys. Rev. Lett. **113**, 130503 (2014).
[12] K. M. S. Nathan Wiebe, Ashish Kapoor, arXiv:1412.3489 (2015).
[13] S. Aaronson, Nature Physics **11**, 291 (2015), commentary.
[14] S. H. Adachi and M. P. Henderson, arXiv:1510.06356 (2015).
[15] N. Chancellor, S. Szoke, W. Vinci, G. Aeppli, and P. A. Warburton, arXiv:1506.08140 (2015).
[16] Mohammad H. Amin and Evgeny Andriyash and Jason Rolfe and Bohdan Kulchytskyy and Roger Melko, arXiv:1601.02036 (2016).
[17] A. Finnila, M. Gomez, C. Sebenik, C. Stenson, and J. Doll, Chemical Physics Letters **219**, 343 (1994).
[18] T. Kadowaki and H. Nishimori, Phys. Rev. E. **58**, 5355 (1998).
[19] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, Science **292**, 472 (2001).
[20] F. Gaitan and L. Clark, Phys. Rev. Lett. **108**, 010501 (2012).
[21] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, Sci. Rep. **2**, 571 (2012).
[22] Z. Bian, F. Chudak, R. Israel, B. Lackey, W. G. Macready, and A. Roy, Frontiers in Physics **2** (2014), ISSN 2296-424X.
[23] B. O'Gorman, R. Babbush, A. Perdomo-Ortiz, A. Aspuru-Guzik, and V. Smelyanskiy, The European Physical Journal Special Topics **224**, 163 (2015), ISSN 1951-6355.
[24] E. G. Rieffel, D. Venturelli, B. O'Gorman, M. B. Do, E. M. Prystay, and V. N. Smelyanskiy, Quantum Information Processing **14**, 1 (2015), ISSN 1570-0755.
[25] Perdomo-Ortiz, A., Fluegemann, J., Narasimhan, S., Biswas, R., and Smelyanskiy, V.N., Eur. Phys. J. Special Topics **224**, 131 (2015).
[26] A. Perdomo-Ortiz, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, arXiv:1503.01083 (2015).
[27] D. Venturelli, D. J. Marchand, and G. Rojo, arXiv:1506.08479 (2015).
[28] M. H. Amin, Phys. Rev. A **92**, 052323 (2015).
[29] A. Perdomo-Ortiz, B. O'Gorman, J. Fluegemann, R. Biswas, and V. N. Smelyanskiy, Sci. Rep. **6**, 18628 (2016).
[30] J. E. Dorband, in *ITNG, Washington DC* (2015), pp. 703–707.
[31] A. Sinclair and M. Jerrum, Inf. Comput. **82**, 93 (1989), ISSN 0890-5401, URL `http://dx.doi.org/10.1016/0890-5401(89)90067-9`.
[32] A. Frigessi, F. Martinelli, and J. Stander, Biometrika **84**, 1 (1997).
[33] P. M. Long and R. Servedio, in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, edited by J. Frnkranz and T. Joachims (Omnipress, 2010), pp. 703–710.
[34] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).
[35] P. Smolensky, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, edited by D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group (MIT Press, Cambridge, MA, USA, 1986), chap. Information Processing in Dynamical Systems: Foundations of Harmony Theory, pp. 194–281, ISBN 0-262-68053-X.
[36] G. E. Hinton, Neural Computation **14**, 1771 (2002).
[37] G. E. Hinton and T. J. Sejnowski (MIT Press, Cambridge, MA, USA, 1986), chap. Learning and Relearning in Boltzmann Machines, pp. 282–317, ISBN 0-262-68053-X.
[38] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms* (Cambridge University Press, New York, NY, USA, 2002), ISBN 0521642981.
[39] A. Fischer and C. Igel, in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (Springer, 2012), pp. 14–36.
[40] N. Le Roux and Y. Bengio, Neural Computation **20**, 1631 (2008), ISSN 0899-7667.
[41] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, et al., Nature **473**, 194 (2011), ISSN 0028-0836.
[42] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley,

J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, et al., Phys. Rev. B **82**, 024511 (2010).
[43] P. Shor, in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on* (1994), pp. 124–134.
[44] Y. Bengio and O. Delalleau, Neural Computation **21**, 1601 (2009).
[45] A. Fischer and C. Igel, in *Proceedings of the 20th International Conference on Artificial Neural Networks: Part III* (Springer-Verlag, Berlin, Heidelberg, 2010), ICANN'10, pp. 208–217, ISBN 3-642-15824-2, 978-3-642-15824-7.
[46] I. Goodfellow, Y. Bengio, and A. Courville (2016), book in preparation for MIT Press, URL `http://goodfeli.github.io/dlbook/`.
[47] T. Albash, S. Boixo, D. A. Lidar, and P. Zanardi, New Journal of Physics **14**, 123016 (2012).
[48] V. N. Smelyanskiy, D. Venturelli, A. Perdomo-Ortiz, S. Knysh, and M. I. Dykman, arXiv:1511.02581 (2015).
[49] S. Boixo, V. N. Smelyanskiy, A. Shabani, S. V. Isakov, M. Dykman, V. S. Denchev, M. Amin, A. Smirnov, M. Mohseni, and H. Neven, Nature Communications **7**, 10327 (2014).
[50] V. Balasubramanian, Neural Comput. **9**, 349 (1997), ISSN 0899-7667.
[51] I. J. Myung, V. Balasubramanian, and M. A. Pitt, Proceedings of the National Academy of Sciences **97**, 11170 (2000).
[52] I. Mastromatteo, ArXiv e-prints (2013), 1311.0190.
[53] M. Habeck, arXiv:1504.00053 (2015).
[54] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006).
[55] Y. Bengio, Found. Trends Mach. Learn. **2**, 1 (2009).
[56] H. Schulz, A. C. Müller, and S. Behnke, in *ESANN* (2010).
[57] R. Salakhutdinov, Tech. Rep. (2008).
[58] H. Huang and T. Toyoizumi, Phys. Rev. E **91**, 050101 (2015).
[59] M. Gabrié, E. W. Tramel, and F. Krzakala, arXiv:1506.02914 (2015).
[60] G. E. Hinton, in *Neural Networks: Tricks of the Trade (2nd ed.)*, edited by G. Montavon, G. B. Orr, and K.-R. Mller (Springer, 2012), vol. 7700 of *Lecture Notes in Computer Science*, pp. 599–619.
[61] M. Benedetti, Master's thesis, Université Lumière Lyon 2, France (2015).
[62] G. Rose, *First ever DBM trained using a quantum computer*, `https://dwave.wordpress.com/2014/01/06/first-ever-dbm-trained-using-a-quantum-computer/` (2014), [Online; accessed 22-October-2015].
[63] J. Raymond, S. Yarkoni, and E. Andriyash, arXiv:1606.00919 (2016).
[64] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, Nature **440**, 1007 (2006).
[65] M. Mézard and T. Mora, Journal of Physiology-Paris **103**, 107 (2009), ISSN 0928-4257, Neuromathematics of Vision.
[66] F. Ricci-Tersenghi, Journal of Statistical Mechanics: Theory and Experiment **2012**, P08015 (2012).
[67] H. C. Nguyen and J. Berg, Journal of Statistical Mechanics: Theory and Experiment **2012**, P03004 (2012).
[68] E. Aurell and M. Ekeberg, Phys. Rev. Lett. **108**, 090201 (2012).
[69] A. Decelle and F. Ricci-Tersenghi, arXiv:1501.03034 (2015).
[70] M. Mézard and A. Montanari, *Information, Physics, and Computation*, Oxford Graduate Texts (Oxford University Press, Oxford, 2009), ISBN 9780198570837.