AFRL-RI-RS-TR-2019-148

DESIGN AND APPLICATION OF QUANTUM ANNEALING SAMPLING ALGORITHMS

BOOZ ALLEN HAMILTON INC.

JULY 2019

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the 88th ABW, Wright-Patterson AFB Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2019-148 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /

PATRICK M. HURLEY
Work Unit Manager

/ S /

QING WU
Technical Advisor, Computing
& Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
JULY 2019	FINAL TECHNICAL REPORT	MAY 2018 – MAY 2019

4. TITLE AND SUBTITLE

DESIGN AND APPLICATION OF QUANTUM ANNEALING SAMPLING ALGORITHMS

5a. CONTRACT NUMBER
FA8750-18-C-0097

5b. GRANT NUMBER
N/A

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)

Casey Tomlin
JD Dulny
Joseph Riolo

5d. PROJECT NUMBER
CYDT

5e. TASK NUMBER
BA

5f. WORK UNIT NUMBER
H1

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Booz Allen Hamilton Inc.
Suite 300, 8283 Greensboro Dr.
Mclean, VA 22102-3830

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RITA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2019-148

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA# 88ABW-2019-3296
Date Cleared: 03 July 2019

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of this effort was to investigate the utility of hardware quantum annealing devices in two near-term applications: Circuit fault diagnosis; and machine learning. This report summarizes the technical work performed, results published, software developed, lessons learned, and future directions of study. Key accomplishments include:

- A new efficient algorithm for domain decomposition, allowing large optimization and sampling problems to be solved on small quantum hardware;
- A library of optimal Hamiltonians for common circuits;
- Software for rapid experimentation in quantum-assisted unsupervised Boltzmann machine training; and
- Training of several quantum hardware-native and non-native Boltzmann machines with state-of-the-art performance on standard benchmarks.

15. SUBJECT TERMS

quantum annealing, Boltzmann machines, quantum sampling, circuit fault diagnosis, and machine learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**PATRICK M. HURLEY**
U	U	U	UU	18	19b. TELEPHONE NUMBER (*Include area code*) **315-330-3624**

TABLE OF CONTENTS

TABLE OF FIGURES

1 SUMMARY

The objective of this effort was to investigate the utility of hardware quantum annealing devices in two near-term applications: Circuit fault diagnosis; and machine learning. This report summarizes the technical work performed, results published, software developed, lessons learned, and future directions of study. Key accomplishments include

- A new efficient algorithm for domain decomposition, allowing large optimization and sampling problems to be solved on small quantum hardware;
- A library of optimal Hamiltonians for common circuits;
- Software for rapid experimentation in quantum-assisted unsupervised Boltzmann machine training; and
- Training of several quantum hardware-native and non-native Boltzmann machines with state-of-the-art performance on standard benchmarks.

This document is the Final Technical Report (FTR - CDRL A006) for the Design and Application Of Quantum Annealing Sampling Algorithms effort. Section 2 discusses our work on quantum annealing for circuit fault diagnosis, while Section 3 discusses quantum sampling applications to machine learning. Each contains recommendations detailing important lessons and suggestions for future inquiry.

2 INTRODUCTION

2.1 Circuit Fault Diagnosis

A main application of quantum annealing has been solving constrained optimization. In our previous work, we focused on two easily formulated constrained optimization problems: decoding low-density parity check codes [2] and circuit fault diagnosis [3]. In both works we emphasized that the two main obstacles to the use of quantum annealing are

1. Efficiently mapping problem constraints to unconstrained optimization functions consistent with the hardware topology; and
2. Methods for decomposing industrial-sized problems into subproblems that fit on current hardware and reconstituting solutions found by an annealer on individual subproblems into a solution of the whole problem.

In a follow-on work [6] we showed using quantum annealers for circuit fault diagnosis shows promise for a quantum speed up, however neither of this core obstacles were addressed. The purpose of this research was to make in-roads into both these problems with an eye to solving large circuit fault diagnosis problems.

The challenge of mapping hard constraints to unconstrained optimization functions is often called the “inverse Ising” problem: given all configurations that satisfy the constraint, produce a Hamiltonian whose ground state manifold is precisely these configurations. This is a very difficult problem in general. However, for many applications, including circuit fault diagnosis, the constraints are given as many local sparse conditions. In this case an individual constraint involves only a few problem variables and so we can map each of these separately and sum the resulting Hamiltonians. Ad hoc methods are used for mapping, which generally leads to suboptimal Hamiltonians; Appendix C of [6] gives examples for circuit fault diagnosis problems. As argued in [2,3] a metric for measuring the quality of a mapping is the spectral gap of the resulting

Hamiltonian. The larger the spectral gap, the greater the suppression of sampling from states that do not satisfy the desired constraint. Our goal was to codify what it means for a Hamiltonian to be optimal with respect to this metric and develop new methods to construct them.

The most limiting factor on the use of quantum annealers is the modest size of hardware available. Given overheads in mapping and embedding real world problems, current hardware can typically only handle problems with a few hundred binary variables. Hence most studies on the use of quantum annealers are relegated to toy problem sizes. The current D-Wave API [9] offers the method qsage, which allows one to solve larger problems by iteratively selecting subproblems by clamping all but a few hundred variables and using the hardware to minimize over the unclamped variables; in [2] we referred to this method as large neighborhood local search. In this work and later in [3], we explored several methods of decomposing problems that were amenable to the use of quantum annealing on the resulting subproblems. This included preliminary work on adapting belief propagation to problem decomposition and reconstitution. It was thought that this approach would be particularly powerful since belief propagation, and related algorithms such as survey propagation, are some of the best methods for solving constraint satisfaction problems such as SAT. However, the results of those studies did not support this tenet. Our goal was to re-examine the method of domain decomposition and develop an effective algorithm for decomposition and reconstitution.

3 METHODS, ASSUMPTIONS, AND PROCEDURES

3.1 Convex Geometry And Optimal Hamiltonians

Current quantum annealing hardware is restricted to Ising Hamiltonians

$$H(s) = \sum_{i=0}^{n} h_i s_i + \sum_{(i,j)\in E} J_{jk} s_j s_k. \quad (1)$$

where the set of allowed edges E is very sparse. As a consequence, much effort has been dedicated to embedding: ways to represent a general Ising Hamiltonian as one of form (1) using graph minors. Since this is mature compared to mapping methods, we decided not to place restrictions on our sets of edges. Instead we focused just on the second main constraint: the allowable ranges for h_i and J_{jk} are bounded. In the current D-Wave hardware, these are restricted to $-2 \leq h_i \leq 2$ and $-1 \leq J_{jk} \leq 1$; sending a problem with coefficients out of this range results in the controller rescaling the entire problem so these conditions are met. Bounds such as this imply that the set of allowable Hamiltonians (1) we may use forms a convex set.

Our goal was to find the largest spectral gap among the Hamiltonians of a given convex set. The constraints on the coefficients as given above are linear, as is the requirement that a given state be a ground state or excited state. Hence, we can recast the problem of finding optimal Hamiltonians for a given constraint as a linear program, albeit one with exponentially many facets. In the case of constraints arising from circuit fault diagnosis, these only have 3 – 6 variables and so such linear programs are easy to solve when the problem has a solution.

Given a collection of states, there generally is not a Hamiltonian of form (1) for which these states compose the ground state manifold. Stated in terms of the associated linear program, the maximal spectral gap is zero: there is no separation between the states that satisfy the constraint versus those that do not. To find Hamiltonians with positive spectral gap, we must add ancillary

spins to our system, however we do know a priori how to set the values of the ancillary variables. This results in an exponential number (in the number of ground states and ancillary variables) of linear programs to be solved. Fortunately, circuit fault diagnosis constraints are just small enough that this is just feasible using enumerative methods.

3.2 Symmetry And Noncommutative Cohomology

Enumerating over the possible ancillary settings and solving the resulting linear programs results in hundreds, and sometimes thousands, of optimal Hamiltonians for a given constraint. The main factor contributing to this phenomenon is symmetry. Consider the case of a NAND gate

$$(2) x_0 \leftarrow \neg(x_1 \wedge x_2). \tag{2}$$

The associated constraint consists of all configurations that are valid for this computation: $(x_0, x_1, x_2) = (1,0,0), (1,0,1), (1,1,0), (0,1,1)$. This relation has a clear symmetry: we can exchange $x_1 \leftrightarrow x_2$ and leave the relation invariant. Therefore, if we find one optimal Hamiltonians, then exchange the spins $s_1 \leftrightarrow s_2$ produces another optimal Hamiltonian. While this is not much of a problem for a feasible linear program such as NAND in (2), for gates where we must add ancillary spins, and collection optimal Hamiltonians for hundreds of possible setting, even a small symmetry group can produce an explosion of morally equivalent Hamiltonians.

To combat this, we explored how symmetry effects the search for optimal Hamiltonians. For feasible linear programs, this allowed us to do a full symmetry reduction and search for optimal Hamiltonians in the invariant ring of the symmetry group. We also formalized how to lift a symmetry group, such as the exchange symmetry (2) above, to the system with ancillary spins attached. The action on the ancillary spins was proven to satisfy a cocycle condition, and so the collection of spin configurations can be partitioned according to the induced noncommutative group cohomology class they define.

3.3 Regional Free Energy And Domain Decomposition

Given an Ising model, there are two canonical ways to cut it up. One can partition its set of spins and consider the resulting Ising models on each part; interactions between variables in different parts are lost and need to be treated somehow. We explored a different method: one partitions the set of interactions and using the supported spins of each part derives an associated Ising model. In this latter "domain decomposition" method no interactions are lost, however spins that appear in the support of interactions in different regions will be duplicated. This is a common feature in message passing algorithms [1]. In belief propagation, information about how a variable is set in other subproblems is passed to a given subproblem and incorporated into its solution [5].

A well-known technique for deriving the belief propagation algorithm is through minimizing the free energy [8]. This method is particularly well suited to this study as the free energy minimum is the Boltzmann distribution. We explored the effect of decomposing free energies using a generalized analogue of the Bethe approximation [7]. Using such we developed a new algorithm where the ideal message is an entire Boltzmann distribution. Consequently, sampling from a Boltzmann distribution arises as the natural method for computing a message.

4 RESULTS AND DISCUSSION

In our previous work on domain decomposition [2,3] we used heuristic methods to develop belief propagation style algorithms. Consequently, normalization factors and message definitions were assumed by analogy to the usual algorithms. In rigorously deriving the algorithm from minimizing free energy approximation we found that some of these assumptions were wrong. We proved that if the new algorithm converges then one can reconstruct a distribution that minimized an analogue of the Bethe approximation. We also illustrated how one can empirically estimate the messages from all the duplicated variables of a region using a single run of the quantum annealer, opposed to in our prior work where one would need to solve a different subproblem for each variable. This work was accepted for a contributed talk at the conference Adiabatic Quantum Computing 2019.

We developed code that produces a complete library of optimal Hamiltonians for all the gates (NOOP, NOT, AND, OR, NAND, NOR, and XOR) and error models (stuck-at-0, SA0, or stuck-at-1, SA1) that appear in the ISCAS ‘85 synthetic benchmark [4]. As indicated above there are typically several hundred models associated to each of the more complicated examples. Where possible we have given parameterized Hamiltonians, anticipating changes to the bounds of the Hamiltonian coefficients of future hardware.

Along the way we proved a powerful theorem: if a constraint has an optimal Hamilton, then it has one that is invariant under the natural symmetry group of the constraint. This allows us to symmetry reduce, and search for optimal Hamiltonians among the group-invariant ones. Unfortunately, this technique currently applies only to systems that do not use ancillary spins.

For certain gates and error models using ancillary spins (namely XOR, NOR>SA0, NOR>SA1, and NOR>SA0>SA1), we have identified the associated symmetry group and have selected a characteristic example from each orbit. Hamiltonians from different orbits typically have different interaction graphs and including such models in a library should be useful when embedding on non-Chimera topologies.

4.1 Software

We have developed a very early prototype implementation of the above domain decomposition algorithm, and a user guide accompanying this report details its functionality.

5 CONCLUSIONS

Symmetry reduction will be a critical tool when searching for optimal Hamiltonians larger than those used in circuit fault diagnosis benchmarks. We have proved how to lift symmetries when adding ancillary spins to a system using noncommutative group cohomology, however owing to the difficult nature of the subject have only been able to treat one example completely (adding a single ancillary spin for an XOR gate). To make this technique usable at scale, we need to develop software that can compute such cohomology sets and their actions. For the constraints in circuit fault diagnosis problems, we are on the cusp of being able to do so. Further afield is characterizing any relationship between the orbits and optimal Hamiltonians: when does an orbit contain an optimal Hamiltonian?

With the rigorous formulation of a sound domain decomposition, we need only implement it to test its efficiency. For fault diagnosis there is still some work to be done in combining optimal

Hamiltonians in a way to make embedding onto a Chimera or Pegasus topology efficient before we can test the decomposition method. However, in machine learning, we are just at the stage of using the algorithm.

5.1 Machine Learning Applications

In this section, after a brief introduction and literature survey, we describe our investigation into hardware quantum annealer-assisted unsupervised training of Boltzmann machines. We refer the reader to [7] for further details on classical Boltzmann machines.

Traditional wisdom is that the goal of quantum annealing is to find the ground state of a user-programmable Ising Hamiltonian; today however, available hardware does not appear to offer a significant advantage when used to find the absolute ground state [10]. The current generation of quantum annealers are instead able to sample very rapidly from low energy configurations [11], and there is evidence of significant advantage over software techniques at tasks related to sampling from these states [12].

One such sampling task appears in the unsupervised training of Boltzmann machines [13–18]. Training these models to exactly maximize dataset likelihood is classically intractible, and instead one usually resorts to approximating certain quantities that appear during training with samples from the relevant Boltzmann distributions, a classically expensive task itself [19]. Studies suggest that quantum annealers (under certain conditions) generate approximately Boltzmann-distributed samples [20–22], a fact that has led to several recent investigations of the feasibility of using such hardware for their training [23–36]. Experiments have also indicated some non-classical behavior in these systems [37], and therefore one should not expect perfect Boltzmann statistics. The going thesis is that algorithms that require sampling from Boltzmann or Boltzmann-like distributions will see a decrease in cost if they can be tailored to use quantum annealing hardware.

Efficient sampling from a Boltzmann-like distribution can be used to estimate the parameters of a restricted Boltzmann machine (RBM) [26,38,39]. RBMs can be used to build powerful classifier programs for use in a variety of crucial applications, including, for example, image classification, data filtering, and identification of friend or foe [52-54]. More generally they can be used to model arbitrary probability distributions in unsupervised machine learning pipelines [16]. Even with techniques such as Markov chain Monte Carlo, certain types of Bayesian estimation can often be intractable given one must evaluate high-dimensional integrals/sums. Methods like rejection sampling are also inefficient as the dimension of the space grows; a large majority of the samples generated end up being rejected, resulting in significant computational waste. Using quantum annealing hardware as an efficient sampler gives the user access to efficient Bayesian estimation via sampling that would otherwise be inaccessible. Given the recent success and proliferation of work in neural network-based image recognition [40,55] and other forms of deep learning, quantum annealing hardware shows promise as a practical tool for machine learning and data science. The hardware can be used for the efficient generation of Boltzmann-like distributions that can be tuned and specified by the graphical network of quantum bits (qubits) in the chip itself.

There are, however, several well-known limitations imposed by current hardware [41,42]. Limited qubit connectivity is a major hurdle, and as real analog devices, the adverse effects of noise, imprecise control, and non-zero operating temperatures are expected [43]. Devices can be regarded as approximately operating at some effective inverse temperature, β_{eff}, but in general

this is different from physical temperature, and indeed even problem instance-specific and non-constant across a chip [20,22,36,44–46]. To compete with classical techniques, previous studies of annealer-trained RBMs found it necessary to estimate β_{eff}, which can become a prohibitively expensive calculation as the problem size grows [39]. As described in [36], cheaper methods are available, or one may work with simpler, fully visible models in order to avoid calculating β_{eff} [35], though at the cost of model complexity. Other interesting model paradigms [47] use both classical and quantum Boltzmann-distributed samples. In this work we limit our scope to Boltzmann machines where annealing hardware is used as a classical sampler.

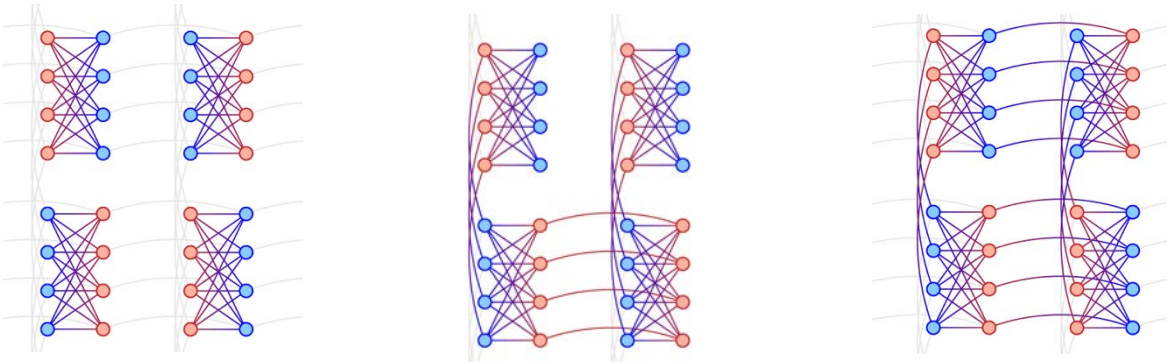
6 METHODS, ASSUMPTIONS, AND PROCEDURES

Here we describe our investigation of efficient and practical implementations of Boltzmann machines on quantum annealers. We propose and evaluate a scheme in which one need not compute effective temperatures, without comprimising model complexity, and which remains applicable to general (non-restricted) Boltzmann machines, without significant computational overhead. We provide the necessary gadgets that enable such a scheme.

We evaluate a selection of hardware-natively embeddable as well as minor-embeddable architectures to begin to tease out the role of edge density and latent structure in training quality. Conventional wisdom indicates that sparse connectivity and models with fewer latent variables always lead to poor training. We test this assumption where underlying hardware architecture may bring out unexpected trends.

Figure 1 displays a subset of the native architectures considered. In addition to these, we have initiated experiments with a few more general models, featuring connected chains/loops of visible qubits, and structures with arbitrary (as allowed by the hardware) depth (see utils.py in the accompanying code for details and other architecture generators).

a) b) c)



d)

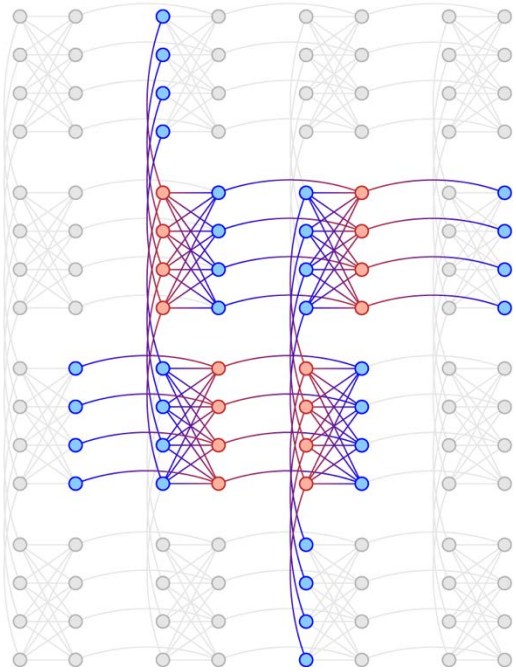


Figure 1: Natively embeddable Boltzmann machine graphs.

Red and blue nodes represent visible and hidden nodes respectively. (a) Disconnected RBM. Each tile is a fully-connected bipartite $\boldsymbol{K_{4,4}}$ graph, but there are no inter-tile connections. (b) Tight semi-restricted Boltzmann machine (SRBM). Each tile is itself an RBM, but the inter-tile connections link some visible units. (c) Tight RBM. Same as the disconnected version but allows all inter-tile connections. (d) Maximal RBM. The tight RBM neglects some available hidden units, which this case includes. These are however effectively biases since each additional hidden unit is only connected to a single visible node.

For example, a 'deep RBM' starts with the visible qubits as in Figure 1 (c) and adds hidden qubits by joining all subgraphs of radius 2 centered on each visible qubit.

We trained models on the bars and stripes (BAS) dataset, a standard toy image data set often used to test generative modeling techniques. Each member consists either of some (black and white) horizontal stripes or vertical bars. Which bars or stripes will be white (on a black background) in the $n \times n$ pixel case is determined by a uniform choice of subset of $[n]$, along with the uniform choice of whether it will be bars or stripes, so the complete data set has $T = 2^{n+1}$ members. Each non-uniform pattern appears once in the data set, while the two uniform patterns (all black or all white) each appear twice. This allows us to compute the exact log likelihood,

$$\begin{aligned} \ell &= (T-4)\ln(1/T) + 4\ln(2/T) \\ &= (4 - 2^{n+1}(n+1))\ln(2), \end{aligned} \tag{3}$$

which serves as an upper bound against which to compare any trained model. In the 4×4 case we primarily considered, that bound is ~ -108.

Our experiments were conducted on two D-Wave 2000Q quantum annealers (housed at NASA Ames and LANL), using the D-Wave-supplied post-processing and spin reversal transform features. In post-processing, raw samples are refined by a mild blocked Gibbs sampler, while spin reversal transforms apply gauge transformations to the problem instance in order to mitigate the effect of certain hardware biases [9]. Under certain situations, sample post-processing is expected to bring the raw sample distribution closer to Boltzmann at the user-specified post-processing inverse temperature β_{pp} [9,36]. To achieve reasonable results, the post-processing temperature β_{pp} must be chosen close to β_{eff} so that raw samples are quickly shuffled to β_{pp} by the post-processing [9,36]. We found by trial and error that $\beta_{\mathrm{pp}} = 3.0$ was ideal on both machines. 16 spin reversal transformations (equal to the number of visible nodes in the data set we considered) were used (we did not perform exhaustive experiments to determine an optimal number of spin reversal transforms).

We measure D-Wave sampling performance against simulated annealing (SA)-based sampling with comparable budgets. For both D-Wave and SA, 500 samples were obtained to compute the clamped and model averages. An annealing time of 1000 μs was used. For SA, a geometric β schedule with 2000 steps from 0.1 to 1.0 was used to decrease the temperature.

6.1 Software

To allow rapid experimentation and extensions of the work described here, we have written a few small extendible Python modules that implement the training of general Boltzmann machines via various sampling algorithms (classical and quantum), and allow one to monitor training progress (e.g., log likelihood loss, weight histograms). A user guide accompanying this report details their functionality and gives examples of the training runs below.

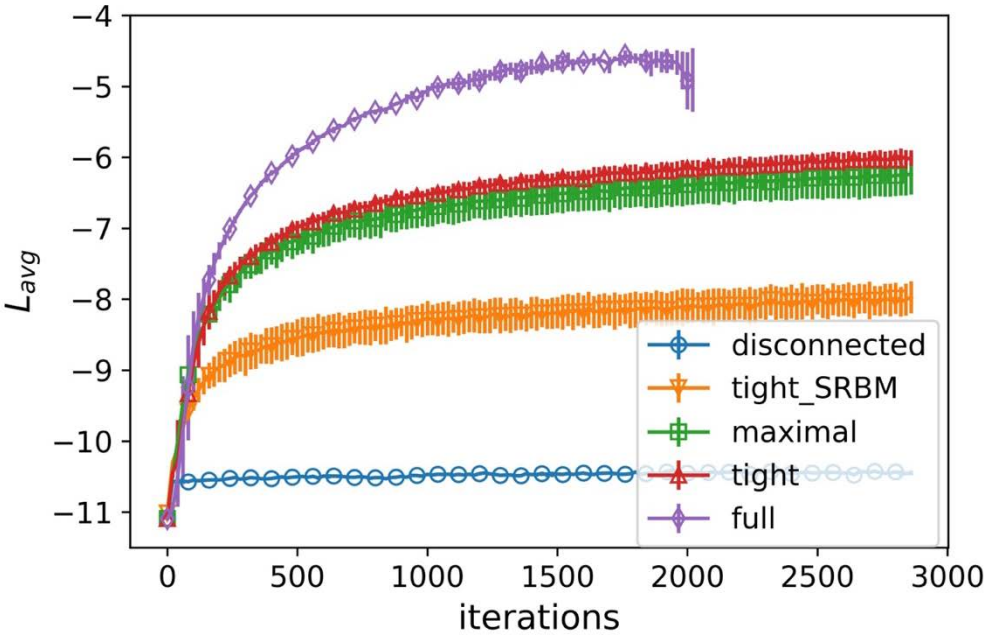


Figure 2: Evolution of log-likelihood with iterations during the training phase using QA hardware as a sampler

In Figure 2, we present average log-likelihood curves for each connectivity case. All were run at a learning rate of 0.08, except the full RBM where a learning rate of 0.02 was used. Momentum was set to 0.5 and weight decay to 0. In [34], an average log-likelihood of ~ -5.5 was reported for the tight RBM case when the parameters of the model were scaled by β_{eff}. [34] used a pre-processing scheme that maps pixels to the qubits on the hardware in a 'convolutional' fashion. Encouragingly, without any pre-processing or β_{eff} scaling, we see comparable performance. We see that the full-RBM case plateaus at a log-likelihood value of ~ -4.5 (note that for this BAS model, the average log-likelihood upper bound is ~ -3.38 [41]). State-of-the-art classical algorithms report plateau values close to the upper bound [56]. For the full-RBM case we saw onset of divergence after around 2000 iterations. The divergence observed in the log-likelihood plots is a known issue [56]. Early stopping and the use of decay term are standard procedures suggested to circumvent this issue [56]. Here we use early stopping. The wider uncertainty bars towards the end of training in the full case is indicative of onset of such divergence.

It is evident in the plot that the tight and maximal cases hit similar log-likelihood plateau values. Note that the maximal case has more hidden nodes compared to the tight case, though as mentioned above, these act as effective biases. From the plot it is clear that the disconnected case is the worst performing. Despite a small disparity between the density of edges we see a huge gap in the curves between disconnected and tight/maximal cases.

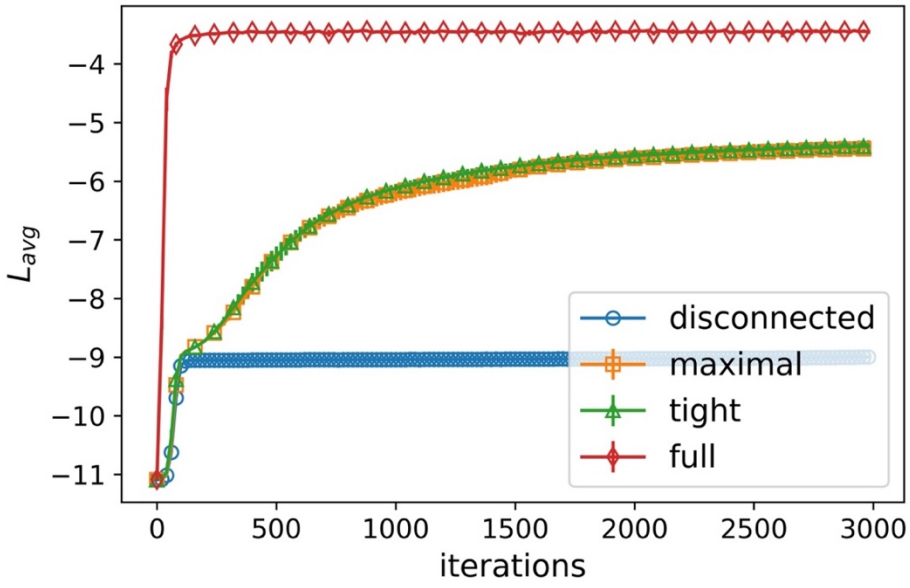


Figure 3: Evolution of log-likelihood with iterations during the training phase using SA as a sampler

In Figure 3, we show log-likelihood curves obtained using SA as a sampler. The tight and maximal cases hit plateau values similar to what was observed for QA hardware. Note that the tight and maximal cases use graphs that directly embed on the device. It is interesting to see that for such cases there is not much difference in the sampling performance between SA and QA hardware. SA-based log-likelihood values are higher compared to the QA hardware for the disconnected case. The absence of edges between the qubit blocks appears to have a severe impact on the log-likelihood values on the hardware compared to SA. As indicated in Figure 3, the SA-

based full case performs much better than the QA based runs. SA-based runs reach log-likelihood values close to the upper bound. It is possible that this QA performance hit is brought out by the use of chains to embed the full case graph on to the hardware.

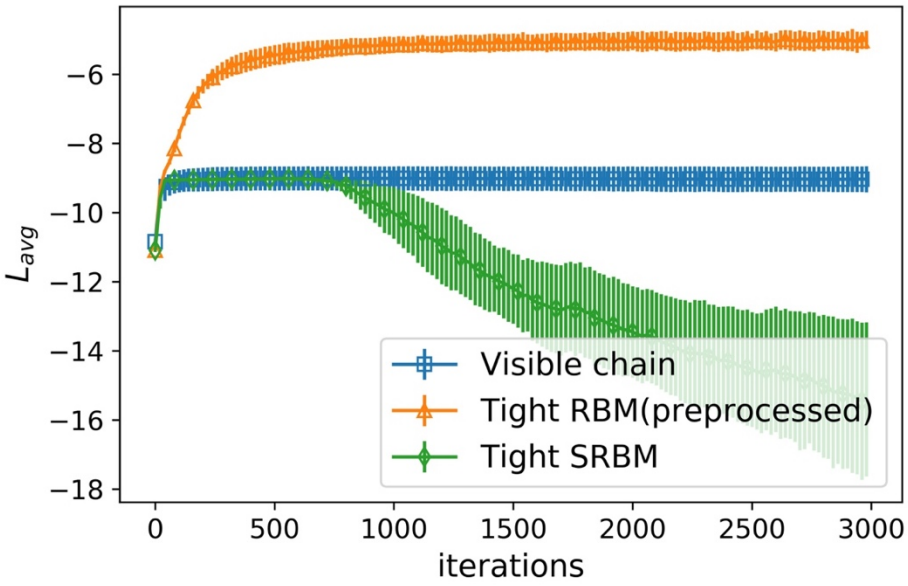


Figure 4: Evolution of log-likelihood with iterations during the training phase using SA as a sampler

In Figure 4, we present results for some additional models we studied using SA as a sampler. The tight RBM (preprocessed) plot indicates log-likelihood values obtained for the tight RBM case where the image pixels were mapped to qubits on the hardware as in Ref [34]. We see pre-processed case converges relatively rapidly to similar log-likelihood values compared to the tight RBM case in Figure 3 where no pre-processing was used. The tight SRBM case diverges after reaching a plateau value similar to QA-based run as seen in Figure 2. The "visible chain" instance uses an embedding of the visible units as a connected chain generated by a (self-avoiding) random walk on the D-Wave chip, while the hidden units are given by all neighbors of this chain (in this case disallowing hidden-hidden connections). This architecture generally suffers from sparse connectivity, but initial results suggest that allowing hidden-hidden structure can drastically improve performance.

Finally, this study opens up several interesting research avenues. We observed slight performance gains with some very light data preprocessing that tailors input to the underlying hardware graph. As described in our previous reports, we have carried out some preliminary investigations in this direction, and we believe this to be a fruitful direction.

Current hardware offers limited size and connectivity. Domain decomposition (described in section 2.2.3) allows sampling from larger graphs via partitioning. The framework we have developed can in principle be combined with the domain decomposition technique to train larger Boltzmann machines. Moreover, our software implementation can be easily extended to D-Wave's upcoming Pegasus architecture with minimal retooling, and simulations based on this new architecture can start effectively immediately. Extending this work to the training of deep Boltzmann machines is another exciting path to pursue. Such extensions are promising because of

the minimal overhead involved in training deep architectures when clamped and model sampling are carried out on the hardware. We have not yet conducted thorough tests of the benefits of on-chip clamped sampling, and indeed the training updates to purely visible model parameters do obviously support such a computation method, making these natural follow-up questions to be addressed.

Initial experiments with fully-connected non-restricted architectures (with a reduced number of hidden units) suggest that they outperform fully-connected bipartite RBMs, despite a naïve comparative lack of resources (here in the form of edges). It is an interesting (also purely classical) question whether this trend persists across datasets and problem sizes, and it is a relatively simple extension of the work described here.

We would be remiss if we did not put the general technique of Boltzmann machines in context, as they have been largely superceded by more modern techniques, such as variational autoencoders and generative adversarial networks. However, if quantum sampling remains robust as problem sizes grow and architectures complexify, the low resource requirements may give Boltzmann machines a competitive edge against these newcomers. A metastudy of such accuracy and resource requirements, paired with projections of expected annealer maturation may be in order.

On the purely operational side of this effort, we experienced frequent D-Wave connection losses, and unfortunately could not avoid manual restarts in most cases due to security features in place. A more automated solution to mitigate such issues would be a major time saver, but may not be possible given the 2-factor authentication needs of organizations hosting the hardware. Some disconnects were however not due to broken SSH pipes, so a (partial) software solution should be easily implementable.

7 REFERENCES

[1] Aji, Srinivas, and Robert J McEliece, *IEEE Trans. On Inf. Theory* **46** (2): 325-343.

[2] Bian, Zhengbin, Fabian Chudak, Robert Isreal, Brad Lackey, William G. Macready, and Aiden Roy. *Frontiers in Physics* **2**: 56, (2014)

[3] Bian, Zhengbin, Fabian Chudak, Robert Isreal, Brad Lackey, William G. Macready, and Aiden Roy, *Frontiers in ICT* **3**: 14 (2016).

[4] Brglez, F. and Fujiwara, *IEEE Int. Symp. On Circuits and Systems* (1985).

[5] Pearl, Judea, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan-Kauffmann (2014).

[6] Perdomo-Ortiz, Alejandro, Alexander Feldman, Asier Ozaeta, Sergei V Isakov, Zheng Zhu, Bryan O'Gorman, Helmut G Katzgraber, Alexander Diedrich, Hartmut Neven, Johan de Kleer, Brad Lackey, and Rupak Biswas., arXiv preprint arXiv:1708.09780 (2017).

[7] Yedidia, Johnathan S., William T. Freeman, and Yair Weiss, *NIPS* **13** (2000).

[8] Yedidia, Johnathan S., William T. Freeman, and Yair Weiss, *IEEE Trans. On Inf. Theory* **51** (7): 2282-2312 (2005).

[9] "Technical description of the d-wave qpu," (), https://docs.dwavesys.com/docs/latest/doc_qpu.html.

[10] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, Science 345, 420 (2014).

[11] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, and others, Nature 473, 194 (2011).

[12] J. King, S. Yarkoni, M. M. Nevisi, J. P. Hilton, and C. C. McGeoch, arXiv Preprint arXiv:1508.05087 (2015).

[13] G. E. Hinton and T. J. Sejnowski, in edited by D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group (MIT Press, Cambridge, MA, USA, 1986), pp. 282–317.

[14] T. Tieleman, in Proceedings of the 25th International Conference on Machine Learning (ACM, New York, NY, USA, 2008), pp. 1064–1071.

[15] P. Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory (Colorado Univ at Boulder Dept of Computer Science, 1986).

[16] A. Fischer and C. Igel, in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, edited by L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), pp. 14–36.

[17] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, Cognitive Science 9, 147 (1985).

[18] E. Aarts and J. Korst, (1988).

[19] P. M. Long and R. A. Servedio, (2010).

[20] M. H. Amin, Physical Review A 92, 052323 (2015).

[21] V. N. Smelyanskiy, D. Venturelli, A. Perdomo-Ortiz, S. Knysh, and M. I. Dykman, Physical Review Letters 118, 066802 (2017).

[22] L. C. Venuti, T. Albash, D. A. Lidar, and P. Zanardi, Physical Review A 93, 032118 (2016).

[23] Z. Bian, F. Chudak, W. G. Macready, and G. Rose, D-Wave Systems 2, (2010).

[24] M. Denil and N. de Freitas, in (NIPS, 2011).

[25] V. Dumoulin, I. Goodfellow, A. Courville, and Y. Bengio, (2014).

[26] S. H. Adachi and M. P. Henderson, arXiv Preprint arXiv:1510.06356 (2015).

[27] J. E. Dorband, in 2015 12th International Conference on Information Technology-New Generations (IEEE, 2015), pp. 703–707.

[28] D. Korenkevych, Y. Xue, Z. Bian, F. Chudak, W. G. Macready, J. Rolfe, and E. Andriyash, arXiv Preprint arXiv:1611.04528 (2016).

[29] J. Liu, F. Spedalieri, K.-T. Yao, T. Potok, C. Schuman, S. Young, R. Patton, G. Rose, and G. Chamka, Entropy 20, 380 (2018).

[30] W. Vinci and D. A. Lidar, arXiv Preprint arXiv:1710.07871 (2017).

[31] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Physical Review X (2018).

[32] M. Kieferová and N. Wiebe, Physical Review A (2017).

[33] B. Gardas, M. M. Rams, and J. Dziarmaga, Physical Review B 98, 184304 (2018).

[34] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Phys. Rev. A 94, 22308 (2016).

[35] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Physical Review X (2017).

[36] J. Raymond, S. Yarkoni, and E. Andriyash, (2016).

[37] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, Nature Communications 4, 2067 (2013).

[38] T. Tieleman, in Proceedings of the 25th International Conference on Machine Learning (ACM, 2008), pp. 1064–1071.

[39] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Physical Review A 94, 022308 (2016).

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in Advances in Neural Information Processing Systems (2012), pp. 1097–1105.

[41] (n.d.).

[42] A. D. King and C. C. McGeoch, arXiv Preprint arXiv:1410.2628 (2014).

[43] A. D. King and C. C. McGeoch, CoRR abs/1410.2628, (2014).

[44] N. Chancellor, S. Szoke, W. Vinci, G. Aeppli, and P. A. Warburton, Scientific Reports 6, (2016).

[45] J. Marshall, E. G. Rieffel, and I. Hen, arXiv Preprint arXiv:1703.03902 (2017).

[46] T. Albash, V. Martin-Mayor, and I. Hen, arXiv Preprint arXiv:1703.03871 (2017).

[47] M. Benedetti, J. Realpe-Gómez, and A. Perdomo-Ortiz, Quantum Science and Technology 3, 034007 (2018).

[48] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Physical Review X 7, 041052 (2017).

[49] M. A. Carreira-Perpinan and G. E. Hinton, in Aistats (Citeseer, 2005), pp. 33–40.

[50] J. Melchior, A. Fischer, and L. Wiskott, (2013).

[51] B. Lackey, arXiv Preprint arXiv:1810.10005 (2018).

[52] Hinton, G. E., Osindero, S., & Teh, Y. W., Neural computation, 18(7), 1527-1554 (2006).

[53] Tzortzis, G. and Likas, A., In 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007) (Vol. 2, pp. 306-309). IEEE (2007).

[54] Fiore, U., Palmieri, F., Castiglione, A. and De Santis, A., Neurocomputing, 122, pp.13-23 (2013).

[55] McCann, M.T., Jin, K.H. and Unser, M., IEEE Signal Processing Magazine, 34(6), pp.85-95 (2017).

[56] Fischer, A. and Igel, C.. In International Conference on Artificial Neural Networks (pp. 208-217). Springer, Berlin, Heidelberg (2010).

8 LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

ACRONYM	DEFINITION
BAS	Bars and Stripes
COTR	Contracting Officer Technical Representative
CDRL	Contract Data Requirements List
FTR	Final Technical Report
QA	Quantum Annealing
RBM	Restricted Boltzmann Machine
SA	Simulated Annealing
SAT	The Boolean Satisfiability problem
SRBM	Semi-Restricted Boltzmann Machine