Restricted Boltzmann Machines for galaxy morphology classification with a quantum annealer

João Caldeira,[1, *] Joshua Job,[2, *] Steven H. Adachi,[2] Brian Nord,[1, 3, 4] and Gabriel N. Perdue[1]

[1]*Fermi National Accelerator Laboratory, Batavia, IL 60510*
[2]*Lockheed Martin Advanced Technology Center, Sunnyvale, CA 94089*
[3]*Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637*
[4]*Department of Astronomy and Astrophysics, University of Chicago, Chicago, IL 60637*
(Dated: November 15, 2019)

We present the application of Restricted Boltzmann Machines (RBMs) to the task of astronomical image classification using a quantum annealer built by D-Wave Systems. Morphological analysis of galaxies provides critical information for studying their formation and evolution across cosmic time scales. We compress the images using principal component analysis to fit a representation on the quantum hardware. Then, we train RBMs with discriminative and generative algorithms, including contrastive divergence and hybrid generative-discriminative approaches. We compare these methods to Quantum Annealing (QA), Markov Chain Monte Carlo (MCMC) Gibbs Sampling, Simulated Annealing (SA) as well as machine learning algorithms like gradient boosted decision trees. We find that RBMs implemented on D-wave hardware perform well, and that they show some classification performance advantages on small datasets, but they don't offer a broadly strategic advantage for this task. During this exploration, we analyzed the steps required for Boltzmann sampling with the D-Wave 2000Q, including a study of temperature estimation, and examined the impact of qubit noise by comparing and contrasting the original D-Wave 2000Q to the lower-noise version recently made available. While these analyses ultimately had minimal impact on the performance of the RBMs, we include them for reference.

I. INTRODUCTION

Machine learning techniques are being used increasingly in high energy physics [e.g., 1] and astrophysics [e.g., 2] for applications such as event detection, particle identification, data analysis, and simulation of detector responses. In many cases, machine learning provides an efficient alternative to analytical models, which are intractable, or Monte Carlo-based simulations, which can be computationally expensive. Data analysis tasks in cosmology are extremely computing intensive and will become even more so as new instruments like LSST [3] come online, motivating advances in data analysis techniques.

Feynman first proposed the idea of using quantum computers to simulate physical systems [4]. More recently, a variety of approaches have been studied for combining quantum computing with machine learning techniques [5]. While quantum computing hardware is still in the early stages of development, initial attempts have been made to apply quantum machine learning to high energy physics, e.g. classifying Higgs decay events in Large Hadron Collider data [6].

In this study, we focus on the challenge of morphological classification of galaxies via astronomical images using the D-Wave 2000Q quantum annealer [7]. Galaxies exhibit morphologies (structure in their shape) that tend to correlate with their evolutionary state and history. For example, spiral galaxies (typically blue) have higher rates of star formation often visible in clumpy regions of their spiral arms, irregulars have quasi-randomly distributed clumps of star formation, while elliptical galaxies (typically red) tend to have ceased making their stars. Star formation occurs more readily in relaxed kinematic environments, where gravity has sufficient relative influence to pull together cold material that can fuse into stellar cores. Highly energetic or dense environments, like the cores of galaxy clusters or filaments of the cosmic web, may cause galaxy mergers or other disruptive events that can slow or halt star formation. Galaxies evolve rapidly in stellar mass in the range of cosmic redshifts (measures of cosmic age) $1 < z < 3$, where star formation density peaks near $z \sim 2.5$. The rate of star formation is one of the primary measures of cosmic energy exchange, and structural and morphological analysis of galaxies permits a critical avenue of investigation cosmic evolution. The accurate classification of galaxies based on morphology is a critical step in this analysis. Please see the review in [8] for more details.

Classical methods for morphological analysis have typically relied on a) visual examination, such as conducted through the Galaxy Zoo project [9]; multi-wavelength model-fitting [10]; and structural proxies, like concentration, asymmetry, and clumpiness [8]. Recent advancements in deep learning permit the usage of convolutional neural networks for morphological classification, which

---
* J. Caldeira and J. Job contributed equally to this work; caldeira@fnal.gov; joshua.job@lmco.com

have become the state of art [11–13]. The conventional convolutional neural network does not yet have an efficient implementation on the D-Wave quantum annealer. In this work, we use a different type of machine learning model, the Restricted Boltzmann Machine (RBM) [14]. While there are many other types of machine learning models, the RBM model has stochastic binary variables and a quadratic energy functional, which can be efficiently implemented using the relatively small number of qubits available on near-term quantum computing devices, such as the D-Wave quantum annealer [15, 16]. Training an RBM is classically hard, but there is reason to believe quantum annealers may eventually offer performance advantages [17].

While quantum annealers are generally used for solving optimization problems, they have also been used in a machine learning context, where the quantum annealer is programmed with coefficients derived from the RBM, and used as a sampling engine to generate samples from the Boltzmann distribution [17, 18]. As in classical machine learning, an iterative training process is used to refine the RBM coefficients.

Quantum annealers offer a way to leverage the power of quantum computers while avoiding the complexity of a gate-based programming model, making them an attractive tool for domain scientists. However, given the limitations of present-day quantum annealers, this approach presents a number of challenges. Input data must be severely compressed to fit the available qubits. Samples from the quantum annealer may be not be Boltzmann distributed, in which case post-processing or temperature estimation techniques may need to be applied.

A. Overview of the paper

This paper consists of two main results. First we show the outcome of some studies of the distribution of states coming from the D-Wave. We considered a variety of post-processing techniques to bring the output distributions closer to Boltzmann distributions, which are theoretically necessary for training RBMs. Second, we show the results of trained RBMs and other algorithms for the galaxy morphology classification problem. We also include a discussion of the data and compression methods employed.

Specifically, in Section II we briefly review the galaxy morphology classification datasets used for training and testing, and the techniques used to compress the data for the D-Wave 2000Q. In Section III we discuss the training algorithms used to prepare RBMs for classification. We compare a variety of options for training RBMs, including various combinations of generative and discriminative training. In Section IV, we discuss a variety of post-processing steps for producing a Boltzmann-distributed set of energy states using the D-Wave quantum annealer. We also compare two versions of the D-Wave 2000Q, one of which featured lower levels of noise. In Section V we

study the performance of RBMs on the quantum device and using classical resources, and we compare the performance of other classical machine learning algorithms. Finally, in Section VI we conclude and offer thoughts on future directions.

II. ASTRONOMICAL DATA AND COMPRESSION

A. Data: Galaxy Zoo

We use data from the Galaxy Zoo 2 data release, which contains 304,122 galaxies that are taken from the Sloan Digital Sky Survey [9]. For each image of a galaxy, this dataset includes crowdsourced answers to a set of 11 questions characterizing the galaxy's morphology. There are 16 million classifications of morphology, with features like bulges, disks, bars, spirals. We simplify the problem into a binary classification problem by picking spiral galaxies (those with more than 50% "yes" answers to "Is there a spiral pattern?") and rounded smooth galaxies (those with more than 50% "completely round" answers to the question "How rounded is it?"). These classes contain 10397 and 8434 galaxies, respectively. We select 5000 random images of each of the two classes. Before applying any data compression algorithm, we crop the images into 200 by 200 pixel images.

B. Compression and Manipulation

Raw images are 200×200 RGB pixels in size, which is far too large to encode in the binary variables available on the D-Wave 2000Q. There are a number of interesting compression schemes available, including, for example, discrete variational autoencoders [19, 20]. In practice, we found no appreciable advantage with different compression schemes while reducing data dimensionality to the level where we could encode the essential information about a given image into the binary variables available. Therefore, we relied on principal component analysis (PCA) on the basis that the method is very simple and easy to explain and understand.

We used 5000 images to train a PCA model using Scikit-learn [21], and applied it on the remaining 5000 images to obtain the dataset we used to train and test the RBM. The ratio of explained variance added by each PCA component is shown in Fig. 1. We can see that the information contained in each additional component rapidly decays.

The encoded components defined by PCA are 64-bit floating point numbers. We would like to transform them into a more compact representation. We will do this by linearly mapping the range of each PCA component in the training set to the interval [15, 240]. We can then round to the nearest integer and transform the data into unsigned 8-bit integers, which support numbers between
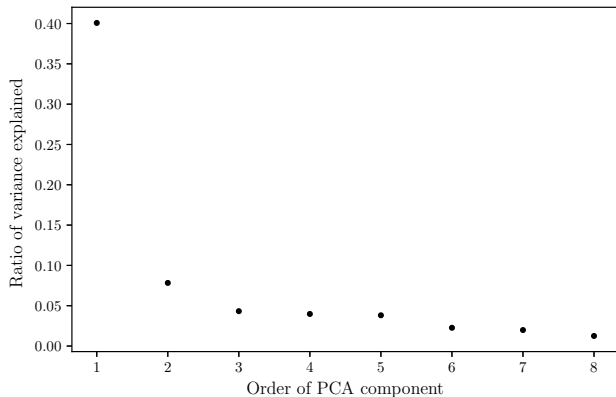
FIG. 1. The ratio of the total variance in the PCA training dataset explained by each additional PCA component.

0 and 255. The range we map the training set into was chosen to be safely inside the [0, 255] interval in order to accomodate outliers beyond the ranges present in the training set. See Fig. 2 for an example of the compressed data.

To these compact representations of the images, we add a bit representing the class (0 for rounded smooth galaxies and 1 for spiral galaxies). This means that if our RBM has n visible units, the first $n - 1$ will correspond to the first bits in the compressed images, while the last visible unit will encode the class of each image.

During the analysis we were concerned that the digitization scheme employed was putting extra weight on the most significant bits of each encoded PCA component, but this was not information we could easily share directly with the RBM algorithm. We tested several different re-ordering schemes for the bits and also tested preferentially keeping the most significant bits only for higher order components as a way of including information from those components when working with small RBMs. However, the re-ordering schemes generally slightly degraded performance, and attempts to include a larger number of components using only the most significant bits did not offer any performance advantages.

## III. GENERATIVE AND DISCRIMINATIVE TRAINING

A. RBM training

Restricted Boltzmann machines (RBMs) are generally speaking a generative model, where one attempts to approximate one's target distribution over a string of binary variables $\vec{v}$, $p(\vec{v})$, as the marginal distribution of a larger system $p(v) = \sum_{\vec{h}} p(\vec{v}, \vec{h})$ composed of binary variables $\vec{v}$ and latent variables $\vec{h}$, with an ansatz such that

$$p(v, h) \propto exp(-v'Wh + b'v + c'h) \qquad (1)$$
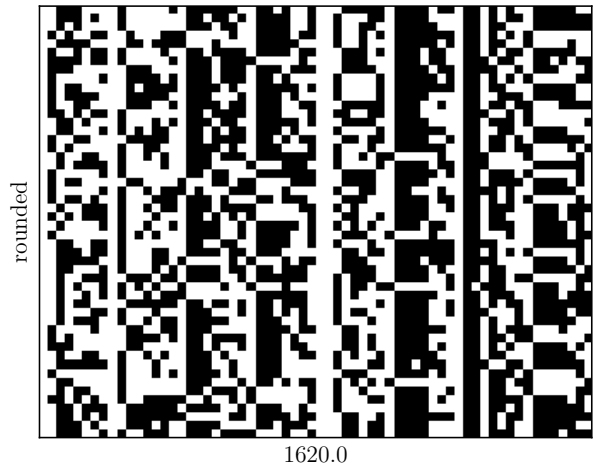
(a) Galaxies with label 0, corresponding to rounded smooth galaxies.
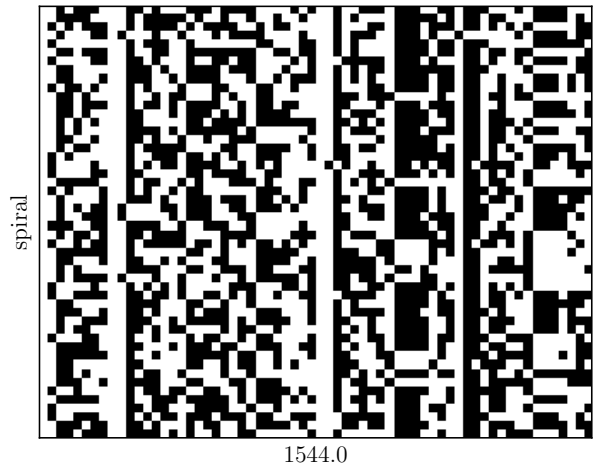
(b) Galaxies with label 1, corresponding to spiral galaxies.

FIG. 2. Compressed "minibatches" of data. Here each row in each figure represents one event. There are fifty events (rows) in each figure. Each column represents the binary value of the compressed data bit, with dark colors indicating zero and bright colors indicating one. Here we have 64 bits of width, with each PCA component represented by an 8-bit discretization of the floating point value, for a total of eight PCA components represented, with the most important component on the left side of the figure. The x-axis labels report the sum of all the binary values across the entire minibatch (so the maximum possible is 3,200).

for some bias vectors b, c, and a weight matrix W. This corresponds to a complete bipartite graph with local biases and interactions along the edges. The binary variable $\vec{v}$ are called the visible nodes, as they compose the distribution of interest, while the $\vec{h}$ are the hidden nodes.

By variationally maximizing the log-likelihood of the data of the RBM model with respect to the weights and biases, we can train the RBM to better approximate the data distribution. It can be readily shown that max-

imizing the log-likelihood corresponds to matching the one and two-point correlation functions of the model between states conditioned on the data distribution and the free generative model. Defining the loss as the negative log-likelihood, dubbed L, we get derivatives for the variational parameters

$$\frac{\partial L}{\partial b^i} = \left\langle v^i \right\rangle_{\text{data}} - \left\langle v^i \right\rangle_{\text{model}} \tag{2}$$

$$\frac{\partial L}{\partial c^i} = \left\langle h^i \right\rangle_{\text{data}} - \left\langle h^i \right\rangle_{\text{model}} \tag{3}$$

$$\frac{\partial L}{\partial W_i^j} = \left\langle v^i h^j \right\rangle_{\text{data}} - \left\langle v^i h^j \right\rangle_{\text{model}} \tag{4}$$

Collectively these derivatives form the gradient, to be used in gradient descent to the adjustments for b, c, and W respectively. Here the expectations are computed over the training set and the the model (also called the positive and negative phases).

Once the RBM has been trained, we can use it to make a prediction on the class of unseen images. To do this, we calculate the free energies of the RBM with the visible units set to the compressed image representation and both options for the class. The class corresponding to the lowest free energy is then the most likely class for that image. This type of discriminative RBM was introduced in [22].

B. Classical training algorithms

In general, one cannot compute expectations over the model directly, as it takes a time that scales as $2^{\min n_v, n_h}$ where n_v and n_h represent the number of visible and hidden units, respectively. This is generally intractable. However, we can use a variety of algorithms to perform training.

1. Contrastive divergence

We may perform efficiently block sampling updates of $p(v|h)$ and $p(h|v)$, as the conditional distributions reduce to single-spin probabilities that can be sampled in linear time with the number of variables. Initializing a Markov chain performing such block Gibbs sampling at each training datapoint and taking expectations over the resulting chains is the basis of the contrastive divergence (CD) algorithm, first put forward in [23]. Using CD, one can often train RBMs of quite large size reasonably efficiently.

2. Discriminative training

In this work, we are interested in using RBMs not as a strictly generative model, but as a classification algorithm. In essence, we wish to be able to input an image and sampling the posterior distribution for the class

of that image using the RBM. Thus, rather than our directly modeling the full $p(v)$, as is standard practice, we are really interested in only $p(v_{class}|v_{image})$. Rather than training a model to represent the entire distribution over v, we can instead directly train to maximize the log-likelihood of the $p(v_{class}|v_{image})$ distribution, as was proposed in [22].

The training process is much the same as before, except that now the sample over the model is vastly simplified, as one is still taking an expectation by fixing the image dataset, reducing the effective number of variables to merely that of the number of bits used to represent the class. In our case, where we use a single variable, we thus can contract the graph in linear time to get an exact gradient. In general, one can contract in a time scaling no worse than $2^{n_{class}}$ for unary encoding of the classes. Using a binary representation for the classes, one can do this in linear time in the number of classes, and thus training the discriminative model can be done efficiently on a classical computer in an exact fashion, with no Markov chains required.

3. Hybrid approaches

Finally, one may consider a hybrid approach. For instance, an approach where one takes a combination of both the aforementioned gradients, generative and discriminative, so as to better approximate $p(v_{class}|v_{image})$ while still representing the full distribution efficiently. In this, we can set a value λ which combines the the gradients $\boldsymbol{\nabla}_{gen}$ and $\boldsymbol{\nabla}_{disc}$ for the generative and discriminative models as

$$\boldsymbol{\nabla}_{hybrid} = \frac{\lambda}{1+\lambda} \boldsymbol{\nabla}_{gen} + \frac{1}{1+\lambda} \boldsymbol{\nabla}_{disc}. \tag{5}$$

This approach was also investigated in [22] and found to be beneficial at small values of λ.

We additionally investigate another hybrid approach, where we use generative training as a kind of pretraining and then follow it with pure discriminative training, which we dub "annealed hybrid" training, even though if one is considering it as annealing the λ parameter it is better thought of as a quench. This was motivated by our observations of the performance of generative and discriminative training.

C. Generative training with quantum annealing

We also compare against a quantum annealing (QA) based model for estimating the negative phase (the intractable model expectation values) and alternatives to QA, including a pure Gibbs sampling MCMC algorithm initialized at a random position, and simulated annealing. In essence we seek to understand what causes observed QA performance by testing against other annealing algorithms.

In training via quantum annealing, we map our RBM energy function, which is in the form of a QUBO (quadratic unconstrained binary optimization), and use D-Wave's provided embedding function to map this QUBO into the physical architecture of the D-Wave device, called a Chimera graph, see Fig. 3. This is done because the Chimera graph has a maximum complete bipartite subgraph of 4x4.

By minor embedding [24] the graph we identify a chain of qubits and bind them tightly together so that they act approximately as a single large spin. Each programming cycle we use 100 samples drawn from the D-Wave to take our gradient estimate, and apply a varying number (typically 2) post-anneal Gibbs sweeps over the variables to aid in additional thermalization.

IV. BOLTZMANN DISTRIBUTIONS ON THE D-WAVE QUANTUM ANNEALER

In order to train a Boltzmann machine, we need to sample expectation values from a Boltzmann distribution with β set to 1. We use the Kolmogorov-Smirnov (KS) test to check the statistical consistency of our sample distribution with that of a Boltzmann distribution.

A. Comparisons between initializing sampling with an annealer vs a random bitstring

The raw distribution of states coming from a D-Wave 2000Q is often not close to a Boltzmann distribution with $\beta = 1$. It is "colder", with a higher propensity for producing states at the lowest energy levels. This energy shift may be advantageous in optimization problems, but RBM training relies on being able to sample from a Boltzmann distribution, so post-processing is generally required.

For us, this process will consist of taking a few steps of Gibbs sampling as a post-processing step. In this section, to check how many steps is enough, we carry out the KS test after each step and keep taking Gibbs steps until the KS p-value rises above 0.05.

The advantage in starting the post-processing using samples from a D-Wave 2000Q is not clear in some of the RBM shown in Fig. 4, for instance in Fig. 4d. On the other hand, Fig. 4b and Fig. 4c show some advantages. In all cases, however, there are regions of couplings for which we need quite a few steps, as shown in Fig. 4.

B. Temperature estimation

It is possible that the D-Wave returns a Boltzmann distribution, but at a temperature that needs to be determined. If we know the effective inverse temperature β_{eff}, we can sample from a distribution with $\beta = 1$ and couplings (W, b, c) by setting the couplings $J =$ $(W/\beta_{\mathrm{eff}}, b/\beta_{\mathrm{eff}}, c/\beta_{\mathrm{eff}})$ on the D-Wave. The effective temperature of the D-Wave has been shown to be problem-dependent and different from the physical temperature of the annealer [25]. In this work, we will follow a modification of the temperature estimation recipe proposed in [18].

The algorithm follows the following steps:

1. At each step, take RBM couplings $A = (W, b, c)$. Set couplings on D-Wave to $J_1 = A/\beta_0$, with β_0 estimated at the previous step (on the first step, we need to take a guess).

2. Take one set of n samples. We bin the samples into $\lceil \sqrt{2n} \rceil$ bins according to their energy, obtaining probability density estimates n_1/n.

3. We want a second set that will provide different "enough" samples for distinguishability. Following [18], we take $J_2 = xJ_1$, with $x = 1 + 1/(\beta_0\sigma)$,[1] where σ is the standard deviation of the first sample.

4. Take a second set of samples and use the same bins as in step 2 to obtain probability density estimates n_2/n.

5. Denoting the Ising energy of each state with couplings A as E, note that

$$\frac{n_2}{n_1} = \frac{e^{-x\beta_{\mathrm{eff}}E/\beta_0}}{Z_2}\frac{Z_1}{e^{-\beta_{\mathrm{eff}}E/\beta_0}}$$
$$\Rightarrow \log\frac{n_2}{n_1} = \log\frac{Z_1}{Z_2} + (1-x)\frac{\beta_{\mathrm{eff}}}{\beta_0}E. \qquad (6)$$

With this in mind, we can extract an estimate of β_{eff} from the slope of the linear regression between $\log n_2/n_1$ and the bin energies, as exemplified in Fig. 5. In order to reduce noise caused by bins with a small number of samples, we limit the regression to bins with at least five samples in both draws.

We can see the results of this temperature estimation procedure throughout training in Fig. 6.

We found some pitfalls in this procedure. Namely, as the couplings of the RBM and therefore the magnitude of the energies involved grow, the distribution of states becomes more and more skewed towards the lower energy states. This is a desirable outcome of training an RBM. However, this leaves the higher-energy bins with a small number of samples, causing large variance in the estimates of $\log(n_2/n_1)$. In all our training runs, this leads to a step where $\log(n_2/n_1)$ happens to fluctuate

----

[1] [18] suggests transitioning to a $-$ sign in the expression for x once the RBM couplings get large enough. We found that even at late stages, this would result in values of x that are close to zero.
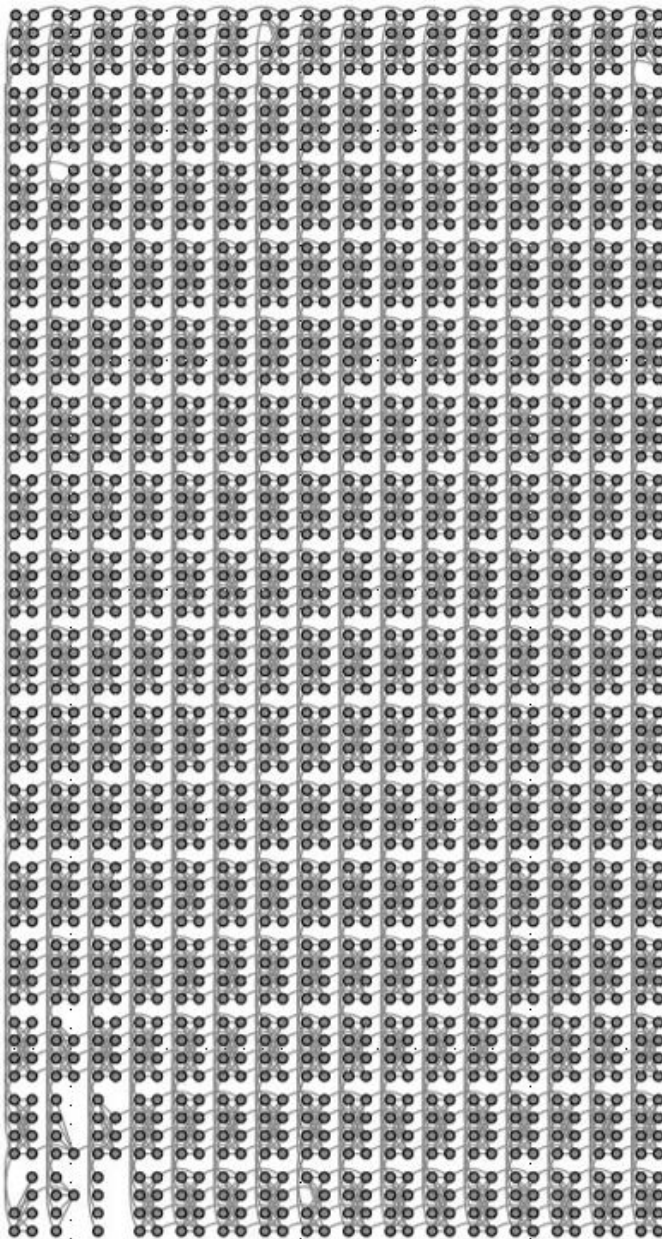
FIG. 3. An example Chimera lattice, from the low-noise DW2000Q we used. Each line is a coupler, each node an active qubit. The ideal graph is a tile of $K_{4,4}$ graphs with vertical connections between qubits in the same position in unit cells above and below from the left-hand side and horizontal connections between qubits in the same position in unit cells left and right from the right-hand side of the unit cell.

to a larger value than usual for some of the larger energy bins. This causes β_{eff} to be underestimated at that step. The effect compounds in a few training steps, often leading to negative estimates of β_{eff} and a crash of the algorithm.

Potential solutions include:

- some regularization to keep the weights from growing. This successfully kept the temperature estimation routine from crashing, but at the cost of impairing the classifier performance of our RBM. This

is to be expected, as a well-trained RBM should strongly separate the energies of different states.

- only estimating β during the initial stages of training. This can be a good solution, since β does not seem to change by a large amount during training, as we can see in Fig. 6.

Even without a temperature estimation routine, weights growing to be too large is a problem with the algorithm on a QA in general. This is because if weights grow above the maximum coupling that can be imple-
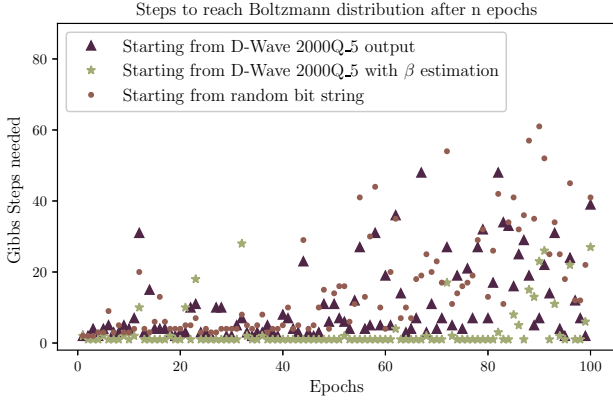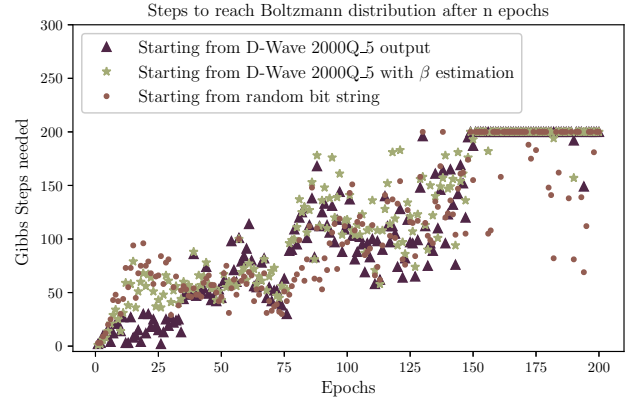
(a) Couplings are obtained by training a 12×12 RBM on the 2000Q with 10 Gibbs steps. On this test, there is some advantage to using a D-Wave, especially after estimating β.
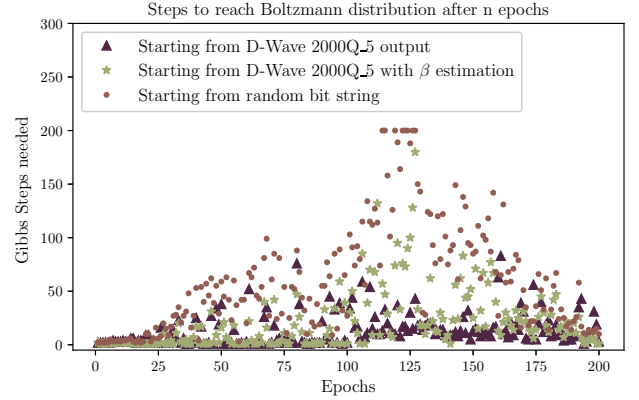
(b) Couplings are obtained by training a 12×12 RBM on the low-noise 2000Q with 10 Gibbs steps. On this test, the D-Wave needs fewer steps than starting from a random string, though applying β estimation does not help further.
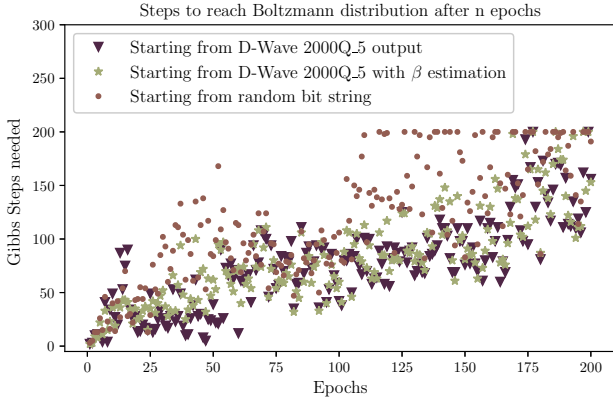
(c) Couplings are obtained by training a 48×48 RBM on the 2000Q with 10 Gibbs steps. In this case, the D-Wave shows some advantage over random strings.

(d) Couplings are obtained by training a 48×48 RBM on the low-noise 2000Q with 10 Gibbs steps. For these RBM couplings, starting from a random string or the D-Wave samples does not lead to a significantly different number of Gibbs samples needed.

FIG. 4. We present results of the test described in section IV A, applied to the D-Wave samples after setting the D-Wave couplings to the actual RBM couplings, and to the RBM couplings scaled by an estimated temperature as in section IV B. Note that the number of Gibbs steps taken until a p-value of 0.05 was reached was capped at 200, and that is why there is some bunching of values at 200 Gibbs steps.

mented on the D-Wave, we must rescale the weights as in order to set coupling constants on the D-Wave. However, discretization of the coupling constants means that if one weight is very large, subtle variations between much smaller weights are lost. Another possible solution would be to turn off weight rescaling, but not let couplings grow beyond what is physically implementable on the D-Wave. Conceptually, this is equivalent to allowing the RBM to learn chains of logical qubits that are strongly coupled. Either of these solutions can impair classifier performance because sometimes RBM might just need very large weights, or might need a large ratio across some weights, to reproduce the probability distribution of the data.

Finally, we have tested whether temperature estima-

tion allows us to take fewer Gibbs steps to reach a Boltzmann distribution. The results of this test are shown in Fig. 4. Once again, the results do not always show a decisive advantage in the number of post-processing steps needed when using temperature estimation.

C. Noise and RBMs

D-Wave has recently released a low-noise version of its 2000Q quantum computer, with claims to enhancing tunneling rates by a factor of 7.4 [26]. It is claimed that this leads to a larger diversity of states returned by the machine, as well as a larger proportion of lower-energy states. In this section, we test whether these lower-
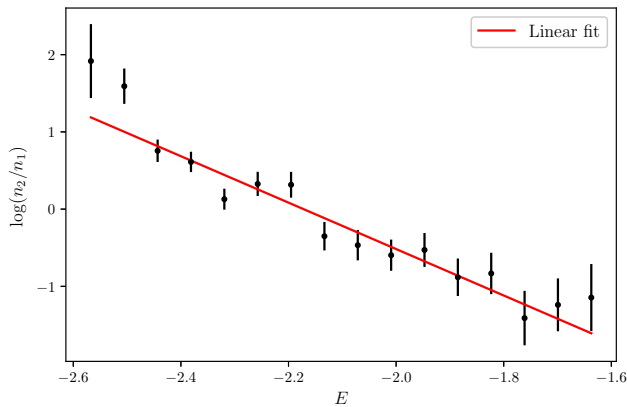
FIG. 5. Linear regression obtained from equation (6) for an example step in training a restricted Boltzmann machine.
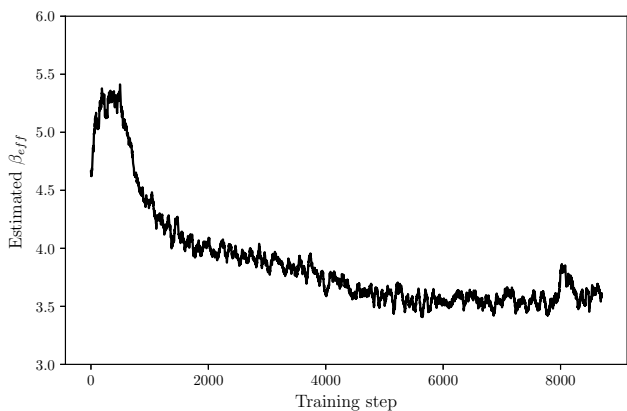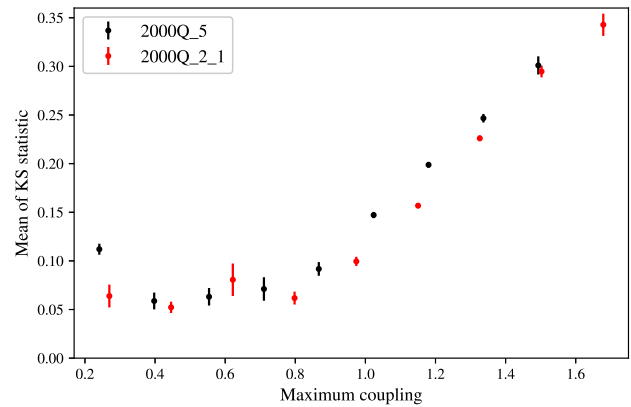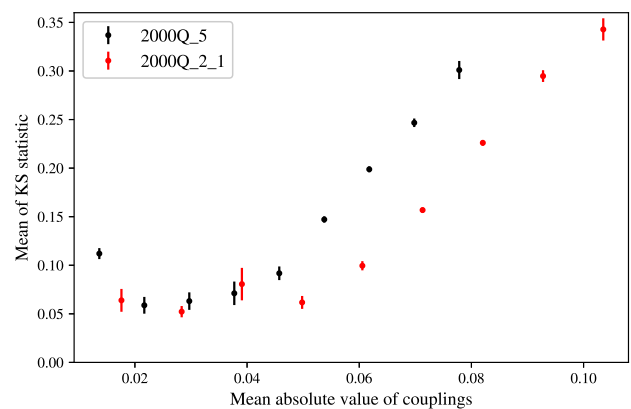

FIG. 6. Temperature estimates over 70 epochs of training (or 8750 training steps) for a 48×48 RBM, plotted using a rolling average over the last 50 steps. It can be seen that the temperature estimates vary significantly over the first stages of training, and later stabilize. This can likely be used to estimate the temperature less often than at every training step.

(a) KS statistic as a function of maximum RBM coupling after scaling by the effective β.

(b) KS statistic as a function of mean RBM coupling after scaling by the effective β.

FIG. 7. We compare the KS statistics between Boltzmann samples and D-Wave samples for two 12 by 12 RBM, one trained on the original 2000Q and one trained on the low-noise 2000Q. We see no advantage in finding a Boltzmann distribution from using the low-noise 2000Q. Note each machine was tested using couplings of an RBM trained on that same machine, so the range of tested couplings differs slightly.

noise properties also help us obtain a more Boltzmann-like distribution from the D-Wave output. To do this, we train two 12×12 RBM using the temperature estimation techniques described in Sec. IV B. One of the RBM was trained using the original 2000Q, and the other RBM using the low-noise machine. At each 20 training steps, we compare the distribution obtained using the D-Wave machine with a Boltzmann distribution obtained from analytically calculated energies for the current RBM couplings. To compare the distributions, we use the Kolmogorov-Smirnov statistic, which should be close to zero for samples drawn from the same distribution. We compare the KS values as a function of the RBM weight distribution in each machine.

In Fig. 7, we show the mean of the KS statistic binned as a function of the mean and maximum RBM coupling.

We see no advantage from using the lower-noise 2000Q in how Boltzmann-like the returned distributions are. For both machines, samples returned are not far from Boltzmann distributions (with KS statistics below 0.1) for low RBM weights, but the distributions diverge from Boltzmann as the weights grow larger.

We also try a test similar to Fig. 4, initializing the Gibbs steps with samples from either 2000Q machine. The results are shown in Fig. 8.

V. RESULTS AND DISCUSSION

Here we focus on comparing training accuracy, defined as the total fraction of all testing data correctly classified by the trained RBMs, between the wide va-
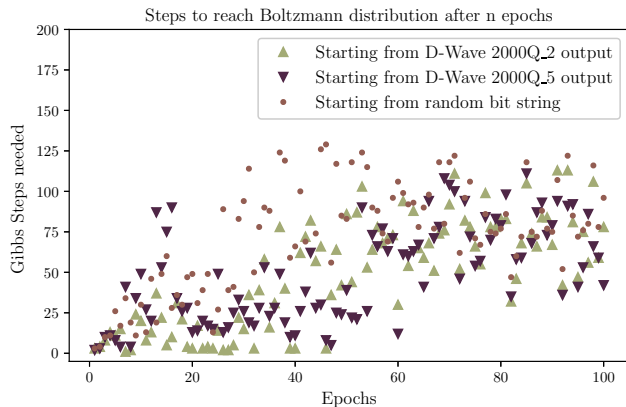
FIG. 8. Couplings are obtained by training a 48×48 RBM on the 2000Q with 10 Gibbs steps. We see no significant difference between using the 2000Q and the low-noise 2000Q.

riety of algorithms discussed in section III, along with straightforward logistic regression and gradient boosted decision trees, as a function of training epoch. For results given in this section, we use two Gibbs sweeps of post-processing for quantum annealing samples, except as otherwise stated.

We present a comprehensive figure of our primary results in Fig. 9. In Fig. 9a we see that QA requires at least two Gibbs sweeps to perform competitively. Moreover there we also see, all RBM-structured models (which are upper-bounded, as seen in Fig. 9b, by discriminative training) are outperformed for this problem by simple logistic regression and particularly gradient boosted decision trees. (Note: "epoch" for gradient boosted decision trees corresponds to the number of trees in these plots.) Looking at Fig. 9b (where we focus on RBM-structured models), we show that while QA appears to achieve higher accuracy at early stages of training than other RBM-structured models/training methods, as training progresses it either matches (for smaller size RBMs) or underperforms other algorithms such as purely discriminative and hybrid training. Moreover, it appears to lend little if anything to discriminative training to incorporate hybrid updates using QA, or to transition from QA to discriminative training near the observed crossover point of performance. Unless one is only going to run training for a short period (on the order of 25 epochs), one observes no improvement from the incorporation of QA. We also examine performance of MCMC Gibbs sampling (ie directly taking expectation values from a Markov chain of appropriate length) and simulated annealing, and observe broadly similar performance as QA, with small improvements for MCMC and SA over QA at larger size RBMs.

We also present Fig. 10 to highlight in more detail the relative performance of the various RBM-structured algorithms, wherein we take the ratio of their accuracy at each step of training with the accuracy of the quantum annealing training. As this figure makes clear, QA achieves better early training results but fails to maintain that advantage with additional training. It also makes clear that MCMC Gibbs sampling and SA with sufficiently many sweeps outperforms QA slightly at larger RBM sizes. In that figure we exclude logistic regression and gradient boosted decision trees as they dominate all models.

Inspired by the observations in [6], and by the advantages shown very early in training by the QA approach, we also performed a study using a very small training set. As shown in Fig. 11, the RBM approach actually shows a decisive advantage under this condition. When the training set is restricted to 250 events (reduced from 5000), we find strong overfitting when using logistic regression or gradient boosted trees, but good performance by the RBM. We again see stronger performance by the QA early in the training process, but as the number of epochs of training increases, discriminatively trained models take over.

Curiously, this suggests that if there is a need to train a classifier based on very small datasets and if the number of epochs is limited (perhaps by time concerns in a situation where we were able to operate training for the QA resource with no network latency), then a QA-trained RBM shows promise as a superior algorithm.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we explored a classification application of importance in cosmology, and studied the distribution of energy states coming from the D-Wave 2000Q. We tested several post-processing methods that aimed to bring the output distributions closer to Boltzmann distributions, which are theoretically required for training RBMs, but we found little impact from post-processing on RBM performance. As a consequence and for simplicity of interpretation, we subsequently minimized post-processing. We presented the results of trained RBMs, and other ML algorithms, for galaxy morphology classification. While we ultimately find RBMs implemented on D-Wave hardware perform well, we don't find compelling evidence of algorithmic performance advantage with this dataset for this problem over the most likely training scenarios.

We do not believe this result is an indictment of the performance of the quantum resources — we found regions of phase space in the training where the quantum computer performed better than its classical counterpart. In particular, for small datasets and for limited numbers of training repetitions, QA-based RBMs performed very well and outperformed the alternative classical algorithms studied here (logistic regression and gradient boosted trees), and outperformed classically trained RBMs. However, outside of these rather special training scenarios, RBMs (regardless of the classical or quantum nature of the training algorithm) did not outperform the gradient boosted tree algorithm.

Perhaps more complex and higher dimensional data would be more challenging for algorithms like gradient boosted trees and regression, but here we find that they are able to handle the dataset well. In the cases where significantly less or no compression is required due to significantly larger quantum resources, we may see a performance advantage for this algorithm. This line of investigation will be interesting to revisit on future versions of quantum hardware, or perhaps on a digital annealer [27, 28] with substantially larger RBMs. For this data, due to the compression mechanisms involved, enlarging the RBM significantly did not lead to performance improvements, but with much larger numbers of qubits available, it would be possible to pursue different compression mechanisms or perhaps avoid compression altogether. Another option to pursue with less compressed data would be increasing the network connectivity and adding additional layers. There is evidence in classical machine learning [29] that multi-layer networks are able to construct hierarchical representations that often offer some advantage in data analysis tasks.

ACKNOWLEDGMENTS

[1] K. Albertsson, P. Altoe, D. Anderson, J. Anderson, M. Andrews, J. P. Araque Espinosa, A. Aurisano, L. Basara, A. Bevan, W. Bhimji et al., Machine Learning in High Energy Physics Community White Paper, arXiv e-print (2018). arXiv:1807.02876[physics.comp-ph].
URL https://arxiv.org/abs/1807.02876

[2] M. Ntampaka, C. Avestruz, S. Boada, J. Caldeira, J. Cisewski-Kehe, R. Di Stefano, C. Dvorkin, A. E. Evrard, A. Farahi, D. Finkbeiner et al., The Role of Machine Learning in the Next Decade of Cosmology, arXiv e-print (2019). arXiv:1902.10159[astro-ph.IM].
URL https://arxiv.org/abs/1902.10159

[3] Large Synoptic Survey Telescope, https://www.lsst.org/ (2019).

[4] R. P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics 21 (6) (1982) 467–488. doi:10.1007/BF02650179.
URL https://doi.org/10.1007/BF02650179

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature 549 (2017) 195 EP –.
URL https://doi.org/10.1038/nature23474

[6] A. Mott, J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, Solving a higgs optimization problem with quantum annealing for machine learning, Nature 550 (2017) 375 EP –.
URL https://doi.org/10.1038/nature24047

[7] D-Wave Systems, The D-Wave 2000Q^{TM} system, https://www.dwavesys.com/d-wave-two-system (2019).

[8] C. J. Conselice, The evolution of galaxy structure over cosmic time, Annual Review of Astronomy and Astrophysics 52 (1) (2014) 291–337. doi:10.1146/annurev-astro-081913-040037.
URL https://doi.org/10.1146/annurev-astro-081913-040037

[9] K. W. Willett, C. J. Lintott, S. P. Bamford, K. L. Masters, B. D. Simmons, K. R. V. Casteels, E. M. Edmondson, L. F. Fortson, S. Kaviraj, W. C. Keel et al., Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey, Monthly Notices of the Royal Astronomical Society 435 (4) (2013) 2835–2860. doi:10.1093/mnras/stt1458.

[10] Vika, Marina, Vulcani, Benedetta, Bamford, Steven P., Häußler, Boris, and Rojas, Alex L., MegaMorph: classifying galaxy morphology using multi-wavelength Sérsic profile fits, A&A 577 (2015) A97. doi:10.1051/0004-6361/201425174.
URL https://doi.org/10.1051/0004-6361/201425174

[11] S. Dieleman, K. W. Willett, and J. Dambre, Rotation-invariant convolutional neural networks for galaxy morphology prediction, Monthly Notices of the Royal Astronomical Society 450 (2) (2015) 1441–1459. arXiv:1503.07077, doi:10.1093/mnras/stv632.

[12] D. Tuccillo, M. Huertas-Company, E. Decencière, and S. Velasco-Forero, Deep learning for studies of galaxy morphology, in: M. Brescia, S. G. Djorgovski, E. D. Feigelson, G. Longo, and S. Cavuoti (Eds.), Astroinformatics, Vol. 325 of IAU Symposium, 2017, pp. 191–196. arXiv:1701.05917, doi:10.1017/S1743921317000552.

[13] P. H. Barchi, R. R. de Carvalho, R. R. Rosa, R. Sautter, M. Soares-Santos, B. A. D. Marques, E. Clua, T. S. Gonçalves, C. de Sá-Freitas, and T. C. Moura, Machine and Deep Learning Applied to Galaxy Morphology – A Comparative Study, arXiv e-print (Jan 2019). arXiv:1901.07047[astro-ph.IM].
URL https://arxiv.org/abs/1901.07047

[14] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, in: D. E. Rumelhart and J. L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 194–281.

[15] M. Johnson, M. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. Berkley, J. Johansson,

P. Bunyk et al., Quantum annealing with manufactured spins, Nature 473 (2011) 194–8. `doi:10.1038/nature10012`.

[16] K. L. Pudenz and D. A. Lidar, Quantum adiabatic machine learning, Quantum Information Processing 12 (5) (2013) 2027–2070. `doi:10.1007/s11128-012-0506-4`.
URL `https://doi.org/10.1007/s11128-012-0506-4`

[17] S. H. Adachi and M. P. Henderson, Application of quantum annealing to training of deep neural networks, arXiv e-print (2015). `arXiv:1510.06356[quant-ph]`.
URL `https://arxiv.org/abs/1510.06356`

[18] M. Benedetti, J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning, Phys. Rev. A 94 (2016) 022308. `doi:10.1103/PhysRevA.94.022308`.
URL `https://link.aps.org/doi/10.1103/PhysRevA.94.022308`

[19] J. T. Rolfe, Discrete Variational Autoencoders, arXiv e-print (2016). `arXiv:1609.02200[stat.ML]`.
URL `https://arxiv.org/abs/1609.02200`

[20] A. Vahdat, E. Andriyash, and W. G. Macready, DVAE#: Discrete Variational Autoencoders with Relaxed Boltzmann Priors, arXiv e-print (2018). `arXiv:1805.07445[stat.ML]`.
URL `https://arxiv.org/abs/1805.07445`

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[22] H. Larochelle and Y. Bengio, Classification using discriminative restricted Boltzmann machines, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, ACM, New York, NY, USA, 2008, pp. 536–543. `doi:10.1145/1390156.1390224`.
URL `http://doi.acm.org/10.1145/1390156.1390224`

[23] G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (8) (2002) 1771–1800. `doi:10.1162/089976602760128018`.
URL `https://doi.org/10.1162/089976602760128018`

[24] V. Choi, Minor-embedding in adiabatic quantum computation: I. the parameter setting problem, Quantum Information Processing 7 (2008) 193–209.

[25] M. H. Amin, Searching for quantum speedup in quasistatic quantum annealers, Phys. Rev. A 92 (2015) 052323. `doi:10.1103/PhysRevA.92.052323`.
URL `https://link.aps.org/doi/10.1103/PhysRevA.92.052323`

[26] C. McGeoch, Comparison of 2000Q to Lower-Noise 2000Q, Qubits North America 2019 (2019).
URL `https://www.dwavesys.com/sites/default/files/14-DWMcGeogh.pdf`

[27] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, P. L. McMahon, T. Umeki, K. Enbutsu et al., A coherent ising machine for 2000-node optimization problems, Science 354 (6312) (2016) 603–606. `doi:10.1126/science.aah4243`.
URL `https://science.sciencemag.org/content/354/6312/603`

[28] M. Aramon, G. Rosenberg, E. Valiante, T. Miyazawa, H. Tamura, and H. G. Katzgraber, Physics-inspired optimization for quadratic unconstrained problems using a digital annealer, Frontiers in Physics 7 (2019) 48. `doi:10.3389/fphy.2019.00048`.
URL `https://www.frontiersin.org/article/10.3389/fphy.2019.00048`

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12, Curran Associates Inc., USA, 2012, pp. 1097–1105, `http://dl.acm.org/citation.cfm?id=2999134.2999257`.
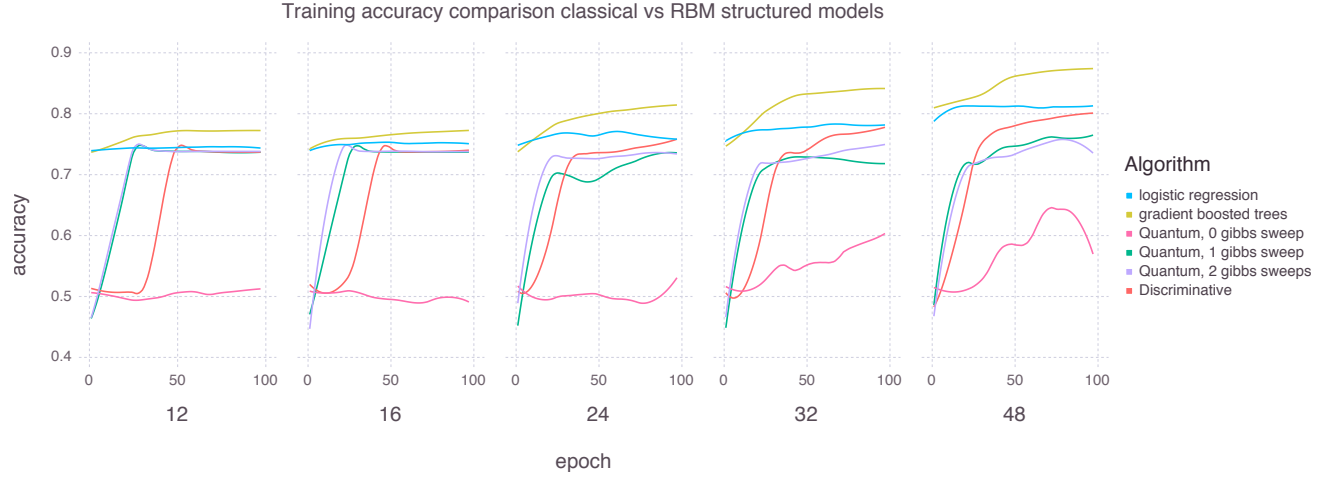
(a) Comparison of algorithm accuracy for standard classification techniques, logistic regression and gradient boosted decision trees, against quantum-based RBM training along with RBM training using efficient discriminative RBM training for different sized RBMs. The classical models strictly outperform all RBM-basd models. Note: epoch for the tree-based model corresponds to the number of decisions trees.
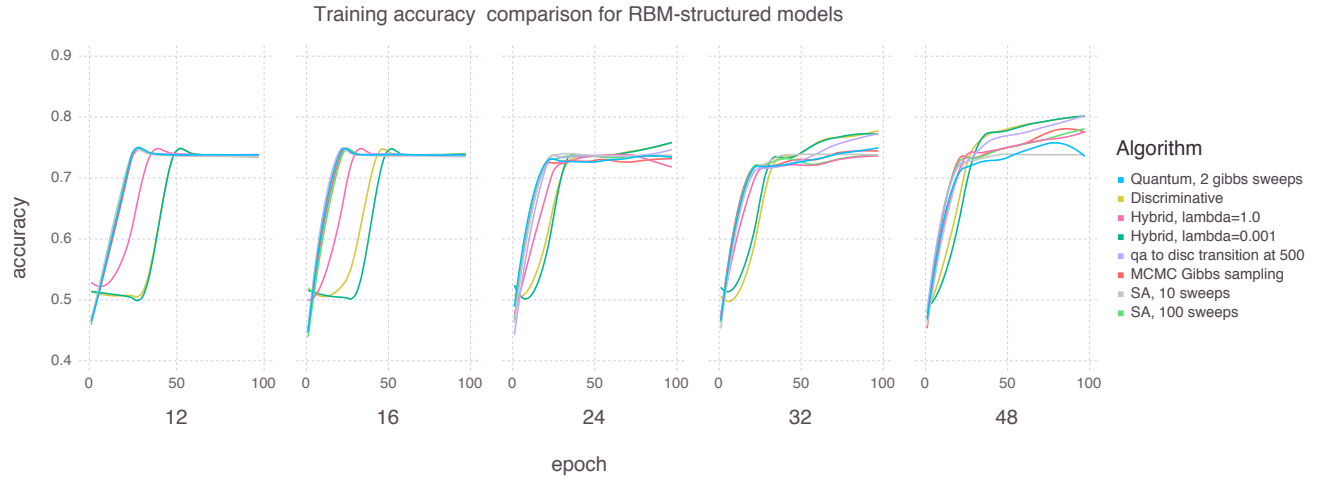
(b) Comparison of various algorithms accuracy as a function of the number of epochs for difference sized RBM-structured models. In summary, quantum training appears to broadly achieve higher accuracy early in training but either match (for small RBMs) or underperform relative to pure discriminative and hybrid training with small values of λ (for large RBMs). Transitioning from QA to discriminative training at approximately the localtion of the crossover in performance does not yield as high an accuracy at the end of 100 epochs as pure discriminative training. Data shown is for batch size of 128. Moreover, QA-style results seem to be approximately reproducible with fairly brief simulated annealing runs. The best final performance is from purely discriminative training.
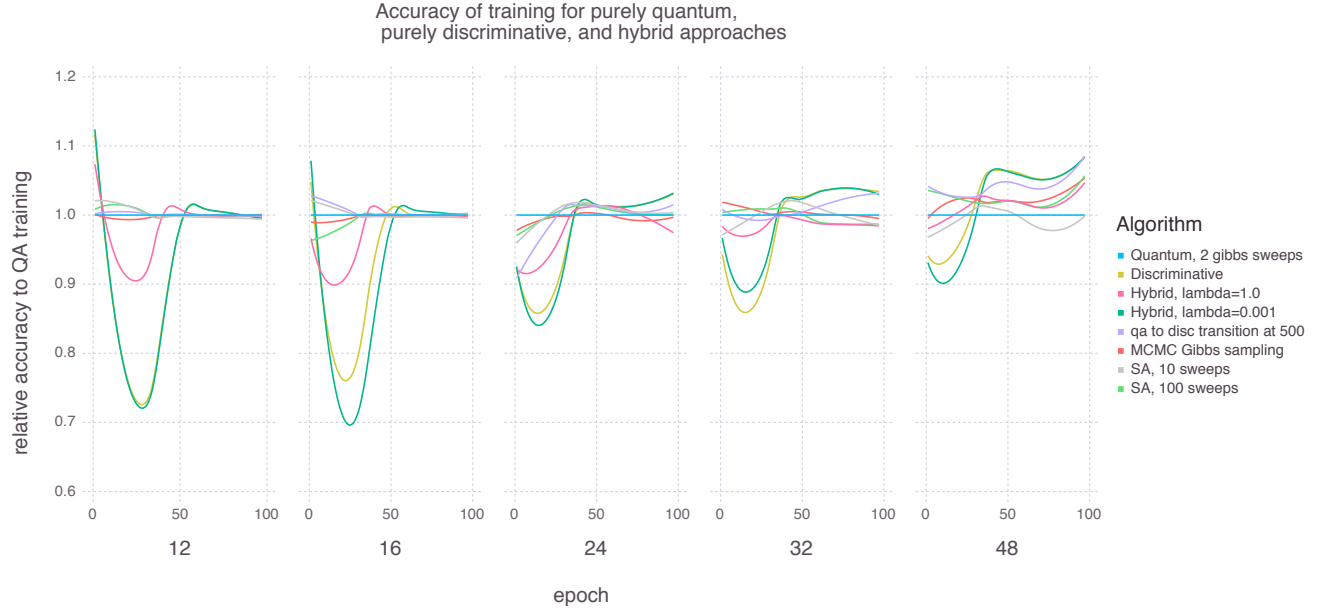
FIG. 9.

FIG. 10. Comparison of the ratio of various algorithms accuracy against the accuracy of QA training (with 2 Gibbs sweeps) per epoch. Values larger than one imply higher accuracy for the algorithm than QA, below one worse accuracy. This only displays RBM-structured models. Logistic regression and gradient boosted trees dominate RBM-structured models and are not included so as to maximize resulution in the comparison among RBM-structured models. Note: epoch for the tree-based model corresponds to the number of decisions trees.
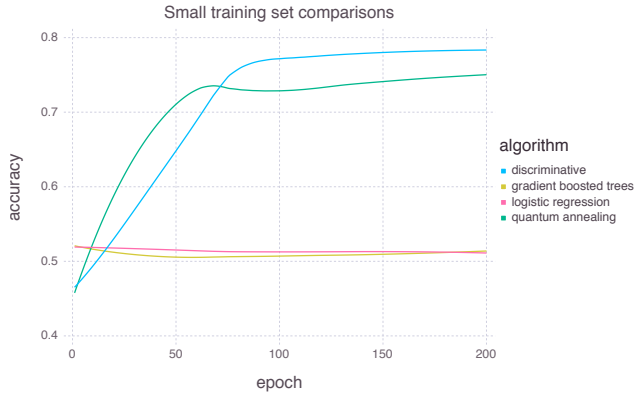

FIG. 11. Comparison of the different algorithms' accuracies on the test set when trained on only 250 training set examples. We see that RBM beat the two algorithms we compare against, which both overfit the training set. This is the case even for the gradient boosted classifier which is known as being robust to overfitting. The two RBM training routines show a similar pattern to that of the larger training set.