Computational Materials Science 173 (2020) 109429

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Boltzmann machine modeling of layered MoS_2 synthesis on a quantum annealer

Jeremy Liu[a,b], Ankith Mohan[a], Rajiv K. Kalia[c], Aiichiro Nakano[c], Ken-ichi Nomura[c,*], Priya Vashishta[c], Ke-Thia Yao[a]

[a] Department of Computer Science, University of Southern California, Los Angeles, CA 90089, United States
[b] Information Sciences Institute, University of Southern California, Los Angeles, CA 90089, United States
[c] Collaboratory of Advanced Computing and Simulations, University of Southern California, Los Angeles, CA 90089, United States

ARTICLE INFO

Keywords:
Layered material
Boltzmann machine
Quantum annealer

ABSTRACT

Two-dimensional layered materials are usually synthesized by chemical vapor deposition (CVD). Here we use the Restricted Boltzmann Machine (RBM) and its enhancement called Limited Boltzmann Machine (LBM) on a quantum computer to model the chemical vapor deposition (CVD) growth data for a MoS_2 monolayer. The CVD data for Boltzmann machines are generated by reactive molecular dynamics (RMD) simulations. The Boltzmann machines give insight into the semiconducting (2H) and metallic (1T) phases and defects generated during the CVD growth of MoS_2. We compare the performances of the two types of Boltzmann machine models, i.e. RBM versus augmented topologies with limited intra-layer coupling between the hidden units (LBM). We show that limited topologies have a performance advantage over restricted topologies. We examine connectivity within our topological variants, explore hardware qubit mapping schemes, and discuss what performance differences may imply about locality within the data without prior knowledge.

1. Introduction

Technologies based on two-dimensional (2D) materials such as graphene and transition metal dichalcogenides (TMDCs) will play a crucial role in the development of the next-generation devices [1,2]. Typical 2D materials are atomically thin, comprising a single or a few layers that are weakly bonded by van der Walls interaction. In contrast, the intra-layer atomic bonding in layered materials is strong and usually covalent or ionic in nature. A large variety of experimental work has shown that TMDCs display unique mechanical, electrical and optical properties. For example, the switching capability of a MoS_2 monolayer between semiconducting and metallic phases can be controlled by a laser. Furthermore, unlike bulk MoS_2 with an indirect band gap of 1.28 eV, 2D MoS_2 exhibits a direct band gap of 1.8 eV.

A MoS_2 monolayer has several polymorphs, such as 1T and 2H phases, with different intra-layer structures. The 1T phase consists of octahedral prismatic molecular units with ABC stacking and the 2H phase is trigonal prismatic with ABA stacking (see Fig. 1). This difference in the stacking-order endows MoS_2 with distinct materials properties, e.g. the 1T phase is metallic and the 2H phase is semiconducting.

A robust and high-yield synthesis procedure is critically needed to generate single-layered 2D materials for TMDC-based engineering applications and devices. A variety of experimental methods, e.g. chemical vaper deposition (CVD), mechanical and chemical exfoliation, and hydrothermal synthesis have been used to synthesize TMDC monolayers. Efforts to synthesize device-scale monolayer samples tend to produce polycrystalline domains and multi-layered heterogenous structures. Grain boundaries and defects introduce undesired carrier trap states, which are known to deteriorate mechanical and electrical properties of 2D TMDCs.

CVD is the most widely used approach to synthesize highly uniform and large-scale samples of 2D TMDCs [3]. We have performed large-scale reactive molecular dynamics (RMD) simulations to study the CVD growth of a MoS_2 monolayer on an alumina substrate and used two types of Boltzmann machines (BMs) to model the RMD data (Fig. 2). BMs are composed of many interconnected units. Typically, these are binary but have been extended to continuous variables [4]. BMs are most often used to pre-train layers of deeper networks. In practical applications, BMs make simplifying modifications to their architecture, discussed later, for computational tractability.

Some units of a BM are dedicated to representing data and are called visible units. The remaining so-called hidden units can be seen as representing latent factors that influence the distribution of other unit states. Sampling the network, a process essential to both training and

* Corresponding author.
E-mail address: knomura@usc.edu (K.-i. Nomura).

https://doi.org/10.1016/j.commatsci.2019.109429
Received 16 October 2019; Received in revised form 20 November 2019; Accepted 21 November 2019

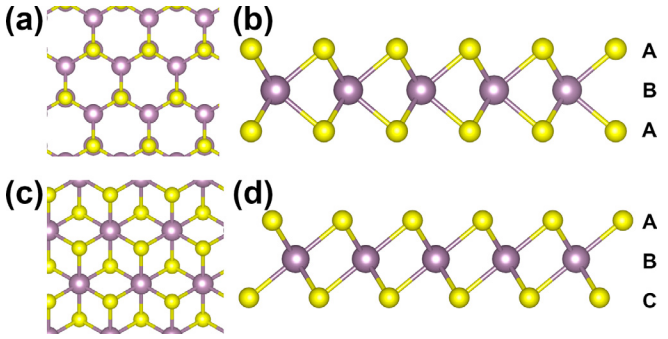

Fig. 1. 1T and 2H structures of 2D MoS_2 consisting of a single layer of molybdenum atoms (purple) sandwiched by two layers of sulfur atoms (yellow). (a) & (b) are top and side views of the 2H phase, and (c) & (d) are the top and side views of the 1T phase.

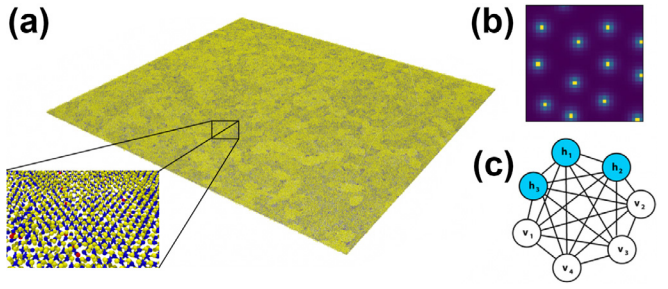

Fig. 2. MoS_2 monolayer structure obtained by RMD simulation and thermal annealing of CVD process (a). Yellow spheres represent S (sulfur) atoms, blue spheres are Mo (molybdenum), and red spheres are O (oxygen) atoms. Atomic configurations are converted into image representations (b) and then passed to a Boltzmann machine (c). The visible units $\boldsymbol{v}$ of a Boltzmann machine are set to represent a given image.

output, becomes intractable if the units are fully connected as shown in Fig. 2c. However, a network topology restricted to a bipartite connectivity between visible and hidden units creates conditional independence that allows for efficient inference of unit distributions [5]. This modified network is called a restricted BM (RBM).

We also use an alternative to the RBM called the limited BM (LBM). This topological variant allows sparse connectivity among the hidden units, something normally disallowed under the RBM topology as it breaks conditional independence. The LBM is a connection superset of the RBM and should enjoy a performance advantage since it can utilize connections normally unavailable to the RBM. Since inference is no longer straightforward in the LBM, we resort to sampling to estimate unit state distributions and train a model. Here we use D-Wave's adiabatic quantum annealer for our sampling needs.

The annealer samples from Ising spin-glass problems defined by the user, which we find useful because BMs can be easily restated as Ising models. Our implementation of BMs using the quantum annealer differs slightly from other work in that we only place hidden units on the hardware instead of the whole BM [6]. Our purpose is not to prove any sort of quantum speedup or supremacy but to practically utilize an interesting, available tool and apply it to our work. Our implementation of LBMs was designed with the annealer's architectural limitations in mind.

Whereas mathematically there is no limit to the number of other units a given unit can be connected to in a BM, D-Wave's annealer has to contend with real-world physical limitations. The annealer arranges its quantum bits, or qubits (each of which would represent one unit in a BM), according to a chimera topology [7]; see Fig. S1 in the Supplementary material. This is done for practical engineering purposes, being a relatively newer technology with limited connectivity between qubits as a result. A more detailed description of both the topology and how it affected our implementation is provided in the Supplementary material.

2. Simulation method and workflow

We use a combination of BMs and quantum annealers to model RMD simulation data for the CVD growth of 2D MoS_2. The RMD simulation involved 1,497,600 O atoms, 2,347,200 S atoms and 460,800 Mo atoms and the system dimensions were 211.0 × 96.3 × 14.5 (nm^3). The simulation was based on a newly developed ReaxFF force field, especially optimized to model the CVD process for MoS_2. The force-field parameters were validated by extensive theoretical and experimental data to reproduce structural properties as well as the key chemical reaction pathways in the reduction and sulfurdization steps of MoO_3 substrate. Details of the simulation schedule and CVD process modeling are described elsewhere [8]. The RMD simulation was performed on 524,288 IBM BlueGene/Q cores of Mira and Intel Xeon Knights Landing (KNL) 262,144 cores of Theta machines at Argonne Leadership Computing Facility.

Developing a Boltzmann machine to restore corrupt/missing data is a scientifically important and interesting problem because the ability to do so implies the model has some deeper understanding of the rules that govern the data. The RBM atomic configurations were converted to a form understood by BMs. From the position of each Mo atom, three channels of 32×32 pixel images were generated with a kernel of $e^{-r/\sigma}$, where r is the distance between the position of an atom and the pixel location and σ is chosen to be 0.3 Å. The distribution represents the density of either S or Mo atom. A brighter pixel indicates its proximity to an atomic position in the Cartesian coordinates. The 32×32 pixels correspond to 1 nm^2 region on the x-y plane in the RMD simulation. We assigned one channel for the top S layer, one for the middle Mo layer, and the third channel for the bottom S layer to better capture the characteristic features of the dataset as well as to avoid the ambiguity in the image intensity if the three channels were overlapping (see Fig. 1(b) & (d)). The training data were labeled as 1T phase, 2H phase, and disordered (uncategorized) phases. We used 4500 images for training and 500 images for testing.

Training the BMs creates a model that captures important

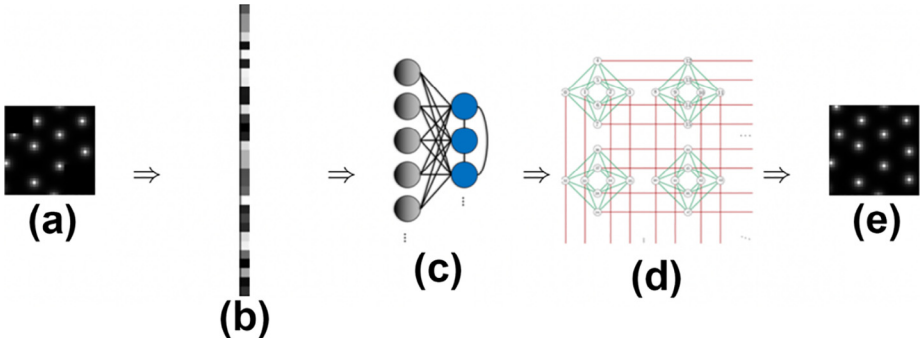

Fig. 3. Conceptual workflow for recovering missing data. In (a) we have the input image with the top-left and bottom-right corners missing. This is converted into the vector representation seen in (b). This vector is passed to the visible units of the Boltzmann machine in (c). The hidden units of the Boltzmann machine are mapped onto an adiabatic quantum annealer in (d), which returns an image with the missing corners filled in in (e).

relationships among the data. A model trained in this way is then able to perform a denoising task and fill in missing or corrupted portions of an input image. Fig. 3 shows the entire process for training RBM and LBM. RMD data is converted to a pixel image format with removed or corrupt portions of the images. The corrupted input is translated into a vector format that can be understood by a BM. Once converted, the vector is passed to a BM and the states of the visible units are set accordingly to represent the data. The parameters defining the BM are translated into equivalent Ising spin-glass parameters, and logical BM units are mapped to qubits on the annealer hardware. After several iterations of Gibbs sampling on the annealer, we recover the resulting visible states representing the restored and denoised input image.

3. Limited boltzmann machine (LBM) modeling on a quantum annealer

BMs are defined by an energy function $E(\boldsymbol{v}, \boldsymbol{h}; \theta)$ which depends on visible units $\boldsymbol{v}$ containing the data and hidden layer units $\boldsymbol{h}$. Here θ represents hyper-parameters which include the weight matrix W connecting visible and hidden units, and biases for all the visible and hidden units. The probability of a given state is governed by the Boltzmann distribution $\boldsymbol{P}(\boldsymbol{v}, \boldsymbol{h}, \theta) = \boldsymbol{e}^{-E(\boldsymbol{v},\boldsymbol{h},\theta)}/\boldsymbol{Z}$, where the denominator is known as the partition function: $\boldsymbol{Z} = \sum_{\boldsymbol{v},\boldsymbol{h}} \boldsymbol{e}^{-E(\boldsymbol{v},\boldsymbol{h},\theta)}$.

One method of training BMs to learn a given distribution is called contrastive divergence (CD) [5]. In CD, the weight matrix W is updated according to the rule

$$\Delta \boldsymbol{W_{ij}} = \boldsymbol{\epsilon}\left(\langle \boldsymbol{h_i v_j}\rangle_{\boldsymbol{data}} - \langle \boldsymbol{h_i v_j}\rangle_{\boldsymbol{recon}}\right) \tag{1}$$

where $\boldsymbol{\epsilon}$ is a learning rate hyper-parameter. CD can be split into positive and negative phases, each corresponding to calculating $\langle \boldsymbol{h_i v_j}\rangle_{\boldsymbol{data}}$ and $\langle \boldsymbol{h_i v_j}\rangle_{\boldsymbol{recon}}$ respectively. The expression $\langle \boldsymbol{h_i v_j}\rangle$ is the probability both units i and j are "on" together. In the positive phase (where network activity is being driven by input *data*), the values of the visible units are clamped to a training input's value; in the negative phase (where the network activity is being driven by *reconstruction* efforts), no units are clamped and the network is allowed to settle.

It is here that network topologies diverge in consideration of computational tractability. In the case of unrestricted connectivity, calculating $P(h_i = 1|v)$ is very difficult because $P(h_i = 1|v)$ is still dependent on other hidden units in the network. This is an exponentially large space that is only viable for small networks. However, if we are allowed to make simplifying assumptions about the network, we can introduce independence in the calculation of $P(h_i = 1|v)$ to make it much more straightforward. In a RBM, the only allowed connection type is between a visible unit v_i and a hidden unit h_j. Therefore,

$$P(h_i{=}1|v) = \prod_{j=1}^{n} P(h_i{=}1|v_j) \tag{2}$$

Likewise, we can write

$$P(v_i{=}1|h) = \prod_{j=1}^{n} P(v_i{=}1|h_j) \tag{3}$$

With these expressions and an assumption of a simplified connectivity topology, training RBMs [9] becomes efficient. There is some cost to the representational power of the RBM, although connections between units of the same type that may have contributed valuable information to the BM's performance are deliberately removed for the sake of tractability.

We modified the RBM by adding some sparse connections between hidden units; we call this a limited BM, or LBM. Under the CD training procedure, such a change would run into tractability problems again when calculating $P(h|v)$ because the conditional independence is lost (although calculating $P(v|h)$ remains the same as before). Because we would be unable to infer the proper distribution of hidden units, we opted to use an adiabatic quantum annealer to perform the sampling task instead. The annealer is described in the Supplementary material.

Our RBM and LBM implementations use 32 × 32 × 3 = 3072 visible units and 128 hidden units. The visible unit count arises from the need to represent our input data (32 × 32 pixel images in 3 slices), and the hidden unit count arises from the chimera topology of the annealer (16 cells each containing 8 qubits). The weight matrix W in both RBM and LBM are subsequently 3072 × 128 in the number of parameters; the LBM has an additional 352 couplers which connect hidden units together[1].

We randomly initialized the weight matrix using a normal distribution. Our learning rate hyper-parameters for visible-to-hidden connection weights (ε_w), hidden-to-hidden couplers (ε_c), visible unit biases (ε_{vb}), and hidden unit biases (ε_{hb}) were all set to 0.1. The noise control learning parameter was set to $\alpha = 1$ and the variance to $\sigma_\alpha = 0.1$. We also used the ADAM hyper-parameter tuning algorithm [10] and set its parameters to $\alpha_{ADAM} = 0.005$, $\varepsilon_{ADAM} = 10^{-6}$, $\beta_1 = 0.9$, and $\beta_2 = 0.9$. We trained BMs over 50 epochs for a data set of 4500 training examples and 500 test images.[1]

4. Results

We were able to successfully train BMs to model MoS_2 data. One sign of a trained network is if it is able to perform an encoding-decoding task. That is, a BM is exposed to an input image by clamping visible unit values to correspond to the image pixel values. This produces a set of hidden unit states, at which point the network releases the clamps on the visible unit states and allows the states to settle. The set of hidden unit states, the encoding of the input image, is said to drive the network to produce a new set of visible unit states, the decoding. The decoding result should be similar to the input. Fig. 2b shows the result of one encode-decode task that looks very similar to the input data originally derived from the RMD simulation shown in Fig. 2a.

In addition to the encoding-decoding task, BMs can perform a task called denoising. Given an image with some portions omitted or corrupted, a BM can fill in or correct the bad portions of the input. BMs can perform this task due to their nature as generative models of data as opposed to discriminative models more commonly found in machine learning. Recovering data can be done using Markov-chain Monte-Carlo methods. For our BMs, we clamp the pixel values of our given data and impute random values to the missing portions. This induces a distribution of hidden states, which we then use to calculate new values for our visible units representing the pixel values. We repeatedly sample the hidden and visible states until the network converges on some values for the missing visible units. Fig. 4a shows an example of a BM performing this denoising task. As seen, the results are reasonable completion of the missing data. We also compared RBM against LBM on this encoding-decoding task. Fig. 4b shows an error plot of the two topologies in which the LBM shows a slight advantage over the RBM.

Next, we discuss the results of a qubit remapping experiment. Earlier we introduced the idea of changing the assignment of qubits to chimera cells in the hopes of materially affecting our end results, reasoning that as qubits begin to specialize as proto feature we might find a good way to pair them together. These feature detectors may have useful information for other detectors, but they may be unable to communicate such information due to chimera topology connectivity restrictions, hence our remapping of qubits to chimera cells.

We chose to use correlation in hidden unit activities to create a qubit remapping strategy. When training a BM, we recorded the hidden unit states for all images. We halted training after improvements in reconstruction error dropped below 500 per iteration and calculated Pearson correlation coefficients for the qubits. We used these

[1] For an annealer with $M \times N$ chimera cells containing 2 partitions of size L qubits each, we have $L^2MN + LM(N-1) + LN(M-1)$ couplers. In our case, $M = 4$, $N = 4$, $L = 4$. Each cell contains L^2 connections between the 2 partitions and has L connections with qubits in the adjacent cell, except for cells on the edges of the chimera cell grid.

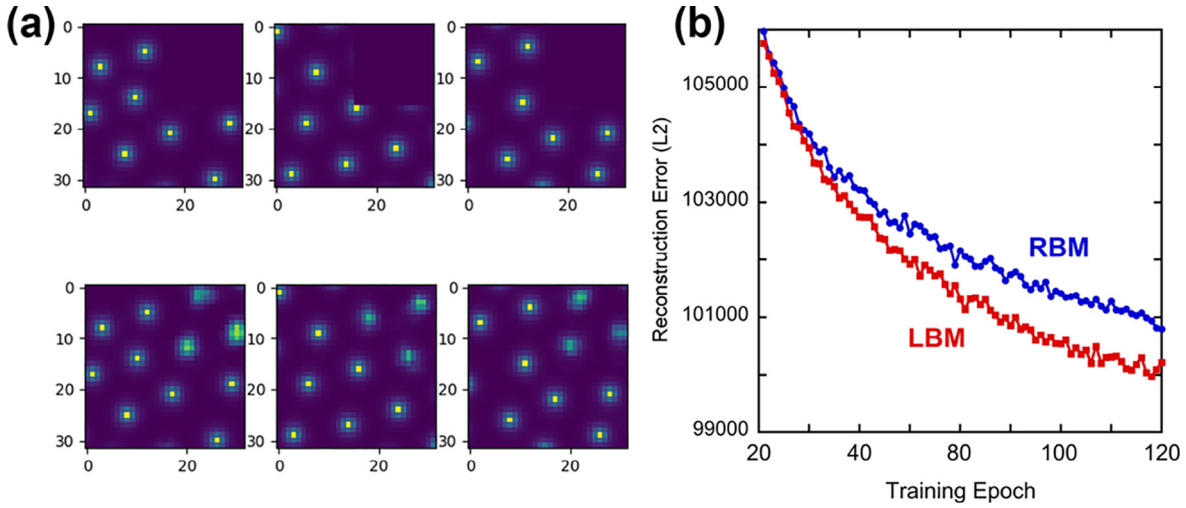

Fig. 4. A comparison of RBM and LBM performance using L2 reconstruction loss, the squared sum of per-pixel differences between the data set and the BM's resulting encoding-decoding output. (a) Boltzmann machines can fill in missing or noisy data. In the top row are images of the simulation subspace with the top-right quarter removed. The Boltzmann machine then returns the completed images seen in the bottom row, which appear to be reasonable estimates. (b) A comparison of RBM and LBM performance using L2 reconstruction loss. LBMs show a small but consistent advantage over RBMs, suggesting the intralayer connectivity between hidden units is beneficial for learning.

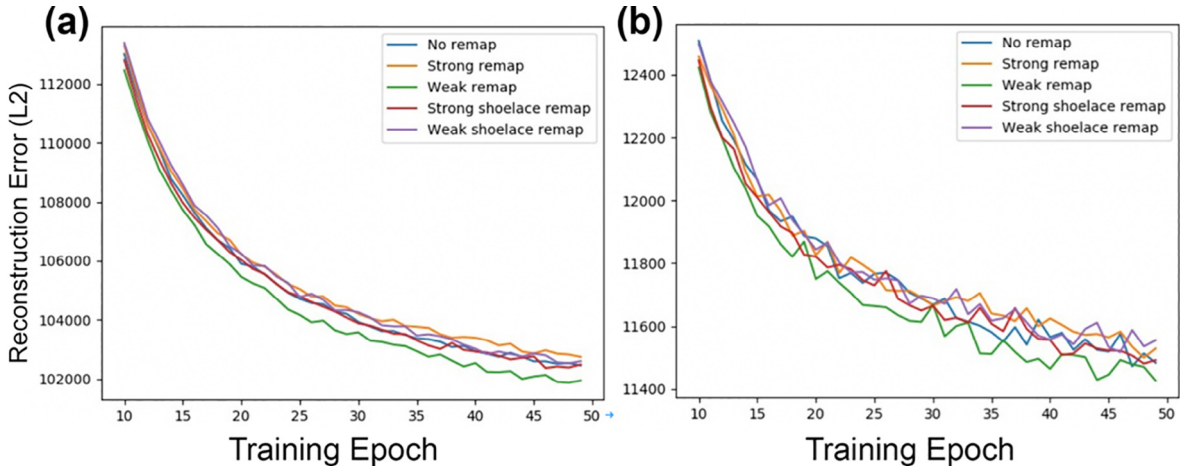

Fig. 5. (a) Remapping results on a training data set. Weak correlation remapping produces the overall best result on the training data and strong correlation remapping yields a worse result. (b) Remapping results on a test data set. Note that the test data set is 10% the size of the training data set, thus the discrepancy in the error numbers compared to the training data set. The same trends found in Fig. 5a continue in these results from the test data set.

coefficients to find either a maximally or minimally correlated pair of qubits and seeded them into opposing partitions of a chimera cell. Here, maximal correlation means the qubits are either strongly positively or negatively correlated, and minimal correlation means the qubits have correlation coefficients nearer to 0. We filled the remainder of a chimera cell with either maximally or minimally correlated qubits relative to the seed qubits. We referred to the maximal correlation method as strong remapping and the minimal correlation method as weak remapping. We also created an alternative method of filling the cells we called shoelace remapping. Instead of completely filling each partition based on a pair of seed qubits, we would only add one new pair and consider them new seed qubits, repeating until the cell was filled. This causes the cell to be filled two qubits at a time, resembling the tying of shoelaces. After performing these remappings, we continued training the BMs, the results of which can be seen in Fig. 5.

In general, the weak remapping gave better results. We reason this is the case because qubits that are highly correlated have a lower chance of having relevant information for each other. In an extreme case, suppose two qubits are perfectly correlated and linked to each other. In such a case, neither qubit is useful to the other and we could delete one qubit without losing any information. If we do a weak remapping, however, we avoid such an effect and give qubits new opportunities to interact with different neighbors and incorporate new, relevant information into their own activations.

Classification of 2H and 1T phases and defects in CVD growth of MoS_2 and other 2D materials such as $MoSe_2$ and $MoWSe_2$ is of great interest. The BMs provide a slight benefit in classification results. Though not the main focus of this paper, the use of a support vector machine (SVM) with a radial basis function kernel (RBF) on raw image data gave a classification accuracy of 66.2%. We then trained a LBM to model the images and recorded the hidden unit states produced by the BM in response to exposure to the input images – the encodings. We used these encodings as input to another SVM, still using the RBF kernel, which yielded a classification accuracy of 77.0%. Doubtlessly, such classification efforts can be tuned and optimized much better, but this suffices to show that BMs can successfully classify defects, and 2H and 1T phases in CVD growth of MoS_2.

5. Conclusion

In this study, we have shown that the RBM and its variant LBM can successfully model RMD simulation data for CVD growth data of a MoS_2 monolayer. Since scientific problems can be simulated on classical machines in most cases, the quantum annealer might not be strictly necessary to implement the Boltzmann machines. However, researchers are constantly looking for cases where quantum devices produce results that classical simulations cannot. The annealer might be a useful tool if confronted with some especially complex data. Some of our research is directed towards how we might implement networks with additional connectivity, therefore we found it appropriate to try out the annealer. Another reason we might use the annealer instead of a classical solution is that simulators may not scale well. Using the quantum annealer requires adjusting parameters on individual qubits. Additionally, because the annealer is a real-world piece of hardware, not all components are operational. Some qubits are broken, some couplers are broken, and this can affect how to program problems onto the chip. The route we chose was to find contiguous areas of the chip that have fully operational components and constrain our problems to fit within that area.

We have also demonstrated that the LBM maps well on a D-Wave adiabatic quantum annealer and performs better than the RBM. We explored different qubit mapping schemes and their effect on the performance of LBM. In addition to the encoding-decoding task, we have demonstrated that both RBM and LBM can perform denoising and fill in the missing data in RMD simulations of MoS_2 growth by the CVD process. These unsupervised learning capabilities of Boltzmann machines hold a great deal of promise for modeling material structures and characterization data from various kinds of instruments, in particular electron microscopy and X-ray scattering probes.

CRediT authorship contribution statement

Jeremy Liu: Conceptualization, Investigation, Methodology, Software, Validation, Writing - original draft. **Ankith Mohan:** Investigation, Methodology, Software, Validation. **Rajiv K. Kalia:** Conceptualization, Investigation, Methodology, Formal analysis, Writing - original draft, Supervision, Funding acquisition. **Aiichiro Nakano:** Conceptualization, Resources, Funding acquisition. **Ken-ichi Nomura:** Investigation, Methodology, Software, Writing - original draft. **Priya Vashishta:** Conceptualization, Resources, Funding acquisition. **Ke-Thia Yao:** Conceptualization, Methodology, Resources, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences, Materials Science and Engineering Division, Grant DE-SC0018195. This work was performed, in part, at the Center for Integrated Nanotechnologies, an Office of Science User Facility operated for the U.S. Department of Energy (DOE) Office of Science. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. DOE's National Nuclear Security Administration under contract DE-NA-0003525. The views expressed in the article do not necessarily represent the views of the U.S. DOE or the United States Government. RMD simulations were performed at the Argonne Leadership Computing Facility under the DOE INCITE and Aurora Early Science programs, and at the Center for High Performance Computing of the University of Southern California.

Data availability

The raw data required to reproduce these findings are available to download from doi: 10.17632/8pc8x7c3g6.1

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.commatsci.2019.109429.

References

[1] A.K. Geim, I.V. Grigorieva, Van der Waals heterostructures, Nature 499 (7459) (2013) 419–425.
[2] I. Tung, et al., Anisotropic structural dynamics of monolayer crystals revealed by femtosecond surface x-ray scattering, Nat. Photon. (2019) published online.
[3] S. Hong, et al., Computational synthesis of MoS2 layers by reactive molecular dynamics simulations: initial sulfidation of MoO3 surfaces, Nano Lett. 17 (8) (2017) 4866–4872.
[4] G.E. Hinton, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507.
[5] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800.
[6] M. Benedetti, J. Realpe-Gómez, R. Biswas, A. Perdomo-Ortiz, Quantum-assisted learning of hardware-embedded probabilistic graphical models, Phys. Rev. X 7 (2017) 041052.
[7] R. Harris, et al., Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor, Phys. Rev. B 82 (2) (2010) 15.
[8] S. Hong, et al., Defect healing in layered materials: a machine learning-assisted characterization of MoS2 crystal-phases, J. Phys. Chem. Lett. (2019).
[9] R. Hamerly, et al., Experimental investigation of performance differences between Coherent Ising Machines and a quantum annealer, 2018, pp. 1–26.
[10] D.P. Kingma, J. Ba Adam: A Method for Stochastic Optimization, 2014, 58–62.