Machine Learning the Square-Lattice Ising Model

Burak Çivitcioğlu[1], Rudolf A. Römer[2] and Andreas Honecker[1]

[1] Laboratoire de Physique Théorique et Modélisation, CNRS UMR 8089, CY Cergy Paris Université, Cergy-Pontoise, France
[2] Department of Physics, University of Warwick, Coventry, CV4 7AL, United Kingdom

E-mail: `burak.civitcioglu@cyu.fr`

Abstract. Recently, machine-learning methods have been shown to be successful in identifying and classifying different phases of the square-lattice Ising model. We study the performance and limits of classification and regression models. In particular, we investigate how accurately the correlation length, energy and magnetisation can be recovered from a given configuration. We find that a supervised learning study of a regression model yields good predictions for magnetisation and energy, and acceptable predictions for the correlation length.

1. Introduction

Condensed-matter physics studies the properties of matter with many constituents and interactions [1]. One central task in condensed-matter and statistical physics is to identify different phases of the system and locate the critical points separating these in the space of parameters. Machine learning (ML) is the study of algorithms that improve their performance by gaining experience from data. It has been used for image classification, object detection, self-driving cars, speech recognition and many other tasks that rely on data [2]. ML is commonly categorised into supervised, unsupervised and reinforcement learning. Recently, ML methods have been employed to tackle condensed-matter problems and have shown promising predictions in topics such as studying the phases of the Ising model [3–5], disordered quantum systems [6,7], phase transitions in the Bose-Hubbard [8,9], the Blume-Capel model [10], a highly degenerate biquadratic-exchange spin-one Ising variant, and the two-dimensional (2D) XY model [10], as well as material properties [11].

The square-lattice Ising model is an exactly solved model [12]. Notwithstanding its exact solution, it is commonly employed in the ML context [3–5, 13–18]. Motivations for such ML studies of the Ising model in 2D include the readily available comparison to the exact solution and that training data can be easily generated using standard Monte-Carlo simulations. Here, we continue along these lines and investigate the problem of predicting the temperature dependence of energy E, magnetisation m and correlation length ξ for the square-lattice Ising model by ML of configurations.

2. Ising Model, Methods, Data and the ML Model

The ferromagnetic Ising model is defined by the Hamiltonian
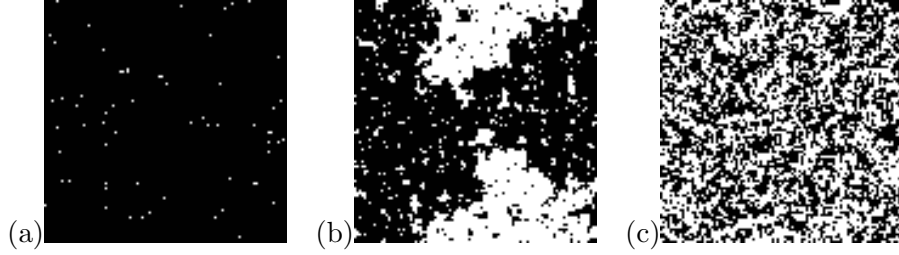
$$E = -\sum_{\langle i,j \rangle} s_i s_j, \tag{1}$$

Figure 1. Configurations of the Ising model on a 100×100 square lattice with periodic boundary conditions at temperatures (a) $T = 1.5 < T_c$, (b) $2.3 \sim T_c$ and (c) $4 > T_c$. In each configuration, black and white pixels represent spin up and spin down states, respectively. Panel (a) shows a ferromagnetic configuration at low temperature, (b) a critical configuration close to T_c and (c) a paramagnetic configuration at high temperatures.

where s_i is the spin at site i that can point up ($+1$) or down (-1) and $\langle i, j \rangle$ are nearest-neighbour pairs [19,20]. Here we focus on 100×100 square lattices subject to periodic boundary conditions. This finite-sized choice is mainly motivated by the ML context where the codes have been optimised for treating images whose size does not exceed a few hundred times a few hundred pixels. For a given configuration, we define the magnetisation m and the correlation length ξ, respectively, as

$$m = \frac{1}{N} \left| \sum_i s_i \right|, \quad \xi^2 = \frac{\sum_r r^2 g(r)}{\sum_r g(r)}, \tag{2}$$

where the two-point correlation function and the distance between lattice sites are denoted as $g(r)$ and r, respectively [21]. We note that the correlation length ξ quantifies the characteristic scale of a configuration, as encoded in the correlation function $g(r)$.

We employ the Metropolis algorithm to generate many 100×100 configurations as data for the ML approach [19,20,22]. We start the simulations from an initial, hot temperature $T = 100$. This is subsequently slowly reduced to $T = 4$ in steps of $\Delta T = 0.09$. Staying at constant $T = 4$, five data samples are collected with $\Delta t_{\mathrm{MC}} = 1000$ Monte-Carlo sweeps between each measurement (as usual, a Monte-Carlo sweep is defined as one attempt on average to flip each spin in the system). Next, T is further reduced in steps of 0.001 to $T = 3.95$. This procedure is repeated at each $\Delta T = 0.05$ in the interval $[4.0, 1.5]$. we have checked that this procedure produces well-equilibrated configurations. The training and cross-validation data consists of 2080 configurations for each $T = 1.5, 1.55, 1.6, \dots, 4$ such we are having a total 51 distinct temperatures. 10% of this data is reserved as cross-validation data for ML. The test set consists of 250 configurations for each T with the same parameters as the training and cross-validation data. Each sample consists of the spatial configuration and its m, E, ξ and $g(r)$ values. Examples of configurations are shown in Fig. 1 with (b) close to the critical temperature, exactly known in the thermodynamic limit to be $T_c = 2/\ln(1 + \sqrt{2}) \approx 2.269$ [12], (a) in the ferromagnetic phase, $T < T_c$ and (b) in the paramagnetic phase, $T > T_c$.

For the ML classification and regression tasks, we use the same convolutional neural network (CNN) with eight convolutional layers (with a 3×3 kernel, stride 1 in each layer and the number of feature maps/filters increasing as $16, 16, 32, 32, \dots, 128$), and 2×2 pooling layers with $\mathrm{ReLU}(x) = \max(0, x)$ activation function, and 25% dropout between layers. This is capped by three dense layers (of size 512, 128, while the last dense layer has variable size K for classification and 1 for regression). For classification, K expresses the number of categories, and the softmax activation function $S : \mathbb{R}^K \to [0, 1]^K, S(\mathbf{r})_i = e^{r_i} / \sum_{j=1} e^{r_j}$, with $\mathbf{r} = (r_1, r_2, \dots, r_K) \in \mathbb{R}^K$, $i = (1, 2, \dots, K)$, is used in the output layer. The loss function is categorical cross entropy [2].
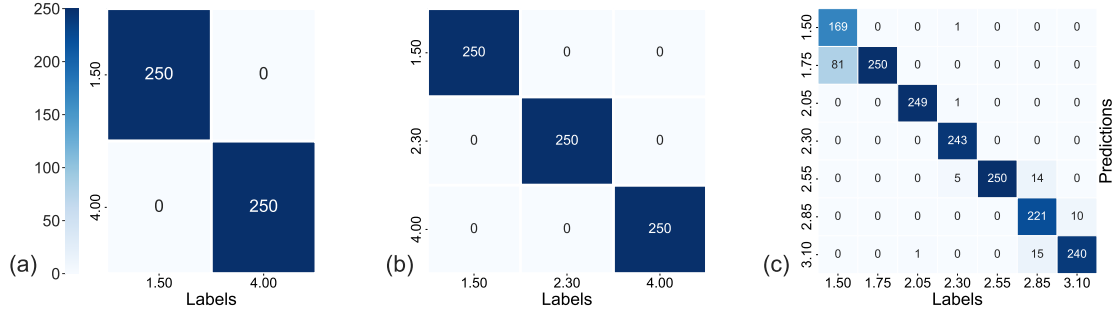
Figure 2. Confusion matrices for temperature classification. Horizontal axes denote the true values while the vertical axes indicate the predicted ones. The colour bar represents the number of samples in each entry. (a) Two classes: $T \in \{1.5, 4.0\}$, (b) three classes: $T = 1.5, 2.3 \sim T_c$ and 4.0, (c) seven classes: $T \in \{1.50, 1.75, 2.05, 2.30, 2.55, 2.85, 3.10\}$.

For regression, we stay with the same CNN architecture, only adjusting the output layer. Instead of the softmax activated output layer, a single dense node is used with no activation function. This single node returns the desired numerical estimate. As loss function we now work with the mean squared error (MSE). We have used the TENSORFLOW implementation 2.4.1 for all machine learning tasks described here [23].

3. Results

Our results for the confusion matrices of the classification model for different numbers of categories are presented in Fig. 2. A "confusion matrix" represents the predicted label of the class as a function of the true one in matrix form, with an error-free prediction corresponding to a diagonal matrix. In Fig. 2(a), the categories of the phases, namely the ferromagnetic phase at $T = 1.5$ and the paramagnetic phase at $T = 4.0$, are correctly identified for all the test samples. Similarly, in Fig. 2(b), the categories are classified correctly for all the test samples, namely $T = 1.5$, $T = 2.3 \sim T_c$ and $T = 4.0$. When the number of categories is increased to seven, the confusion matrix develops off-diagonal terms, indicating "confusion", i.e. false positives and false negatives, mostly with neighbouring temperatures as shown in Fig. 2(c). Such misclassification in the $T > T_c$, paramagnetic phase is observed only among neighbouring temperatures, with a total of 39 misclassifications corresponding to 5.2% of all cases. In the $T < T_c$, ferromagnetic phase, there are 82 misclassifications corresponding to 10.9%, and only one of them is confused with a temperature that is not a neighbouring one, namely a $T = 2.05$ case is misclassified as $T = 3.1$. For $T = 2.3 \sim T_c$, only 7 out of 250 cases, i.e., 2.8% are misclassified, and only one of them ($T = 1.5$) is not a neighbouring temperature. We note that most of the confusion occurs at the highest and lowest T's. Indeed, the probability of a given configuration is controlled by its energy E. For a given T and on a finite system, the energy distribution has a certain width σ_E, and, as shown below in Fig. 4 for our case, these are consistent with the order of the confusion that we observe, at least between $T = 1.5 \leftrightarrow 1.75$ and $2.85 \leftrightarrow 3.1$, respectively. Thus, the classification according to 7 distinct temperatures reaches the physical limits of the 100×100 lattice.

We now turn to the regression models. Results for ξ, E and m are presented in Figs. 3, 4 and 5, respectively. Averages of the test data are denoted by black stars and their respective standard deviations σ_ξ, σ_E and σ_m, respectively, are indicated by the grey shaded region in the background. These standard deviations indicate the fluctuations for each quantity on the 100×100 lattice. The fluctuations are largest for ξ (Fig. 3), still significant for m around T_c (Fig. 5) and smallest but still non-negligible for E (Fig. 4). Values for one concrete configuration are
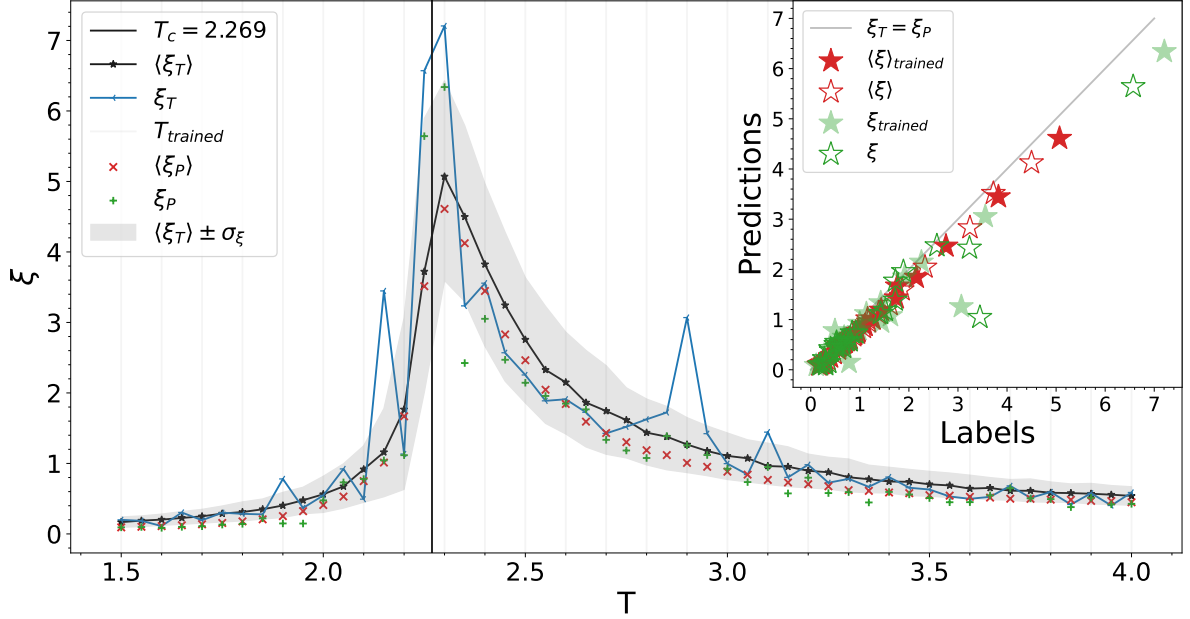
Figure 3. Correlation length ξ as a function of T. We show averages $\langle \xi_T \rangle$ ($\star$) obtained via Monte-Carlo simulations as well as predictions $\langle \xi_P \rangle$ (red $\times$) for the test data. The grey shaded region denotes the standard deviation of the training data. ξ_T (blue) and ξ_P (green $+$) are the Monte-Carlo results and ML predictions for a one specific sample, respectively. The thin black vertical line indicates the critical temperature T_c of an infinite system. Lines connecting data values are guides-to-the-eye only. The training was done only for the temperature values that are indicated by a grey vertical line in the background. The inset shows the relation of the true values and the predicted values with the diagonal giving the perfect result $\xi_T = \xi_P$.

shown in blue and as a function of T one observes indeed fluctuations on the scale of the standard deviations. We have used only every second temperature value for training; these are marked by the grey vertical lines in the background. Averages of the predictions are shown by red crosses. These lie within or close to the grey shaded region, showing that the network is indeed able to reconstruct ξ, E and m from a given configuration. This is also valid for the intermediate temperatures for which the network has not been trained, thus demonstrating the ability of the network to generalise. For more detail, we also show by green pluses the predicted values for the specific sample whose raw Monte-Carlo values were shown in blue. The fluctuations of these predictions generally follow those of the true values. Just for ξ (Fig. 3), the predictions do not fully reproduce the outliers of the raw data, i.e., ML appears to be trying to actually smooth out the sample-to-sample fluctuations. The insets of Figs. 3, 4 and 5 show the correlation between the predicted values and the true labels. Here, red symbols are for averages and green ones for the selected specific sample; closed symbols denote cases for which the CNN has been trained, open ones for temperature values where this is not the case. Again, one observes overall good performance of the network at measuring these values. Nevertheless, for ξ (inset of Fig. 3) one observes a systematic trend of the predicted values to lie below the true ones. For m, in particular intermediate values tend to be predicted slightly larger than their true value (inset of Fig. 5). We have observed other but comparable deviations in other trained CNNs and thus believe these deviations from the diagonal to represent the limits of the accuracy of the neural network.
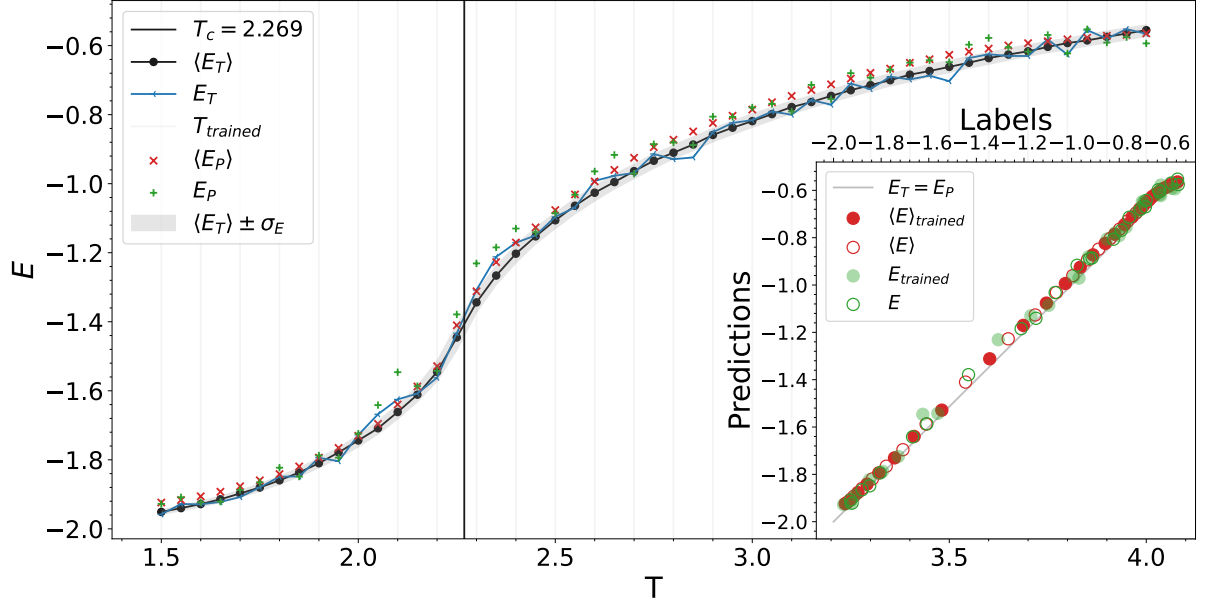
Figure 4. Energy per site E as a function of T. Averages $\langle E_T \rangle$, $\langle E_P \rangle$ and single-sample values E_T and E_P are indicated by symbols, as in Fig. 3. The inset shows the deviations from the perfect result $E_T = E_P$.

Figure 5. Magnetisation m as a function of T. Averages $\langle m_T \rangle$, $\langle m_P \rangle$ and single-sample values m_T and m_P are indicated by symbols, as in Fig. 3. The inset shows the deviations from a perfect result $m_T = m_P$.
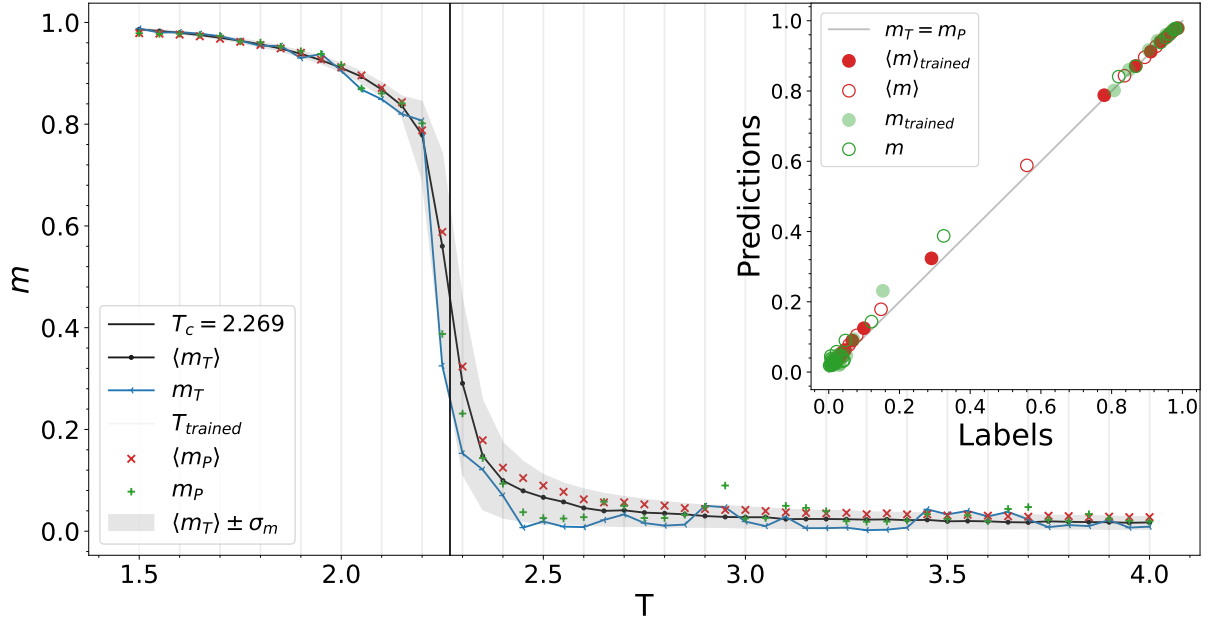
4. Discussion and Conclusions

We have shown that the specified CNN model is able to successfully classify configurations according to their temperatures until it reaches the physical limits of a 100×100 system for

our choice of the 7 categories. We also found that a regression model is able to extract the correlation length ξ, energy E and magnetisation m with an accuracy that is consistent with the fluctuations of these quantities on the 100×100 lattice. The fact that this also works for intermediate temperatures, where the network has not been trained, demonstrates the ability of the CNN to learn general features and thus to generalise. One possible next step is to investigate transfer learning for the models that are trained here. Indeed, the case investigated here is the $J_1 = 1$, $J_2 = 0$ special case of the J_1-J_2 Ising model that has recently been investigated from the machine-learning perspective [17,18], and it would be interesting to see how our neural networks trained on the case $J_2/J_1 = 0$ perform for other values of J_2/J_1.

References

[1] Domb C and Green M S 1972–1976 *Phase Transitions and Critical Phenomena* vols. 1, 2, 5B (Academic Press)

[2] Alpaydin E 2004 *Introduction to Machine Learning* 4th ed (The MIT Press) ISBN 9780262012119 URL https://mitpress.mit.edu/books/introduction-machine-learning

[3] Tanaka A and Tomiya A 2017 *J. Phys. Soc. Jpn.* **86** 063001 URL http://doi.org/10.7566/JPSJ.86.063001

[4] Walker N, Tam K M, Novak B and Jarrell M 2018 *Phys. Rev. E* **98** 053305 URL https://doi.org/10.1103/PhysRevE.98.053305

[5] Alexandrou C, Athenodorou A, Chrysostomou C and Paul S 2020 *Eur. Phys. J. B* **93** URL http://doi.org/10.1140/epjb/e2020-100506-5

[6] Ohtsuki T and Ohtsuki T 2016 *J. Phys. Soc. Jpn.* **85** 123706 URL http://doi.org/10.7566/JPSJ.85.123706

[7] Ohtsuki T and Mano T 2020 *J. Phys. Soc. Jpn.* **89** 022001 URL http://doi.org/10.7566/JPSJ.89.022001

[8] Huembeli P, Dauphin A and Wittek P 2018 *Phys. Rev. B* **97** URL http://doi.org/10.1103/PhysRevB.97.134109

[9] Dong X Y, Pollmann F and Zhang X F 2019 *Phys. Rev. B* **99** URL http://doi.org/10.1103/PhysRevB.99.121104

[10] Hu W, Singh R R P and Scalettar R T 2017 *Phys. Rev. E* **95** URL http://doi.org/10.1103/PhysRevE.95.062122

[11] Pilania G, Wang C, Jiang X, Rajasekaran S and Ramprasad R 2013 *Sci. Rep.* **3** 2810 URL https://doi.org/10.1038/srep02810

[12] Onsager L 1944 *Phys. Rev.* **65** 117–149 URL https://doi.org/10.1103/PhysRev.65.117

[13] Morningstar A and Melko R G 2018 *J. Mach. Learn. Res.* **18** 1–17 URL http://jmlr.org/papers/v18/17-527.html

[14] Walker N, Tam K M and Jarrell M 2020 *Sci. Rep.* **10** 13047 URL https://doi.org/10.1038/s41598-020-69848-5

[15] DAngelo F and Bttcher L 2020 *Phys. Rev. Research* **2** URL http://doi.org/10.1103/PhysRevResearch.2.023266

[16] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431434 URL http://doi.org/10.1038/nphys4035

[17] Corte I, Acevedo S, Arlego M and Lamas C 2021 *Comput. Mater. Sci.* **198** 110702 URL https://doi.org/10.1016/j.commatsci.2021.110702

[18] Acevedo S, Arlego M and Lamas C A 2021 *Phys. Rev. B* **103** 134422 URL https://doi.org/10.1103/PhysRevB.103.134422

[19] Adler M 2010 *Monte Carlo Simulations of the Ising Model* (Hamburg, Bedey Media GmbH) ISBN 9783954898794 URL https://www.anchor-publishing.com/document/320337

[20] Landau D P and Binder K 2014 *A Guide to Monte Carlo Simulations in Statistical Physics* 4th ed (Cambridge University Press) URL https://doi.org/10.1017/CBO9781139696463

[21] Montroll E W, Potts R B and Ward J C 1963 *J. Math. Phys.* **4** 308–322 URL https://doi.org/10.1063/1.1703955

[22] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 *J. Chem. Phys.* **21** 1087–1092 URL https://doi.org/10.1063/1.1699114

[23] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems software available from tensorflow.org URL https://www.tensorflow.org/