Rapid detection of phase transitions from Monte Carlo samples before equilibrium

Jiewei Ding[1,2], Ho-Kin Tang[3] ⋆ and Wing Chi Yu[1,2] †

1 Department of Physics, City University of Hong Kong, Kowloon, Hong Kong
2 City Univeristy of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China
3 School of Science, Harbin Institute of Technology, Shenzhen, 518055, China
⋆ denghaojian@hit.edu.cn † wingcyu@cityu.edu.hk

March 21, 2022

Abstract

We found that Bidirectional LSTM and Transformer can classify different phases of condensed matter models and determine the phase transition points by learning features in the Monte Carlo raw data before equilibrium. Our method can significantly reduce the time and computational resources required for probing phase transitions as compared to the conventional Monte Carlo simulation. We also provide evidence that the method is robust and the performance of the deep learning model is insensitive to the type of input data (we tested spin configurations of classical models and green functions of a quantum model), and it also performs well in detecting Kosterlitz–Thouless phase transitions.

Contents

---

1 Introduction

The study of phase transitions in many-body systems is one of the hottest research topics in condensed matter physics. Microscopic constituents can couple and interact with each other

in many different ways, giving rise to various phases of matter having intriguing macroscopic physical properties. Studying the transitions between different phases can give us deeper understandings of condensed matter physics especially in some non-trivial phases like topological phases where the order parameter is not readily available [1]. Monte Carlo(MC) simulation has been one of the most popular numerical techniques adapted to explore the physical properties of different phases in condensed matter models by the established Markov chain process. In addition, the large amount of data generated by MC simulations can be used for data-driven physics research, such as using machine learning to discover new physics from the data.

In the past few years, the classification of phases of matter using machine learning emerged as a prosperous research field in physics [2, 3]. Recent studies have shown that data-driven machine learning models can classify different phases by finding unknown features of condensed matter models, and further locate the phase transition points using both supervised and unsupervised learning techniques. Unsupervised learning does not require prior labelling of the data. This is particular suitable for the task of determining the number of phases in the phase diagram of new models. Examples of common unsupervised learning techniques include principal component analysis [4–6], meta-heuristic optimization [7], machine learning clustering [8] and deep autoencoder [6, 8, 9]. On the other hand, although supervised learning requires prior knowledge to label the training data, it can locate the transition points with high accuracy. Previous work has demonstrated the success of empolying supervised learning in determining the phase transition points of, for examples, the Ising models [10], the XY model [11] and the Hubbard model [12].

However, when using MC simulation, in some cases, it requires to consume extensive computing resource to generate converged set of equilibrium data. For examples, near the phase transition where critical slowing down occurs and thermal fluctuation diverges, or when we do quantum MC Simulation, a longer time is also required for the simulation to reach equilibrium due to the computational complexity of the algorithm, which sometimes accompanied with the thorny sign problem [13]. To obtain the result in the thermodynamic limit, larger systems size is needed to locate the phase transition point accurately [14]. However, when the system size increases, the time required for simulation to reach equilibrium will also increase sharply, which results in a great increase in the time cost and computing resources for generating the training data. Our method being discuss here provides a novel approach to locate the phase transition points using the input data before equilibrium, thus saving the lengthy computational time.

In this article, we tested some of the most up-to-date deep learning models, namely the Bidirectional LSTM and Transformer, which focus on analysing the time-domain data. The fact that deep learning has become a hot research area in recent years has a lot to do with its success in image classification. AlexNet proposed by Alex Krizhevsky *et. al.* won the championship in an image classification competition in 2012, and its performance far outperformed other non-deep learning algorithms [15]. Later, it was found that deep learning algorithms are also outstanding in many tasks such as image generation, image segmentation and object detection. At the same time, people began to explore the application of deep learning on tasks involving time-series data, such as text translation and speech recognition and have proposed models such as Recurrent Neural Network (RNN) [16] and Long short-term memory (LSTM) [17]. Transformer has shown great potential in the field of natural language processing in recent years, and many improved deep learning models based on Transformer prove it to be more suitable for extracting features from very long sequences than LSTM [18].

Unlike previous approaches which used a large amount of spin configurations generated from MC simulations after equilibrium is reached as the training data, here we used MC simulated spin configurations of the first m steps that are far away from the phase transition temperature T_c. We labelled the data as 0 or 1 according to the first phase or the second phase, e.g. above or below T_c. The deep learning models are then trained to classify the labelled data by learning the features of different phases. When we feed the unlabeled spin configuration near the phase transition into the well-trained machine, the machine is confused and outputs 0.5 as the classification result, from which we could determine the phase transition temperature T_c. As a benchmark, we first employed our scheme to locate the phase transition in the two-dimensional (2D) Ising model on a square lattice. We then further extend the scope of data source and the condensed matter model with non-trivial phases like the topological phase in the XY model for the deep learning analysis. Surprisingly, we find our method not only works using spin configurations from classical MC, but also the Green function obtained with quantum MC process.

In addition, we also compared the performance of the Bidirectional LSTM and Transformer with other commonly used deep learning models, namely the Fully Connected Neural Network (FCN) and Convolutional Neural Network (CNN). We found that Bidirectional LSTM and Transformer performed far better than FCN and CNN in classifying the phases with the m steps MC data. We also find that the Bidirectional LSTM and Transformer can correctly classify the phases with smaller m as compared to FCN and CNN.

The paper is organised as the followings. In Section 2, we introduce our proposed method and the deep learning models in detail. We applied our scheme to detect the phase transition in the Ising model and compare the performance using different deep learning models in Sec. 3. Section 4 presents the results when employing our method to other condensed matter models with non-trivial type of phase transitions and to the quantum Hubbard model where inputs with the Green functions generated from quantum MC is used to determine the phase transition point. A conclusion is given in Sec. 5.

2 Deep learning models for time domain analysis

Figure 1 shows a schematic illustration of our proposed learning model. We use MC method to simulate the sequential data before reaching the equilibrium state and when the system is far away from the phase transition point as the training input for the deep learning models. Taking a classical spin model as an example, the sequence is formed by randomly select a site in the system and take its spin configuration in the first m steps in the MC simulation. The deep learning model extracts features from the sequential data through LSTM block or Transformer block and performs binary classification of the phases. The trained deep learning model is then fed with data near the phase transition point to predict which phase the data belongs to. When the deep learning output a probability value of 0.5, the corresponding value of the driving parameter is the phase transition point of the system.

When one uses equilibrium spin configurations as the training data for the deep learning model, CNN can easily capture the spatial information of the configuration, such as vortexes in the XY model. However, in our task, the deep learning model needs to extract information from long sequences, and LSTM and Transformer are just suitable for such tasks [17, 18], so our deep learning model is followed by a Bidirectional LSTM Block or Transformer Block after

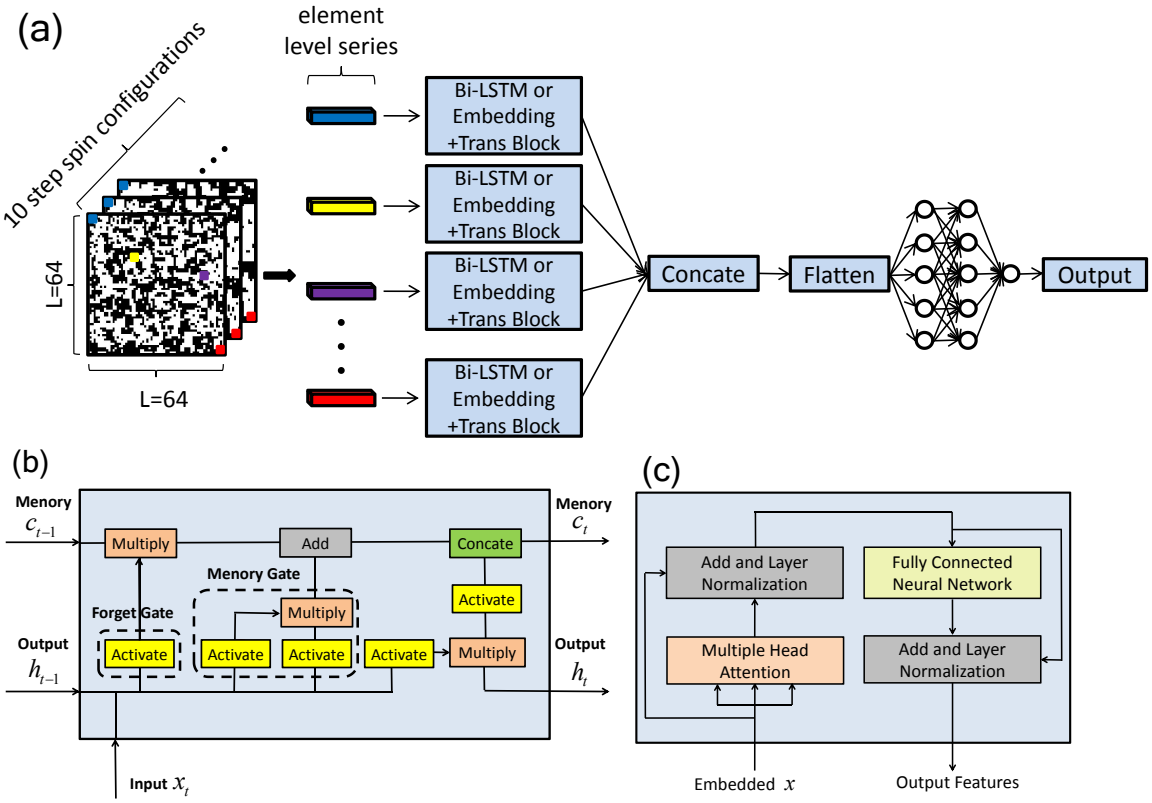

Figure 1: (a) Overall architecture of the deep learning model we used. (b) and (c) are the sub-architectures of the LSTM block and the Transformer block, respectively. The input to the model is 20 sequences obtained from MC simulation before reaching equilibrium, followed by 20 LSTM blocks or Transformer blocks to extract features from each sequence data, and finally fully connected layers are used to map the obtained features of the sequences into binary output to predict which phase the input data belongs to.

the input layer, then use FCN to process the extracted sequence information and output the probability of binary classification. The structure of the deep learning model is shown in Fig.1 (a).

Despite RNN is often used to process sequence data, its simple internal structure makes it impossible to extract long-range correlated features in the data. On the other hand, LSTM has a more complex internal structure, therefore it can capture long-range correlated features better. The architecture of LSTM is shown in Fig.1 (b). The internal structure of LSTM mainly consists of memory cells, forgetting gates, memory gates and output gates [17]. The function of memory cells (c_T) is to store important information in the input sequence (x_T) from $t = 0$ to $t = T$, the T here is the time we feed the latest data into LSTM blocks. The forget gate then judge whether there is invalid information in the memory cell according to the input sequence (x_T) at time $t = T$ and the LSTM output (h_{T-1}) at time $t = T-1$, and set the vector value of invalid information to be 0. The memory gate will judge what information needs to be added to the memory cell according to h_{T-1} and x_T. The output gate then combines the information of c_T, h_{T-1} and x_T to determine the output of LSTM at time $t = T$.

Sometimes LSTM may fail in extracting long-range related features of the sequence because the sequence's elements has to be read one by one. This will overflow the memory

cell information if the sequence is too long or leads to zeros in the gradient during backward propagation in the learning process. On the other hand, Transformer can extract all ranges of correlations in the sequence since it reads the elements of the entire sequence at one time. The correlation between all the elements of the sequence is then learnt through a self-attention layer and more advanced features can be extracted. The architecture of Transformer is shown in Fig.1(c) [18]. The self-attention layer is a key part of the Transformer. In principle, it can extract infinitely long-range correlated features.

It is worth noting that before the sequence data is fed into the Transformer Block, each element of the sequence needs to be mapped to a higher or lower dimension through the embedding layer, and the positional information of each element in the sequence is also added to the mapped information. However, we found that position embedding is not necessary for our task. The details are discussed in Appendix A.

In the following, we applied the above machine learning scheme to the classical and quantum many-body systems. For classical spin models with $L \times L$ sites, we used the spin configurations of each MC step as input data, which are $L \times L$ matrices. For the Hubbard model with N sites, we used the Green functions as input, which are $N \times N$ matrices. To let the deep learning model only focus on the information of the input data in the time dimension instead of the pattern in the space, we did not use the entire matrix as input. For each input sample, we randomly choose 20 elements in the matrix and pick the simulation result of these matrix elements in m MC steps to form a tensor with of the shape $(20, m)$, as shown in Fig. 1(a). We only selected 20 elements because we found that selecting more elements did not improve the performance of the deep learning model.

3 Machine performance in phase transition detection from data before equilibrium

The Ising model on a square lattice is a pedagogical model capturing the physics of a classical phase transition in condensed matter. The model describes spin-1/2 particles in a lattice system where each spin interacts with its nearest neighbours. The Hamiltonian of the Ising model is given by

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \tag{1}$$

where $\sigma_i \in$ {-1,1} denotes spin-down and spin-up and the sum is over all the nearest neighbouring spins. J characterises the coupling strength between two nearest spins and is taken to be $J = 1$ in the following.

In an infinite size square lattice, the system exhibits a phase transition between the paramagnetic phase and the ferromagnetic phase at a temperature of $T_c = 2/\log(1+\sqrt{2}) \approx 2.269$ [19]. When the temperature is close to zero, the interactions between the spins dominates and all the spins tend to align in the same direction. The system is in the ferromagnetic phase with an average magnetization $M \in$ {-1,1}. When the temperature is much higher than T_c, the direction of the spins becomes random due to the strong thermal perturbation. The average magnetization of the system is approximately equal to zero and the system is in the paramagnetic phase. Given a randomly initialized spin configuration, one can simulate how this spin configuration reaches one of the two phases at different temperatures in equilibrium step by

step using MC method.

Spin configurations of the Ising model at equilibrium have very obvious difference between ferromagnetic phase and paramagnetic phase. Previous work has shown that after supervised training of a convolutional neural network using such equilibrium data, the neural network can easily locate the phase transition temperature of the Ising model [10]. However, for the Ising model, a randomly initialized spin configuration typically requires 1000 MC steps to reach its equilibrium configuration at a given temperature. In more complex models, the time and computational resources required for the MC simulation to reach equilibrium will even be more. In the following, we explored whether the neural network can also accurately determine the phase transition temperature if it is trained with spin configurations from only the first few MC steps that are far before equilibrium is reached.

We used MC simulated data in the temperature range $T \in ([0,1] \cup [4,5])$ as the training set. The system size of the Ising model is $L = 256$. Five hundreds raw samples were generated in each temperature range, therefore we have a total of 1000 raw samples. As mentioned in Sec. 2, each input data of our deep learning model comes from the spin configurations of 20 randomly selected sites. For each raw sample, we can repeatedly select 20 sites randomly to obtain multiple training samples. Altogether, we obtained 10000 training samples from the MC raw samples.

Figure 2 shows the output of the trained neural networks when fed with testing data from full temperature range. We tested the performance of FCN, CNN, Bi-LSTM and Transformer with different MC steps $m \in \{10, 20, 30, 40\}$. From Fig. 2 (a) and (b) respectively, we found that Bi-LSTM and Transformer can accurately predict the transition temperature of model to be $T_c \approx 2.269$ as determined by an output value of 0.5 (indicated by the horizontal dashed line in the figures) from the machine. On the other hand, CNN can only predict that T_c is around 2.269 and the results are also more sensitive to the number of MC steps (Fig. 2(c)). The performance of FCN is even worse. The predicted T_c is significantly larger than the expected value (Fig. 2(c)). Besides, FCN is also less confident in classifying the two phases as one can see for test samples far away from T_c, its outputs do not reach 0 or 1. It is worth noting that in order to fairly evaluate the performance of each deep learning model, we controlled the number of parameters of the model to be in the same order of magnitude.

Figure 3 shows the predicted transition temperature by the four deep learning models using various number of MC steps m as the input training data. Specifically, for each value of m, we trained the model 10 times. After each training, in order to find the predicted value of T_c, we took the data whose model output value is in the range [0.1, 0.9] and performed a linear regression on those data. The temperature corresponds to a fitted value of 0.5 from the linear regression is taken as the T_c predicted by the model. From the figure, we find that except for FCN, the predicted value of T_c from all the models converges to the expected value 2.269 within a MC step of 20. Although the prediction of CNN is close to 2.269, the convergence with respect to the MC steps is relatively slower than Bi-LSTM and Transformer. The error in T_c predicted by CNN is also much larger than that of Bi-LSTM and Transformer, which shows CNN is less stable than the other two models on this task, while Bi-LSTM and Transformer performs similarly well.

We are not surprised by the poor performance of FCN as FCN has a natural disadvantage when dealing with sequence data. Suppose there is an FCN model with sequential element

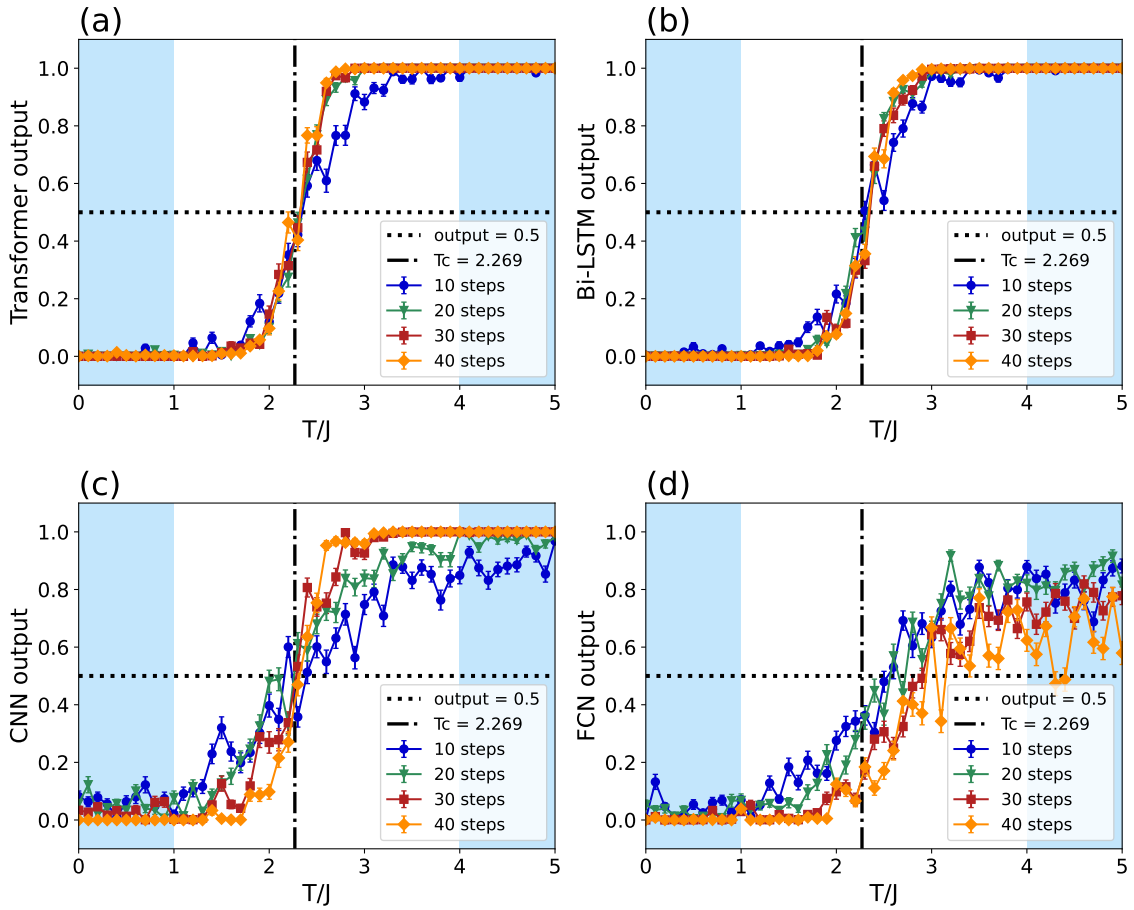

Figure 2: Testing result for Ising model on square lattice using (a) transformer, (b) Bi-LSTM, (c) CNN, (d) FCN. The light blue area represents the temperature range of our training data. Vertical black dashed line indicates the theoretically predicted transition temperature. The machine predicted transition temperature is given by the intersect of the horizontal dashed line and the curves. Transformer and Bi-LSTM accurately predicted the phase transition temperature $T_c \approx 2.269$. The performance of CNN is relatively unstable. Even if the input data contains more MC steps, the predicted T_c by CNN still has a relatively large error. FCN performed the worst, not only failed to predict the phase transition point, but also for the test samples within the training temperature range. FCN was unable to correctly classify the samples with high confidence.

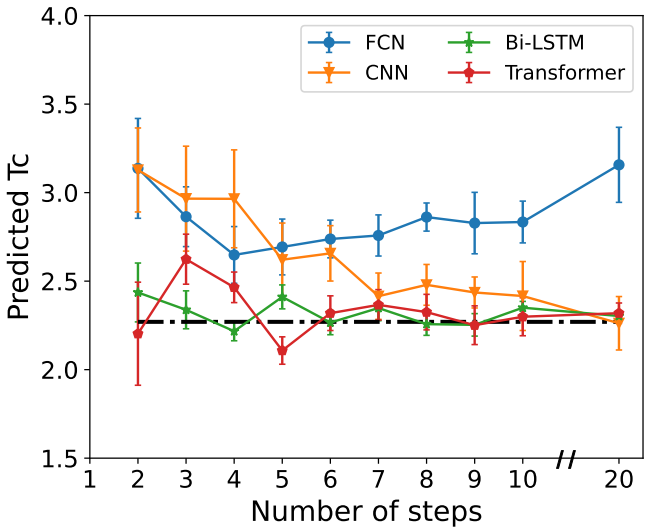

Figure 3: Machine predicted value of the phase transition temperature T_c in the classical Ising model on a square lattice using different MC steps. The lines joining the data points are just guides for the eyes. Transformer and Bi-LSTM accurately predict the phase transition point in fewer MC steps, while CNN predict the phase transition point in 20 steps but with a larger error. The T_c predicted by FCN is sufficiently larger than the theoretical value (horizontal black dashed line).

input, the output z of the first neuron in the first hidden layer is given by

$$z = \sum_i w_i x_i, \tag{2}$$

where w_i is the weight of each input element x_i. After the FCN model is trained, its weights w_i will be fixed, and these weights will only depend on the position of the elements in the sequence. In other words, the weight of the first element of all input sequences will be the same. However, in our training data, the information contained in the elements at the same position in different sequences as well as their importance can be different in general.

The CNN model can solve the above problems by increasing the kernel number, while LSTM and Transformer convert the weight into an input-related function through a complex model architecture. Specifically, the hidden layer outputs of LSTM and Transforemr become

$$z = \sum_i f(x_i) x_i, \tag{3}$$

where $f(x_i)$ is different for different deep learning models. In our task, $f(x_i)$ can help the deep learning model to better locate the key input data, and extract features from this key information to better complete the binary classification. In fact, the Transformer and Bi-LSTM has very similar performance as we observed above. This is because the input sequences' length, which is $m \in [2, 40]$, is within the storing capacity of the memory cell in Bi-LSTM and thus the full sequential information can be retained, making the advantage of Transformer not obvious.

4 Generalizability to other models and raw data source in time domain

We also applied our scheme to more complicated condensed matter models to test the robustness of our method. The models are the Ising model on a honeycomb lattice and on a triangle lattice, the XY model on a square lattice and the quantum Hubbard model on a honeycomb lattice. The system size of all classical models is $N = 256 \times 256$, while that of the Hubbard model is $N = 12 \times 12$.

The Hamiltonian of the XY model is given by

$$H = -J \sum_{\langle i,j \rangle} \cos(\sigma_i - \sigma_j), \tag{4}$$

where J is the interaction strength between two nearest spins, $\sigma_i \in (0, 2\pi]$ represents the spin angle on the lattice's plane at the site i. Unlike the Ising model discussed in the previous section, the phase transition occurring at $T_c = 0.89$ in the XY model is Kosterlitz–Thouless (KT) type [20]. Above T_c, the spin correlation decays exponentially while it shows a power-law decaying behavior at temperatures below T_c. Vortexes and antivortexes with winding numbers equal to 1 and -1 respectively are formed in the system [21]. At low temperatures, the vortex and antivortex are tight to each other and tend to annihilate to minimize system's energy. The phase transition is associated with the unbinding of the vortex-antivortex pairs at the critical temperatures when the temperature increases. Intuitively, we shall expect these non-local spatial feature needs to be obtained through CNN using the entire spin configuration of the system. However, as discussed in Sec. 2, the input data used in our method is element level series in the simulation time domain. It will be interesting to test whether the deep learning model can still extract relevant information about the phases and accurately determine the phase transition that is of a topological character.

The Hubbard model, on the other hand, is a quantum model whose Hamiltonian is given by

$$H = -t \sum_{\langle i,j \rangle} (c^\dagger_{i\uparrow} c_{j\uparrow} + c^\dagger_{i\downarrow} c_{j\downarrow} + h.c.) + U \sum_i n_{i\uparrow} n_{i\downarrow} - \mu \sum_i (n_{i\uparrow} + n_{i\downarrow}), \tag{5}$$

where $c^\dagger_{i\sigma}(c_{j\sigma})$is the creation (annihilation) fermion operator of spin $\sigma = \{\uparrow, \downarrow\}$ at site i, $n_{i\uparrow} = c^\dagger_{i\uparrow} c_{i\uparrow}$ is the number operator, t is the nearest neighbor hopping amplitude, U characterises the Coulomb interaction strength between two electrons of opposite spins at the same site, μ is the chemical potential. In this article, we considered $\mu = 0$, which corresponds to the case of half-filling. For $U > 0$, the system exhibit a quantum phase transitions from paramagnetic phase to anti-ferromagnetic phase at $U_c \approx 3.9$ [22]. Since the model is quantum in nature, input data is generated from quantum MC in which the raw output are the Green functions. We would like to investigate whether our method can be applied to this type of source data. Existing work has proven that Green functions after the equilibrium is reached in the quantum MC simulation are suitable input data for deep learning models to learn about the phase transition point [12]. As a data source for the deep learning model, the physical interpretation of Green function is very different from the spin configuration. For example, the spin configuration represents the spatial feature of the system in the real space, while the Green function represents the correlation between the creation and annihilation of the electron from different sites in the Hubbard model. During the training, we use the time-series of randomly picked element from the data source. In the case of the spin configuration, the transnational symmetries in the model helps preserve the universality of the training data. But

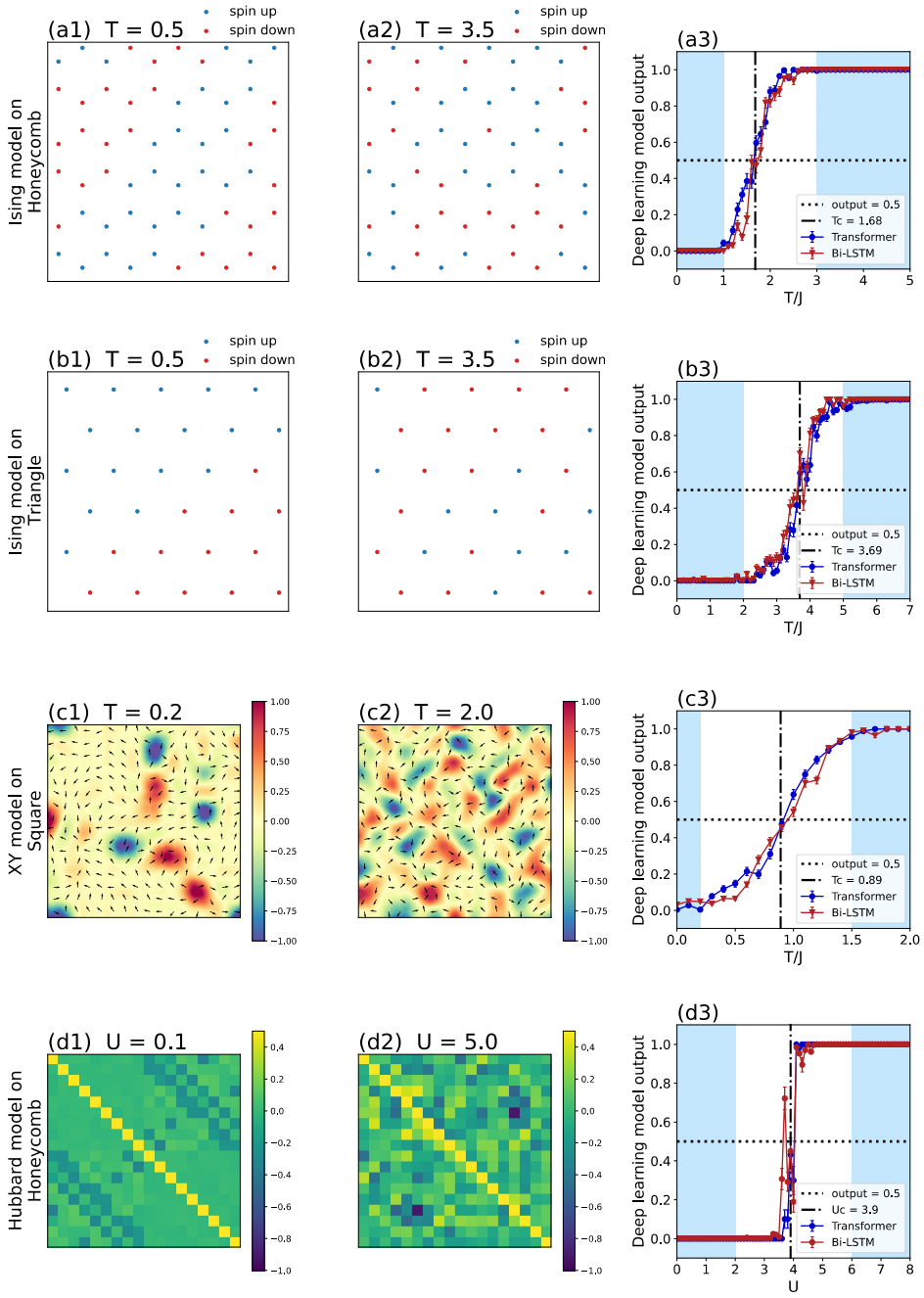

Figure 4: Figure (a1)-(a2), (b1)-(b2) and (c1)-(c2) are the spin configurations at the tenth MC steps in the Ising model on honeycomb lattice, Ising model on triangle lattice and XY model, respectively, away from the phase transition temperature. Figure (d1)-(d2) are Green functions of the Hubbard model away from the quantum phase transition point. In the Ising models, the blue dots represent up spin and the red dots represent down spin. In the XY model, arrows represent spin orientation and color scale represents the magnitude of the local winding number. In the Hubbard model, the color scale represents magnitude of Green function matrix element. Figure (a3), (b3), (c3) and (d3) shows the output of the Transformer and Bi-LSTM in the corresponding condensed matter models respectively. Transformer and Bi-LSTM in the three classical models perform similarly to the Ising model on a square lattice. However, their performance in the Hubbard model is significantly different. The output value of the deep learning model is unstable and it shows a step-like abrupt change in the close vicinity of the transition point.

this is not the case for the Green function, the time variation of correlator highly depends on the which element in the Green function we picked. This may be an adverse factor when we train the model using the Green function.

Figure 4 shows the results of the above mentioned models using Bi-LSTM and Transformer. Our deep learning model performs well on all the three classical spin models (Fig. 4(a)-(c)). In the region far from the phase transition point, the deep learning model can distinguish the two phases with a high level of confidence, while near the phase transition point, the deep learning model has a very smooth change due to confusion. When the output of the deep learning model is equal to 0.5, its corresponding temperature T is approximately consistent with those obtained in the literature. The performance of our proposed deep learning model in the Ising models are in line with our expectations. Since the only difference among these models is in the geometry of their lattice and only the nearest-neighbor ferromagnetic interactions are presence in the models, one shall expect the same type of phase transition between the FM and PM to take place. However, to our surprise, the deep learning model also performs well on the XY model where the input data we used to train the model contains no spatial information. Even the model cannot capture any information about topological quantities in the data, the model can still classify the two phases well. This suggests the sequence information present in the MC steps before equilibrium may be related to the topological character of the XY model at equilibrium. Unfortunately, it is hard to interpret what features has the deep learning models learnt from the sequences due to the complexity of the network itself.

In the case of the Hubbard model (Fig. 4(d)), the input data to the deep learning model is a sequence of length 100, which is about 0.01% of the quantum Monte Carlo(QMC) steps required to reach equilibrium. We found that the probability values output by the deep learning models do not change gradually as in the classical models near the phase transition point, but exhibit a significant increase from 0 to 1 as temperature increases, as shown in Fig. 4 (d3). Moreover, the probability values output by the deep learning model is unstable in the vicinity of the quantum phase transition. For example, samples in the paramagnetic phase are sometimes classified as an anti-ferromagnetic phase with high confidence. We can be sure that this difference is not because each matrix element in the Green function contains much less information than the spin configuration of the classical models. As we can see from the figure, most samples near the transition are correctly classified with very high confidence. If the information in the input data is not enough, the deep learning model will output a probability that is much greater than 0 and much less than 1 due to confusion. Instead, the large fluctuation of the output in the close vicinity of the transition point may attribute to the fact that the Green functions fluctuate a lot (as compared to the classical model) due to the quantum fluctuations when they are far from equilibrium, resulting in the misclassification of some test samples [23, 24].

5 Conclusion

In this work, we show that deep learning models can rapidly classify phases in various condensed matter models using MC data before equilibrium and locate the critical points with high accuracy. Among the deep learning models we have investigated, the performance of Bi-LSTM and Transformer are found to be the best in probing phase transition points. Both models are mainly constructed to extract feature of time-domain data. Unlike the CNN counterparts, their success relies on the efficient features extraction from the long sequential input

data, which have not been explored in the mission of phase transition detection in previous studies.

We also investigated the generalizability of our method to the Ising model on a honeycomb lattice and a triangle lattice, the XY model, which undergoes a KT transition, and the Hubbard model where the input training data comes from the Green function generated by quantum MC. Bi-LSTM and Transformer can determine the critical points of these condensed matter models accurately. The results evidence that our proposed method is robust in detecting various types of phase transitions in condensed matter models and in using different types of source data.

We would like to remark that the data generated by the Markov Chain Monte Carlo simulation in this study contains no time order. Bi-LSTM and Transformer do not extract features in time order, but only information on the sequence elements. For future works, it will be worthwhile to apply the method we proposed here to real time-ordered simulation data, such as studying the dynamics of disordered systems or glass transitions using molecular dynamics simulation data.

Acknowledgements

We acknowledge financial support from National Natural Science Foundation of China (Grant No. 12005179), Harbin Institute of Technology Shenzhen (Grant No. ZX20210478, Grant No. X20220001) and City University of Hong Kong (Grant No. 9610438).

References

[1] X.-G. Wen, *Quantum field theory of many-body systems: from the origin of sound to an origin of light and electrons*, Oxford University Press, doi:10.1063/1.2155760 (2004).

[2] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto and L. Zdeborová, *Machine learning and the physical sciences*, Rev. Mod. Phys. **91**, 45002 (2019), doi:10.1103/RevModPhys.91.045002.

[3] J. Carrasquilla, *Machine learning for quantum matter*, Adv. Phys.:X **5**, 1797528 (2020), doi:10.1080/23746149.2020.1797528.

[4] L. Wang, *Discovering phase transitions with unsupervised learning*, Phys. Rev. B **94**, 195105 (2016), doi:10.1103/PhysRevB.94.195105.

[5] C. Wang and H. Zhai, *Machine learning of frustrated classical spin models. i. principal component analysis*, Phys. Rev. B **96**, 144432 (2017), doi:10.1103/PhysRevB.96.144432.

[6] W. Hu, R. R. Singh and R. T. Scalettar, *Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination*, Phys. Rev. E **95**, 062122 (2017), doi:10.1103/PhysRevE.95.062122.

[7] H.-K. Tang, M. A. Marashli and W. C. Yu, *Unveiling quantum phase transitions by fidelity mapping*, Phys. Rev. B **104**, 075142 (2021), doi:10.1103/PhysRevB.104.075142.

[8] K. Ch'ng, N. Vazquez and E. Khatami, *Unsupervised machine learning account of magnetic transitions in the hubbard model*, Phys. Rev. E **97**, 013306 (2018), doi:10.1103/PhysRevE.97.013306.

[9] K. Kottmann, P. Huembeli, M. Lewenstein and A. Acín, *Unsupervised phase discovery with deep anomaly detection*, Phys. Rev. Lett. **125**, 170603 (2020), doi:10.1103/PhysRevLett.125.170603.

[10] J. Carrasquilla and R. G. Melko, *Machine learning phases of matter*, Nat. Phys. **13**, 431 (2017), doi:10.1038/nphys4035.

[11] M. J. Beach, A. Golubeva and R. G. Melko, *Machine learning vortices at the kosterlitz-thouless transition*, Phys. Rev. B **97**, 045207 (2018), doi:10.1103/PhysRevB.97.045207.

[12] P. Broecker, J. Carrasquilla, R. G. Melko and S. Trebst, *Machine learning quantum phases of matter beyond the fermion sign problem*, Sci. Rep. **7**, 8823 (2017), doi:10.1038/s41598-017-09098-0.

[13] E. Loh Jr, J. Gubernatis, R. Scalettar, S. White, D. Scalapino and R. Sugar, *Sign problem in the numerical simulation of many-electron systems*, Phys. Rev. B **41**, 9301 (1990), doi:10.1103/PhysRevB.41.9301.

[14] E. Brezin and J. Zinn-Justin, *Finite size effects in phase transitions*, Nucl. Phys. B **257**, 867 (1985), doi:10.1016/0550-3213(85)90379-7.

[15] A. Krizhevsky, I. Sutskever and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, NIPS **25**, 1097 (2012), doi:10.1145/3065386.

[16] W. Zaremba, I. Sutskever and O. Vinyals, *Recurrent neural network regularization*, arXiv:1409.2329 (2014), doi:10.48550/arXiv.1409.2329.

[17] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural. Comput. **9**, 1735 (1997), doi:10.1162/neco.1997.9.8.1735.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Attention is all you need*, Adv. Neural Inf. Process. Syst. **30**, 5998–6008 (2017), doi:10.48550/arXiv.1706.03762.

[19] L. Onsager, *Crystal statistics. i. a two-dimensional model with an order-disorder transition*, Phys. Rev. **65**, 117 (1944), doi:10.1103/PhysRev.65.117.

[20] P. Olsson, *Monte carlo analysis of the two-dimensional xy model. ii. comparison with the kosterlitz renormalization-group equations*, Phys. Rev. B **52**, 4526 (1995), doi:10.1103/PhysRevB.52.4526.

[21] P. M. Chaikin, T. C. Lubensky and T. A. Witten, *Principles of condensed matter physics*, vol. 10, Cambridge university press Cambridge, doi:10.1017/CBO9780511813467 (1995).

[22] Y. Otsuka, S. Yunoki and S. Sorella, *Universal quantum criticality in the metal-insulator transition of two-dimensional interacting dirac electrons*, Phys. Rev. X **6**, 011029 (2016), doi:10.1103/PhysRevX.6.011029.

[23] R. R. d. Santos, *Introduction to quantum monte carlo simulations for fermionic systems*, Braz. J. Phys. **33**, 36 (2003), doi:10.1590/S0103-97332003000100003.

[24] F. Assaad and H. Evertz, *World-line and determinantal quantum monte carlo methods for spins, phonons and electrons*, In *Computational Many-Particle Physics*, pp. 277–356. Springer, doi:10.1007/978-3-540-74686-7_10 (2008).

A Analysis of model performance

A.1 Shuffle time dimension input data

We experimented that not using positional embedding or shuffling the data in the time dimension will not change the performance of the deep learning model. This is because the data is real time independent. MC simulation is a hidden Markov chain process, the action of each step during simulation is independent of previous steps. Thus, there is no time-related feature in the input sequence.

Figure 5 shows the outputs of the deep learning models for the Ising model on a square lattice if we shuffle the input data in the simulation time dimension randomly. We can see that the performance of Transformer and Bi-LSTM are not affected by the shuffling.

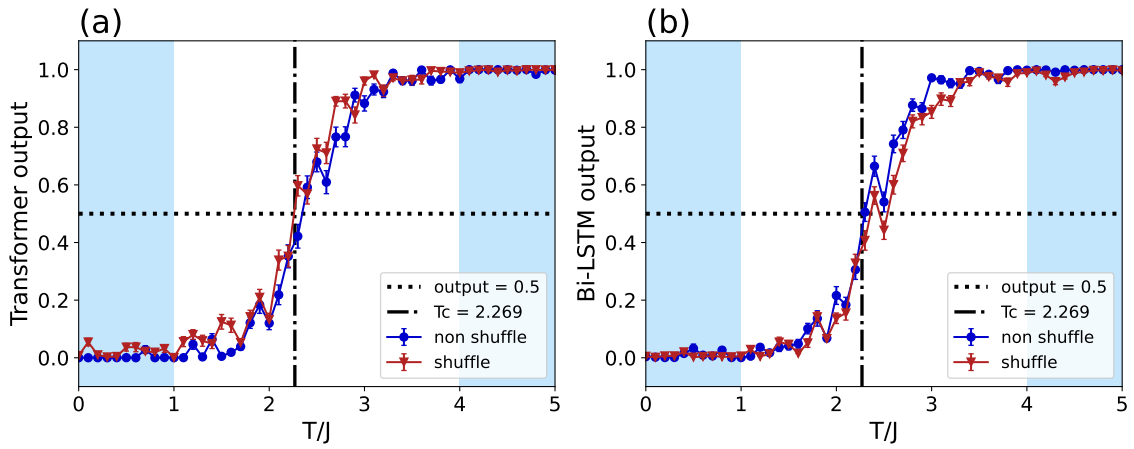

Figure 5: (a) and (b) show the performance of Transformer and Bi-LSTM on Ising model on square lattice, respectively. Here MC steps $m = 10$ is used. The light blue area represents the temperature range of our training data. Shuffling the data in the simulation time dimension does not affect the performance of Transformer and Bi-LSTM.

A.2 Variation of training range

We also tested whether our method is sensitive to the range of the training data. We used the first 10 Monte Carlo steps of the Ising model to feed the Bi-LSTM and Tranaformer. As shown in Fig. 6, Bi-LSTM and Tranaformer perform very similarly in the four different sets of training ranges.

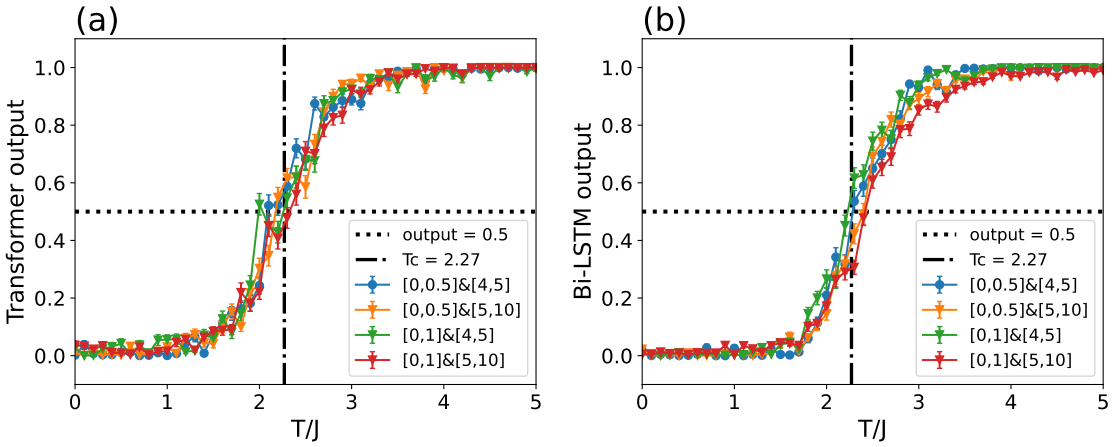

Figure 6: (a) and (b) show the output of Transformer and Bi-LSTM on Ising model on square lattice, respectively. The legend indicates the range of temperatures in which the training samples are taken from. It can been seen that the performance of the machine is not sensitive to the training range.