Classifying Quantum States of Matter with Machine Learning

Julian A. Calder

Department of Physics, Middlebury College, Middlebury, VT

Project report for PHYS 0704

May 21, 2024

Phase transitions are a highly studied area within numerous scientific fields. While it is often easy to identify the physical phase of a material as a solid, liquid, or gas, it proves more challenging to identify magnetic phases by observation alone. This paper expands on previous research which has demonstrated that machine learning models can prove to be effective tools in classifying magnetic phases, implementing binary image classification methods to identify simulated spin configuration data as corresponding to the ferromagnetic or paramagnetic phases. First, we validate previous results using machine learning methodologies on the Ising model of a magnetic material before expanding to the study of the XY model, which exhibits a less easily-identifiable phase transition. We find that fairly simple neural network architectures are able to classify simulated magnetic spin configuration data with a high degree of accuracy, strengthening the case for further study of the implementation of machine learning methods in scientific use-cases such as this one.

Date Accepted: ______________________________

I Introduction

When identifying phase of matter one often uses properties such as temperature, mass, volume, and other easily measurable or observable properties. Phase diagrams are commonly used for a wide range of compounds to describe when they may transition between a solid, liquid or gas as a function of temperature and pressure. These phase transitions typically correspond to a change in the arrangement in the compound's underlying atoms, taking, for example, the grid-like arrangement of water molecules as it transitions from a liquid to solid at 0 °C.

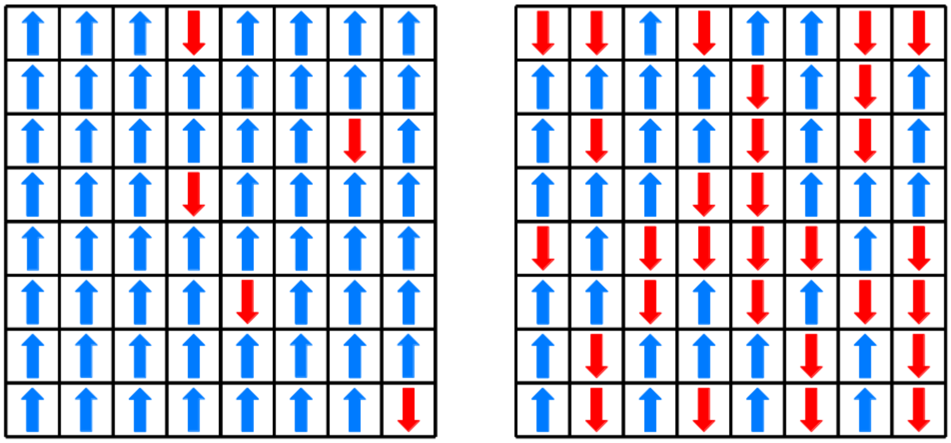
Phase transitions also occur in the domain of magnetic materials as the individual magnetic moments of atoms within a material may rearrange when exposed to changes in temperature or external magnetic fields. One can imagine a magnetic material as a series of atoms each possessing a magnetic moment. The overall energy of a given material corresponds to the degree to which the magnetic moments (also called spins) of the atoms align with each other. When the majority of the spins within a material are aligned, this is referred to as the ferromagnetic phase. At higher temperatures, the spins within the material become randomized in what is known as the paramagnetic phase. Similar to the known temperatures at which water freezes and boils, there too exist distinct temperatures at which magnetic materials transition from ferromagnetic to paramagnetic phase.

This paper aims to validate the results of and expand on previous research using machine learning tools to classify magnetic phase from the spin configurations of a simulated magnetic system. We begin by exploring the theory behind both the Ising and XY models of magnetic materials and explaining why the magnetization cannot function as an order parameter for both systems. We then move to validate earlier results [1, 2] demonstrating the effectiveness of a neural network when performing binary classification on simulated magnetic spin configurations for the Ising model. We then extend this neural network classification to the XY model of magnetic materials, exploring which neural network architecture can most effectively identify magnetic phase based on images of spin configurations and use this to predict the temperature at which our system transitions from ferromagnetic to paramagnetic. [3].

II Theory

II. (i) The Ising Model

The simplest mathematical representation of a magnetic material is known as the Ising model. One can visualize the two-dimensional Ising model as a lattice of atoms, where each atom can be either spin up or spin down. When the majority of spins are either all spin up or all spin down, our material is classified as ferromagnetic. Conversely, when spins are a random combination of up and down our material is classified as paramagnetic. Fig. 1 shows visualizations of the Ising model for a material in the ferromagnetic and paramagnetic phase, respectively.



(a) Ferromagnetic phase. Low energy, high magnetization

(b) Paramagnetic phase. High energy, low magnetization

Fig. 1. Spin configurations for two-dimensional square lattice Ising model. The ferromagnetic phase (a) consists of mostly aligned spins and has low energy and high magnetization (as described by equations (1) and (3) respectively). Conversely, the paramagnetic phase (b) consists of randomly aligned spins and higher energy, with a magnetization value approaching 0.

The energy of the Ising model is determined by the interaction between neighboring spins and is related to the temperature of the system being represented. The total energy E is given by

$$E = -J \sum_{\langle i,j \rangle} S_i S_j, \tag{1}$$

where $S = \pm 1$ represents our atom being spin up or spin down and J is a positive constant with units of energy that describes the strength of the interaction between adjacent atoms. Possible energy values for pairs of neighboring spins are thus

$$E(\uparrow\uparrow) = E(\downarrow\downarrow) = -J \text{ for aligned spins}$$
$$E(\uparrow\downarrow) = E(\downarrow\uparrow) = +J \text{ for misaligned spins.}$$

This summation is taken over all pairs of neighboring points within the lattice.

The Boltzmann factor tells us that the probability of a certain energy state is proportional to

$$P(E) \propto e^{-E/k_b T}, \tag{2}$$

where E is the energy of the system given by equation 1, k_b is the Boltzmann constant, and T is the temperature of the system. This equation tells us that at low temperature, systems with low energy are favored which correspond to lattices in which the spins are mostly aligned as in Fig. 1a. As temperature increases, the probability of each energy state approaches the same value and each state becomes equally likely resulting in random spin configurations as in Fig. 1b.

The magnetization can be thought of as the degree to which spins are aligned, ranging in value from 0 (spins randomized) to 1 (spins fully aligned). In the Ising model, magnetization can be calculated by averaging the spins across the entire lattice

$$M = \frac{1}{N}\sum_i S_i, \tag{3}$$

where $S_i = \pm 1$ corresponds to the spin of a given point within our lattice, and N is the total number of points within our lattice. A plot of the magnetization as a function of temperature for a variety of lattice dimensions ranging from $L = 4 \times 4$ to $L = 64 \times 64$ is shown in Fig. 2. We see a dramatic drop in magnetization at around $T \approx 2.3\ k_b T/J$, which corresponds to the transition from the ferromagnetic to paramagnetic phase. This value, known as the critical temperature T_c, is given by [4]

$$T_c = \frac{2}{ln(1+\sqrt{2})} \approx 2.269\ J/k_b. \tag{4}$$

Note that as lattice dimension L increases, the magnetization drops increasingly abruptly around the critical temperature T_c.

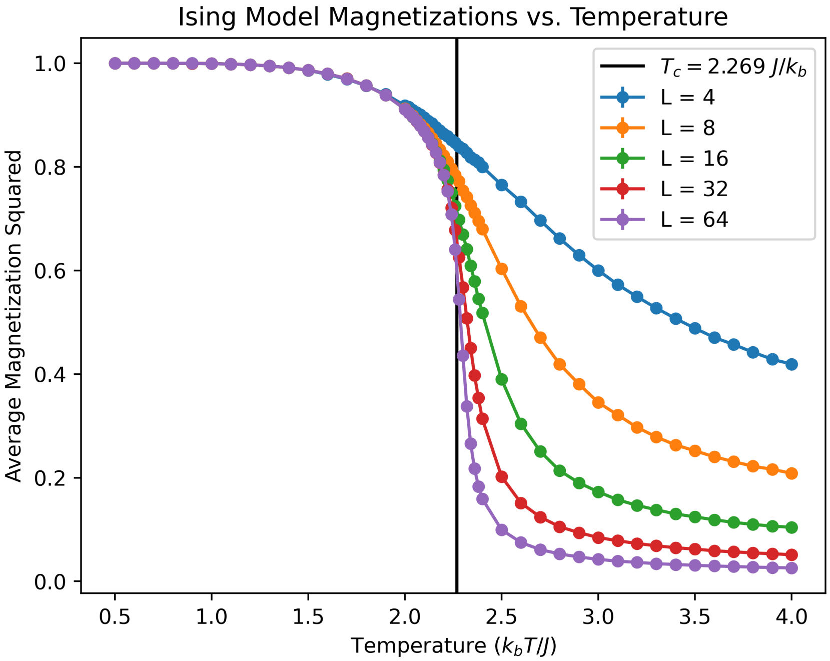


Fig. 2. Average magnetization squared $\langle M^2 \rangle$ as a function of temperature for an $L = 4 \times 4$ to $L = 64 \times 64$ two-dimensional square Ising lattice, as calculated with equation (3). The data used in this calculation was produced by a Monte Carlo simulation as described in section III. (i). The vertical line in this plot corresponds to the value of the critical temperature, T_c. Note that this plot does include standard error estimates but that the values are too small to be visible.

The Ising model is widely used across many disciplines, though it is well studied for this application representing magnetic materials. The Ising model can also be easily simulated computationally, as is explained in Section III. (i). However, this model substantially oversimplifies the physics behind atomic interactions within magnetic materials such that it is worth exploring alternative models of magnetic materials to more adequately represent the underlying physics.

II. (ii) The XY Model

The XY model, much like the Ising model, is a mathematical model which can be used to represent spins within a magnetic material. The important difference between the Ising and XY models, though, is that while spins in the Ising model exist only as binary value of up or down, spins in the XY model can be any orientation in the xy plane. This adds additional complexity compared to the Ising model when describing the interaction between neighboring spins, though many of the calculations regarding system energy and magnetization are similar.

As with the Ising model, aligned spins in the XY model correspond to a ferromagnetic material while randomized spins correspond to a paramagnetic material as shown in Fig. 3. The probability of a given energy state in the XY model is still dictated by the Boltzmann factor in equation (2). The total energy of a system according to the XY model is

$$E = -J \sum_{\langle i,j \rangle} \vec{S_i} \cdot \vec{S_j} = -J \sum_{\langle i,j \rangle} cos(\theta_i - \theta_j), \tag{5}$$

where $\vec{S_i}$ and $\vec{S_j}$ are the unit vectors corresponding to our spin angles of neighboring points within the lattice and J is a positive constant with units of energy that describes the strength of interaction between adjacent points in the lattice.

One important difference in the possible energy values between the Ising and XY models is that the energy values in the Ising model are discrete, depending on whether a given point within the lattice is spin up or spin down. This makes it more difficult to disorder the system. The XY model, in contrast, allows for slight perturbations to individual spins which allows for smaller fluctuations in energy.

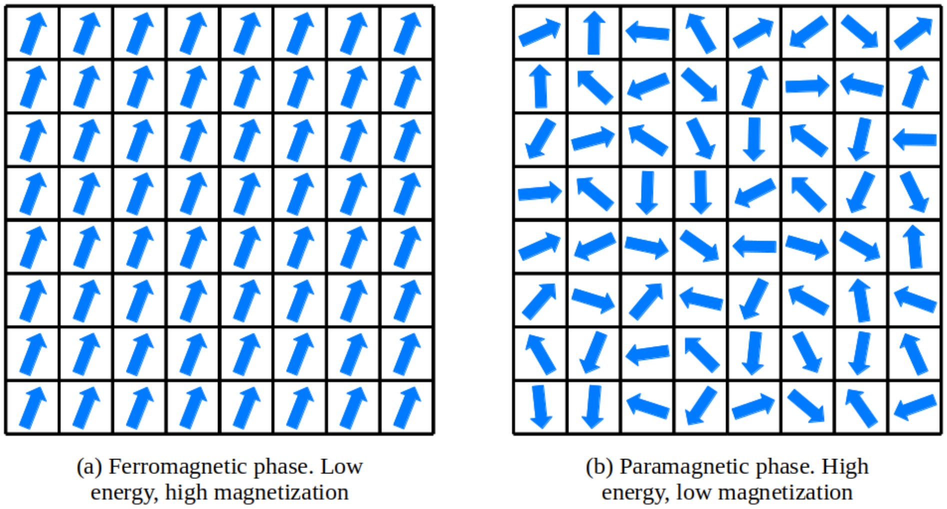
The calculation of magnetization for the XY model involves separately adding the x and y components of the spin vectors together

$$\langle M_x \rangle = \frac{1}{N} \sum_i cos(\theta_i) \text{ and } \langle M_y \rangle = \frac{1}{N} \sum_i sin(\theta_i), \tag{6}$$

where θ_i is the angle of our spin vector. The calculation for average magnetization squared $\langle M^2 \rangle$ follows from equation (6)

$$\langle M^2 \rangle = \langle M_x^2 \rangle + \langle M_y^2 \rangle \, . \tag{7}$$

A plot of magnetization squared as a function of temperature for a variety of lattice dimensions ranging from $L = 4 \times 4$ to $L = 64 \times 64$ is shown in Fig. 4. The XY model also experiences


(a) Ferromagnetic phase. Low energy, high magnetization

(b) Paramagnetic phase. High energy, low magnetization

Fig. 3. Spin configurations for a two-dimensional square lattice XY model where $L = 8 \times 8$. The ferromagnetic phase (a) consists of mostly aligned spins and has low energy and high magnetization (as described by equations (5) and (7) respectively). Conversely, the paramagnetic phase (b) consists of randomly aligned spins and higher energy, with a magnetization value approaching 0.

an abrupt ferromagnetic-paramagnetic phase transition as described by the Kosterlitz-Thouless transition at a numerically-determined critical temperature $T_c \approx 0.8816$ [5]. As lattice size increases magnetization values begin to drop more abruptly such that for very large system sizes the magnetization approaches 0 for all temperatures.

It is important to note that unlike magnetization in the Ising model, the XY model magnetization does not exhibit the same dramatic drop around T_c and is not well suited for identifying the magnetic phase of our simulated magnetic material. Fig. 4 shows that as system size increases the average magnetization curve shifts down and to the left, and for infinite system size the magnetization would approach 0 at all temperatures. While there are other values relevant to the XY model such as vorticity and helicity modulus which can be used to identify the critical temperature, these require additional computation to determine and would introduce potentially unnecessary complexity. Previous research has demonstrated that neural networks can be effective tools for classifying magnetic materials using the Ising model [1, 2], and this paper seeks to explore the application of neural networks to the more complex XY model.

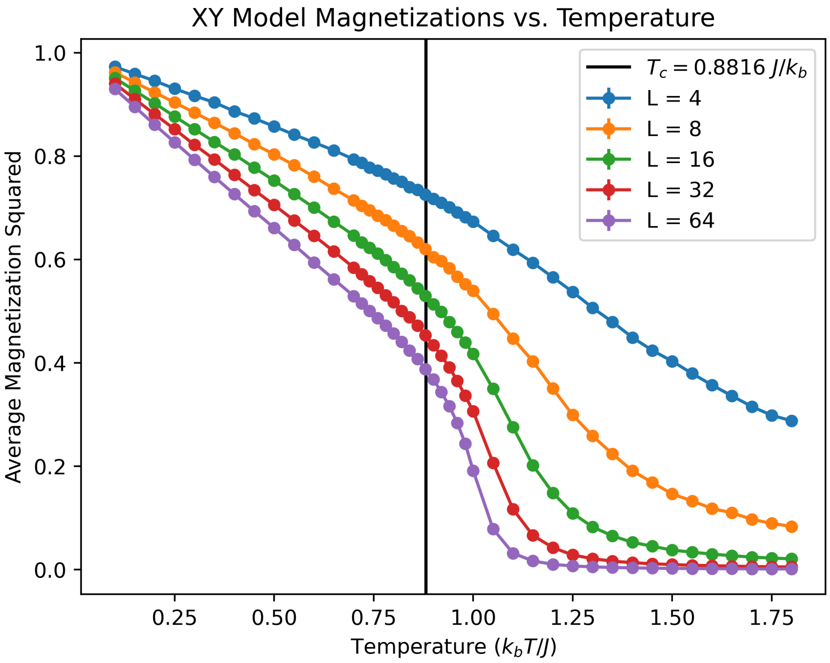


Fig. 4. Average magnetization squared $\langle M^2 \rangle$ as a function of temperature for an $L = 4 \times 4$ to $L = 64 \times 64$ two-dimensional square XY-model lattice, as calculated with equations (6) and (7). The data used in this calculation was produced by a Monte Carlo simulation as described in section III. (i). The vertical line in this plot corresponds to the value of the critical temperature, T_c. Note that this plot does include standard error estimates but that the values are too small to be visible.

II. (iii) Overview of Machine Learning

Machine learning refers to the broad field of using computer models and algorithms to learn, interpret and make decisions based on data without explicit guidance. One can imagine a machine learning model as a black-box function which takes in arbitrary data as an input and outputs a result based on how that data is interpreted, which can be visualized in Fig. 5. There exist several different subcategories within machine learning based on how the input data is configured and the degree to which the process of ”learning” is supervised by humans. This research implements a supervised machine learning architecture known as a neural network which is well suited for the classification problem which we are presented with.

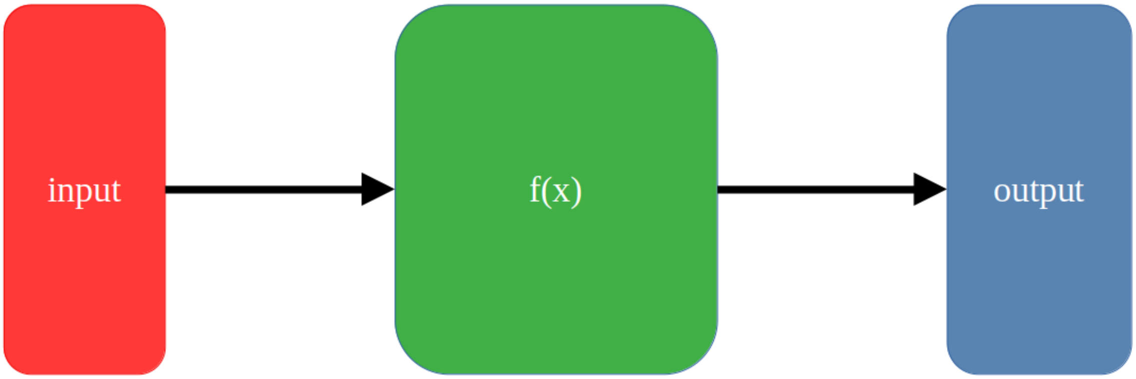


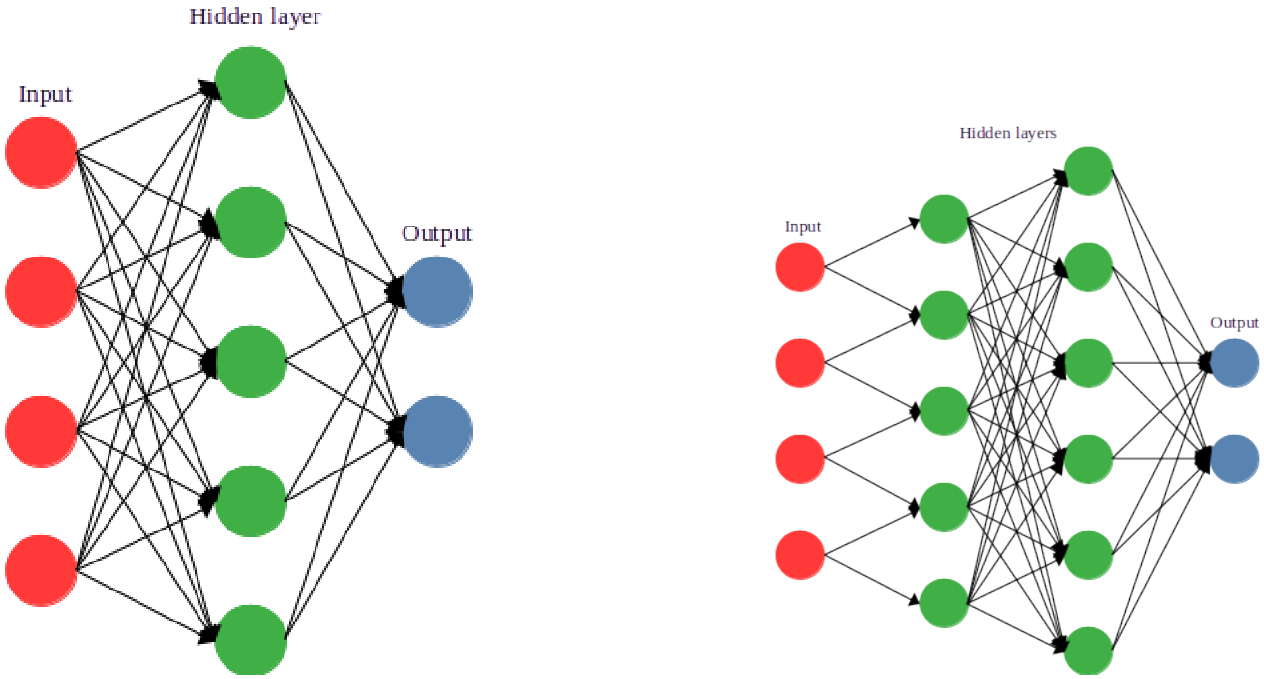
Fig. 5. Input data is passed into our machine learning function, which performs a series of mathematical operations to adjust its parameters to minimize the deviation between its output and the desired output.

II. (iv) Neural Networks

A neural network is a kind of supervised machine learning model which is designed to process and interpret data in a way analogous to neurons within the brain. Input data is passed into a series of nodes, which are then connected to an arbitrary number of additional ”hidden layers” to perform loss minimization before ultimately being passed into a number of output nodes. In the case of a binary classification problem such as the one explored in this project, that number of output nodes is two. Two common neural network architectures are shown in Fig. 6.

A fully connected neural network is one in which each node in a given layer is connected to all of the nodes in the following layer, as shown in Fig. 6a. A convolutional neural network (CNN), on the other hand, consists of layers of nodes which are not necessarily connected to all nodes in the subsequent layer (as shown in Fig. 6b). This can be advantageous when dealing with input data which may depend on locality, as a CNN may be better equipped to identify geographic clusters within the input data.

A neural network is trained by taking in a series of input data points and their corresponding labels. In this research we pass in an array of numerical values corresponding to the spin configurations generated for both the Ising and XY models, along with the appropriate label of whether a given spin corresponds to a system which is above or below the critical temperature T_c. Neural networks are typically trained over a series of epochs, in which a subset of the training data is processed and used to update the parameter weights to improve the accuracy



(a) Fully connected neural network

(b) Convolutional neural network

Fig. 6. Two common neural network architectures. In a fully connected network (a), every node is connected to every node in the adjacent layer such that locality is not preserved. In a convolutional network (b), input nodes are connected only to subsequent pairs of nodes, preserving information about locality as opposed to the fully-connected network.

of the model.

In the case of a binary classification such as this one, the output of the neural network is a pair of values corresponding to the probability of the input data corresponding to each of the two possible labels (in this case, whether the system is above or below T_c. This probability value can also be interpreted as a level of "confidence" that the neural network has when classifying a given input in either of the two ways. Together, these two probability values add to 1.

III Methods

This research was divided into two primary phases. First, we validated previous results which demonstrated the success of using neural networks to classify spin configurations for the Ising model [1, 2]. After confirming these results, we explored implementing various neural network architectures for the XY model of magnetic materials to determine which architecture has the greatest accuracy when predicting the value of T_c [3]. An overview of the experimental workflow is below:

1. Create parameter file and use C++ simulation to generate spin configuration data.
2. Iterate through spin configuration file extracting raw spin values and organizing by temperature.
3. Generate magnetization plots calculated using equations (3) and (7).
4. Convert data into appropriate form to input into neural network and begin training process.
5. Test model on new randomized spin configuration data to determine accuracy of model.
6. Generate plots of confidence vs. temperature for each network architecture and system size.
7. Perform curve fitting to identify the temperature at which confidence values intersect, corresponding to the neural network's estimate for T_c.
8. Interpolate T_c estimates across multiple system sizes to determine estimate for T_c at infinite system size.

III. (i) Data Generation

Spin configuration data was generated using a preexisting Markov Chain Monte Carlo simulation [6] which uses statistical sampling to produce a spin for every point within the simulated lattice with probabilities dictated by equation (2). Each array corresponding to the spin configuration of a given lattice was then given a label of above or below the critical temperature T_c depending on the temperature which was input into the Monte Carlo simulation.

The dataset for the Ising model consisted of a range of temperatures from $T = 0.5\ J/k_b$ to $T = 4.0\ J/k_b$ at increments of $T = 0.1\ J/k_b$, except near the critical temperature where data was produced at increments of $T = 0.02\ J/k_b$. A similar process was employed for the XY model, where data was generated for the temperature range $T = 0.1\ J/k_b$ to $T = 1.8\ J/k_b$ at intervals of $T = 0.05\ J/k_b$. Near the critical temperature, increments of $T = 0.02\ J/k_b$ were again used.

In both the Ising and XY models, 5000 statistically independent spin configurations were generated for each of these temperatures to ensure sufficient data for training and testing, with 4000 spin configurations being used for training and the remaining 1000 being used for testing at each temperature. Simulation data was produced for lattice sizes ranging from $L = 4 \times 4$ up to $L = 64 \times 64$ to allow for the study of how model accuracy varied with system size.

III. (ii) Neural Network Training

The TensorFlow machine learning python package was used to generate the neural networks used throughout this research [7]. Training on data for the Ising model was kept brief, aiming only to validate previous results. We employed a simple three-layer model, the first consisting of L^2 input nodes to create a one-to-one correspondence between each point in our lattice and each node in our model. Next, we utilized a 128-node fully connected layer followed by a two-node output layer corresponding to whether the model thought that the spin configuration represented a system that was above or below the critical temperature T_c.

We used four different neural network architectures for the XY model, described below as follows:

1. **Fully connected - cos:** A fully connected neural network consisting of L^2 input nodes, each taking in a single float value representing the cosine value of the spin vector. Input nodes are fully connected to 128 hidden nodes, which then connect to two output nodes.
2. **Fully connected - sin:** A fully connected neural network consisting of L^2 input nodes, each taking in a single float value representing the sine value of the spin vector. Input nodes are fully connected to 128 hidden nodes, which then connect to two output nodes.
3. **Fully connected - combined:** A fully connected neural network consisting of $2L^2$ input nodes, where one set of L^2 nodes takes in a single float value representing the cosine value of the spin vector and the next L^2 taking a single float representing the sine value of the spin vector. Input nodes are fully connected to 128 hidden nodes, which then connect to two output nodes.
4. **CNN - 2 channel:** A two-channel convolutional neural network (CNN) consisting of L^2 input nodes, each structured to take in two float values corresponding to the cosine and

sine values of the spin vector. These input nodes are then connected to pairs of subsequent hidden nodes, which are then fully connected to 128 nodes and then to the pair of output nodes. See Fig. 6b for a visualization of this neural network.

In all cases, training was done over 20 epochs. Due to limitations in computing power, we were unable to perform neural network analysis on the $L = 64 \times 64$ XY model data.

III. (iii) Estimating T_c

After the models were trained, we were able to test them by passing in previously unseen spin configurations which were then classified by the models and compared their outputs with the correct classification label. For each spin configuration, the neural network output values of confidence above and below the critical temperature which were grouped and averaged by system temperature and used to produce a plot of confidence as a function of temperature (an example of which is shown in Fig. 9. We used the `scipy.curve_fit` function to perform a linear fit on the confidence plots for values near T_c from which we could solve for the temperature at which the confidence values equalled 50%. This value was then interpreted as the estimate for T_c, from which we could produce a plot of estimated T_c values as a function of neural network architecture for various system sizes. After determining T_c estimates at a variety of different system sizes for each neural network architecture, we could plot these estimates and find a line of best fit which allowed us to determine a value of T_c at infinite system size.

IV Results

We saw encouraging results across both the Ising and XY models regardless of specific neural network architecture and system size. Fig 7 shows a single plot of confidence vs. temperature for $L = 64 \times 64$ for the two-dimensional Ising model. Data Figs. 8 and 9 show plots of confidence versus temperature for system sizes ranging from $L = 4 \times 4$ up to $L = 64 \times 64$ and $L = 32 \times 32$ for the Ising and XY models, respectively. The Ising model was trained on a fully-connected network as described in section III. (ii), while the XY model utilized a two-channel convolutional neural network (CNN), also described in section III. (ii). At temperatures far from T_c, the models correctly identified the simulated data as above or below T_c with a high degree of confidence. In both the Ising and XY models, the neural network experiences a loss of confidence at values quite close to the critical temperature of each model ($T_c = 2.269\ J/k_b$ for Ising and $T_c = 0.8816\ J/k_b$ for XY). We also see that the confidence values for both models are much lower for temperatures away from T_c when $L = 4$, indicating that the neural networks struggle to make accurate classifications with limited input data in a small system size.

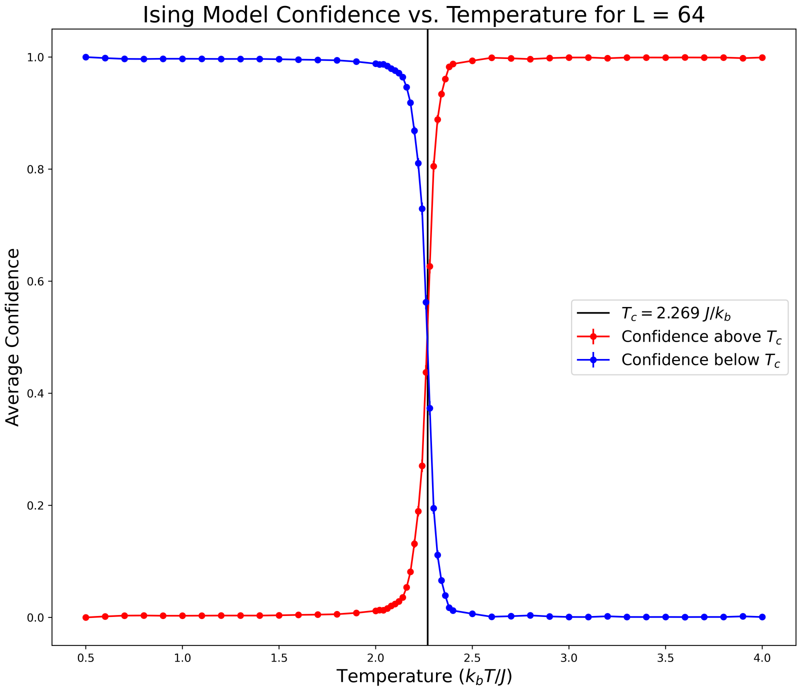


Fig. 7. Plot of classification confidence versus temperature for the two-dimensional Ising model for $L = 64 \times 64$. Note that plot does contain standard error estimates but these values are too small to be visible.

Fig. 10 shows that for larger system sizes in the XY model, we see a sharper drop in the confidence for the two-channel CNN than when utilizing a fully-connected network train on cos values, sin values, or a combination of both. This confirms results seen by [3], indicating that a convolutional neural network is better suited for classification with the XY model than a fully-connected neural network.

We see a similar advantage of the CNN when estimating T_c from our confidence plots as described in section III. (iii). Our CNN produced an estimated critical temperature value of $T_{c,2channel} = 0.886 \pm 0.005\ J/k_b$ for an infinite system size which, when accounting for uncertainty,

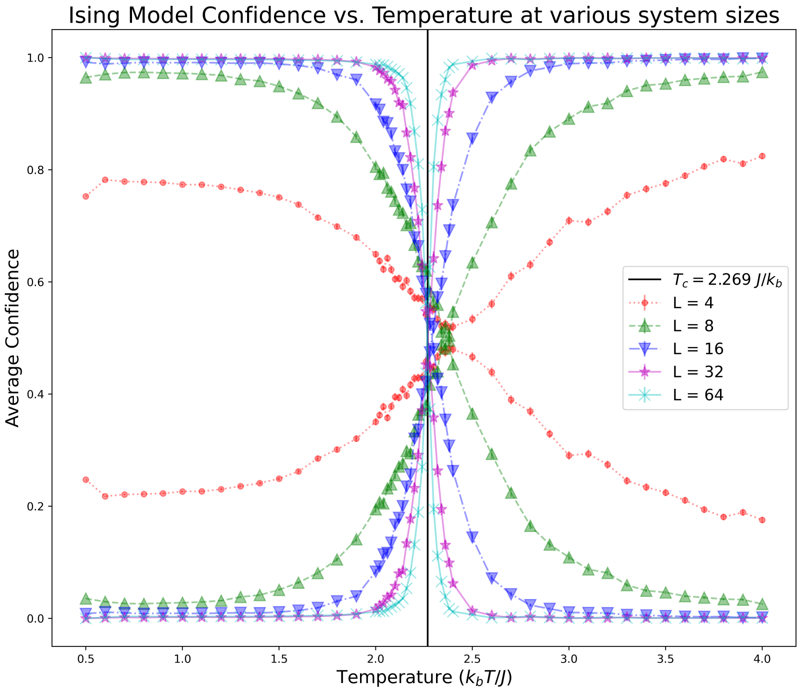


Fig. 8. Plot of classification confidence versus temperature for the two-dimensional Ising model at lattice sizes from $L = 4 \times 4$ to $L = 64 \times 64$. Model is trained on a fully connected neural network as shown in Fig. 6a. Notice how curve sharpness increases with system size and how the point at which the confidence curves cross aligns closely with the critical temperature, plotted in black. Note that plot does contain standard error estimates but these values are too small to be visible.

agrees with the accepted value of $T_c = 0.8816\ J/k_b$. There is also a general trend towards smaller uncertainties as system size increases as confidence plots become sharper and the intersection regions grow more linear.

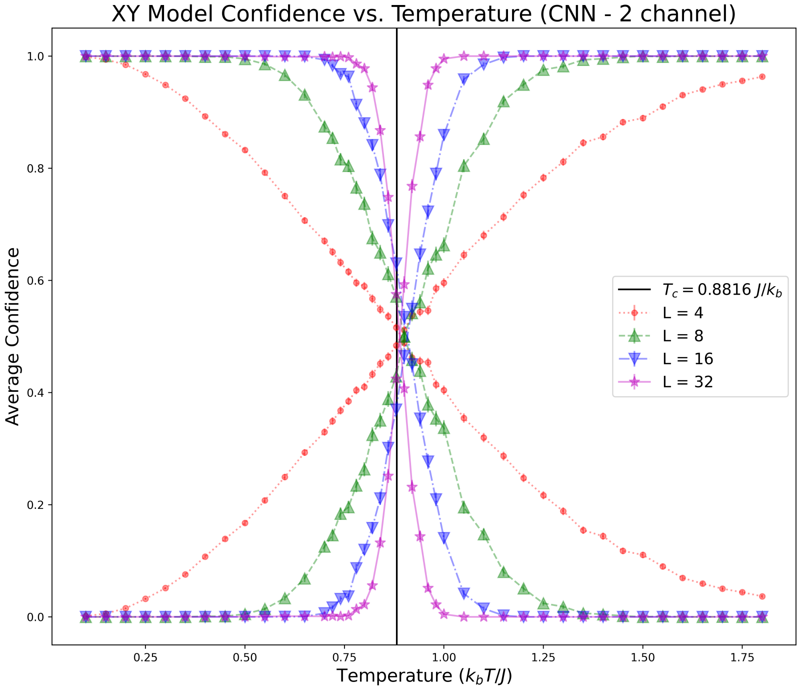


Fig. 9. Plot of classification confidence versus temperature for the two-dimensional XY model at lattice sizes from $L = 4 \times 4$ to $L = 32 \times 32$. Model is trained on a two-channel convolutional neural network as shown in Fig. 6b. Notice how curve sharpness increases with system size and how the point at which the confidence curves cross aligns closely with the critical temperature, plotted in black. Note that plot does contain standard error estimates but these values are too small to be visible.

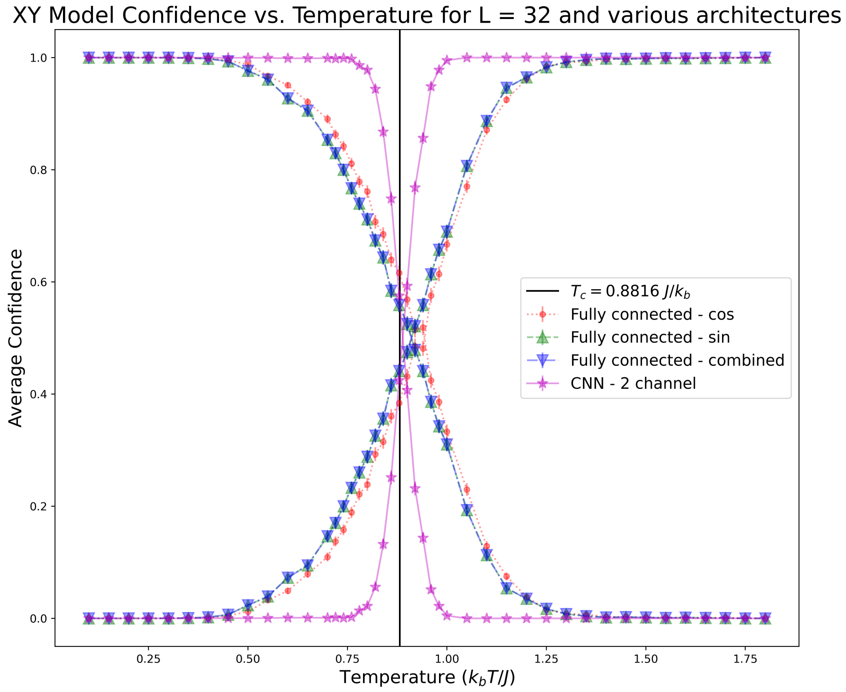


Fig. 10. Neural network confidence as a function of temperature for a simulated $L = 32 \times 32$ lattice across various network architectures. Red, green and blue curves were trained on a fully-connected neural network with a 128-node hidden layer, while the purple curve corresponds to a convolutional neural network trained on cos and sin values passed in together.

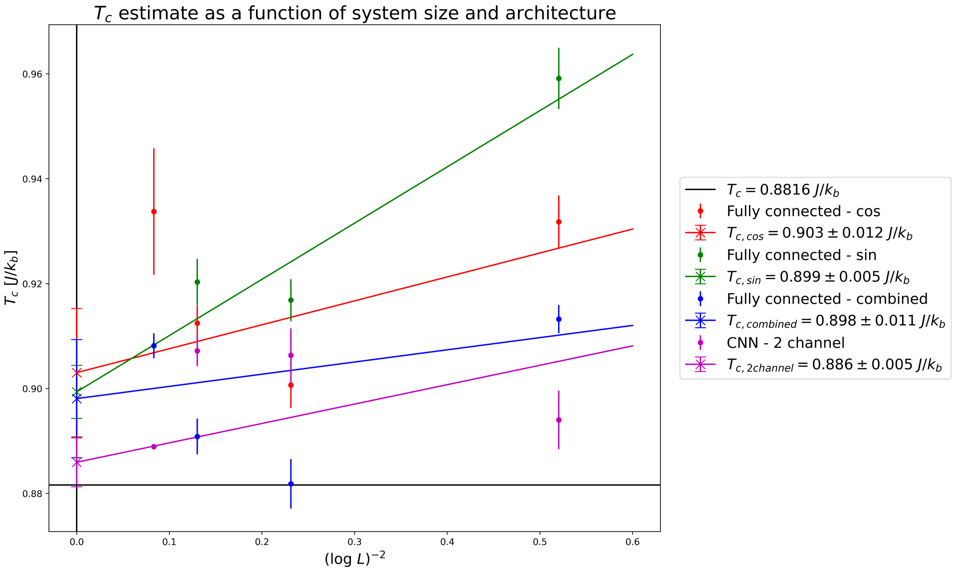


Fig. 11. T_c estimates as function of system size for various neural network architectures. Linear fit of T_c estimates for each system size was interpolated to find the y-intercept, representing the value of T_c at an infinitely large system size.

V Discussion and Conclusion

This research aimed to explore the implementation of neural networks in studying and classifying magnetic phase based on numerical representations of spin configurations. Our results have demonstrated that even simple neural networks can be very effective at classifying magnetic phase and indirectly identifying the critical temperature T_c for different models of magnetic materials, confirming prior results by Carrasquilla et al. [1] and Beach et al [3]. We confirmed that the CNN provided the most accurate estimate for T_c compared to the fully-connected network architectures, indicating that this architecture is perhaps better suited for more complex magnetic models. Future research should explore implementing more sophisticated CNN architectures which can more accurately account for the complexity of the atomic interactions within the XY model, as well as extending this methodology to other models of magnetic materials.

VI Acknowledgements

Thank you to Chris Herdman for all of the support throughout the duration of this project. Thank you also to the Middlebury Physics department for providing the workstation computers on which the computation for this research was performed.

Appendix

Code

The code used for generating the data, training the neural networks and processing the results is available at `https://github.com/julian-calder/phys-704`.

Monte Carlo Simulation Parameters

Both the Ising and XY model simulations shared the following parameters:

- **Seed:** 0
- **Number of Warm-up Sweeps:** 10,000
- **Sweeps per Measurement:** 100
- **Measurements per Bin:** 10
- **Number of Bins:** 5,000
- **Lattice Parameters:** D = 2, L = [4, 4], [8, 8], [16, 16] and [32, 32] (to run simulation on different system sizes)
- **Model Parameters:** J = 1, h = 0

The Ising model used temperatures from 0.5 to 4.0 at intervals of 0.1, except for the range of 2.0 to 2.4 which was spaced every 0.02. The XY model, on the other hand, used temperature values from 0.1 to 1.8 at intervals of 0.05, except for the range 0.7 to 1.0 which was spaced every 0.02. The ”Spin Dimension” parameter was 1 for the Ising model, and 2 for XY.

Neural Network Parameters

The connections between nodes within a neural network each possess their own weights which are adjusted with every epoch that the model is trained such that it is tuned to minimize the deviation between the model output and the expected output. Each node within the neural network also contains its own activation function which is used to introduce non-linearity into the network as it decides whether or not to incorporate the weight of a given neuron.

Both Ising and XY models were trained with the `relu` activation function for the fully-connected hidden layer, though the XY model CNN architecture replaced the original L^2 input nodes with a `Conv2D` input layer with a 3×3 convolution window which also utilized the `relu` activation function. Both models were compiled with the `adam` optimizer with the `SparseCategoricalCrossentropy` loss function.

Error Calculation

Uncertainty in estimates for neural network confidence vs. temperature were calculated with the standard error of the mean:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{N}}, \tag{8}$$

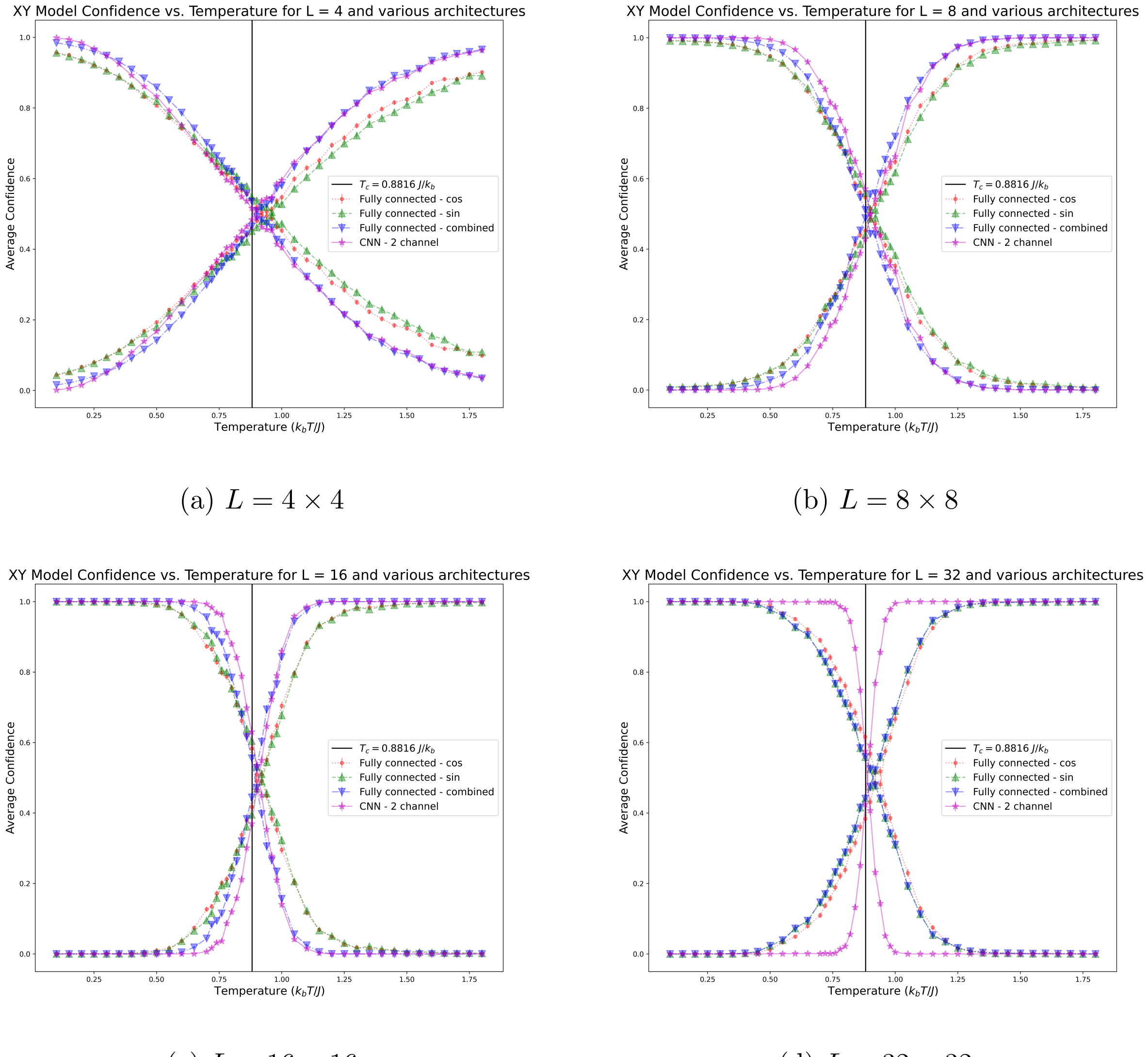
where σ is the standard deviation of the data and N is the number of samples. Uncertainty in T_c values were determined by the variance in parameters A and B (from equation $y = A + Bx$) which were output by `scipy.curve_fit` and propagated as described in [8].

Plots of Confidence vs. Temperature

Plots of confidence versus temperature were generated for all system sizes and neural network architectures, as shown in Figs. 12 and 13. All plots contain standard error calculated using equation (8), though values are too small to be visible.

T_c Estimation From Confidence Plots

The process of estimating T_c from the plots of confidence vs. temperature involved producing linear fit functions in the data region where the confidence plots crossed, as shown in Fig. 14. Setting the fit functions equal to each other allowed us to determine an estimate for the intersection point, representing the estimated value for T_c. This process was repeated for each neural network architecture and system size. These values were then plotted as shown in Fig. 11, and a linear fit on those estimates let us interpolate a value for T_c at $L \approx \infty$.



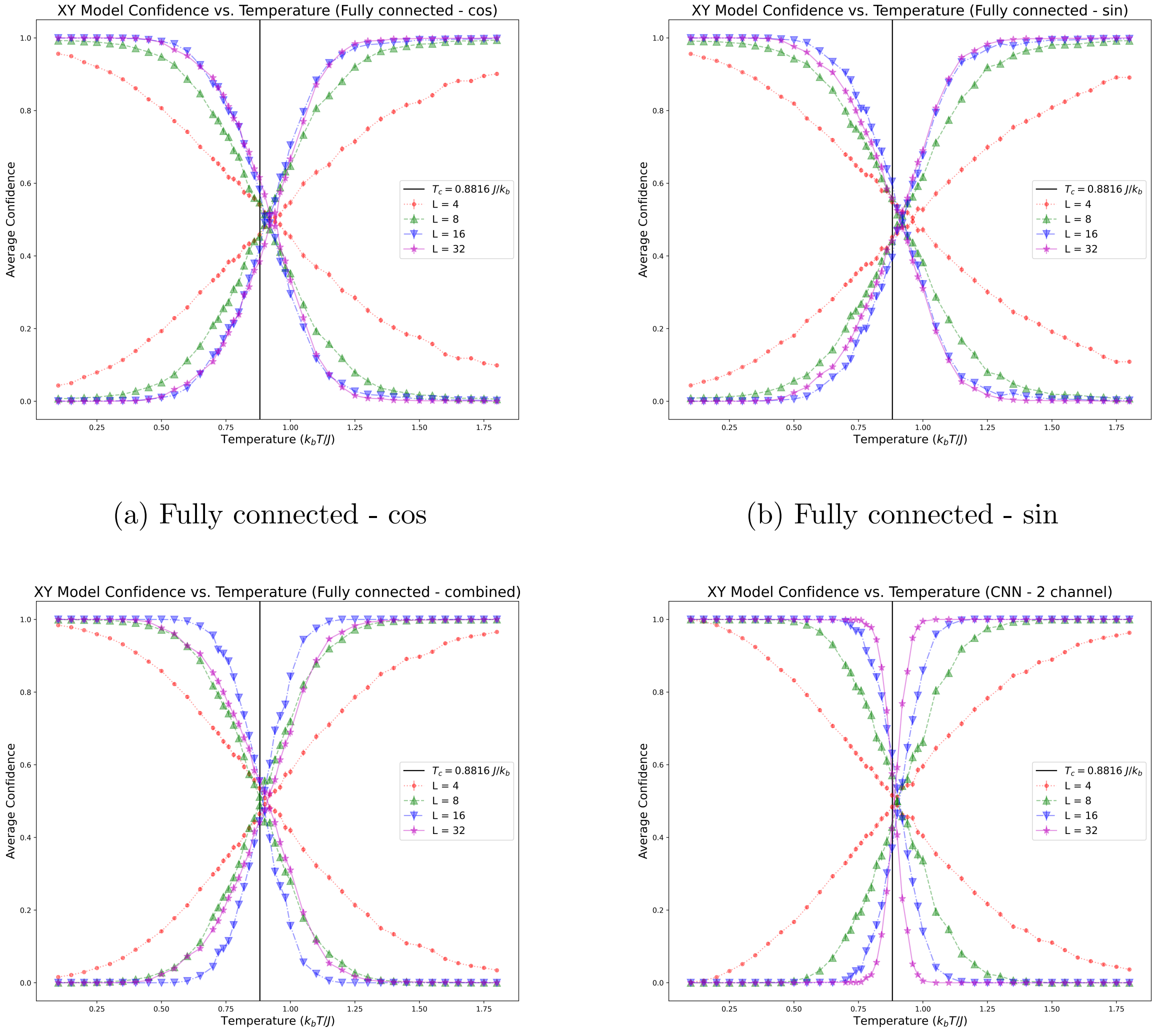
(a) $L = 4 \times 4$

(b) $L = 8 \times 8$

(c) $L = 16 \times 16$

(d) $L = 32 \times 32$

Fig. 12. Confidence vs. temperature at various system sizes for each of the XY model neural network architectures.



(a) Fully connected - cos

(b) Fully connected - sin

(c) Fully connected - combined

(d) CNN - 2 channel

Fig. 13. Confidence vs. temperature for different neural network architectures at various system sizes for the XY model.

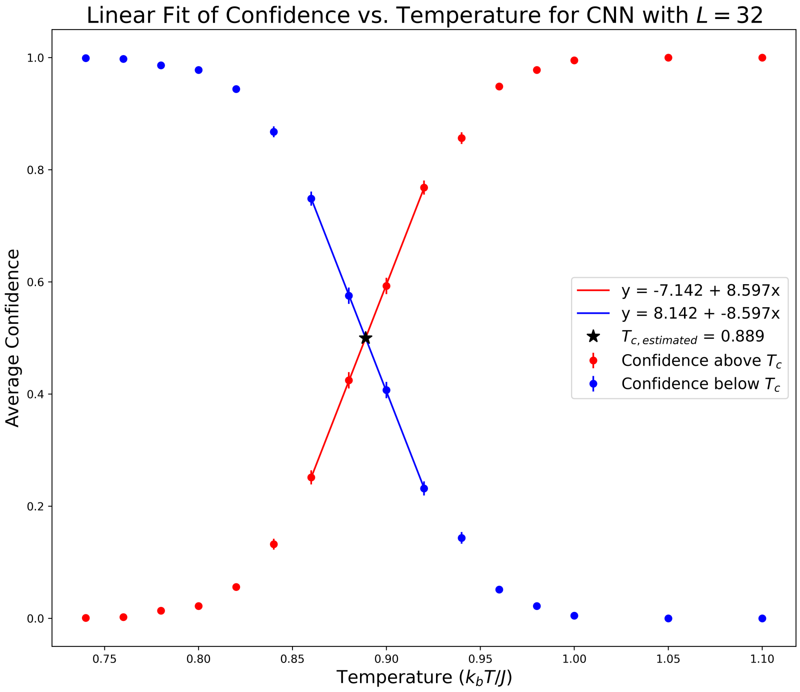


Fig. 14. Example linear fit functions in the region where confidence values intersect for a CNN architecture when $L = 32$. Finding the intersection point between these linear functions yielded the $T_{c,estimated}$, which was then used to produce the plot seen in Fig 11.

Bibliography

[1] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, pp. 431–434, Feb 2017.

[2] R. Datar, "Using Machine Learning To Study Phases Of Matter," 2023.

[3] M. J. S. Beach, A. Golubeva, and R. G. Melko, "Machine learning vortices at the Kosterlitz-Thouless transition," *Phys. Rev. B*, vol. 97, p. 045207, Jan 2018.

[4] L. Onsager, "Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition," *Phys. Rev.*, vol. 65, pp. 117–149, Feb 1944.

[5] E. Lieb, T. Schultz, and D. Mattis, "Two soluble models of an antiferromagnetic chain," *Annals of Physics*, vol. 16, no. 3, pp. 407–466, 1961.

[6] L. Hayward, "ON model." available `https://github.com/lhayward/ON_Model`.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015. Software available from `tensorflow.org`.

[8] Wikipedia contributors, "Propagation of uncertainty — Wikipedia, the free encyclopedia," 2024. [Online; accessed 8-May-2024].