



Exploring neural network training strategies to determine phase transitions in frustrated magnetic models

I. Corte^{a,b}, S. Acevedo^{c,b}, M. Arlego^c, C.A. Lamas^{b,c,*}

^a Departamento de Física, Universidad Nacional de La Plata, C.C. 67, 1900 La Plata, Argentina

^b Departamento de Ciencias Básicas, Facultad de Ingeniería Universidad Nacional de La Plata, La Plata, Argentina

^c IFLP - CONICET, Departamento de Física, Universidad Nacional de La Plata, C.C. 67, 1900 La Plata, Argentina

ARTICLE INFO

Keywords:

Frustrated magnetism
Machine learning
Ising model
Honeycomb lattice
Square lattice
Neural networks

ABSTRACT

The transfer learning of a neural network is one of its most outstanding aspects and has given supervised learning with neural networks a prominent place in data science. Here we explore this feature in the context of strongly interacting many-body systems. Through case studies, we test the potential of this deep learning technique to detect phases and their transitions in frustrated spin systems, using fully-connected and convolutional neural networks. In addition, we explore a recently-introduced technique, which is at the middle point of supervised and unsupervised learning. It consists in evaluating the performance of a neural network that has been deliberately “confused” during its training. To properly demonstrate the capability of the “confusion” and transfer learning techniques, we apply them to a paradigmatic model of frustrated magnetism in two dimensions, to determine its phase diagram and compare it with high-performance Monte Carlo simulations.

1. Introduction

The field of machine learning, in particular deep learning, has gained a prominent place in practically all areas associated with technology [1]. This is the result of the symbiosis between data generation, computing power, and algorithm development. What makes machine learning techniques especially useful in many applications is the automatic search for patterns and underlying models in the data. These models are then used to classify, predict, generate, and make decisions about new events or data.

The characteristic elements of the many-body interacting systems, such as high dimension, correlations, symmetries, and phase transitions, naturally emerge in data science and machine learning [2–5].

For this reason, it is evident that neural network techniques, which have been fundamental in data science and machine learning, will also play an important role in the physics of interacting many-body systems. This fact is reflected in some recent articles [6–8]. Our work points in this direction, taking a step forward in the implementation of neural networks to study frustrated magnetism (systems where cannot simultaneously minimize the energy contribution of all the magnetic couplings).

In this paper, motivated by the work of Carrasquilla and Melko [9]

and the research carried out recently [10–29], to cite a few, we analyze a variety of correlated classical spin models using neural networks. This provides a complement to more traditional methods, which include diverse analytical and computational tools [30,31].

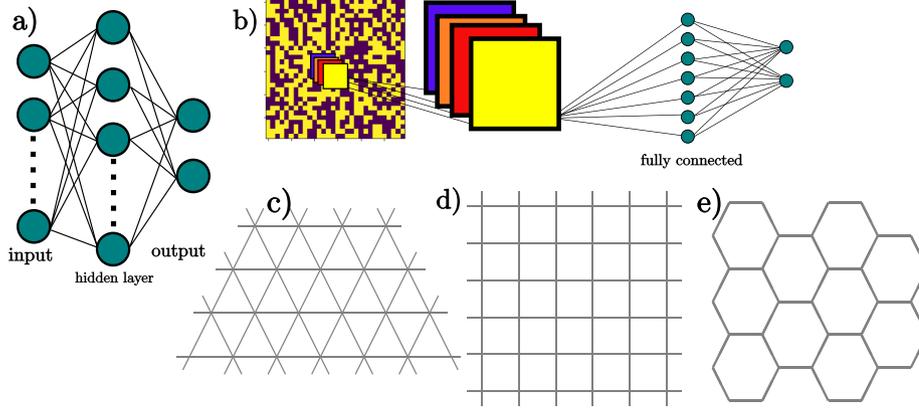
This work focuses on the flexibility and the generalization power provided by the use of fully-connected neural networks, also called dense neural networks (DNN)[32] and convolutional (CNN) neural networks. These architectures are schematically indicated in Fig. 1 (a) and (b), respectively. We aim to explore the possibilities that these techniques can offer in the study of correlated systems, beyond a particular and precise determination of a phase transition in a specific model and lattice. A key ingredient in this study is the incorporation of frustration into the system. That is why we choose to study purely antiferromagnetic systems where frustration emerges naturally. Unlike previous studies where the performance of neural networks in ferromagnetic systems is explored, when studying a frustrated system, whose order parameter is more complex, some architectures begin to fail and more complex neural networks must be used.

Throughout this paper, different types of spin lattices are explored (Fig. 1 (c-e)), together with several groups of hyperparameters [2], to test the performance of a given neural network against a classification task. The data in our work is synthetically generated by Monte Carlo

* Corresponding author.

E-mail address: lamas@fisica.unlp.edu.ar (C.A. Lamas).

simulations [33], which we implement for the different models and lattices studied.



lattices.

The last part of our work explores a technique that could be Fig. 1. Schematic representation of the neural networks and lattices implemented in this work. (a) Fully-Connected Neural Network, or Dense Neural Network (DNN). (b) Convolutional Neural Network (CNN). The input features correspond to 2D system configurations which are re-shaped into a 1D structure for the DNN, while in the CNN the input is the 2D “image” of the configuration. In both cases, the two-neurons output layer corresponds to order–disorder probabilities. (c) Triangular, (d) Square and (e) Honeycomb lattice.

For the supervised-network training, we test different types of input data, such as local spin configurations and correlations. This allows us to analyze the advantages and possibilities offered by the different types of features and their relevance in each particular case.

Most of the computations performed in this work are open source [34]. The corresponding codes are freely accessible via GitHub and data is available upon request to the authors. We have relegated most of the technical details about training and network architectures to the Appendix.

A significant part of this work is devoted to studying the “transfer learning” ability of neural networks in this kind of phase classification. Transfer learning consists of exploiting the predictive capacity of a neural network beyond the context in which it has been trained. In the framework of machine learning, the concept of transfer learning is broad [2]. It may be *indirect*, such as fine-tuning transfer learning, in which a pre-trained model in a certain domain is then fine-tuned in another domain, and feature-based transfer learning, in which certain intermediate features of a model are transferred to another domain. In order to clearly differentiate the scenarios where we will test the ability to generalize the knowledge obtained in the training for a neural network, we will use different labels. We use the label *direct* transfer learning, where the model trained in one domain is directly applied to another domain. This presents several levels of complexity. At a more basic level, networks trained in a restricted sector of a given model and lattice, for example at low and high temperatures, are used to predict its transition temperature [24]. At a higher level, networks trained with a certain model and lattice are used to predict properties, such as critical transition temperature, of the same lattice but in another model [13], we label this scenario as *model* transfer learning. Further increasing the complexity, networks trained with a given model and lattice are used to predict the same properties in a different model and lattice [9], we label this scenario as *model-lattice* transfer learning. We note that, although these are not standard names, their use will make the explanation of the results clearer.

As we have mentioned, the different transfer processes has been used in other works. However, the novelty here is its application to more complex systems, where frustration induces high degeneracy, representing a greater challenge to the power of generalization of the network as a classifier. We analyze these transfer learning processes on different realizations of the antiferromagnetic Ising model, including first- and second-neighbor interactions on the square, honeycomb, and triangular

considered a middle point between supervised and unsupervised learning. This method [35] of “learning by confusion” exploits the variability of a network performance that has been deliberately trained with incorrect labels. The advantage of this technique is that it does not need the correct labels for learning and can detect phase transitions, or at least significant pattern variations that suggest such a transition. This is the reason why it can be considered an unsupervised learning method. With the confusion technique, we teach a network to classify the phases of the AFM Ising model on the square lattice with second-neighbors interactions, which is an archetype of classical frustrated two-dimensional magnetism. This study complements the phase diagram determined with a CNN by *model transfer*.

For completeness, in the following we present a brief discussion of the prominent aspects of the Ising model family of interacting systems that we are going to analyze using neural networks.

We consider the Ising model in the absence of magnetic field described by the following Hamiltonian H , representing the energy of the system

$$H = \sum_{ij} J_{ij} \sigma_i \sigma_j, \quad (1)$$

in several two-dimensional lattices, where J_{ij} is the coupling between spins σ_i and σ_j (which can take values $\{\pm 1\}$) on sites i and j . Throughout the text, we simplify the notation by denoting first neighbors couplings as J and we use J_2 for second neighbors. Here, we will be particularly focused on the AFM case ($J_{ij} > 0$), due to its higher complexity and richness. The ferromagnetic case ($J_{ij} < 0$), although equally important, offers fewer difficulties and has been much more explored using neural networks, driven by Carrasquilla and Melko’s work [9], among others. The two-dimensional Ising model on the square lattice with first-neighbor interactions J was analytically solved by Onsager in 1944 [36]. He showed that there is an order–disorder phase transition for the infinite square lattice at the critical temperature $T_c = \frac{2|J|}{\ln(1+\sqrt{2})} \simeq 2.269|J|$ in units of Boltzmann constant k_B . On the honeycomb lattice, the critical temperature is also analytically available [37], $T_c = \frac{2|J|}{\ln(2+\sqrt{3})} \simeq 1.519|J|$. In both cases the result is valid for FM ($J < 0$) and AFM ($J > 0$) interactions. Solutions for general lattice topologies and couplings are obtained by means of series expansions and numerical methods, such as Monte Carlo [33].

The concept of frustration, which accounts for the impossibility of

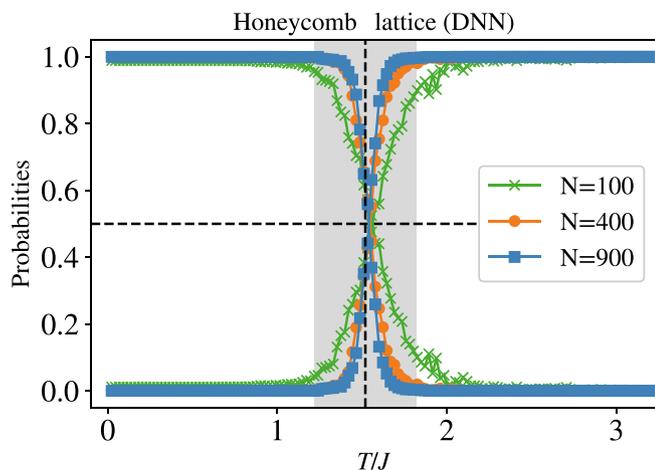


Fig. 2. Local (non-frustrated) DNN transfer. DNN output layer probabilities averaged over the test dataset as a function of T for the AFM-Honeycomb lattice. The lattice sizes are $N = 100, 400$ and 900 sites. The training data corresponds to the ranges $0.02 < T/J < 1.22$ and $1.82 < T/J < 4.53$. The shaded gray region represents the hidden set of data during training, and the vertical dotted black line shows the analytical result for the honeycomb lattice critical temperature in the thermodynamic limit. Validation accuracy, i.e. the ratio of correct predictions over the total number of predictions at the validation stage is 0.99, for $N = 100, 400$ and 900 sites (see Appendix for further details).

simultaneously minimizing all the couplings in the Hamiltonian from Eq. 1, takes the Ising model to a higher order of complexity. Frustration can take place in different ways, either due to the structure of the lattice itself or due to the inclusion of interactions beyond first neighbors, with additional finite couplings J_2, J_3 , etc. The AFM Ising model with first-neighbor couplings on the triangular lattice is an example of the first type. In this case, it is not possible to arrange the spins so that all interactions between them are antiparallel. Even in a single triangle of the lattice, if two antiparallel spins are placed on two vertices, the third vertex cannot be antiparallel to the other two. The system is “frustrated” because it cannot simultaneously minimize the energy contribution of all three couplings. As a consequence, there is no single way to minimize the system energy and it becomes degenerate at $T = 0$ [38]. This is a crucial difference with the FM counterpart, where the energy is minimized by placing all the spins in parallel.

On the other hand, the AFM Ising model with second-neighbor couplings on the square and honeycomb lattice illustrates the second case of frustration. Here the first neighbors can be arranged antiparallel, but if second neighbors are included this cannot be satisfied anymore. These are some of the cases we analyze in this work.

A direct consequence of frustration is the high degeneracy of the ground state, resulting in a wider variety and complexity of structures [39]. This makes frustrated systems ideal to explore the power of classification and generalization of neural networks in condensed matter interacting systems.

2. Training I: Local transfer

In this section, we begin our study of the phase diagrams with neural networks by exploring their performance to classify ordered and disordered phases by *local transfer*. We also present the general procedure for generating and labeling data, as well as the training-test scheme that we follow in the supervised learning part of the work.

2.1. Honeycomb lattice

To begin with, we considered the AFM Ising model on the honeycomb lattice. We start by considering the simplest neural network, a DNN.

We found that to classify the ordered and disordered phases in the AFM Ising model on the honeycomb lattice, it is enough to use a DNN with a single hidden layer of 16 neurons. As input variables, in this case, we use the local spin configurations that we compute from Monte Carlo simulations [33]. Since this is a binary classification problem involving ordered and disordered configurations, we employ the binary cross-entropy cost function, together with an L_2 regularization to further control overfitting (see Appendix for further details).

Monte Carlo generation of tagged data is performed as follows. We run 400 independent simulations starting from the high-temperature phase. The whole temperature range is partitioned, and for each temperature, the spin configuration and the temperature are saved once equilibrium is reached. Data with $T < T_c$ is labeled with 0 and data with $T > T_c$ is labeled with 1.

The data generated in the simulations is split 70% for training and validation and 30% for the test (prediction). Note that the temperature information is not introduced explicitly during training, since the DNN only uses the local spin configurations as input features and the labels 0 and 1 for minimizing the cost function. The temperature is used only in the test stage to analyze the performance of the classification and prediction of the critical temperature.

To be able to say if a prediction is correct or incorrect using supervised learning architectures, one has to compare it with the sample label. There are several ways to measure the performance of a neural network. Here we employ the *accuracy*, which is defined as the ratio between the number of well-classified samples and the total number of samples. The accuracy may depend on several factors such as the characteristics of the system near the transition temperature, the size of the training set, the geometry of the network used, etc. In any case, a validation accuracy of 0.99, for instance, tells us that the DNN correctly classified 99 out of 100 snapshots of the validation set (which is not used for training).

To test the network’s ability to predict beyond the data in which it is trained, we have trained the network in a range of temperatures that excludes a window of width w , centered at the transition temperature T_c . This case illustrates the *local transfer* process. Dense neural networks have shown that they can determine the transition temperature of a ferromagnetic system in this way. They show that they can learn simple order parameters such as magnetization. However, in the antiferromagnetic case, the order parameter is not so simple and depends on the lattice geometry.

Fig. 2 presents the results obtained for the classification of the ordered and disordered phases in the AFM Ising honeycomb lattice, for three different lattice sizes N of 100, 400, and 900 spins. In particular, it shows the probability of belonging to each phase as a function of temperature. This probability is obtained for each temperature by averaging the values predicted by the network on test data (not used for training) corresponding to this temperature. The shaded gray region indicates the range of data excluded for training (*local transfer*). Since the classification is binary, only the probability curve of one class is required for each size. For example, the blue curve starting at the top left of Fig. 2 indicates the probability of belonging to the ordered class for $N = 900$, while the lower blue curve to the left indicates the probability of belonging to the disordered class. Both curves add probability 1 and therefore are not independent (they are symmetric with respect to the line $p = 0.5$). However, throughout work, we show both probability curves as a visual guide to locate the predicted critical temperature, i.e. the crossing

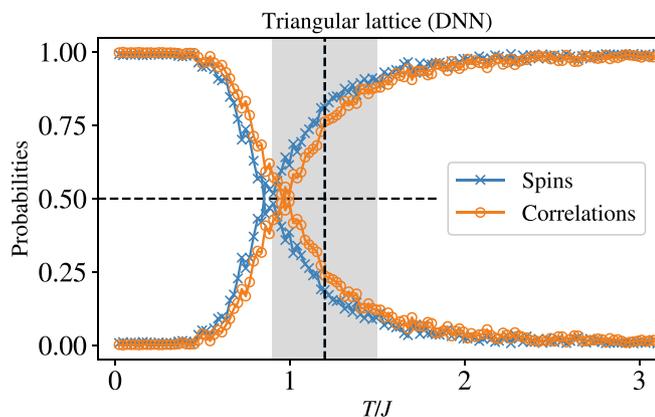


Fig. 3. Local (frustrated) DNN transfer. DNN output layer probabilities averaged over the test set, as a function of temperature for the AFM-Triangular lattice. The blue crosses and orange circles correspond to training with local spin configurations and correlations, respectively. The shaded gray region indicates the range of data that was excluded for training. The vertical dashed black line corresponds to the temperature T^* where the specific heat of the system reaches its maximum value in the Monte Carlo simulation. In both cases the DNN underestimates T^* , i.e. the crossing of both probability curves due to the additional complexity that frustration introduces in the triangular lattice. Validation accuracy: 0.97 for spin configurations and 0.98 for correlations.

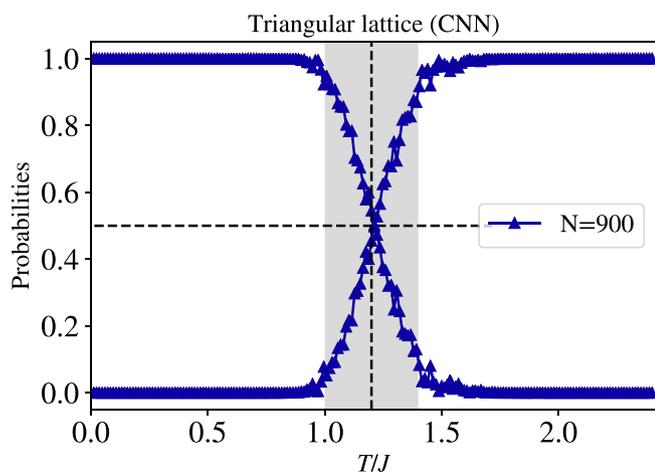


Fig. 4. Local (frustrated) CNN transfer: Output probabilities as a function of temperature, for the AFM-triangular lattice using a CNN. The shaded gray area corresponds to the range of data that was not used for training. The vertical line denotes the value of T^* , where the specific heat of the system reaches its maximum value with $N = 900$ sites. Validation accuracy of 0.99.

between both curves. The analytical critical temperature for the Ising honeycomb lattice (in units of k_B), $T_c \simeq 1.519J$, is indicated by the black dotted line in Fig. 2.

First, note that the DNN clearly separates the ordered and disordered classes for temperatures far from the transition. This is reflected in a probability prediction of approximately 1 (0) for the ordered (disordered) phase on the left and vice versa on the right.

However, as the temperature approaches T_c , the DNN finds it more difficult to differentiate between the two phases, which is indicated by the approach of the curves to the intermediate zone of the Fig. 2. The crossing of both curves at $p = 1/2$ defines the transition temperature predicted by the DNN for each size.

As it can be observed in Fig. 2, the prediction of the critical temperature is very close and slightly shifted to the right of the critical value, for each size. However, for $N = 100$ (green curves) the loss of predictive power falls faster and is noisier than in the other sizes. This

reflects the limitations and finite-size effects when using a smaller lattice.

The most important aspect to emphasize here is the performance of the DNN in the intermediate zone around the critical temperature, denoted by the gray area. Given that this intermediate zone is deliberately removed from training, the predictions made in this region manifest the generalizing power of the network beyond training. In other words, the DNN predicts the critical temperature with high accuracy despite never having seen data from the transition zone. This example illustrates the efficiency of *local transfer* with DNNs on non-frustrated lattices.

2.2. Triangular lattice

To test the performance of DNNs in a more complex context, we analyzed the AFM Ising model on the triangular lattice.

In the thermodynamic limit, this system has finite zero-point entropy, is disordered at all temperatures, and has no Curie point [38]. However, the Monte Carlo simulation of the model shows a maximum in the specific heat at a certain temperature T^* , below which short-range correlations emerge. The task of the neural network, in this case, is to differentiate low and high-temperature configurations, i.e. on both sides of T^* , when trained with samples of Monte Carlo simulations, excluding a window around T^* .

In the following, we set system sizes to $N = 900$ sites. Fig. 3 shows our results for the triangular lattice, trained with two different input features, using a DNN with a hidden layer of 32 neurons. The blue line shows the results obtained by training the network with local spin configurations, while the orange line corresponds to the ones obtained by training with correlations. The latter has been shown to be useful in multicomponent models, such as the Potts model [15]. For the Ising model we have evaluated the correlations $C_{xy} = \sigma_{xy}\sigma_{L/2,L/2}$, relative to the center $(L/2, L/2)$ of a $L \times L$ lattice with periodic boundary conditions. Subscripts x, y denote the spin position, relative to the vectors of the lattice unit cell. As can be observed in Fig. 3, in both cases, the DNN underestimate the position of T^* , indicated by the dashed vertical line in the figure. Using local spins as input features, the DNN predicts the T^* that falls slightly outside the left zone excluded from training (shaded gray area). The use of correlations in the training process improves the prediction of T^* , which falls inside the zone excluded from training. However, the additional complexity that frustration introduces in the triangular lattice reduces the DNN's performance to determine T^* in both cases. The overall performance of the network is quite insensitive to differences in the choice of hyperparameters of the network or alternative ways of evaluating correlations [15]. The AFM Ising model on the triangular lattice shows the limitations of DNNs in the presence of frustration and suggests the usage of higher-complexity neural network architecture would be more appropriate for this problem.

To address the previously-mentioned problem of limited DNN's performance in a frustrated lattice, we analyzed the Ising AFM model on the triangular lattice with a CNN. Convolutional networks are well-adapted to classify images, which in our case are the local spin configurations for each temperature value. The CNN extracts the relevant features from the input image through a successive application of pre-processing filters and then these feature maps serve as input for a final dense network. The filters and DNN parameters are learned during training.

Fig. 4 shows the results obtained using a CNN for the classification of low and high-temperature configurations of the AFM Ising model on the triangular lattice. The results were obtained by implementing a CNN with a convolutional layer of 10 filters, followed by a dense layer of 10 neurons. As before, we have applied the strategy of training excluding a window around T^* , limited by the shaded gray area. Unlike the DNN results in Fig. 3, the probability curves cross around T^* , indicated by the dashed vertical line.

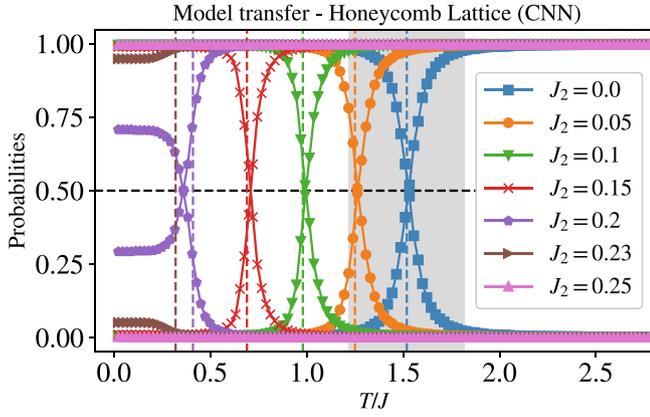


Fig. 5. Model transfer on the honeycomb lattice: CNN-predicted probabilities for ordered and disordered phases as a function of temperature in the AFM-honeycomb lattice. The CNN was trained using data for which $J_2/J = 0$ and temperatures outside the shaded gray area. Vertical dotted lines represent the Monte Carlo estimation of the transition temperatures. The network is able to correctly estimate the transition temperature of the model with second-neighbor interactions, despite being trained in the nearest-neighbor model. As J_2/J approaches 0.25 the change in the probability output suggests the presence of a phase transition. In the legend the J_2 values are in units of J . Validation accuracy: 0.99.

Since the CNNs show higher performance than DNNs in frustrated systems, in the following sections we will concentrate on the implementation of CNNs in these systems. In particular, we will analyze other aspects associated with the performance of CNNs and their power of generalization, not only at the local transfer level as we have done here. We will fully explore the potential of CNNs to predict transitions in other models as well as model and lattices in which the network has not been trained, i.e. CNN *model transfer*, and *model-and-lattice transfer* respectively.

We would like to end this section by making a general comment on the number of layers and the depth of the networks used. It is possible to increase the number of layers in the network, however it does not significantly improve the accuracy for the cases under study. In contrast, as the neural network gets deeper the number of trainable parameters augments exponentially, which significantly increases the computational cost and the likelihood of overfitting. In our cases, the neural network must be able to detect ordered phases (or low temperature

regimes) outside the training range. For this reason, it is very important to avoid overfitting.

3. Training II: Model transfer

As we discussed in the previous section, the CNNs can be very powerful to differentiate between low and high-temperature configurations in frustrated models as the AFM-triangular lattice, by training at temperatures away from T^* (local transfer). One step further would be to apply pre-trained neural networks in a given model and lattice to predict transitions in other models or lattices. In this section, we address *model transfer* by analyzing the performance of a pre-trained CNN to study another model, different from the one in which it was trained.

We start by training the CNN on the AFM Ising model on the honeycomb lattice with first-neighbor couplings (J), excluding a region around the critical temperature. To this end, we map the honeycomb lattice to a square array as detailed in Appendix B. This allows us to carry out comparisons and the lattice transfer in this work. Next, we evaluate its performance on data generated in the honeycomb lattice with first- and second-neighbor interactions, i.e. finite J_2 . If the ratio of second- to first-neighbor coupling J_2/J is small, the system remains in the same phase and it is reasonable to think that the CNN will be able to recognize the order. However, the transition temperature is a function of J_2/J and a priori it is not clear that the CNN can correctly detect the T_c behavior.

In Fig. 5 we show results of the probabilities predicted for the Néel ordered and disordered phases as a function of temperature. Each curve corresponds to a different value of the frustrating coupling J_2/J and the dashed vertical lines indicate the respective Monte Carlo estimations of the critical temperatures (See Appendix for details).

Let us recall that the CNN has not been trained with the presence of frustrating interactions, nor near the transition of the first-neighbor model, and yet for small values of J_2 the convolutional network prediction is remarkable.

The system has a low-temperature phase transition at $J_2/J = 0.25$, from the Néel phase to another phase where there is no long-range order due to the ground state degeneracy [40]. As J_2/J approaches 0.25 in Fig. 5 the CNN probability output changes progressively. At $J_2/J = 0.25$, there is zero probability prediction that the system is ordered for all temperatures. As the neural network was trained to learn the difference between Néel order and disorder, this outcome suggests that the system abandons the Néel ordered phase at $J_2/J = 0.25$.

To further explore *model transfer*, we have also analyzed the Ising model with first and second-neighbor couplings on the square lattice,

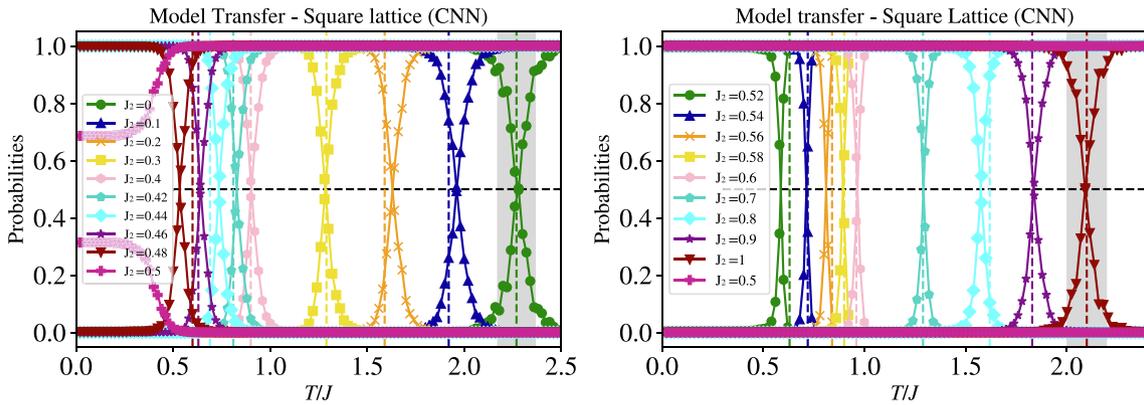


Fig. 6. Model transfer on the square lattice: Left: CNN output layer probabilities for Néel ordered and disordered phases for different values of $0 \leq J_2/J < 0.5$. The CNN was trained only with $J_2/J = 0$ data and temperatures outside the shaded gray area. Validation accuracy: 0.98. The CNN captures correctly the behavior of the transition temperature with respect to the parameter that regulates the frustration of the system, J_2/J . Right: CNN output layer probabilities for collinear-ordered and disordered phases for different values of $0.5 < J_2/J \leq 1$. The CNN was trained only for $J_2/J = 1$ and temperatures outside the shaded gray area. Validation accuracy: 0.99. We observe a similar behavior as in the left panel but as the ratio J_2/J decreases. In both panels, for $J_2 \approx 0.5J$ the change in the probability predictions suggests a phase transition, consistent with the zero-temperature Néel - collinear transition at $J_2/J = 0.5$. Dashed vertical lines indicate the Monte Carlo estimation for the transition temperatures. In the legends the J_2 values are in units of J .

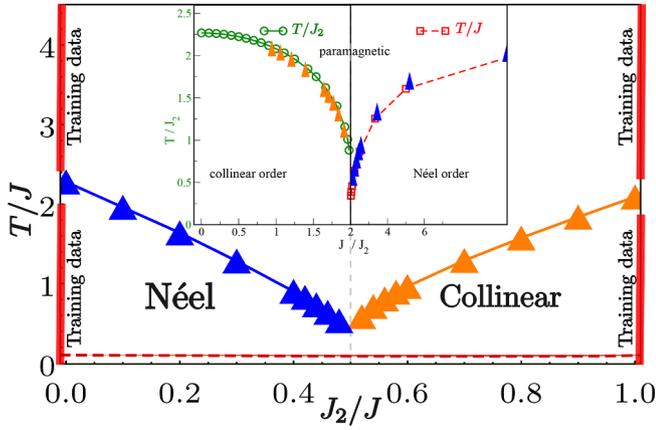


Fig. 7. Frustrated phase diagram by transfer learning. Main panel: CNN predicted phase diagram of the frustrated AFM Ising model on the square lattice with critical temperatures corresponding to the crossings at $p = 0.5$ of Fig. 6. Blue (orange) triangles indicate transition points from the Néel (collinear) to disordered phase, predicted via transfer learning from $J_2/J = 0$ ($J_2/J = 1$). The horizontal red dashed line indicates the temperature value used for confusion method training shown in Section 5. Inset: The green circles and red squares display high-precision Monte Carlo results of Fig. 2 from Ref. [41]. Orange and blue triangles display data from the main panel and were added for comparison. As can be observed, the resulting temperatures are in excellent agreement with the reference values despite training only with data from a very limited region of the phase diagram, indicated by thick vertical red lines at $J_2 = 0$ and $J_2 = J$.

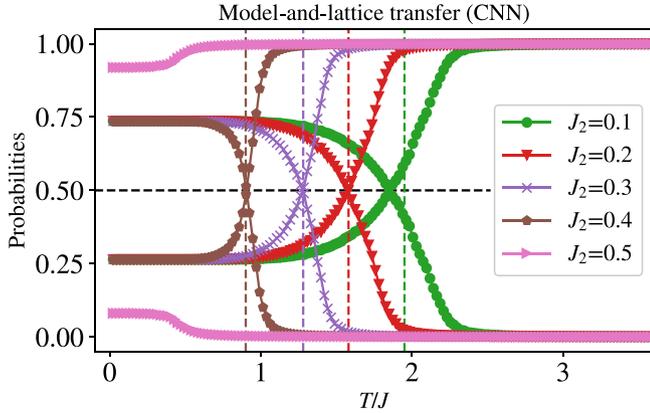


Fig. 8. Model-and-lattice transfer: CNN output layer probabilities on the *square lattice* as a function of temperature for different frustration values J_2 . The CNN was only trained with the $J_2 = 0$ *honeycomb lattice* data, with temperatures in the range $0.02 < T/J < 1.22$ and $1.82 < T/J < 4.53$ (the same as in Fig. 2). Vertical dotted lines indicate the corresponding Monte Carlo estimation of transition temperatures. In the legend the J_2 values are in units of J . Validation accuracy: 0.99.

starting from two different limiting cases. On the one hand, we trained a CNN with AFM Ising model data on the square lattice with first-neighbor couplings excluding as before the zone around the transition. Next we analyzed the network performance on the square lattice, including frustrating second-neighbor interactions J_2 . The left panel of Fig. 6 shows the results of the network predictions for the transition temperature between the disordered and Néel phases for different values of frustration J_2/J , whereas the dashed vertical lines indicate the Monte Carlo estimated critical temperatures. As can be observed, the CNN predicts accurately the change in the critical temperature for small J_2/J . For $J_2/J \approx 0.5$ the prediction becomes less accurate. This is consistent with the $T = 0$ transition from Néel to collinear phase at $J_2/J = 0.5$, characterized by a highly degenerate ground state [41].

On the other hand, we trained a CNN at $J_2 = J$, i.e. far from the

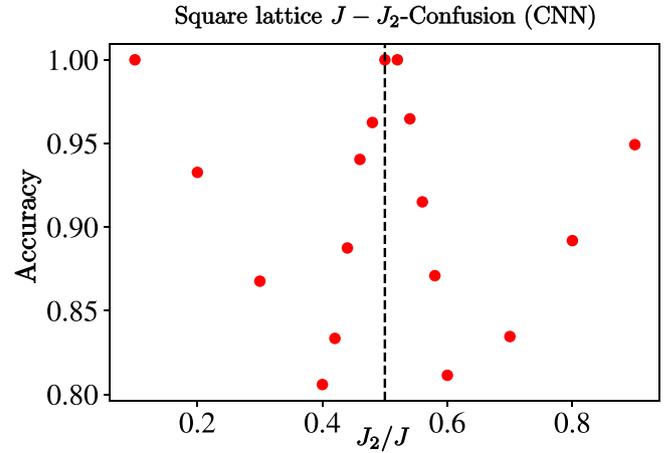


Fig. 9. Learning frustration by confusion: Results were obtained by applying the confusion method to the low-temperature phases of the frustrated $J - J_2$ AFM Ising model on the square lattice. The training was implemented at $T/J = 0.02$, indicated by the horizontal red dashed line in Fig. 7. The central peak of the characteristic “W-shape” is very close to $1/2$, which reflects the zero-temperature phase transition at $J_2/J = 1/2$ from the Néel to the collinear phase. This indicates that the confusion method is a powerful technique to determine phase transitions even in the presence of frustration.

maximal frustration point, $J_2/J = 0.5$, excluding as before the transition zone. The transition between the collinear and disordered phases exhibits a different nature (first or second-order transition depending on J_2) and has been studied in detail [41–43]. Following the same procedure as before, we evaluated the CNN performance by varying J_2/J from 1 up to $J_2/J \approx 0.5$. Results are shown in the right panel of Fig. 6 and a behavior similar to the left panel can be observed but on the other side of the transition.

With the transition temperatures predicted by the CNN, shown in Fig. 6, we built the temperature vs. frustration phase diagram of the frustrated Ising model on the square lattice, presented in Fig. 7. There are three phases, where the system presents Néel-order, collinear-order, or is disordered (paramagnetic phase). The blue (orange) line that separates the Néel-ordered (collinear-ordered) phase from the paramagnetic phase is constructed with the critical temperatures predicted by the CNN trained only in the region indicated by the thick vertical red lines on the left (right), with $J_2 = 0$ ($J_2 = J$). The CNN-predicted critical temperatures denoted here by blue (orange) triangles correspond to the intersections at $p = 0.5$ from the left (right) panel of Fig. 6. The inset in Fig. 7 shows the same transition temperatures, plotted with the scale of Fig. 2 of Ref. [41], along with that Figure. The comparison made in the inset shows the excellent agreement between our CNN predictions and the high-precision Monte Carlo results of Ref. [41], indicated with green circles and red squares. This is remarkable considering that almost the complete phase diagram was obtained by transfer learning since only spin configurations at temperatures within a restricted region (thick vertical red lines) were used as training data.

This result shows the robustness of the CNNs to generalize their predictions to different models from those in which they were trained, performing equally well in phase transitions of different nature, as illustrated in this example for the frustrated Ising model on the square lattice.

In the following section we further test the generalization power of CNNs by simultaneously performing a transfer in model and lattice.

4. Training III: Model-and-lattice transfer

We conclude the transfer learning study by analyzing the generalization power of a CNN trained with a given lattice and model and evaluating its performance on another lattice and another model

simultaneously, i.e. *model-and-lattice transfer*.

Keeping the number of training parameters as low as possible, the network can be forced to learn only the main features from the training data corresponding to a certain model and lattice. In this way, it is capable of correctly predict the critical temperature from data corresponding to a different model and lattice (see details in the Appendix).

We train a CNN with Monte Carlo spin configurations of the AFM nearest-neighbor Ising model on the honeycomb lattice excluding the phase transition zone, and we use it to predict the transition temperatures in the frustrated Ising model on the square lattice. As we map the honeycomb lattice to a square array (see Appendix B), the paramagnetic phase in both systems looks the same to the network, but the two low-temperature orderings are different. Hence, in general, a trained model cannot classify correctly the Néel ordered snapshots from the square lattice. Nonetheless, performing several pieces of training it is possible to obtain a model that achieves this task, and the number of tries necessary was found to be no greater than 80, which takes no longer than 2 h. To find this model the process is as follows: Once the network is trained on the honeycomb lattice, we use it to predict the order–disorder probabilities on the square lattice at some fixed value of J_2 , say $J_2 = 0.1$. If these two probabilities do not cross each other at any temperature, the trained model does not predict a transition, it is discarded, and a new training on the honeycomb lattice is performed. If, in contrast, the two probabilities cross each other, the model is saved. We emphasize here that this selection process does not utilize any a priori knowledge on the transition temperatures in the square lattice.

Fig. 8 shows the results of the CNN predictions on the frustrated square lattice for different frustration values J_2 as a function of temperature. As before, the dotted lines indicate the corresponding Monte Carlo estimation of the critical temperatures. The CNN’s T_c estimations are very close to those of the Monte Carlo simulations up to J_2 values near to the transition between the Néel and the collinear phases at $J_2 = 0.5J$ [41].

5. Training IV: Learning frustration by confusion

To further explore the magnetic phases in frustrated systems via deep learning, in this section we discuss the “confusion” learning technique [35]. This method can be considered a hybrid between supervised and unsupervised learning methods. The central idea of confusion learning is based on evaluating the accuracy, i.e. the ratio of correct predictions over the total number of predictions, of a network trained with a set of intentionally incorrect labels. For completeness, we briefly describe the method. Supposing that data depends on a parameter that lies in the range $[a, b]$, in which there is a phase transition at the critical point c^* , the method consists of proposing an arbitrary critical point c and training the network by giving the label 0 to all data with parameters smaller than c , and giving the label 1 to the rest. Next, the accuracy of the trained network on the complete training set, $P(c)$, is evaluated with respect to the proposed critical point c .

By sweeping the c values and repeating the process over the range $[a, b]$, the function $P(c)$ is obtained, which exhibits a W-shape with its intermediate peak located at the true critical point c^* .

The reason for this is as follows. When $c = a$ or $c = b$ all data is labeled with the same label, and therefore the network predicts with 100% accuracy. This explains the two ends of the W-shape. The central maximum in the network performance occurs when the proposed critical point c coincides with the true critical point c^* . In this case, the training process is equivalent to the standard supervised learning of the network.

Fig. 9 shows the results obtained by applying the confusion method to the low-temperature phases of the frustrated $J - J_2$ AFM Ising model on the square lattice. In this case, the training process was performed at $T/J = 0.02$, indicated by the horizontal red dashed line in Fig. 7 (see the Appendix for the CNN implementation details). For $T \rightarrow 0$, the model presents a phase transition at $J_2/J = 1/2$ from Néel to collinear order [41]. Note how the central peak of the W-shape emerges very close to $1/2$ in Fig. 9, indicating that the method can precisely determine the transition.

It is important to recall that the confusion method can detect changes in data patterns that are not necessarily associated with a physical phase transition. In this sense, the method is an indicator of a transition, which should be contrasted with other methods or prior knowledge of the physics involved.

6. Conclusions

In this work, we have explored the ability of neural networks to generalize knowledge beyond their training. This aspect of transfer learning, which has become a cornerstone in data science, has been discussed here in the context of antiferromagnetic Ising spin systems.

The transfer learning analysis was carried out in three stages of increasing order of generality demand.

We call the first case *local transfer*. Here, given a model on a lattice, we train a neural network to distinguish spin configurations far from the corresponding order–disorder transition temperature. Then we analyze the performance of the neural network to differentiate both phases in the complete range of temperatures, and we determine the transition temperature. This target is outside the training zone, although within the same model and lattice, hence the term “local” transfer. Our result is as follows. For non-frustrated lattices, *local transfer* works properly with DNNs. In this work, we illustrate this case with the AFM first neighbors Ising model on the honeycomb lattice (Fig. 2). However, for frustrated lattices, DNNs are not sufficiently accurate to guarantee adequate local transfer. This was tested not only with spin configurations but also with correlations as input features to train the neural network. We exemplified this case with the AFM first neighbors Ising model on the triangular lattice. The structure of the lattice gives rise in this case to frustration and consequently high degeneracy. This makes it difficult for the DNN to classify properly (Fig. 3). This difficulty is overcome using a convolutional network, whose pre-processing filter architecture extracts more representative features directly from the image of the configurations, making it suitable in high degeneracy cases (Fig. 4).

The second case analyzed was *model transfer*, in which we train a CNN using data on a given lattice (excluding the transition zone) and we then test its capability to classify data of a more general model on the same lattice. We illustrated this case with three examples. In the first one, we train the CNN with data corresponding to the non-frustrated AFM Ising model on the honeycomb lattice and we use it to classify data adding finite next-neighbors antiferromagnetic interactions.

(Fig. 5). For the other two examples we carried out a similar procedure, but for the AFM Ising model on the square lattice. We train the CNN far away from the maximally frustrated point $J_2 = J/2$, i.e., with data with $J_2 = 0$, or $J_2 = J$, respectively. With these two pieces of training, we construct the phase diagram of Fig. 7. The agreement of our CNN results with high-precision Monte Carlo results [41] depicted in the figure inset is significant, considering the small region of training.

These results indicate that CNNs generalize appropriately from the features learned in the restricted models where they were trained. The CNN not only quantitatively identifies the transitions between the

ordered and high-temperature phases but also gives evidence of the transitions between the low-temperature phases.

Finally, for the third and most demanding case, we considered *model-and-lattice transfer*. To this end, we train a convolutional network with spin configurations corresponding to the non-frustrated AFM Ising model on the honeycomb lattice, and we test the CNN performance on the frustrated AFM Ising model on the square lattice. The results in Fig. 8 show that it is possible to find a model that can generalize adequately, finding the order–disorder transitions for different frustration values, and signaling the low-temperature phase transition induced by frustration.

The transfer learning results presented in this work indicate that neural networks, in particular convolutional networks, can be adequate generic classifiers, exhibiting high performance when properly trained in minimal architectures, even in cases of high degeneracy such as the frustrated systems already analyzed. We plan to apply similar methods to other frustrated models at classical [44] and quantum level [45–47].

In addition to the implementation of supervised transfer learning, in this paper, we addressed the “learning by confusion” technique.

We implemented learning by confusion on the $J-J_2$ model on the square lattice using a CNN. Our results, depicted in Fig. 9, clearly show that the method can detect the emergence of the transition between the low-temperature phases. This is evidenced by the intermediate peak of the characteristic “W” shape of the predicted accuracy in Fig. 9, which is located very close to the transition point at $J_2 = 0.5J$.

The previous example highlights the main advantage of the confusion method, i.e. not having to rely on correct labeling beforehand to

detect pattern changes in the configuration data. In this case, the neural network learns to differentiate the low-temperature phases when frustration generates high degeneracy. In this sense, learning by confusion offers a complementary tool to the supervised learning techniques presented above.

CRedit authorship contribution statement

I. Corte: Data curation, Software, Writing – original draft. **S. Acevedo:** Data curation, Conceptualization, Software, Validation, Writing – original draft. **M. Arlego:** Supervision, Funding acquisition, Writing – original draft. **C.A. Lamas:** Supervision, Funding acquisition, Writing – original draft.

Declaration of Competing Interest

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

Acknowledgments

This research was partially supported by CONICET (P-UE 22920170100066CO), UNLP (Grant No. 11/X678, 11/X806). We would like to thank Andreas Honecker and Mateus Schmidt for very valuable information exchange. We also thank Andreas Honecker, Ansgar Kalz, and Marion Moliner for allowing us to use one of their Figures.

Appendix A. Methods

The computations performed in this paper are open source and written in C and Python, using specific libraries such as Keras for deep learning.

Monte Carlo simulations: The Monte Carlo generation of tagged data, using a Metropolis Algorithm and single-spin-flip dynamics, is performed as follows. We run 400 independent simulations starting from the high-temperature phase (in general $T_0 = 4.5J$). A set of 200 evenly-spaced temperature values is obtained from the range $[0, T_0]$, and for each temperature, the spin configuration and the temperature are saved once equilibrium is reached. Thus, our dataset for each lattice and each model considered consists of 80000 samples.

We use the analytical expressions available for the critical temperature in the cases of the non-frustrated Ising model on the square and honeycomb lattices. In the other cases, the location of the Monte Carlo specific heat maximum, for $N = 900$ sites, is taken as an estimator for the order–disorder transitions. In this work, this procedure was carried out for the frustrated Ising model on the square, honeycomb, and triangular lattices. In the latter, the above-mentioned maximum does not represent an order–disorder transition. In this case, we identify this temperature as T^* to avoid confusion.

Training procedure: The data generated by Monte Carlo simulations is split 70% for training (10% of which is taken for validation) and 30% for the test, i.e. prediction. Data with $T < T_c$ is labeled with 0 and data with $T > T_c$ is labeled with 1. The same exclusion process is carried out around T^* for the triangular lattice. Temperature values are only used in the test stage, to analyze the performance of the classification, and to estimate the critical temperature. We have trained the network in a range of temperatures that excludes a window $|T - T_c| < wJ$, with $0.1 \leq w \leq 0.3$. Therefore, no information about the critical region is introduced during training (*local transfer*). For *model transfer* and *model-and-lattice transfer* we used a similar procedure but validating on a different model as well as a different model and lattice to the training one, respectively. In the range of windows analyzed there are no significant variations in the accuracy.

Dense Neural Networks (DNN): In Fig. 2 we use as input a vector with the local spin configurations that we compute from the Monte Carlo simulations, normalized to 0 (spin down) or 1 (spin up). Three DNNs have an input layer having $N = 100, 400$ and 900 nodes, corresponding to three system sizes, respectively. The single hidden layer contains 16 neurons with ReLU activation functions. The stochastic optimization method is Adam, and the loss function is categorical cross-entropy. For training, we use roughly 50000–70000 configurations, 10% of which is used for validation during training. The learning rate is order 10^{-4} , the batch size is 128, the L_2 regularization factor is order 10^{-5} , and the number of epochs is 10, giving a final validation accuracy of 0.99 and a validation loss of order 10^{-2} . The output layer has two neurons (binary classification), with Softmax activation functions. The probability curves in Figs. 2 and 3 are the result of the average over roughly 150 independent samples from the test set for each temperature. In Fig. 3 both spin configurations and spin correlations input variables are normalized to 0 or 1. The hyperparameters that changed with respect to the previously mentioned are the number of neurons in the hidden layer, increased from 16 to 32, and the number of epochs increased from 10 to 20. The final validation accuracy is roughly 0.98 and there is a validation loss of order 10^{-1} over the training dataset.

Convolutional Neural Networks (CNN): The input is a 30x30 matrix, with normalized spin configurations. Data segmentation in train, validation, and test sets, is made as in the DNN case. Several architectures are suitable for classification tasks. The CNNs used consist of firstly in one or two convolutional layers of 3 to 10 filters of size 3x3, each followed by a max- or average-pooling layer. Then, the data is flattened to a one-dimensional vector which is the input of a dense layer with 3 to 16 neurons with ReLU activation functions. The optimization method is Adam, and the loss function is again the categorical cross-entropy. The batch sizes are between 128 and 512, the number of epochs is less than 5 to prevent overfitting, and the learning rate is order $10^{-3} - 10^{-4}$. The output layer has again two neurons with Softmax activation functions. Validation accuracy in training is higher than 0.99 and validation loss is order 10^{-2} . As before, the probability predictions over test datasets are averaged over roughly 150 independent samples from the test set for each temperature.

Source data for all the figures in the paper and all training and test data used are available upon request from the authors. Source code for training and evaluating our neural networks is available upon request from the authors.

Appendix B. Mapping the Honeycomb lattice to a square array

The size of the system is 900 sites and the Monte Carlo simulation was performed using periodic boundary conditions. Each unit cell is indexed by two integers (i, j) , where $0 \leq i < N_1 = 30$ and $0 \leq j < N_2 = 15$ as we show in Fig. B.10.

We map the honeycomb lattice to a square 30×30 array A as follows: Each spin in the honeycomb lattice is indexed by three integers in the 3-rank tensor S_{ij}^k , where i and j determine the unit cell (i, j) , and $k = 0$ or $k = 1$ corresponds to the left or right spin in the unit cell, respectively. Then, we can construct the square 30×30 array A by taking

$$A_{mn} = S_{m \lfloor *n/2 \rfloor}^{\text{mod}(n,2)}, \quad (\text{B.2})$$

with $0 \leq m, n < 30$, being $\text{mod}(n, 2)$ the rest in the division of n by two, and $\lfloor *n/2 \rfloor$ the floor function which gets the integer part in the division.

Using the mapping (B.2), we can visualize snapshots of the system in a 2D image as we show in Fig. B.11. The left panel corresponds to a Néel ordered spin configuration at $T = 0.02$ whereas the right panel corresponds to a disordered spin configuration at $T = 4.53$.

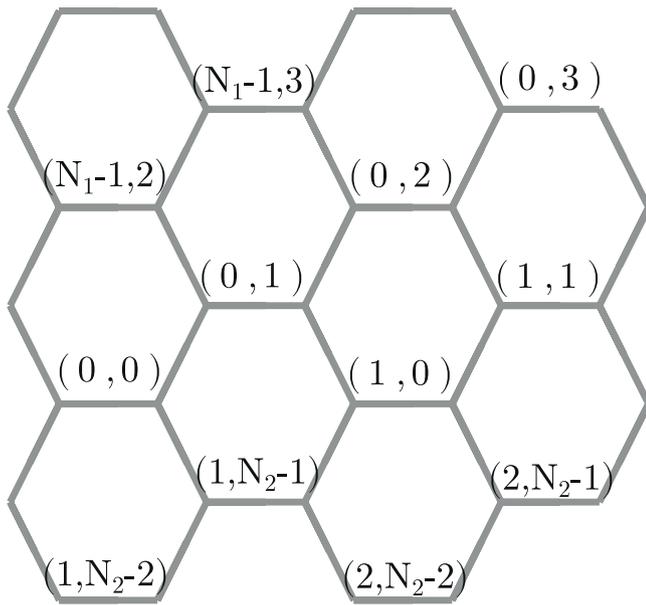


Fig. B.10. Honeycomb lattice. Ordered pairs correspond to the indexing for each cell used in this work.



Fig. B.11. Snapshots of the system mapped to a 30×30 pixel image. Each black or white pixel corresponds to a down or up spin, respectively. The left panel corresponds to a Néel ordered spin configuration at $T = 0.02$ whereas the right panel corresponds to a disordered spin configuration at $T = 4.53$.

References

- [1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [2] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT press (2016).
- [3] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, Springer Science & Business Media, 2009.
- [4] W. Li, Random texts exhibit zipf's-law-like word frequency distribution, *IEEE Trans. Information Theory* 38 (6) (1992) 1842–1845.
- [5] L. Saitta, A. Giordana, A. Cornuéjols, Phase Transitions in Machine Learning, Cambridge University Press, 2011. URL: <https://books.google.com.ar/books?id=wG4UnmfsdWQC>.
- [6] P. Mehta, M. Bukov, C.-H. Wang, A.G. Day, C. Richardson, C.K. Fisher, D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Phys. Rep.* 810 (2019) 1–124, <https://doi.org/10.1016/j.physrep.2019.03.001>. URL: <http://www.sciencedirect.com/science/article/pii/S037015719300766>.
- [7] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* 91 (4) (2019), 045002.
- [8] J. Carrasquilla, Machine Learning for Quantum Matter, arXiv e-prints (2020) arXiv:2003.11040 arXiv:2003.11040.
- [9] J. Carrasquilla, R.G. Melko, Machine learning phases of matter, *Nat. Phys.* 13 (5) (2017) 431–434.
- [10] C. Wang, H. Zhai, Machine learning of frustrated classical spin models. i. principal component analysis, *Phys. Rev. B* 96 (14) (2017), 144432.
- [11] C. Wang, H. Zhai, Machine learning of frustrated classical spin models (ii): Kernel principal component analysis, *Front. Phys.* 13 (5) (2018) 1–7.
- [12] M.J. Beach, A. Golubeva, R.G. Melko, Machine learning vortices at the kosterlitz-thouless transition, *Phys. Rev. B* 97 (4) (2018), 045207.
- [13] K. Ch'ng, J. Carrasquilla, R.G. Melko, E. Khatami, Machine learning phases of strongly correlated fermions, *Phys. Rev. X* 7 (3) (2017), 031038.
- [14] Y. Zhang, R.G. Melko, E.-A. Kim, Machine learning z 2 quantum spin liquids with quasiparticle statistics, *Phys. Rev. B* 96 (24) (2017), 245119.
- [15] K. Shiina, H. Mori, Y. Okabe, H.K. Lee, Machine-learning studies on spin models, *Scientific reports* 10 (1) (2020) 1–6.
- [16] P. Ponte, R.G. Melko, Kernel methods for interpretable machine learning of order parameters, *Phys. Rev. B* 96 (20) (2017), 205146.
- [17] P. Suchsland, S. Wessel, Parameter diagnostics of phases and phase transition learning by neural networks, *Phys. Rev. B* 97 (17) (2018), 174435.
- [18] J. Greitemann, K. Liu, L. Pollet, et al., Probing hidden spin order with interpretable machine learning, *Phys. Rev. B* 99 (6) (2019), 060404.
- [19] K. Ch'ng, N. Vazquez, E. Khatami, Unsupervised machine learning account of magnetic transitions in the hubbard model, *Phys. Rev. E* 97 (1) (2018), 013306.
- [20] P. Broecker, F.F. Assaad, S. Trebst, Quantum phase recognition via unsupervised machine learning, arXiv preprint arXiv:1707.00663.
- [21] P. Huembeli, A. Dauphin, P. Wittek, Identifying quantum phase transitions with adversarial neural networks, *Phys. Rev. B* 97 (13) (2018), 134109.
- [22] K. Liu, J. Greitemann, L. Pollet, et al., Learning multiple order parameters with interpretable machines, *Phys. Rev. B* 99 (10) (2019), 104410.
- [23] Y.-T. Hsu, X. Li, D.-L. Deng, S.D. Sarma, Machine learning many-body localization: Search for the elusive nonergodic metal, *Phys. Rev. Lett.* 121 (24) (2018), 245701.
- [24] P. Broecker, J. Carrasquilla, R.G. Melko, S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, *Scientific reports* 7 (1) (2017) 1–10.

- [25] W. Zhang, L. Wang, Z. Wang, Interpretable machine learning study of the many-body localization transition in disordered quantum ising spin chains, *Phys. Rev. B* 99 (5) (2019), 054208.
- [26] Q. Ni, M. Tang, Y. Liu, Y.-C. Lai, Machine learning dynamical phase transitions in complex networks, *Phys. Rev. E* 100 (5) (2019), 052312.
- [27] B.S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, C. Weitenberg, Identifying quantum phase transitions using artificial neural networks on experimental data, *Nat. Phys.* 15 (9) (2019) 917–920.
- [28] S.J. Wetzel, R.G. Melko, J. Scott, M. Panju, V. Ganesh, Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks, arXiv preprint arXiv:2003.04299.
- [29] L. Wang, Discovering phase transitions with unsupervised learning, *Phys. Rev. B* 94 (19) (2016), 195105.
- [30] A. Auerbach, *Interacting Electrons and Quantum Magnetism*, Graduate Texts in Contemporary Physics, Springer-Verlag, New York (1994), <https://doi.org/10.1007/978-1-4612-0869-3>.
- [31] H. Fehske, R. Schneider, A. Weiße, *Computational Many-Particle Physics*, Lecture Notes in Physics, Springer, Berlin Heidelberg, 2010. URL: <https://books.google.com.ar/books?id=2TugcQAACAAJ>.
- [32] F. Chollet, *Deep Learning with Python*, Manning Publications Company, 2017. URL: <https://books.google.com.ar/books?id=Yo3CAQAACAAJ>.
- [33] D.P. Landau, K. Binder, *A guide to Monte Carlo simulations in statistical physics*, Cambridge University Press, 2014.
- [34] F. Chollet, et al., Keras, <https://github.com/fchollet/keras> (2015).
- [35] E.P. Van Nieuwenburg, Y.-H. Liu, S.D. Huber, Learning phase transitions by confusion, *Nat. Phys.* 13 (5) (2017) 435–439.
- [36] L. Onsager, Crystal statistics. i. a two-dimensional model with an order-disorder transition, *Phys. Rev.* 65 (3–4) (1944) 117.
- [37] R.M.F. Houtappel, Order-disorder in hexagonal lattices, *Physica* 16 (5) (1950) 425–455.
- [38] G. Wannier, Antiferromagnetism. the triangular ising net, *Physical Review* 79 (2) (1950) 357.
- [39] C. Lacroix, P. Mendels, F. Mila, *Introduction to frustrated magnetism*, Vol. 164, Springer Science & Business Media, 2011.
- [40] A. Bobák, T. Lučivjanský, M. Žukovič, M. Borovský, T. Balcerzak, Tricritical behaviour of the frustrated ising antiferromagnet on the honeycomb lattice, *Phys. Lett. A* 380 (34) (2016) 2693–2697.
- [41] A. Kalz, A. Honecker, M. Moliner, Analysis of the phase transition for the ising model on the frustrated square lattice, *Phys. Rev. B* 84 (17) (2011), 174407.
- [42] A. Kalz, A. Honecker, Location of the potts-critical end point in the frustrated ising model on the square lattice, *Phys. Rev. B* 86 (13) (2012), 134410.
- [43] S. Jin, A. Sen, A.W. Sandvik, Ashkin-teller criticality and pseudo-first-order behavior in a frustrated ising model on the square lattice, *Phys. Rev. Lett.* 108 (4) (2012), 045702.
- [44] A. Kalz, M. Arlego, D. Cabra, A. Honecker, G. Rossini, Anisotropic frustrated heisenberg model on the honeycomb lattice, *Phys. Rev. B* 85 (2012), 104505, <https://doi.org/10.1103/PhysRevB.85.104505>. URL: <https://link.aps.org/doi/10.1103/PhysRevB.85.104505>.
- [45] D.C. Cabra, C.A. Lamas, H.D. Rosales, Quantum disordered phase on the frustrated honeycomb lattice, *Phys. Rev. B* 83 (2011), 094506, <https://doi.org/10.1103/PhysRevB.83.094506>. URL: <https://link.aps.org/doi/10.1103/PhysRevB.83.094506>.
- [46] C.A. Lamas, A. Ralko, M. Oshikawa, D. Poilblanc, P. Pujol, Hole statistics and superfluid phases in quantum dimer models, *Phys. Rev. B* 87 (2013), 104512, <https://doi.org/10.1103/PhysRevB.87.104512>. URL: <https://link.aps.org/doi/10.1103/PhysRevB.87.104512>.
- [47] H. Zhang, C.A. Lamas, M. Arlego, W. Brenig, Nematic quantum phases in the bilayer honeycomb antiferromagnet, *Phys. Rev. B* 97 (2018), 235123, <https://doi.org/10.1103/PhysRevB.97.235123>. URL: <https://link.aps.org/doi/10.1103/PhysRevB.97.235123>.