Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory

Sebastian J. Wetzel and Manuel Scherzer
Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany
(Received 27 May 2017; revised manuscript received 27 October 2017; published 8 November 2017)

We present a solution to the problem of interpreting neural networks classifying phases of matter. We devise a procedure for reconstructing the decision function of an artificial neural network as a simple function of the input, provided the decision function is sufficiently symmetric. In this case one can easily deduce the quantity by which the neural network classifies the input. The method is applied to the Ising model and SU(2) lattice gauge theory. In both systems we deduce the explicit expressions of the order parameters from the decision functions of the neural networks. We assume no prior knowledge about the Hamiltonian or the order parameters except Monte Carlo–sampled configurations.

DOI: 10.1103/PhysRevB.96.184410

I. INTRODUCTION

Machine learning enables computers to learn from experience and generalize their gained knowledge to previously unseen problems. The development of better hardware and algorithms, most notably artificial neural networks, propelled machine learning to one of the most transformative disciplines of this century. Nowadays such algorithms are used to classify images [1], to recognize language [2] or to beat humans in complex games [3]. Recently, machine learning has even been successfully employed to tackle highly complex problems in physics [4–15] and in turn physics has also inspired developments in machine learning [16–24]. It is now possible to classify phases of matter in the context of supervised learning [25–31] only from Monte Carlo samples. Phases can also be found without any information about their existence by unsupervised learning [32–35].

These algorithms suffer from a huge drawback: there is no comprehensive theoretical understanding of what they actually learn [36–40]. Without knowing if the neural networks base their decision on physical quantities one has no reason to trust the results if applied to an unknown system. Previous works suggest that machine learning discriminates phases of the Ising model by the order parameter [25,41]; others use the weights of the neural network to formulate a new order parameter [30].

In this paper (a) we propose a method to fully interpret neural networks, provided their decision function is sufficiently symmetric, (b) we explain this method at the Ising model and demonstrate its power at SU(2) gauge at theory, (c) we thereby justify the use of neural networks to classify phases, (d) this method yields such a clear interpretation that it can be used to determine the nature of the ordered phase.

To this end we introduce the correlation probing neural network. It can reduce the complexity of sufficiently symmetric decision functions. Since physical quantities are typically highly symmetric, this network is ideal for probing whether a physical quantity is responsible for the learned decision function. After reducing the complexity, we show that it is possible to fully reconstruct the explicit mathematical expression of the decision function in a simple form. From this expression one can extract the quantities by which the neural network distinguishes between phases.

This procedure is introduced at the Ising model, where we show that neural networks at distinguish between phases by the expected energy per spin (dominant) and the magnetization (subleading). We apply our method to SU(2) lattice gauge theory, where we find that the decision function is based on a nonlocal order parameter, the Polyakov loop.

II. ARTIFICIAL NEURAL NETWORKS

In this work we employ feed-forward artificial neural networks as a tool to distinguish between two classes in the context of supervised learning. After being successfully trained, the algorithm is able to predict the class of unseen test samples with high accuracy. We consider a neural network as an approximation of the decision function D. The decision function assigns to each sample S a probability $P \in [0,1]$ to be in class 1. The decision boundary is a hyperplane in the space of the parameters of sample configurations defined by $D(S) = 0.5$, where the neural network is most unsure about the correct label. If there exists an explicit quantity $Q(S)$ which is learned by the neural network, and which is responsible for the distinction between phases, we expect that a change in the quantity Q is always related to a change in the prediction probability, hence $\nabla Q || \nabla D$ in the vicinity of the decision boundary. In our neural networks the output can be written as $D(S) = \mathrm{sigmoid}(\xi(S))$, where $\mathrm{sigmoid}(x) = 1/[1+\exp(-x)]$ maps the latent prediction $\xi(S)$ to a probability. It follows that $\nabla Q || \nabla \xi$ and thus Q can be expressed as a linear function of ξ in a linearized regime close to the decision boundary $\xi(S) = w\, Q(S) + b$. The decision function of neural networks is encoded in a highly elusive and highly nonlinear way. In order to decode the decision function, we present a type of neural network that is tailored to probe if specific correlations between different variables contribute to the decision function of the neural network. We call it the *correlation probing neural network*; see Fig. 1. The neural network architecture can be found in Appendix B.

The idea is to construct a tunable neural network which is able to interpolate between a traditional feed-forward neural network in one limit and an optimal minimal neural network, that still yields a similar classification performance, in the other limit. A neural network is an algorithm that excels in identifying hierarchical structure on data. These hierarchical functions can in principle be decomposed into simpler subfunctions. To this end the correlation probing neural network is decomposed into subnetworks of which each can only learn a specific function. The subfunctions are unique

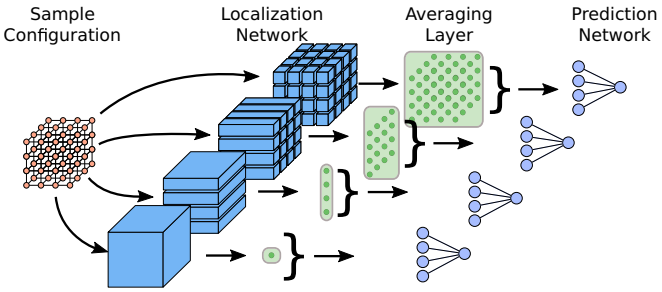

FIG. 1. The correlation probing neural network consists of three types of neural networks stacked on top of each other. The localization network is a fully convolutional neural network which prohibits connections outside of the receptive field of each output neuron and thus only recognizes correlations in the receptive field. The averaging layer averages over the input from the localization network, similarly to how the magnetization averages over all spins. The prediction network is a fully connected neural network, which transforms the output of the averaging layer to a prediction probability.

up to a linear transformation. The procedure for finding the optimal minimal neural network is to reduce the capacity of each of these subnetworks in an ordered manner until the neural network experiences a significant drop in the classification performance. The decision function of the optimal minimal neural network can then be written in a simplified form. If the quantity by which the neural network classifies the input is highly symmetric it is often possible to read off the quantity from the decision function. This is the case in the classification of phases in many physical systems.

III. ISING MODEL

The Hamiltonian of the ferromagnetic nearest-neighbor Ising model on the square lattice with vanishing external magnetic field is

$$H(S) = -J \sum_{\langle i,j\rangle_{nn}} s_i s_j, \tag{1}$$

with $J = 1$, $S = (s_1, \ldots, s_N)$ denotes a spin configuration, where $s_i \in \{1, -1\}$. It is a simple, well studied, and exactly solvable model from statistical physics that undergoes a second-order phase transition at $T_c = 2/[k_B \ln(1+\sqrt{2})]$ [42]. At T_c the specific heat $C_V = \partial\langle E\rangle/\partial\beta$ diverges, as does the temperature derivative of the expectation value of the absolute value of the magnetization $\langle M\rangle$, where $M(S) = |1/N \sum_i s_i|$.

The existence of different phases in the Ising model in the low- and high-temperature limit is known from unsupervised learning [32,33,35]. Using this knowledge we train the correlation probing neural network to predict phases of Monte Carlo–sampled configurations of size 28×28 below $T = 1.6$ in the ordered phase and above $T = 2.9$ in the unordered phase. More information about Monte Carlo simulation can be found in Appendix A. Using the full receptive field of 28×28, we allow the neural network to learn all possible spin correlations to approximate its decision function. In this case, the correlation probing network is equivalent to a standard convolutional neural network. Training and validation losses close to zero indicate that the neural network has found all necessary information it needs to reliably classify the phases. By successively lowering the receptive field size, we do not observe a drop in performance, except from 1×2 to 1×1 and from 1×1 to the baseline classifier; see Table I. In each of these steps the neural network loses important information about the samples. In Fig. 2(c) we can see the average classification probability, as a function of the temperature, of both networks. The phase-transition temperature can be found where $P = 0.5$. This is at $T = 2.5 \pm 0.5$ for the 1×1 network and $T = 2.25 \pm 0.25$ for the 1×2 network. An accurate estimation can be found in [25]. We however focus on examining what information got lost while lowering the receptive field size.

TABLE I. Ising model: Losses of neural networks with different receptive fields of the neurons in the localization network. Smaller numbers mean better performance. The baseline classifier is a random classifier which predicts each phase with a probability of $p = 0.5$.

Receptive field size	Train loss	Validation loss
28×28	6.1588×10^{-4}	0.0232
1×2	**1.2559×10^{-4}**	**1.2105×10^{-7}**
1×1	**0.2015**	**0.1886**
Baseline	0.6931	0.6931

By construction, the decision function D of the 1×1 neural network can be expressed as

$$D(S) = F\left(\frac{1}{N}\sum_i f(s_i)\right) = \text{sigmoid}\left[\xi\left(\frac{1}{N}\sum_i f(s_i)\right)\right], \tag{2}$$

where F is the function approximated by the prediction network and f is the function approximated by the localization network. The function f can be Taylor expanded:

$$f(s_i) = f_0 + f_1 s_i + f_2 \underbrace{s_i^2}_{1} + f_3 \underbrace{s_i^3}_{s_i} + \cdots. \tag{3}$$

Since $s_i^2 = 1$, all higher-order terms can be neglected. The constants f_0 and f_1 can be absorbed by the bias and the weights of the prediction network approximating F. Thus, the decision function reduces to

$$D(S) = F\left(\frac{1}{N}\sum_i s_i\right). \tag{4}$$

In order to determine the function F, we compare the latent prediction ξ of the neural network, with the argument of F: $1/N \sum_i s_i$, in the vicinity of the decision boundary; see Fig. 2(a). This knowledge allows us to construct the decision function

$$D(S) \approx \text{sigmoid}\left(w\left|\frac{1}{N}\sum_i s_i\right| + b\right), \tag{5}$$

with weight w and bias b of the prediction neuron. The perfect correlation between the latent prediction $\xi(S)$ and $|1/N \sum_i s_i|$ further reinforces that our above deduction was correct. Until this point we have not used any information about the Ising model except Monte Carlo configurations. We have found that the decision function determines the phase by the quantity $Q(S) = |1/N \sum_i s_i|$. This function is the magnetization.

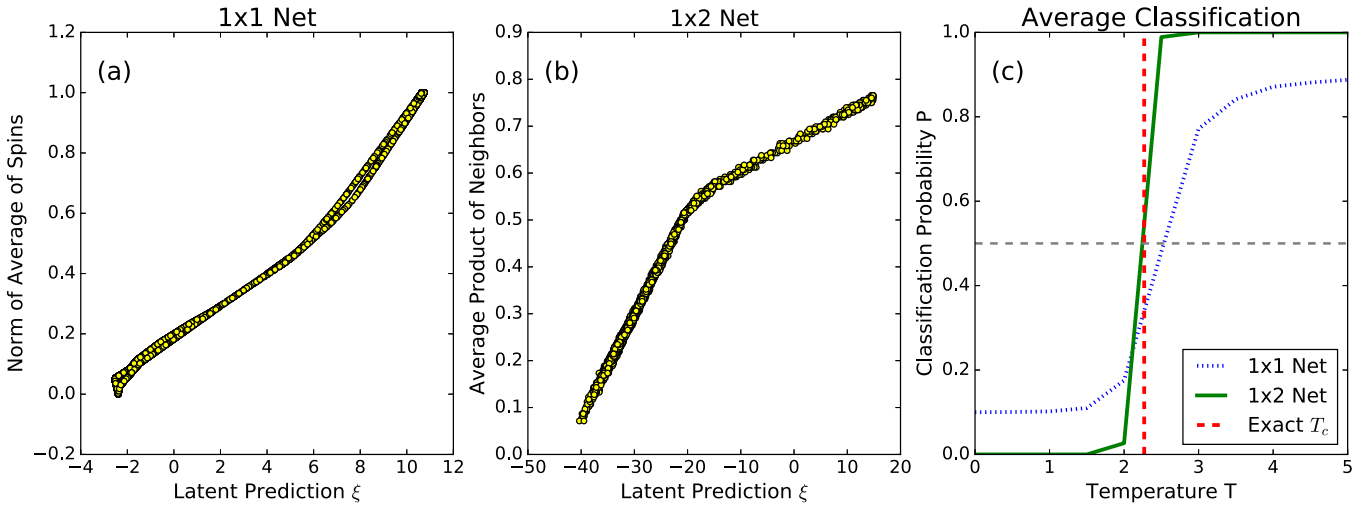

FIG. 2. Results of the correlation probing neural network applied to the Ising model. The latent prediction ξ is the argument of the sigmoid function in the last layer of the prediction network. (a) The latent prediction is perfectly correlated with the absolute value of the average of spins, i.e., the magnetization in the 1×1 network, for all sampled configurations. (b) The latent prediction of the 1×2 network is perfectly correlated with the average product of neighbors, i.e., the expected energy per site.

By examining the 1×2 network, we require by construction that the decision function is of the form

$$D(S) = F\left(\frac{1}{N}\sum_{\langle i,j\rangle_T} f(s_i,s_j)\right). \tag{6}$$

Here the sum only goes over transversal nearest neighbors, collecting each spin only once. The Taylor expansion,

$$\begin{aligned} f(s_i,s_j) =& f_{0,0} + f_{1,0}\,s_i + f_{0,1}\,s_j \\ &+ f_{2,0}\,s_i^2 + f_{1,1}\,s_i\,s_j + f_{0,2}\,s_j^2 + \cdots, \end{aligned} \tag{7}$$

contains only three terms of note; all other terms can be reduced to simpler ones by using $s_i^2 = 1$. The terms $f_{1,0}\,s_i$ and $f_{0,1}\,s_j$ represent the magnetization. From Table I and the analysis of the 1×1 network, we know that these terms contain less information than the quantity we are looking for. So the leading term must be $f_{1,1}s_is_j$. Thus, the decision function can be written as

$$D(S) \approx F\left(\frac{1}{N}\sum_{\langle i,j\rangle_T} s_i s_j\right). \tag{8}$$

In Fig. 2(b) we see the perfect correlation between the latent prediction $\xi(S)$ and $1/N\sum_{\langle i,j\rangle_T} s_i s_j$. This also means that the correction from the subleading terms $f_{1,0}\,s_i$ and $f_{0,1}\,s_j$ is indeed negligible. Hence, we end up with the decision function

$$D(S) \approx \text{sigmoid}\left[w\left(\frac{1}{N}\sum_{\langle i,j\rangle_T} s_i s_j\right) + b\right]. \tag{9}$$

By translational and rotational symmetry, the sum can be generalized to all neighbors $Q(S) = \frac{1}{N}\sum_{\langle i,j\rangle_{nn}} s_i s_j$. This quantity is, up to a minus sign, the average energy per spin site. It is worth noting that the energy per site can be used to distinguish between phases more reliably than the magnetization; see Table I.

IV. SU(2) LATTICE GAUGE THEORY

We examine SU(2) lattice gauge theory, which shows confinement, one of the most distinct features of QCD. It builds on the idea of discretizing the Euclidean path integral of SU(2) Yang-Mills theory. Lattice configurations are defined by a set of link variables $U_\mu^x \in \text{SU}(2)$. Each matrix connects two sites on a four-dimensional $x \in N_\tau \times N_s^3$ space-time lattice with $N_\tau = 2$ (temporal direction) and $N_s = 8$ (spatial volume). The direction is indicated by $\mu \in \{\tau,x,y,z\}$. A sample lattice configuration collects all link variables on the lattice $S = (\{U_\mu^x\})$. Each U_μ^x is parametrized by four real parameters,

$$U_\mu^x = a_\mu^x \mathbb{1} + i\left(b_\mu^x\sigma_1 + c_\mu^x\sigma_2 + d_\mu^x\sigma_3\right), \tag{10}$$

where σ_i are the Pauli matrices; the coefficients obey $(a_\mu^x)^2 + (b_\mu^x)^2 + (c_\mu^x)^2 + (d_\mu^x)^2 = 1$. The trace of U_μ^x is given by $2\,a_\mu^x$, since the Pauli matrices are traceless. We employ the lattice version of the Yang-Mills action, the Wilson action [43],

$$S_{\text{Wilson}}[U] = \beta_{\text{latt}}\sum_x\sum_{\mu<\nu}\text{Re tr}\left(\mathbb{1} - U_{\mu\nu}^x\right), \tag{11}$$

where β_{latt} is the lattice coupling. Here $U_{\mu\nu}^x = U_\mu^x U_\nu^{x+\hat{\mu}} U_{-\mu}^{x+\hat{\mu}+\hat{\nu}} U_{-\nu}^{x+\hat{\nu}}$ is the smallest possible closed rectangular loop. The order parameter for the deconfinement

TABLE II. SU(2): Losses of neural networks with different receptive fields of the neurons in the localization network (* no hidden layers in the prediction net).

Receptive field size	Train loss	Validation loss
$2 \times 8 \times 8 \times 8$	1.0004×10^{-4}	2.6266×10^{-4}
$\mathbf{2 \times 1 \times 1 \times 1}$	$\mathbf{8.8104} \times 10^{-8}$	$\mathbf{6.8276} \times 10^{-8}$
$\mathbf{2 \times 1 \times 1 \times 1}^*$	$\mathbf{2.2292} \times 10^{-7}$	$\mathbf{4.2958} \times 10^{-7}$
$1 \times 1 \times 1 \times 1$	0.6620	0.9482
Baseline	0.6931	0.6931

phase transition is the expectation value of the Polyakov loop

$$L(\vec{x}) = \mathrm{tr}\left(\prod_{x_0=0}^{N_\tau-1} U_\tau^x\right) \overset{N_\tau=2}{=} \mathrm{tr}\big(U_\tau^{0,\vec{x}} U_\tau^{1,\vec{x}}\big)$$
$$= 2\big(a_\tau^{0,\vec{x}} a_\tau^{1,\vec{x}} - b_\tau^{0,\vec{x}} b_\tau^{1,\vec{x}} - c_\tau^{0,\vec{x}} c_\tau^{1,\vec{x}} - d_\tau^{0,\vec{x}} d_\tau^{1,\vec{x}}\big). \quad (12)$$

It is the trace of a closed loop that winds around time direction using periodic boundary conditions. The expectation value of the Polyakov loop is zero in the confined phase and finite in the deconfined phase. More details on the simulations can be found in Appendix A.

The existence of different phases in SU(2) lattice gauge theory can be found by unsupervised learning; see Appendix D. This knowledge allows us to train the correlation probing neural networks with different receptive fields, to classify phases on Monte Carlo–sampled configurations at lattice coupling $\beta \in [1,1.2]$ in one phase and $\beta \in [3.3,3.5]$ in the other phase. We test the neural network in $\beta \in [1.3,3.2]$ to predict a phase transition at $\beta = 1.99 \pm 0.10$ ($2\times1\times1\times1$ network) and $\beta = 1.97 \pm 0.10$ ($2\times8\times8\times8$ network); see Fig. 3(c). Our direct lattice calculation reveals $\beta = 1.880 \pm 0.025$. By successively lowering the receptive field size we lose important information for classifying phases below $2\times1\times1\times1$; see Table II. This means that crucial information about the phase transition is contained in this specific structure.

The decision function of the $2\times1\times1\times1$ network is constrained to

$$D(S) = F\left(\frac{2}{N}\sum_{\vec{x}} f\big(\{U_\mu^{x_0,\vec{x}}\}\big)\right), \quad (13)$$

where the arguments of f are eight matrices at spatial location $\vec{x}$. A general approach to find F and f is presented in Appendix F. A simpler approach is based on the observation that we do not need any hidden layers in the prediction network, i.e., we only keep the output neuron; see Table II. Then the decision function simplifies to $D(S) = \mathrm{sigmoid}[w\,Q(S) + b]$, where

$$Q(S) = \frac{2}{N}\sum_{\vec{x}} f\big(\{U_\mu^{x_0,\vec{x}}\}\big) \quad (14)$$

reduces to a sum of functions acting only on a single patch of size $2\times1\times1\times1$ each. This allows us to split all samples to a minimum size of $2\times1\times1\times1$. We train a new local neural network to classify the phases of each local sample. By performing a regression on the latent prediction of the local neural network, we find that a second-order polynomial performs best (a comparison of different algorithms is found in Appendix E). The regression approximates the latent prediction by a sum of 561 terms,

$$f\big(\{U_\mu^{x_0}\}\big) \approx +7.3816\, a_\tau^0 a_\tau^1 + 0.2529\, a_\tau^1 b_\tau^1 + \cdots$$
$$- 0.2869\, d_\tau^0 c_\tau^1 - 7.2279\, b_\tau^0 b_\tau^1$$
$$- 7.3005\, c_\tau^0 c_\tau^1 - 7.4642\, d_\tau^0 d_\tau^1. \quad (15)$$

We only keep the leading contributions and assume that the differences between the leading contributions originate from approximation errors. Since overall factors and intercepts can be absorbed in the weights and biases of the neural network, we can simply rescale the above parameter to

$$f\big(\{U_\mu^{x_0}\}\big) \approx a_\tau^0 a_\tau^1 - b_\tau^0 b_\tau^1 - c_\tau^0 c_\tau^1 - d_\tau^0 d_\tau^1 = \mathrm{tr}\big(U_\tau^0 U_\tau^1\big). \quad (16)$$

This is the Polyakov loop on a single spatial lattice site (12). We promote $f(\{U_\mu^{x_0}\}) \to f(\{U_\mu^{x_0,\vec{x}}\})$ to act on the full lattice, such that we can formulate the decision function of the neural network with the full receptive field as

$$D(S) \approx \mathrm{sigmoid}\left[w\left(\frac{2}{N}\sum_{\vec{x}} f\big(\{U_\mu^{x_0,\vec{x}}\}\big)\right) + b\right]. \quad (17)$$

Here $Q(S) = \frac{2}{N}\sum_{\vec{x}} f(\{U_\mu^{x_0,\vec{x}}\})$ is the Polyakov loop on the full lattice. A confirmation of this deduction can be seen in the perfect correlation between the latent prediction and the Polyakov loop in Figs. 3(a) and 3(b).

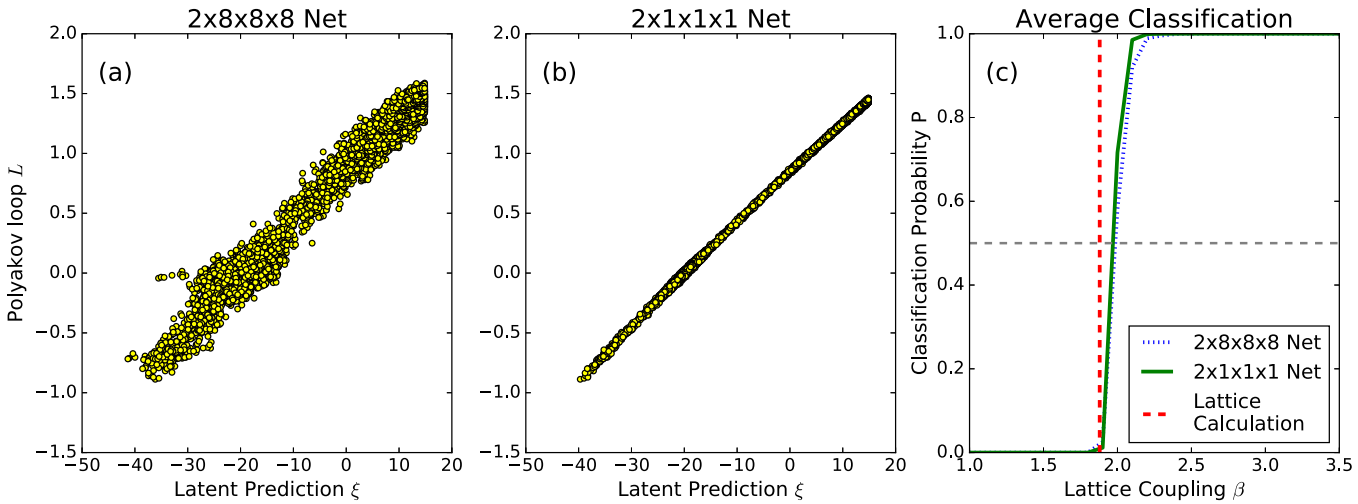

FIG. 3. Results of the correlation probing network applied to SU(2) lattice gauge theory. (a),(b) The latent prediction shows a strong correlation with the Polyakov loop in both the $2\times8\times8\times8$ network and the $2\times1\times1\times1$ network. (c) The average prediction probability of the two networks.

TABLE III. Ising model neural network. A, B, C determine the receptive field size of each neuron in the averaging layer.

Layer	Output shape	Kernel size
InputLayer	(784, 1)	
Convolution1D	$(784/(A),n_A)$	A
Convolution1D	$(784/(A \times B),n_B)$	B
Convolution1D	$(784/(A \times B \times C),n_C)$	C
Average pooling	$(1, n_C)$	
Flatten	(n_C)	
Dense	(n_D)	
Dense	(1)	

TABLE V. Scores of different regression algorithms. Higher is better.

Order of regression	Train score	Validation score
Polynomial regression		
1	0.00128	-0.00042
2	**0.72025**	**0.72395**
3	0.75675	0.69129
Support vector regression		
1	-0.08943	-0.08988
2	0.64048	0.65367
3	-0.08434	-0.08963

V. CONCLUSION

We proposed and demonstrated a method to fully interpret neural networks, which is based on the correlation probing neural network. The method was introduced at the Ising model on the square lattice, where the neural network predicts phases via the magnetization (5) or the expected energy per site (9). We then demonstrated the power of this method at SU(2) lattice gauge theory, where the reconstructed decision function reveals the explicit mathematical expression of the Polyakov loop (17), a nonlinear, nonlocal order parameter. This method provides the means to judge whether neural networks have learned physical properties and thus whether their results can be trusted. Furthermore, our procedure can be used to deduce the explicit formulas of physical order parameters. Since our approach is vastly different than conventional methods, it could determine the nature of phases where conventional methods have not yet succeeded.

A first application could be identifying if machine learning methods classify sign problematic models by physical quantities [26]. Then we could reliably determine the phase diagram of QCD at finite density [44–46] or examine the pseudogap [47–50] or the competition between d-wave and antiferromagnetic order [51–54] in the two-dimensional Hubbard model.

ACKNOWLEDGMENTS

We would like to thank J. M. Pawlowski, M. Salmhofer, I.-O. Stamatescu, and C. Wetterich for useful discussions. We thank M. Neidig and S. Nkongolo for reviewing the manuscript. S.J.W. acknowledges support by the Heidelberg Graduate School of Fundamental Physics. M.S. was supported by the DFG via Project No. STA283/16-2.

TABLE IV. SU(2) Neural network. A, B, C determine the receptive field size of each neuron in the averaging layer.

Layer	Output shape	Kernel size
InputLayer	(1024, 16)	
Convolution1D	$(1024/(A),n_A)$	A
Convolution1D	$(1024/(A \times B),n_B)$	B
Convolution1D	$(1024/(A \times B \times C),n_C)$	C
Average pooling	$(1,n_C)$	
Flatten	(n_C)	
Dense	(n_D)	
Dense	(1)	

APPENDIX A: MONTE CARLO SIMULATIONS

In statistical physics and lattice gauge theory, Markov Chain Monte Carlo algorithms are used to sample lattice configurations from the Boltzmann factor. This is done by constructing a stochastic sequence that starts at some random initial configuration. This stochastic sequence is constructed such that the configurations obey Boltzmann statistics in the equilibrium. For more details on algorithm requirements and algorithms see, e.g., [55].

Observables are then computed by taking the average over many spin or lattice configurations from the equilibrium distribution

$$\langle \mathcal{O} \rangle = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathcal{O}_i. \tag{A1}$$

Taking the limit in the last equality is practically not possible. Hence, the expectation value of the observable is approximated by large N and gives rise to a statistical error. It is important to take enough configurations such that ergodicity is achieved. In the case of two distinct regions of phase space, this can take a very long simulation time.

For the Ising model, we produced a total of 55 000 spin configurations, of size 28×28, equally distributed over 11 equidistant temperature values $T \in [0,5]$ by employing the Metropolis-Hastings algorithm [56] with simulated annealing.

For SU(2), we used the Heatbath algorithm [57] to produce a total of 15 600 decorrelated configurations equally distributed over 26 values in the range of $\beta_{\text{latt}} = 4/g^2 \in [1,3.5]$. In the context of this paper it is important to have decorrelated data, since neural networks are good at finding structures, and thus correlations between configurations, if existent. Due to center symmetry breaking, in the deconfined phase the average Polyakov loop can take either positive or negative values of equal magnitude. In our simulations, we initiated all links with the unit matrix, hence we introduced a bias for large values of β_{latt}, i.e., our simulations are not fully ergodic. At large values of β_{latt}, this will prevent a full exploration of phase space. If we were to employ neural networks to extract the position of the phase transition, this nonergodicity leads to a shift in the value of critical β_{latt}. Generally speaking, ergodicity can be retained by doing more simulations and employing algorithms such as

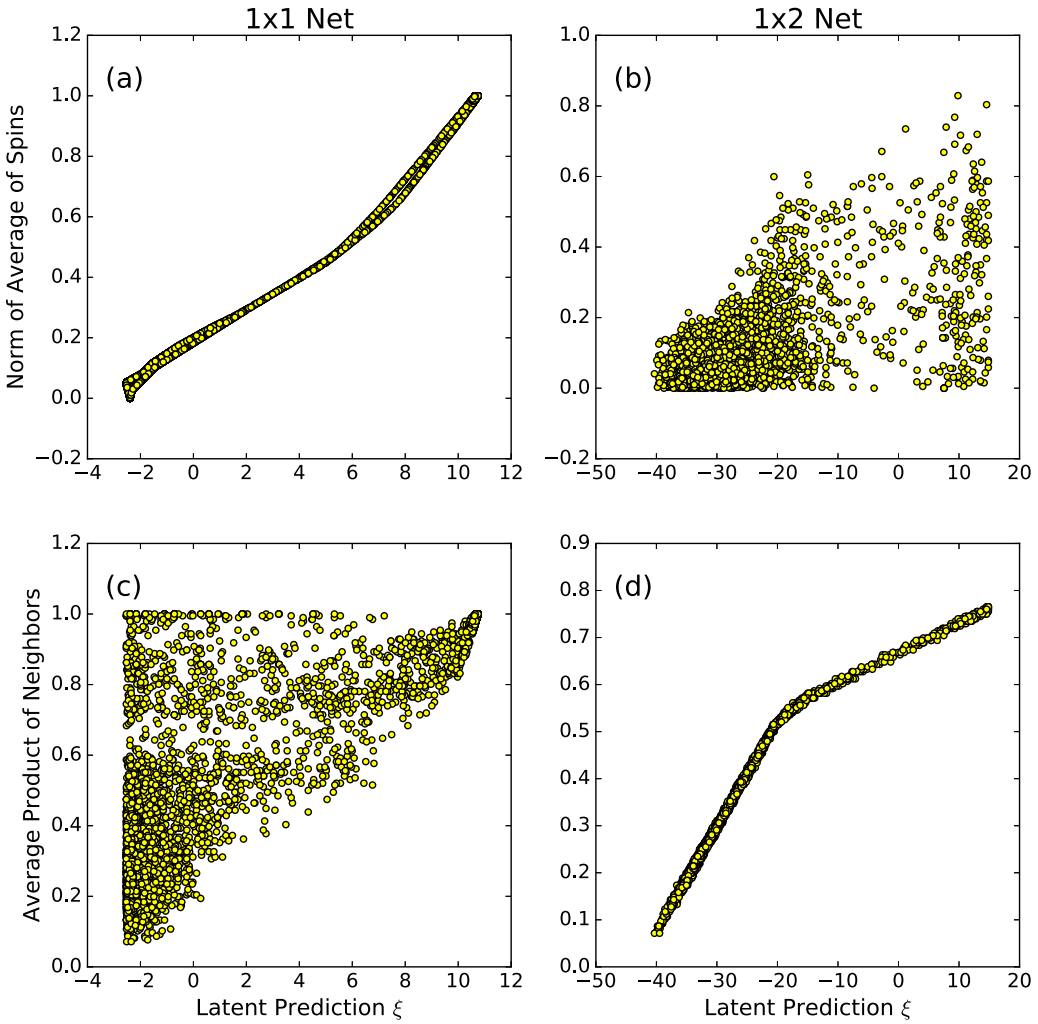

FIG. 4. Results of the correlation probing neural network applied to the Ising model. (a) The latent prediction is perfectly correlated with the absolute value of the average of spins, i.e., the magnetization in the 1×1 network. (b) The latent prediction of the 1×2 network is not correlated with the absolute value of the average of spins. (d) The latent prediction of the 1×2 network is perfectly correlated with the average product of neighbors, i.e., the expected energy per site. (c) The latent prediction of the 1×1 network is not correlated with the average of neighbors.

simulated annealing or overrelaxation, thus in principle it is possible to extract the critical temperature reliably.

APPENDIX B: NEURAL NETWORK ARCHITECTURE

We constructed our machine learning pipeline using Scikit-learn [58] and Keras [59]. The neural network architectures are presented in Tables III and IV. Since there is no Convolutional4D in Keras, we just rearranged our samples to fit a Convolutional1D layer. We used neural networks with number of filters $n_A,n_B,n_D \in \{1,4,8,32,256,1024\}$. The kernel sizes A, B, C are used to set the receptive field size. For our problems, $n_C = 1$ is sufficient to capture the structure of the order parameter. This was probed in the same manner as finding the optimal receptive field size. In other models one might need a higher n_C, e.g., in the Heisenberg model, $n_C = 3$ could be optimal. Hence, this can already be an early indicator for the type of the broken symmetry. The activation functions are rectified linear units $\mathrm{relu}(x) = \max(0,x)$ between all layers and the sigmoid function $\mathrm{sigmoid}(x) = 1/[1 + \exp(-x)]$ in the last layer. We do not employ any sort of regularization. The training objective is minimizing the binary cross entropy loss function

$$C(Y,P) = -\frac{1}{N}\sum_i [y_i \ln p_i + (1-y)\ln(1-p_i)], \qquad \text{(B1)}$$

where $Y = y_i$ is a list of labels and $P = p_i$ is the corresponding list of predictions. Our baseline classifier is the classifier which assigns each label with a probability of $p_i = 0.5$. This means that this classifier just assigns a label to each sample randomly. The binary cross entropy then evaluates to 0.6931. The neural networks learn by optimizing the weights and biases via RMSprop gradient descent. The neural networks were trained for 300 epochs or less, if the loss already saturated in earlier epochs. The validation set is 20% of the training data.

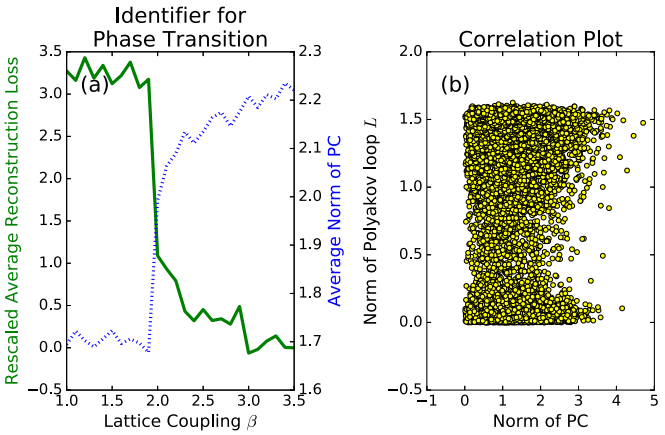

FIG. 5. (a) Finding a possible phase transition in SU(2) lattice gauge theory with PCA. Green solid: The average mean squared error reconstruction loss as a function of temperature is a universal identifier for a phase transition. It was calculated in 100 independent incremental PCA runs with two principal components (PC), measured in units of $\times 10^{-5}$ and shifted by the value at $\beta = 3.5$. Blue dotted: The average norm of the PC also indicates a phase transition. (b) There is no correlation between the principal components and the Polyakov loop.

APPENDIX C: CROSS COMPARISON OF ISING MODEL NEURAL NETWORK RESULTS

In Fig. 4 we show that latent parameters of the neural networks applied to the Ising model cannot be simultaneously correlated with the magnetization and the average energy per spin.

APPENDIX D: UNSUPERVISED LEARNING OF PHASE TRANSITIONS IN SU(2) LATTICE GAUGE THEORY

We assume no prior knowledge of the phase transition, even its existence. Hence, we employ unsupervised learning to find any possible indications for a phase transition. For the sake of simplicity we employ principal component analysis (PCA) [32,60] with two principal components. PCA is an orthogonal linear transformation of the input samples to a set of variables, sorted by their variance. Here, unsupervised learning algorithms that are based on the reconstruction loss like autoencoders [33] are doomed to fail, since the states are gauge invariant. The autoencoder would need to predict a matrix which is not unique.

Even though the Polyakov loop is a nonlinear order parameter, PCA captures indications of a phase transition at $\beta \in [1.8, 2.2]$, which is demonstrated in Fig. 5(a). Here we employed the average reconstruction loss [33] and the Euclidean norm of the principal components as identifiers for a phase transition. In Fig. 5(b) we show that there is no correlation between the Polyakov loop and the principal components.

It is worth noting that this example shows that PCA can capture phase indicators even when the principal components cannot approximate any order parameter.

APPENDIX E: REGRESSION OF THE POLYAKOV LOOP IN THE LOCAL NEURAL NETWORK

We perform a regression on the latent prediction of the local neural network on only 1% of the local samples of size $2 \times 1 \times 1 \times 1$ and use another 1% as validation set. By comparing different algorithms, we find that a second-order polynomial regression gives the best results; see Table V.

APPENDIX F: FULL REGRESSION OF THE POLYAKOV LOOP IN THE GLOBAL NEURAL NETWORK

Here we present the general procedure for reconstructing the decision function of a neural network applied to SU(2) gauge theory. Since it requires separating the correlation probing network into subnetworks, and transferring weights between different networks, it requires more advanced knowledge of artificial neural network architecture.

The decision function of the $2 \times 1 \times 1 \times 1$ neural network which predicts the lattice SU(2) phase transition is by construction

$$D(S) = F\Bigg(\frac{2}{N}\sum_{\vec{x}} f\big(U_\tau^{0,\vec{x}}, U_x^{0,\vec{x}}, U_y^{0,\vec{x}}, U_z^{0,\vec{x}}, U_\tau^{1,\vec{x}}, U_x^{1,\vec{x}}, U_y^{1,\vec{x}}, U_z^{1,\vec{x}}\big)\Bigg). \tag{F1}$$

In general, we cannot assume that the prediction network consists only of the output neuron. Therefore, we suggest a different procedure for constructing the decision function. We split the full correlation probing net into subnetworks: we extract the localization network plus averaging layer and the prediction network as separate networks. In order to determine $F(S) = \mathrm{sigmoid}(\xi(S))$, we use polynomial regression to fit the latent prediction of the prediction network to the output of the averaging layer. We find a polynomial of degree 1 is enough to fit the data, and ξ is approximated by

$$\xi(x) \approx -0.7101\, x + 9.851\,434\,19. \tag{F2}$$

The slope and intercept can be absorbed by the weight w and bias b of the output neuron, such that we can infer

$$\xi(x) \approx w\, x + b. \tag{F3}$$

The function f requires us to build a new local neural network which only acts on patches of size $2 \times 1 \times 1 \times 1$. By construction this network has the same number of weights and biases as the full neural network acting on the input of size $2 \times 8 \times 8 \times 8$. Instead of training the local neural network, we transfer the weights and biases from the full correlation probing network to the local neural network. Hence, one can obtain the output of the localization network for each patch separately. Again, we employ polynomial regression to fit the input from the local patches to the output of the localization network. The result of a regression of degree 2 with 561 parameters yields

$$\begin{aligned} f\big(\{U_\mu^{x_0}\}\big) \approx &- 26.8354\, a_\tau^0 a_\tau^1 - 2.4972\, d_\tau^0 c_\tau^1 + \cdots \\ &+ 1.5653\, b_\tau^0 c_\tau^0 + 26.5908\, b_\tau^0 b_\tau^1 \\ &+ 27.7054\, c_\tau^0 c_\tau^1 + 27.8939\, d_\tau^0 d_\tau^1. \end{aligned} \tag{F4}$$

After absorbing overall factors and the intercept by the weights and biases of the prediction network and neglecting the subleading terms, we rewrite f as

$$f\left(\left\{U_\mu^{x_0}\right\}\right) \approx a_\tau^0 a_\tau^1 - b_\tau^0 b_\tau^1 - c_\tau^0 c_\tau^1 - d_\tau^0 d_\tau^1. \quad \text{(F5)}$$

This is the Polyakov loop on a single lattice site. By employing (F5) as an argument of (F3), we can promote $f(\{U_\mu^{x_0}\}) \to f(\{U_\mu^{x_0,\vec{x}}\})$ to depend on space again. We obtain the definition of the decision function

$$D(S) \approx \text{sigmoid}\left[w\left(\frac{2}{N} \sum_{\vec{x}} f\left(\left\{U_\mu^{x_0,\vec{x}}\right\}\right)\right) + b \right], \quad \text{(F6)}$$

where $Q(S) = [\frac{2}{N}\sum_{\vec{x}} f(\{U_\mu^{x_0,\vec{x}}\})]$ is the Polyakov loop on the full lattice.

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, in *Advances in Neural Information Processing Systems 25*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., Lake Tahoe, Nevada, USA, 2012), pp. 1097–1105.
[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, IEEE Signal Process. Mag. **29**, 82 (2012).
[3] D. Silver *et al.*, Nature (London) **529**, 484 (2016).
[4] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[5] G. Torlai and R. G. Melko, Phys. Rev. B **94**, 165134 (2016).
[6] D.-L. Deng, X. Li, and S. Das Sarma, arXiv:1609.09060.
[7] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. X **7**, 021021 (2017).
[8] X. Gao and L.-M. Duan, Nat. Commun. **8**, 662 (2017).
[9] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, arXiv:1703.05334.
[10] K.-I. Aoki and T. Kobayashi, Mod. Phys. Lett. B **30**, 1650401 (2016).
[11] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, J. High Energy Phys. 05 (2017) 006.
[12] L. Huang and L. Wang, Phys. Rev. B **95**, 035105 (2017).
[13] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, Phys. Rev. B **95**, 041101(R) (2017).
[14] N. Portman and I. Tamblyn, J. Comput. Phys. **350**, 871 (2017).
[15] L. Li, T. E. Baker, S. R. White, and K. Burke, Phys. Rev. B **94**, 245129 (2016).
[16] Y. Levine, D. Yakira, N. Cohen, and A. Shashua, arXiv:1704.01552.
[17] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature **549**, 195 (2017).
[18] S. Lloyd, M. Mohseni, and P. Rebentrost, Nat. Phys. **10**, 631 (2014).
[19] M. Kieferova and N. Wiebe, arXiv:1612.05204.
[20] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, arXiv:1612.05695.
[21] H. W. Lin, M. Tegmark, and D. Rolnick, J. Stat. Phys. **168**, 1223 (2017).
[22] E. Miles Stoudenmire and D. J. Schwab, *Advances in Neural Information Processing Systems 29*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Curran Associates, Inc., New York, 2016), pp. 4799–4807.
[23] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, arXiv:1601.02036.
[24] P. Mehta and D. J. Schwab, arXiv:1410.3831.
[25] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[26] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Sci. Rep. **7**, 8823 (2017).
[27] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[28] Y. Zhang and E.-A. Kim, Phys. Rev. Lett. **118**, 216401 (2017).
[29] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).
[30] A. Tanaka and A. Tomiya, J. Phys. Soc. Jpn. **86**, 063001 (2017).
[31] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).
[32] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[33] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[34] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).
[35] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).
[36] G. Towell and J. W. Shavlik, in *Advances in Neural Information Processing Systems 4*, edited by J. E. Moody, S. J. Hanson, and R. P. Lippmann (Morgan-Kaufmann, Denver, CO, USA, 1992), pp. 977–984.
[37] A. Mahendran and A. Vedaldi, arXiv:1412.0035.
[38] Z. Ghahramani, Nature (London) **521**, 452 (2015).
[39] Z. C. Lipton, arXiv:1606.03490.
[40] R. Shwartz-Ziv and N. Tishby, arXiv:1703.00810.
[41] P. Ponte and R. G. Melko, arXiv:1704.05848.
[42] L. Onsager, Phys. Rev. **65**, 117 (1944).
[43] K. G. Wilson, Phys. Rev. D **10**, 2445 (1974).
[44] M. G. Alford, A. Schmitt, K. Rajagopal, and T. Schäfer, Rev. Mod. Phys. **80**, 1455 (2008).
[45] M. A. Stephanov, Int. J. Mod. Phys. A **20**, 4387 (2005).
[46] J. B. Kogut and M. A. Stephanov, Cambridge Monogr. Part. Phys., Nucl. Phys., Cosmol. **21**, 1 (2004).
[47] C. M. Varma, Phys. Rev. B **73**, 155113 (2006).
[48] L. Taillefer, Annu. Rev. Condens. Matter Phys. **1**, 51 (2010).
[49] Y. Wang, N. P. Ong, Z. A. Xu, T. Kakeshita, S. Uchida, D. A. Bonn, R. Liang, and W. N. Hardy, Phys. Rev. Lett. **88**, 257003 (2002).
[50] S. Hüfner, M. A. Hossain, A. Damascelli, and G. A. Sawatzky, Rep. Prog. Phys. **71**, 062501 (2008).
[51] S. Friederich, H. C. Krahl, and C. Wetterich, Phys. Rev. B **83**, 155125 (2011).
[52] T. A. Maier, M. Jarrell, T. C. Schulthess, P. R. C. Kent, and J. B. White, Phys. Rev. Lett. **95**, 237001 (2005).
[53] S. Raghu and S. A. Kivelson, Phys. Rev. B **83**, 094518 (2011).
[54] C. J. Halboth and W. Metzner, Phys. Rev. Lett. **85**, 5162 (2000).
[55] C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice* (Springer, Berlin, 2010).
[56] N. Metropolis and S. Ulam, J. Am. Stat. Assoc. **44**, 335 (1949).
[57] M. Creutz, Phys. Rev. D **21**, 2308 (1980).
[58] F. Pedregosa *et al.*, J. Machine Learn. Res. **12**, 2825 (2011).
[59] F. Chollet, "keras", https://github.com/fchollet/keras (2015).
[60] K. Pearson, Philos. Mag. J. Ser. **2**, 559 (1901).