

# Towards Novel Insights in Lattice Field Theory with Explainable Machine Learning

Stefan Blücher,<sup>1</sup> Lukas Kades,<sup>1</sup> Jan M. Pawłowski,<sup>1,2</sup> Nils Strodthoff,<sup>3</sup> and Julian M. Urban<sup>1</sup>

<sup>1</sup>*Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 16, 69120 Heidelberg, Germany*

<sup>2</sup>*ExtreMe Matter Institute EMMI, GSI, Planckstraße 1, D-64291 Darmstadt, Germany*

<sup>3</sup>*Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*

Machine learning has the potential to aid our understanding of phase structures in lattice quantum field theories through the statistical analysis of Monte Carlo samples. Available algorithms, in particular those based on deep learning, often demonstrate remarkable performance in the search for previously unidentified features, but tend to lack transparency if applied naively. To address these shortcomings, we propose representation learning in combination with interpretability methods as a framework for the identification of observables. More specifically, we investigate action parameter regression as a pretext task while using layer-wise relevance propagation (LRP) to identify the most important observables depending on the location in the phase diagram. The approach is put to work in the context of a scalar Yukawa model in (2+1)d. First, we investigate a multilayer perceptron to determine an importance hierarchy of several predefined, standard observables. The method is then applied directly to the raw field configurations using a convolutional network, demonstrating the ability to reconstruct all order parameters from the learned filter weights. Based on our results, we argue that due to its broad applicability, attribution methods such as LRP could prove a useful and versatile tool in our search for new physical insights. In the case of the Yukawa model, it facilitates the construction of an observable that characterises the symmetric phase.

## I. INTRODUCTION

Lattice simulations of quantum field theories have proven essential for the theoretical understanding of fundamental interactions from first principles, perhaps most prominently so in quantum chromodynamics. However, an in-depth understanding of the emergent dynamics is often difficult. In cases where such an understanding remains elusive, it may be instructive to search for so far unidentified structures in the data to better characterise the dynamics.

In this quest towards new physical insight, we turn to machine learning (ML) approaches, in particular from the subfield of deep learning [1]. These methods have proven capable of efficiently identifying high-level features in a broad range of data types—in many cases, such as speech or image recognition, with spectacular success [2–5]. Accordingly, there is growing interest in the lattice community to harness the capabilities of these algorithms, both for high energy physics and condensed matter systems. Applications include predictive objectives, such as detecting phase transitions from lattice configurations, as well as generative modeling [6–35]. We recommend [36] as an introduction to ML for physicists and [37] as a general review for ML applications in physics.

One ansatz for the identification of relevant observables from lattice data is through representation learning, i.e. by training on a pretext task. The rationale behind this approach is that the ML algorithm learns to recognise patterns which can be leveraged to construct observables from low-level features that characterise different phases. However, solving a given task does by itself not lead to physical insights, since the inner structure of the algorithm typically remains opaque. This issue can at least partially be resolved by the use of “explainable AI” tech-

niques, which have recently attracted considerable interest in the ML community and beyond. In this work, we focus on layer-wise relevance propagation (LRP) [38]. It is one of several popular post-hoc attribution methods that propagate the prediction back to the input domain, thereby highlighting features that influence the algorithm towards/against a particular classification decision.

We test this approach in the context of Yukawa theory in (2+1) dimensions, using inference of an action parameter as a pretext task in order to identify relevant observables. In a first step, we demonstrate that this is at least partially possible by training a multilayer perceptron (MLP) on a set of standard observables. Here, we show that the relevance of features in different phases, as determined by LRP, agrees with physical expectations. We benchmark our results with a similar method based on random forests. Subsequently, we demonstrate that the action parameter can be inferred directly from field configurations using a convolutional neural network (CNN). We use LRP to identify relevant filters and discuss how these align with physical knowledge. This also allows us to construct an observable that appears to be a distinctive feature of the paramagnetic phase.

The paper is organised as follows. In Section II we briefly review scalar Yukawa theory on the lattice and define important quantities. Section III serves as an introduction to the topic of explainable AI and discusses LRP in order to convey the rationale behind our approach. Numerical results for the MLP and a random forest benchmark are presented in Section IV A. In Section IV B we conduct an analysis of the CNN and subsequently demonstrate how all order parameters, as well as the aforementioned observable relevant for the paramagnetic phase, can be extracted from the filters. We discuss our findings and possible future work in Section V.

## II. YUKAWA THEORY

We consider a scalar Yukawa model defined on a  $(2+1)d$  cubic lattice with periodic boundary conditions. The theory is comprised of a real-valued scalar field with quartic self-interaction coupled to Dirac fermions. The action for the scalar field can be cast into the following dimensionless form,

$$S_{\text{KG}}[\phi] = \sum_{n \in \Lambda} \left[ -2\kappa \sum_{\mu=1}^d \phi(n) \phi(n + \hat{\mu}) + (1 - 2\lambda) \phi(n)^2 + \lambda \phi(n)^4 \right], \quad (1)$$

where  $\Lambda$  denotes the set of all lattice sites. Here,  $\kappa$  is called the hopping parameter and  $\lambda$  takes the role of the coupling constant.

In order to ensure positivity of the partition function, one needs a minimum of two degenerate fermion flavors. Due to their bilinear contributions to the action, the fermionic d.o.f. can be integrated out, yielding the determinant of the discretised Dirac operator,

$$D_{nm}[\phi] = \sum_{\mu=1}^d \eta_{\mu}(n) \frac{\delta(n - m + \hat{\mu}) - \delta(n - m - \hat{\mu})}{2} + \delta(n - m) (M_f + g \phi(n)), \quad (2)$$

as a multiplicative contribution to the statistical weight. The Euclidean Dirac  $\gamma$ -matrices are absorbed by the staggered transformation, yielding the scalars  $\eta_{\mu}(n)$  that mix the spatial and spinor degrees of freedom. They are given by  $\eta_1(n) = 1$  and  $\eta_l(n) = (-1)^{n_1} \dots (-1)^{n_{l-1}}$ .  $M_f$  denotes the fermion mass and  $g$  is the Yukawa coupling to the bosonic field. The expectation value of an observable  $\mathcal{O}$  can then be expressed as the path integral

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}\phi \det(D^T D) \exp(-S_{\text{KG}}) \mathcal{O}(\phi), \quad (3)$$

where  $\mathcal{Z}$  denotes the partition function. Important observables characterising phases and critical phenomena in scalar  $\phi^4$ -theory include the magnetisation,

$$M = \frac{1}{|\Lambda|} \sum_{n \in \Lambda} \phi(n), \quad (4)$$

as well as the staggered magnetisation

$$M_s = \frac{1}{|\Lambda|} \sum_{n \in \Lambda} (-1)^{n_1 + \dots + n_d} \phi(n), \quad (5)$$

which is relevant for negative  $\kappa$ . The scalar part of Equa-

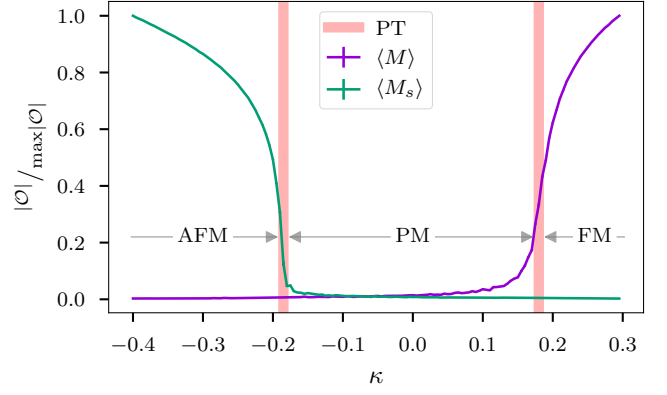


FIG. 1: Slice of the phase diagram for fixed Yukawa coupling  $g = 0.25$  using normalised values of  $\langle M \rangle$  and  $\langle M_s \rangle$ . Phase transitions are highlighted by the shaded bars. We distinguish an antiferromagnetic (AFM), a paramagnetic (PM) and a ferromagnetic (FM) phase.

tion (1) features the additional staggered symmetry

$$\kappa \mapsto -\kappa \quad \text{and} \quad \phi(n) \mapsto (-1)^n \phi(n), \quad (6)$$

which connects both magnetisations. The fermionic part explicitly breaks this symmetry.

A slice of the phase diagram at fixed Yukawa coupling is shown in Figure 1. The theory exhibits an interesting structure, with two broken phases of ferromagnetic (FM) and antiferromagnetic (AFM) nature, where  $M$  and  $M_s$  respectively acquire non-zero expectation values. They are separated by a symmetric, paramagnetic (PM) phase, where both quantities vanish.

We also consider the connected two-point correlation function

$$G_c(n, m) = \langle \phi(n) \phi(m) \rangle - \langle \phi(n) \rangle \langle \phi(m) \rangle. \quad (7)$$

While the expectation value of the magnetisation can be estimated from a single field configuration at reasonable lattice sizes, signals of n-point correlators are naturally much more suppressed due to statistical noise and cannot be reasonably approximated from one sample. Therefore, we introduce the time-sliced correlator  $G_c(t)$ , which is defined by

$$G_c(t) = \frac{1}{|\Lambda|} \sum_{\vec{n}} G_c((t, \vec{n}), (0, \vec{0})), \quad (8)$$

where the sum runs over spacelike components. It measures correlations only in the temporal direction, which leads to a better signal-to-noise ratio due to the averaging procedure.

Some aspects of the derivation and simulation are given in Appendix A. For a comprehensive treatment of Yukawa theory on the lattice, we recommend [39].

### III. INSIGHTS FROM EXPLAINABLE AI

Simple methods from statistics and ML often lack the capability to model complex data, whereas sophisticated algorithms typically tend to be less transparent. A commonly used example is principal component analysis (PCA). It has been successfully applied to the extraction of (albeit already known) order parameters for various systems [6, 9, 12]. However, its linear structure prohibits the identification of complex non-linear features, e.g. Wilson loops in gauge theories. Hence, we require tools capable of modeling non-linearities, such as deep neural networks [17]. They allow for a more comprehensive treatment of complex systems, which has been demonstrated e.g. for fermionic theories in [7, 11]. The approach also enables novel procedures, such as learning by confusion and similar techniques, to locate phase transitions in a semi-supervised manner [15, 34]. For lattice QCD, action parameters can be extracted from field configurations [24]. Overall, deep learning tools seem particularly well-suited to grasp relevant information about quantum field dynamics in a completely data-driven approach, by learning abstract internal representations of relevant features.

However, their lack of transparency is frequently a major drawback of using such methods, which prohibits access to and comprehension of these representations. A unified understanding of how and what these architectures learn, and why it seems to work so well in a wide range of applications, is still pending. To better understand the processes behind neural network-driven phase detection in lattice models, multiple proposals have been made, such as pruning [8, 25, 31], utilising (kernel) support vector machines [16, 32], and saliency maps [33].

Also, in the broader scope of ML research, there has been growing interest in interpretability approaches, most of them focusing on post-hoc explanations for trained models. So-called attribution methods typically assign a relevance score to each of the input features that quantifies—depending on the attribution algorithm—which features the classifier was particularly sensitive to, or influenced the algorithm towards/against an individual classification decision. In the domain of image recognition, such attribution maps are typically visualised as heatmaps overlaying the input image. The development of attribution algorithms is a very active field of research in the ML community. Therefore, we refer to dedicated research articles for more in-depth treatments [40, 41]. Very broadly, the most important types of such local interpretability methods can be categorised as: 1. Gradient-based, such as saliency maps [42] obtained by computing the derivative of a particular class score with respect to the input features or integrated gradients [43]. 2. Decomposition-based, such as layer-wise relevance propagation (LRP) [38] or DeepLift [44]. 3. Perturbation-based, as in [45], investigating the change in class scores when occluding parts of the input features.

In this work, we focus on LRP, as mentioned a particu-

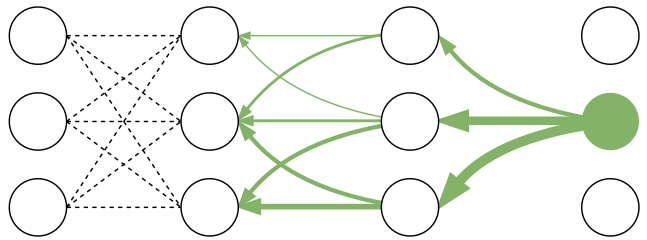


FIG. 2: Sketch of LRP through the last two layers of a classification network that predicts one-hot vectors. Relevance is indicated by arrow width. The conservation law requires the sum of widths to remain constant during backpropagation. Diagram adapted from [46].

lar variant of decomposition-based attribution methods. However, we stress that the qualitative finds do not rely on the specific choice of method. The general idea of LRP is to start from a relevance assignment in the output layer and subsequently propagate this relevance back to the input in a layer-wise fashion using certain propagation rules, see the sketch in Figure 2 and Appendix C for details. In this way, the method assigns a relevance score to each neuron, where positive (negative) entries strongly influence the classifier towards (against) a particular classification decision.

### IV. RESULTS

In this section, numerical results are presented which corroborate our rationale. We train a multilayer perceptron (MLP) and a convolutional neural network (CNN) to infer the associated hopping parameter  $\kappa$  from a set of known observables (Approach A), as well as solely from the raw field configurations (Approach B), akin to [24]. In the first case, without providing any prior knowledge of the phase boundaries, LRP manages to reveal the underlying phase structure and returns a phase-dependent importance hierarchy of the observables that is in accordance with physical expert knowledge. In the second case, by calculating the relevances of the learned filters, we can associate each of them with one of the physical phases and thereby extract the known order parameters. Moreover, it facilitates the construction of an observable that characterises the symmetric phase. Both variants of our strategy are sketched in Figure 3. Predicting action parameters from samples is an ill-conditioned inverse problem, since the same field configuration can in principle be sampled from different combinations of couplings. Hence, the optimisation objective is formulated in terms of maximum likelihood estimation. Assuming a Gaussian distribution with fixed variance, this objective reduces to minimising the mean squared error (MSE), which we use as loss function in the following. In addition, we apply weight regularisation, see Appendix E for details.

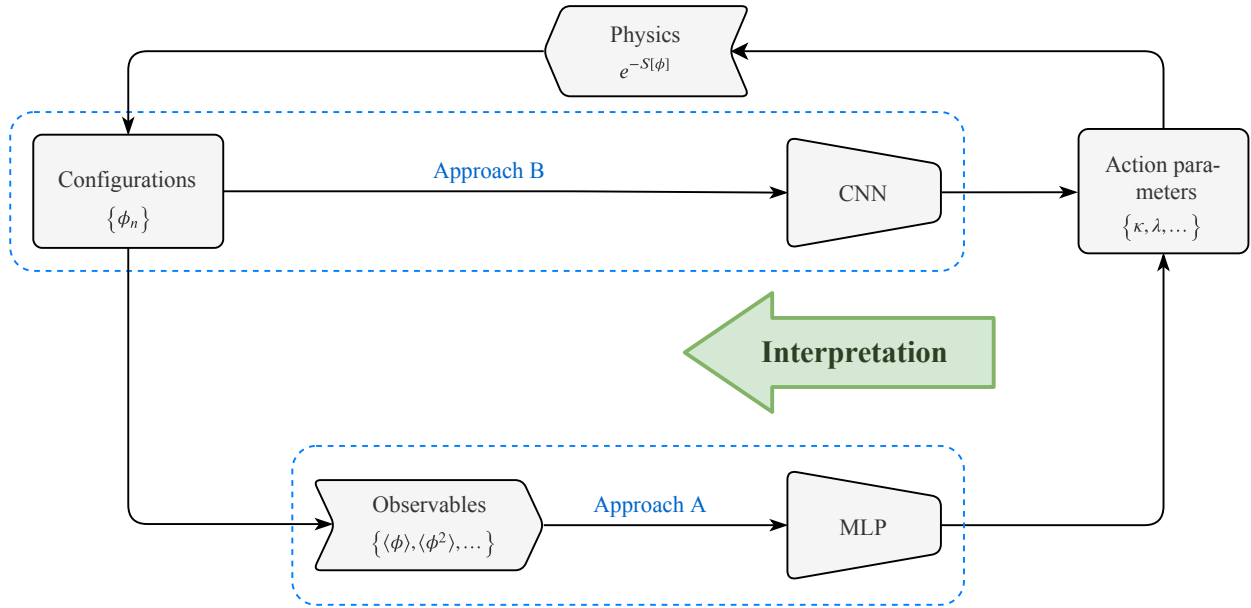


FIG. 3: Sketch of our strategy to learn meaningful structures from the simulation data by analysing the networks trained for action parameter inference. Field configurations used for training are either preprocessed into observables for the MLP (Approach A) or directly operated upon with a CNN (Approach B). Obtaining accurate predictions for the parameters indicates approximate cycle consistency in the above diagram, which supports the notion that the networks have successfully identified characteristic features. These can then be extracted in a subsequent interpretation step using LRP.

### A. Importance Hierarchies of Known Observables

In Section II we introduced a set of standard observables, consisting of the normal and staggered magnetisation as well as the time-sliced two-point correlation function.<sup>1</sup> It seems reasonable to assume that much of the relevant information characterising the phase structure and dynamics of the theory is encoded in these quantities. To check this, we create an ordered dataset of measurements of these quantities at various, evenly spaced values of  $\kappa$  (see Appendix B for details on the dataset) and use it to perform a regression analysis. We employ a MLP, also called fully-connected neural network (see Appendix E for details on the specific architecture). The method is compared against a random forest regressor as a baseline, which is a standard method based on the optimisation of decision trees [47] (see Appendix D for details). The results for both approaches, shown in Figures 4 and 5, will be discussed in the following.

We observe qualitatively similar accuracy on the training and test data in the broken FM and AFM phases. This is expected, since we know from Figure 1 that always one of the two types of magnetisations is strictly

monotonic in the respective phase and can therefore determine  $\kappa$  uniquely. However, both approaches yield at best mediocre performance in the symmetric PM phase. Here, both magnetisations tend to zero and therefore do not contain much relevant information. Moreover, the two-point correlator exhibits approximately symmetric properties around  $\kappa = 0$ . Therefore, it also does not provide a unique mapping. This issue is resembled in the prediction for both methods. The random forest yields a symmetric discrepancy around  $\kappa = 0$ . In comparison, the MLP shows an improved performance for  $\kappa < 0$ , albeit at the price of a larger variance for  $\kappa > 0$ . At this point, we can already see that the chosen set of observables suffices to characterise the theory only in the broken phases, whereas in the symmetric phase, additional information appears to be necessary.

Before we embark on the search for the missing piece, let us first examine the results further to verify that the learned decision rules conform to the physical interpretation given above. We begin with the relevances as determined by LRP, shown in Figure 4 (bottom), and later compare to the random forest benchmark below. As expected,  $M$  and  $M_s$  are relevant in the FM and AFM phases, respectively. There, considerable relevance is also assigned to the observable  $G_c(t=0)$ . However, the contribution appears to diminish when going deeper into the broken phases. Its comparably large relevance in the symmetric PM phase shows that it contains most of the information used for the noisy prediction. As de-

<sup>1</sup> We use a slightly modified definition of the time-sliced correlator in order to remove lattice artifacts from the data, see Appendix B.

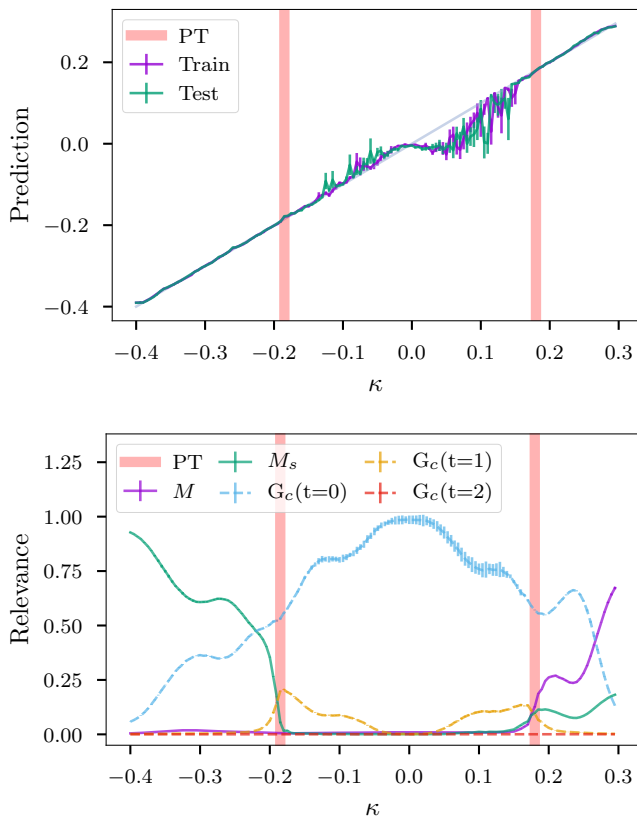


FIG. 4: Results for the MLP. Top: predictions, bottom: normalised LRP relevances of individual features. Error bars here and throughout this work are obtained with the statistical jackknife method.

scribed above, the mediocre performance in this phase indicates that although the network seems to find weak signals, the chosen set of observables cannot be optimal.

The interpretation sketched above is further supported by the results obtained through random forest regression [47]. Analogously to the previously introduced relevance for LRP, we can determine nominal contributions of input features to the prediction and hence a measure of local feature importance (see Appendix D for details), which is shown in Figure 5 (bottom). In the broken FM and AFM phases, the respective contributions of  $M$  and  $M_s$  demonstrate a linear dependence on  $\kappa$ . Again, this clearly indicates that these quantities characterise the associated phases. For the symmetric PM phase, the situation appears more challenging, since no such clear dependence is observed for any of the observables. The non-zero contributions of features in the PM phase imply that they add some valuable information to the decision here. However, this has to be weighted against the observation that the accuracy in this region is poor. This further confirms our previous conclusion that relevant information to characterise this phase is largely lost in the preprocessing step, assuming that it was initially present in the raw field configurations. It is worthwhile stressing that this analysis represents an independent confirmation

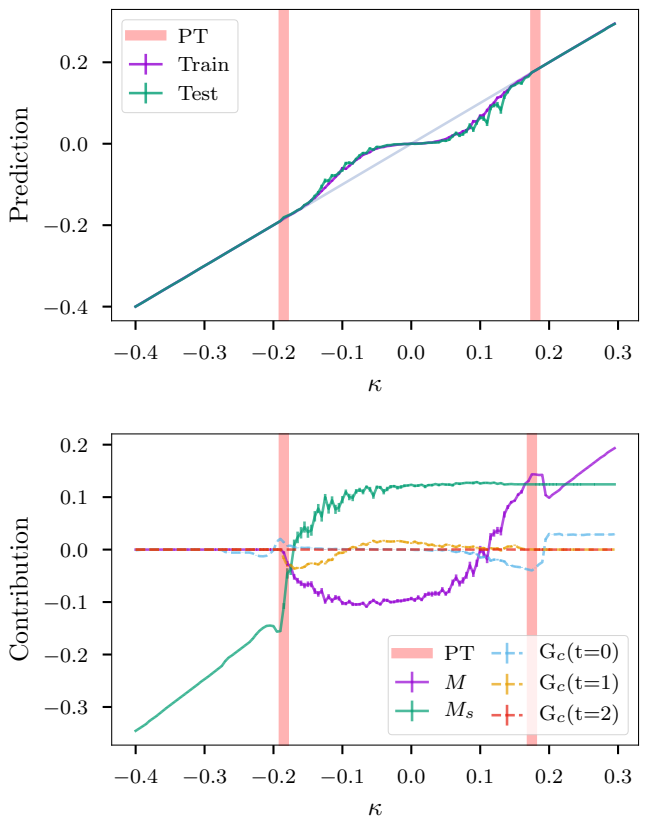


FIG. 5: Benchmark results for the random forest. Top: predictions, bottom: nominal contributions of individual features.

of the results obtained above. Both algorithms (MLP vs. random forest) rely on fundamentally different principles. We use a model-intrinsic interpretability measure for the random forest, whereas for the MLP we rely on LRP, i.e. a post-hoc attribution method.

## B. Extracting Observables from Convolutional Filters

In the previous section, we used a dataset of known observables to reconstruct  $\kappa$ . Calculating such quantities corresponds to heavy preprocessing of the high-dimensional field configuration data. The resulting low-dimensional features are far less noisy, implying distillation of relevant information. This is a common procedure in the field of data science, and may become unavoidable for large lattices and/or theories with more degrees of freedom. E.g. in state-of-the-art simulations of lattice QCD, the required memory to store a single field configuration can easily reach  $\mathcal{O}(10^9)$  floating point numbers. Nevertheless, using preprocessed data in the form of standard observables introduces strong biases towards known structures. If our perception of the problem or generally our physical intuition is flawed, machine learning cannot help us—the relevant information may very well be

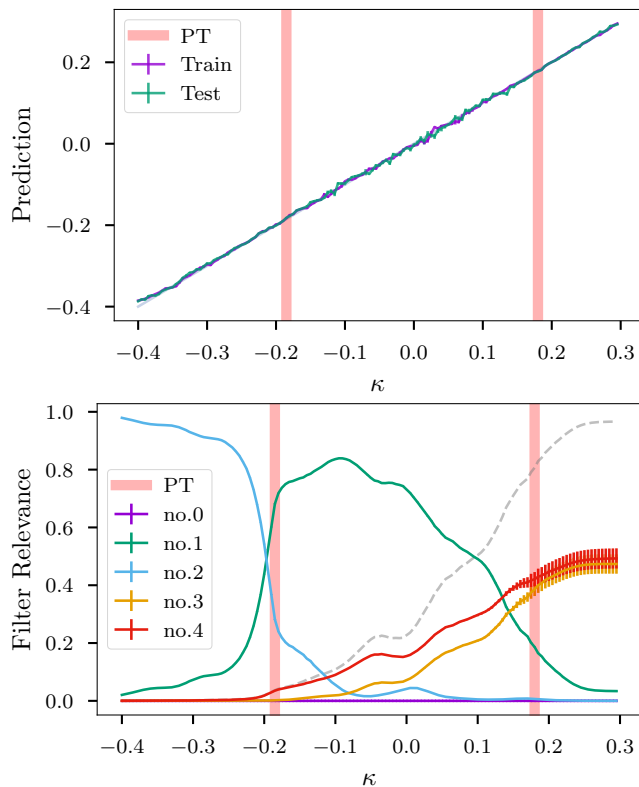


FIG. 6: Results for the CNN. Top: prediction, bottom: normalised relevances of individual filters. The dashed curve corresponds to the cumulative relevance of filter 3 and 4.

lost in the preprocessing step. In the present case specifically, it appears that important features in the PM phase are neglected by this procedure, assuming that structures characterising this phase do in fact exist. Therefore, it is instructive to search for signals of such structures by training neural networks directly on field configurations.

As a starting point for this search, we first perform a PCA on the field configuration dataset. As previously mentioned, this has been done before with promising results [6, 9, 12], albeit not in exactly the same physical setting. PCA immediately identifies the normal and staggered magnetisations as dominant features, essentially reproducing the work of [9]. All higher order principal components show a vanishing explained variance ratio, implying that no other relevant, purely linear features are present in the data. This observation indicates that, if a quantity exists which parametrises the symmetric PM phase, it cannot simply be a linear combination of the field variables.

Our improved approach is based on a convolutional neural network (CNN). The training procedure is largely equivalent to that for the MLP in the previous section, with the observable dataset replaced by the full field configurations. We train a CNN using five convolutional filters with a shape of  $2 \times 2 \times 2$  and a stride of 1. In order to support explainability, we encourage weight sparsity by

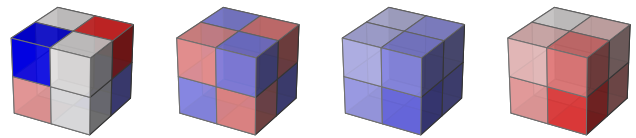


FIG. 7: Learned weights of convolutional filters. Left to right: (no.1, PM); (no.2, AFM); (no.3, FM); (no.4, FM). The colour map is symmetric around zero. Red (blue) corresponds to positive (negative) weights.

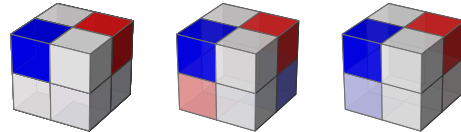


FIG. 8: Comparison of PM filters for three independent training runs of the CNN.

adding the  $L^1$  norm to the loss—also known as LASSO regularisation—as suggested in [33] (see Appendix E for details). Due to the nature of the convolution operation, learned filters have a direct interpretation i.t.o. first-order linear approximations of relevant observables. Hence, we expect the CNN to reproduce the PCA results at the very least, and aim for the identification of other, non-linear quantities, which the network can encode in subsequent layers. It is important to understand this difference between the approaches, even though both extract only linear signals in a first approximation.

The model predictions are shown in Figure 6 (top). We can immediately observe a superior performance in the PM phase compared to our previous results. The CNN succeeds to consistently infer  $\kappa$  from the field configuration data with high accuracy. This indicates that it indeed manages to construct internal representations suitable not only to discern the different phases, which would be sufficient for classification purposes, but also for an ordering of data points within each phase.

In order to interpret the predictions and extract knowledge about the learned representations, we have to customise LRP to our needs. In image recognition, as previously mentioned, one mostly aims at highlighting important regions in the input domain, leading to superimposed heatmaps. This is based on the inherent heterogeneity common to image data, where relevant features are usually localised. For field configurations on the lattice, due to the translational symmetry of the action and the resulting homogeneity, no particularly distinguished, localised region should be apparent in any given sample. However, each convolutional filter encodes an activation map that is in fact sensitive to a specific feature present in a lattice configuration. In contrast to the usual ansatz, the spatial homogeneity promotes global pooling over the relevances associated with each filter weight. Hence, instead of assigning relevances to input pixels, we are interested in the cumulative filter relevance which indicates their individual importance for a particular prediction.



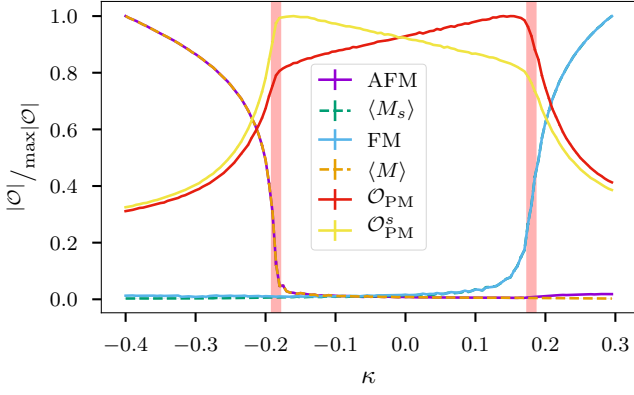


FIG. 9: Normalised observables reconstructed from the learned filters. The quantities associated with the FM and AFM phases are compared to  $M$  and  $M_s$ .  $\mathcal{O}_{\text{PM}}$  and  $\mathcal{O}_{\text{PM}}^s$  are related by Equation (6) and exhibit an approximate mirror symmetry around  $\kappa = 0$ .

Analogously to the rationale of the previous section, we can use this approach to build importance hierarchies of filters, thereby facilitating their physical interpretation as signals of relevant observables.

Figure 6 (bottom) shows each filter relevance as a function of  $\kappa$ . We can recognise some similarities to the relevances in Figure 4, highlighting the underlying phase structure of the Yukawa theory. It appears that the model can parametrise each phase individually using one or a small subset of filters, while the others show small or insignificant relevances in the respective region. The learned weight maps are shown in Figure 7, where we also assign names to the filters depending on the corresponding associated phase, with the exception of filter no.0 because it exhibits completely vanishing weights and relevance. It seems to have been dropped entirely by the network, indicating that four filters are sufficient to characterise all phases seen in the data. This reduction is an effect of the regularisation, and constitutes a recurring pattern also when more filters are initially used, providing a first hint towards the number of independent quantities utilised by the network.

Let us begin by examining the results that directly correspond to known quantities. We observe that the FM1 and FM2 filters have entries of roughly uniform magnitude with a globally flipped sign. Accordingly, we can identify them as signals of the negative and positive branches of the magnetisation  $M$ , respectively. This is corroborated by their dominating relevances in the FM phase. The AFM filter exhibits alternating entries of uniform magnitude and therefore corresponds to the staggered magnetisation  $M_s$ , which accordingly dominates the AFM phase. Hence, both order parameters can be explicitly reconstructed from the CNN. The appearance of two filters for the magnetisation is easily understood by inspection of the network architecture in Table III, the crucial point being the application of a ReLU activation after the convolution operation. Consider the action

of a positively-valued filter to negatively magnetised field configurations, or vice versa. The resulting negative activation map is subsequently defaulted to zero by the ReLU. Hence, in order to take both branches of  $M$  into account, two equivalent filters with opposing signs are required. The comparably large error bars in this region stem from the presence of positively and negatively magnetised samples in the dataset, which lead to a higher per-filter variance. Therefore, we additionally plot the cumulative relevance of both filters.

We now discuss the main object of interest, namely the PM filter. It supplies the dominant signal for the characterisation of this phase. A linear application of this filter to the configurations, as done for the FM and AFM filters, does not produce a monotonic quantity, which would be required for a unique ordering. This further supports the aforementioned evidence gathered by PCA for the absence of an additional, purely linear observable. Hence, the simple reconstruction scheme outlined in the previous paragraphs cannot be applied in this case. Instead, we undertake a heuristic attempt to reconstruct the relevant quantity. To this end, we note that the ReLU activation applied to the convolutional layer's output can effectively correspond to the absolute value function, albeit with less statistics, if the entries of the activation map are distributed accordingly. Inspired by this observation, we define the following observable,

$$\mathcal{O}_{\text{PM}} = \frac{1}{|\Lambda|} \sum_{n \in \Lambda} \left| \left[ \phi(n) + \phi(n + \hat{\mu}_1) \right] - \left[ \phi(n + \hat{\mu}_2 + \hat{\mu}_3) + \phi(n + \hat{\mu}_1 + \hat{\mu}_2 + \hat{\mu}_3) \right] \right|. \quad (9)$$

As with  $M$  and  $M_s$ , we obtain the corresponding staggered form  $\mathcal{O}_{\text{PM}}^s$  by applying the transformation given in Equation (6). The resulting pair of quantities is visualised by the following idealised filters.

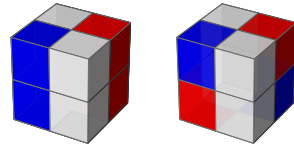


FIG. 10: Convolutional filters corresponding to the observable  $\mathcal{O}_{\text{PM}}$  defined in Equation (9) (left) and its corresponding staggered counterpart  $\mathcal{O}_{\text{PM}}^s$  (right).

The observable  $\mathcal{O}_{\text{PM}}$  defined in Equation (9) is the sum over all lattice sites of the lattice derivative in the diagonal  $\hat{\mu}_2 + \hat{\mu}_3$  direction of blocks in the  $\hat{\mu}_1$  direction. This already explains the modulus, as otherwise  $\mathcal{O}_{\text{PM}}$  would be the sum over all sites of a total derivative, which vanishes identically. We also remark that  $\mathcal{O}_{\text{PM}}$  can be made isotropic by summing over all directions.

We now discuss the properties of the theory that are measured by  $\mathcal{O}_{\text{PM}}$ : In the continuum limit,  $\mathcal{O}_{\text{PM}}$  naively tends towards the volume integral over  $|\nabla\phi|$ . Due to

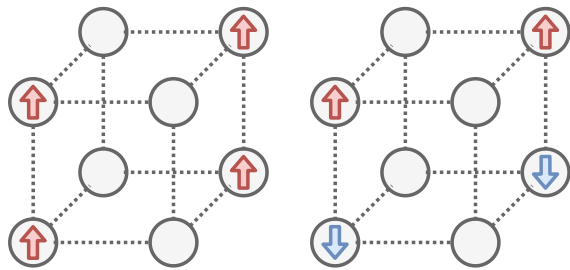


FIG. 11: Qualitative visualisation of local structures in field configurations operated upon by the PM filter. Sign is encoded by arrow orientation/colour. Diagonal neighbours tend to share the same sign everywhere in the phase diagram. On the contrary, nearest neighbours show a preference towards either same (left) or opposite (right) orientations.  $\mathcal{O}_{\text{PM}}$  is particularly sensitive towards the local presence/absence of such sign flips in the PM phase, without the need for a globally non-zero expectation value of the magnetisations.

the modulus of the derivative,  $\langle \mathcal{O}_{\text{PM}} \rangle$  carries the same information as the expectation value of the kinetic term.

The blocking in the  $\hat{\mu}_1$ -direction leads to a sensitivity of  $\mathcal{O}_{\text{PM}}$  to sign flips of nearest-neighbours. While no continuum observable is sensitive to these sign flips, the continuum limit of  $\langle \mathcal{O}_{\text{PM}} \rangle$  maintains this information. Accordingly,  $\langle \mathcal{O}_{\text{PM}} \rangle$  exhibits a distinct behavior in the presence of localised, (anti-)magnetised regions, even if the expectation values vanish globally. Possible local field alignments resulting in different values of  $\mathcal{O}_{\text{PM}}$ , but not of the standard derivative, are visualised in Figure 11.

The construction and discussed sensitivities of  $\langle \mathcal{O}_{\text{PM}} \rangle$  demonstrates again the usefulness of LRP: we can identify the learned representation as a feature of the dataset arising from the lattice discretisation.  $\langle \mathcal{O}_{\text{PM}} \rangle$  and  $\langle \mathcal{O}_{\text{PM}}^s \rangle$  as functions of  $\kappa$  are shown in Figure 9 together with the other reconstructed observables and their respective analytical counterparts. A monotonic, roughly linear dependence is observed in the PM phase, indicating that the quantity indeed provides a unique mapping which aids the  $\kappa$  inference. In fact, if  $\mathcal{O}_{\text{PM}}$  is included in the set of predefined observables for the inference approach detailed in the previous section, the prediction accuracy of the MLP accordingly becomes comparable to the CNN in this phase.

In conclusion, we find that the CNN characterises the PM phase by additionally measuring kinetic contributions in the described manner, rather than only expectation values of the condensate like in the broken phases. Still,  $M$  and  $M_s$  are being utilised as well, judging from the comparably large relevances of the FM filters in this region. Due to the opacity of the fully-connected layers following the convolution, some ambiguity remains regarding the precise decision rules that the network implements based on these quantities. This residual lack of clarity can likely be resolved by manually enforcing locality in the internal operations, e.g. by introducing artificial bottlenecks into the network. Of course, the form

of  $\mathcal{O}_{\text{PM}}$  is also not exactly equivalent to the operations of the CNN, even though they share many important features. In particular, there is a mismatch between the averaging procedure and the MaxPool layer. Effects associated with the choice of different activation functions and pooling layers, which may be tailored more specifically towards certain types of observables, should be investigated in the future. However, our analysis shows that the overlap with the learned internal representation is significant.

## V. CONCLUSIONS AND OUTLOOK

We have investigated the application of interpretability methods to deep neural network classifiers as a general-purpose framework for the identification of physical features from lattice data. The approach facilitates an interpretation of a network's predictions, permitting a quantitative understanding of the internal representations that the network learns in order to solve a pretext task—in this case, inference of action parameters. This culminates in the extraction of relevant observables from the data, leading to insights about the phase structure.

First, both types of magnetisations and the time-sliced, connected two-point correlator were used as training data for a MLP (see Figure 4). Inference of the hopping parameter was shown to work in each of the two broken phases, respectively. However, in the symmetric phase, the network was observed to suffer from bad accuracy. This indicates that the amount of relevant information present in the dataset is insufficient for the network to fully capture the dynamics of the theory. Using layer-wise relevance propagation, we determined a  $\kappa$ -dependent importance hierarchy of the observables. Using this approach we were able to confirm our physics expectations about order parameters being relevant within their associated phases. Moreover, while the two-point correlation function is sensitive to the PM phase, this signal is insufficient for attaining high accuracy for the MLP. Our numerical results and interpretation thereof were further verified by a random forest regression benchmark performed on the same dataset, which demonstrated qualitatively comparable accuracy (see Figure 5).

Next, we trained a CNN directly on the field configurations. In contrast to aforementioned results, the CNN was shown to yield superior accuracy for the same inference task (see Figure 6). Therefore, the set of observables chosen previously must have neglected important information, which the network managed to distill from the raw data. Employing LRP, a cumulative relevance was assigned to the individual convolutional filters, revealing a distinctive pattern that explains the decision process. In particular, we observed that the network specifically assigned filters to the each of the phases of the theory, with small to vanishing relevances in the remaining phases. This also indicates where phase transitions are located. We confirmed that the learned filters corre-



spond to representations of the known order parameters by examining the weight maps (see Figures 7 and 8), essentially reproducing previous results.

Guided by the filter analysis, we constructed an observable that characterises the symmetric phase. In a heuristic attempt to find the exact form of this quantity, we defined  $\mathcal{O}_{\text{PM}}$  in Equation (9) and showed that it exhibits several interesting properties (see Figure 9). We interpreted this quantity as a particular measure of local fluctuations that is also sensitive to nearest-neighbour sign flips. This further validates our physical intuition, since in the PM phase, we expect that relevant information for its characterisation is encoded by kinetic contributions. As discussed in detail below Equation (9), the naive continuum limit of  $\mathcal{O}_{\text{PM}}$  is simply the volume integral of  $|\nabla\phi|$ . Hence, it has lost the information about nearest-neighbour sign flips, while the continuum limit of its expectation value,  $\langle\mathcal{O}_{\text{PM}}\rangle$ , keeps its sensitivity towards this property. Accordingly, the construction of this observable guided by the filter analysis is non-trivial evidence for the potential power of the present approach: the results demonstrate that we can identify relevant structures which may otherwise stay hidden. At this point, LRP has indeed facilitated a deeper understanding of the CNN, by explaining the origin of its comparably high accuracy w.r.t. the MLP. With these results, we have conclusively established the value of interpretability methods in deep learning analyses of lattice data.

In the present work, the emphasis was put on the methodological aspects of the analysis in order to form a comprehensive basis for future efforts. Many interesting aspects, such as an investigation of the fermionic sector, were barely discussed. Instead, we have focused on the inference of the hopping parameter. Including other action parameters into the labels, such as the Yukawa coupling or chemical potential, is a promising endeavour for the future, as it will likely lead to an improvement in comparison to the current results. This is necessary in order to pave the way towards an application to more interesting scenarios, such as QCD at finite density or competing order regimes in the Hubbard model. Moreover, the introduced ML pipeline has the potential to provide insight also in various other areas of computational physics.

### Acknowledgements

We thank M. Scherzer, I.-O. Stamatescu, S. J. Wetzel and F. P.G. Ziegler for discussions. This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster) and under the Collaborative Research Centre SFB 1225 (ISOQUANT), EMMI, the BMBF grants 05P18VHFCA, 01IS14013A (Berlin Big Data Center), and 01IS18037I (Berlin Center for Machine Learning).

## Appendix A: Theory and Simulation Details

### 1. Dimensionless Form of the Klein-Gordon Action

The lattice action for real, scalar  $\phi^4$ -theory in  $d$  dimensions is defined as

$$S_{\text{KG}}[\phi_0] = \sum_{n \in \Lambda} a^d \left[ \frac{1}{2} \sum_{\mu=1}^d \frac{(\phi_0(n + a\hat{\mu}) - \phi_0(n))^2}{a^2} + \frac{m_0^2}{2} \phi_0^2 + \frac{g_0}{4!} \phi_0^4 \right], \quad (\text{A1})$$

where  $a$  is the lattice spacing,  $\phi_0, m_0, g_0$  correspond to the bare field, mass and coupling constant, and  $\hat{\mu}$  is the unit vector in  $\mu$ -direction. The action can be cast into a dimensionless form through the following transformation:

$$\begin{aligned} a^{\frac{d-2}{2}} \phi_0 &= (2\kappa)^{1/2} \phi \\ (am_0)^2 &= \frac{1-2\lambda}{\kappa} - 2d \\ a^{-d+4} \lambda_0 &= \frac{6\lambda}{\kappa^2}. \end{aligned} \quad (\text{A2})$$

Here,  $\kappa$  is commonly called the hopping parameter and  $\lambda$  now takes the role of the coupling constant. Applying this transformation results in

$$S_{\text{KG}}[\phi] = \sum_{n \in \Lambda} \left[ -2\kappa \sum_{\mu=1}^d \phi(n) \phi(n + \hat{\mu}) + (1-2\lambda) \phi(n)^2 + \lambda \phi(n)^4 \right]. \quad (\text{A3})$$

### 2. Simulating Fermions

Calculating the determinant of the discretised Dirac operator (Equation (2)) exactly and repeatedly, which is in principle necessary for importance sampling, is computationally intractable even for moderate lattice sizes. The usual approach is to approximate its value stochastically, e.g. by introducing auxiliary bosonic field variables (commonly called pseudo-fermions), which guarantees an asymptotically exact distribution. Simulations based on the numerical solution of differential equations, such as the Hybrid Monte Carlo (HMC) algorithm or Langevin dynamics, can exploit the comparably low cost of computing only the matrix inverse with the conjugate gradient method. In this work, we exclusively employ the HMC algorithm to generate data.

## Appendix B: Lattice Datasets

$N$	$\lambda$	$M$	$g$	$\Delta\kappa$	#samples per $\kappa$
16	1.1	20	0.25	0.005	200 - Training set 100 - Test set

TABLE I: Action/simulation parameters used for training and test dataset.

All field configurations composing the datasets used in this work are generated with the parameters listed in Table I. A single, labeled sample is given by the mapping

$$(\phi, \kappa) : \quad \{\phi_n\} = \{\phi_n \mid n \in \Lambda\} \longrightarrow \kappa. \quad (\text{B1})$$

In order to explicitly enforce  $\mathbb{Z}_2$  symmetry onto the neural networks, we use the same configurations twice in the dataset, just with a globally flipped sign. This raw data is directly used to train the CNN. For the MLP, the samples are preprocessed by computing the chosen set of observables for each configuration,

$$(\mathcal{O}, \kappa) : \quad \{|M|, |M_s|, G_c(t)\} \longrightarrow \kappa. \quad (\text{B2})$$

In this case, we can simply take the modulus of the magnetisations without losing information, since only two branches with exactly opposite signs are present in the phase diagram. Due to the finite expectation value of the staggered magnetisation, the AFM phase contains unphysical negative correlations. In order to remove these lattice artifacts, we adapt the usual time-sliced two-point correlator to

$$G_c(t) = \left| \langle \phi(t)\phi(0) \rangle - M^2 - (-1)^t M_s^2 \right|. \quad (\text{B3})$$

Generally, LRP is designed for classification problems. Therefore, we discretise  $\kappa$  to facilitate the formulation of the inference objective as a classification task. All values of  $\kappa$  are transformed into individual bins and the networks are tasked to predict the correct bin. In order to retain a notion of locality, the true bins are additionally smeared out with a Gaussian distribution, resulting in the target labels

$$\kappa \longrightarrow y_b = e^{-\frac{(\kappa_b - \kappa_{\text{True}})^2}{2\sigma^2}}. \quad (\text{B4})$$

Here,  $b$  denotes the bin number, and the variance was set to  $\sigma = 3\Delta\kappa$ . In combination with a MSE loss, we obtain qualitatively similar prediction results compared to a standard regression approach.

## Appendix C: Propagation Rules

This section contains a summary of the mathematical background of LRP, in particular regarding the propagation rules. Generally, the relevance  $R_j$  depends on the activation of the previous layer  $x_i$ . Given some input to the network, its predicted class  $f$  is identified by the output neuron with the largest response. This neuron's activation  $R_f^{\text{out}}$ , along with  $R_i^{\text{out}} = 0$  for all other classes  $i \neq f$ , defines the relevance vector. This output layer relevance can then be backpropagated through the whole network, which results in the aforementioned heatmap on the input. Importantly, the propagation rules are designed such that the total relevance is conserved,

$$\sum_i R_i^n = \sum_i R_i^{\text{out}} \equiv R_f^{\text{out}}, \quad (\text{C1})$$

where the index  $n$  can indicate any layer. This conservation law ensures that explanations from all layers are closely related and prohibits additional sources of relevance during the backpropagation. A Taylor expansion of this conservation law yields

$$\sum_j R_j(x_i) = \underbrace{\sum_j R_j(\tilde{x}_i)}_{=0} + \underbrace{\sum_i \sum_j \frac{\partial R_j}{\partial x_i} \Big|_{\{\tilde{x}_i\}}}_{R_i} (x_i - \tilde{x}_i). \quad (\text{C2})$$

Here, we choose  $\tilde{x}_i$  to be a so-called root point, which corresponds to an activation with vanishing consecutive layer relevance  $R_j(\tilde{x}_i) = 0$ . By definition, it is localised on the layer's decision boundary, which constitutes a hypersurface in the activation space. Hence, the root point is not uniquely defined and we need to impose an additional criterion. However, given such a point, we can identify the first order term as the relevance propagation rule  $R_j \mapsto R_i$ . The remaining root point dependence gives rise to a variety of possible propagation rules. For instance, the  $w^2$  rule minimises the Euclidean distance between neuron activation  $x_i$  and the decision boundary in order to single out a root point. Visualisations of root points, as well as essential derivations and analytical expressions for propagation rules, can be found in [41].

## Appendix D: Random Forest Details

Random forests [47] denote a predictive ML approach based on ensembles of decision trees. They utilise the majority vote of multiple randomised trees in order to arrive at a prediction. This greatly improves the generalisation performance compared to using a single tree. The elementary building block is a node performing binary decisions based on a single feature criterion. New nodes are

connected sequentially with so-called branches. A single decision tree is grown iteratively from a root node to multiple leaf nodes. A concrete prediction corresponds to a unique path from the root to a single leaf. Each node on the path is associated with a specific feature. Hence, we can sum up the contributions to the decision separately for each feature by moving along the path,

$$prediction = bias + \sum_i (feature\_contribution)_i. \quad (D1)$$

Here, the bias corresponds to the average prediction at the root node.

We employ the SCIKIT-LEARN implementation [48] in combination with a TREEINTERPRETER extension [49]. The latter reference also provides an excellent introduction to the concept of feature contributions. The random forest is initialised with 10 trees and a maximum tree depth of 10. This parameter is essential for regularisation, since an unconstrained depth causes overfitting and thus results in poor generalisation performance.

### Appendix E: Network Architectures and Implementation Details

We use the PYTORCH framework [50]. The machinery of LRP is included by defining a custom `torch.nn.Module` and equipping all layers with a relevance propagation rule. Furthermore, all biases are restricted to negative values in order to ensure the existence of a root point. For training, we employ the Adam optimiser [51] with default hyperparameters and an initial learning rate of 0.001, using a batch size of 16.

For both networks, the first layer undergoes least absolute shrinkage and selection operator (LASSO) regularisation during training, which encourages sparsity and thereby enhances interpretability. This corresponds to simply adding the  $L^1$  norm of the respective weights  $w_{ij}$  to the MSE loss, which accordingly takes the form

$$L = \frac{1}{d} \sum_{f=1}^d (y_f - \hat{y}_f)^2 + \lambda_{\text{Lasso}} \sum_{ij} |w_{ij}|. \quad (E1)$$

Here,  $y_f, \hat{y}_f$  denote the prediction and ground truth labels, and  $i, j$  the input and output nodes of the first layer. The quantity  $\lambda_{\text{Lasso}}$  parametrises the strength of the regularisation.

The network architectures used in this work are given in the following tables.

Layer	Specification	Propagation rule
<i>Linear</i>	in=18, out=256	$w^2$ -rule
<i>ReLU</i>		$R_i = R_j$
<i>Linear</i>	in=256, out=128	$z^+$ -rule
<i>ReLU</i>		$R_i = R_j$
<i>Linear</i>	in=128, out=140	$z^+$ -rule
<i>LeakyReLU</i>	negative slope=0.01	$R_i = R_j$

TABLE II: Network architecture of the MLP. The first layer undergoes  $L^1$  regularisation with  $\lambda_{\text{Lasso}} = 5$ .

Layer	Specification	Propagation rule
<i>Conv3d</i>	#filter = 5 kernel=B, strides=A	$w^2$ -rule
<i>ReLU</i>		$R_i = R_j$
<i>MaxPool3d</i>	kernel=B, strides=B	$z^+$ -rule
<i>Linear</i>	in=1715, out=256	$z^+$ -rule
<i>ReLU</i>		$R_i = R_j$
<i>Linear</i>	in=256, out=140	$z^+$ -rule
<i>ReLU</i>		$R_i = R_j$

TABLE III: Network architecture of the CNN, with  $A = (1 \times 1 \times 1)$ ,  $B = (2 \times 2 \times 2)$ . The first layer undergoes  $L^1$  regularisation with  $\lambda_{\text{Lasso}} = 10$ .

- 
- |  |   |
|--|---|
| <p>[1] Y. LeCun, Y. Bengio, and G. Hinton <i>Nature</i> <b>521</b> no. 7553, (May, 2015) 436–444.</p> <p>[2] A. Graves, A.-r. Mohamed, and G. Hinton in <i>2013 IEEE international conference on acoustics, speech and signal processing</i>, pp. 6645–6649, IEEE. 2013. <a href="#">arXiv:1303.5778 [cs.NE]</a>.</p> <p>[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton in <i>Advances in Neural Information Processing Systems 25</i>, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., pp. 1097–1105. Curran Associates, Inc., 2012.</p> | <p>[4] S. Ren, K. He, R. Girshick, and J. Sun in <i>Advances in Neural Information Processing Systems 28</i>, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds., pp. 91–99. Curran Associates, Inc., 2015. <a href="#">arXiv:1506.01497 [cs.CV]</a>.</p> <p>[5] K. He, X. Zhang, S. Ren, and J. Sun in <i>Proceedings of the IEEE conference on computer vision and pattern recognition</i>, pp. 770–778. 2016.</p> <p>[6] L. Wang <i>Physical Review B</i> <b>94</b> no. 19, (Nov, 2016) , <a href="#">arXiv:1606.00318 [cond-mat.stat-mech]</a>.</p> <p>[7] P. Broecker, J. Carrasquilla, R. Melko, and S. Trebst</p> |
|--|---|

- Scientific Reports* **7** (08, 2016) , [arXiv:1608.07848](#) [cond-mat.str-el].
- [8] S. J. Wetzel and M. Scherzer *Phys. Rev.* **B96** no. 18, (2017) 184410, [arXiv:1705.05582](#) [cond-mat.stat-mech].
- [9] S. J. Wetzel *Physical Review E* **96** no. 2, (Aug, 2017) , [arXiv:1703.02435](#) [cond-mat.stat-mech].
- [10] M. Cristoforetti, G. Jurman, A. I. Nardelli, and C. Furlanello [arXiv:1705.09524](#) [hep-lat].
- [11] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami *Phys. Rev. X* **7** (Aug, 2017) 031038, [arXiv:1609.02552](#) [cond-mat.str-el].
- [12] W. Hu, R. R. P. Singh, and R. T. Scalettar *Physical Review E* **95** no. 6, (Jun, 2017) , [arXiv:1704.00080](#) [cond-mat.stat-mech].
- [13] Z. Liu, S. P. Rodrigues, and W. Cai [arXiv:1710.04987](#) [cond-mat.dis-nn].
- [14] G. Carleo and M. Troyer *Science* **355** no. 6325, (Feb, 2017) 602–606.
- [15] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber *Nature Physics* **13** no. 5, (Feb, 2017) 435–439, [arXiv:1610.02048](#) [cond-mat.dis-nn].
- [16] P. Ponte and R. Melko *Physical Review B* **96** (04, 2017) , [arXiv:1704.05848](#) [cond-mat.stat-mech].
- [17] J. Carrasquilla and R. G. Melko *Nature Physics* **13** no. 5, (Feb, 2017) 431–434, [arXiv:1605.01735](#) [cond-mat.str-el].
- [18] A. Morningstar and R. G. Melko *J. Mach. Learn. Res.* **18** no. 1, (Jan., 2017) 5975–5991, [arXiv:1708.04622](#) [cond-mat.dis-nn].
- [19] A. Tanaka and A. Tomiya [arXiv:1712.03893](#) [hep-lat].
- [20] L. Huang and L. Wang *Phys. Rev. B* **95** (Jan, 2017) 035105, [arXiv:1610.02746](#) [physics.comp-ph].
- [21] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu *Phys. Rev. B* **95** (Jan, 2017) 041101, [arXiv:1610.03137](#) [cond-mat.str-el].
- [22] D. Wu, L. Wang, and P. Zhang *Physical Review Letters* **122** no. 8, (Feb, 2019) , [arXiv:1809.10606](#) [cond-mat.stat-mech].
- [23] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker *Physical Review D* **100** no. 1, (Jul, 2019) , [arXiv:1810.12879](#) [hep-lat].
- [24] P. E. Shanahan, D. Trewartha, and W. Detmold *Phys. Rev.* **D97** no. 9, (2018) 094506, [arXiv:1801.05784](#) [hep-lat].
- [25] P. Suchsland and S. Wessel *Physical Review B* **97** no. 17, (May, 2018) , [arXiv:1802.09876](#) [cond-mat.stat-mech].
- [26] J. M. Urban and J. M. Pawłowski [arXiv:1811.03533](#) [hep-lat].
- [27] F. Noé, S. Olsson, J. Köhler, and H. Wu *Science* **365** no. 6457, (Sept., 2019) eaaw1147, [arXiv:1812.01729](#) [stat.ML].
- [28] K. Nicoli, P. Kessel, N. Strodthoff, W. Samek, K.-R. Müller, and S. Nakajima [arXiv:1903.11048](#) [cond-mat.stat-mech].
- [29] M. S. Albergo, G. Kanwar, and P. E. Shanahan *Phys. Rev.* **D100** no. 3, (2019) 034515, [arXiv:1904.12072](#) [hep-lat].
- [30] L. Kades, J. M. Pawłowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. Ziegler [arXiv:1905.04305](#) [physics.comp-ph].
- [31] K. Kashiwa, Y. Kikuchi, and A. Tomiya *PTEP* **2019** no. 8, (2019) 083A04, [arXiv:1812.01522](#) [cond-mat.dis-nn].
- [32] K. Liu, J. Greitemann, and L. Pollet *Physical Review B* **99** no. 10, (Mar, 2019) , [arXiv:1810.05538](#) [cond-mat.stat-mech].
- [33] C. Casert, T. Vieijra, J. Nys, and J. Ryckebusch *Physical Review E* **99** no. 2, (Feb, 2019) , [arXiv:1807.02468](#) [cond-mat.stat-mech].
- [34] W. Rządowski, N. Defenu, S. Chiacchiera, A. Trombettoni, and G. Bighin [arXiv:1907.05417](#) [cond-mat.dis-nn].
- [35] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel *Phys. Rev. E* **101** (Feb, 2020) 023304, [arXiv:1910.13496](#) [cond-mat.stat-mech].
- [36] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab *Physics reports* **810** (2019) 1–124, [arXiv:1803.08823](#) [physics.comp-ph].
- [37] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová *Reviews of Modern Physics* **91** no. 4, (Dec., 2019) , [arXiv:1903.10563](#) [physics.comp-ph].
- [38] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek *PLOS ONE* **10** no. 7, (2015) e0130140.
- [39] I. Montvay and G. Münster, *Quantum Fields on a Lattice*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1994.
- [40] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross in *NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning - Now What?* ETH Zurich, 2017. [arXiv:1711.06104](#). <http://hdl.handle.net/20.500.11850/237705>.
- [41] G. Montavon, W. Samek, and K.-R. Müller *Digital Signal Processing* **73** (2018) 1–15. <https://doi.org/10.1016/j.dsp.2017.10.011>.
- [42] K. Simonyan, A. Vedaldi, and A. Zisserman in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, eds. 2014. [arXiv:1312.6034](#) [cs.CV].
- [43] M. Sundararajan, A. Taly, and Q. Yan in *ICML*. 2017. [arXiv:1703.01365](#) [cs.LG].
- [44] A. Shrikumar, P. Greenside, and A. Kundaje in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, eds., vol. 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, International Convention Centre, Sydney, Australia, 06–11 aug, 2017. [arXiv:1704.02685](#) [cs.CV].
- [45] M. D. Zeiler and R. Fergus in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds., vol. 8689 of *Lecture Notes in Computer Science*, pp. 818–833. Springer, 2014. [arXiv:1311.2901](#) [cs.CV].
- [46] T. B. Fraunhofer HHI and S. Singapore. <http://heatmapping.org/>. accessed: 2020-01-07.
- [47] L. Breiman *Mach. Learn.* **45** no. 1, (Oct., 2001) 5–32.
- [48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

- E. Duchesnay *Journal of Machine Learning Research* **12** (2011) 2825–2830.
- [49] A. Saabas, <https://github.com/andos/treeinterpreter>, 2015.
- [50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimsheine, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, eds., pp. 8024–8035. Curran Associates, Inc., 2019. [arXiv:1912.01703](https://arxiv.org/abs/1912.01703) [cs.LG].
- [51] D. P. Kingma and J. Ba [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG].