

Towards novel insights in lattice field theory with explainable machine learning

Stefan Blücher,¹ Lukas Kades¹,¹ Jan M. Pawłowski,^{1,2} Nils Strodthoff,³ and Julian M. Urban¹

¹*Institut für Theoretische Physik, Universität Heidelberg,
Philosophenweg 16, 69120 Heidelberg, Germany*

²*ExtreMe Matter Institute EMMI, GSI, Planckstraße 1, D-64291 Darmstadt, Germany*

³*Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany*



(Received 11 March 2020; accepted 7 May 2020; published 20 May 2020)

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 characterizes the symmetric phase.

DOI: [10.1103/PhysRevD.101.094507](https://doi.org/10.1103/PhysRevD.101.094507)

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 characterize the dynamics.

In this quest toward 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 [6–39]. We recommend [40] as an introduction to ML for physicists and [41] 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 recognize patterns which can be leveraged to construct observables from low-level features that characterize 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” techniques, which have recently attracted considerable interest in the ML community and beyond. In this work, we focus on layer-wise relevance propagation (LRP) [42]. 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 toward/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

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

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 organized as follows. In Sec. 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 Sec. IV A. In Sec. 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 Sec. 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 Λ denotes the set of all lattice sites. Here, κ is called the hopping parameter and λ 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 discretized 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 γ -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 $\eta_i(n) = (-1)^{n_1} \cdots (-1)^{n_{i-1}}$. 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 ϕ^4 -theory include the magnetization,

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

as well as the staggered magnetization

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

which is relevant for negative κ . The scalar part of Eq. (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 magnetizations. The fermionic part explicitly breaks this symmetry.

A slice of the phase diagram at fixed Yukawa coupling is shown in Fig. 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 nonzero expectation values. They are

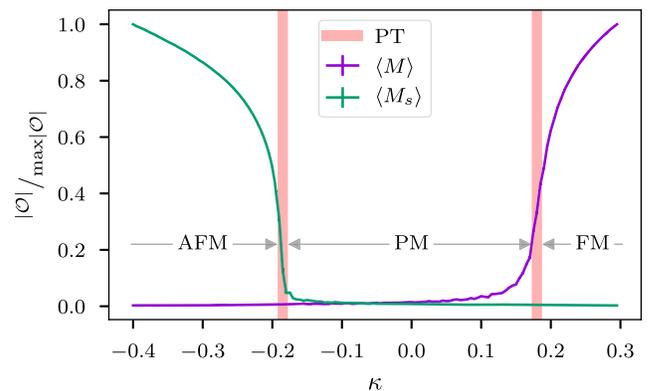


FIG. 1. Slice of the phase diagram for fixed Yukawa coupling $g = 0.25$ using normalized values of $\langle M \rangle$ and $\langle M_s \rangle$. Phase transitions are highlighted by the shaded bars. We distinguish an antiferromagnetic (AFM), a paramagnetic (PM) and a ferromagnetic (FM) phase.

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 magnetization 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 the Appendix A. For a comprehensive treatment of Yukawa theory on the lattice, we recommend [43].

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 [6,9,13]. However, its linear structure prohibits the identification of complex nonlinear features, e.g., Wilson loops in gauge theories. Hence, we require tools capable of modeling nonlinearities, such as deep neural networks [18]. They allow for a more comprehensive treatment of complex systems, which has been demonstrated e.g., for fermionic theories in [7,12]. The approach also enables novel procedures, such as learning by confusion and similar techniques, to locate phase transitions in a semisupervised manner [16,36]. For lattice QCD, action parameters can be extracted from field configurations [26]. 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 [10,27,33], utilizing (kernel) support vector machines [17,34], and saliency maps [35]. Interpretability is also investigated for other applications in theoretical physics, e.g., by employing twin neural networks [44].

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 which features the classifier was particularly sensitive to, or influenced the algorithm toward/against an individual classification decision. In the domain of image recognition, such attribution maps are typically visualized 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 [45,46]. Very broadly, the most important types of such local interpretability methods can be categorized as: 1. Gradient-based, such as saliency maps [47] obtained by computing the derivative of a particular class score with respect to the input features or integrated gradients [48]. 2. Decomposition-based, such as layer-wise relevance propagation (LRP) [42] or DeepLift [49]. 3. Perturbation-based, as in [50], investigating the change in class scores when occluding parts of the input features.

In this work, we focus on LRP, a particular variant of decomposition-based attribution methods, which has been successfully applied to other problems in physics and chemistry, e.g., in the context of atomistic systems [51]. Nevertheless, we stress that qualitative findings are expected to agree for all decomposition- and gradient-based methods [52]. 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 using certain propagation rules, see the sketch in Fig. 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 toward (against) a particular classification decision.

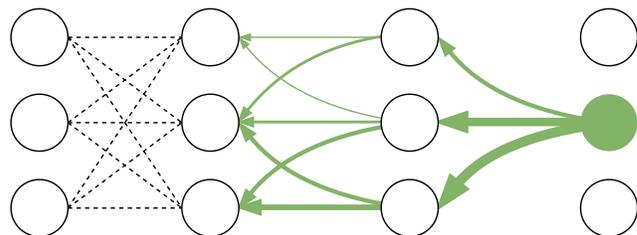


FIG. 2. Sketch of LRP through the last two layers of a classification network that predicts one-hot vectors. Relevance is indicated by arrow width. The conservation law requires the sum of widths to remain constant during back propagation. Diagram adapted from [53].

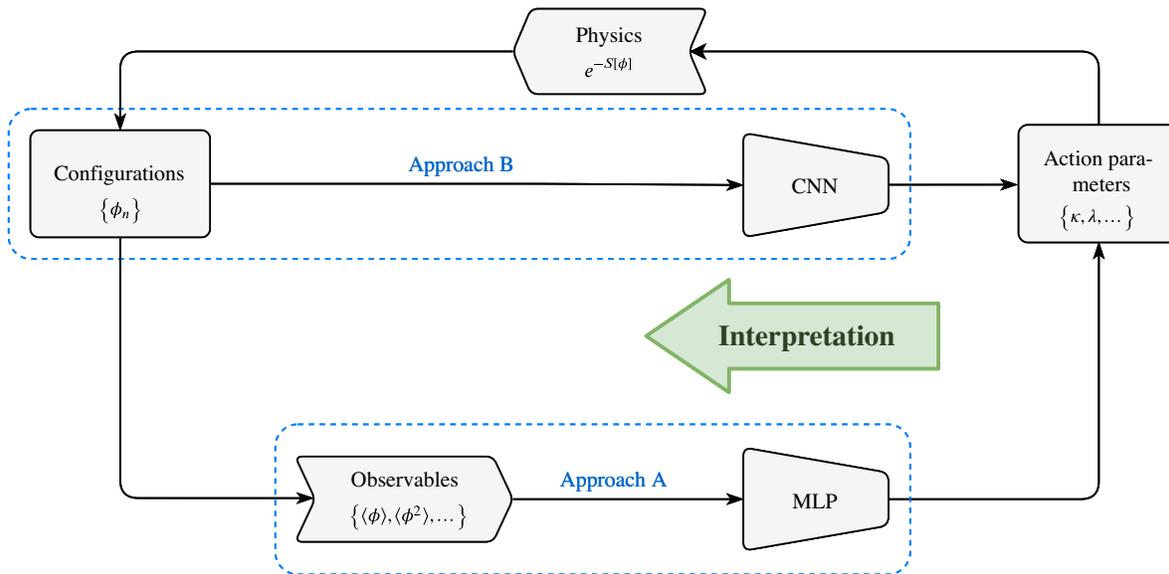


FIG. 3. Sketch of our strategy to learn meaningful structures from the simulation data by analysing the networks trained for action parameter inference. Field configurations used for training are either preprocessed into observables for the MLP (Approach A) or directly operated upon with a CNN (Approach 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.

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 κ from a set of known observables (Approach A), as well as solely from the raw field configurations (Approach B), akin to [26]. 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 in accordance with physical expert knowledge. In the second case, by calculating the relevances of the learnt 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 characterizes the symmetric phase. Both variants of our strategy are sketched in Fig. 3. Due to the ill-conditioned nature of the action parameter prediction problem, the optimization 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 regularization, see Appendix E for details.

A. Importance hierarchies of known observables

In Sec. II we introduced a set of standard observables, consisting of the normal and staggered magnetization as

well as the time-sliced two-point correlation function.¹ 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 κ (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 optimization of decision trees [54] (see Appendix D for details). The results for both approaches, shown in Figs. 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 Fig. 1 that always one of the two types of magnetizations is strictly monotonic in the respective phase and can therefore determine κ uniquely. However, both approaches yield at best mediocre performance in the symmetric PM phase. Here, both magnetizations 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.

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

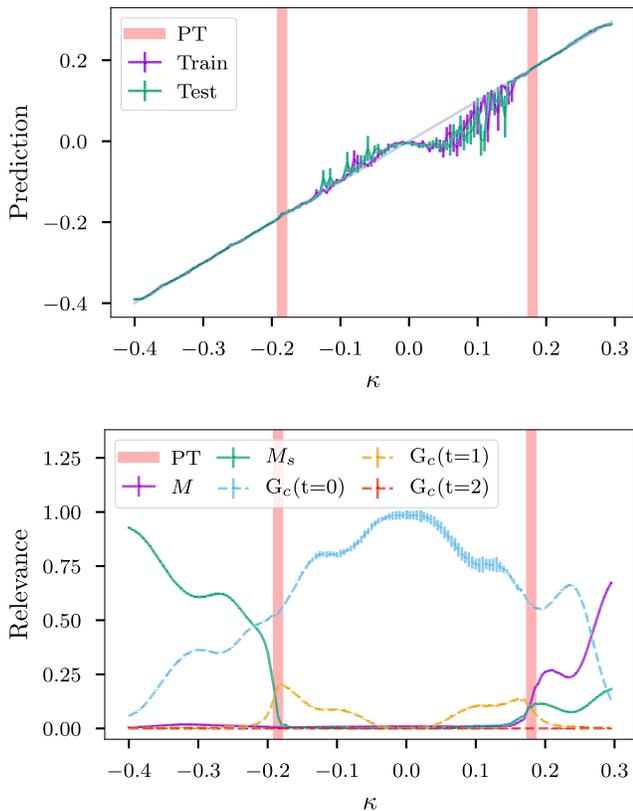


FIG. 4. Results for the MLP. Top: predictions, bottom: normalized LRP relevances of individual features. Error bars here and throughout this work are obtained with the statistical jack-knife method.

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 characterize 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 learnt decision rules conform to the physical interpretation given above. We begin with the relevances as determined by LRP, shown in Fig. 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 described 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.

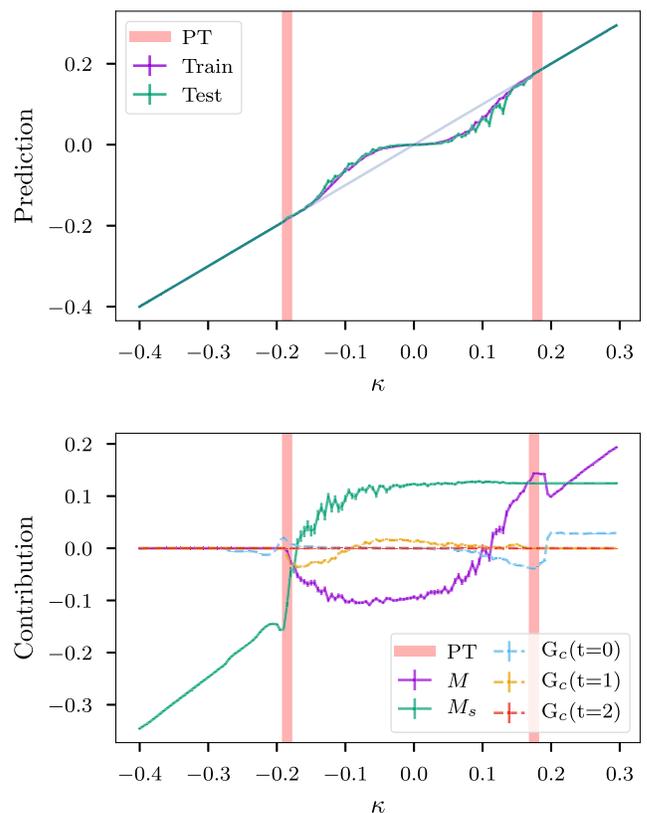


FIG. 5. Benchmark results for the random forest. Top: predictions, bottom: nominal contributions of individual features.

The interpretation sketched above is further supported by the results obtained through random forest regression [54]. 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 Fig. 5 (bottom). In the broken FM and AFM phases, the respective contributions of M and M_s demonstrate a linear dependence on κ . Again, this clearly indicates that these quantities characterize 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 nonzero 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 characterize 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 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 κ . 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 toward 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 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 [6,9,13], albeit not in exactly the same physical setting. PCA immediately identifies the normal and staggered magnetizations 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 parametrizes 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 adding the L^1 norm to the loss—also known as LASSO regularization—as suggested in [35] (see Appendix E for details). Due to the nature of the convolution operation, learnt 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, nonlinear 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 Fig. 6 (top). We can immediately observe a superior performance in the PM

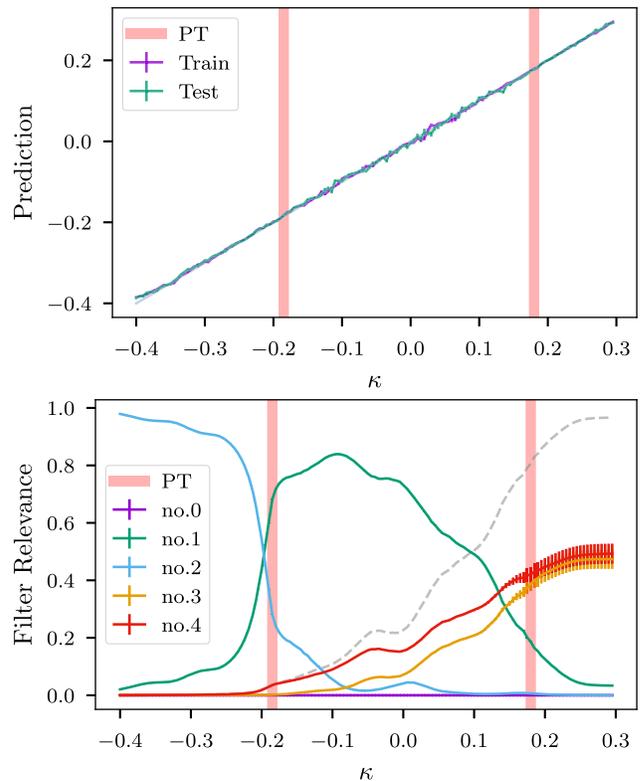


FIG. 6. Results for the CNN. Top: prediction, bottom: normalized relevances of individual filters. The dashed curve corresponds to the cumulative relevance of filter 3 and 4.

phase compared to our previous results. The CNN succeeds to consistently infer κ 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 learnt representations, we have to customize 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 localized. For field configurations on the lattice, due to the translational symmetry of the action and the resulting homogeneity, no particularly distinguished, localized 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.

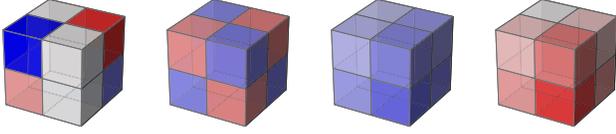


FIG. 7. learnt weights of convolutional filters. Left to right: (no.1, PM); (no.2, AFM); (no.3, FM); (no.4, FM). The colour map is symmetric around zero. Red (blue) corresponds to positive (negative) weights.

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 κ . We can recognize some similarities to the relevances in Fig. 4, highlighting the underlying phase structure of the Yukawa theory. It appears that the model can parametrize each phase individually using one or a small subset of filters, while the others show small or insignificant relevances in the respective region. The learnt weight maps are shown in Fig. 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 characterize all phases seen in the data. This reduction is an effect of the regularization, and constitutes a recurring pattern also when more filters are initially used, providing a first hint toward the number of independent quantities utilized 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 magnetization 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 magnetization 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 magnetization 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 magnetized 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 magnetized 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 characterization 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,

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

As with M and M_s , we obtain the corresponding staggered form $\mathcal{O}_{\text{PM}}^s$ by applying the transformation given in Eq. (6). The resulting pair of quantities is visualised by the idealised filters in Fig. 10.

The observable \mathcal{O}_{PM} defined in Eq. (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}_{PM} would be the sum over all sites of a total derivative, which vanishes identically. We also remark that \mathcal{O}_{PM} can be made isotropic by summing over all directions.

We now discuss the properties of the theory that are measured by \mathcal{O}_{PM} : In the continuum limit, \mathcal{O}_{PM} naively tends toward the volume integral over $|\nabla\phi|$. Due to 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}_{PM} to sign flips of nearest-neighbors. 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 localized, (anti-)magnetized regions, even if the expectation values vanish globally. Possible local field

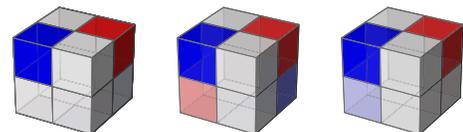


FIG. 8. Comparison of PM filters for three independent training runs of the CNN.

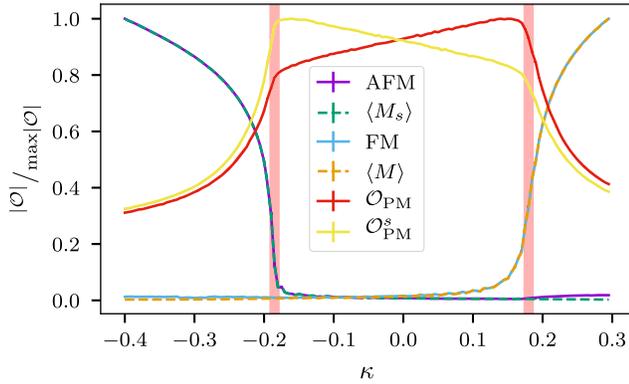


FIG. 9. Normalized observables reconstructed from the learnt filters. The quantities associated with the FM and AFM phases are compared to M and M_s . \mathcal{O}_{PM} and $\mathcal{O}_{\text{PM}}^s$ are related by Eq. (6) and exhibit an approximate mirror symmetry around $\kappa = 0$.

alignments resulting in different values of \mathcal{O}_{PM} , but not of the standard derivative, are visualized in Fig. 11.

The construction and discussed sensitivities of $\langle \mathcal{O}_{\text{PM}} \rangle$ demonstrates again the usefulness of LRP: we can identify the learnt representation as a feature of the dataset arising from the lattice discretization. $\langle \mathcal{O}_{\text{PM}} \rangle$ and $\langle \mathcal{O}_{\text{PM}}^s \rangle$ as functions of κ are shown in Fig. 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 κ inference. In fact, if \mathcal{O}_{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 characterizes 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 utilized 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}_{PM} is also not exactly

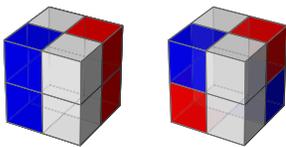


FIG. 10. Convolutional filters corresponding to the observable \mathcal{O}_{PM} defined in Eq. (9) (left) and its corresponding staggered counterpart $\mathcal{O}_{\text{PM}}^s$ (right).

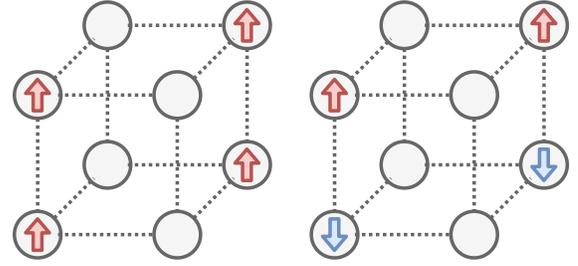


FIG. 11. Qualitative visualization of local structures in field configurations operated upon by the PM filter. Sign is encoded by arrow orientation/colour. Diagonal neighbors tend to share the same sign everywhere in the phase diagram. On the contrary, nearest neighbors show a preference toward either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

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 toward 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 magnetizations and the time-sliced, connected two-point correlator were used as training data for a MLP (see Fig. 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 κ -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 Fig. 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 Fig. 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 correspond to representations of the known order parameters by examining the weight maps (see Figs. 7 and 8), essentially reproducing previous results.

Guided by the filter analysis, we constructed an observable that characterizes the symmetric phase. In a heuristic attempt to find the exact form of this quantity, we defined \mathcal{O}_{PM} in Eq. (9) and showed that it exhibits several interesting properties (see Fig. 9). We interpreted this quantity as a particular measure of local fluctuations that is also sensitive to nearest-neighbor sign flips. This further validates our physical intuition, since in the PM phase, we expect that relevant information for its characterization is encoded by kinetic contributions. As discussed in detail below Eq. (9), the naive continuum limit of \mathcal{O}_{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 toward this property. Accordingly, the construction of this observable guided by the filter analysis is nontrivial 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 endeavor 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 toward 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.

ACKNOWLEDGMENTS

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 No. 05P18VHFCA, No. 01IS14013A (Berlin Big Data Center), and No. 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 ϕ^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, ϕ_0, m_0, g_0 correspond to the bare field, mass and coupling constant, and $\hat{\mu}$ is the unit vector in μ -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, κ is commonly called the hopping parameter and λ 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 discretized Dirac operator [Eq. (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

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

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

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,

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

In this case, we can simply take the modulus of the magnetizations 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 magnetization, the AFM phase contains unphysical negative correlations. In order to remove these lattice artifacts, we adapt the usual time-sliced two-point correlator to

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

Generally, LRP is designed for classification problems. Therefore, we discretize κ to facilitate the formulation of the inference objective as a classification task. All values of κ 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

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

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.

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

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

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^{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

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

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 localized 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 minimizes the Euclidean distance between neuron activation x_i and the decision boundary in order to single out a root point. Visualizations of root points, as well as essential derivations and analytical expressions for propagation rules, can be found in [46].

APPENDIX D: RANDOM FOREST DETAILS

Random forests [54] denote a predictive ML approach based on ensembles of decision trees. They utilize the

majority vote of multiple randomized trees in order to arrive at a prediction. This greatly improves the generalization 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,

$$\text{prediction} = \text{bias} + \sum_i (\text{feature contribution})_i. \quad (\text{D1})$$

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

We employ the SCIKIT-LEARN implementation [55] in combination with a TREEINTERPRETER extension [56]. The latter reference also provides an excellent introduction to the concept of feature contributions.

The random forest is initialized with 10 trees and a maximum tree depth of 10. This parameter is essential for regularization, since an unconstrained depth causes overfitting and thus results in poor generalization performance. In order to fix this parameter, we start at a large value and successively reduce it until the training and test accuracy reach a similar level. This way we can retain as much expressive power as possible in the random forest while simultaneously eliminating systematic errors resulting from overfitting. However, we emphasize that the specific choice of this parameter not relevant to our argument.

APPENDIX E: NETWORK ARCHITECTURES AND IMPLEMENTATION DETAILS

We use the PYTORCH framework [57]. 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

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

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	$R_i = R_j$

optimizer [58] 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) regularization 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

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

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 λ_{Lasso} parametrizes the strength of the regularization.

The network architectures used in this work are given in Tables II and III.

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 regularization with $\lambda_{\text{Lasso}} = 10$.

Layer	Specification	Propagation rule
<i>Conv3d</i>	$\#_{\text{filter}} = 5$ 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>		$R_i = R_j$

- [1] Y. LeCun, Y. Bengio, and G. Hinton, *Nature (London)* **521**, 436 (2015).
 [2] A. Graves, A.-r. Mohamed, and G. Hinton, in *2013 IEEE international conference on acoustics, speech and signal processing* (IEEE, New York, 2013), pp. 6645–6649.
 [3] 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., 2012), pp. 1097–1105.
 [4] S. Ren, K. He, R. Girshick, and J. Sun, in *Advances in Neural Information Processing Systems 28*, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Curran Associates, Inc., 2015), pp. 91–99.

- [5] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.
- [6] L. Wang, *Phys. Rev. B* **94**, 195105 (2016).
- [7] P. Broecker, J. Carrasquilla, R. Melko, and S. Trebst, *Sci. Rep.* **7**, 8823 (2017).
- [8] L.-G. Pang, K. Zhou, N. Su, H. Petersen, H. Stöcker, and X.-N. Wang, *Nat. Commun.* **9**, 210 (2018).
- [9] S. J. Wetzel, *Phys. Rev. E* **96**, 022140 (2017).
- [10] S. J. Wetzel and M. Scherzer, *Phys. Rev. B* **96**, 184410 (2017).
- [11] M. Cristoforetti, G. Jurman, A. I. Nardelli, and C. Furlanello, [arXiv:1705.09524](https://arxiv.org/abs/1705.09524).
- [12] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, *Phys. Rev. X* **7**, 031038 (2017).
- [13] W. Hu, R. R. P. Singh, and R. T. Scalettar, *Phys. Rev. E* **95**, 062122 (2017).
- [14] Z. Liu, S. P. Rodrigues, and W. Cai, [arXiv:1710.04987](https://arxiv.org/abs/1710.04987).
- [15] G. Carleo and M. Troyer, *Science* **355**, 602 (2017).
- [16] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Nat. Phys.* **13**, 435 (2017).
- [17] P. Ponte and R. Melko, *Phys. Rev. B* **96**, 205146 (2017).
- [18] J. Carrasquilla and R. G. Melko, *Nat. Phys.* **13**, 431 (2017).
- [19] A. Morningstar and R. G. Melko, *J. Mach. Learn. Res.* **18**, 5975 (2017).
- [20] A. Tanaka and A. Tomiya, [arXiv:1712.03893](https://arxiv.org/abs/1712.03893).
- [21] A. Tanaka and A. Tomiya, *J. Phys. Soc. Jpn.* **86**, 063001 (2017).
- [22] L. Huang and L. Wang, *Phys. Rev. B* **95**, 035105 (2017).
- [23] J. Liu, Y. Qi, Z. Y. Meng, and L. Fu, *Phys. Rev. B* **95**, 041101 (2017).
- [24] D. Wu, L. Wang, and P. Zhang, *Phys. Rev. Lett.* **122**, 080602 (2019).
- [25] K. Zhou, G. Endrődi, L.-G. Pang, and H. Stöcker, *Phys. Rev. D* **100**, 011501 (2019).
- [26] P. E. Shanahan, D. Trewartha, and W. Detmold, *Phys. Rev. D* **97**, 094506 (2018).
- [27] P. Suchsland and S. Wessel, *Phys. Rev. B* **97**, 174435 (2018).
- [28] J. M. Urban and J. M. Pawłowski, [arXiv:1811.03533](https://arxiv.org/abs/1811.03533).
- [29] F. Noé, S. Olsson, J. Köhler, and H. Wu, *Science* **365**, eaaw1147 (2019).
- [30] K. Nicoli, P. Kessel, N. Strodthoff, W. Samek, K.-R. Müller, and S. Nakajima, [arXiv:1903.11048](https://arxiv.org/abs/1903.11048).
- [31] M. S. Albero, G. Kanwar, and P. E. Shanahan, *Phys. Rev. D* **100**, 034515 (2019).
- [32] L. Kades, J. M. Pawłowski, A. Rothkopf, M. Scherzer, J. M. Urban, S. J. Wetzel, N. Wink, and F. Ziegler, [arXiv:1905.04305](https://arxiv.org/abs/1905.04305).
- [33] K. Kashiwa, Y. Kikuchi, and A. Tomiya, *Prog. Theor. Exp. Phys.* **2019**, 083A04 (2019).
- [34] K. Liu, J. Greitemann, and L. Pollet, *Phys. Rev. B* **99**, 104410 (2019).
- [35] C. Casert, T. Vieijra, J. Nys, and J. Ryckebusch, *Phys. Rev. E* **99**, 023304 (2019).
- [36] W. Rządkowski, N. Defenu, S. Chiacchiera, A. Trombettoni, and G. Bighin, [arXiv:1907.05417](https://arxiv.org/abs/1907.05417).
- [37] K. A. Nicoli, S. Nakajima, N. Strodthoff, W. Samek, K.-R. Müller, and P. Kessel, *Phys. Rev. E* **101**, 023304 (2020).
- [38] K. Kashiwa, Y. Kikuchi, and A. Tomiya, *Prog. Theor. Exp. Phys.* **2019**, 083A04 (2019).
- [39] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. Huber, *New J. Phys.* **22**, 045003 (2020).
- [40] P. Mehta, M. Bukov, C.-H. Wang, A. G. R. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, *Phys. Rep.* **810**, 1 (2019).
- [41] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [42] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, *PLoS One* **10**, e0130140 (2015).
- [43] I. Montvay and G. Münster, *Quantum Fields on a Lattice*, Cambridge Monographs on Mathematical Physics (Cambridge University Press, Cambridge, England, 1994).
- [44] S. J. Wetzel, R. G. Melko, J. Scott, M. Panju, and V. Ganesh, [arXiv:2003.04299](https://arxiv.org/abs/2003.04299).
- [45] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, in *NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning—Now What?* (ETH Zurich, 2017).
- [46] G. Montavon, W. Samek, and K.-R. Müller, *Digit. Signal Process.* **73**, 1 (2018).
- [47] 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*, edited by Y. Bengio and Y. LeCun (2014).
- [48] M. Sundararajan, A. Taly, and Q. Yan, [arXiv:1703.01365](https://arxiv.org/abs/1703.01365).
- [49] A. Shrikumar, P. Greenside, and A. Kundaje, in *Proceedings of the 34th International Conference on Machine Learning*, edited by D. Precup and Y. W. Teh, Vol. 70 of Proceedings of Machine Learning Research (PMLR, International Convention Centre, Sydney, Australia, 2017), pp. 3145–3153.
- [50] M. D. Zeiler and R. Fergus, in *Computer Vision—ECCV 2014—13th European Conference, Zurich, Switzerland, 2014, Proceedings, Part I*, edited by D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Vol. 8689 of Lecture Notes in Computer Science (Springer, New York, 2014), pp. 818–833.
- [51] K. Nicoli, P. Kessel, M. Gastegger, and K. T. Schutt, 2018.
- [52] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, [arXiv:1605.01713](https://arxiv.org/abs/1605.01713).
- [53] T. B. Fraunhofer HHI and S. Singapore, <http://heatmapping.org/> (2020).
- [54] L. Breiman, *Mach. Learn.* **45**, 5 (2001).
- [55] 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, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [56] A. Saabas, <https://github.com/andosaa/treeinterpreter> (2015).
- [57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, 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*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), pp. 8024–8035.
- [58] D. P. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).