Interpretable Machine Learning for Inferring the Phase Boundaries in a Non-equilibrium System

C. Casert,* T. Vieijra, J. Nys, and J. Ryckebusch
Department of Physics and Astronomy, Ghent University, Belgium

At the heart of much debate is the question whether machine learning is capable of going beyond black-box modeling of complex physical systems. We investigate the generalizing and interpretability properties of learning algorithms. To this end we use supervised and unsupervised learning to infer the phase boundaries of the Active Ising Model (AIM) starting from an ensemble of configurations of the system. We illustrate that unsupervised learning techniques are powerful at identifying the phase boundaries in the phase space of control variables, even in situations of coexistent phases. It is demonstrated that supervised learning with neural networks is capable of learning the characteristics about the phase diagram, such that the obtained knowledge at a specific set of control variables can be used to trace the phase boundaries across the phase diagram. In this way we demonstrate that properly designed supervised learning provides predictive power to regions in the phase space of control variables that are not included in the training phase of the algorithm. We show that by scrutinizing the inner workings of the classifier, we can extract the physically relevant density and magnetization patterns.

Introduction—Machine learning has recently shown its great potential for addressing non-trivial problems in statistical and many-body physics. Successful applications include detecting phase transitions using neural networks [1–9] and unsupervised learning [10–13], mapping the ground-state wave function of quantum many-body systems and performing quantum state tomography [14–16], and exploiting the apparent similarities between neural networks and the theory of the renormalization group [17, 18]. Due to its expressive power, deep learning has proven to be a powerful tool to identify phase boundaries. Yet, interpretability remains an issue, as it is often unclear how to transfer the features learned by the algorithm to comprehensible physical properties. Thus far, interpretable machine-learning methods for physical systems have often drawn on the use of more transparent (albeit less expressive) learning methods, such as support vector machines [19, 20]. Although promising, little effort has been devoted to obtaining new insight into the properties of a physical system from deep learning [4, 7]. Gaining a more general insight into whether a deep neural network's classification can be built on non-trivial physical features would hence be a major step forward to the development of an interpretable deep learning methodology for selected physics applications.

In this Letter, we sketch a road map for such an interpretable learning methodology that is capable of inferring the high-level features of a system in the phase space of the control variables by merely starting from an ensemble of system configurations. We propose a two-step procedure: (*i*) first, we apply unsupervised learning to identify the phase boundaries in a slice of the phase diagram, (*ii*) subsequently, we use supervised methods to extract the relevant features of the phases labeled in step (*i*). Thereby, we select models that have learned characteristic properties of the physical system by interpreting its internal representation of the phases, and use these to complete the phase diagram. As a prototypical example, we apply our methodology to configurations of the Active Ising Model (AIM), a non-equilibrium spin system with a non-trivial phase diagram. The AIM describes the generic features of collective motion emerging from local interactions in a lattice gas. Collective motion has played a preeminent role in the study of active matter, and the flocking transition has attracted widespread attention due to its universal properties [21–24]. The nature of this phase transition was, however, long uncertain but has now been established to be comparable to a liquid-gas transition. A crucial factor in the understanding of this transition was the introduction of the (2D) Active Ising Model [25–27]. In the AIM, particles with spin projections $s = \pm 1$ undergo biased diffusion along the x-axis, and diffuse freely along the y-axis. Particles hop to the left (right) at a rate $D(1 \mp \epsilon s)$, where D is a diffusion coefficient and ϵ a measure for the self-propulsion. This mechanism tends to spatially separate the two spin configurations along the x-axis. Hopping along the y-axis is symmetric at a rate D. The number $n_{\pm,i}$ of $s = \pm 1$ spins on a lattice site i determines the local density $\rho_i = n_{+,i} + n_{-,i}$ and the local magnetization $m_i = n_{+,i} - n_{-,i}$. Particles on a particular lattice site tend to align their spin through a ferromagnetic interaction: a spin flip occurs at a rate $\exp\left(-s\beta\frac{m_i}{\rho_i}\right)$, with $\beta = 1/T$ the inverse temperature. The global density $\rho_0 = \sum_i \rho_i / L^2$ is fixed. The phase space of control variables of the dynamic system under investigation is defined by the variables (ρ_0, T, D, ϵ). At a fixed $\epsilon > 0$ and $D > 0$, the phase diagram in the (ρ_0, T)-plane has three distinct regions. At high ρ_0 and low T, the system acts like a homogeneous polar liquid (phase 'L'). For low ρ_0 and high T, collective motion is absent and the system behaves like a homogeneous gas (phase 'G') with $\overline{m} = \frac{1}{\rho_0 L^2}\left|\sum_{i=1}^{L^2} m_i\right| \approx 0$. For intermediate values of ρ_0

and T, phase separation is observed in the form of an ordered, high-density band moving through a disorderly, dilute gas (phase 'L+G'). This phase transition is similar to a liquid-gas transition. Its critical point, where the system can continuously transform between liquid and gas, lies at $(\rho_{0,c} = \infty, T_c = 1)$ and no super-critical region exists. In the following, we use $D = 1$ and $\epsilon = 0.9$ without any loss of generality, as these variables only affect the location of the phase boundaries.

Unsupervised learning—We now explore to what extent unsupervised machine learning is capable of uncovering the non-trivial phase diagram of the AIM. First, we consider dimensionality reduction methods to identify the relevant subspaces that characterize the different phases at varying temperatures T and fixed $\rho_0 = 3$. Using unsupervised machine-learning techniques such as principal component analysis (PCA) and uniform manifold approximation and projection (UMAP), we illustrate that one can uncover the characteristics of the three phases. To this end, we introduce the data matrix $\mathcal{D}$, containing N configurations. $\mathcal{D}_{ij}$ represents the magnetization at site j for a temperature T_i. The rows of $\mathcal{D}$ correspond to 50 uncorrelated configurations per temperature $T \in [0.2, 1.0]$ with a temperature spacing $\Delta T = 0.01$. Hence, for an $L \times L$ lattice, $\mathcal{D}$ is an $N \times L^2$ matrix, where $N = 4050$. The uncorrelated AIM configurations in $\mathcal{D}$ are sampled using Markov Chain Monte Carlo.

PCA identifies the dominant features of a data set as the orthogonal and linearly uncorrelated variables (principal components) by which the variance can best be explained [10–12]. The principal components are the orthonormal eigenvectors $\mathbf{w}_i$ of the correlation matrix with the largest eigenvalues λ_i. The results of the PCA analysis of the AIM configurations are displayed in Figs. 1 and 2. Figure 1 shows the explained variance ratio $\bar{\lambda}_i = \lambda_i / \sum_{j=1}^{L^2} \lambda_j$ of the first 25 principal components, where the λ_i are sorted in descending order. The 7D subspace of the L^2-dimensional configurations spanned by $\mathbf{w}_{1-7}$ explains more than 99.9% of the variance in $\mathcal{D}$. The first principal component is given by $\mathbf{w}_1 = \frac{1}{L}[1, \ldots, 1]_{L\times L}$, and corresponds to the total magnetization. The next few leading components ($\mathbf{w}_2$–$\mathbf{w}_7$) appear in pairs with equal $\bar{\lambda}$ and are periodic along the x-direction with a period of resp. L, $L/2$, and $L/3$. The pairwise occurrence of these components is required to describe the band structures in the 'L+G' phase in the translationally invariant system. The small fluctuations along the y-direction are represented by the higher $\mathbf{w}_{i>7}$. Each AIM configuration (i.e. row of $\mathcal{D}$) can be described by a set of projection coefficients p_i, which are the components of the AIM configuration in the lower-dimensional space spanned by the reduced set of principal components (see Fig. 2 (a)-(c)). We denote $\langle p_i \rangle$ as the fixed-temperature average of p_i (Fig. 2 (d)-(e)). From the previous discussion, it is clear that

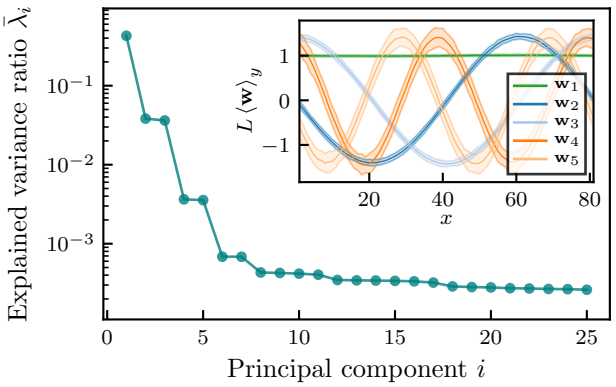

FIG. 1. The explained variance ratio $\bar{\lambda}_i$ for the first 25 principal components of the data matrix $\mathcal{D}$, for $L = 81$ and $\rho_0 = 3$. Inset: The x-dependence of the first five principal components, averaged along the y-direction. The shaded region corresponds to three standard deviations on this average.

$\langle |p_1| \rangle / (\rho_0 L)$ is equal to the traditional order parameter $\overline{m}$. Similarly, as $\left\langle \sqrt{p_2^2 + p_3^2} \right\rangle$ is an indicator for the presence of large-scale inhomogeneities in the magnetization, it is non-zero for temperatures corresponding to the 'L+G' phase. The maximum of $\left\langle \sqrt{p_2^2 + p_3^2} \right\rangle$ allows us to infer the temperature for which the spatial liquid-gas ratio is equal to 1/2, as the components with a period of L dominate for that temperature.

Though PCA is easily interpretable, it requires prior knowledge and physical intuition to separate the phases. Unlike non-linear learning methods, it does not preserve local distances which are described by non-linear correlations when projecting from a high-dimensional to a low-dimensional space. Figure 2(f) shows the result of dimensionality reduction applied to the dataset $\mathcal{D}$ of AIM configurations with a state-of-the-art non-linear technique known as UMAP [28]. This algorithm constructs a fuzzy topological structure for the manifold on which the data is distributed, and then learns a low-dimensional projection of the data of which the fuzzy topological structure best resembles the high-dimensional one. The UMAP is highly efficient in uncovering the different phases of the AIM. Indeed, in the constructed representation with two UMAP components UMAP-1 and UMAP-2, the AIM configurations clearly cluster in five well separated groups. One group contains configurations from the 'G' phase. The symmetry breaking in the 'L' and 'L+G' phases is uncovered by the UMAP algorithm by clearly separating the configurations with positive and negative magnetization. Unlike PCA, UMAP is able to efficiently learn the translational invariance of the bands in the 'L+G' phase and hence requires only two variables to classify the AIM configurations in the three phases. By identifying the temperature ranges of the different clus-

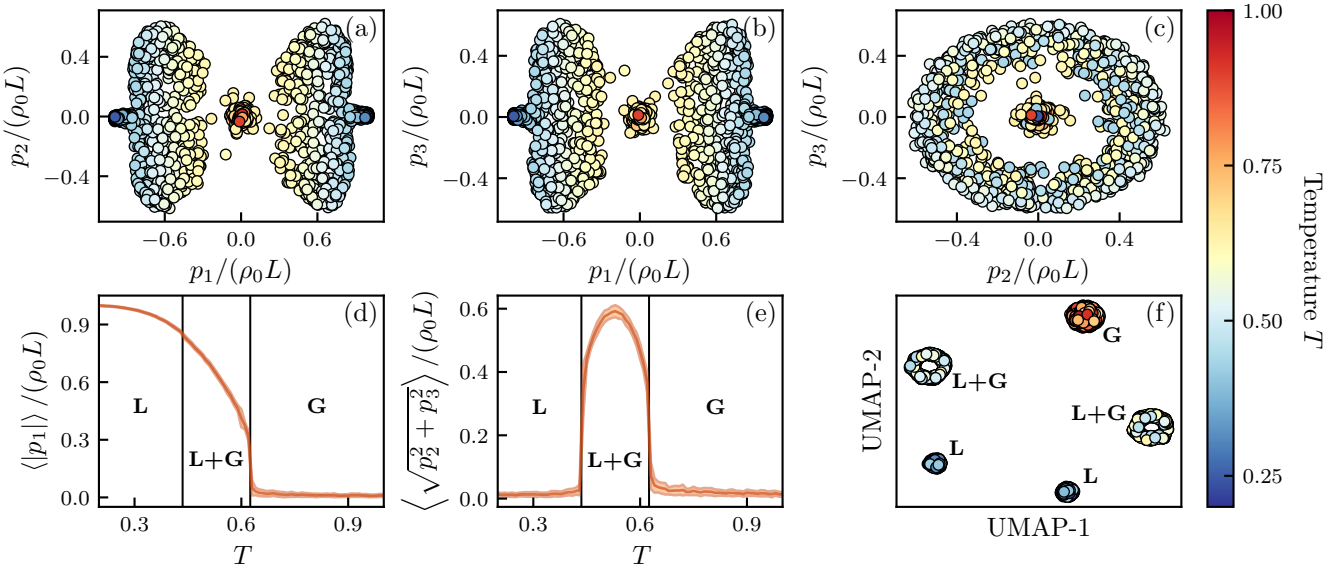

FIG. 2. Classification of AIM configurations at $\rho_0 = 3$ and various temperatures $0.2 \leq T \leq 1$ with the unsupervised PCA (panels (a) to (e)) and UMAP (panel (f)) techniques. (a), (b) and (c) Scatter plots with the projection of the 4050 AIM spin configurations on the first three principal components. (d) Fixed-temperature average of $|p_1|$ in function of T. The shaded region corresponds to one standard deviation. (e) Fixed-temperature average of $\sqrt{p_2^2 + p_3^2}$ in function of T. (f) Clustering of the 4050 AIM configurations with the UMAP algorithm.

ters in the UMAP subspace, we can now easily infer the transition points between the three phases.

Supervised learning—The presented analysis clearly showed that unsupervised learning can determine the temperature boundaries for the different AIM phases at a fixed global density ρ_0. In the forthcoming, we show that supervised learning trained with phase-labeled AIM configurations at a fixed ρ_0 is capable of predicting the phase boundaries in a wide range of ρ_0 values. To this end, a convolutional neural network (CNN) is first trained on a data set of AIM configurations generated at $\rho_0 = 3$. This training is supervised, since the configurations are now labeled with their respective phase, using the unsupervised UMAP approach. Given an input configuration, the network assigns a class score S_c to each of the three phases $c =$ 'L', 'G', or 'L+G', from which the probability P_c for the configuration to belong to phase c can be found after a softmax operation: $P_c = e^{S_c} / \sum_{c'} e^{S_{c'}}$. During training, the loss function $\mathcal{L}$ is minimized by optimizing the model's weights $w \in \mathcal{W}$ which connect the different layers. For each AIM configuration i, the loss function reads $\mathcal{L}_i(Q_i; \mathcal{W}) = \mathcal{H}(Q_i, P_i(\mathcal{W}))$, where $\mathcal{H}$ is the cross-entropy between the predicted $(P_i(\mathcal{W}))$ and the true (Q_i) class probabilities. An additional L2-regularization function with strength λ is included in the total loss function: $\mathcal{L} = \frac{1}{N} \sum_i \mathcal{L}_i + \lambda \sum_{w \in \mathcal{W}} w^2$. The regularization term is introduced to limit the model's complexity and prevents overfitting to the training data.

In order to figure out the features that the CNN has captured, we feed the model with AIM configurations of unknown phase labeling sampled at ρ_0 values not included during training. Networks failing to predict the phase boundaries under those circumstances are likely to have learned trivial features from the $\rho_0 = 3$ data, *e.g.* the local magnetizations m_i crossing a threshold. The phase boundary between two phases A and B is inferred from the temperature for which the predicted class probabilities P_A and P_B coincide [2]. Remarkably, as shown in Fig. 3, the network is able to extrapolate the boundaries it has learned at $\rho_0 = 3$ to a range of densities $0.5 \leq \rho_0 \leq 8.0$ extending over more than one order of magnitude. Networks trained without regularization ($\lambda = 0$) can perfectly classify unseen AIM configurations sampled at the control parameters ($\rho_0 = 3$ and $T \in [0.2, 1.0]$) used during the training phase of the CNN. Yet, they often fail in classifying the proper phase for configurations with control variables that were not included during training. Thereby, the minimum of the loss function heavily focuses on details specific to the $\rho_0 = 3$ configurations, but has failed to grasp the more general features of the AIM. The regularizing term in the loss function limits the model's complexity, but does not impact its classification accuracy on both the training and test set. We find that for $\lambda > 0$ the CNN extracts the more physically relevant characteristics and gains the potential to accurately determine phase boundaries at ρ_0 far away from the training set. On top of that, its inferred extrapolation of the phase boundaries is

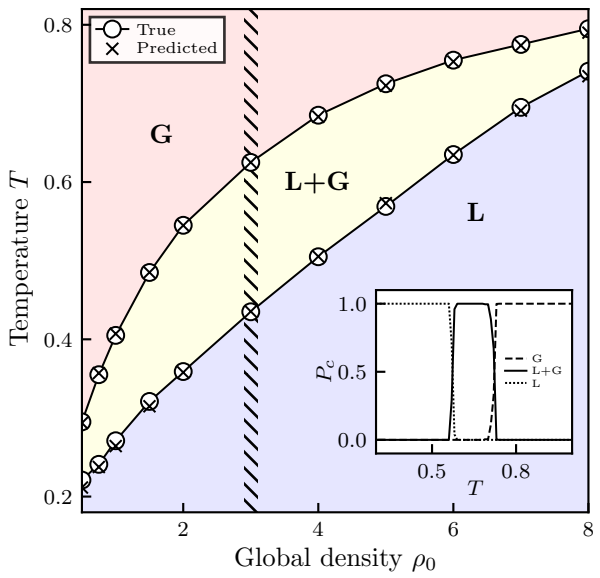

FIG. 3. The phase boundaries of the AIM in the (ρ_0, T)-plane. The circles are the true phase boundaries, while the crosses are inferred with a CNN that is only trained on configurations with $\rho_0 = 3$ (shaded region). Inset: The CNN's prediction of the temperature dependence of the probability P_c of an AIM configuration to belong to the 'L', 'G', 'L+G' phases for $\rho_0 = 5$.

more robust, meaning that the results depend less on the initial weight parameters and choices with regard to the training set. In other words, each run of the optimization routine results in a particular set of weight parameters. Although they reach a similar accuracy, only a selected subset succeeds in making a physically relevant classification. The latter illustrates the pitfall of merely using classification accuracy for model selection, without scrutinizing the learned features.

Interpretability—The question arises about the physical relevance of the features learned during the network's training, and the specific role of the hyperparameters in this. For this purpose, we introduce saliency maps [29], a technique that is commonly used in image recognition to gain insight into "black-box" classifiers. Given an AIM configuration I of phase c, to which the network assigns a class score S_c, we compute the gradient $|\partial S_c/\partial I|$ through back propagation. As a result, we can highlight the regions of I that heavily impact the classification. Those regions are interpreted by the CNN as phase-characteristic and reminiscent of the physical features. To illustrate the potential of saliency maps in phase characterization, we first train a network on AIM configurations for all global densities shown in Fig. 3. For the 'L' and 'G' phases, the gradient $|\partial S_c/\partial I|$ attains only small values, which reflects that the model has captured the homogeneity of these phases. The 'L+G' phase is much more challenging with regard to phase classification. A prototypical saliency map for the 'L+G' phase is shown in Fig. 4 for a vanishing and non-vanishing λ. The magnitude of λ strongly impacts the locality of the extracted characteristics. Saliency maps are a powerful instrument to distill the global emergent properties, *i.e.* the diffuse edges between liquid and gas, given appropriate regularization.

Summary—We have demonstrated that a sequential application of unsupervised and supervised machine learning is a powerful instrument to infer and characterize the phase diagram of a liquid-gas transition, without any a priori knowledge of its phase boundaries. Advanced dimensionality reduction methods, such as UMAP, clearly cluster system configurations into the physical phases and recognize the presence of symmetry breaking. By feeding a CNN with phase-labeled configurations, we demonstrated that well-designed networks, trained to learn the phase boundaries for fixed control parameters, are capable of extrapolating the phase boundaries to complete the phase diagram in a large range of control parameters. Thereby, it is of crucial importance to properly select the network architectures and hyperparameters. Indeed, we have demonstrated that networks of comparable classification

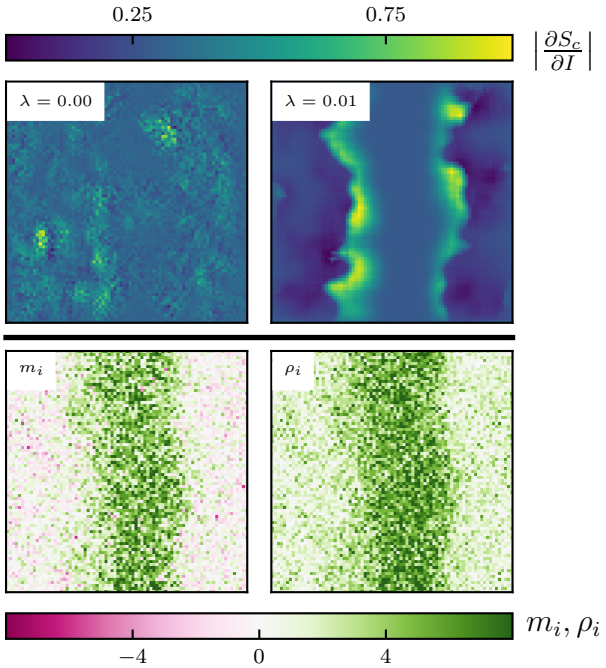

FIG. 4. Saliency maps and physical properties of an 81×81 AIM configuration in the 'L+G' phase with $\rho_0 = 3$, $T = 0.56$. Top panels: $|\partial S_c/\partial I|$, normalized between 0 and 1, for a network trained with L2-regularization strength $\lambda = 0$ and $\lambda = 0.01$. Bottom panels: local magnetization and local density.

performance can either learn physically relevant features or meaningless properties. The selectivity of hyperparameters, such as regularization in the loss function, is a tool to discriminate between these networks. By employing interpretation tools, such as saliency maps, the inferred classification can be connected with the essential physical features, *e.g.* the phase-characteristic magnetization and density patterns.

We are indebted to Benjamin Vandermarliere, Andres Belaza, Ken Bastiaensen and Wesley De Neve for useful discussions. The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center). This work was supported by Ghent University, Research Foundation Flanders (FWO-Flanders) and the Flemish Government – department EWI. J. Nys was supported as an 'FWO-aspirant'.

* corneel.casert@ugent.be

[1] J. Carrasquilla and R. G. Melko, Nature Physics **13**, 431 (2017).
[2] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nature Physics **13**, 435 (2017).
[3] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).
[4] P. Zhang, H. Shen, and H. Zhai, Phys. Rev. Lett. **120**, 066401 (2018).
[5] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[6] M. J. S. Beach, A. Golubeva, and R. G. Melko, Phys. Rev. B **97**, 045207 (2018).
[7] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[8] P. Suchsland and S. Wessel, Phys. Rev. B **97**, 174435 (2018).
[9] J. Venderley, V. Khemani, and E.-A. Kim, Phys. Rev. Lett. **120**, 257204 (2018).
[10] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[11] W. Hu, R. R. P. Singh, and R. T. Scalettar, Phys. Rev. E **95**, 062122 (2017).
[12] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[13] K. Ch'ng, N. Vazquez, and E. Khatami, Phys. Rev. E **97**, 013306 (2018).
[14] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[15] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Nature Physics **14**, 447 (2018).
[16] G. Torlai and R. G. Melko, Phys. Rev. Lett. **120**, 240503 (2018).
[17] P. Mehta and D. J. Schwab, arXiv:1410.3831 (2014).
[18] M. Koch-Janusz and Z. Ringel, Nature Physics **14**, 578 (2018).
[19] P. Ponte and R. G. Melko, Phys. Rev. B **96**, 205146 (2017).
[20] J. Greitemann, K. Liu, and L. Pollet, arXiv:1804.08557 (2018).
[21] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, Phys. Rev. Lett. **75**, 1226 (1995).
[22] H. Chaté, F. Ginelli, G. Grégoire, and F. Raynaud, Phys. Rev. E **77**, 046113 (2008).
[23] T. Vicsek and A. Zafeiris, Physics Reports **517**, 71 (2012).
[24] A. Filella, F. Nadal, C. Sire, E. Kanso, and C. Eloy, Phys. Rev. Lett. **120**, 198101 (2018).
[25] A. P. Solon and J. Tailleur, Phys. Rev. Lett. **111**, 078101 (2013).
[26] A. P. Solon, H. Chaté, and J. Tailleur, Phys. Rev. Lett. **114**, 068101 (2015).
[27] A. P. Solon and J. Tailleur, Phys. Rev. E **92**, 042119 (2015).
[28] L. McInnes and J. Healy, arXiv:1802.03426 (2018).
[29] K. Simonyan, A. Vedaldi, and A. Zisserman, arXiv:1312.6034 (2013).
[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015), software available from tensorflow.org.
[31] F. Chollet *et al.*, "Keras," `https://keras.io` (2015).

Configurations of the Active Ising Model

An example of the magnetization and density in a configuration for each of three phases of the AIM is depicted in Fig. 5. We generate the AIM configurations with a random-sequential-update algorithm, where the time step $\Delta t = (4D + \exp\beta)^{-1}$ is chosen in order to minimize the probability that no state change occurs during a single update [27]. Since data generation is computationally expensive, we use data augmentation. Hereby, new configurations are generated from the original $N = 4050$ samples, by uniformly shifting these in both the x and y direction. These are appended as additional rows to the data matrix $\mathcal{D}$.

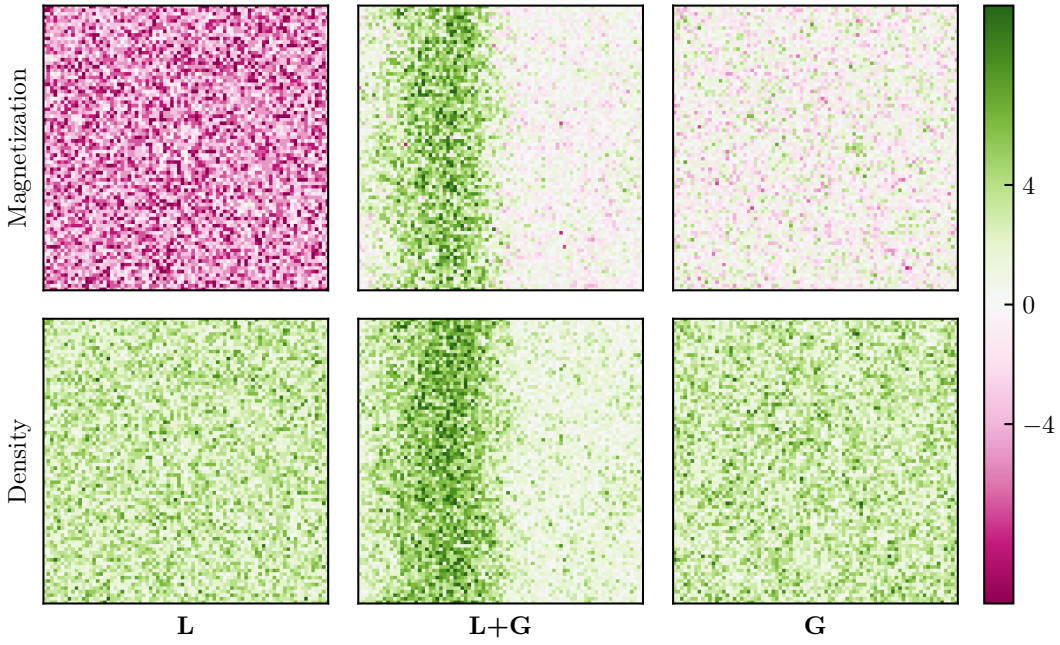

FIG. 5. Magnetization (top row) and density (bottom row) maps of AIM configurations in the three phases: (from left to right) 'L' ($\rho_0 = 3$, $T = 0.25$), 'L+G' ($\rho_0 = 3$, $T = 0.55$) and 'G' ($\rho_0 = 3$, $T = 0.95$).

PCA dimensionality reduction

In this section, we provide illustrations that accompany the discussion on PCA decomposition in the main text. In particular, we demonstrate the effect of projecting an AIM configuration from the original $L \times L$ space to the 7D subspace defined by the principal components, and of backprojecting to the large L^2-dimensional configuration space. In Fig. 6, we illustrate the decomposition of a given AIM magnetization map into the basis of principal components. As can be clearly observed in the right-most panel of Fig. 6, the PCA projection essentially washes out all statistical fluctuations from the original configuration.

CNN architecture and training

The convolutional neural network architecture used to determine and characterize the phase boundaries is shown in Fig. 7. The input layer consists of two channels: magnetization and density. The first two convolutional layers (C1) each have 6 filters with a (5×5) kernel and have a ReLU activation function. These layers are followed by a max pooling layer, with a (3×3) kernel and stride 3. Pooling is included to reduce the model complexity and to make the observed features less orientation and scale dependent. The next two convolutional layers (C2) also contain 6 filters with ReLU activation, but now with a (3×3) kernel, and are followed by the same max pooling operation. The flattened feature vector is then sent through a fully-connected network, where the first layer has 16 hidden nodes with ReLU activations. The output layer has three nodes, one for each of the three different phases, and a softmax

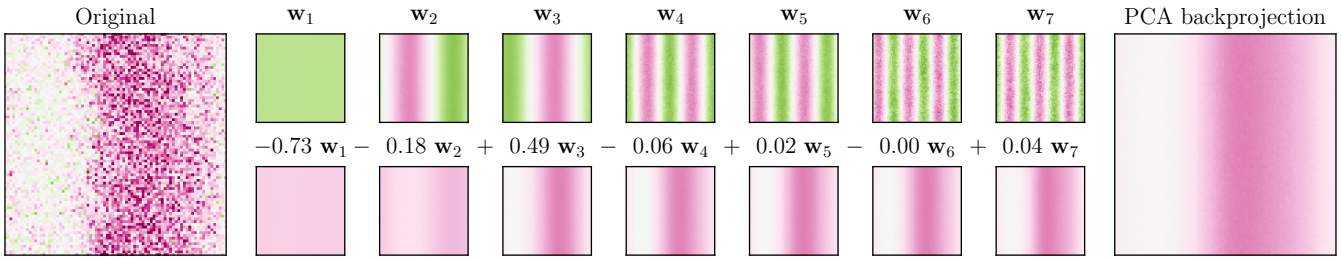

FIG. 6. For a given magnetization map of an AIM configuration in the 'L+G' phase (left), we illustrate its PCA backprojected contributions (bottom row) related to the normalized principal components (top row). The numerical projection coefficients are given explicitly for every component. The result is a denoised configuration (right).

activation. The network is defined by a total of $\left(\frac{32}{27}L^2 + 1939\right)$ weights and biases, which are trained using an Adam optimizer with learning rate $\alpha = 10^{-3}$. We found that the learned classification was rather insensitive to the value of the learning rate α. The data is split into a training set (60% of the total data), a validation set (20%) and a test set (20%). To avoid overfitting on the training set, the loss function is evaluated on the validation set after every training epoch. The model with the lowest loss on the validation set is kept. When no decrease in validation loss is detected for 100 consecutive training epochs, the training is terminated ("early stopping") and the network is evaluated on the independent test set. The neural network and its training are implemented using TensorFlow [30] and Keras [31].

In addition to the saliency maps, we interpret the inner workings of the CNN by visualizing the kernels of the first convolutional layer in Figure 8. This clearly illustrates that the first layer of the CNN trained with $\lambda = 0.01$ detects more robust features (*e.g.* gradients in the local density values) compared to the non-regularized network.

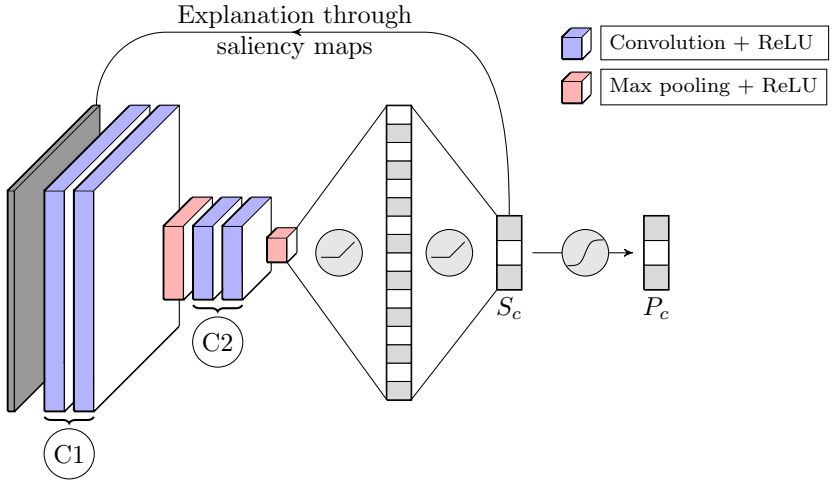

FIG. 7. Architecture of the CNN used for inferring the phase boundaries in the control parameter space of the AIM.

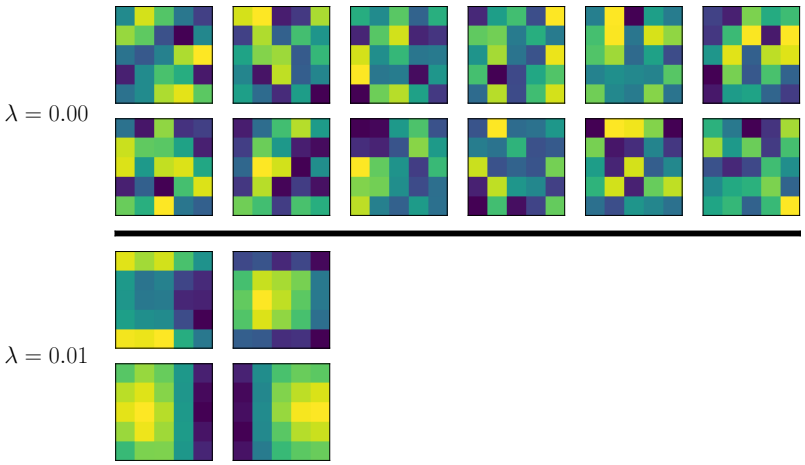

FIG. 8. Normalized kernels of the first layer of the CNN after training. Only kernels with absolute values of the weights larger than 0.01 are shown. The top (bottom) two rows correspond to $\lambda = 0$ ($\lambda = 0.01$). The top and bottom row for each λ represent the filters operating on the magnetization and density, respectively.