Identifying quantum phase transitions with adversarial neural networks

Patrick Huembeli, Alexandre Dauphin, and Peter Wittek
ICFO-Institut de Ciencies Fotoniques, Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain

(Received 15 November 2017; revised manuscript received 1 February 2018; published 11 April 2018)

The identification of phases of matter is a challenging task, especially in quantum mechanics, where the complexity of the ground state appears to grow exponentially with the size of the system. Traditionally, physicists have to identify the relevant order parameters for the classification of the different phases. We here follow a radically different approach: we address this problem with a state-of-the-art deep learning technique, adversarial domain adaptation. We derive the phase diagram of the whole parameter space starting from a fixed and known subspace using unsupervised learning. This method has the advantage that the input of the algorithm can be directly the ground state without any *ad hoc* feature engineering. Furthermore, the dimension of the parameter space is unrestricted. More specifically, the input data set contains both labeled and unlabeled data instances. The first kind is a system that admits an accurate analytical or numerical solution, and one can recover its phase diagram. The second type is the physical system with an unknown phase diagram. Adversarial domain adaptation uses both types of data to create invariant feature extracting layers in a deep learning architecture. Once these layers are trained, we can attach an unsupervised learner to the network to find phase transitions. We show the success of this technique by applying it on several paradigmatic models: the Ising model with different temperatures, the Bose-Hubbard model, and the Su-Schrieffer-Heeger model with disorder. The method finds unknown transitions successfully and predicts transition points in close agreement with standard methods. This study opens the door to the classification of physical systems where the phase boundaries are complex such as the many-body localization problem or the Bose glass phase.

DOI: 10.1103/PhysRevB.97.134109

I. INTRODUCTION

The intersection of many-body physics and machine learning is an emergent area of research that has produced spectacular successes in a short span of time. Generative machine learning models are able to represent the many-body wave function even with long-range correlations [1–7], tensor networks commonly used in many-body physics are also useful for machine learning [8,9], and machine learning is effective in studying phase transitions in many-body systems [10–21]. This latter direction is the one we pursue.

A phase diagram shows qualitative changes in many-body systems as functions of parameters in a physical system. The task of physicists is to identify the correct order parameters. For example, in the Landau theory of phase transitions, a discontinuity of the local order parameter or of one of its derivatives indicates a phase transition. In more exotic systems, the order parameters are global as it is the case for topological phases or topological insulators. The search of the *right* order parameters and the derivation of the phase diagram in terms of the parameters of the Hamiltonian prove to be very challenging tasks. Already for noninteracting Hamiltonians where the addition of disorder or quasiperiodic disorder can lead to Anderson localization [22,23] or to topological phase transitions [24,25], distinguishing the phases can be demanding. Even more surprisingly, the interplay of disorder and interactions can give rise to many-body localization [26].

We can think of quantum states matching a particular choice of parameters as data instances with a label that is the corresponding phase. This approach provides a link to machine learning, where the task is to discriminate data instances with different labels. Different strategies can be adopted for the choice of the inputs of the neural network. The first one would be to feed the order parameter or several order parameters to the machine and let it find the phase transition points. This approach is very intuitive for physicists but its main weakness is the requirement of *ad hoc* engineering as one has to know which are the relevant order parameters. The second one would be to feed directly the ground state of the Hamiltonian to the algorithm and let the machine itself discover the order parameters and the phase transition points. In this work, we follow the second strategy.

The discrimination of the phases can happen via supervised training, when the labels are known in advance, or via unsupervised training, when the labels are unknown. The latter is clearly harder, but it is also more interesting from a physics perspective, since it would allow us to map out an unknown phase diagram. The feasibility of unsupervised training has already been demonstrated. Standard unsupervised methods, such as principal component analysis or t-distributed stochastic neighbor embedding (t-SNE), have been used to characterize phase transitions in several systems such as the Ising model, the XY model, or the Hubbard model [10,16,17]. Other works used shallow neural networks, i.e., fully connected neural networks with a few layers, to characterize models such as the Ising model or the Bose-Hubbard model [11,15–17]. The latter approaches all used fully connected learners, which do not scale well with the input size and depth of the network and have limited ability to extract features from the input [27]. Therefore the input of the neural network had to be either

small or hand-crafted (as in the case of using the entanglement spectrum or the correlation function).

This stands in contrast to deep learning, which revolutionized machine learning and artificial intelligence by providing automated means of extracting high-quality feature spaces from raw data [28]. Deep learning networks, however, struggle with the unsupervised scenario, and they are mainly applied in supervised problems. A body of work studied classical [12] and quantum [29,30] phase transitions, and even topological phases [31,32] with deep architectures this way.

Since automated feature extraction is desirable to investigate more complex systems, a few recent works ventured into using unsupervised deep learning techniques for studying phase transitions. Boltzmann machines are a computationally expensive but highly expressive method [33], and computationally efficient feedforward convolutional neural networks (CNNs) can be tweaked in some cases to perform unsupervised learning [34] or so called transfer learning [30].

In this work, we show that adversarial domain adaptation [35] unleashes the power of deep learning in a wide range of many-body physics problems to find the phase transition in an unsupervised manner. This approach avoids *ad hoc* feature engineering and does not make assumptions about the input data, relying on deep learning to extract an expressive feature space. Furthermore our architecture allows us to scale the size of the input because of the convolutional neural network. The unsupervised approach presented in Refs. [11,15] on the other hand is only viable for shallow networks as a deep neural network would be able to learn any mislabeled distribution with a high accuracy. Furthermore, deep approaches are much more efficient in computational resources as the network does not have to be retrained for every point in the parameter space and for a series of different labelings. This enables building much deeper neural networks for automated feature extraction and learning more complex distributions, and once the representation is extracted, the scheme is fully unsupervised.

We illustrate the different machine learning techniques presented in the introduction on a concrete example that will be studied in detail in this work, the Su-Schrieffer-Heeger model with disorder, and discuss the strengths and weaknesses. The first approach is to feed order parameters to the neural network, as for instance the entanglement spectrum, and train a shallow neural network for the case without disorder and apply the trained neural network to the case with disorder. The main weakness of this approach is the *ad hoc* engineering where one has to know the relevant order parameters. The second is to train a deep convolutional neural network directly on the ground state of the system without disorder and apply transfer learning to learn the phase diagram with disorder. This method has been shown to be very powerful [30], but works poorly when applied to systems with disorder, as we will see later in this work. The third is the domain adversarial adaptation where we have two data sets. One contains the labeled source states, which are the ground states without disorder, and the other contains the unlabeled target states, which are the ground states with disorder. The objective of this technique is to extract invariant features from the ground states of the two data sets. By construction, this algorithm is very efficient for classifying noisy data and therefore outperforms transfer learning.

The rest of the article is structured as follows. Section II reviews the idea of domain adversarial adaptation and discusses how this algorithm can be a powerful tool for the classification of phase transitions. Section III demonstrates the efficiency of the technique on several paradigmatic models: the Ising model with different temperatures, the Bose-Hubbard (BH) model, the Su-Schrieffer-Heeger (SSH) model with disorder, and long-range hopping. Section IV provides the technical details of the algorithms. Finally, Sec. V is dedicated to the conclusions and outlook.

II. UNSUPERVISED LEARNING WITH DOMAIN ADVERSARIAL NEURAL NETWORKS

A. General idea

The core idea is letting a deep architecture develop an intuition on a physical problem, and transfer it to a different system. The simple form of this is called transfer learning, that is, training CNNs on one domain and fine-tuning them on another, and it is known to give good results. This is true even if the second domain only uses the feature extraction layers that were trained on the first domain, in combination with unsupervised learning [36]. A more focused approach is what we follow: we use a domain adversarial neural network (DANN)—also known as adversarial domain adaptation—where the feature extraction layers of a CNN are trained to be invariant between a supervised source data distribution and a potentially unsupervised target data distribution [35].

A DANN consists of three parts: a feature extractor, a label predictor, and a domain classifier (see Fig. 1). The neural network is trained such that the feature representations of the two domains are invariant and the domain classifier cannot distinguish between them anymore. The label classifier is only trained on the source data.

We predict the labels of the target distribution, after the training, by feeding the target states to the feature extractor and the label predictor, without the domain classifier. Alternatively we can apply unsupervised algorithms such as t-SNE [37,38], k-means clustering [39], or density-based spatial clustering of applications with noise [40] directly on the feature representation.

Intuitively, this method should allow the efficient study of models with disorder or noise, where transfer learning tends to fail. Furthermore the invariant feature space allows further studies with unsupervised methods to even detect new phases, as we will show in Sec. III D.

B. Domain adversarial neural networks

The three parts of the domain adversarial neural network (DANN) are the feature extractor $G_f(\mathbf{x},\Theta_f)$, the label predictor $G_y(\mathbf{x}',\Theta_y)$, and the domain classifier $G_d(\mathbf{x}',\Theta_d)$ (see Fig. 1), where Θ_d,Θ_y are the domain classifier and label predictor parameters and Θ_f are the feature extractor parameters.

The labeled input data of the well known model are called the source distribution $\mathcal{S} = \{(x_s, y_s)\}$, where the distribution of the unknown model, without labels, is called the target distribution $\mathcal{T} = \{x_t\}$. Our goal will be to predict the labels y_t for given inputs x_t of our target distribution. To distinguish whether the input x_i is coming from the source or target

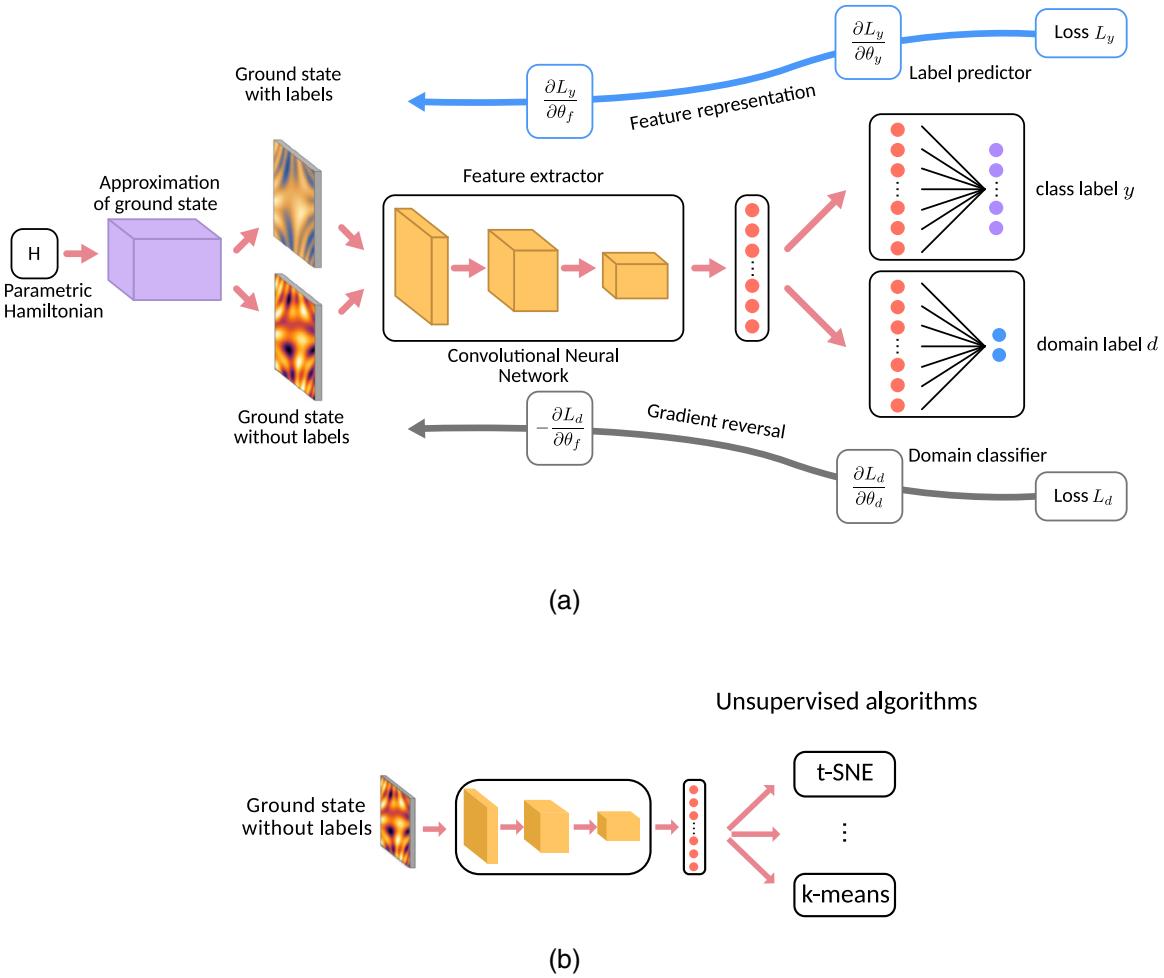

FIG. 1. Schematic representation of our architecture. (a) Given a parametric Hamiltonian, we find the ground states of two different distributions. For one of them—the source—we know the labels. For the other one—the target—we do not. A convolutional neural network is used as a feature extractor. The final layer of the representation is fed into a domain and a label classifier to find the correct phase labeling and to identify which domain the data comes from, respectively. The gradient reverse layer adds a negative constant to the back propagation of the domain classifier, which makes the feature distributions of the two domains similar. (b) We send the unlabeled examples across the trained feature extractor, and feed the high-dimensional representation to unsupervised learning methods to identify the phase transition.

distribution, we introduce the domain label d_i, which is $d_i = 0$ if x_i is from our source distribution or $d_i = 1$ if x_i is from the target distribution. During the training of the DANN, we feed the input $x \in \mathcal{S} \cup \mathcal{T}$ into the feature extractor where it is mapped to a high-dimensional feature vector $\mathbf{f} = G_f(\mathbf{x}, \Theta_f)$.

The feature extractor consists of convolutional neural networks, composed of many different filters. Compared to a fully connected neural network, in a CNN, for each filter only a small number of weights are trained, defining a receptive field that is slid across the whole image.

After a convolutional layer, we apply a max-pooling layer to further reduce the dimensionality of the input. This is achieved by forwarding the maximum value of a fixed-sized tiling window that scans the image.

Following a series of convolutional and pooling layers, we obtain an abstract, high-level feature representation. The feature vector $\mathbf{f}$ is fed into the label predictor $G_y(\mathbf{f}, \Theta_y)$ to output the labels y and into the domain classifier $G_d(\mathbf{f}, \Theta_d)$. Since there are only labeled data for the source part of the input x, the loss of the label predictor can only be calculated by the source part of the feature vector $\mathbf{f}$. The loss of the domain classifiers can be calculated on the full input $\mathcal{S} \cup \mathcal{T}$.

To train the network we define the domain and classifier losses L_d, L_y. As described in Ref. [35], the domain classifier loss is a regularization of the label predictor. Therefore the training of the DANN optimizes $E(\Theta_f, \Theta_y, \Theta_d) = L_y(\Theta_f, \Theta_y) - L_d(\Theta_f, \Theta_d)$ by finding the saddle point

$$(\Theta_f, \Theta_y) = \underset{\Theta_f, \Theta_y}{\operatorname{argmin}} E(\Theta_f, \Theta_y, \Theta_d), \tag{1}$$

$$(\Theta_d) = \underset{\Theta_d}{\operatorname{argmax}} E(\Theta_f, \Theta_y, \Theta_d). \tag{2}$$

The update rule for the feature extractor therefore has the form

$$\Theta_f \leftarrow \Theta_f - \mu\left(\frac{\partial L_y}{\partial \Theta_f} - \frac{\partial L_d}{\partial \Theta_f}\right), \tag{3}$$

which can be implemented via stochastic gradient descent and the gradient reversal layer [35].

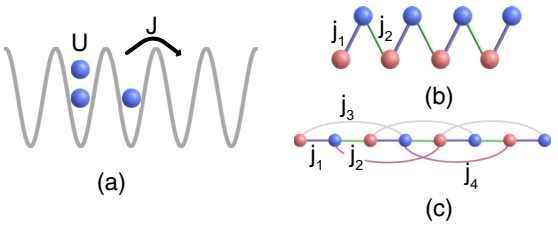

FIG. 2. Sketches of the Bose-Hubbard model (a), the SSH model (b), and the SSH model with long-range hopping (c).

The domain classifier should not be able to distinguish the two domains because their feature representation is invariant. This is achieved by training the parameters of the domain classifier Θ_d such that the domain loss L_d is minimal. At the same time, the parameters Θ_f of the feature extractor are identified by minimizing the function $E(\Theta_f,\Theta_y,\Theta_d)$. Since the domain loss also depends on the feature extraction parameters Θ_f, this optimization problem has an adversarial character and leads to a competition between optimizing the domain classifier and the label prediction loss or $E(\Theta_f,\Theta_y,\Theta_d)$. This results in a domain classifier that is well trained, but is unable to distinguish the domains, as the feature representation of the two domains is invariant. For the label predictor's output, the training is similar except that both parameters Θ_f and Θ_y minimize the classifier loss.

To predict the labels of the target distribution, we can either apply the label predictor or directly use unsupervised methods such as t-SNE or k-means on the feature representation.

III. RESULTS

We now apply our method to several paradigmatic models to benchmark its performance.

A. Ising model

We study the 2D square-lattice Ising model in the presence of a local random magnetic field [41], $H = -J\sum_{\langle i,j\rangle}\sigma_i\sigma_j - \sum_j h_j\sigma_j$, where σ_i are classical spins, J is the interaction, and $h_i \in [-h,h]$ are local random magnetic fields. The presence of random fields shifts the critical temperature T_c associated with the phase transition. We generate samples of configurations for 20×20 sites with Monte Carlo simulations. The phase transition for $h = 0$ can be found analytically and provides a labeled source data. The configuration in the presence of random fields is the unlabeled target data. The phase transition found by the algorithm agrees with the literature [41]. We notice, however, that in this simple case, a convolutional neural network without domain adaptation has the same performance. In other words, elementary transfer learning suffices (see Sec. A 1 of the Appendix).

B. Bose-Hubbard model

As a next benchmark for the performance of the DANN algorithm we choose the Bose-Hubbard model with a mean-field treatment, which has also been used as a benchmark of Ref. [15]. We investigate the 2D Bose-Hubbard model

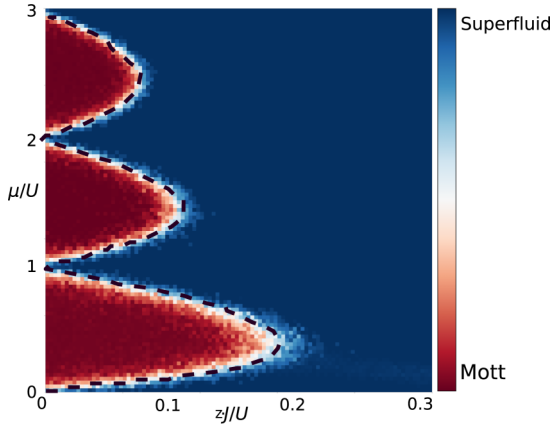

FIG. 3. Phase diagram of the Bose-Hubbard model predicted by the label classifier of the DANN. The dashed line represents the phase transition directly calculated via the compressibility κ. The predictions of the DANN and the exact value are in good agreement.

[Fig. 2(a)] with Hamiltonian

$$H = -J\sum_{\langle i,j\rangle}(b_i^\dagger b_j + b_j^\dagger b_i) + U\sum_i n_i(n_i-1) - \mu\sum_i n_i, \tag{4}$$

chemical potential μ, nearest neighbor hopping J, and on-site interaction strength U. This model experiences phase transitions at zero temperature from Mott insulating to superfluid phases [42]. The inputs of the neural network are the Gutzwiller coefficients [43] with a maximum number of bosons per site of $n = 20$. The Gutzwiller coefficients have been found with a simulated annealing method [44]. Since the Gutzwiller approach maps the 2D Bose-Hubbard model to a string of coefficients the input data are one-dimensional and therefore the convolutional neural network is also one-dimensional. An arbitrary line of the phase diagram at a fixed $z\,J/U = 0.005$ is labeled for all the values of μ with the help of the compressibility $\kappa = \partial\langle n_i\rangle/\partial\mu$ [45]. Here, $z = 4$ is the number of nearest neighbors of each site, which is 2 for the one-dimensional case. The target samples are unlabeled states for a different value of $z\,J/U = 0.1$. After training on these sets, we apply the domain adaptation algorithm on states of the whole phase diagram. Results are presented in Fig. 3. The algorithm recovers the celebrated Mott lobes [42], and the predicted phase transitions match the ones obtained from the literature, as well as the phase transition obtained directly from the compressibility (dashed line). At the tip of the first Mott lobe the phase transition occurs at $J/U = 1/(5.8z)$ [42,46]. For the higher Mott lobes the transition point is at around $J/U = 4\bar{n}z$, where $\bar{n}$ is the boson density and at the same time the number of the lobe.

C. SSH model with disorder

The SSH model [Fig. 2(b)] is a one-dimensional chiral model that exhibits topological properties: this system is characterized by a global topological invariant, the winding number. The latter predicts the number of protected edge states appearing at each edge of a finite-size chain with

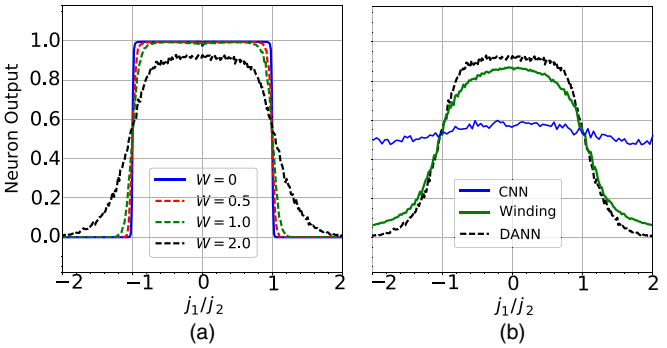

FIG. 4. SSH with open boundary conditions. (a) Neuron output of the SSH model for different disorder strength W and for a system of 64 sites with open boundary conditions. The results are averaged over 1000 disorder realizations. (b) Comparison of the phase predictions of a convolutional neural network without domain adaptation (CNN), with the domain adversarial approach (DANN), and with the winding number for different values of j_1 and fixed $W = 2$. While the transfer learning fails, the phase transitions predicted by the DANN are in good agreement with the winding number.

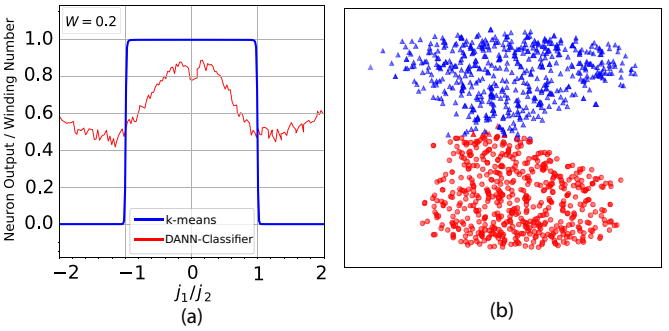

FIG. 5. (a) k-means classification applied on the feature space of the DANN trained on SSH model with periodic boundary conditions compared to the DANN label classifier output. For periodic boundaries. The classifier of the trained DANN cannot distinguish the phases of states with disorder. (b) Clustering of the two phases with t-SNE. The shapes indicate the correct labeling; the colors show the labeling found by k-means. If we apply k-means clustering on the low-dimensional embedding provided by t-SNE on the feature space, the labeling works well and the phase boundary can be found with an accuracy of $j_1/j_2 = 1 \pm 0.01$. The plot shows the SSH model with periodic boundary conditions with disorder strength $W = 0.2$.

open boundary conditions. We apply the DANN to study the phase diagram in the presence of disorder. In this case, the Hamiltonian of noninteracting spinless fermions reads

$$H_{\mathrm{SSH}} = \sum_n j_{1,n} c_n^\dagger \sigma_1 c_n + j_{2,n} [c_n^\dagger \sigma_+ c_{n+1} + \mathrm{H.c.}], \quad (5)$$

where the σ_i are the Pauli matrices and $\sigma_+ = \sigma_1 + i\sigma_2$. The disorder appears in the hopping parameters $j_{1,n} = j_1 + W_1\omega_n$ and $j_{2,n} = j_2 + W_2\omega_n'$, where ω_n and ω_n' are randomly distributed numbers in the interval $[-0.5,0.5]$. In the following, we set $j_2 = 1$.

For this noninteracting system, the input state is composed of all the eigenstates below the Fermi energy $E_F = 0$ of a system of 64 sites. We find numerically these occupied states with the help of exact diagonalization. The input data of the DANN are a matrix where each column is an eigenstate of the Hamiltonian below the Fermi energy. We generate source states for $W = 0$ and label them analytically [47]: the states in the trivial phase have label 0 and the states in the topological phase have label 1. We then generate target states in the presence of disorder $W_1 = 2W_2 = W = 2$ where the correct labeling is unknown.

1. Open boundary conditions

We first apply the algorithm for the system with open boundary conditions. Figure 4(a) shows the classifier output for different disorder strengths averaged over 1000 disorder realizations. We correctly identify a shift of the topological phase transition with increasing disorder, which is in accordance with Ref. [25]. Furthermore, we compared the phase transition points with the one obtained from the winding number defined in Ref. [25], shown in Fig. 4(b). Remarkably, the DANN predicts precisely the transition point.

2. Periodic boundary conditions

We then focus on the case of periodic boundary conditions. Here, the label predictor fails to accurately predict the phase transition, as presented in Fig. 5(a). The label classifier does not show real plateaus nor is the phase boundary agreeing with the literature. This is related to the fact that, within periodic boundary conditions, the classifier has to find a global property of the bulk of the system. Nevertheless, we can still perform unsupervised learning directly on the feature representation. We first apply the t-SNE algorithm [37] which allows us to reduce the dimension of the feature representation to 2. Figure 5(b) shows the t-SNE plot for one realization of disorder $W = 0.2$. The trivial (circles) and topological states (triangles) form two clearly separated clusters that can be labeled with k-means clustering. This method allows us to find the phase transition with good accuracy. In Sec. A 3 of the Appendix, we also show that transfer learning is portable between two different models: the SSH and the Kitaev model. In this case, domain adaptation works because both models show edge states with open boundary conditions.

3. Comparison between transfer learning and domain adversarial adaptation

To compare the efficiency of domain adversarial adaptation to the one of transfer learning, we train a neural network composed of a feature extractor and a classifier on the states without disorder. The architecture of the feature extractor and the classifier is chosen to be the same as the one of the domain adaptation. Figure 4(b) compares the predictions of the phase diagram of the SSH model with disorder $W = 2.0$ with transfer learning (blue), with domain adaptation (dashed dark), and with the winding number (green). In this case, the transfer learning fails to reproduce the topological phase transition in the presence of disorder.

D. SSH model with long-range hopping

We now consider the SSH model with nearest neighbor hopping j_1 and j_2, and third nearest neighbor hopping j_3 and

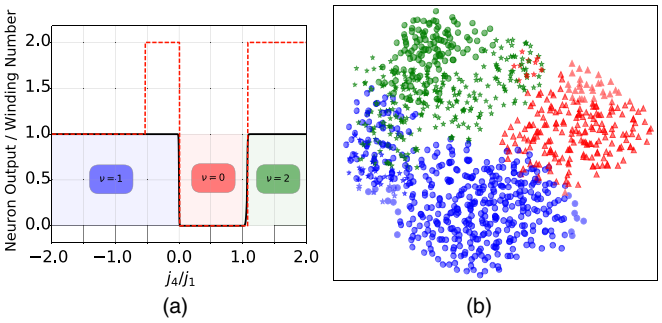

FIG. 6. SSH long-range label prediction of the classifier (solid line). If we fix $j_1 = j_2 = 1$ and $j_3 = 0$ we can find phase transitions $\nu = 1 \to 0$ at $j_4 = 0$ and $\nu = 0 \to 2$ at $j_4 = 1$ [48]. The dashed line shows the labeling found by k-means directly on the feature space. We can see that there is a mislabeling at the boundary between $\nu = 1$ and $\nu = 0$. In color are the effective winding numbers. j_1 and j_2 are the nearest neighbor hopping terms; j_3 and j_4 are the third nearest neighbor hopping terms. In the right panel is the SSH long-range feature space classification via k-means and graphical embedding by t-SNE. The shapes indicate the correct labeling; the colors show the labeling found by k-means.

j_4, as shown in Fig. 2(c), which has the Hamiltonian

$$H = H_{\mathrm{SSH}} + \sum_n j_{3,n} c_n^\dagger \sigma_1 c_{n+1} + j_{4,n}[c_n^\dagger \sigma_+ c_{n+2} + \mathrm{H.c.}]. \tag{6}$$

In this case, the phase diagram becomes richer with higher winding numbers [48]. By considering third nearest neighbor hopping j_3 and j_4, in addition to the winding numbers $\nu = 0, 1$, we can also obtain winding number $\nu = \pm 1, \pm 2$. Our purpose is to see whether our scheme allows one to predict unseen phases. As before, we generate source states for the SSH model for $j_2 = 1$, $j_3 = j_4 = 0$ and label them analytically with windings 0 and 1. We then produce target states for the SSH with long-range hopping for $j_2 = j_1 = 1$ and $j_3 = 0$. Although the classifier has been trained to distinguish data points with windings 0 and 1, it accurately detects phase transitions between trivial and topological phases, as shown in Fig. 6(a) (solid line). Furthermore, when analyzing the feature space directly, in addition to the clustering trivial/topological phases we find a subclustering in the topological phase. The k-means algorithm can predict the labels of the trivial phase ($\nu = 0$) with high accuracy. The transition between winding numbers $\nu = 1$ and $\nu = 2$, on the other hand, is not accurate close to the phase transition, as shown in Fig. 6(a) (dashed line). Nevertheless, far from the phase transition, the k-means algorithm labels the phases correctly.

IV. METHODS

To ensure the reproducibility of our results, we made the source code available under an open source license [49].

A. Neural network

The feature extractor of our DANN consists of two convolutional layers each with 32 filters. For two-dimensional inputs (SSH), the receptive field size is 3×3 and the pooling size is 2×2. For the Bose-Hubbard model we choose one-dimensional convolutional networks with a receptive field of length 3 and pooling size 2. The activation functions for the convolutional layers are rectified linear units (ReLUs). The label predictor and the domain classifier are built in the same way: they contain 128 hidden ReLU neurons and 2 softmax output neurons. The difference between them is the gradient reversal layer between the feature extractor and the domain classifier. The input size in 2D is for every model 64×64, and for the one-dimensional Gutzwiller coefficients, the input size is 21. To prevent overfitting we use dropout and the cost function is the categorical cross-entropy. The learning rate is similar to [35] slowly decaying and defined as $\mu = \mu_0/(1 + \alpha \cdot p)^\beta$, where $\mu_0 = 0.001$, $\alpha = 10$, $\beta = 0.75$, and p is the training progress linearly changing from 0 to 1.

B. Input data

We produced the input data via different approaches, dependent on the model. The SSH can be diagonalized exactly. The input data for the DANN are the fermionic occupied states, which are the states with negative energy eigenvalues and the zero-energy state. We arrange these states in a matrix, where the eigenstates are the columns. For practical reasons we use the states two times to have a square matrix of the size $N \times N$, where N is the system size.

The configurations of the 2D-Ising model were found via Monte Carlo methods. The input data of the DANN are simply the square lattice configuration.

For the Bose-Hubbard model we choose the Gutzwiller ansatz [43]. The maximum boson number per site is fixed at $n_{\max} = 20$. We calculate the Gutzwiller coefficients for every configuration (J, U, μ), via simulated annealing. Since our input data in this case are a 1D vector of coefficients, we use 1D convolutional layers instead of a 2D one.

V. CONCLUSION

As humans, we often gain an intuition on a physical system using a special case that is analytically or numerically easy to treat. Then we generalize the insights to the more complex cases. Domain adaptation captures this idea: a deep learning system extracts intuition on a well-understood system and applies it to a more perplexing one. This is a subtle, targeted application of machine learning, with the explicit purpose of avoiding brute force numerical methods. We demonstrated the applicability of the method on several paradigmatic models: the 2D Ising model, the Bose-Hubbard model, and the SSH model. The phase diagram found by the algorithm is in very good agreement with the one obtained with standard methods and even with analytical calculations. Therefore it allows the characterization of classical, quantum, and topological phase transitions. Furthermore, the algorithm can even predict new phases as shown in the long-range SSH model. In future studies, we will focus on interacting Hamiltonians, Bose glass, and many-body localization. We will also study the scaling of the order parameter predicted by the neural network in terms of the system's size.

ACKNOWLEDGMENTS

We thank Antonio Acín, Jacob Biamonte, Maciej Lewenstein, Gorka Muñoz-Gil, James Quach, and Gabriel Senno for comments on the manuscript. The authors acknowledge financial support from Spanish MINECO Severo Ochoa Grant No. SEV-2015-0522, Generalitat de Catalunya Grants No. 874 and No. 875, the CERCA Program, and Fundación Cellex. P.H. and P.W. were further supported by QIBEQI FIS2016-80773-P. A.D. is financed by a Cellex-ICFO-MPQ fellowship, and acknowledges financial support from FisicaTeAMO FIS2016-79508-P and EU grants OSYRIS (ERC-2013-AdG Grant No. 339106) and QUIC (H2020-FETProAct-2014 641122). P.W. acknowledges financial support from the ERC (Consolidator Grant QITBOX) and a hardware donation by Nvidia Corporation.

APPENDIX

1. Ising model

The classical 2D Ising Hamiltonian reads

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - \sum_j h_j \sigma_j, \tag{A1}$$

with the classical spin representation $\sigma_i \in \{-1,1\}$, the interaction J, and the magnetic field h_i. This model has a well known phase transition at the temperature $T \approx 2.27J$ for $h_i = 0$. If we apply a noisy random magnetic field according to Ref. [41] the critical temperature shifts to lower values for increasing h. We first sample lattice configurations $\{\vec{s}_{h=0}\}$ from Monte Carlo simulation for zero magnetic field and different temperatures. Since the critical temperature of the phase transition is well known, we can label this configuration accordingly. The configurations with magnetic field are again provided by Monte Carlo techniques, whereas the magnetic field is randomly drawn from $\{-h,h\}$ for each site and each Monte Carlo sampling step.

We notice however that, in this case, the domain adaptation is not necessary to find the phase transition. A convolutional neural network can already learn the Ising order parameter, which is essentially the sum over all spins. Therefore a CNN that is trained on $\{\vec{s}_{h=0}\}$ can already extract the features of Ising configurations $\{\vec{s}_{h=1.5}\}$ without any fine-tuning for transfer learning. Figure 7 shows the output of our neural network where we did not apply domain adaptation.

2. Bose-Hubbard

a. Effective phase boundary

The effective phase boundary of the produced states can be calculated via the compressibility $\partial \langle n \rangle / \partial \mu$, which is equal to 0 in the Mott phase and is different from 0 in the superfluid phase, whereas $\langle n \rangle = \langle \sum_i n_i \rangle$ is the sum over the thermal averages of the occupation number of each site. Besides that the particle number is constant in the Mott phase, we also know that it has integer values for each Mott lobe.

b. Generation of the data

Figure 3 has been found by the DANN with the labeled source data along the line $J = 0.005$. The target has been chosen arbitrarily to be along $J = 0.1$. After training the DANN, we can feed Gutzwiller coefficients calculated from any point in the parameter space into the DANN and we find the phase diagram. We want to emphasize here that this figure is the direct output of the DANN for Gutzwiller coefficients calculated for a 100×100 grid in the parameter space $(\mu/U, J/U)$ averaged over 20 realizations without further data processing. The noise of the output comes from the way the Gutzwiller coefficients are calculated. The simulated annealing we used is a stochastic method and can sometimes get stuck in a local minimum.

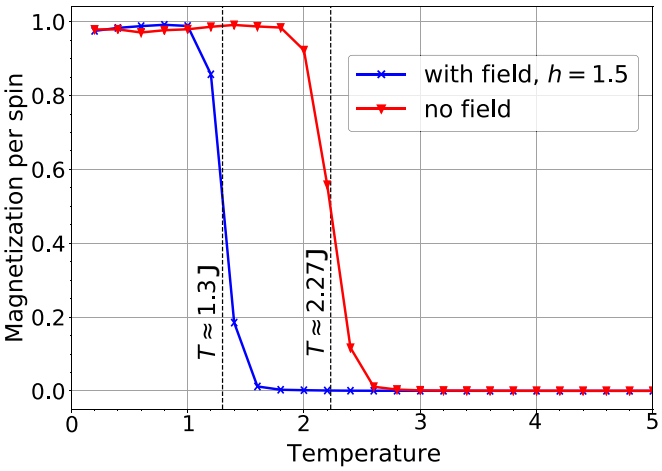

FIG. 7. CNN phase prediction of Ising configurations with and without random magnetic field. Triangular markers show the training set with $T_{\text{crit}} \approx 2.27$ and the cross markers show the test set with $T_{\text{crit}} \approx 1.3$.

3. Kitaev model for spinless fermions

In the SSH model, we can find the phase transition by adapting the domain from a well understood case (without disorder)

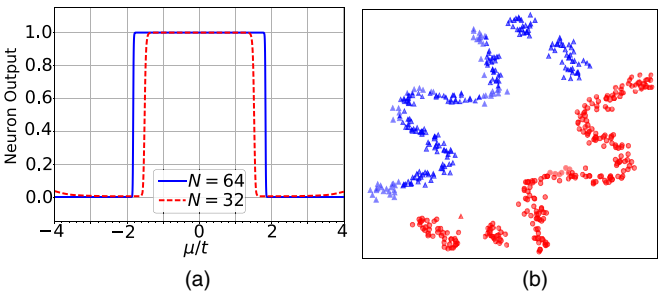

FIG. 8. (a) DANN label prediction of the classifier for the Kitaev model. The learned phase transition of the SSH model can be transferred via domain adaptation. The prediction of the transition point gets better for bigger system sizes. The inaccuracy of the predictions comes from the fact that the DANN classifier was not trained on the Kitaev inputs, only on SSH. (b) k-means applied to the feature space directly after the convolutional layers shows an accurate labeling. The shapes indicate the correct labeling; the colors show the labeling found by k-means. The effective phase transition occurs at $\mu/t = |2|$. The k-means algorithm finds almost all labels correctly (colors). We obtain an error of $\Delta\mu/t = -0.01$. The spatial clustering has been done by the t-SNE method.

to an unknown case (with disorder). The next step will be to show that domain adaptation under certain restrictions even works within two different models. For this case we study the Kitaev model for spinless fermions, which is an important prototypical example for topologically protected edge states. The Hamiltonian reads

$$H = -t \sum_{\langle i,j \rangle} (c_i^\dagger c_j) + \Delta \sum_{\langle i,j \rangle} (c_i^\dagger c_j^\dagger + c_i c_j) - \mu \sum_i c_i^\dagger c_i, \tag{A2}$$

with the hopping parameter t, the pairing Δ, and the chemical potential μ. For simplicity, we choose $t = \Delta$. In this case the ground state has a phase transition from a topologically trivial phase for $|\mu| > 2t$, to a nontrivial phase at $|\mu| < 2t$. Again we choose the fermionic ground states of the SSH model without disorder to be our source input of the DANN and the ground states of the Kitaev model are the target input.

The domain adaptation from the SSH to the Kitaev model works, because they both have topological phases with edge states. Figure 8(a) shows the label prediction of the trained DANN of the Kitaev input instances.

The label prediction shows a clear phase transition, but it is slightly shifted with respect to the analytical predictions. The accuracy of the transition point can be improved with bigger system sizes. In Fig. 8(b), we analyze directly the feature space via the k-means algorithm for a system size $N = 64$ and find the transition point with an error of $\Delta\mu/t = -0.01$. The direct evaluation of the feature space leads to better results than the DANN classifier. This is due to the fact that the DANN classifier never was trained on the target distribution.

[1] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[2] D.-L. Deng, X. Li, and S. D. Sarma, Quantum Entanglement in Neural Network States, Phys. Rev. X **7**, 021021 (2017).

[3] X. Gao and L.-M. Duan, Efficient representation of quantum many-body states with deep neural networks, Nat. Commun. **8**, 662 (2017).

[4] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, Equivalence of restricted Boltzmann machines and tensor network states, Phys. Rev. B **97**, 085104 (2018).

[5] Y. Nomura, A. Darmawan, Y. Yamaji, and M. Imada, Restricted-Boltzmann-machine learning for solving strongly correlated quantum systems, Phys. Rev. B **96**, 205152 (2017).

[6] H. Saito, Solving the Bose-Hubbard model with machine learning, J. Phys. Soc. Jpn. **86**, 093001 (2017).

[7] H. Saito and M. Kato, Machine learning technique to find quantum many-body ground states of bosons on a lattice, J. Phys. Soc. Jpn. **87**, 014001 (2018).

[8] E. M. Stoudenmire and D. J. Schwab, Supervised learning with tensor networks, Advances in Neural Information Processing Systems **29**, 4799 (2016).

[9] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Unsupervised generative modeling using matrix product states, arXiv:1709.01662.

[10] L. Wang, Discovering phase transitions with unsupervised learning, Phys. Rev. B **94**, 195105 (2016).

[11] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. **13**, 435 (2017).

[12] J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. **13**, 431 (2017).

[13] L. Zdeborová, Machine learning: New tool in the box, Nat. Phys. **13**, 420 (2017).

[14] F. Schindler, N. Regnault, and T. Neupert, Probing many-body localization with neural networks, Phys. Rev. B **95**, 245134 (2017).

[15] Y.-H. Liu and E. P. L. van Nieuwenburg, Self-learning phase boundaries by active contours, arXiv:1706.08111.

[16] S. J. Wetzel, Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders, Phys. Rev. E **96**, 022140 (2017).

[17] K. Ch'ng, N. Vazquez, and E. Khatami, Unsupervised machine learning account of magnetic transitions in the Hubbard model, Phys. Rev. E **97**, 013306 (2018).

[18] M. Koch-Janusz and Z. Ringel, Mutual information, neural networks and the renormalization group, arXiv:1704.06279.

[19] Ce Wang and H. Zhai, Machine learning of frustrated classical spin models. I. Principal component analysis, Phys. Rev. B **96**, 144432 (2017).

[20] W. Hu, R. R. P. Singh, and R. T. Scalettar, Discovering phases, phase transitions, and crossovers through unsupervised machine learning: A critical examination, Phys. Rev. E **95**, 062122 (2017).

[21] N. C. Costa, W. Hu, Z. J. Bai, R. T. Scalettar, and R. R. P. Singh, Principal component analysis for fermionic critical points, Phys. Rev. B **96**, 195138 (2017).

[22] P. W. Anderson, Absence of diffusion in certain random lattices, Phys. Rev. **109**, 1492 (1958).

[23] S. Aubry and G. André, Analyticity breaking and Anderson localization in incommensurate lattices, Ann. Israel Phys. Soc. **3**, 18 (1980).

[24] J. Li, R.-L. Chu, J. K. Jain, and S.-Q. Shen, Topological Anderson Insulator, Phys. Rev. Lett. **102**, 136806 (2009).

[25] I. Mondragon-Shem, T. L. Hughes, J. Song, and E. Prodan, Topological Criticality in the Chiral-Symmetric AIII Class at Strong Disorder, Phys. Rev. Lett. **113**, 046802 (2014).

[26] R. Nandkishore and D. A. Huse, Many-body localization and thermalization in quantum statistical mechanics, Annu. Rev. Condens. Matter Phys. **6**, 15 (2015).

[27] Y. Bengio and Y. LeCun, Scaling learning algorithms towards AI, in *Large-Scale Kernel Machines*, edited by L. Bottou and O. Chapelle (MIT Press, Cambridge, 2007).

[28] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature (London) **521**, 436 (2015).

[29] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Machine learning quantum phases of matter beyond the fermion sign problem, Sci. Rep. **7**, 8823 (2017).

[30] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Machine Learning Phases of Strongly Correlated Fermions, Phys. Rev. X **7**, 031038 (2017).

[31] D.-L. Deng, X. Li, and S. D. Sarma, Exact machine learning topological states, Phys. Rev. B **96**, 195145 (2017).

[32] P. Zhang, H. Shen, and H. Zhai, Machine Learning Topological Invariants with Neural Networks, Phys. Rev. Lett. **120**, 066401 (2018).

[33] A. Morningstar and R. G. Melko, Deep learning the Ising model near criticality, arXiv:1708.04622.

[34] P. Broecker, F. F. Assaad, and S. Trebst, Quantum phase recognition via unsupervised machine learning, arXiv:1707.00663.

[35] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, Domain-adversarial training of neural networks, Journal of Machine Learning Research **17**, 1 (2016).

[36] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri, CNN features are also great at unsupervised classification, arXiv:1707.01700.

[37] L. Van der Maaten and G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research **9**, 2579 (2008).

[38] M. Wattenberg, F. Viégas, and I. Johnson, How to use t-SNE effectively, Distill (2016), doi:10.23915/distill.00002.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer, New York, 2002).

[40] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in *Proceedings of SIGKDD-96, 2nd International Conference on Knowledge Discovery and Data Mining* (ACM, New York, NY, 1996), pp. 226–231.

[41] N. Crokidakis, First-order phase transition in a 2D random-field Ising model with conflicting dynamics, J. Stat. Mech.: Theor. Exp. (2009) P02058.

[42] M. Lewenstein, A. Sanpera, and V. Ahufinger, Ultracold Gases in Optical Lattices: Basic Concepts, in *Ultracold Atoms in Optical Lattices: Simulating Quantum Many Body Systems* (Oxford University Press, Oxford, 2017).

[43] W. Krauth, M. Caffarel, and J.-P. Bouchaud, Gutzwiller wave function for a model of strongly interacting bosons, Phys. Rev. B **45**, 3137 (1992).

[44] T. Comparin, https://github.com/tcompa/BoseHubbard Gutzwiller/tree/v1.0.2, 2017.

[45] Y. Khorramzadeh, F. Lin, and V. W. Scarola, Boson core compressibility, Phys. Rev. A **85**, 043610 (2012).

[46] W. Zwerger, Mott-Hubbard transition of cold atoms in optical lattices, J. Opt. B **5**, S9 (2003).

[47] J. K. Asbóth, L. Oroszlány, and A. Pályi, *A Short Course on Topological Insulators: Band-structure Topology and Edge States in One and Two Dimensions* (Springer, Berlin, 2016).

[48] M. Maffei, A. Dauphin, F. Cardano, M. Lewenstein, and P. Massignan, Topological characterization of chiral models through their long time dynamics, New J. Phys. **20**, 013023 (2018).

[49] P. Huembeli, A. Dauphin, and P. Wittek, PatrickHuembeli/Adversarial-Domain-Adaptation-for-Identifying-Phase-Transitions: DANN_Arxiv_Version_01, 2017, https://github.com/PatrickHuembeli/Adversarial-Domain-Adaptation-for-Identifying-Phase-Transitions.