MACHINE LEARNING
Science and Technology

PAPER

Characterizing out-of-distribution generalization of neural networks: application to the disordered Su–Schrieffer–Heeger model

Kacper Cybiński[1], Marcin Płodzień[2], Michał Tomza[1], Maciej Lewenstein[2,3], Alexandre Dauphin[2,4] and Anna Dawid[5,*]

[1] Faculty of Physics, University of Warsaw, Pasteura 5, 02-093 Warsaw, Poland
[2] ICFO—Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology, Av. Carl Friedrich Gauss 3, 08860 Castelldefels (Barcelona), Spain
[3] ICREA, Pg. Lluís Campanys 23, 08010 Barcelona, Spain
[4] PASQAL SAS, 7 rue Léonard de Vinci, 91300 Massy, Paris, France
[5] Center for Computational Quantum Physics, Flatiron Institute, 162 Fifth Avenue, New York, NY 10010, United States of America
* Author to whom any correspondence should be addressed.

E-mail: a.m.dawid@liacs.leidenuniv.nl

Keywords: out-of-distribution generalization, interpretability, disorder, topological phases of matter

OPEN ACCESS

RECEIVED
26 June 2024

REVISED
17 September 2024

ACCEPTED FOR PUBLICATION
8 November 2024

PUBLISHED
22 January 2025

Abstract

Machine learning (ML) is a promising tool for the detection of phases of matter. However, ML models are also known for their black-box construction, which hinders understanding of what they learn from the data and makes their application to novel data risky. Moreover, the central challenge of ML is to ensure its good generalization abilities, i.e. good performance on data outside the training set. Here, we show how the informed use of an interpretability method called class activation mapping, and the analysis of the latent representation of the data with the principal component analysis can increase trust in predictions of a neural network (NN) trained to classify quantum phases. In particular, we show that we can ensure better out-of-distribution (OOD) generalization in the complex classification problem by choosing such an NN that, in the simplified version of the problem, learns a known characteristic of the phase. We also discuss the characteristics of the data representation learned by a network that are predictors of its good OOD generalization. We show this on an example of the topological Su–Schrieffer–Heeger model with and without disorder, which turned out to be surprisingly challenging for NNs trained in a supervised way. This work is an example of how the systematic use of interpretability methods can improve the performance of NNs in scientific problems.

1. Introduction and motivation

Machine learning (ML) promises a revolution in science, similar to the current revolution in industry [1]. In quantum physics, neural networks (NNs) serve as a flexible and promising representation of quantum states [2–4] and a booster for quantum technologies [5–7], e.g. as entanglement classifiers [8, 9]. Neural networks are especially promising in the detection of phases of matter and have been used for classical [10–15], quantum [16–30], and topological [31–39] phase transitions with supervised [10, 11, 27–33] and unsupervised [12–26, 34–39] approaches as well as for experimental data [40–44]. Other examples include ML approaches relying on dimensionality reduction [45–48], kernel methods [49–51], topological data analysis [52–55], and quantum NNs [56–58].

However, before NNs join standard toolboxes for the analysis of phases of matter, they need to become more interpretable (so we understand what they learn) and reliable (so we can trust their predictions). Note that interpretability is a stronger condition than reliability (if we understand exactly how an NN makes its predictions, we usually can trust it). There are extensive efforts to make automated approaches more interpretable [17, 23, 37, 54, 59–63], which should ultimately lead to learning phases of matter and assisting

physicists in understanding the learned order parameters [43, 64–67]. However, the community focuses predominantly on a special representation of data, that is, spin configurations. Therefore, the question of the interpretability and reliability of NNs applied to different quantum data remains wide open.

Reliable NNs are expected to generalize robustly to new scenarios [68]. It is especially challenging in the case of the out-of-distribution (OOD) generalization when the test data come from a different distribution than the training data. Naturally, no OOD generalization should be expected for unrelated training and test distributions. However, a robust model also performs well under a limited distribution shift. Such an OOD generalization of a network can be checked when we have enough information on the test distribution; for example, in a supervised scenario, we have access to labels of some OOD test data. When we do not have such labels, our trust in the OOD generalization of an NN has to be limited, especially in the presence of spurious correlations in the training data.

In this work, we show how we can increase trust in the OOD generalization of an NN in the absence of labeled OOD data. To this end, we study explanations of NN predictions and the representation of data learned by an NN and discuss patterns that correlate well with the network robustness. We perform this analysis on an example of data coming from a prototypical topological Hamiltonian, that is, the Su–Schrieffer–Heeger (SSH) model. The training data come from the standard SSH model, while the OOD test data are from the SSH model with the disorder, as presented schematically in figure 1(a). In both regimes, the network task is the same, i.e. it is to classify topological and topologically trivial phases. Therefore, this work belongs to a subfield of transfer learning, called domain adaptation [69].

Previous work [34] showed that standard convolutional NNs (CNNs) trained on the data from the SSH model struggle to generalize under the disorder. The authors solved this problem by using a domain adversarial NN. Here, we do not aim to improve this solution. Instead, we want to understand the reason for the failures of standard CNNs and propose tools that can inform the user whether to expect an OOD generalization from a trained CNN or not without labeled data.

This paper is structured as follows. We start by describing the data set and the learning task of a CNN in sections 2.1 and 2.2. In section 2.3, we discuss an interpretability technique called class activation mapping (CAM) [70], which provides explanations of CNN predictions as sketched in figure 1(b). To study the data representation learned by a CNN, we need a dimensionality reduction technique such as principal component analysis (PCA), presented schematically in figure 1(c), which we describe in section 2.4. We present and discuss our results in section 3 and conclude in section 4.

2. Methods

2.1. The model Hamiltonian

The model proposed by Su, Schrieffer, and Heeger [72] describes spinless fermions on a one-dimensional chain in a tight-binding approximation, with staggered nearest-neighbor tunneling amplitudes. It is described by the Hamiltonian

$$\hat{H}_0 = v\sum_{n=1}^{N}\hat{c}_n^\dagger\hat{\sigma}_x\hat{c}_n + w\sum_{n=1}^{N-1}\left(\frac{1}{2}\hat{c}_n^\dagger\left(\hat{\sigma}_x + i\hat{\sigma}_y\right)\hat{c}_{n+1} + \text{h.c.}\right). \tag{1}$$

The SSH Hamiltonian can have alternative definitions depending on the conventions [73–75] (see appendix A for more details). We consider a chain of N unit cells. Within each unit cell, there are two sites that belong to sublattice A and B, respectively. The operator $\hat{c}_i^{(\dagger)} = \left(\hat{c}_{A,i}^{(\dagger)}, \hat{c}_{B,i}^{(\dagger)}\right)$ denotes the annihilation(creation) operators on the respective sublattices in the system. σ_i denotes the Pauli matrices. The tunneling amplitudes are different for intercell tunneling (between neighboring cells) – w and for intracell tunneling (within a single unit cell) – v. For the remainder of this work, we fix the intercell tunneling to $w = 1$.

Such a system exhibits two distinct phases: topological and trivial. They are characterized by a different value of a *topological invariant*, called the winding number, which arises from the chiral symmetry present in the system. For periodic boundary conditions (PBCs), the invariant is *winding number* ϑ. For open boundary conditions (OBCs), the topological phase is characterized by the real space winding number and by the presence of zero-energy eigenstates, corresponding to the half-filling of the system [74]. These states are exponentially localized at the edges of the system. For this reason, they are referred to as *edge states* or *edge modes*. The number of such zero-energy edge states on one edge is equal the winding number of the bulk. It can be shown by bulk-boundary correspondence [72], that both formulations are equivalent; therefore, in this work we use both approaches.

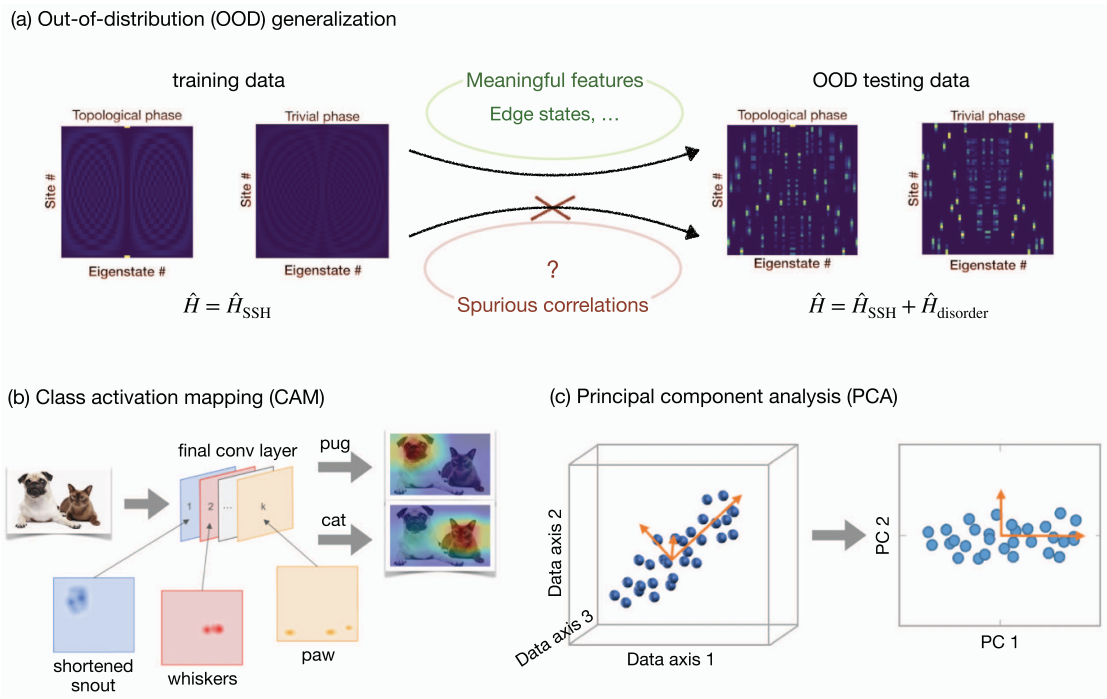

Figure 1. Unsupervised study of out-of-distribution generalization with class activation mapping and principal component analysis. (a) We train multiple randomly initialized CNNs to predict phases in the SSH model without the disorder (simple regime) and study their OOD generalization to phases in the SSH model with disorder (complex regime). Only a handful of trained CNNs learn meaningful features (instead of spurious correlations) that allow for successful OOD generalization. (b) We analyze what data features are the most important for predictions of well- and poorly-generalizing CNNs with class activation mapping (CAM) – an interpretability technique producing heatmaps that after overlaying on input samples indicate the most important data elements for the prediction. (c) With principal component analysis (PCA), we study the similarity of the representation learned by poorly- and well-generalizing CNNs for the data from the simple and complex regime. Panels (b) and (c) are inspired by [70] and Reproduced from [71]. CC BY 4.0.

The interplay between the topological invariants of the system and the presence of the disorder can be studied with the following Hamiltonian [73],

$$\hat{H}_1 = \sum_{n=1}^{N-1} t_n \left(\frac{1}{2} \hat{c}_n^\dagger \left(\hat{\sigma}_x + i \hat{\sigma}_y \right) \hat{c}_{n+1} + \mathrm{h.c.} \right) + \sum_{n=1}^{N} m_n \hat{c}_n^\dagger \hat{\sigma}_x \hat{c}_n. \tag{2}$$

Here, the tunneling amplitudes t_n and m_n vary from site to site. They include the added disorder in the following way

$$t_n = w + 2W\omega_n, \quad m_n = v + W\omega_n'. \tag{3}$$

The parameter W is the strength of the disorder, and ω_n is a random variable drawn from the uniform distribution on the interval $[-0.5, 0.5]$. The topological invariants, to some extent, are robust to disorder. However, for a strong enough value of the disorder, they are likely to change. Therefore, to account for disorder in winding number calculation, we use a handy approximation based on infinite system winding number calculation [76] (see appendix B for details). The relationship between winding number ϑ and disorder strength W/w for systems of the symmetry class such as equation (2) has been thoroughly studied in [73], where they proposed a phase diagram for this scenario. The phase diagram we calculate using the infinite system winding number approximation is in good agreement with their results.

2.2. Learning task and data set

In this work, we task a CNN with learning a mapping between a full set of eigenstates of the SSH Hamiltonian from equation (1) and the corresponding winding number. We describe the CNN architecture and training hyperparameters in detail in appendix C. To obtain the input data, we compute the eigenstates of equation (1) using exact diagonalization [77] with OBC. The corresponding winding numbers are computed by taking the same Hamiltonian with the PBC imposed. This is an approximate calculation, which translates into calculating the winding number of an infinite system [76].

The input data are matrices whose columns are squared moduli of the eigenstates of the system, sorted by energy. We present exemplary input data samples in figure 1(a). The labels are corresponding winding numbers. Because of the presence of chiral symmetry, the energy spectrum is symmetric, and the zero-energy edge states correspond to the two middle columns of an input data sample. The amplitude of the edge states peaks at the edges and decays exponentially into the bulk; therefore, we expect the edge pixels of the middle columns to be the most indicative of these states in the topological phase.

We want to understand the general behavior of NNs, so we study multiple instances of CNNs obtained by training them using the same hyperparameters but from different random initializations, following the default uniform initialization in PyTorch [78]. We train and test our CNNs on data from a system without added disorder, described by equation (1). We train them on data sampled in equal numbers from both topological and trivial phases, far from the phase transition. The loss function we optimize in this task is defined by the binary cross-entropy function [79]. We test CNNs on data sampled uniformly over a whole range of v/w, including the close vicinity of the phase transition. The v/w values used to generate data points in the three data sets (training, validation, and test) were different to prevent possible leakage of information. Such choice of testing points allows us to assess the networks' *generalization*, that is, its performance on previously unseen examples that are sampled from the same underlying distribution of the data. Our understanding is that data that are drawn from the same distribution share the same structure and set of features.

A much more challenging task for NNs than the in-distribution generalization is to achieve OOD generalization, that is, to perform well on data that come from a different distribution than the training data. We expect ODD generalization when OOD data have a similar structure and set of meaningful features as training data. In our case, these OOD data are data samples constructed from eigenstates of the disordered SSH model described by equation (2). Many new features are added to the OOD data that arise from the disorder-induced Anderson localization [80]. This results in more highly localized eigenstates, which render the edge states less distinguishable with increasing disorder strength. Especially dangerous to the OOD generalization are *spurious correlations*, which can be present in the training data. They refer to a situation where some features of the input data and the label appear to be related to each other, but the relationship is coincidental or confounded by an external variable [81].

To evaluate the network OOD generalization, we check its ability to recreate the phase diagram faithfully. A viable error metric is the root mean square error (RMSE). In our case, it is computed as the square root mean of an element-wise square difference between the target (y) and the predicted ($\hat{y}$) winding numbers, for N_s^v values of intracell tunneling v/w, N_s^W values of the disorder amplitude W/w, and N_r realizations of each disorder amplitude W/w. The number of entries in the y and $\hat{y}$ arrays is therefore, $N_\mathrm{el} = N_\mathrm{s}^W \cdot N_\mathrm{s}^v \cdot N_\mathrm{r}$, and the RMSE formula reads

$$\mathrm{RMSE}\,(y,\hat{y}) = \sqrt{\frac{1}{N_\mathrm{el}}\sum_{i=1}^{N_\mathrm{el}}(y_i-\hat{y}_i)^2}\,. \tag{4}$$

Another metric we use is OOD accuracy, defined as the percentage of correct predictions out of N_el in the winding number prediction task. Note that the network can output only binary predictions, however, when we plot phase diagrams predicted by networks there are non-binary entries coming from averaging over N_r realizations of each disorder amplitude W/w.

2.3. CAM

ML methods are usually black-box tools. They accomplish the tasks that we give them at the cost of not being able to justify the outcome they provide. This shortcoming is known as the lack of *interpretability*. Interpretability can be understood as 'the degree to which a human can consistently predict the model's result' [82]. There are numerous interpretability techniques that aim at a better understanding of the network reasoning. We recommend their overview in [83]. In this work, we apply CAM, a simple pixel attribution technique. CAM, given an input sample, produces a map highlighting the areas of the input that contribute the most to the prediction of a considered class.

CAM is an attribution technique tailored for use with CNNs and leverages their design. A CNN produces different representations of the input data during a forward pass through the network and encodes them in different channels. CAM relies on the latent representation of the data present in the output channels of the last convolutional layer in a network, visualized in figure 1(b) as colored activation maps, and performs their weighted sum with weights α_k [84]. The resulting map, after rescaling back to the original data size, highlights the areas that were the most influential in predicting a considered class. The crucial part is the choice of the weights α_k. The original formulation of the method relies on reducing each channel output of the last convolutional layer to a single number with global average pooling (GAP) layer [70]. The weights of

fully-connected layer connecting those numbers with the output corresponding to the considered class C are then used as α_k for the weighted average of convolved images (see appendix F for more details). An obvious limitation of this approach is the necessity for the NN architecture to include the GAP layer, which can limit the applicability of CAM to existing trained networks.

This technique can be made architecture-agnostic [85–90]. The simplest extension is Grad-CAM [86], which replaces the need for the GAP layer with gradients computed by backpropagation of the class output to the last convolutional layer. We applied both techniques in our analysis, but they gave quantitatively the same results, which is why in the discussion we only present the results obtained with CAM.

2.4. Dimensionality reduction

The information learned by an NN is stored throughout all its parameters. Analyzing the data representation at different layers, i.e. how different input samples activate various network's neurons, can help the researchers extract this knowledge [91–94]. However, even for small NNs, the dimensionality of the activation space is on the order of hundreds and more, and humans have difficulty comprehending data in high dimensions. Thus, reducing data to a small number of dimensions is helpful for visualization purposes and allows us to gain valuable information on the inner workings of NNs. An example of such insight might be to extract which areas in latent space correspond to given learned concepts and, conversely, the relations and distance between them that the network has learned [95, 96]. Dimensionality reduction of the data representation is also used in a cluster-based interpretation technique [97] that aims to explain cluster assignments within layers in terms of input features. To visualize and analyze data structure across NN layers, we use PCA - an established linear dimensionality reduction technique.

PCA [98] relies on computing the principal components (PCs) of the data that are first stacked to form a multidimensional tensor. The PCs are eigenvectors of the covariance matrix of this tensor, represented schematically as orange arrows in figure 1(c). The magnitude of their corresponding eigenvalues orders them, so the first PC represents the direction of the highest data variance, the second PC is orthogonal to it and describes the direction with the second highest variance, etc. The original tensor is then projected into a subspace spanned by the selected number of its PCs. The result is a tangible low-dimensional representation of the data that presents the maximal variance of the original high-dimensional space. The low-dimensional representation is needed as a preprocessing step for some ML algorithms or visualization purposes. Because of the linearity of this reduction technique, the distance between the points and the density is primarily preserved. It can be treated as representative of the distance in the original high-dimensional space.

3. Results and discussion

3.1. CNNs fail to generalize to data with disorder

CNNs succeeded in the training task posed by the disorderless SSH system. The training of all CNN instances converged, and the networks achieved perfect (100%) accuracy on both the training and validation data sets. The performance was also very good for all the trained instances when tasked with making predictions for in-distribution data. The accuracy score for the in-distribution testing was greater than 95%. The only regions where some test samples were misclassified were for data in the vicinity of the phase transition, which were excluded from the training set.

Despite promising results within the training distribution, OOD generalization proved to be a challenging task, as we show in table 1. We quantify the OOD generalization by testing the trained networks on the phase diagram of the SSH model, generated for 25 disorder realizations, and presented in the bottom-left corner of figure 2. Upon many initializations, we have found that only as little as 5% of them manage to reproduce the shape of the phase diagram correctly. We used RMSE as a metric of distinction between the well- and poorly-generalizing networks, with the threshold set at 0.2. We divide these networks into two such groups and report their OOD accuracy. The well-generalizing networks achieve an OOD accuracy of 84% and more on the data from the whole phase diagram. In contrast, a large majority (95%) of CNNs have OOD accuracy only marginally higher than a random guess, as seen in the third row of table 1. Moreover, this failure in the OOD generalization seems to be tied to convolutional layers as fully-connected networks do not exhibit this behavior (see appendix D for details). Interestingly, well-generalizing CNNs tend to predict the phase transition at values below $v/w = 1$, closer to $v/w \approx 0.9 - 0.95$, as visible in figures 2(h) and (j). They may take into account the finite-size effects that the infinite-system approximation of the winding number fails to account for.

We can improve this poor statistic by introducing slightly disordered ($W/w \in \{0.01, 0.05\}$) data into the training data set. We treat this range of small disorders as a 'perturbative' regime in which we can label the slightly disordered data with winding numbers calculated for respective disorderless points of the phase diagram. This is a simplistic attempt at the domain adaptation, where we use 'unlabeled' data from test

Table 1. Statistics (means ± standard deviations) of 100 randomly initialized networks trained only on data without disorder, $W = 0$, and on data including also slightly disordered samples, $W/w \in \{0, 0.01, 0.05\}$.

	CNNs trained on $W = 0$		CNNs trained on $W/w \in \{0, 0.01, 0.05\}$	
OOD generalization	Well-generalizing	Poorly-generalizing	Well-generalizing	Poorly-generalizing
Number of CNNs	5 / 100	95 / 100	22 / 100	78 / 100
Training accuracy	∼100 %	∼100 %	∼100 %	∼100 %
Test accuracy	95.5 ± 1.5 %	97 ± 2 %	94.7 ± 1.9 %	97 ± 2 %
OOD accuracy	84 ± 3 %	55 ± 8 %	86 ± 3 %	61 ± 9 %
RMSE	0.168 ± 0.025	0.46 ± 0.07	0.153 ± 0.031	0.41 ± 0.10

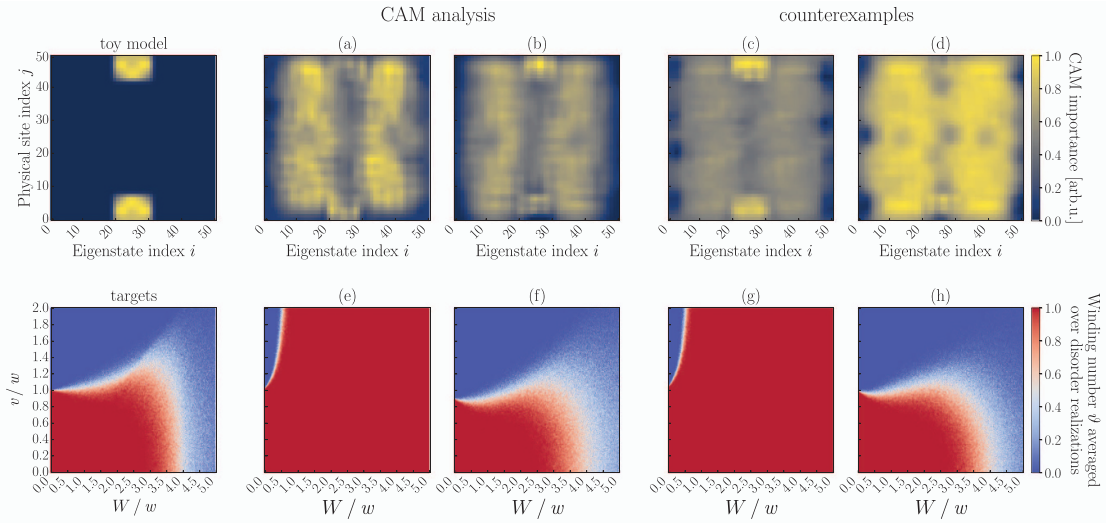

Figure 2. Comparison between a well- and poorly-generalizing CNN through the lens of CAM. First column: the expected CAM result for a CNN that focuses on the edge states, obtained via a toy model in appendix G (first row), and the target phase diagram (second row). (a) CAM-based explanation of the prediction of one of many poorly-generalizing CNNs (here, the OOD accuracy of 50.3%) for an input sample from the disorderless SSH model at $v/w = 0.27$. CNN places more importance on states around the one- and three-fourths of the spectrum. (b) One of the well-generalizing CNNs (OOD accuracy of 90.1%) in the same input sample pays more attention to the localization of the edge states. However, (c) poorly-generalizing CNN (OOD accuracy of 52.3%) can also pay attention to edge states, and (d) well-generalizing CNN (OOD accuracy of 91.7%) can ignore edge states, rendering the CAM analysis inconclusive. Panels (e)–(h) present the predicted phase diagram of the corresponding CNNs. The phase diagrams are predicted winding numbers ϑ averaged over $N_r = 25$ realizations of disorder, with $N_s^v = 200$ slices in v/w, and $N_s^W = 500$ slices in W/w directions. The CNNs selected for analysis were the ones trained on data with no and slight disorder, $W/w \in \{0, 0.01, 0.05\}$.

distribution to improve the network performance between domains. As a result, the ratio of successful CNNs increases from 5% to 22%. Other statistics remain qualitatively the same in both well- and poorly-generating models. A performance comparison of 100 randomly initialized CNNs trained only on disorderless data and CNNs trained also with slightly disordered data is in table 1. There are better domain-adaptation techniques that could help in this setup. We have already mentioned a domain adversarial NN used in [34]. In appendix E, we test the correlation alignment (CORAL) domain adaptation method and, interestingly, while CORAL improves CNN performance on disordered data, it does so at the cost of lower performance in the disorderless regime.

The failure of the trained CNNs should be contrasted with other works in which networks were trained in a simple regime of a problem and successfully generalized out of the training distribution. For example, [99] demonstrated that an NN trained in a supervised way with the noninteracting data accurately predicted all topological phases in the interacting system for a variety of topological insulators in different dimensions and symmetry classes. The input data in this work was the curvature function's values at high-symmetry points in either momentum or momentum-frequency space. A follow-up work [100] showed that a similar feat is possible when a network is provided with single-particle correlation functions in a noninteracting regime. We hypothesize that generalizing from no disorder to disordered data is more challenging for deep networks than from noninteracting to interacting data because the disorder is more likely to trigger the sensitivity of NNs to adversarial perturbations [101]. On the other hand, [102] presented that when a network is provided with the entanglement spectrum, it can successfully generalize from weak to strong

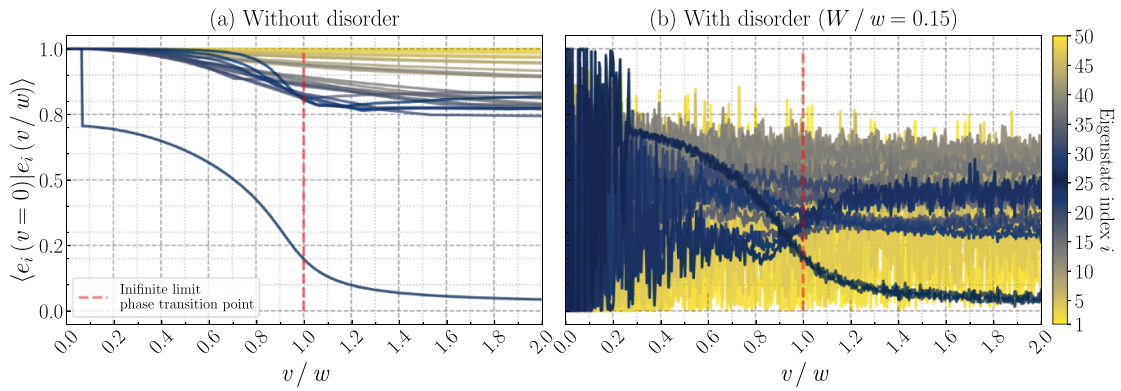

Figure 3. Fidelity between the eigenstate e_i for $v = 0$ and the respective eigenstate e_i for different values v/w, showcasing an example of correlations in data coming from the SSH model that can be used by an NN to predict a label in a simple regime. (a) Apart from the rapidly changing fidelity expected of the half-filled edge state, we can also observe the changing fidelity of eigenstates closer to the extremes of the spectrum. (b) Already in the weakly disordered case ($W/w = 0.15$) fidelity of the states closer to the extremes of the spectrum becomes a poor predictor of the phases. Only the disorder-robust fidelity of edge states remains a valid feature. Note: The drop in fidelity of edge states around $v/w = 0.08$ in panel (a) and rapid oscillations for small v/w in panel (b) come from the hybrydization of the edge states due to finite-size effects [72].

disorder in the task of classifying topological phases. Their result highlights the advantage of choosing a good representation of data that allows for better network performance.

To sum up, the networks fail to generalize to the disordered data despite showing stellar performance both in the training regime and during the in-domain testing. Most CNNs fail even though we provide them with all the necessary information to carry this performance to the disordered regime, such as the disorder-robust connection between zero-energy edge states and winding number. Moreover, the physically motivated remedy only slightly improved the number of well-generalizing networks, while CORAL did it at the expense of worse performance on disorderless data. To understand these phenomena, we must lift the lid of the 'black box' and ask why CNNs do not generalize well.

3.2. Analysis of the CNNs generalization's failure

Most successfully trained CNNs failed to generalize from the simple training regime to the disordered regime. This is especially surprising because there is a robust indicator of the topological phase, namely the presence of edge states. To understand what other features of the input data are instead leveraged by networks to make their predictions, we use CAM [70], described in section 2.3.

A large majority of trained networks tend to ignore edge states in the middle of the spectrum as well as the extremes of the spectrum, that is, the ground state and the highest excited state (maximally filled state). Instead, they look at the remaining states of the system. We see an example of such a typical network in figure 2(a). Networks that ignore edge states in the disorderless SSH model usually fail to generalize to the system with the disorder, as seen in the corresponding predicted phase diagram in figure 2(e). In contrast, a small number of networks focus more on the edge states in the middle of the spectrum, corresponding to the half-filled system and constituting the topological invariant of the system. An example of such a network is shown in figure 2(b), which pays closer attention to the localization at one of the system edges. This group of networks is more likely to generalize well to the disordered data, see the predicted phase diagram in figure 2(f), as it detects something related to the known topological invariant. This analysis shows that we can increase our trust in the OOD generalization of the network by making sure it looks at relevant features in the known regime of the problem.

The analysis so far showed that networks tend to ignore edge states, but why is this the case? We hypothesize that features must exist in the bulk that allow a network to solve the learning task in the simple regime, i.e. that correlate well with the label in the simple regime. Prompted by this analysis, we compare each eigenstate of the SSH model for intracell tunneling $v = 0$ to its respective eigenstate for different intracell tunneling values. We plot the described fidelity of the pair of eigenstates in figure 3(a). As expected, we see that the fidelity of the edge states at $v = 0$ and their counterparts for different v/w changes rapidly throughout the phase diagram. However, we observe that the fidelity of the bulk states also changes across v/w and, as such, can be used by a CNN to predict a label in the system without disorder. It is only an example of a possible feature, and there can be any number of more complex ones that the networks pick up from the data. Importantly, they no longer correlate with the label in the disordered system, as seen in

figure 3(b). For this reason, we consider them to be *spurious correlations* in the task of distinguishing between the topological and topologically trivial phases.

Apparently, in this task, CNNs tend to learn some combination of features related to numerous non-edge states, even if they are weakly correlated with the label when taken separately. This observation echoes the results of the ML community on the inherent trade-off of classifiers between the accuracy and robustness. Tsipras *et al* [103] showed rigorously on a simple example that classifiers learn a combination of weakly correlated features to achieve perfect accuracy, instead of relying on a single strongly correlated feature that is also present in the data but does not allow for perfect accuracy. When disorder enters the data, it can easily distort the combination of weakly correlated features, harming robustness of a classifier. A solution to achieving robust NNs is adversarial training [101], which is reminiscent of the original solution from [34], which used a domain adversarial NN. In our data distribution, though, the single feature (edge states) should be perfectly correlated with the label (up to finite-size effects), so the task of maximizing accuracy should not lead to ignoring this feature. We do not know what causes CNNs to favor the combination of weak correlators over the single predictive feature, but we have two hypotheses. First, the preference towards multiple features may be due to regularization that prevents NNs from relying on single features. Second, it may come from the iterative nature of the training, which causes multiple features with weak correlation to have an increasingly strong training signal as they boost each other.

3.3. Unsupervised analysis of OOD generalizations of NNs

In the previous section, we have described how CAM guided us into understanding that the majority of CNNs learn some spurious combination of weakly correlated features (related to non-edge states) instead of a single strong feature (related to edge states). An appropriate combination of weakly correlated features allows for perfect accuracy in the simple regime but ceases to correlate with the label in the disordered regime. The same CAM analysis can be used to assess the OOD generalization of CNNs without access to the labels. For example, if a CNN focuses on the edge state, this increases the chance of a better OOD generalization of the model.

Although, overall, CAM is useful and can be used as the first step of the generalization study, we have found that the CAM analysis is unfortunately inconclusive and can only indicate CNN OOD generalization tendencies. In particular, not every CNN that pays attention to the edge states exhibits good OOD generalization—evidence for this is in panels (c) and (g) of figure 2. We even find an opposite example where the CNN that does not put special focus on edge states achieves great OOD accuracy, as seen in panels (d) and (h) of figure 2. We share the CAM results for all 100 trained CNNs for all test data online [104] for the reader interested in various possible behaviors of the discussed CNNs. The reliability of CAM as the method to assess the generalization could be improved if used on the test data from the disordered SSH model. However, this would bring this explanation method into a noisy regime where CAM is known to be fragile and unreliable. We advise the reader against using this explanation method in the presence of noise and elaborate on this topic in appendix F.

As a second technique to validate the OOD generalization of a network, we propose to study the structure of data representation learned by trained CNNs. When an NN processes an input sample, it generates different activations at every network layer. We want to understand the relation between the activations generated by data without the disorder and those generated by the disordered data. We follow here the logic that the data the network views as similar should generate similar activations or, in other words, should have similar learned representations [95]. If an NN sees any meaningful relation between the training data without the disorder and the OOD test data with the disorder, their representations should be somewhat similar. Such a result increases our trust in the OOD prediction made by a network.

The space of CNN activations is high-dimensional. In the case of a fully connected layer, the dimension depends on the number of neurons. In the case of a convolutional layer, the dimension is equal to the size of the convolved data point. To study the relation between points in such a high-dimensional space, we apply dimensionality reduction techniques that can bring the space dimension down to, e.g. two while preserving some structure of the original data. We use here PCA, explained in section 2.4. To verify the results, we also used uniform manifold approximation and projection (UMAP), which is a non-linear dimensionality reduction technique. We discuss obtained results in appendix H.

We study the representation of the data learned by the penultimate layer of CNN, that is, the neuron activations after applying the GAP and before entering the last fully connected layer, with the softmax function, which plays the role of a classifier. To this end, in figure 4 we present the clustering obtained with PCA for data with different disorder strengths in the two-dimensional versions of the high-dimensional representation space of the penultimate CNN layer. The dimensionality reduction is applied to the

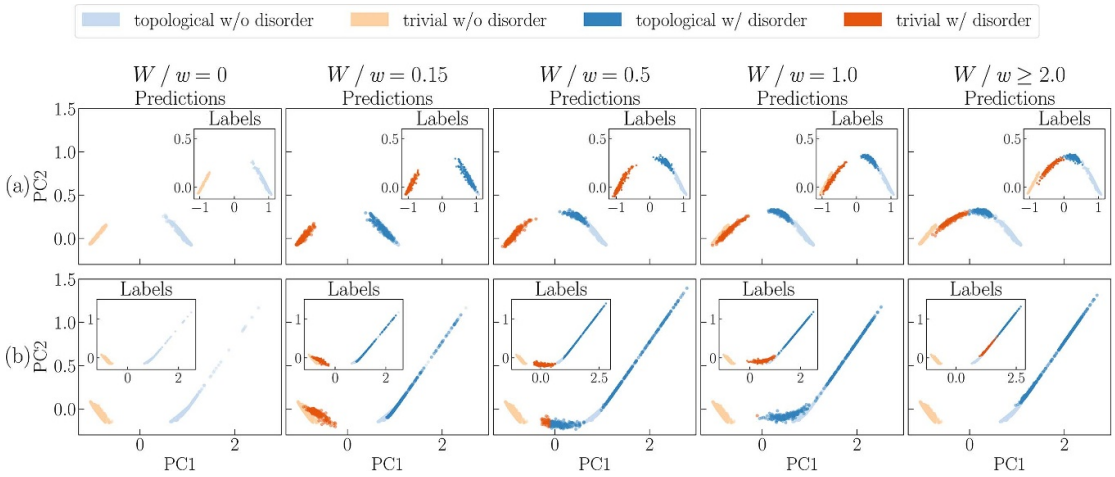

Figure 4. Two-dimensional representation of the data learned by the penultimate layer of a poorly- and well-generalizing CNN with inconclusive CAM results from figure 2, obtained with PCA. (a) The well-generalizing CNN, which according to CAM ignores the edge states, learned the representation in which the disorderless data (first column) are clustered in two well-separated clusters corresponding to two phases. Data with increasing disorder falls firstly in the same two clusters formed by disorderless data, and then slowly separates from them but keeps a weak connection. (b) The representation of test data learned by the poorly-generalizing CNN, which according to CAM pays close attention to the edge states, already for disorderless data (first column) lacks well-defined two clusters. The data with disorder, forms more and more separated third cluster, only weakly connected to one of the clusters of disorderless data. Insets represent labels obtained with the infinite-system approximation of the winding number. The first two PCs explain together 100% of the data variance.

activations generated by all test data simultaneously, but we plot data subsets for different W/w's in separate subplots to make the analysis clearer. We exclude test data without disorder coming from the vicinity of the transition ($v/w \in [0.9; 1.1]$) to further simplify the analysis.

Let us first focus on the data representation learned by the penultimate layer of a well-generalizing CNN in figure 4(a). The first thing to notice is the representation of the disorderless data (first column) that relies on two well-separated clusters corresponding to two phases (light orange is trivial, light blue is topological). In the next columns, we plot both the disorderless data and the data with increasing disorder strength (indicated with darker orange and blue). For small and intermediate disorder ($W/w = 0.15$ and 0.5), the disordered data are represented very similarly to the data without disorder, in other words, their representation overlaps. With increasing disorder ($W/w = 2$ and more), the data start to disconnect from the original clusters. Interestingly, the topological data with disorder separate faster from the disorderless data than the data from the topologically trivial phase. Finally, only for large disorder the data form a separate third cluster. This means that the network has a disjointed representation only of strongly disordered data.

In contrast, the data representation learned by poorly-generalizing CNNs already at the level of disorderless data do not form two well-defined clusters corresponding to two phases. Especially the topological data without disorder form quite a disconnected cluster, as seen with PCA in figure 4(b). The data with disorder of $W/w = 0.5$ (third column) is already separate from the original clusters. For strong disorders, data form three or four separate clusters in the representation space of the penultimate layer. This suggests that the network does not see continuity in the data as a function of the disorder strength.

Notably, the two CNNs whose learned data representations we studied above are the same CNNs whose CAM analysis rendered inconclusive results. That is, the well-generalizing CNN is the one from figures 2(d) and (h) that focuses less on edge states. The poorly-generalizing CNN is the one from figures 2(c) and (g) that focuses prominently on edge states. This shows that data representation analysis together with CAM can make stronger statements about the OOD generalization of trained networks.

Finally, we make an analogous analysis for data representation across layers of the well- and poorly-generalizing CNNs in appendix H, this time with UMAP. It renders the same conclusions as PCA. However, we make an additional interesting observation that UMAP performed on the input data with and without disorder does a better job in forming clusters corresponding to topological and trivial phases than a majority of CNNs trained in a supervised way on data without disorder. The success of UMAP, in light of CNN's failures, highlights the power of unsupervised data analysis and the development of a useful data representation as the first step of an ML pipeline.

4. Conclusion and outlook

In this work, we have presented how to increase the trust in predictions of an NN when the predictions are made on data from a different distribution than the NN's training data distribution and without access to ground-truth labels. Such an OOD generalization is a desired property of a robust and reliable network and is difficult to achieve or validate. We have shown how one can study the network to understand the way it processes the data with two different tools. The first is an interpretability technique that highlights parts of the data that were important to the network classification decision, called CAM. The second is an analysis of the data representation learned by the network, facilitated by the dimensionality reduction techniques, such as PCA, applied to the representation space of the network. The data we have tackled are eigenstates of the SSH Hamiltonian, coming from two regimes: with and without disorder. We trained hundreds of CNNs on the data without disorder and checked their in-distribution and OOD generalization to data without and with disorder, respectively. We have made the following observations.

- An overwhelming majority (95%) of successfully trained CNNs failed to generalize to data with disorder. This behavior seems to be tied to convolutional layers because it is shared across various CNN architectures and ResNets, which contain convolutional layers. Expanding the training set with data with slight disorder ($W/w \leqslant 0.05$) decreased the number of poorly-generalizing CNNs only slightly (78%). While the CORAL domain adaptation method significantly improves CNN performance on disordered data, it does so at the cost of lower performance in the disorderless regime.
- The reason CNNs tend to fail is that they focus on non-generalizable features of data that are nevertheless useful for the training task, as indicated by the CAM analysis of CNNs' predictions. For example, eigenvectors other than the edge states carry correlations that can be leveraged to classify phases in the disorderless data regime, but they disappear when disorder is added.
- CAM analysis can be used to assert our trust in network prediction. If in the disorderless regime a network pays attention to edge states, which are known to be useful features, such a network tends to exhibit a better OOD generalization. Surprisingly, such networks can still fail to generalize to disordered data, making the CAM analysis inconclusive.
- Moreover, CAM itself is also a fragile interpretation technique that performs poorly in the presence of noise, which prevented its use to understand the network predictions in the disordered regime.
- Dimensionality reduction techniques such as PCA can be used to visualize the data representation learned by networks. If a CNN has a disconnected (very different) representation of disordered data with no connection to the disorderless training data, the trust in its predictions on the disordered data should be limited. A good predictor of the OOD generalization quality is instead when a CNN represents the data with slight disorder similarly to its disorderless training data and then the representation gets smoothly disconnected with increasing disorder strength.

We conclude that, together, CAM and data representation analysis serve as useful tools to gain additional insight and assess trust in NN predictions. Given their low cost, we believe that their routine use can only benefit members of the scientific community who have already added deep learning to their computational toolbox. The described predictors of the good OOD generalization can, in principle, guide an NN architecture selection by rejecting architectures that do not exhibit expected positive behaviors, such as a similar representation of data without and with a small disorder. Such an analysis of NNs is especially needed in light of our surprising observations that CNNs fail in a relatively simple task that is generalizing to slight disorder when trained on the disorderless SSH model, whereas physicists know that there exists a robust feature to solve this task, such as the presence of edge states. This observation also highlights that scientific data sets create new challenges for deep learning [105, 106] which can unravel unexpected behaviors and failure modes of NNs. An interesting development would be to understand why deep networks prefer many features that are weakly correlated with the label over a single strongly correlated feature, even when focusing on a single feature does not come at the cost of accuracy as in our work, in contrast to the setting studied in [103]. Another interesting direction would be to study the OOD generalization of simple transformers in this setting, as their attention layer would remain interpretable also in the disordered regime, contrary to CAM, possibly containing much better predictors for good OOD generalization in the absence of labels. At the same time, the community needs to develop more robust tools to assess NN performance in regimes without known ground truths, bearing in mind that some of them fail in the presence of random or adversarial noise.

Data availability statement

The source code and data that support the findings of this study is openly available at https://doi.org/10.5281/zenodo.12518289 [104].

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.12518289.

Acknowledgments

K C acknowledges the financial support from the Polish Ministry of Science and Higher Education within the 'Excellence initiative—research university' program. M T acknowledge the National Science Centre Poland (Grant No. 2020/38/E/ST2/00564) for the financial support and the Poland's high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support (computational Grant No. PLG/2023/016878). ICFO group acknowledges support from European Research Council AdG NOQIA; MCIN/AEI (PGC2018-0910.13039 /501100011033, CEX2019-000910-S/10.13039/501100011033, Plan National FIDEUA PID2019-106901GB-I00, Plan National STAMEENA PID2022-139099NB, I00, Project funded by MCIN/AEI/10.13039/501100011033 and by the 'European Union NextGenerationEU/PRTR' (PRTR-C17.I1), FPI); QUANTERA MAQS PCI2019-111828-2; QUANTERA DYNAMITE PCI2022-132919, QuantERA II Programme co-funded by European Union's Horizon 2020 program under Grant Agreement No 101017733; Ministry for Digital Transformation and of Civil Service of the Spanish Government through the QUANTUM ENIA project call—Quantum Spain project, and by the European Union through the Recovery, Transformation and Resilience Plan—NextGenerationEU within the framework of the Digital Spain 2026 Agenda; Fundació Cellex; Fundació Mir-Puig; Generalitat de Catalunya (European Social Fund FEDER and CERCA program, AGAUR Grant No. 2021 SGR 01452, QuantumCAT U16-011424, co-funded by ERDF Operational Program of Catalonia 2014-2020); Barcelona Supercomputing Center MareNostrum (FI-2023-3-0024); Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, European Commission, European Climate, Infrastructure and Environment Executive Agency (CINEA), or any other granting authority. Neither the European Union nor any granting authority can be held responsible for them (HORIZON-CL4-2022-QUANTUM-02-SGA PASQuanS2.1, 101113690, EU Horizon 2020 FET-OPEN OPTOlogic, Grant No 899794); This project has received funding from the European Union's Horizon Europe research and innovation program under Grant Agreement No 101080086 NeQSTGrant Agreement 101080086 - NeQST; ICFO Internal 'QuantumGaudi' project; European Union's Horizon 2020 program under the Marie Sklodowska-Curie Grant Agreement No 847648; 'La Caixa' Junior Leaders fellowships, 'La Caixa' Foundation (ID 100010434): CF/BQ/PR23/119 80043. Al D acknowledges the financial support from a fellowship granted by la Caixa Foundation (ID 100010434, fellowship code LCF/BQ/PR20/11770012). An.D. acknowledges the financial support from the National Science Centre Poland (Grant No. 2020/36/T/ST2/00588) and from the Foundation for Polish Science. The Flatiron Institute is a division of the Simons Foundation.

Appendix A. Alternate SSH model formulations

The basic formulations of the SSH model [107] we consider is discussed by Asboth *et al* [72]:

$$\hat{H} = v\sum_{m=1}^{N} |m\rangle\langle m| \otimes \hat{\sigma}_x + w\sum_{m=1}^{N-1}\left(|m+1\rangle\langle m| \otimes \frac{\hat{\sigma}_x + i\hat{\sigma}_y}{2} + \text{h.c.}\right). \tag{A1}$$

In $\hat{c}_i^{(\dagger)} = \left(\hat{c}_{A,i}^{(\dagger)}, \hat{c}_{B,i}^{(\dagger)}\right)$ operators formulation it is,

$$\hat{H} = v\sum_{n=1}^{N} \hat{c}_n^{\dagger}\hat{\sigma}_x\hat{c}_n + w\sum_{n=1}^{N-1}\left(\hat{c}_n^{\dagger}\frac{\hat{\sigma}_x + i\hat{\sigma}_y}{2}\hat{c}_{n+1} + \text{h.c.}\right). \tag{A2}$$

Other works we used for comparison of our results formulate it differently:

1. Mondragon *et al* [73]:

$$\hat{H}_{\text{SSH}} = \sum_n \left\{ m_n\hat{c}_n^{\dagger}\hat{\sigma}_y\hat{c}_n + t_n\left(\hat{c}_n^{\dagger}\frac{\hat{\sigma}_x + i\hat{\sigma}_y}{2}\hat{c}_{n+1} + \text{h.c.}\right)\right\}, \tag{A3}$$

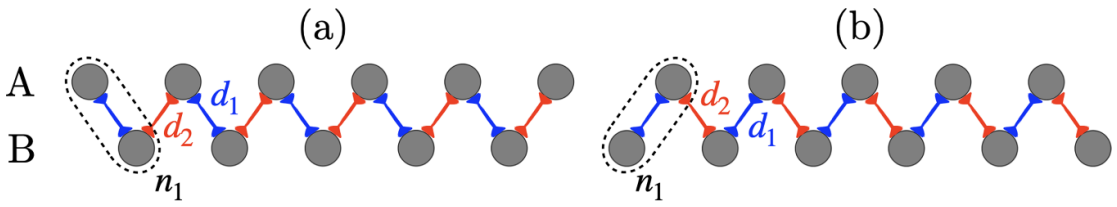

Figure 5. Difference between SSH formulations from [73] and [74] is due to different sublattice order, as seen in panels (a) and (b), respectively. Both define distance within unit cell $d_1 = 0$, and distance between unit cells as $d_2 = 1$. Clarification of this choice is needed for the further transition to momentum space, as the ordering of A and B sites in each unit cell is swapped.

2. Meier *et al* [74]:

$$\hat{H}_{\mathrm{SSH}} = \sum_n \left\{ m_n \hat{c}_n^\dagger \hat{\sigma}_x \hat{c}_n + t_n \left(\hat{c}_{n+1}^\dagger \frac{\hat{\sigma}_x - i\hat{\sigma}_y}{2} \hat{c}_n + \mathrm{h.c.} \right) \right\}, \tag{A4}$$

3. Le *et al* [75]:

$$\hat{H}_{\mathrm{SSH}} = \sum_{j,\sigma=A,B} \left\{ -J \left[1 + (-1)^j \Delta t \right] \hat{c}_{j+1,\sigma}^\dagger \hat{c}_{j,\sigma} + \mathrm{h.c.} \right\}. \tag{A5}$$

Equations drawn from [73] and [74], allow for the presence of disorder in the system through t_n and m_n tunneling amplitudes, corresponding to the same *inter-* or *intra-* cell tunneling as w and v. The correspondence is:

- Intercell tunneling $w \to t_n$,
- Intracell tunneling $v \to m_n$.

The sign discrepancy between models from [73] and [74] in the intercell part is due to the different site numeration. The former uses $BABABA\ldots$ convention, and the latter uses $ABABAB\ldots$ convention, which is the one we use. The difference is seen in figure 5. Another difference we note is that in [73], the intracell tunneling amplitudes are purely imaginary m_n, while in [74], they are purely real m_n. Both yield the same results as long as one convention is kept. In this work, we opt for real tunneling amplitudes.

Appendix B. Calculation of winding number

Calculation demonstration. The routine we employ for the calculation of the winding number [76] involves several steps. This section is to serve as a step-by-step demonstration. It is accompanied by a mirror Jupyter Notebook in our GitHub repository [104].

Demonstration parameters. In order to keep the calculations tractable, we choose system size to be 6 sites (3 unit cells). We set the boundary conditions to periodic. We generate the occupational basis, and sort it lexicographically.

Winding number formula. We want to use here the equation for the winding number in the form [76]:

$$\vartheta = \frac{1}{2\pi i} \oint_{\mathrm{BZ}} \mathrm{Tr}\left(h^{-1} \partial_k h \right). \tag{B1}$$

It is defined as a 1st Brillouin zone integral of a transformed momentum space Hamiltonian. In the following sections we will derive it, starting from Hamiltonian definition presented in equation (1).

Occupational basis Hamiltonian. First we define the Hamiltonian in the occupational basis according to equation (1). In matrix form it is,

$$H_{\mathrm{occ}} = \begin{bmatrix} 0 & v & 0 & 0 & 0 & w \\ v & 0 & w & 0 & 0 & 0 \\ 0 & w & 0 & v & 0 & 0 \\ 0 & 0 & v & 0 & w & 0 \\ 0 & 0 & 0 & w & 0 & v \\ w & 0 & 0 & 0 & v & 0 \end{bmatrix}. \tag{B2}$$

Momentum basis Hamiltonian. The winding number is defined as a 1st Brillouin zone integral, so we must transfer the Hamiltonian to momentum space. The condition to be met here is that a fermion only accumulates phase when crossing between the neighboring Brillouin zones. Given our choice of boundary conditions, this is achieved by addition of e^{ik} multiplication to the corner terms. These are the terms corresponding to transitioning the periodic boundary. The resulting Hamiltonian is

$$H^{\mathrm{k}}_{\mathrm{occ}} = \begin{bmatrix} 0 & v & 0 & 0 & 0 & we^{ik} \\ v & 0 & w & 0 & 0 & 0 \\ 0 & w & 0 & v & 0 & 0 \\ 0 & 0 & v & 0 & w & 0 \\ 0 & 0 & 0 & w & 0 & v \\ we^{ik} & 0 & 0 & 0 & v & 0 \end{bmatrix}. \tag{B3}$$

Chiral symmetry operator. The next step is to define the chiral symmetry operator Γ. In this system it has the form

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \tag{B4}$$

Hamiltonian in chiral symmetry basis. We need to rewrite our Hamiltonian in the eigenbasis of Γ. To this end, we must ensure the eigenvectors of Γ are also sorted lexicographically. The resulting change-of-basis matrices $\chi, \chi^\dagger$ have the form

$$\chi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad \chi^\dagger = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{B5}$$

Once we apply them to the Hamiltonian H it becomes block off-diagonal,

$$H = \chi H^{\mathrm{k}}_{\mathrm{occ}} \chi^\dagger = \begin{bmatrix} 0 & 0 & 0 & v & 0 & we^{-ik} \\ 0 & 0 & 0 & w & v & 0 \\ 0 & 0 & 0 & 0 & w & v \\ v & w & 0 & 0 & 0 & 0 \\ 0 & v & w & 0 & 0 & 0 \\ we^{ik} & 0 & v & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & h^\dagger \\ h & 0 \end{bmatrix}. \tag{B6}$$

Final calculation. Now the elements we need for the calculations according to equation (B1) have the form:

$$h = \begin{bmatrix} v & w & 0 \\ 0 & v & w \\ we^{ik} & 0 & v \end{bmatrix}, \quad h^{-1} = \frac{1}{v^3 + w^3 e^{ik}} \begin{bmatrix} v^2 & -vw & w^2 \\ w^2 e^{ik} & v^2 & -vw \\ -vwe^{ik} & w^2 e^{ik} & v^2 \end{bmatrix}, \quad \partial_k h = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ iwe^{ik} & 0 & 0 \end{bmatrix},$$

$$h^{-1}\partial_k h = \frac{ie^{ik}}{v^3 + w^3 e^{ik}} \begin{bmatrix} w^2 & 0 & 0 \\ -vw & 0 & 0 \\ v^2 & 0 & 0 \end{bmatrix}, \quad \mathrm{tr}\left[h^{-1}\partial_k h\right] = \frac{iw^2 e^{ik}}{v^3 + w^3 e^{ik}}.$$

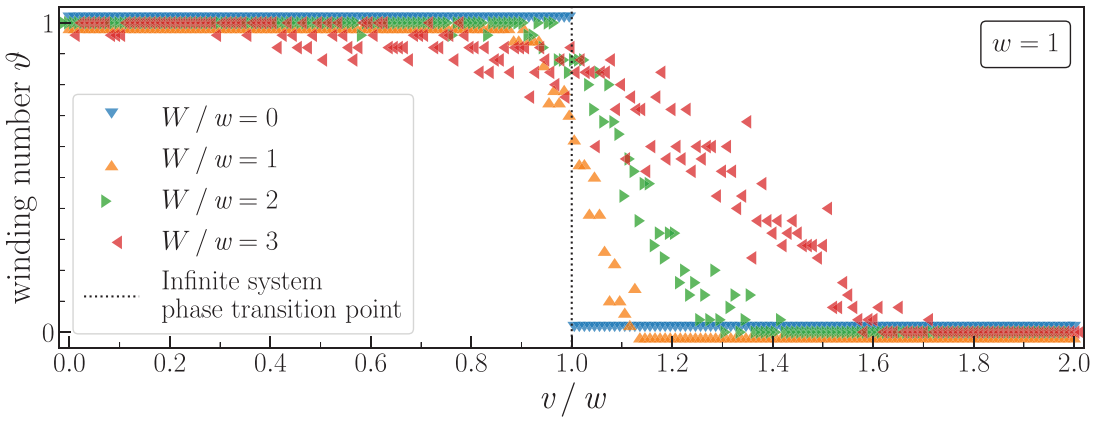

Figure 6. Phase transition vs disorder strength. Comparison of winding number computed for different disorder amplitudes W/w, averaged over $N_r = 25$ realizations of disorder. With increased disorder, the phase transition point shifts to $v/w \geqslant 1$, as discussed in section 2.1. Use of infinite system calculation is visible by the position of phase transition point, computed to happen at $v/w = 1$ in disorderless system ($W = 0$). In a finite system, it would happen for a smaller v/w ratio [72].

We have arrived at the final formula for integration to obtain the winding number. Now, it needs to be integrated over the 1st Brillouin Zone. Due to software stability reasons, we integrated it numerically for $k \in [0, 2\pi - \epsilon]$.

Universality. This calculation of the winding number also applies to the disordered SSH Hamiltonian, defined by equation (2). The phase transition point then drifts to values of $v/w \geqslant 1$ with an increase of disorder amplitude W/w. This is presented in figure 6.

Appendix C. Architectures and hyperparameters

Data point. Single data point supplied to the network is a matrix of squared coefficients of eigenstates of Hamiltonians defined by equation (1) (represented in a lexicographically sorted occupational basis), (2) and its corresponding label. Each column is a single normalized eigenstate. The label is the winding number. A step-by-step demonstration of the calculation of the winding number is presented in appendix B.

Basic training parameters. All throughout this work we keep the intercell tunneling amplitude $w = 1$. The two phases of the SSH system are probed by varying $v/w \in (0, 2)$. In all data sets (training, validation, test) both phases are equally numerous – 50% of all data points represent each phase. To ensure linear independence of data in all three sets, the intervals of v/w were chosen with a slightly offset initial point. That is $0.001, 0.002, 0.003$ for training, validation, and test set, respectively. This ensures they all represent distinct points in the phase diagram, and that in disorderless setting they belong to the same distribution.

Data set composition. A part of what we would like to achieve is for the network to learn the correct phase transition point by itself. Therefore, the training and validation data sets do not contain data from the direct vicinity of the phase transition. In these two sets, we vary the parameter v in the range $v/w \in (0, 0.8) \cup (1.2, 2)$. The composition of disorderless data sets is presented in the left column of table 2.

Training extension—disordered data. The extension that proved fruitful was an addition of disordered data to the training. In this setting, we included data from low-disorder regimes ($W/w \in 0.01, 0.05$) in training and validation sets. For each disorder amplitude, the data added to the training set remained in 5 : 1 ratio to the disorderless data. For the validation set, the ratio was set at 4 : 1. The composition of data sets used in both regimes of training are presented in table 2.

Generalization assessment. To assess the networks' generalization, we tasked it with recreating the target v vs W phase diagram. To this end, we generated $N_r = 25$ disorder realizations of the input data and corresponding labels. We covered the range $v/w \in [0, 2]$, and $W/w \in [0, 5]$ for each realization. The step size in disorder amplitude strength is $\Delta W/w = 0.1$, while the step size in intracell tunneling amplitude is $\Delta v/w = 0.01$. After collecting predictions for all N_r realizations, the values in the generated phase diagram are then averaged over N_r predicted labels for each pair of $(v/w, W/w)$.

Table 2. Composition of data sets used in networks' training. The base setting involved only using data coming from the SSH model in equation (1). In an extension to this approach, data from disordered SSH model was added, see equation (2).

Approach	Disorderless	Disordered data		
Disorder	$W=0$	$W=0$	$W/w=0.01$	$W/w=0.05$
Training points	5000	5000	1000	1000
Validation points	1000	1000	250	250

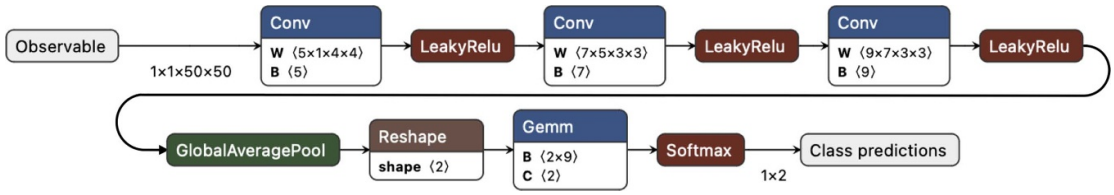

Figure 7. Architecture of the CNN used in this work.

Architecture. The architecture we use is a shallow CNN, as presented in figure 7. It consists of three convolutional layers with symmetric kernels. The initial convolution filter is of shape (4, 4), followed by convolutions with (3, 3) kernels. Before the final fully-connected classification layer, we position the GAP layer in order to accommodate the requirements of the CAM attribution technique.

Hyperparameters. We trained the networks using Stochastic Gradient Descent (SGD). We explored various combinations of learning rate (LR), momentum, weight decay, and batch size. The range of LRs we evaluated was $10^{-6} - 10^{0}$. The tested batch sizes were 64 and 500. The hyperparameters we picked for training were: LR $= 10^{-4}$, momentum $= 0.1$, weight decay $= 0.1$, batch size $= 64$. The networks were trained for up to 500 epochs, with early stopping allowed after 10 consecutive epochs of validation loss rise, with warm-up of 50 epochs.

Appendix D. Comparison with other architectures

Aim of the comparison. In section 3.1, we demonstrate that that CNNs fail in a relatively simple task, with a known solution, that is generalizing to slight disorder when trained on the disorderless SSH model. There, we tested this claim with a small CNN ($\mathcal{O}(10^3)$ parameters) presented in figure 7, which we used throughout the manuscript's main body. Here, we test this claim with four larger networks of the following types: convolution-based, i.e. a deeper CNN and ResNet-type architecture, as well as generalized linear and fully-connected NNs.

Hyperparameters We train 100 randomly initialized networks for each architecture. We present architectures of the deeper CNN, generalized linear NN, and fully-connected NN in figures 8(a)–(c), respectively. The used ResNet architecture is an adaptation of ResNet18 [108] with layer 'conv5' omitted due to our input size being smaller than the ImageNet one. For CNN, ResNet, and the linear network, we use the same hyperparameters as described in appendix C. The only exception is a fully-connected NN, where we increase the LR to 10^{-3}.

Fully-connected and linear networks outperform convolutions. We present the comparison results in table 3. The fully-connected and generalized linear networks exhibit excellent OOD accuracy on the phase diagram of the disordered SSH model, when trained only on the disorderless data. All trained instances perform perfectly in the disorderless regime and accurately predict the whole phase diagram (∼82%), on par with the best CNNs. Additionally, contrary to the CNNs, their good OOD performance is a general trend, not a stochastic property. The convolution-based architectures pale in comparison—the ResNet learns perfectly the disorderless regime, but consistently fails to classify correctly the whole phase diagram (all 100 trained networks fall into the poorly-generalizing group). The deeper CNN outperforms its shallower counterpart with as much as 41% of the networks generalizing well. However, increased depth comes at the

Table 3. Statistics (means ± standard deviations) of four classes of 100 randomly initialized networks trained only on data without disorder, $W = 0$. The compared architectures are presented in figure 8.

	Deep CNN		ResNet	Generalized Linear NN	Fully-connected NN
Number of parameters	42 306		2 868 162	5 002	1 364 434
OOD generalization	Good	Poor	Poor	Good	Good
Number of CNNs	41 / 100	59 / 100	100 / 100	100 / 100	100 / 100
Training accuracy	∼100 %	∼100 %	∼100 %	∼100 %	∼100 %
Validation accuracy	∼100 %	∼100 %	∼100 %	∼100 %	∼100 %
Test accuracy	94.9 ± 2.1 %	94.1 ± 1.9 %	93.81 ± 0.34 %	∼ 93 %	∼94 %
OOD accuracy	86.6 ± 3.3 %	71 ± 6 %	50.29 ± 0.30 %	81.396 ± 0.020 %	82.51 ± 0.04 %
RMSE	0.143 ± 0.033	0.30 ± 0.06	0.5078 ± 0.0033	0.1839 ± 0.0004	0.1692 ± 0.0007

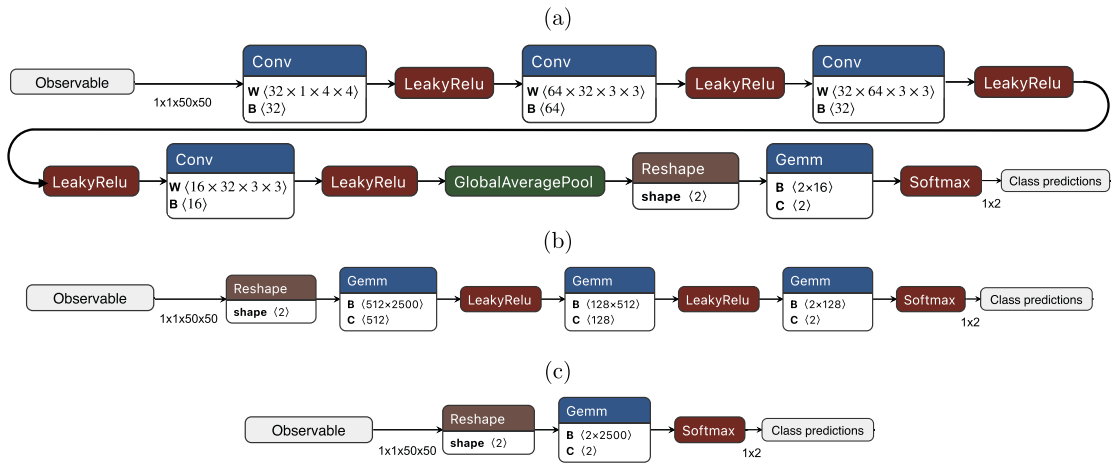

Figure 8. Tested alternative architectures. (a) This CNN is of the same type as presented in figure 7 but has an increased depth. (b) A deep fully-connected NN. (c) The generalized linear NN. Here, the softmax layer only acts as an output normalization, allowing to interpret it as class probabilities.

cost of a 40-fold increase in the number of trainable parameters, and the deeper CNN still underperforms compared to the two fully-connected networks.

Convolutions, edge states, and disorder-induced localization. The comparison between architectures shows that generally networks with convolutional layers struggle with the task of classifying phases of the disordered SSH model, when trained only on disorderless data. This might be because the physically meaningful discriminative features—the zero-energy edge states—always appear in the same pixels of the input image. If not for their position, they can be confused with disorder-induced localization of particles. Convolutional networks are translationally invariant, so they cannot immediately leverage the knowledge of the localization position in the image, contrary to fully-connected networks, which may use the information about the position directly. Convolutional networks need to learn a more subtle representation of the localization to carry out the classification task successfully, which proves to be a challenging and non-deterministic task.

Appendix E. CORAL for architecture-agnostic domain adaptation

CORAL as a way for domain adaptation. CORAL stands for CORrelation ALignment domain adaptation method [109–111]. It aims to align network representations of in- and OOD data by matching their

Table 4. Statistics (means ± standard deviations) of 100 randomly initialized networks trained on disorderless data with CORAL loss added either from the beginning or only when ACC on training and validation dataset exceeded 95%. The addition of CORAL to training drastically worsened the performance on the disorderless test datasets. The table also displays the number of well- and poorly-generalizing networks that meet the necessary condition of good performance (ACC > 90%) on the disorderless test dataset. Their performance statistics are displayed in table 5.

	CNNs trained with CORAL from the start		CNNs trained with CORAL from training ACC > 95%	
OOD generalization	Well-generalizing	Poorly-generalizing	Well-generalizing	Poorly-generalizing
Number of CNNs	14 / 100	86 / 100	43 / 100	57 / 100
Training accuracy	∼100 %	∼100 %	∼100 %	∼100 %
Validation accuracy	75 ± 25 %	59 ± 18 %	93 ± 14 %	67 ± 22 %
Test accuracy	73 ± 23 %	58 ± 16 %	87 ± 13 %	64 ± 19 %
OOD accuracy	84.0 ± 2.0 %	64 ± 9 %	87.5 ± 3.5 %	68 ± 9 %
RMSE	0.165 ± 0.021	0.36 ± 0.09	0.14 ± 0.04	0.34 ± 0.10
Number of CNNs (ACC > 90%)	6 / 14	8 / 86	26 / 43	13 / 57
Percentage of group (ACC > 90%)	43%	9%	60%	23%

second-order statistics. In the case of a linear classifier, it can be done analytically [110]. A visually appealing explanation of the linear version of this method can be seen in figure 2 of [110].

CORAL in NNs. When applied to an NN, CORAL consists of adding to the training loss function a term describing the distance between covariance matrices of activations of training and OOD data [111]:

$$\ell_{\mathrm{CORAL}} = \frac{1}{4d^2}\|C_{\mathrm{train}} - C_{\mathrm{OOD}}\|_{\mathrm{F}}^2, \tag{E1}$$

where d is the number of channels in the output of an NN layer, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm, and C_{train} and C_{OOD} are covariance matrices of activations of training and OOD data, as defined in equations (2) and (3) of [111]. These activations can be studied at different network layers or at many t layers simultaneously, each weighted by a hyperparameter λ_i. Then, the total loss is a sum of the classification loss and CORAL terms coming from different layers as defined by equation (E2).

$$\ell = \ell_{\mathrm{class.}} + \sum_{i=1}^{t} \lambda_i \ell_{\mathrm{CORAL}}\,. \tag{E2}$$

Implementation details. We test two approaches to incorporate CORAL into the training routine. The first one is to train the network from scratch with the loss function defined by equation (E2), and the second is only to add the CORAL term to the classification loss when the training and validation accuracies surpass 95%. The second approach serves as a way of fine-tuning architectures already leaning towards a classification loss minimum. As per [111], we choose the value of λ so that at the end of the training, the $\ell_{\mathrm{class.}}$ and $\sum_{i=1}^{t}\lambda_i\ell_{\mathrm{CORAM}}$ reach an equilibrium and their values are 'roughly the same'. Following [111], we apply ℓ_{CORAL} to only one layer, i.e. the last convolution before GAP. Note that the OOD data used for the CORAL is unlabeled and sampled across the whole target phase diagram, where for every value of $W/w \in [0,5]$ 20 randomly chosen samples from each class were picked.

CORAL worsens the disorderless test accuracy. In both cases, the test accuracy on disorderless data drops significantly, with the vast majority of trained networks performing under 90% disorderless test accuracy. To mitigate the effect, several λ values were tested. When λ is large, $\mathcal{O}(10)$, the optimization process focuses only on aligning representations, completely neglecting the classification loss. On the other hand, when λ is small, $\mathcal{O}(10^{-5})$, CNNs heavily overfit the disorderless training data. We report the results for intermediate λ of $\mathcal{O}(10^{-2})$ in table 4. Most CNNs (86%) still severely overfit (with test accuracy below 60%), but gain slightly better OOD accuracy (≈64%). Finally, of the networks that perform well on the disorderless dataset, the good generalization is encountered more often, as presented in table 5. Therefore, CORAL improves CNN performance on disordered data, but usually at the cost of lower performance in the disorderless regime.

Table 5. Statistics (means ± standard deviations) of the networks that meet the necessary condition of good performance (ACC > 90%) on the disorderless test dataset out of 100 random initializations. The networks were trained on disorderless data with CORAL loss added from the beginning or only when ACC on the training and validation dataset exceeded 95%.

	CNNs trained with CORAL from the start		CNNs trained with CORAL from training ACC > 95%	
OOD generalization	Well-generalizing	Poorly-generalizing	Well-generalizing	Poorly-generalizing
Number of CNNs	6 / 14	8 / 14	26 / 43	13 / 43
Training accuracy	∼100 %	∼100 %	∼100 %	∼100 %
Validation accuracy	∼100 %	∼100 %	∼100 %	∼100 %
Test accuracy	95.7 ± 3.0 %	97.8 ± 2.2 %	95.2 ± 3.2 %	95.0 ± 2.4 %
OOD accuracy	85.0 ± 2.1 %	75 ± 6 %	88.7 ± 2.8 %	71 ± 8 %
RMSE	0.157 ± 0.020	0.27 ± 0.06	0.130 ± 0.033	0.34 ± 0.09

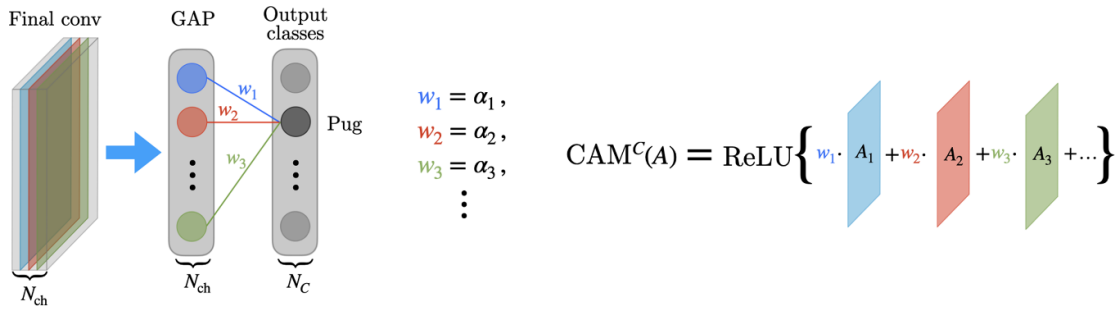

Figure 9. Class activation mapping (CAM) generates a map that highlights relevant parts of the input data for predicting a selected class C (out of N_C classes), e.g. a pug out of all 1000 ImageNet categories. In the final convolutional layer of the network, each of the N_{ch} channels generates a convolved image, A_k (three out of N_{ch} are marked with blue, red, and green squares in the figure). Each activation can encode some data features relevant to some of the classes. An intuitive example of such features could be the shortened snout of a pug or cat's whiskers. The GAP layer then reduces each of these activations to its mean value (e.g. each colored square got reduced to the scalar of the same color). Finally, the fully-connected layer maps GAP values to the network outputs and its weights are used as weights for the CAM. Inspired by [70].

Appendix F. Gradient-based interpretability techniques are fragile

Introduction to CAM. CAM is an attribution technique tailored for use with CNNs and leverages their design. A CNN produces different representations of the input data during a forward pass through the network and encodes them in different channels. CAM relies on the latent representation of the data present in the N_{ch} output channels of the last convolutional layer in a network (activations A_k), by performing their weighted sum with weights α_k [84],

$$\mathrm{CAM}^C(A) = \mathrm{ReLU}\left(\sum_{k=1}^{N_{ch}} \alpha_k A_k\right). \tag{F1}$$

The resulting map, after rescaling back to the original data size, highlights the areas that were the most influential in predicting a considered class C. The crucial part is the choice of the weights α_k. The original formulation of the method relies on reducing each channel output of the last convolutional layer to a single number with GAP layer [70]. The weights of fully-connected layer connecting those numbers with the output corresponding to the considered class C are then used as α_k for the weighted average of convolved images. We present this explanation graphically in figure 9.

CAM fails for the disordered regime. As discussed before, CAM provides invaluable insight into possible correlations in disorderless data. However, the results proved unreliable when applied to data from a disordered regime ($W/w \leqslant 0.05$). Contrary to the previously observed and expected tendencies, CAM importance maps were asymmetric and varied significantly between neighboring v values and multiple realizations of the same point in phase space. This example is presented in figures 10(c) and (d).

NN's fragility. The CNNs have been shown to be prone to 'adversarial noise' attacks [114, 115]. They can take various intricate forms, as discussed by [116]. The simplest scenario of an adversarial noise attack is one where the input image is perturbed in a deliberately engineered way to fool the CNN and make it predict a

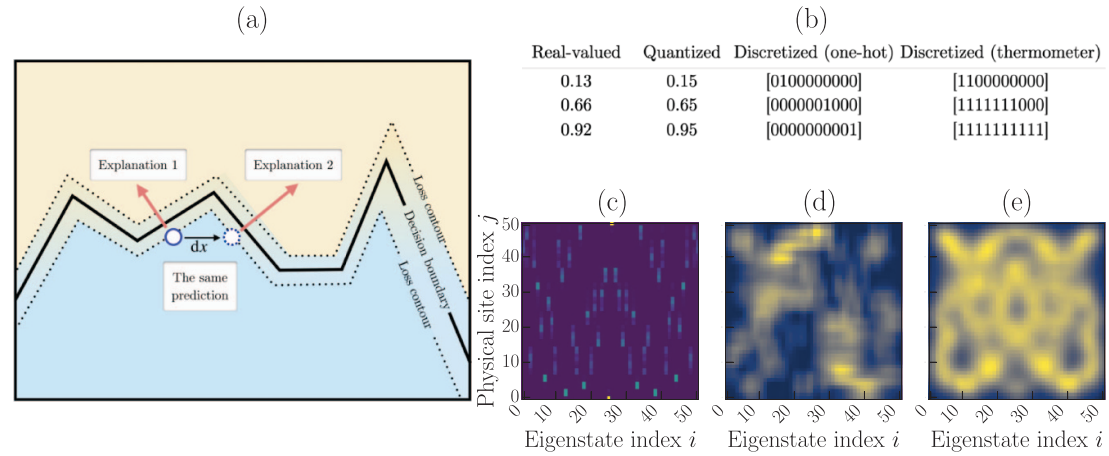

Figure 10. Interpretation fragility. (a) Gradient-based attribution techniques (CAM included) rely on the direction of the gradient (pink arrow), so small perturbation dx of a test image can vary the basis of explanation, leaving class prediction unaffected. Panel inspired by [112]. (b) Thermometer encoding, a solution to this problem proposed by Buckman *et al* [113], is a discretization technique for encoding a floating point input variable as a vector representing its magnitude. Reproduced from [113]. CC BY 4.0. (d) In a disordered regime, CAM yields flawed interpretations. (c) The input is axially symmetric—a property that the interpretation should preserve. (e) This is remedied by using thermometer encoding at the cost of limited network performance and a rise in computational cost.

different class while keeping the input visually indistinguishable for a human. This class of attacks targets the predictive power of CNN. However, the problem in our case is the more subtle—predictions remain factual, and the gradient-based interpretation fails.

Interpretation fragility. The topic of attacks on interpretation alone has already been addressed by the computer science community [112, 117, 118]. The intuition provided by [112] is as follows. Gradient-based techniques, like CAM, rely on the direction one must go from any given test example to the decision boundary. This is the geometric interpretation of the gradient in parameter space. The decision boundary, being highly non-convex, makes it very easy to perturb a given test image while still within the boundaries of its class. This slight shift is enough for the gradient vector to point in another direction, producing a different interpretation. See figure 10(a).

Thermometer encoding. Buckman *et al* [113] have proposed using thermometer encoding of input to remedy this problem. It belongs to a broader class of approaches to reducing problem complexity via casting input with floating point entries to discrete variables. A standard input discretization technique, one-hot encoding, encodes a quantized floating point number as a vector with the only non-zero entry marking the range in which it falls. [113] argues that by mimicking the behavior of a mercury thermometer, encoding the magnitude of the encoded input, the network gradients are better behaved. This change, they argue, reduces the fragility of gradient-based network attribution techniques. The thermometer encoding process is demonstrated in figure 10(b).

Thermometer recovers CAM. We have implemented thermometer encoding with the number of discretization bins ranging from 10 to 100 to test its applicability to our problem. We have successfully recovered the insight provided by CAM, as displayed in figure 10(e). We have found the 100 discretization bins to be the optimal value for the problem. This technique retrieves CAM interpretation for all disorders in the range addressed by this study.

Cost of CAM interpretability. While gradient-based interpretation is recovered, there is collateral damage done to network performance and training costs. None out of 100 networks trained using this thermometer encoding generalized well to data with disorder, even when trained on data with added small disorder $W/w \in 0, 0.01, 0.05$. This is a significant performance loss when compared to 22% of well-generalizing networks obtained when the architecture is not adjusted to the thermometer encoding. There is also a substantial rise in the computational cost of network training. For 100 discretization bins, the number of trainable parameters rises from $\mathcal{O}(10^3)$ to $\mathcal{O}(10^9)$. This, in turn, extended the training time of a single network from $\mathcal{O}(1\text{min})$ to $\mathcal{O}(2\text{h})$. The training was performed in PyTorch [78] on a CUDA-enabled NVIDIA GeForce GTX 1050 graphics card with 4 GB of GDDR5 VRAM.

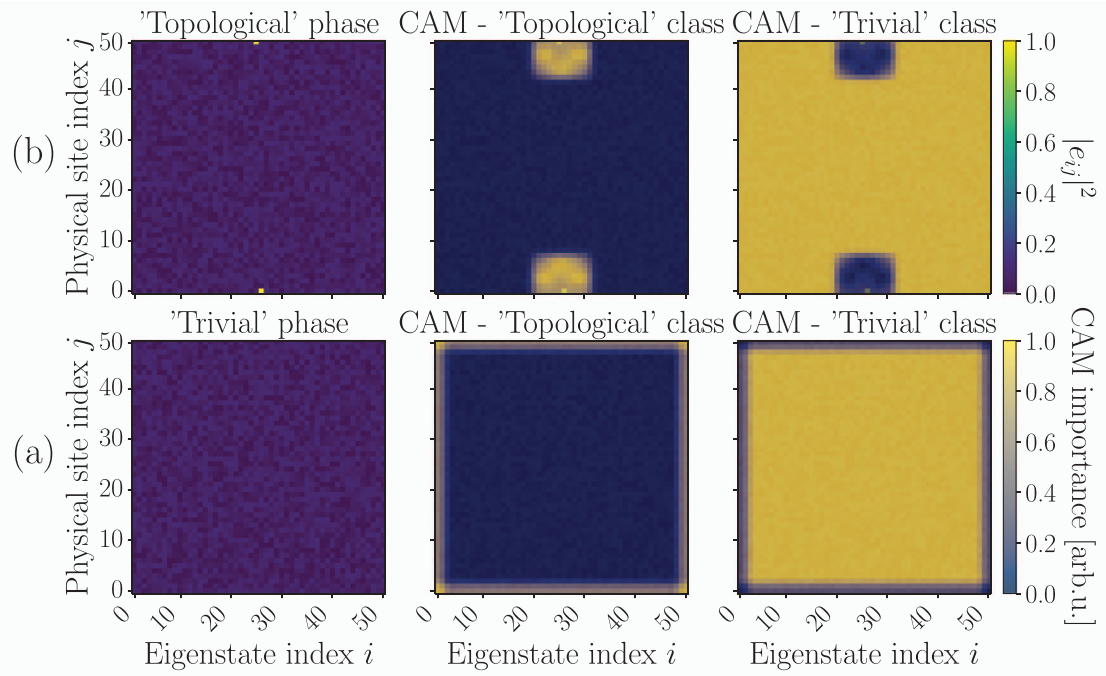

Figure 11. CAM analysis of a network trained and tested in a simplified setting. All entries in data points are drawn from the uniform distribution $U[0, 0.1]$. In half of the data, we set the edge entries of two middle columns to 1. This constitutes a 'Topological' phase, with the second part labeled 'Trivial' phase. CAM analysis of this setting confirms that the features the network uses for class discrimination are the bright pixels.

Appendix G. Toy model for CAM performance assessment

The motivation behind our use of CAM can be demonstrated using a toy model of the problem at hand, imitating the SSH model with a slight disorder. In this setting, we create an artificial data set consisting of data points, where all entries are drawn from the uniform distribution $U[0, 0.1]$. We then separate it into two parts and set the edge entries of two middle columns to 1. In this setting, the phases are not distinguished by any correlations other than the two bright pixels we set. CAM analysis of a network trained and tested in this setting shows that the features the network uses for class discrimination are the bright pixels. This result is shown in figure 11.

Appendix H. Evolution of the data representation across layers of a NN

Representation varies between network layers. We can also study the evolution of the data representation across layers, which gives additional insights into the behavior of different networks. To this end, we now investigate outputs of the three consecutive convolutional layers (Conv1, Conv2, Conv3), next to the output of the GAP layer, which we have studied in the main text in figure 4. Contrary to the discussion in the main text, here we do not remove the data from the vicinity of the transition point. We also inspect the respective representation of the test data without disorder and OOD test data with a single realization of multiple disorder amplitudes ($W/w \in 0.15, 0.5, 1, 2$). However, not all data sets can be visualized accurately in a few dimensions using only linear transformations. Therefore, we chose a nonlinear technique to visualize the underlying structure of the latent space representations in the CNNs more accurately than PCA, described in section 2.4. To this end, we use the UMAP [119].

UMAP—nonlinear dimensionality reduction. As demonstrated in [120], UMAP is a two-step routine: first, it computes a graph that represents the data and then learns a low-dimensional representation of the graph while trying to preserve the local, global, and topological structure of the data. UMAP is good at visualizing high-dimensional nonlinear data sets, but it has some drawbacks. First, the embedding learning process is stochastic, which means that the reproducibility of the results is limited. Second, it can distort the true nature of data connectivity and density [121–123], so it is essential to be cautious when using it to conclude the clustering or concentration of data. It is a good practice to verify that the resulting number of clusters agrees with what we expect to find in the data based on results from other techniques. Despite its limitations, UMAP has proven to be computationally inexpensive and is often used because of its ability to visualize

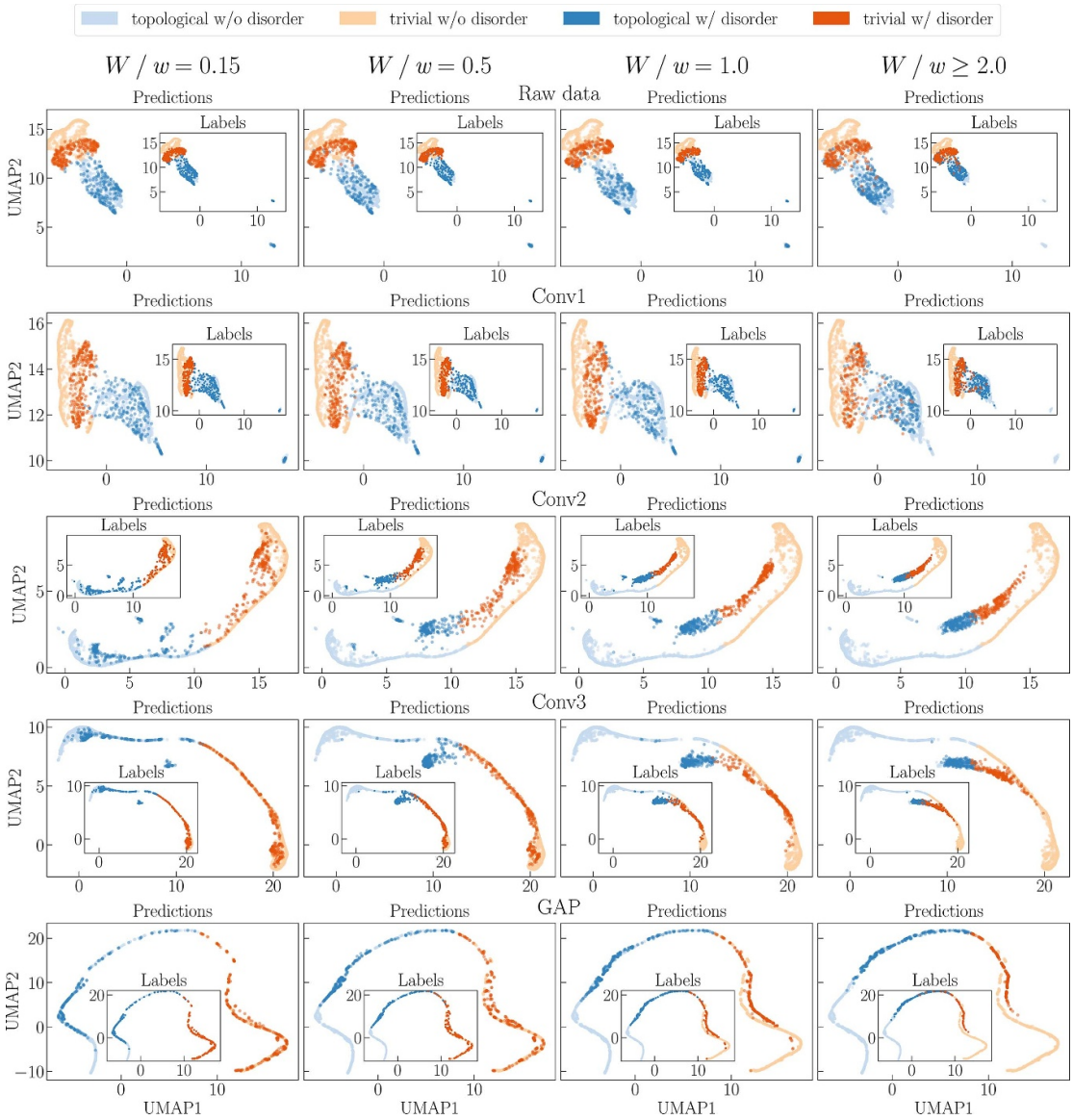

Figure 12. Evolution of the internal data representation with UMAP within different layers of the well-generalizing CNN that does not focus on edge states, as seen figures 2(d) and (h).

high-dimensional data efficiently. These are the reasons behind our decision to use it to analyze the latent space representations in the CNNs we studied. The UMAP embeddings are visualized in figures 12 and 13 for well- and poorly-generalizing networks, respectively.

UMAP results for raw data. Analysis of raw data with UMAP (first row of figures 12 and 13) allows for two interesting observations. Firstly, surprisingly, the UMAP embedding is itself a better classifier of data from disordered regimes than the majority of CNNs. The embedding of both classes in such data overlaps with the embeddings of classes from a disorderless system. This could allow for an informed decision about proper labels in the OOD regime. Secondly, it is a testament to the previously mentioned stochasticity of low-dimensional representations. The raw data subject to analysis are the same in both figures, but their embedding is slightly different. This is a result of UMAP's limited reproducibility. Upon careful inspection, these UMAP embeddings led to one more observation: that the data coming from the topological class formed two disjoint clusters. This phenomenon is due to the hybridization of the edge states [72].

Well-generalizing CNNs represent data with and without disorder similarly. Let us first focus on the two-dimensional data representation generated by UMAP for the well-generalizing network, as seen in figure 12. In the consecutive columns, we plot the disorderless data and the data with increasing disorder strength (indicated with darker orange and blue). In the first convolutional layer (Conv1), the disordered data are represented very similarly to those without the disorder. In other words, their representation

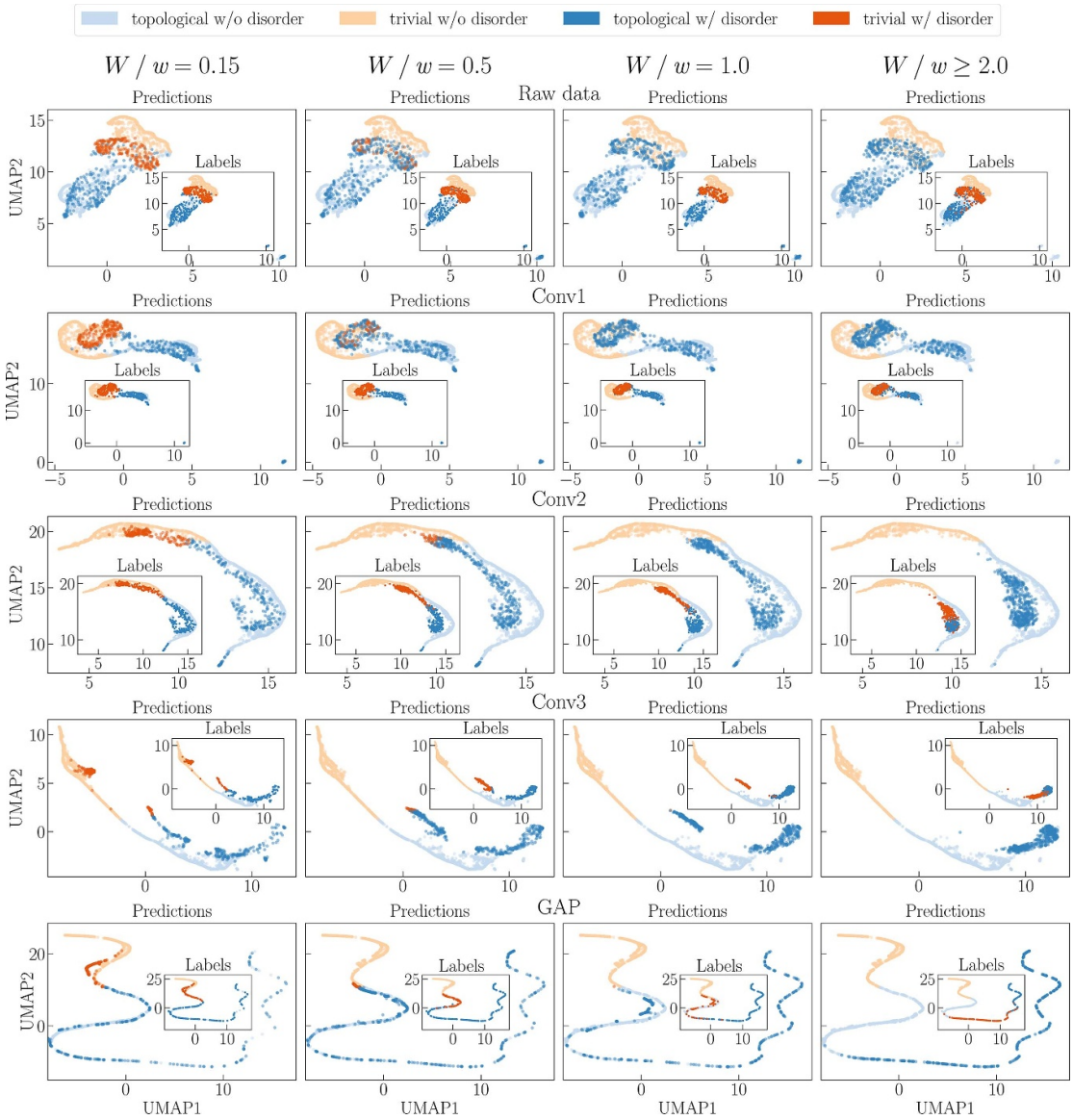

Figure 13. Evolution of the internal data representation with UMAP within different layers of the poorly-generalizing CNN that focuses on edge states, as seen figures 2(c) and (g).

overlaps. In the following two layers (Conv2, Conv3), the class representations overlap for intermediate disorder amplitudes ($W/w \in 0.5, 1$. With increasing disorder amplitude ($W/w \geqslant 2$), the embedded data start to disconnect from the original clusters. In particular, the embedded Conv3 layer activations, which initially clustered with the topological class, begin forming a class of its own. This may signify a change in the data structure the network has learned to observe. It also means that at this stage of the forward pass, the network has a disjointed representation only of strongly disordered data.

The lowest row presents embedded activations from the GAP layer, a qualitatively different picture. Knowing the nature of GAP output, it seems safe to assume that what we see here is an embedding of a unidimensional manifold. As disorder increases, the initially disjoint clusters of disorderless data gradually connect with data from increasingly disordered regimes. This captures well the perceived similarity of both classes an observer, artificial or human, has when observing the input data.

Poorly-generalizing CNNs have very different representation of data with and without disorder. Let us now move on to the two-dimensional data representation generated by UMAP for the poorly-generalizing network, as seen in figure 13. In the first convolutional layer (Conv1), the disordered data are represented very similarly to those without the disorder. This behaviour is consistent with the one observed for the well-generalizing network. In the following two layers (Conv2, Conv3), the class representations overlap for intermediate disorder amplitudes ($W/w \in 0.5, 1$. With increasing disorder amplitude ($W/w \geqslant 2$), the embedded data start to disconnect from the original clusters. Given the poor generalizability we expect from

this network, it is not surprising to see them merge together, and then finally be positioned near just one of the initial distributions. The fact that it gets positioned near the topological phase representation might suggest that this would be the phase predicted in high disorder regime, and this notion turns out to be true here.

The lowest row of figure 12 is once again a qualitatively different picture from the previous layers, but of the same nature as embedding of GAP layer activations from well-generalizing network. This further strengthens the assumption that is an embedding of a unidimensional manifold. Here, however, it is hard to say surely that the initial clusters of disorderless data are well separated. Their, already low, separation vanishes as disorder increases. The disordered data then proceed to merge together and position in the range of topological class. This, once again, correctly suggests that this would be the phase predicted in high disorder regime.

ORCID iDs

Kacper Cybiński https://orcid.org/0000-0003-2600-7473
Marcin Płodzień https://orcid.org/0000-0002-0835-1644
Michał Tomza https://orcid.org/0000-0003-1792-8043
Maciej Lewenstein https://orcid.org/0000-0002-0210-7800
Alexandre Dauphin https://orcid.org/0000-0003-4996-2561
Anna Dawid https://orcid.org/0000-0001-9498-1732

References

[1] Dawid A and LeCun Y 2023 Introduction to latent variable energy-based models: a path towards autonomous machine intelligence (arXiv:2306.02572 [cs.LG])
[2] Hermann J, Spencer J, Choo K, Mezzacapo A, Foulkes W M C, Pfau D, Carleo G and Noé F 2023 *Ab initio* quantum chemistry with neural-network wavefunctions *Nat. Rev. Chem.* **7** 692–709
[3] Medvidović M and Moreno J R 2024 Neural-network quantum states for many-body physics *Eur. Phys. J. Plus* **139** 631
[4] Lange H, Van de Walle A, Abedinnia A and Bohrdt A 2024 From architectures to applications: a review of neural quantum states *Quantum Sci. Technol.* **9** 040501
[5] Dawid A *et al* 2023 Modern applications of machine learning in quantum sciences (arXiv:2204.04198 [quant-ph])
[6] Krenn M, Landgraf J, Foesel T and Marquardt F 2023 Artificial intelligence and machine learning for quantum technologies *Phys. Rev.* A **107** 010101
[7] Mafu M 2024 Advances in artificial intelligence and machine learning for quantum communication applications *IET Quantum Commun.* **5** 202–31
[8] Lu S, Huang S, Li K, Li J, Chen J, Lu D, Ji Z, Shen Y, Zhou D and Zeng B 2018 Separability-entanglement classifier via machine learning *Phys. Rev.* A **98** 012315
[9] Sanavio C, Tignone E and Ercolessi E 2023 Entanglement classification via witness operators generated by support vector machine *Eur. Phys. J. Plus* **138** 936
[10] Carrasquilla J and Melko R G 2017 Machine learning phases of matter *Nat. Phys.* **13** 431
[11] Li C-D, Tan D-R and Jiang F-J 2018 Applications of neural networks to the studies of phase transitions of two-dimensional Potts models *Ann. Phys., NY* **391** 312
[12] Schäfer F and Lörch N 2019 Vector field divergence of predictive model output as indication of phase transitions *Phys. Rev.* E **99** 062107
[13] Bachtis D, Aarts G and Lucini B 2020 Mapping distinct phase transitions to a neural network *Phys. Rev.* E **102** 053306
[14] Liu K, Sadoune N, Rao N, Greitemann J and Pollet L 2021 Revealing the phase diagram of Kitaev materials by machine learning: cooperation and competition between spin liquids *Phys. Rev. Res.* **3** 023016
[15] Richter-Laskowska M, Kurpas M and Maśka M M 2023 Learning by confusion approach to identification of discontinuous phase transitions *Phys. Rev.* E **108** 024113
[16] Van Nieuwenburg E P L, Liu Y-H and Huber S D 2017 Learning phase transitions by confusion *Nat. Phys.* **13** 435
[17] Wetzel S J 2017 Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders *Phys. Rev.* E **96** 022140
[18] Liu Y-H and van Nieuwenburg E P L 2018 Discriminative cooperative networks for detecting phase transitions *Phys. Rev. Lett.* **120** 176401
[19] Ch'ng K, Vazquez N and Khatami E 2018 Unsupervised machine learning account of magnetic transitions in the Hubbard model *Phys. Rev.* E **97** 013306
[20] Huembeli P, Dauphin A, Wittek P and Gogolin C 2019 Automated discovery of characteristic features of phase transitions in many-body localization *Phys. Rev.* B **99** 104106
[21] Kottmann K, Huembeli P, Lewenstein M and Acín A 2020 Unsupervised phase discovery with deep anomaly detection *Phys. Rev. Lett.* **125** 170603
[22] Kottmann K, Corboz P, Lewenstein M and Acín A 2021 Unsupervised mapping of phase diagrams of 2D systems from infinite projected entangled-pair states via deep anomaly detection *SciPost Phys.* **11** 025
[23] Arnold J, Schäfer F, Žonda M and Lode A U J 2021 Interpretable and unsupervised phase classification *Phys. Rev. Res.* **3** 033052
[24] Patel Z, Merali E and Wetzel S J 2022 Unsupervised learning of Rydberg atom array phase diagram with Siamese neural networks *New J. Phys.* **24** 113021
[25] Szodra T, Sierant P, Kottmann K, Lewenstein M and Zakrzewski J 2021 Detecting ergodic bubbles at the crossover to many-body localization using neural networks *Phys. Rev.* B **104** L140202

[26] Szodra T, Sierant P, Lewenstein M and Zakrzewski J 2022 Unsupervised detection of decoupled subspaces: many-body scars and beyond *Phys. Rev.* B **105** 224205
[27] Broecker P, Carrasquilla J, Melko R G and Trebst S 2017 Machine learning quantum phases of matter beyond the fermion sign problem *Sci. Rep.* **7** 8823
[28] Théveniaut H and Alet F 2019 Neural network setups for a precise detection of the many-body localization transition: finite-size scaling and limitations *Phys. Rev.* B **100** 224202
[29] Dong X-Y, Pollmann F and Zhang X-F 2019 Machine learning of quantum phase transitions *Phys. Rev.* B **99** 121104
[30] Blücher S, Kades L, Pawlowski J M, Strodthoff N and Urban J M 2020 Towards novel insights in lattice field theory with explainable machine learning *Phys. Rev.* D **101** 094507
[31] Zhang P, Shen H and Zhai H 2018 Machine learning topological invariants with neural networks *Phys. Rev. Lett.* **120** 066401
[32] Tsai Y-H, Yu M-Z, Hsu Y-H and Chung M-C 2020 Deep learning of topological phase transitions from entanglement aspects *Phys. Rev.* B **102** 054512
[33] Baireuther P, Płodzień M, Ojanen T, Tworzydło J and Hyart T 2023 Identifying Chern numbers of superconductors from local measurements *SciPost Phys. Core* **6** 087
[34] Huembeli P, Dauphin A and Wittek P 2018 Identifying quantum phase transitions with adversarial neural networks *Phys. Rev.* B **97** 134109
[35] Rodriguez-Nieva J F and Scheurer M S 2019 Identifying topological order through unsupervised machine learning *Nat. Phys.* **15** 790
[36] Greplova E, Valenti A, Boschung G, Schäfer F, Lörch N and Huber S D 2020 Unsupervised identification of topological phase transitions using predictive models *New J. Phys.* **22** 045003
[37] Balabanov O and Granath M 2021 Unsupervised interpretable learning of topological indices invariant under permutations of atomic bands *Mach. Learn.: Sci. Technol.* **2** 025008
[38] Yu L-W, Zhang S-Y, Shen P-X and Deng D-L 2023 Unsupervised learning of interacting topological phases from experimental observables *Fundam. Res.* **4** 1086–91
[39] Teng Y, Sachdev S and Scheurer M S 2023 Clustering neural quantum states via diffusion maps *Phys. Rev.* B **108** 205152
[40] Rem B S, Käming N, Tarnowski M, Asteria L, Fläschner N, Becker C, Sengstock K and Weitenberg C 2019 Identifying quantum phase transitions using artificial neural networks on experimental data *Nat. Phys.* **15** 917
[41] Khatami E, Guardado-Sanchez E, Spar B M, Carrasquilla J F, Bakr W S and Scalettar R T 2020 Visualizing strange metallic correlations in the two-dimensional fermi-hubbard model with artificial intelligence *Phys. Rev.* A **102** 033326
[42] Käming N, Dawid A, Kottmann K, Lewenstein M, Sengstock K, Dauphin A and Weitenberg C 2021 Unsupervised machine learning of topological phase transitions from experimental data *Mach. Learn.: Sci. Technol.* **2** 035037
[43] Miles C, Samajdar R, Ebadi S, Wang T T, Pichler H, Sachdev S, Lukin M D, Greiner M, Weinberger K Q and Kim E-A 2023 Machine learning discovery of new phases in programmable quantum simulator snapshots *Phys. Rev. Res.* **5** 013026
[44] Link M, Gao K, Kell A, Breyer M, Eberz D, Rauf B and Köhl M 2023 Machine learning the phase diagram of a strongly interacting Fermi gas *Phys. Rev. Lett.* **130** 203401
[45] Wang L 2016 Discovering phase transitions with unsupervised learning *Phys. Rev.* B **94** 195105
[46] Che Y, Gneiting C, Liu T and Nori F 2020 Topological quantum phase transitions retrieved through unsupervised machine learning *Phys. Rev.* B **102** 134213
[47] Long Y, Ren J and Chen H 2020 Unsupervised manifold clustering of topological phononics *Phys. Rev. Lett.* **124** 185501
[48] Yang Y, Sun Z-Z, Ran S-J and Su G 2021 Visualizing quantum phases and identifying quantum phase transitions by nonlinear dimensional reduction *Phys. Rev.* B **103** 075106
[49] Vargas-Hernández R A, Sous J, Berciu M and Krems R V 2018 Extrapolating quantum observables with machine learning: inferring multiple phase transitions from properties of a single phase *Phys. Rev. Lett.* **121** 255702
[50] Greitemann J, Liu K, Jaubert L D C, Yan H, Shannon N and Pollet L 2019 Identification of emergent constraints and hidden order in frustrated magnets using tensorial kernel methods of machine learning *Phys. Rev.* B **100** 174408
[51] Greitemann J, Liu K and Pollet L 2021 The view of TK-SVM on the phase hierarchy in the classical kagome Heisenberg antiferromagnet *J. Phys.: Condens. Matter* **33** 054002
[52] Leykam D and Angelakis D G 2023 Topological data analysis and machine learning *Adv. Phys.* X **8** 2202331
[53] Scheurer M S and Slager R-J 2020 Unsupervised machine learning and band topology *Phys. Rev. Lett.* **124** 226401
[54] Cole A, Loges G J and Shiu G 2021 Quantitative and interpretable order parameters for phase transitions from persistent homology *Phys. Rev.* B **104** 104426
[55] Park S, Hwang Y and Yang B-J 2022 Unsupervised learning of topological phase diagram using topological data analysis *Phys. Rev.* B **105** 195115
[56] Schuld M and Killoran N 2019 Quantum machine learning in feature Hilbert spaces *Phys. Rev. Lett.* **122** 040504
[57] Caro M C, Huang H-Y, Cerezo M, Sharma K, Sornborger A, Cincio L and Coles P J 2022 Generalization in quantum machine learning from few training data *Nat. Commun.* **13** 4919
[58] Liu Y-J, Smith A, Knap M and Pollmann F 2023 Model-independent learning of quantum phases of matter with quantum convolutional neural networks *Phys. Rev. Lett.* **130** 220603
[59] Dawid A, Huembeli P, Tomza M, Lewenstein M and Dauphin A 2020 Phase detection with neural networks: interpreting the black box *New J. Phys.* **22** 115001
[60] Wetzel S J, Melko R G, Scott J, Panju M and Ganesh V 2020 Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks *Phys. Rev. Res.* **2** 033499
[61] Dawid A, Huembeli P, Tomza M, Lewenstein M and Dauphin A 2021 Hessian-based toolbox for reliable and interpretable machine learning in physics *Mach. Learn.: Sci. Technol.* **3** 015002
[62] Arnold J and Schäfer F 2022 Replacing neural networks by optimal analytical predictors for the detection of phase transitions *Phys. Rev.* X **12** 031044
[63] Wetzel S J 2024 Closed-form interpretation of neural network classifiers with symbolic regression gradients (arXiv:2401.04978 [cs.LG])
[64] Wetzel S J and Scherzer M 2017 Machine learning of explicit order parameters: from the Ising model to SU(2) lattice gauge theory *Phys. Rev.* B **96** 184410
[65] Greitemann J, Liu K and Pollet L 2019 Probing hidden spin order with interpretable machine learning *Phys. Rev.* B **99** 060404
[66] Liu K, Greitemann J and Pollet L 2019 Learning multiple order parameters with interpretable machines *Phys. Rev.* B **99** 104410

[67] Miles C, Bohrdt A, Wu R, Chiu C, Xu M, Ji G, Greiner M, Weinberger K Q, Demler E and Kim E-A 2021 Correlator convolutional neural networks as an interpretable architecture for image-like quantum matter data *Nat. Commun.* **12** 3905
[68] Tran D *et al* 2022 Plex: towards reliability using pretrained large model extensions *First Workshop on Pre-Training: Perspectives, Pitfalls and Paths Forward at ICML 2022* (arXiv:2207.07411)
[69] Redko I, Morvant E, Habrard A, Sebban M and Bennani Y 2022 A survey on domain adaptation theory: learning bounds and theoretical guarantees (arXiv:2004.11829 [cs.LG])
[70] Zhou B, Khosla A, Lapedriza A, Oliva A and Torralba A 2016 Learning deep features for discriminative localization *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition* (available at: http://cnnlocalization.csail.mit.edu/) pp 2921–9
[71] Dawid-Łękowska A M 2022 Quantum many-body physics with ultracold atoms and molecules: exact dynamics and machine learning *PhD Thesis* University of Warsaw, Universitat Politècnica de Catalunya. Institut de Ciències Fotòniques (https://doi.org/10.5821/dissertation-2117-379455)
[72] Asbóth J K, Oroszlány L and Pályi A 2016 *A Short Course on Topological Insulators* (Springer) (https://doi.org/10.1007/978-3-319-25607-8)
[73] Mondragon-Shem I, Hughes T L, Song J and Prodan E 2014 Topological criticality in the chiral-symmetric AIII class at strong disorder *Phys. Rev. Lett.* **113** 046802
[74] Meier E J, An F A, Dauphin A, Maffei M, Massignan P, Hughes T L and Gadway B 2018 Observation of the topological Anderson insulator in disordered atomic wires *Science* **362** 929
[75] Le N H, Fisher A J, Curson N J and Ginossar E 2020 Topological phases of a dimerized Fermi–Hubbard model for semiconductor nano-lattices *npj Quantum Inf.* **6** 24
[76] Fraxanet J, Bhattacharya U, Grass T, Rakshit D, Lewenstein M and Dauphin A 2021 Topological properties of the long-range Kitaev chain with Aubry-André-Harper modulation *Phys. Rev. Res.* **3** 013148
[77] Zhang J M and Dong R X 2010 Exact diagonalization: the Bose-Hubbard model as an example *Eur. J. Phys.* **31** 591
[78] Paszke A *et al* 2019 PyTorch: an imperative style, high-performance deep learning library (arXiv:1912.01703 [cs.LG])
[79] Good I J 1952 Rational decisions *J. R. Stat. Soc.* B **14** 107
[80] Anderson P W 1958 Absence of diffusion in certain random lattices *Phys. Rev.* **109** 1492
[81] Ye W, Zheng G, Cao X, Ma Y, Hu X and Zhang A 2024 Spurious correlations in machine learning: a survey (arXiv:2402.12715 [cs.LG])
[82] Kim B, Khanna R and Koyejo O O 2016 Examples are not enough, learn to criticize! Criticism for interpretability *Advances in Neural Information Processing Systems* vol 29 (Curran Associates, Inc.)
[83] Molnar C 2023 *Interpretable Machine Learning* (Published independently) (available at: https://christophm.github.io/interpretable-ml-book/)
[84] Jung H and Oh Y 2021 Towards better explanations of class activation mapping *2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)* (IEEE Computer Society) pp 1316–24
[85] Smilkov D, Thorat N, Kim B, Viégas F and Wattenberg M 2017 Smoothgrad: removing noise by adding noise (https://doi.org/10.48550/arXiv.1706.03825)
[86] Selvaraju R R, Cogswell M, Das A, Vedantam R, Parikh D and Batra D 2020 Grad-CAM: visual explanations from deep networks via gradient-based localization *Int. J. Comput. Vis.* **128** 336
[87] Chattopadhay A, Sarkar A, Howlader P and Balasubramanian V N 2018 Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks *2018 IEEE Winter Conf. on Applications of Computer Vision (WACV)* pp 839–47
[88] Desai S and Ramaswamy H G 2020 Ablation-CAM: visual explanations for deep convolutional network via gradient-free localization *2020 IEEE Winter Conf. on Applications of Computer Vision (WACV)* pp 972–80
[89] Fu R, Hu Q, Dong X, Guo Y, Gao Y and Li B 2020 Axiom-based grad-CAM: towards accurate visualization and explanation of CNNs (arXiv:2008.02312 [cs.CV])
[90] Wang H, Wang Z, Du M, Yang F, Zhang Z, Ding S, Mardziel P and Hu X 2020 Score-CAM: score-weighted visual explanations for convolutional neural networks *2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)* pp 111–9
[91] Erhan D, Bengio Y, Courville A C and Vincent P 2009 Visualizing higher-layer features of a deep network *2009 Int. Conf. on Machine Learning (ICML)*
[92] Simonyan K, Vedaldi A and Zisserman A 2014 Deep inside convolutional networks: visualising image classification models and saliency maps (arXiv:1312.6034 [cs.CV])
[93] Nguyen A, Yosinski J and Clune J 2015 Deep neural networks are easily fooled: high confidence predictions for unrecognizable images *2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* 427–36 (IEEE)
[94] Olah C, Mordvintsev A and Schubert L 2017 Feature visualization *Distill* **2**
[95] Zhou B, Bau D, Oliva A and Torralba A 2019 Interpreting deep visual representations via network dissection *IEEE Trans. Pattern Anal. Mach. Intell.* **41** 2131
[96] Carter S, Armstrong Z, Schubert L, Johnson I and Olah C 2019 Activation atlas *Distill* **4**
[97] Kauffmann J, Esders M, Ruff L, Montavon G, Samek W and Müller K-R 2024 From clustering to cluster explanations via neural networks *IEEE Trans. Neural Netw. Learn. Syst.* **35** 1926
[98] Pearson K 1901 LIII. On lines and planes of closest fit to systems of points in space *London, Edinburgh Dublin Phil. Mag. J. Sci.* **2** 559
[99] Molignini P, Zegarra A, van Nieuwenburg E, Chitra R and Chen W 2021 A supervised learning algorithm for interacting topological insulators based on local curvature *SciPost Phys.* **11** 073
[100] Tibaldi S, Magnifico G, Vodola D and Ercolessi E 2023 Unsupervised and supervised learning of interacting topological phases from single-particle correlation functions *SciPost Phys.* **14** 005
[101] Madry A, Makelov A, Schmidt L, Tsipras D and Vladu A 2018 Towards deep learning models resistant to adversarial attacks *Int. Conf. on Learning Representations (ICLR)* (https://doi.org/10.48550/arXiv.1706.06083)
[102] Uría-Álvarez A J, Molpeceres-Mingo D and Palacios J J 2022 Deep learning for disordered topological insulators through their entanglement spectrum *Phys. Rev.* B **105** 155128
[103] Tsipras D, Santurkar S, Engstrom L, Turner A and Madry A 2020 Robustness may be at odds with accuracy *Int. Conf. on Learning Representations*
[104] Cybiński K, Płodzień M, Tomza M, Lewenstein M, Dauphin A and Dawid A 2024 Github repository: interpreting_NNs_for_topological_phases_of_matter (Version arXiv 1.0) *Zenodo* https://doi.org/10.5281/zenodo.12518288 (Accessed 24 June 2024)
[105] Ohana R *et al* 2024 The well: a large-scale collection of diverse physics simulations for machine learning (arXiv:2412.02527)

[106] Angeloudi E et al 2024 The multimodal universe: enabling large-scale machine learning with 100 TB of astronomical scientific data (arXiv:2412.02527)
[107] Su W P, Schrieffer J R and Heeger A J 1979 Solitons in polyacetylene *Phys. Rev. Lett.* **42** 1698
[108] He K, Zhang X, Ren S and Sun J 2015 Deep residual learning for image recognition (arXiv:1512.03385 [cs.CV])
[109] Daumé III H 2009 Frustratingly easy domain adaptation (arXiv:0907.1815 [cs.LG])
[110] Sun B, Feng J and Saenko K 2015 Return of frustratingly easy domain adaptation (arXiv:1511.05547 [cs.CV])
[111] Sun B and Saenko K 2016 Deep coral: correlation alignment for deep domain adaptation (arXiv:1607.01719 [cs.CV])
[112] Ghorbani A, Abid A and Zou J 2019 Interpretation of neural networks is fragile *Proc. AAAI Conf. on Artificial Intelligence* vol 33 pp 3681
[113] Buckman J, Roy A, Raffel C and Goodfellow I 2018 Thermometer encoding: one hot way to resist adversarial examples *2018 Int. Conf. on Learning Representations (ICLR)* (available at: https://openreview.net/forum?id=S18Su--CW)
[114] Yao Z, Gholami A, Keutzer K and Mahoney M 2020 PyHessian: neural networks through the lens of the Hessian *2020 IEEE Int. Conf. on Big Data* pp 581–90
[115] Duan R, Mao X, Qin A K, Chen Y, Ye S, He Y and Yang Y 2021 Adversarial laser beam: effective physical-world attack to DNNs in a blink *2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 16057–66
[116] Wang Y, Sun T, Li S, Yuan X, Ni W, Hossain E and Poor H V 2023 Adversarial attacks and defenses in machine learning-powered networks: a contemporary survey (arXiv:2303.06302 [cs.LG])
[117] Dombrowski A-K, Alber M, Anders C, Ackermann M, Müller K-R and Kessel P 2019 Explanations can be manipulated and geometry is to blame *Advances in Neural Information Processing Systems* vol 32, ed H Wallach, H Larochelle, A Beygelzimer, F d Alché-Buc, E Fox and R Garnett (available at: https://proceedings.neurips.cc/paper_files/paper/2019/file/bb836c01cdc9120a9c984c525e4b1a4a-Paper.pdf)
[118] Dombrowski A-K, Anders C J, Müller K-R and Kessel P 2022 Towards robust explanations for deep neural networks *Pattern Recognit.* **121** 108194
[119] McInnes L, Healy J and Melville J 2020 UMAP: uniform manifold approximation and projection for dimension reduction (arXiv:1802.03426 [stat.ML])
[120] Sainburg T, McInnes L and Gentner T Q 2021 Parametric UMAP embeddings for representation and semisupervised learning *Neural Comput.* **33** 2881–907
[121] Cooley S M, Hamilton T, Ray J C J and Deeds E J 2019 A novel metric reveals previously unrecognized distortion in dimensionality reduction of scRNA-seq data (bioRxiv) (https://doi.org/10.1101/689851)
[122] Wattenberg M, Viégas F and Johnson I 2016 How to use t-SNE effectively *Distill* **1**
[123] Chari T and Pachter L 2023 The specious art of single-cell genomics *PLoS Comput. Biol.* **19** e1011288