Unsupervised machine learning of topological phase transitions from experimental data

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
25 January 2021

REVISED
19 April 2021

ACCEPTED FOR PUBLICATION
11 May 2021

PUBLISHED
14 July 2021

PAPER

Unsupervised machine learning of topological phase transitions from experimental data

Niklas Käming[1], Anna Dawid[2,3], Korbinian Kottmann[3], Maciej Lewenstein[3,4], Klaus Sengstock[1,5,6], Alexandre Dauphin[3] and Christof Weitenberg[1,5,*]

[1] ILP—Institut für Laserphysik, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany
[2] Faculty of Physics, University of Warsaw, Pasteura 5, 02-093 Warsaw, Poland
[3] ICFO—Institut de Ciències Fotòniques, The Barcelona Institute of Science and Technology, Av. Carl Friedrich Gauss 3, 08860 Castelldefels (Barcelona), Spain
[4] ICREA, Pg. Lluís Campanys 23, 08010 Barcelona, Spain
[5] The Hamburg Centre for Ultrafast Imaging, Luruper Chaussee 149, 22761 Hamburg, Germany
[6] ZOQ—Zentrum für Optische Quantentechnologien, Universität Hamburg, Luruper Chaussee 149, 22761 Hamburg, Germany
* Author to whom any correspondence should be addressed.

E-mail: christof.weitenberg@physnet.uni-hamburg.de

Keywords: machine learning, unsupervised learning, topological matter, Floquet systems

Abstract

Identifying phase transitions is one of the key challenges in quantum many-body physics. Recently, machine learning methods have been shown to be an alternative way of localising phase boundaries from noisy and imperfect data without the knowledge of the order parameter. Here, we apply different unsupervised machine learning techniques, including anomaly detection and influence functions, to experimental data from ultracold atoms. In this way, we obtain the topological phase diagram of the Haldane model in a completely unbiased fashion. We show that these methods can successfully be applied to experimental data at finite temperatures and to the data of Floquet systems when post-processing the data to a single micromotion phase. Our work provides a benchmark for the unsupervised detection of new exotic phases in complex many-body systems.

1. Introduction

Machine learning techniques have recently achieved remarkable successes in analysing large data sets in various areas. These developments have also led to promising applications in quantum physics [1, 2]. Examples include the efficient representation of quantum many-body states [3], efficient state tomography from restricted experimental data [4–6], the optimisation of experimental preparations [7–10] and the identification of the phases of matter [11–20]. For the latter, machine learning methods have been applied to data both from numerical simulations and from experiments such as scanning tunneling microscopy images of condensed matter systems [21, 22], neutron scattering data from spin ice systems [23], as well as real and momentum-space images of ultracold atomic systems [24–26]. When analysing experimental data, machine learning can unfold its full potential by identifying relevant information despite noise and other imperfections, such as finite temperatures or restricted access to the relevant observables. Another prospect of machine learning analysis is to identify novel phases and order parameters in exotic regimes [26, 27]. While supervised machine learning methods, i.e., with labelled training data, have been broadly applied, unsupervised methods dealing with unlabelled data have so far mainly been restricted to numerical studies [14, 18, 28–36].

Machine learning techniques can be employed for various tasks related to phase transitions, including the comparison of experimental data to competing theoretical descriptions in an unbiased way [25], the analysis of patterns in the trained filters of convolutional neural networks [26, 27], the generation of new images [37], or the extraction of physical parameters and concepts [38, 39]. Finally, the question of the interpretability of neural networks has obtained a new stimulus due to their application to physical problems [15, 27, 40–44].

Quantum simulators based on ultracold atoms in an optical lattice allow a variety of quantum many-body systems to be engineered and probed using detection methods complementary to those used with solid-state systems [45]. In particular, topological systems can be created by adding artificial gauge fields [46, 47] using periodic driving, i.e., the so-called Floquet engineering [48, 49]. The topological phases of matter are an active field of study, but the absence of a local order parameter generically poses a challenge to their detection [50]. Therefore, the classification of topological phases has, in particular, been addressed by machine learning techniques [32, 51–59]. With cold atoms, many detection methods have been demonstrated, including the transverse Hall drift [60–63], Berry phase measurements [64], quantised circular dichroism [65, 66] as well as Bloch state tomography [67–71]. The latter is based on momentum-space images following quench dynamics, which also form the basis of the machine learning analysis in this article.

Here, we apply unsupervised machine learning techniques to experimental data from the topological phases of a Haldane-like [72] model realised in ultracold atomic quantum simulators. We also address the problem of dealing with the micromotion that inherently arises in Floquet systems using machine learning for data post-processing, which allows the micromotion phase of all data to be fixed at the desired value. Fixing the micromotion phase has proved vital for the successful application of unsupervised machine learning methods, which are apparently dominated by a micromotion-induced change of graphic patterns. As expected, unsupervised machine learning is more challenging than supervised machine learning, where the identification of the phase transitions was successful, even in the presence of micromotion [24].

The unsupervised learning of phase transitions can roughly be divided into two categories: clustering-based methods [15, 16, 29, 30, 33, 56, 73–75] and learning-success-based methods [14, 18, 19, 28, 32]. In this work, we apply methods from both categories to the data, which we post-process into a single micromotion phase. Clustering-based methods identify the phases by clustering the data in a suitably chosen space and associating each cluster with a different phase, employing concepts such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), autoencoders and diffusion maps. In this category, we find that a k-means cluster analysis in the latent space of an autoencoder does identify the phase transitions, when it is separately applied to cuts through the phase diagram. This approach, however, cannot distinguish between the different signs of the Chern number. Learning success-based methods use the success of the training process for different trial classifications to judge the similarity of the data. In this context, we use anomaly detection [18] and influence functions [42]. By carefully combining the information from these techniques, we can uncover the full phase diagram from noisy experimental data in a completely unsupervised way. Our results provide an important benchmark for unsupervised machine learning of the phases of matter and evaluate methods that might be useful for revealing new exotic orders in complex systems.

The structure of this article is as follows. We start with a description of the methods used (section 2). In section 2.1, we describe the experimental setup and the protocols used to obtain the data. Section 2.2 gives an overview of the different machine learning methods we use within this work. In the results section 3.1, we first employ latent space analysis to detect the different topological phases. Afterwards, we describe how to post-process the data to a desired micromotion phase in section 3.2 and check the validity of this approach using influence functions in section 3.3. We use the post-processed data to detect the different topological phases again using the method of latent space analysis (section 3.4). Section 3.5 describes an anomaly detection scheme used to separate different topological phases. We finally use the influence functions in section 3.6 to distinguish between the two topologically non-trivial phases.

2. Methods

In the following, we start by explaining the technical details of the experimental setup. We continue by describing the machine learning methods we use within this study.

2.1. Experimental setup and data acquisition

The data are obtained in experiments performed with ultracold atoms in optical lattices [45], which are established as a very controllable system for studying solid-state physics in general and topological phases in particular [46, 47]. The topological Haldane model [72] is realised by the Floquet driving of a honeycomb lattice [62, 69, 76, 77]. In this specific configuration, the experiments start with a hexagonal lattice with a large offset $\Delta_{\mathrm{AB}} = 2\pi \cdot 6.1$ kHz between the two sublattices, realised by a suitable polarisation of the three interfering laser beams that form the optical lattice [69] (figure 1(a)). The lattice is then accelerated on elliptical trajectories by a phase modulation of the lattice beams, characterised by a shaking phase φ between the modulation in the x and y directions. The resulting effective Floquet Hamiltonian features non-trivial Chern numbers and gives rise to a topological phase diagram closely related to the original Haldane model (figure 1(d)) [24, 71]. The control parameters are the shaking phase φ, which gives rise to time-reversal

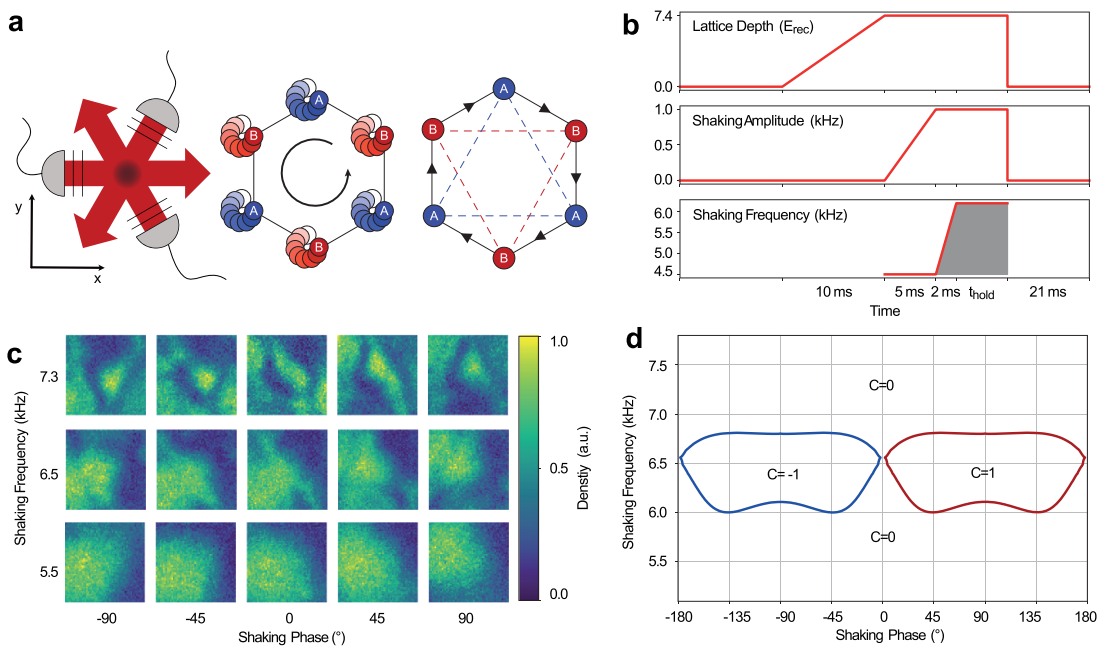

Figure 1. Experimental setup. (a) Three laser beams form a hexagonal optical lattice by interference at 120° (left). The polarisation, and thus the geometry, can be tuned via the two waveplates indicated by the black lines. The hexagonal lattice consists of A and B sites with an energy offset, Δ_{AB}. We periodically drive the lattice on an elliptical trajectory (centre) in order to obtain an effective Floquet Hamiltonian with Peierls phases on the nearest-neighbor tunneling elements (right) giving rise to topological bands. (b) The atoms are adiabatically prepared in the lowest Floquet band by ramping the lattice depth, the shaking amplitude, and the shaking frequency to different final values. Different hold times allow the sampling of different micromotion phases given by the grey area under the curve. The atoms are imaged after a time-of-flight expansion of 21 ms. (c) Typical momentum-space images for different shaking frequencies and shaking phases. The images are centred around zero momentum and have a width of one reciprocal lattice vector. (d) The topological phase diagram for the lowest band as predicted by a numerical Floquet calculation featuring the two lobes of the non-trivial Chern number, $C = \pm 1$, characteristic of the Haldane model.

symmetry breaking, and the shaking frequency f_{sh}, which gives rise to non-trivial Chern numbers $C = \pm 1$ for near-resonant shaking with the sublattice offset $f_{sh} \approx \Delta_{AB}/2\pi$.

The numerical prediction for the phase boundary (figure 1(d)) results from a Floquet calculation for a tight-binding model of the hexagonal lattice based on the shaking parameters and the calibrated parameters of the static lattice. It has been shown to agree well with previous measurements of the topological properties of the system [24, 66, 69, 71], except for a slight shift of the topological region towards higher frequencies for the experimental data. This shift may be due to the uncertainty in the calibration of the static lattice or to contributions from higher bands, which were neglected in the two-band tight-binding model. Note that the calibration uncertainty of the polarisation of the lattice beams of 0.2° leads to an uncertainty of the expected phase-transition points of around 200 Hz [24].

The experiments are performed with ultracold spin-polarised fermionic atoms of ^{40}K with a mass $m = 40$ u prepared in the lowest band of the optical lattice and formed by laser beams with a wavelength of $\lambda = 1064$ nm, as in earlier works [24, 69]. The characteristic energy scale is the recoil energy $E_{rec} = h^2/(2\,m\lambda^2)$. In the transverse direction, the cloud is weakly harmonically confined. In order to adiabatically prepare the lowest band of the Floquet system, we gradually ramp up the Floquet drive in two steps (figure 1(b)): (a) we ramp up the shaking amplitude to 1 kHz within 5 ms at the very off-resonant shaking frequency of $f_{sh}^{ini} = 4.5$ kHz, (b) we ramp the shaking frequency up to the final value f_{sh}^{fin} within $t_{ramp} = 2$ ms at a fixed shaking amplitude. This ramping protocol strategy aims to keep the bandgaps as large as possible and is well-established via earlier experimental and theoretical work [69–71, 78]. Due to Floquet heating, this procedure leads to a typical population of the lowest band of 50%–75%. Previous work on supervised machine learning has shown that the Chern number of the lowest band can be faithfully obtained, despite the non-zero temperature [24].

For the detection of the state, all potentials are switched off, leading to a free expansion of the system known as time-of-flight imaging. The expansion maps the original momentum distribution onto the real-space density, which is then imaged by absorption imaging. This procedure can be related to Bloch-state tomography [68, 69, 71], which is based on quench dynamics after projection onto a static lattice with large Δ_{AB}, realising the special case of a zero hold time in the static lattice. This connection motivates the use of

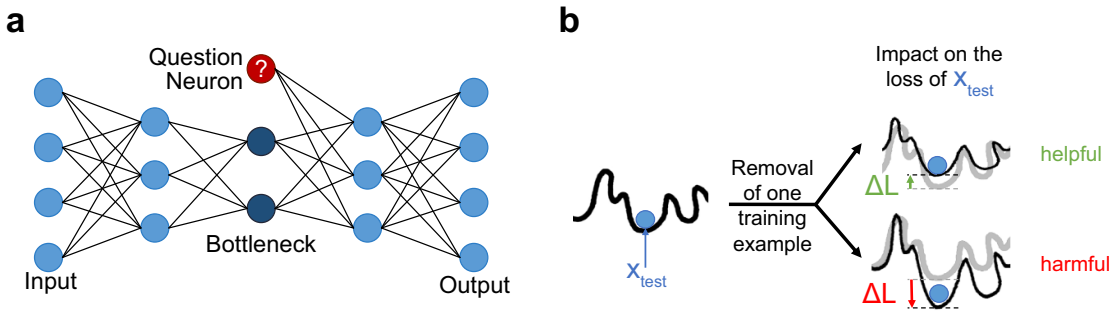

Figure 2. Schemes of the machine learning methods used within this work. (a) The architecture of an autoencoder with a question neuron. The data is encoded in the bottleneck forming the latent space. An additional question neuron can be used to feed additional data to the fully connected layers of the decoder. (b) Influence of a training example on a prediction. The sketches show the loss of a single test point, whose minimum in the space of the network parameters is marked by a circle. Consider removing a single training example from the training set and retraining the model. If the loss at the test point, X_{test}, increases (decreases), the removed training example is helpful (harmful) for making prediction atX_{test}. This procedure can be approximated with influence functions.

the experimental images for detecting topology, although proper Bloch state tomography explicitly relies on the full quench dynamics to disentangle the parameters [68].

In the experimental protocol, we hold the atoms in the Floquet system for different hold times t_{hold} at the final shaking frequency in steps smaller than the Floquet period, in order to sample different instances of the Floquet micromotion ϕ. The micromotion phase is then given by $\phi = f_{\text{sh}}^{\text{fin}}(t_{\text{ramp}}/2 + t_{\text{hold}}) + f_{\text{sh}}^{\text{ini}} t_{\text{ramp}}$. This convention traces the micromotion back to the start of the driving with a kick in a fixed direction and allows the micromotion phases of the data to be related to different shaking frequencies. Micromotion is an intrinsic property of Floquet systems, and while it can give rise to new physics [79, 80], it is often a nuisance when studying the effective Floquet Hamiltonian [78, 81].

For the analysis, we restrict the images to a square region of 56×56 pixels centred around zero momentum, $k = 0$, where 56 pixels corresponds to the length of a reciprocal lattice vector (figure 1(c)). The images are furthermore individually normalised to the interval [0, 1]. In total, we use 10 436 images with varying shaking phases, shaking frequencies, and micromotion phases with just a few images per parameter. While supervised learning often requires an additional large training data set of parameters, which allows for labelling, the unsupervised methods discussed below can identify phase transitions using data homogeneously sampled across the parameter space alone.

2.2. Machine learning methods

In the various machine learning applications of this article, we use deep neural networks (NNs) composed of combinations of fully connected and convolutional layers [82]. After each layer, the output is processed by a non-linear activation function, which in this work is mainly the so-called rectified unit function $\text{ReLU}(x) = (0 \text{ if } x \leqslant 0; x \text{ if } x \geqslant 0)$ or its variations (e.g., leaky ReLU [83]). The archetypical task of an NN is supervised learning, where the network output y_i^{out} is trained to approximate a desired label y_i for every input x_i of the dataset $\mathcal{D} = \{x, y\}$. In order to achieve this task, we define a loss function $l_i = l(y_i^{\text{out}}, y_i)$ that captures the success of this endeavor. Training then comes down to minimising the total loss function $L = 1/N \sum_{i=1}^{N} l_i$ with respect to the trainable parameters $\{\Lambda\}$ of the deep NN. The most commonly used loss functions are mean square error and binary cross-entropy. This high-dimensional optimisation problem can be tackled using gradient descent, where the parameters $\{\Lambda\}$ are iteratively shifted in the direction of the negative gradient, i.e. $\omega \to \omega - \alpha \nabla_\omega L$, where α is the so-called learning rate and a hyperparameter. Here, we mainly use more involved gradient-based optimisation strategies such as adaptive moment estimation [84] in order to speed up the training process.

In this work, we employ a special NN architecture called an autoencoder (AE) [85–87]. An AE is composed of two successive deep NNs, called the encoder and the decoder, as presented in figure 2(a). The neurons z at the output of the encoder are called the bottleneck neurons and the dimension of this so-called latent space is typically chosen to be smaller than the input space. The task of an AE is to find an efficient compression of the data input through the encoder at the bottleneck, from which the decoder is able to reproduce the original input x_i at the output stage y_i^{out}. The loss function $l(x_i, y_i^{\text{out}})$ is therefore defined between the input data and the output of the AE, and there is no need for labelled data. In this work, we found it sufficient to use the mean squared error total loss function, $L_{\text{MSE}}(x_i, y_i^{\text{out}}) = \frac{1}{N} \sum_i |x_i - y_i^{\text{out}}|^2$. AEs are typically used in the context of unsupervised learning for tasks such as dimensionality reduction or

anomaly detection. We will later use the success of this compression by looking at the loss L to differentiate between the phases of the phase diagram in section 3.5. Finally, an AE can also be trained in a supervised way, given a series of inputs associated to their corresponding outputs. Such a supervised method has applications in image denoising or colourisation [88–90].

To improve stability for data transformation, we also employ so-called variational autoencoders (VAEs) [91, 92]. The key difference with respect to the previous architecture is the introduction of an engineered regularisation at the bottleneck. Instead of encoding the input x into a feature $z(x)$ in the latent space, it is encoded into a probability distribution $p(z|x)$. A feature z is then sampled from $p(z|x)$ to be passed to the decoder. This introduces two main advantages. First, the feature space is regularised such that neuron activations at the bottleneck are more interpretable [93]. Second, in this way, we can generate new data after training by sampling at the bottleneck. However, we use the VAE here to gain more stability in the transformation of experimental data. Additionally, one can introduce a question neuron, that is, an extra input neuron that feeds directly into the bottleneck (see figure 2(a)). The information provided there can be, for example, a physical parameter corresponding to the image we provide. We will later use VAEs to post-process the data to a fixed micromotion phase.

The influence function [94, 95] is an interpretability method that can be understood as a numerically feasible approximation of leave-one-out (LOO) training, as presented in figure 2(b). LOO training consists of retraining the model after removing a single training point and checking how this changes the test loss connected to the prediction on the chosen test point. If the prediction got worse (better), i.e., the test loss got larger (smaller), the removed training point was a helpful (harmful) one. However, such an analysis with LOO training is prohibitively expensive because, for the full picture, it requires the number of training procedures to be equal to the training dataset size multiplied by the number of chosen test points. Instead, the most complicated step in the numerical approximation, i.e., influence functions, consists of a single computation of Hessian's inverse of the training loss with respect to the model parameters.

An influence function estimates the influence that a single training point has on a prediction made on a single test point. In this paper, we apply it to a convolutional neural network (CNN) trained in a supervised way. An analysis of the most influential points indicates which data features are dominant in the model's predictions. We use this property in section 3.3. Moreover, we interpret training points that are similarly influential to a particular prediction as similar. In this way, the influence functions' values, $\mathcal{I}$, provide the notion of similarity found by a network trained in a given problem. Thanks to this property, they allow distinctive similarity regions to be found within the same class [42], which indicates improper labelling and proves useful in section 3.6. Analysis of which training points the model regards as similar and which data features are dominant in the model's predictions increases the model's interpretability. The influence function values, $\mathcal{I}(x_{\text{train}}, x_{\text{test}})$, can only be compared for fixed test points and various training points within the same model. Therefore, we need to fix a test point whenever we calculate a set of $\mathcal{I}$ for similarity analysis.

All the machine learning techniques were implemented using NumPy, PyTorch, and Tensorflow [96–98]. The specifics of the architectures with reproducible code for all performed tasks can be found in our notebooks [99].

3. Results

3.1. Latent-space interpretation of AEs

As a first step, we produce and analyse a low-dimensional representation of the data in the latent space of an AE formed by the activations of the bottleneck neurons. AEs are important tools for unsupervised learning [100]. An AE consists of several convolutional layers and a fully connected bottleneck formed by two neurons (figure 3(a)). We choose a 2D latent space to create an easy-to-understand visual representation of the given samples. The complete implementation details can be found in our notebooks [99]. We checked that choosing more dimensions in the latent space does not lead to an improvement. The AE is trained on the complete dataset.

The two-dimensional latent-space representation of all images yields a dense cloud of data points without any apparent clustering (figure 3(b)). The picture becomes clearer when we restrict the data to fixed shaking phases, i.e., vertical cuts through the phase diagram (figures 3(c) and (d)). The data then lie on elliptical structures whose radii are related to the shaking frequency. For further analysis, we fit an ellipse using direct least-squares fitting [101] and perform a coordinate transformation to extract the elliptical coordinate radius r and azimuthal angle θ measured from the major axis of the fitted ellipse.

The azimuthal angle can be clearly connected to the micromotion phase, showing a linear dependence (figure 3(e)) for a shaking phase of $\varphi = 90°$. The same dependence can also be seen with the azimuthal coordinate of the centre of mass of the raw images, which provides a direct connection between the time-of-flight images and the latent space. See appendix A for further details. We furthermore explore

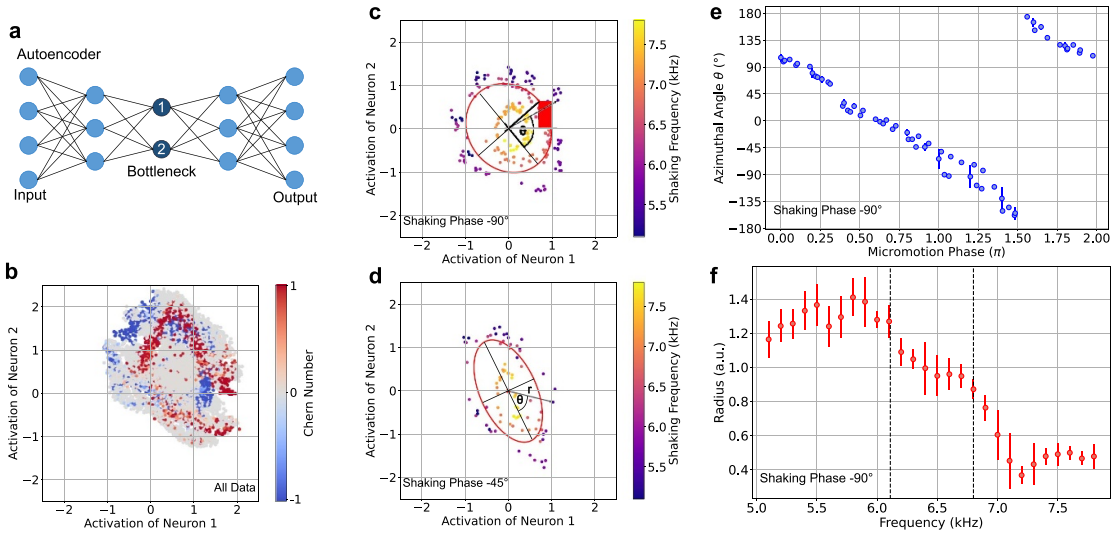

Figure 3. Bottleneck analysis. (a) Sketch of the AE with two bottleneck neurons, which is trained to reproduce the entire dataset. (b) Activations of the two bottleneck neurons form the latent space, in which each point corresponds to one image. In the scatter plot of all data, the Chern number is colour coded according to the numerics, illustrating that the data does not cluster according to the Chern number in latent space. (c), (d) Analysis of single cuts through the phase diagram, i.e., of data with fixed shaking phases of (c) $\varphi = -90°$ and (d) $\varphi = -45°$. The data form rings in latent space and are fitted by an ellipse (red line), which forms a new coordinate system with an azimuthal angle θ measured relative to the longer half axis and a radius r, as indicated by an arrow. (e), (f) Analysis of latent space in elliptical coordinates. (e) The azimuthal angle θ is linearly related to the micromotion phase for $\varphi = 90°$ of the individual images and is independent of their shaking frequency. (f) The mean radial coordinate for a given shaking frequency traces out a monotonously decreasing curve with no clear signature of a phase transition and displays three plateaus in accordance with the phase boundaries. The error bar is the standard deviation taken by averaging over the images with a given shaking frequency. The plots for other shaking frequencies look similar. This association with micromotion means that latent space can be interpreted, but also that micromotion is the dominant feature hiding the possible signatures of topological phase transitions.

possible information hidden in the radial coordinate (figure 3(f)). The mean radius decreases with shaking frequency with some signs of plateaus, but without a sufficiently clear separation as the shaking frequency changes to make a prediction of phase transitions. The latent space representation can thus be physically interpreted via the micromotion, but it cannot provide an identification of the topological phases. We attribute this to the dominance of the micromotion, which we try to eliminate in the following section 3.2.

3.2. Data post-processing to the desired micromotion phase

In Floquet systems, micromotion poses an additional challenge for identifying phase transitions. We find that all attempts to apply unsupervised machine learning methods fail for data with varying micromotion phases. In contrast, as we show in this article, unsupervised machine learning is successful if all images have the same micromotion phase, i.e., the centre-of-mass displacement is in the same direction. Changes in the micromotion phase induce different graphical patterns, including the moving centre of mass, which seems to dominate the models' predictions. The studied experimental data were obtained by sampling various micromotion phases. Therefore, we need to post-process them to the desired single value of the micromotion phase. We show that this can be accomplished by machine learning techniques based on VAEs, which are powerful tools for data transformation and generation [91, 92]. Our architecture uses an additional question neuron in the bottleneck, as sketched in figure 2(a), which has previously been proven successful in identifying relevant physical properties [39]. Here, we use an additional question neuron for supervised training of the VAE. The challenge addressed here is related to tasks such as fringe removal in absorption imaging [102], or the removal of timing jitter in pump-probe experiments [103], but we believe that our method based on VAE is very broadly applicable to post-processing to obtain the desired sampling in different experimental scenarios.

The encoder of the VAE consists of several convolutional stages and several layers of fully connected neurons. The last layer of the encoder has 26 fully connected neurons, thus the latent space covers 13 uncorrelated Gaussian distributions, each defined by its mean and log variance. The decoder also has several fully connected layers followed by a few transposed convolutional stages. The first fully connected layer is also attached to the input of the question neuron. In total, the AE has over three million trainable parameters. The complete implementation details can be found in our notebooks [99]. To optimise the hyperparameters of our AE, we use the hyperparameter optimisation library Optuna [104] and test over 60 000 different network architectures. To identify the best working network, we use the structural similarity

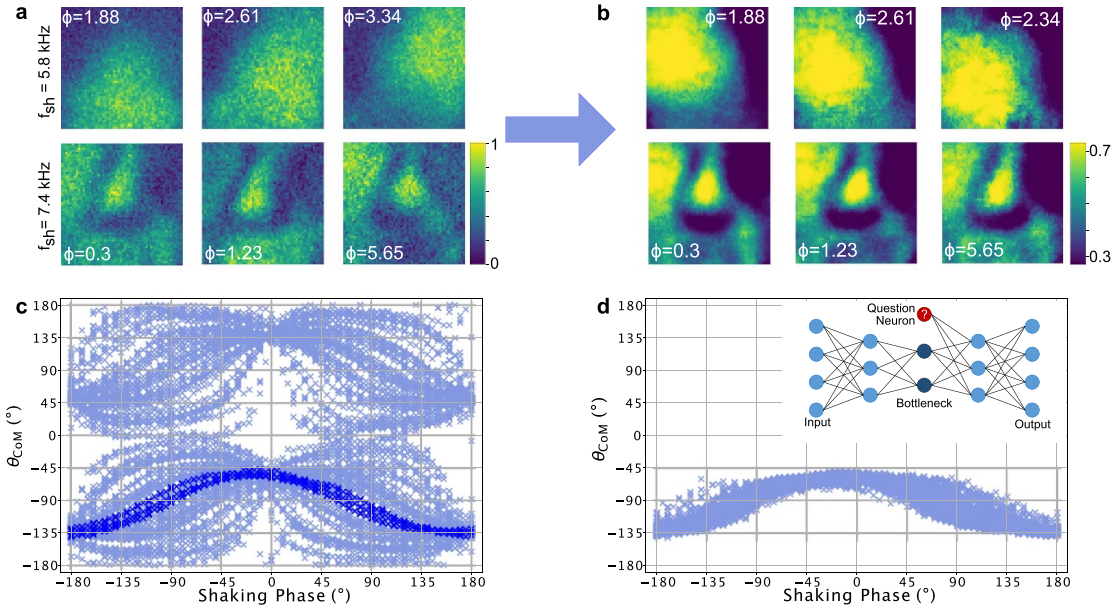

Figure 4. Post-processing data to the desired micromotion phase using a VAE with an additional question neuron. (a) Example images at shaking frequencies of 5.8 kHz and 7.4 kHz and a shaking phase $\varphi = 90°$ for different micromotion phases ϕ illustrating the motion of the centre of mass with the micromotion phase. (b) The same example images as in (a), rephased by the variational autoencoder to a micromotion phase of $\phi_0 = 0.0$ show a centre-of-mass position that is only weakly dependent on the original micromotion phase ϕ. (c) The azimuthal coordinate of the centre of mass of the original images as a function of the shaking phase shows a complicated dependence. (d) The azimuthal coordinate of the centre of mass (θ_{CoM}) of all images forms a narrow band and is only dependent on the shaking phase, but not the original micromotion phase. This dependence on the shaking phase is similar to the dependence of the data for the actual micromotion phase $\phi = 0.0$ before rephasing (dark blue points in (c)). The comparison of the distribution of azimuthal coordinates of the centre of mass in (c) and (d) illustrates that the rephasing of the micromotion was successful. The micromotion phase of the processed data $\phi_0 = 0.0$ is controlled by an additional question neuron in the bottleneck of the VAE (inset in (d)).

index [105] as a measure of performance. For each point in the phase diagram with a fixed shaking frequency and shaking phase, we took several images with different micromotion phases by varying the hold time. Taking these images as a new dataset, we select all combinations and permutations of images for a fixed shaking frequency and shaking phase and calculate their micromotion phase difference $\Delta\phi = \phi_{\mathrm{output}} - \phi_{\mathrm{input}}$. Thus, the dataset includes 63 050 image pairs with a given $\Delta\phi$. Therefore, in contrast to the other AEs we employed in the context of this paper, the input and output are different for the VAE. For validation purposes, we randomly choose 10% of the dataset and hide it from the network during training. We train with one image as the input, which we refer to as the input image, and one image with the same shaking frequency and shaking phase but a different micromotion phase as the output; the micromotion phase difference is the input for the question neuron. Samples of the original images are given in figure 4(a). After training, we use the VAE to transform all the original images in the dataset to a micromotion phase of $\phi_0 = 0.0$ by choosing their corresponding micromotion phase, but with an opposite sign, as the input for the question neuron. The post-processed images with a single micromotion phase are similar to the original data, except for some noise removal and a squeezing of the distribution of pixel values to a range of 0.3–0.7, which we attribute to the non-linear activations in the network (figure 4(b)).

Because the micromotion is directly related to the centre of mass of the images, one can see the success of the post-processing in the example images in figure 4(b): after the micromotion is set to a fixed value, the images with different micromotion phases look very similar and have the same direction of displacement of the centre of mass. This is further confirmed by a comparison of the distributions of the azimuthal centre-of-mass coordinates for all images before and after rephasing (figures 4(c) and (d)). The highlighted blue data in (figure 4(c)) are the centres of mass for the micromotion phase $\phi = 0.0$. The narrow distribution in (figure 4(d)) shows that the rephasing was successful. For the identification of phase transitions using unsupervised machine learning discussed below, it is important that we use the post-processed data with a single micromotion phase.

3.3. Confirming the post-processing to a desired micromotion phase with influence functions

To provide further evidence that the post-processing described in section 3.2 was successful, we confirm it using the influence functions introduced in section 2.2. Influence functions provide an interpretation of the

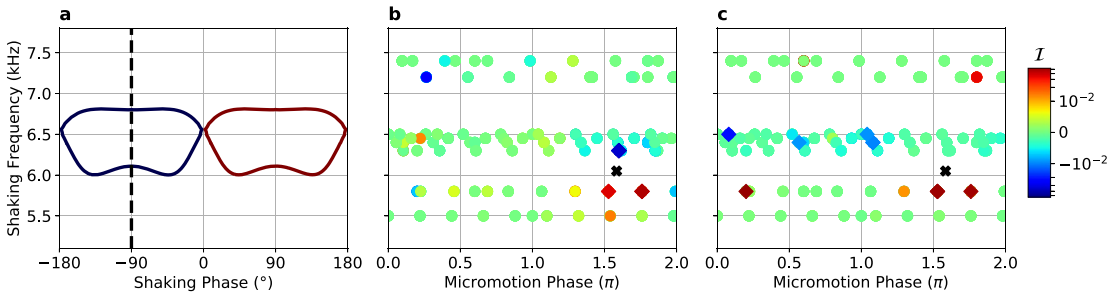

Figure 5. Confirming the post-processing to a desired micromotion phase with influence functions. Panel (a) presents the phase diagram with theoretical boundaries between phases and with the cut at a shaking phase of $-90°$ marked with a black line, which is analysed on the right. Panels (b) and (c) show the colour-coded influence function values for all the training points marked with dots and for the same test point (marked with a black cross) for a single cut at a phase of $-90°$ across the shaking frequency before and after post-processing. The five most helpful (harmful) training points are marked with red (blue) diamonds and they can overlap. (b) The most influential (both helpful and harmful) training points are the points with the shaking frequency and the micromotion phase closest to the test point. (c) After fixing the micromotion phase, the most influential points are distributed across various original micromotion phase values, indicating the successful removal of this property, which now is ignored by the CNN.

machine learning model, by indicating which training points are influential for a chosen prediction. Analysis of the most influential examples can reveal the characteristics that impact the machine learning predictions.

First, we train a convolutional neural network (CNN) in a supervised way to classify original images with various micromotion phases. Instead of analyzing the whole 2D diagram, we only consider a single cut at the fixed shaking phase of $-90°$, which simplifies the visualisation of the results without changing them. Within this single cut, the Chern number of the system changes from 0 to -1 and back to 0 with an increase in the shaking frequency. The labelled training data then only contain two phases ($C = 0$ and $C = -1$). To avoid the influence of experimental imperfections, we exclude data close to the theoretically predicted phase transitions. Using the trained CNN, we calculate the influence functions determining how influential the whole training dataset is for the prediction chosen to be in the transition region. The results are presented in figures 5(b) and (c); the black cross indicates the test point, and dots represent the training data and their colour-coded influence function values. The colours vary from red, for helpful training points, through green for the least influential (ignored), to blue for harmful.

We see in panel (b) that the most influential (both helpful and harmful) data for the chosen prediction are those with both the most similar shaking frequency and micromotion phase. Learning the shaking frequency is to be expected, as it is the parameter governing the phase transition. However, while the CNN also regards the micromotion phase as influential when making a prediction, we know that this property is physically irrelevant for the transition. The micromotion phase is an intrinsic property of the Floquet realisation of the topological Hamiltonian, but does not change the topology of the effective Floquet Hamiltonian.

We perform an analogous analysis for the post-processed training data, i.e., with the micromotion phase removed. Panel (c) in figure 5 shows that the most influential points are now randomly distributed along the original micromotion phase axis. This tells us that the CNN no longer sees this parameter as influential and confirms that the data were successfully post-processed to a constant micromotion phase.

We note that when training on data with or without the micromotion, in both cases, the validation and test accuracy of the trained CNN are similar. Interestingly, this means that in this setup, the predictive power of the network is not impacted by learning a quantity which is physically irrelevant.

3.4. Clustering in latent space

In the following sections, we apply different unsupervised machine learning methods to the post-processed data with a constant micromotion phase and compare their performance. We start with a method based on the clustering category, which identifies clusters in some low-dimensional representation of the data as different phases. Specifically, we employ the same convolutional AE as in section 3.1, but it is now applied to the post-processed data.

The latent-space representation of the complete dataset is shown in figure 6(b). The data are colour encoded according to the theoretical predictions for the Chern number. It appears that different topological classes tend to form ring structures in the 2D latent space. As in section 3.1, we now restrict ourselves to single shaking-phase cuts. Figures 6(c) and (d) show the latent spaces for two such cuts, and we observe three main clusters related to three frequency regimes.

By choosing k-means clustering, a standard method for solving clustering problems, it is possible to automate the clustering process of the different latent space representations of the observed image data. We

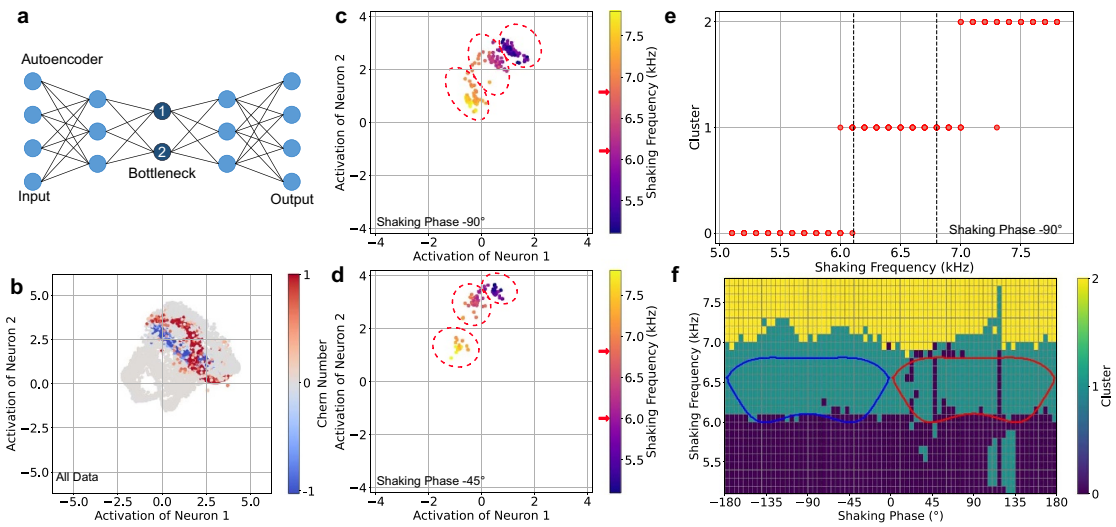

Figure 6. Bottleneck analysis of rephased data. The bottleneck analysis with a simple AE (sketched in (a)) for the data after rephasing to sampling of a fixed micromotion phase. (c), (d) The analysis of single cuts through the phase diagram, i.e. of data with a fixed shaking phase of $\varphi = -90°$ in (b) and $\varphi = -45°$ in (c), shows significant clustering according to the shaking frequency (colour coded). The dotted lines guide the eye to separate the different clusters found by the k-means algorithm. The arrows next to the colour bar mark the theoretically expected phase transitions. (e) The cluster analysis of the latent space. A k-means clustering analysis for $k = 3$ clusters shows that the data are indeed clustered according to the shaking frequency. There are several values for each shaking frequency for the different original micromotion phases, which mostly lie on top of each other in the plot. (f) Ordering the three clusters with increasing frequency and combining the data from all cuts yields a topological phase diagram, which is in reasonable agreement with the numerical prediction (blue and red lines), but does not distinguish between $C = 1$ and $C = -1$.

use the k-means functionality of Scikit-learn [106] and set the number of clusters to three and the maximum number of iterations to 500. All other parameters are set to their standard values according to the documentation. We tried different random seeds to prove stability. The results of the cluster analysis are shown in figures 6(e) and (f).

This allows one to predict phase boundaries in good agreement with the theoretical predictions given by the dashed lines. The slight shift to higher frequencies is in accordance with all other methods and is explained in section 2.1. Evaluating the data for all shaking phases allows reconstructing the complete 2D Haldane phase diagram shown in figure 6(f).

The procedure of separately analysing vertical cuts through the phase diagram is fundamentally unable to distinguish between the $C = 1$ and $C = -1$ phases for positive and negative shaking phases. Furthermore, a similar analysis of the horizontal cuts along constant shaking frequencies throughout the phase diagram does not produce clustering. Therefore, further methods are required to fully identify the topological phases.

3.5. Anomaly detection scheme

We followed the approach outlined in [18] and use an unsupervised learning scheme called anomaly detection to map out the phase diagram in a few training iterations. We use a convolutional AE, similar to the network described in section 3.2. The network consists of an encoder and decoder, each made of two convolutional layers, with a fully connected bottleneck of 50 units. For full details of the model and the process, we refer the interested reader to [99], where all the steps described in this section are exactly reproduced. The idea is as follows: We start by defining a region of the phase diagram in which we trained the AE to encode and decode the images with a low mean-squared error $L_{\mathrm{MSE}}(A^{\mathrm{in}}, A^{\mathrm{out}})$ between the input and output images $A^{\mathrm{in}}, A^{\mathrm{out}} \in \mathbb{R}^{56\times 56}$. The network learns the characteristic features of the phase that the images were taken from, and fails to reproduce images from the other phases. By looking at the loss for all images after training in only part of the phase diagram, we distinguish between the phase it has been trained on and the remaining phases via different plateaus of the loss function. Furthermore, by fitting a sum of tanh functions to the loss curve as a function of the shaking frequency, we obtain a phase boundary. We then repeat this process by training in the region of high loss from the previous training round until we find all boundaries.

We show this process in figure 7, where we start by training in the low-frequency regime (Training 1). We use all values of the shaking phase from $[-180°, 180°]$ and small shaking frequencies of up to 5.5 kHz, as indicated by the cyan rectangle. We identify three different plateaus in the loss, between which we obtain two boundaries. As seen in panel 1(b), where we take a single cut of the phase diagram at $-90°$, the different

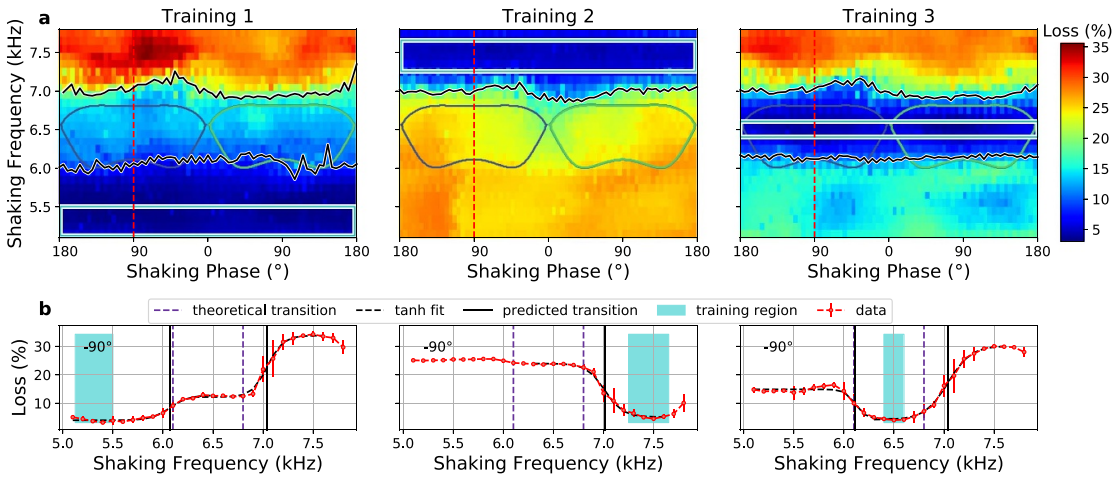

Figure 7. Anomaly detection scheme: we start by training in the topologically trivial phase (Training 1). Due to the symmetry in the shaking phase, we use all angles from $[-180°, 180°]$ and small frequencies up to 5.5 kHz, as indicated by the light blue box in (a). Panel (b) shows a single cut at the shaking phase of $-90°$, where we can see two plateaus which we identify with the topological non-trivial and trivial phases, respectively. Training 2: we continue by training in the region of highest loss in the first iteration for high frequencies. From these two training iterations, we can already narrow down the two boundaries in the phase diagram. Training 3 completes the overall picture and confirms the phases mapped in Training 1 and 2.

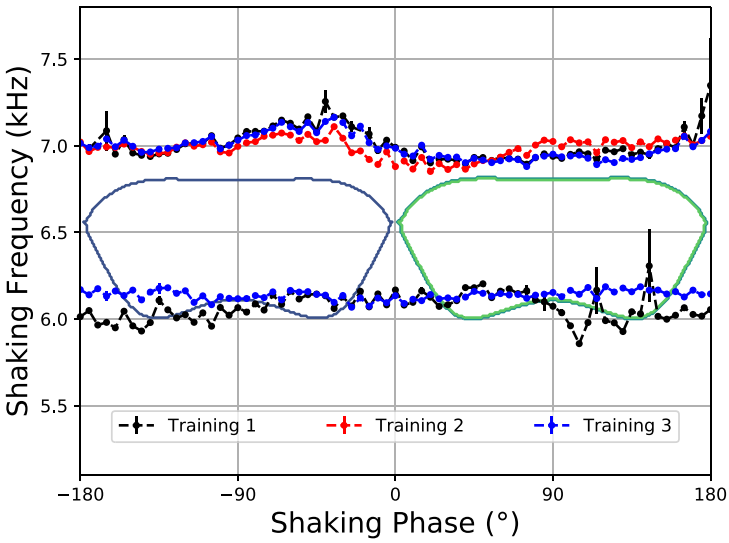

Figure 8. Phase boundaries drawn from four training iterations of anomaly detection in figure 7, separating three different phases, which we identify with the topologically trivial and non-trivial phases. The boundaries obtained from the different training iterations are consistent with each other.

phases make up plateaus. We therefore fit a tanh function in the relevant parameter region to estimate the boundaries, as indicated by the grey lines.

We continue the process and train in the high-frequency regime, where the first iterations yielded the highest loss. As seen in Training 2 (figure 7), we find a reverse picture with a clear boundary slightly above the theoretically predicted transition. This boundary from Training 2 nicely coincides with the second boundary from Training 1. To complete the picture, we also train in the intermediate-frequency regime, which yielded a higher loss in the previous training iterations. Here, the training region is significantly smaller, yet the results still match nicely with the previous training iterations. We stress that the three training procedures are independent. Therefore, the good agreement of the phase boundaries in figure 7 is a strong indication of the method's validity. Note also that generally, the results are robust concerning the size of the training region in frequency. We show this in figure 12 in appendix C. With this method, the images provide sufficient information to separate the different phases and map out the phase diagram. We present the predicted diagram in figure 8. We note that with this method, it is not possible to differentiate the non-trivial topological phases with Chern numbers of 1 and -1, because the trained compression does not generalise well in the shaking phase parameter. We provide further details in appendix C. In the following section, we

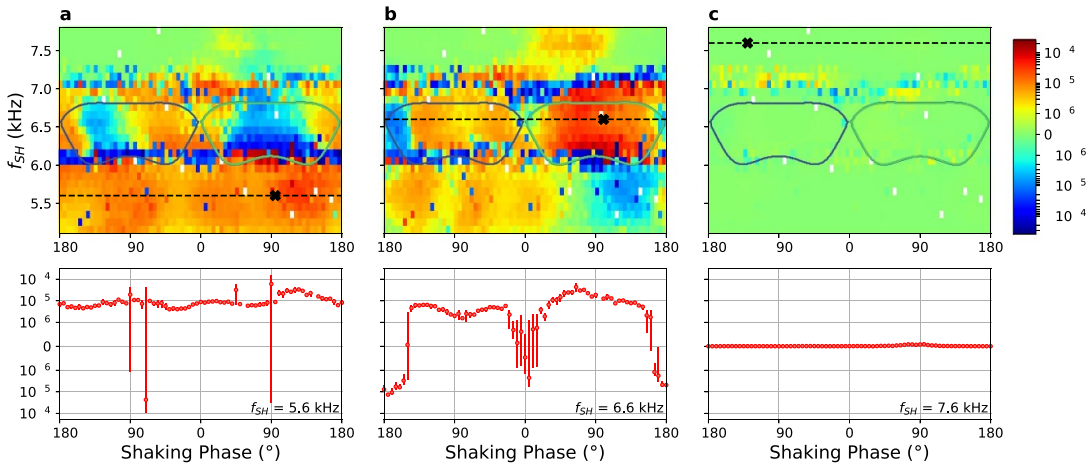

Figure 9. Analysis of data similarity within three anomaly-detected phases with influence functions. The upper row shows the phase diagrams with the colour-coded influence function values for all training data and the prediction for the test point marked with a black cross. The test points belong to (a) the low-frequency, (b) intermediate-frequency, and (c) high-frequency phases. The lower row presents the influence function values from the single cut through a corresponding phase diagram above at the fixed shaking frequency, f_{sh}. Panels (a) and (c) show quite uniform similarity patterns, while (b) suggests the existence of an additional phase. Note the use of a symmetric logarithmic scale. We mark theoretically predicted boundaries with continuous blue and green lines. White pixels correspond to a lack of available training data.

overcome this shortcoming and complete the phase diagram, i.e., we separate the two topological regions by employing the influence functions. We observe that the transition between intermediate and high frequencies for all three training rounds is slightly above the theoretically predicted transition, which is due to a mismatch between theory and experiment, as discussed in section 2.1.

3.6. Analysis of data similarity within three phases with influence functions

After obtaining the phase boundaries from the anomaly detection scheme, as described in the previous section, we analyse how similar the data are within the three distinguished phases. Such an analysis can not only confirm the predictions of unsupervised machine learning schemes but also reveal the existence of additional phase transitions. To this end, we train a CNN on the post-processed experimental data, i.e., with a single micromotion phase, with labels assigned by the anomaly detection scheme. Therefore, we have three labels corresponding to the low-frequency, topological, and high-frequency phases. We employ influence functions, described in section 2.2, to analyse which training data are influential for a given prediction. Similarly influential training data$\{x_{\mathrm{train}}\}$, with similar influence function values, $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ for a particular test point, x_{test}, can then be interpreted as similar from the machine learning model's point of view for a given problem.

To analyse the similarity of the training data, we need to compare $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ and therefore fix the test point for which $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ is calculated. In figure 9, we plot three sets of $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ calculated for all training data and three different test points: one is located in the low-frequency regime (panel (a)), the second in the intermediate-frequency regime (panel (b)), and the final one in the high-frequency regime (panel (c)). Each element of the phase diagrams in the upper row indicates a colour-coded $\mathcal{I}$ value for the corresponding test point marked with a black cross. If an element corresponds to more than one training point (if more measurements are performed for a given frequency and shaking phase), we then plot the mean of $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$. The red (blue) colour indicates the most helpful (harmful) training points for a given prediction. White dots correspond to the lack of available training data. The lower row of figure 9 contains the mean of $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ for a single cut across the phase diagram for a fixed shaking frequency, f_{sh}. The error bars represent the standard deviation, which is non-zero for all shaking phases for which multiple measurements were taken.

In panel (a), we see that the low-frequency training data are all quite similarly influential for the model, while predicting that the black cross test point belongs to the same low-frequency phase. The uniformity in question is also clearly visible in the cut through the phase diagram in the lower panel of figure 9. Apart from a single $\mathcal{I}(x_{\mathrm{train}}, x_{\mathrm{test}})$ with a large variation, indicating experimental outliers in the training set, the similarity pattern formed is quite uniform. Note, however, the symmetric logarithmic scale for $\mathcal{I}$. When ignoring outliers, the $\mathcal{I}$ values span almost one order of magnitude. The lowest $\mathcal{I}$ values of around 5×10^{-6} are located in the negative shaking phase, and the largest $\mathcal{I}$ values of around 3×10^{-5} are for training points which have a positive shaking phase similar to x_{test}. This tells us that the shaking phase is an influential factor

in the predictions for the low-frequency phase. However, it is not a determining one. Otherwise, the largest $\mathcal{I}$ values would be much more localised along the shaking phase axis. We also note that the $\mathcal{I}$ values always highlight the boundaries between phases, for two reasons. First, the data around the phase transitions are usually the most confusing for the model. They are labelled as belonging to either of the phases, being, at the same time, non-representative of any phase. The second reason is of a purely numerical nature. Regardless of whether boundaries are placed in accordance with the physical ones, the data around the boundaries play a unique role in the training, as they contain the most important information for the model. In general, we expect that the confusing phase transition regions, indicated by large $\mathcal{I}$ values in the experimental data should be broader, compared to those of numerical studies [42]. This is due to the fact that the experimental system is finite and inhomogeneous, and therefore, the phase transition is intrinsically broadened.

Panel (c) shows even more uniform behaviour. It contains $\mathcal{I}$ values for the test point localised in the high-frequency regime. What may seem surprising is that almost all $\mathcal{I}$ values are practically zero. This means that none of the training points is of significant influence when making the chosen prediction. This is because the prediction at the test point from panel (c) has an extremely high certainty, which has an impact on the $\mathcal{I}$ values. In fact, the $|\mathcal{I}(x_{\text{train}}, x_{\text{test}})|$ values are proportional to the uncertainty of the prediction made on x_{test}. When the prediction's uncertainty is very low, the $\mathcal{I}$ values are also minimal.

Panel (b) is analogous to previous panels, but this time, the test point for which the $\mathcal{I}$ values are computed is localised in the intermediate-frequency regime (which we know contains two topological phases). A striking feature of panel (b) is the lack of uniformity in the intermediate-frequency regime, which is clearly visible in the lower plot of panel (b), which contains $\mathcal{I}$ values for the single cut through the phase diagram for a fixed frequency of 6.6 kHz. In between the two plateaus, i.e., around the shaking phases of 0 and 180°, there are significant dips in the influence functions' values, which reach negative values. They show that the training data in this part of the diagram are different enough to be harmful to the analysed prediction.

While they are quite different, they are misleading for the model, as they are labelled as being the same (as belonging to the topological phase). This is analogous to the reason why influence functions always highlight boundaries between phases. This leads us to the conclusion that within the anomaly-detected intermediate-frequency regime, there is an additional boundary separating two more phases, which we know to be the $C = +1$ and $C = -1$ phases. Another observation supporting this conclusion is that there are two similarity plateaus on the negative and positive sides of the shaking phase, separated by the detected boundary. They are clearly visible in the lower plot of panel (b). The average values of the two plateaus differ by almost an order of magnitude, indicating two distinct patterns. Simultaneously, these patterns are more similar to each other than to the low- or high-frequency phases, which suggests the similar character of the two phases detected in the intermediate-frequency regime.

The similarity analysis described above reveals the existence of two phases within the anomaly-detected topological phase. We note that this analysis is vastly simplified by removing the micromotion phase from the time-of-flight images. The results from section 3.3 show that the micromotion phase was a very influential factor for the trained CNN before post-processing. Therefore, the analysis would need to include the impact of the micromotion phase on the CNN's predictions.

4. Conclusions

In this article, we have applied different unsupervised machine learning methods to identify topological phase transitions in experimental data of a Haldane-like model realised with ultracold atoms. The topological phase diagram of the elliptically shaken hexagonal lattice hosts topologically non-trivial phases at an intermediate shaking frequency and trivial phases for both low and high shaking frequencies. Furthermore, the sign of the Chern number changes with the sign of the shaking phase, i.e., the orientation of shaking, giving rise to two distinct non-trivial phases.

A necessary step for successful unsupervised learning is to fix the micromotion phase inherent to the Floquet realisation of the topological phases via a VAE with a question neuron. This post-processing of the experimental data to the desired sampling demonstrated here is an exciting tool in its own right. When generalising such a post-processing procedure to data sets, where known physical observables are invariant under the parameter which is removed, it would be interesting to test whether these observables stay the same when evaluated using the pre- and post-processed images.

Both a clustering analysis in an appropriate low-dimensional representation of the data and anomaly detection in the loss function correctly identified the three regions as a function of the shaking frequency. The correct identification of the two regions with opposite signs of the Chern number was only possible by combining this information with insights from an influence function applied to the supervised training of the incomplete phase diagram. In total, the full phase diagram, which can also be identified via supervised

machine learning on labelled data, can be obtained in a fully unsupervised way by combining the different methods.

The successful identification of the phase diagram demonstrates that unsupervised machine learning can correctly identify phases, even for noisy data, and despite the finite temperature of the system. In the future, these methods can be applied to strongly correlated systems to determine corrections to numerical predictions or to exotic quantum many-body systems with unknown phase diagrams or hidden orders [26, 27] to support the interpretation of the data and to guide the experimental exploration of the parameter space.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.4700379.

Acknowledgments

We thank Gorka Muñoz-Gil, Gabriel Fernández-Fernández, Patrick Huembeli, and Ludwig Mathey for insightful discussions and Benno Rem, Matthias Tarnowski, and Luca Asteria for providing the experimental data. The team in Hamburg thanks PHYSnet for providing computational resources.

The work in Hamburg was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) via Research Unit FOR 2414 under project number 277974659 and via the Cluster of Excellence 'CUI: Advanced Imaging of Matter'—EXC 2056—under project number 390715994.

ICFO group acknowledges support from ERC AdG NOQIA, Agencia Estatal de Investigación ("Severo Ochoa" Center of Excellence CEX2019-000910-S, Plan National FIDEUA PID2019-106901GB-I00/10.13039 / 501100011033, FPI), Fundació Privada Cellex, Fundació Mir-Puig, and from Generalitat de Catalunya (AGAUR Grant No. 2017 SGR 1341, CERCA program, QuantumCAT _U16-011424 , co-funded by ERDF Operational Program of Catalonia 2014-2020), MINECO-EU QUANTERA MAQS (funded by State Research Agency (AEI) PCI2019-111828-2 / 10.13039/501100011033), EU Horizon 2020 FET-OPEN OPTOLogic (Grant No. 899794), and the National Science Centre, Poland-Symfonia Grant No. 2016/20/W/ST4/00314, Marie Sklodowska-Curie grant STRETCH No. 101029393.

An D acknowledges the financial support from the National Science Centre, Poland, within the Preludium Grant No. 2019/33/N/ST2/03123 and the Etiuda Grant No. 2020/36/T/ST2/00588 as well as the Foundation for Polish Science within the First Team programme co-financed by the EU Regional Development Fund. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 713729 (K K). Al D acknowledges the financial support from a fellowship granted by la Caixa Foundation (ID 100010434, fellowship code LCF/BQ/PR20/11770012).

Author Contributions

N K, An D, and K K performed the numerical calculation and evaluation of the different machine learning architectures and analysed the data. N K performed the bottleneck analysis of the AE and the data post-processing via the VAE. An D performed the confirmation of the data post-processing and the identification of the different topological phases via the influence function. K K performed the unsupervised identification of phases via anomaly detection. N K, An D, and K K contributed equally to this work. Al D and M L supervised the work at ICFO (An D and K K), C W and K S supervised the work in Hamburg (N K). All authors contributed to the interpretation of the results and to the writing of the manuscript.

Appendix A. Centre of mass and micromotion

In figure 10, we show the dependence of the azimuthal coordinate of the centre of mass on the micromotion phase. For circular shaking, i.e. a shaking phase of $\pm 90°$, the centre of mass moves in a circular fashion yielding a linear dependence between the azimuthal coordinate of the centre of mass and the micromotion phase. For linear shaking, i.e. a shaking phase of $0°$ and $180°$, the centre of mass moves along a diagonal line yielding a constant azimuthal coordinate of the centre of mass at $\pm 45°$, with a phase jump of $180°$ due to small disturbances in the experiment, forcing an exchange of the long and short axes in the fitting scheme of the ellipse. Other shaking phases interpolate between these two behaviours. In conclusion, the movement of the centre of mass of the momentum distribution follows the shaking trajectories, as expected.

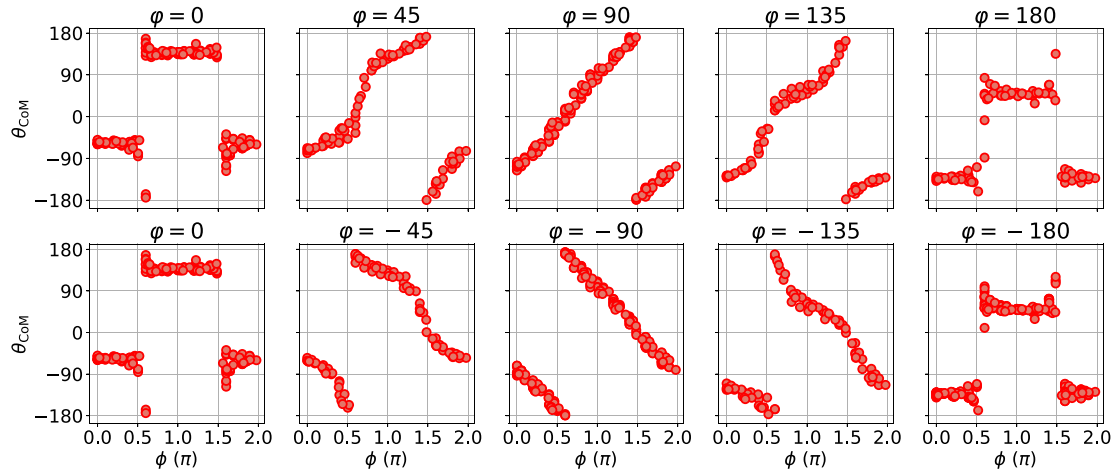

Figure 10. Dependence of the azimuthal coordinate θ_{CoM} of the centre of mass and the micromotion phase ϕ. The dependence can be explained by the elliptical shaking. For the shaking phases $\varphi = 0, \pm 180$, the shaking is linear, thus the cloud can only be displaced in k-space in the direction of the shaking. For a shaking phase of $\varphi = \pm 90$, the shaking is circular and thus the dependence is linear. The sign of φ decides the direction of shaking, which is encoded in the phase jump and the direction of the slope.

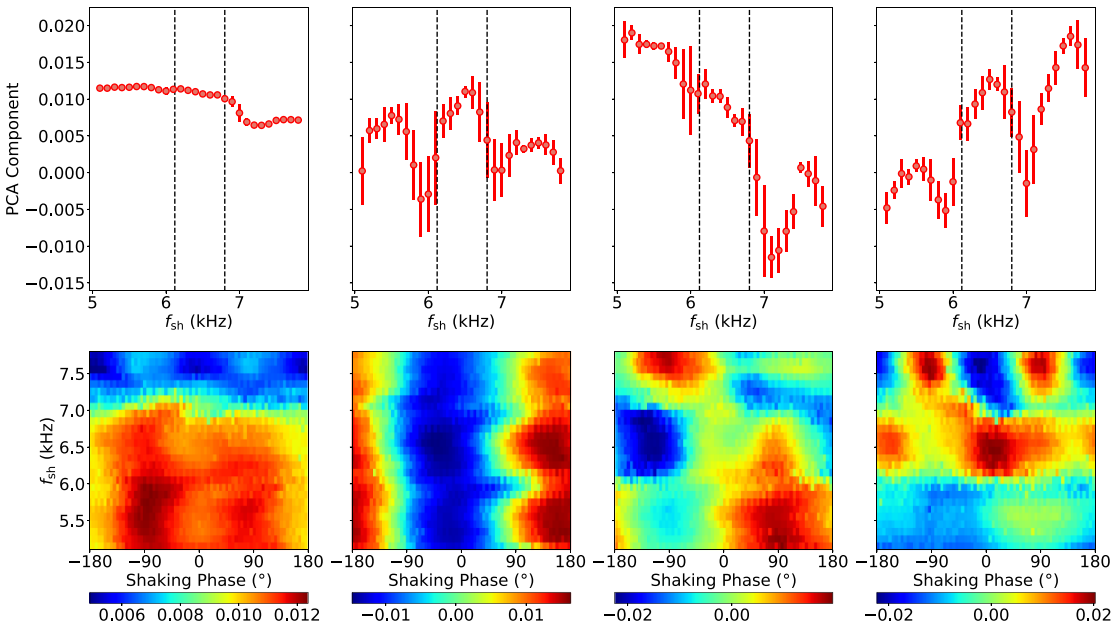

Figure 11. PCA analysis of the ToF images rephased to a fixed micromotion phase. From left to right: PCA components 1, 2, 6, and 8 which have been arbitrarily selected. To plot all other components, see the notebooks in [99]. In the upper row, a cut along a shaking phase of 90° is plotted. The error is the standard deviation of the different values for the component for the given shaking frequency. The theoretical predictions for the phase transitions are given by dashed lines. In the lower row, the averaged components are plotted in the Haldane phase diagram fashion.

Appendix B. PCA analysis

In addition to the methods in the main text, we also used principal component analysis on the processed data. In figure 11, we selected four components of this analysis. The different principle components clearly show features that correspond well with the theoretical predictions, such as sharp local minima at the expected phase transitions. However, the data also show features that are not related to phase transitions, such as a strong dependence on the shaking phase in the trivial regions. Therefore, the data do not provide a clear recipe for identifying the topological phase transitions in a completely unsupervised way. This is particularly true for the choice of the components to be analysed.

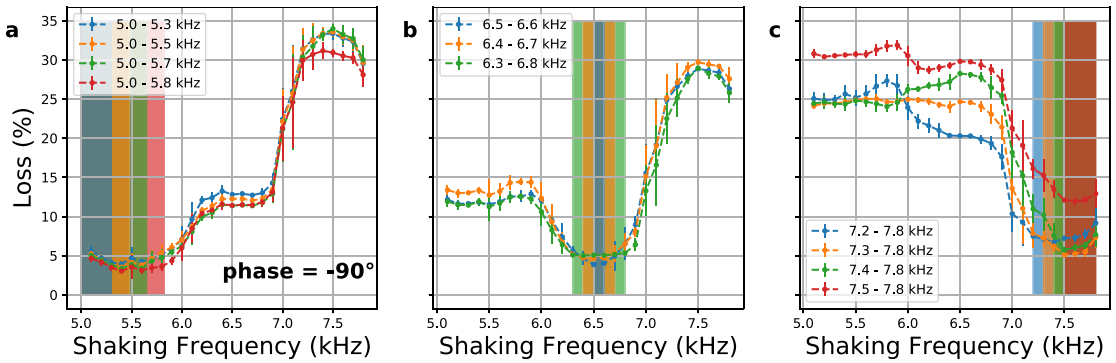

Figure 12. Consistency checks for the obtained phase boundaries for varying sizes of the training region for all three phases (a)–(c). We see that the results are consistent and that changing the training region size does not shift the obtained boundaries.

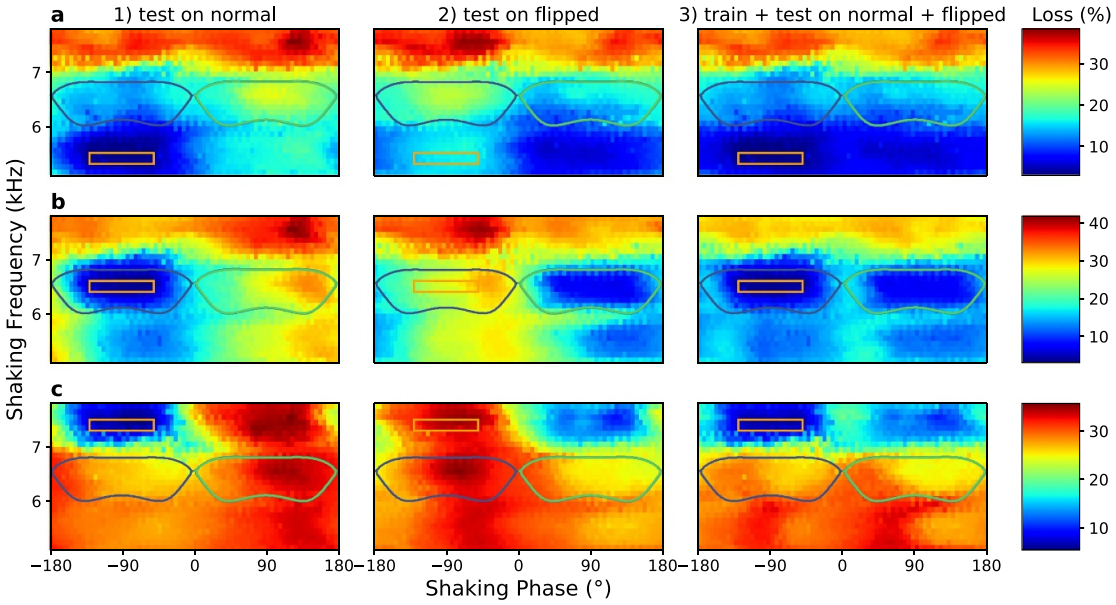

Figure 13. 1(a)–(c) Training with the small box and testing on normal images. The AE does not generalise well within the same phase in the shaking phase parameter. 2(a)–(c) Same AE as in (1), but now tested on flipped images. The loss map is qualitatively mirrored around the phase $= 0°$ axis. 3(a)–(c) Now both training sessions in testing are performed on a dataset consisting of both the normal and flipped images.

Appendix C. Anomaly detection in shaking phase direction

We perform a consistency check to confirm that the method is well behaved in the frequency parameter. For this, we train with different sizes of the training region in the respective three phases. As seen in figure 12, the obtained results are consistent, and we are confident about the three obtained boundaries presented in the main text. In panel (c), there are discrepancies inside the blue-detuned trivial phase where we performed the training. However, the plateau's onset for the topologically non-trivial phase is consistent for all four training region sizes. We find that the method does not generalise well when performing the same analysis in the shaking phase parameter. We show in figures 13 1(a)–(c) how, for smaller boxes in the shaking phase-parameter, the network has problems reproducing images from the same phase but with different and unseen shaking phase parameters. This is why we cannot differentiate between the non-trivial topological phases with Chern numbers of $+1$ and -1. We use the same network and test it on flipped images in figures 13 2(a)–(c). Here, flipping corresponds to the phase-space transformation $(k_x, k_y) \to (-k_x, k_y)$. As expected, this operation physically corresponds to changing the shaking phase φ to $-\varphi$ and should invert the sign of the Chern number. In 2(a)–(c) we train and evaluate the network on a dataset in which we use both normal and flipped images. In 3(a) and (b) we see that we now obtain the expected behaviour, i.e. generalising in the whole red-detuned trivial phase, and separating the Chern number $+1$ and -1 phases. However, in 3(c), the network still does not generalise well in the trivial blue-detuned phase. Therefore, we conclude that with this architecture, we cannot separate the two topologically non-trivial phases from each other.

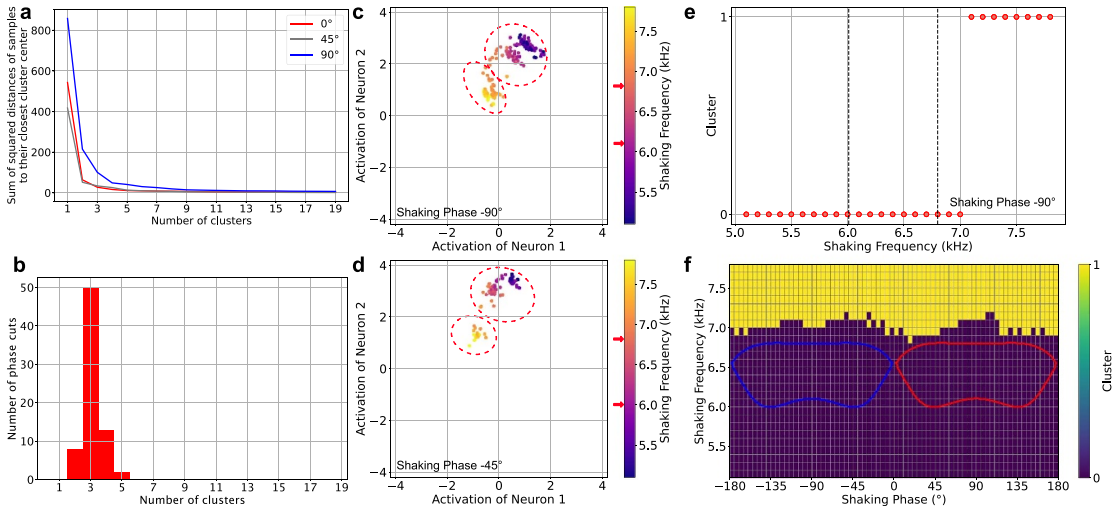

Figure 14. Bottleneck analysis of rephased data with varying numbers of clusters as the input parameter for the k-means algorithm. (a) Kneeling analysis for three different phase cuts. (b) Number of phase cuts with the best choice of clusters by the kneeling analysis. (c), (d) The analysis of single cuts with only two clusters as an input parameter to the k-means algorithm through the phase diagram, i.e. of data with a fixed shaking phase of $\varphi = -90°$ in (b) and $\varphi = -45°$ in (c), shows a significant clustering according to the shaking frequency (colour coded). The dotted lines guide the eye to separate the different clusters found by the k-means algorithm. The arrows next to the colour bar mark the theoretically expected phase transitions. (e) The cluster analysis of latent space. A k-means clustering analysis for $k = 2$ clusters shows that the data is indeed clustered according to shaking frequency. However, in contrast to figure 6, $k = 2$ does not cluster for the lower boundary. (f) Naming the two clusters increasing with frequency and combining the data from all cuts yields half of the topological phase diagram identifying the high-frequency phase boundary.

Appendix D. Optimising the VAE

The VAE employed in section 3.2 is optimised using the optimisation framework Optuna [104]. We use the tree-structured Parzen Estimator for the optimisation. Utilising a median pruner with ten startup trials, eight warmup steps, and an interval step of two allows a reduction of computing time, since fewer networks have to be trained until the end. The hyperparameters optimised by the framework can be found in our notebooks [99].

In total, we tested over 60 000 different network architectures. For the final Optuna run, we started 33 534 trials; 32 974 were stopped or pruned, since their performance was not significantly better than the average time of training, and 560 were trained to the end. In total, we spent 6889.6 h on GPU devices for the optimisation of the VAE. The average training time for the completed training was 11.8 h each. The training time for the VAE we used for the results in our paper was 10.8 h. We used 26 GPUs (Nvidia RTX 2080Ti and RTX 3090).

The network architecture we employ in the main text is the best-performing network from the final Optuna run. We note that it is replaceable by any other network with good performance from the optimisation run with a similar final loss value. Most of the 560 completed trials result in a similar final loss value. We analyse the parameter importance according to [107]. We observe a significantly higher parameter importance of the number of fully connected layers in the decoder. For a complete analysis, see the notebook appendix_optuna_analysis.ipynb in our supporting material [99].

Appendix E. k-means analysis

The number of clusters used for the k-means algorithm is crucial. We employ a so-called kneeling analysis to select the most reasonable number of clusters. To find the optimal number of clusters, we run the k-means algorithm separately with a different number of clusters as an input parameter for each shaking phase cut. Three examples are given in figure 14(a). The most suitable number of clusters is given by the knee of the curve. A histogram of the values for the knees extracted for the different phase cuts shows a clear maximum at three clusters (figure 14(b)). Hence $k = 3$ is a sophisticated choice for the number of clusters. As a reference, we run the analysis for $k = 2$, analogously to section 3.4 (figures 14(c)–(f)). A boundary arises for higher frequencies, but a correct distinction between $C = \pm 1$ and $C = 0$ is impossible.

ORCID iDs

Niklas Käming https://orcid.org/0000-0001-7926-5797
Anna Dawid https://orcid.org/0000-0001-9498-1732
Korbinian Kottmann https://orcid.org/0000-0001-7817-9571
Maciej Lewenstein https://orcid.org/0000-0002-0210-7800
Alexandre Dauphin https://orcid.org/0000-0003-4996-2561
Christof Weitenberg https://orcid.org/0000-0001-9301-2067

References

[1] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
[2] Carrasquilla J 2020 Machine learning for quantum matter *Adv. Phys.* X **5** 1797528
[3] Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602
[4] Torlai G, Mazzola G, Carrasquilla J, Troyer M, Melko R and Carleo G 2018 Neural-network quantum state tomography *Nat. Phys.* **14** 447
[5] Torlai G *et al* 2019 Integrating neural networks with a quantum simulator for state reconstruction *Phys. Rev. Lett.* **123** 19
[6] Neugebauer M, Fischer L, Jäger A, Czischek S, Jochim S, Weidemüller M and Gärttner M 2020 Neural network quantum state tomography in a two-qubit experiment *Phys. Rev.* A **102** 042604
[7] Wigley P B *et al* 2016 Fast machine-learning online optimization of ultra-cold-atom experiments *Sci. Rep.* **6** 25890
[8] Tranter A D *et al* 2018 Multiparameter optimisation of a magneto-optical trap using deep learning *Nat. Commun.* **9** 4360
[9] Bukov M, Day A G, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev.* X **8** 031086
[10] Davletov E T, Tsyganok V V, Khlebnikov V A, Pershin D A, Shaykin D V and Akimov A V 2020 Machine learning for achieving Bose-Einstein condensation of thulium atoms *Phys. Rev.* A **102** 011302(R)
[11] Carrasquilla J and Melko R G 2017 Machine learning phases of matter *Nat. Phys.* **13** 431
[12] Ch'ng K, Carrasquilla J, Melko R G and Khatami E 2017 Machine learning phases of strongly correlated fermions *Phys. Rev.* X **7** 031038
[13] 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
[14] van Nieuwenburg E P L, Liu Y-H and Huber S D 2017 Learning phase transitions by confusion *Nat. Phys.* **13** 435
[15] Wang L 2016 Discovering phase transitions with unsupervised learning *Phys. Rev.* B **94** 195105
[16] Wang C and Zhai H 2017 Machine learning of frustrated classical spin models. I. Principal component analysis *Phys. Rev.* B **96** 144432
[17] Ohtsuki T and Ohtsuki T 2016 Deep learning the quantum phase transitions in random two-dimensional electron systems *J. Phys. Soc. Japan* **85** 123706
[18] Kottmann K, Huembeli P, Lewenstein M and Acín A 2020 Unsupervised phase discovery with deep anomaly detection *Phys. Rev. Lett.* **125** 170603
[19] Huembeli P, Dauphin A and Wittek P 2018 Identifying quantum phase transitions with adversarial neural networks *Phys. Rev.* B **97** 134109
[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] Zhang Y *et al* 2019 Machine learning in electronic-quantum-matter imaging experiments *Nature* **570** 484
[22] Ziatdinov M, Maksov A, Li L, Sefat A S, Maksymovych P and Kalinin S V 2016 Deep data mining in a real space: separation of intertwined electronic responses in a lightly doped $BaFe_2As_2$ *Nanotechnology* **27** 475706
[23] Samarakoon A M *et al* 2020 Machine-learning-assisted insight into spin ice $Dy_2Ti_2O_7$ *Nat. Commun.* **11** 1
[24] 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
[25] Bohrdt A, Chiu C S, Ji G, Xu M, Greif D, Greiner M, Demler E, Grusdt F and Knap M 2019 Classifying snapshots of the doped Hubbard model with machine learning *Nat. Phys.* **15** 921
[26] 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
[27] Miles C, Bohrdt A, Wu R, Chiu C, Xu M, Ji G, Greiner M, Weinberger K Q, Demler E and Kim E-A 2020 Correlator convolutional neural networks: an interpretable architecture for image-like quantum matter data (arXiv:2011.03474 [cond-mat.str-el])
[28] Broecker P, Assaad F F and Trebst S 2017 Quantum phase recognition via unsupervised machine learning (arXiv:1707.00663)
[29] Wetzel S J 2017 Unsupervised learning of phase transitions: from principal component analysis to variational autoencoders *Phys. Rev.* E **96** 022140
[30] 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
[31] Shirinyan A A, Kozin V K, Hellsvik J, Pereiro M, Eriksson O and Yudin D 2019 Self-organizing maps as a method for detecting phase transitions and phase identification *Phys. Rev.* B **99** 041108
[32] 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
[33] Lidiak A and Gong Z 2020 Unsupervised machine learning of quantum phase transitions using diffusion maps *Phys. Rev. Lett.* **125** 225701
[34] Che Y, Gneiting C, Liu T and Nori F 2020 Topological quantum phase transitions retrieved through unsupervised machine learning *Phys. Rev.* B **102** 134213
[35] Arnold J, Schäfer F, Žonda M and Lode A U J 2020 Interpretable and unsupervised phase classification (arXiv:2010.04730 [cond-mat.dis-nn])
[36] Woloshyn R M 2021 Exploring phases of the Su-Schrieffer-Heeger model with tSNE (arXiv:2101.08704 [cond-mat.mes-hall])

[37] Casert C, Mills K, Vieijra T, Ryckebusch J and Tamblyn I 2020 Optical lattice experiments at unobserved conditions and scales through generative adversarial deep learning (arXiv:2002.07055)
[38] Lu P Y, Kim S and Soljačć M 2020 Extracting interpretable physical parameters from spatiotemporal systems using unsupervised learning *Phys. Rev.* X **10** 031056
[39] Iten R, Metger T, Wilming H, del Rio L and Renner R 2020 Discovering physical concepts with neural networks *Phys. Rev. Lett.* **124** 010508
[40] Ponte P and Melko R G 2017 Kernel methods for interpretable machine learning of order parameters *Phys. Rev.* B **96** 205146
[41] Greitemann J, Liu K and Pollet L 2019 Probing hidden spin order with interpretable machine learning *Phys. Rev.* B **99** 060404(R)
[42] 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
[43] Zhang Y, Ginsparg P and Kim E-A 2020 Interpreting machine learning of topological quantum phase transitions *Phys. Rev. Res.* **2** 23283
[44] 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
[45] Lewenstein M, Sanpera A and Ahufinger V 2012 *Ultracold Atoms in Optical Lattice: Simulating Quantum Many-Body Systems* (Oxford: Oxford University Press)
[46] Dalibard J, Gerbier F, Juzeliunas G and Öhberg P 2011 Coll.: artificial gauge potentials for neutral atoms *Rev. Mod. Phys.* **83** 1523
[47] Cooper N R, Dalibard J and Spielman I B 2019 Topological bands for ultracold atoms *Rev. Mod. Phys.* **91** 015005
[48] Bukov M, D'Alessio L and Polkovnikov A 2015 Universal high-frequency behavior of periodically driven systems: from dynamical stabilization to Floquet engineering *Adv. Phys.* **64** 139
[49] Eckardt A 2017 Coll.: atomic quantum gases in periodically driven optical lattices *Rev. Mod. Phys.* **89** 011004
[50] Cooper N R, Dalibard J and Spielman I B 2019 Topological bands for ultracold atoms *Rev. Mod. Phys.* **91** 015005
[51] Zhang Y and Kim E-A 2017 Quantum loop topography for machine learning *Phys. Rev. Lett.* **118** 216401
[52] Deng D-L, Li X and Das Sarma S 2017 Machine learning topological states *Phys. Rev.* B **96** 195145
[53] Zhang P, Shen H and Zhai H 2018 Machine learning topological invariants with neural networks *Phys. Rev. Lett.* **120** 066401
[54] Carvalho D, García-Martínez N A, Lado J L and Fernández-Rossier J 2018 Real-space mapping of topological invariants using artificial neural networks *Phys. Rev.* B **97** 115453
[55] Beach M J S, Golubeva A and Melko R G 2018 Machine learning vortices at the Kosterlitz-Thouless transition *Phys. Rev.* B **97** 045207
[56] Rodriguez-Nieva J F and Scheurer M S 2019 Identifying topological order through unsupervised machine learning *Nat. Phys.* **15** 790
[57] Holanda N L and Griffith M A 2020 Machine learning topological phases in real space *Phys. Rev.* B **102** 054107
[58] Long Y, Ren J and Chen H 2020 Unsupervised manifold clustering of topological phononics *Phys. Rev. Lett.* **124** 185501
[59] Scheurer M S and Slager R-J 2020 Unsupervised machine learning and band topology *Phys. Rev. Lett.* **124** 226401
[60] Price H M and Cooper N R 2012 Mapping the Berry curvature from semiclassical dynamics in optical lattices *Phys. Rev.* A **85** 033620
[61] Dauphin A and Goldman N 2013 Extracting the Chern number from the dynamics of a Fermi gas: implementing a quantum Hall bar for cold atoms *Phys. Rev. Lett.* **111** 135302
[62] Jotzu G, Messer M, Desbuquois R, Lebrat M, Uehlinger T, Greif D and Esslinger T 2014 Experimental realisation of the topological Haldane model with ultracold fermions *Nature* **515** 237
[63] Aidelsburger M, Lohse M, Schweizer C, Atala M, Barreiro J T, Nascimbène S, Cooper N R, Bloch I and Goldman N 2015 Measuring the Chern number of Hofstadter bands with ultracold bosonic atoms *Nat. Phys.* **11** 162
[64] Duca L, Li T, Reitter M, Bloch I, Schleier-Smith M and Schneider U 2015 An Aharonov-Bohm interferometer for determining Bloch band topology *Science* **347** 288
[65] Tran D T, Dauphin A, Grushin A G, Zoller P and Goldman N 2017 Probing topology by 'heating': quantized circular dichroism in ultracold atoms *Sci. Adv.* **3** e1701207
[66] Asteria L, Tran D T, Ozawa T, Tarnowski M, Rem B S, Fläschner N, Sengstock K, Goldman N and Weitenberg C 2019 Measuring quantized circular dichroism in ultracold topological matter *Nat. Phys.* **15** 449
[67] Alba E, Fernandez-Gonzalvo X, Mur-Petit J, Pachos J K and Garcia-Ripoll J J 2011 Seeing topological order in time-of-flight measurements *Phys. Rev. Lett.* **107** 235301
[68] Hauke P, Lewenstein M and Eckardt A 2014 Tomography of band insulators from quench dynamics *Phys. Rev. Lett.* **113** 045303
[69] Fläschner N, Rem B S, Tarnowski M, Vogel D, Lühmann D-S, Sengstock K and Weitenberg C 2016 Experimental reconstruction of the Berry curvature in a Floquet Bloch band *Science* **352** 1091
[70] Fläschner N *et al* 2018 Observation of dynamical vortices after quenches in a system with topology *Nat. Phys.* **14** 265
[71] Tarnowski M, Ünal F N, Fläschner N, Rem B S, Eckardt A, Sengstock K and Weitenberg C 2019 Measuring topology from dynamics by obtaining the Chern number from a linking number *Nat. Commun.* **10** 1728
[72] Haldane F D M 1988 Model for a quantum Hall effect without Landau levels: condensed-matter realization of the 'parity anomaly' *Phys. Rev. Lett.* **61** 2015
[73] Hu W, Singh R R P and Scalettar R T 2017 Discovering phases, phase transitions and crossovers through unsupervised machine learning: a critical examination *Phys. Rev.* E **95** 062122
[74] Ming Y, Lin C T, Bartlett S D and Zhang W W 2019 Quantum topology identification with deep neural networks and quantum walks *npj Comput. Mater.* **5** 88
[75] Rosson P, Kiffner M, Mur-Petit J and Jaksch D 2020 Characterizing the phase diagram of finite-size dipolar Bose-Hubbard systems *Phys. Rev.* A **101** 013616
[76] Oka T and Aoki H 2009 Photovoltaic Hall effect in graphene *Phys. Rev.* B **79** 081406
[77] Rechtsman M C, Zeuner J M, Plotnik Y, Lumer Y, Podolsky D, Dreisow F, Nolte S, Segev M and Szameit A 2013 Photonic Floquet topological insulators *Nature* **496** 196
[78] Dauphin A, Tran D-T, Lewenstein M and Goldman N 2017 Loading ultracold gases in topological Floquet bands: the fate of current and center-of-mass responses *2D Mater.* **4** 024010
[79] Kitagawa T, Berg E, Rudner M and Demler E 2010 Topological characterization of periodically driven quantum systems *Phys. Rev.* B **82** 235114
[80] Rudner M S, Lindner N H, Berg E and Levin M 2013 Anomalous edge states and the bulk-edge correspondence for periodically driven two-dimensional systems *Phys. Rev.* X **3** 031005

[81] Kumar A, Rodriguez-Vega M, Pereg-Barnea T and Seradjeh B 2020 Linear response theory and optical conductivity of Floquet topological insulators *Phys. Rev.* B **101** 174314
[82] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[83] Maas A L, Hannun A Y and Ng A Y 2013 Rectifier nonlinearities improve neural network acoustic models *Proc. 30th Int. Conf. on Machine Learning*
[84] Kingma D P and Ba J L 2015 Adam: a method for stochastic optimization *3rd Int. Conf. on Learning Representations, ICLR 2015—Conf. Track Proc.* (Int. Conf. on Learning Representations, ICLR)
[85] Lecun Y 1987 Modeles connexionnistes de l'apprentissage (connectionist learning models) PhD Thesis Universite P. et M. Curie (Paris 6)
[86] Bourlard H and Kamp Y 1988 Auto-association by multilayer perceptrons and singular value decomposition *Biol. Cybern.* **59** 291
[87] Hinton G E and Zemel R S 1993 Autoencoders, minimum description length and Helmholtz free energy *Proc. 6th Int. Conf. on Neural Information Processing Systems NIPS'93* (San Francisco, CA: Morgan Kaufmann Publishers Inc.) pp 3–10
[88] Vincent P, Larochelle H, Bengio Y and Manzagol P-A 2008 Extracting and composing robust features with denoising autoencoders *Proc. 25th Int. Conf. on Machine Learning, ICML '08* (New York: Association for Computing Machinery) pp 1096–103
[89] Xie J, Xu L and Chen E 2012 Image denoising and inpainting with deep neural networks *Advances in Neural Information Processing Systems* vol 25, ed F Pereira, C J C Burges, L Bottou and K Q Weinberger (Curran Associates, Inc.) pp 341–9
[90] Baldassarre F, Morín D G and Rodés-Guirao L 2017 Deep koalarization: image colorization using CNNs and Inception-ResNet-v2 *CoRR* (arXiv:1712.03400)
[91] Kingma D P and Welling M 2014 Auto-encoding variational bayes (arXiv:1312.6114 [stat.ML])
[92] Rezende D J, Mohamed S and Wierstra D 2014 Stochastic backpropagation and approximate inference in deep generative models (arXiv:1401.4082 [stat.ML])
[93] Doersch C 2016 Tutorial on variational autoencoders (arXiv:1606.05908 [stat.ML])
[94] Cook R D 1977 Detection of influential observation in linear regression *Technometrics* **19** 15
[95] Koh P W and Liang P 2017 Understanding black-box predictions via influence functions *Proc. 34th Int. Conf. on Machine Learning*
[96] van der Walt S, Colbert S C and Varoquaux G 2011 The NumPy array: a structure for efficient numerical computation *Comput. Sci. Eng.* **13** 22
[97] Paszke A *et al* 2017 Automatic differentiation in PyTorch *NIPS-W*
[98] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: http://tensorflow.org/)
[99] Käming N, Dawid A, Kottmann K, Lewenstein M, Sengstock K, Dauphin A and Weitenberg C 2021 Code and data for Unsupervised machine learning of topological phase transitions from experimental data (https://doi.org/10.5281/zenodo.4700379)
[100] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504
[101] Fitzgibbon A, Pilu M and Fisher R B 1999 Direct least square fitting of ellipses *IEEE Trans. Pattern Anal. Mach. Intell.* **21** 476
[102] Ockeloen C F, Tauschinsky A F, Spreeuw R J C and Whitlock S 2010 Detection of small atom numbers through image processing *Phys. Rev.* A **82** 061606
[103] Fung R, Hanna A M, Vendrell O, Ramakrishna S, Seideman T, Santra R and Ourmazd A 2016 Dynamics from noisy data with extreme timing uncertainty *Nature* **532** 471
[104] Akiba T, Sano S, Yanase T, Ohta T and Koyama M 2019 Optuna: a next-generation hyperparameter optimization framework *Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*
[105] Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: from error visibility to structural similarity *IEEE Trans. Image Process.* **13** 600
[106] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825
[107] Hutter F, Hoos H and Leyton-Brown K 2014 An efficient approach for assessing hyperparameter importance *Proc. 31st Int. Conf. on Machine Learning* vol 32 pp 754–62