

Unsupervised interpretable learning of phases from many-qubit systems

Nicolas Sadoune,^{1,2} Giuliano Giudici,^{1,2} Ke Liu (刘科子竞) ^{1,2} and Lode Pollet ^{1,2,3}

¹Arnold Sommerfeld Center for Theoretical Physics, University of Munich, Theresienstr. 37, 80333 München, Germany

²Munich Center for Quantum Science and Technology (MCQST), Schellingstr. 4, 80799 München, Germany

³Wilczek Quantum Center, School of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai 200240, China



(Received 12 September 2022; accepted 19 January 2023; published 6 February 2023)

Experimental progress in qubit manufacturing calls for the development of new theoretical tools to analyze quantum data. We show how an unsupervised machine-learning technique can be used to understand short-range entangled many-qubit systems using data of local measurements. The method successfully constructs the phase diagram of a cluster-state model and detects the respective order parameters of its phases, including string order parameters. For the toric code subject to external magnetic fields, the machine identifies the explicit forms of its two stabilizers. Prior information of the underlying Hamiltonian or the quantum states is not needed; instead, the machine outputs their characteristic observables. Our work opens the door for a first-principles application of hybrid algorithms that aim at strong interpretability without supervision.

DOI: [10.1103/PhysRevResearch.5.013082](https://doi.org/10.1103/PhysRevResearch.5.013082)

I. INTRODUCTION

A shared central theme of quantum computation science and quantum many-body physics is the estimation of a quantum state from a large number of measurements. Standard quantum state tomography quickly becomes infeasible for this task due to the exponential scaling of the many-body Hilbert space [1–3]. Various strategies, such as matrix product state (MPS) tomography [4,5], neural network tomography [6,7], and randomized measurements [8–10] have been proposed to improve the efficiency by restricting the target functions to particular information or specific types of quantum states. A noticeable recent advance in efficiency is the development of shadow tomography [11,12] and the classical shadow scheme [13–15], which promise to estimate a range of observables accurately from considerably less measurements. Nevertheless, the resources required to achieve a certain accuracy still strongly depend on the entanglement properties of the target state and the complexity of the observable [13,16]. Therefore, it is highly desirable to develop an algorithm where the defining features of an unknown complex quantum state can be detected and returned as an interpretable output. Those can then be used to make predictions for new measurements and serve as input for more sophisticated tomography processes.

In this work, we show that the tensorial-kernel support vector machine (TK-SVM) [17,18] can directly analyze positive-operator valued measure (POVM) measurements and identify local entanglement structures in quantum states, in addition to its proven power of detecting hidden tensorial

orders [17,18] and classical ground-state constraints [19]. The *unsupervised* nature and the *strong interpretability* of this machine promise to characterize complex quantum states and construct unknown quantum phase diagrams without knowledge of the states or the Hamiltonian. We demonstrate the capability of our method by mapping out the phase diagram of a cluster model without supervision and extracting its analytical string order parameters for arbitrary ranks using experimentally accessible data. The machine is also applied to a toric code model subject to a magnetic field [20], where we show that it can distinguish the topologically ordered phase from a trivially disordered phase and identify the stabilizer operators even far away from the zero field limit.

II. METHOD

One of the two uses of our framework is the detection of local observables that are translational invariant with respect to a unit cell comprised of a number of adjacent lattice sites. To that end, a support vector machine (SVM) is trained to classify the input samples against a set of fictitious featureless samples. Once the training is completed, the local observables characterizing the input are retrieved from the learning-model internals. In this context, *interpretability* means that the model internals have a direct physical meaning. This is in sharp contrast to neural-network based methods, which often lack interpretability. The second use of TK-SVM consists in exploring completely unknown phase diagrams. For this purpose multiclassification of a predefined grid of phase points is carried out, followed by partitioning of a graph that is built from learning-model parameters as weighted edges and phase points as vertices.

Although standard SVM is a supervised learning method, TK-SVM can be seen as unsupervised, since the labeling of training samples used in TK-SVM carries no information at all about the physical quantities (order parameters) that are

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

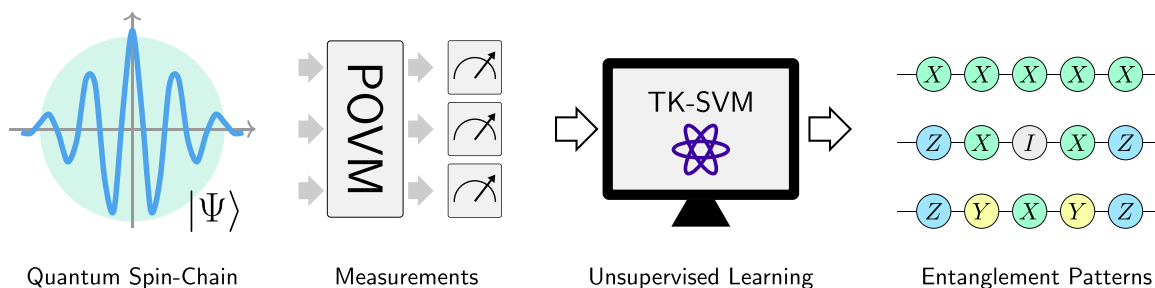


FIG. 1. Work flow of the tensorial-kernel support vector machine (TK-SVM). The learning stage is unsupervised and local correlation patterns as well as local orders in the state can be extracted systematically from the trained model.

being learned. The TK-SVM working scheme is depicted in Fig. 1.

Data and classical map. From a many-qubit system, repetitive POVM measurements are taken to serve as input to TK-SVM. Informational completeness (IC) of measurements is crucial to our approach, which is why we require POVM outcomes as input data (details in Appendix A). The choice of the IC-POVM does not affect the algorithm or the results. Therefore, we consider the experiment-friendly and intuitive Pauli-6 POVM with six possible outcomes, defined by a set of positive semidefinite Hermitian matrices $M^{\uparrow\alpha} = \frac{1}{3}|\uparrow_{\alpha}\rangle\langle\uparrow_{\alpha}|$ and $M^{\downarrow\alpha} = \frac{1}{3}|\downarrow_{\alpha}\rangle\langle\downarrow_{\alpha}|$ with $\alpha \in \{x, y, z\}$ and $\{|\uparrow_{\alpha}\rangle, |\downarrow_{\alpha}\rangle\}$ representing the eigenbases of the Pauli operators X , Y , and Z . Every possible outcome is assigned to a three-component classical vector $\mathbf{S} = (S^x, S^y, S^z)$, e.g., $M^{\uparrow x} \mapsto (+1, 0, 0)$, $M^{\downarrow x} \mapsto (-1, 0, 0)$, $M^{\uparrow y} \mapsto (0, +1, 0)$. Measuring each qubit of the system in a random Pauli basis and mapping the outcomes yields one classical snapshot of a quantum state. This is repeated a number N_s of times to obtain a set of samples used as input data.

Feature vector construction. The SVM at the core of our framework does not directly operate on classical spin configurations, but on a set of feature vectors constructed from these configurations instead. Two hyperparameters, *cluster*—a collection of adjacent lattice sites—and *rank* generally determine the way that a feature vector is constructed. In this section, only the chain lattice is considered, as the generalization to other lattice geometries is straightforward. For the chain lattice, there is only one possible cluster shape: the string. Therefore, the relevant hyperparameters become string length n and rank r . The optimal choice of n and r depends on the precise nature of the local orders present. Nematic order, for instance, can by definition never be captured at rank $r = 1$. In practice finding the optimal choice of hyperparameters requires multiple TK-SVM runs on a trial-and-error basis. Often it is also necessary to combine results from several different choices of hyperparameters to obtain a complete physical picture. The latter is the case for the cluster model.

For a fixed choice of hyperparameters, a classical spin configuration of L sites is partitioned into nonoverlapping clusters $\{S_i^c\} \mapsto \{S_{J,\alpha}^c\}$, where $i \in \{1, \dots, L\}$ labels the lattice sites, $c \in \{x, y, z\}$ the spin component, $J \in \{1, \dots, L/n\}$ labels the clusters, and $\alpha \in \{1, \dots, n\}$ the site within one cluster. Note that, for the partitioning to be meaningful, L/n must be integer. For every cluster J , a feature vector of rank r is then constructed by taking all possible products of different $S_{J,\alpha}^c$

with exactly r factors. This yields a set of feature vectors $\{\phi_J\}$ upon which, in a last step, the cluster average is taken to produce an averaged feature vector ϕ

$$\phi_J = \{S_{J,\alpha_1}^{c_1} \times S_{J,\alpha_2}^{c_2} \times \dots \times S_{J,\alpha_r}^{c_r}\}, \quad (1)$$

$$\phi = \frac{n}{L} \sum_J \phi_J. \quad (2)$$

To exclude trivial products in Eq. (1), any two site indices, e.g., α_1 and α_2 , must be different. This condition forbids r to be greater than n . Additionally, redundancies are removed from Eq. (1) by imposing an ordering $\alpha_1 < \alpha_2 < \dots < \alpha_r$ on the site indices. This simplification is allowed since Pauli operators on different sites commute. The dimension of the resulting feature vector is independent of the system size L and given by $\dim(\phi) = \binom{n}{r} \times 3^r$.

The procedure above describes how to construct one feature vector from one sample. It is however desirable to create a single feature vector from several samples, by extending the average in Eq. (2), to enhance accuracy. Especially with increasing rank, the cluster average should be taken over increasingly many samples. This ensures that the error of the components of ϕ , which can be seen as estimators of r -point correlators, stays arbitrarily small. For more details on the feature vector construction see Appendix B.

A. Learning local orders (binary classification)

When used to learn local orders, the input samples constitute the first class of the training set, whereas the second class of the training set is constructed internally. With the aim to learn physical quantities only from the input samples, the artificial second class is chosen to lack any sort of information. Physically the artificial class might be interpreted as samples from a system at infinite temperature. To emulate such a system, POVM outcomes are generated randomly according to a uniform distribution. After the artificial samples are generated, feature vectors are constructed as described in the previous section. This results in two labeled sets of feature vectors that can now be fed to the SVM.

In our approach, besides quantifying the learning success (Appendix D), the decision function is never used to predict unknown samples. Instead the focus lies solely on the information gained from the input samples, which is encoded in the learning-model internals after training. These model internals

are extracted from the decision function and interpreted as physical observables, characterizing the input samples.

The quadratic-kernel SVM decision function predicting the class of an unknown feature vector $\phi = \{\phi_\mu\}$ reads

$$d(\phi) = \sum_s \lambda_s y_s (\phi^{(s)} \cdot \phi)^2 - b \quad (3)$$

$$= \sum_{\mu\nu} \left(\sum_s \lambda_s y_s \phi_\mu^{(s)} \phi_\nu^{(s)} \right) \phi_\mu \phi_\nu - b \quad (4)$$

$$= \sum_{\mu\nu} C_{\mu\nu} \phi_\mu \phi_\nu - b, \quad (5)$$

where s runs over the combined training set of feature vectors from both classes, $y_s = \pm 1$ represents the class label of each feature vector, and λ_s as well as the bias b are the SVM optimization parameters. The resulting coefficient matrix $\{C_{\mu\nu}\}$ depends on the training data and optimization parameters only and exposes all local orders present in the phase of interest (that the input samples were taken from). From each of its nonvanishing components, an analytical expression interpreted as order parameter or part of an order parameter can be recovered. Typically a small subset of columns or rows of $C_{\mu\nu}$ is sufficient for inferring an order parameter and constructing the entire matrix is not necessary (see Appendix B). This allows us to examine large clusters and high ranks, which is particularly useful for detecting nonlinear and extended orders.

B. Learning a phase diagram (multiclassification)

In general a model's phase diagram might be completely unknown. In this case it is simpler to first get an idea of the phase diagram's topology by roughly determining phase boundaries, instead of extracting order parameters of randomly selected phase points. Using labeled samples from two separate phase points A and B as input (without constructing the artificial class), TK-SVM allows one to decide whether they belong to the same phase, without explicitly extracting local orders from the decision function. Note that the samples are labeled without knowing if they are truly physically different or not, meaning that this application of TK-SVM is still unsupervised. The decision whether two classes belong to the same phase is based on a criterion derived in [21],

$$\begin{aligned} |b_{A,B}| &\gg \text{lif } A, B \text{ in the same phase,} \\ &\lesssim \text{lif } A, B \text{ in different phases,} \end{aligned} \quad (6)$$

which relies on the bias b of the decision function Eq. (3) alone.

To find the approximate topology of an unknown phase diagram, we first fix a grid of M phase points. Then, for all possible $M(M-1)/2$ different pairs of phase points TK-SVM runs a binary classification task, returning a set of biases. Succeeding the classification stage, a graph is constructed using phase points as vertices and the set of absolute biases as weighted edges. Since all points of the same phase are strongly connected (large bias modulus) and points of different phases are weakly connected, partitioning the resulting graph directly yields the topology of the phase diagram. There are many standard methods to partition a graph; the one

used in this work is the spectral graph partitioning method introduced by Fiedler [22,23] (summarized in Appendix C). In practice, this protocol for unsupervised learning of a phase diagram turns out to perform better than the kernel principal component analysis (PCA) [21] and persistent homology [24] methods.

III. APPLICATIONS

A. Cluster model

To demonstrate the abilities of TK-SVM, we consider a spin-1/2 Hamiltonian including a paramagnetic, a symmetry-broken, and a symmetry-protected topological (SPT) phase,

$$H = - \sum_{i=2}^{L-1} Z_{i-1} X_i Z_{i+1} - h_1 \sum_{i=1}^L X_i - h_2 \sum_{i=1}^{L-1} X_i X_{i+1}, \quad (7)$$

where the sums run over the lattice sites of a chain with open boundary conditions and the external field $h_1 \geq 0$. Distinct phases of the Hamiltonian can be understood from the following limits: for $h_1 \rightarrow +\infty$ the system is a paramagnet with all spins pointing along \hat{x} , while for $h_2 \rightarrow -\infty$ it is an Ising antiferromagnet along \hat{x} . The ground state at $h_1 = h_2 = 0$ is known as the cluster state protected by a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry and plays a crucial role in measurement-based quantum computations [25–28].

The data used in this work is generated from density matrix renormalization group (DMRG) simulations [29]; nevertheless we expect the same analysis to apply for POVM measurements of experimental origin. A total of 255 data sets, each containing $N_s = 5000$ POVM samples, are collected uniformly in a region spanning the whole phase diagram, with system size $L = 72$. Since a Hamiltonian can generally host distinct types of phases and a single phase may also develop more than one order, two strategies may be used to construct the phase diagram. The first one is to carry out a multiclassification over the entire data set with different ranks and clusters, each covering several types of orders. The full phase diagram is then obtained by combining the results of the subsequent graph partitioning. Alternatively, one can work iteratively by restricting the multiclassification to data sets belonging to the same component (subgraph) of a preceding classification analysis and repeat until a converged phase diagram topology is reached. Both strategies are more efficient than finding a universal rank and cluster sensitive to all orders and they can be combined in practice. In the current problem, we find that rank $r = 3$ and cluster size $n = 3$ reproduce the expected phase diagram, as shown in Fig. 2. Increasing r and n does not reveal a finer partitioning; hence no other phases are found.

After obtaining the phase diagram, we wish to investigate the order parameters characterizing each phase. We focus our discussion on interpreting the SPT phase, since the paramagnetic and antiferromagnetic phases only lead to very simple features $\langle \sum_{i=1}^L X_i \rangle = 1$ and $\langle \sum_{i=1}^{L/2} X_{2i-1} - X_{2i} \rangle = 1$. We start by discussing the pure cluster model in the limit $h_1 = h_2 = 0$ and analyze ranks and clusters up to $r, n = 9$. The first non-trivial feature emerges at $n = r = 3$ with a structure

$$B_k := Z_{k-1} X_k Z_{k+1}, \quad (8)$$

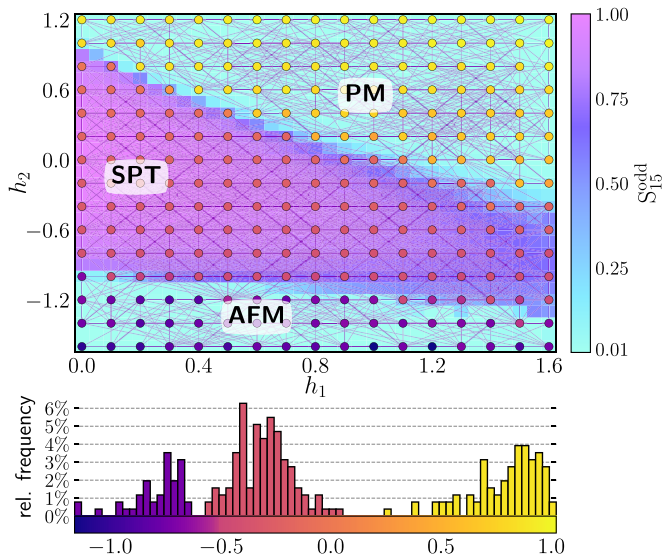


FIG. 2. Upper panel: partitioned graph corresponding to the cluster model phase diagram, learned at rank $r = 3$ and cluster size $n = 3$. The graph consists of a uniform 15×17 grid of phase points (filled circles) as vertices and the set of normalized bias parameters, obtained from all mutual classification tasks between the phase points, as edges. Partitioning the graph leads to three components (sub-graphs), which are identified as the paramagnetic (PM), SPT ordered, and antiferromagnetic (AFM) phases. Every phase point is colored according to the appropriate Fiedler vector entry, which indicates its phase correspondence. The background displays the expectation value of the string order parameter $S_n^{\text{odd}} = \langle Z_1 (\prod_{k=1}^{n-1} X_{2k}) Z_n \rangle$ with length $n = 15$, which is finite in the SPT phase but vanishes elsewhere. Lower panel: a histogram of the Fiedler vector entries (for details, see Appendix C), whose values are color coded (the horizontal axis). The three separated regions correspond to the three phases.

where $k = 2, \dots, n-1$ labels qubits within the cluster. Increasing the rank to $r = 4$ and cluster size to $n \geq 5$, two additional structures are detected (see Fig. 3),

$$B_k B_{k+1} = Z_{k-1} Y_k Y_{k+1} Z_{k+2}, \quad k \in \{2, \dots, n-2\}, \quad (9)$$

$$B_k B_{k+2} = Z_{k-1} X_k X_{k+2} Z_{k+3}, \quad k \in \{2, \dots, n-3\}. \quad (10)$$

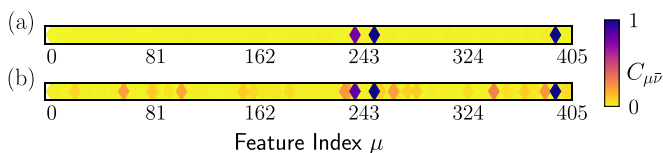


FIG. 3. Nontrivial features in the SPT phase learned at $r = 4$, $n = 5$. The figures show a nonvanishing column of the associated $C_{\mu\bar{\nu}}$ matrix, with $\bar{\nu}$ chosen such that $\phi_{\bar{\nu}} = Z_2 Y_3 Y_4 Z_5$ is a prominent feature. (a) The pure cluster state in the limit $h_1 = h_2 = 0$. Nonvanishing features (diamonds) can be systematically retrieved: the three apparent features correspond to entangled patterns $B_2 B_4$, $B_2 B_3$, and $B_3 B_4$ [cf. Eqs. (9) and (10)]. (b) All phase points belonging to the SPT phase according to the graph partition are pooled and treated as one data set. The features from the pure limit remain dominant, as they reflect the local correlation structure of the whole phase.

TABLE I. Excerpt of nontrivial features learned at different ranks and cluster sizes r, n . The dimension of the feature space is $\dim(\phi) = \binom{n}{r} \times 3^r$. N_f denotes the number of nonvanishing features at fixed r and n .

r, n	$\dim(\phi)$	N_f	Example
3,6	540	4	$B_4 = Z_3 X_4 Z_5$
4,5	405	3	$B_3 B_4 = Z_2 Y_3 Y_4 Z_5$
5,7	5103	8	$B_2 B_4 B_6 = Z_1 X_2 X_4 X_6 Z_7$
6,9	61236	30	$B_2 B_4 B_5 B_7 = Z_1 X_2 Y_4 Y_5 X_7 Z_8$
7,9	78732	35	$B_2 B_3 B_5 B_6 B_8 = Z_1 Y_2 Y_3 Y_5 Y_6 X_8 Z_9$
8,9	59049	22	$B_2 B_3 B_4 B_6 B_7 B_8 = Z_1 Y_2 X_3 Y_4 Y_6 X_7 Y_8 Z_9$
9,9	19683	6	$\prod_{k=2}^8 B_k = -Z_1 Y_2 X_3 X_4 X_5 X_6 X_7 Y_8 Z_9$

In Table I we summarize the features learned for ranks and cluster sizes $r, n \in \{3, \dots, 9\}$. Extrapolating $n, r \rightarrow \infty$, they reproduce the commonly used “dense” and “odd” string order parameters [27,30]

$$S_{n \rightarrow \infty}^{\text{dense}} = \left\langle \prod_{k=2}^{n-1} B_k \right\rangle = (-1)^n \left\langle Z_1 Y_2 \left(\prod_{k=3}^{n-2} X_k \right) Y_{n-1} Z_n \right\rangle, \quad (11)$$

$$S_{n \rightarrow \infty}^{\text{odd}} = \left\langle \prod_{k=1}^{\frac{n-1}{2}} B_{2k} \right\rangle = \left\langle Z_1 \left(\prod_{k=1}^{\frac{n-1}{2}} X_{2k} \right) Z_n \right\rangle. \quad (12)$$

Although the high-rank features can be generated by the building block B_k , it is remarkable that a machine automatically picks up all variants of the string correlations.

To find the order parameter for arbitrary parameter values of h_1, h_2 , all data sets belonging to the same phase according to the machine-learned phase diagram are pooled. For the SPT phase the resulting pooled data set contains 96×5000 samples. Classifying the pooled data set against random samples reveals that the same operators as in Table I remain dominant throughout the phase. Secondary features caused by finite h_1, h_2 are significantly weaker; see Fig. 3. We thus identified characteristic local entanglement structures in a quantum phase, without relying on particular known limits nor any other information about the Hamiltonian. This marks a crucial difference of our machine compared to popular strategies in neural-network based algorithms [31–33], which make use of special known limits during the training stage.

B. Toric code subject to magnetic fields

Characterizing long-range entangled topological phases, e.g., by identifying their anyonic statistics [20,34] or topological classes [35,36], in a purely data-driven manner, arguably remains beyond the capabilities of existing machine learning algorithms [37]. Nevertheless, information about local correlation patterns can bring valuable insight into the state and guide further tomography processes specialized for topological features. To examine what information can be learned from local POVM data, we consider a toric code model subject to magnetic fields $h_x, h_z > 0$ [20],

$$H = - \sum_v \prod_{i \in v} Z_i - \sum_p \prod_{i \in p} X_i - h_x \sum_i X_i - h_z \sum_i Z_i. \quad (13)$$



FIG. 4. Features detected by TK-SVM for the toric code model using a (2×2) -unit cell (eight sites) cluster at rank 4. The prominent coefficient matrix column \bar{v} is given by $\phi_{\bar{v}} = A_v$, with v the only vertex contained in an eight-site cluster. (a) In the zero field limit $h_x, h_z \rightarrow 0$, two features (blue diamonds) reflecting the vertex and plaquette operators A_v and B_p are captured. (b) The same features remain prominent even at sizable magnetic fields $h_x = h_z = 0.3$ close to the phase boundary. (c) In the nontopological phase at $(h_x, h_z) = (0, 1.5)$, multiple trivial features appear due to the strong polarization.

Qubits are located on the bonds of a square lattice and v and p denote vertices and plaquettes of the lattice. This model is topologically ordered for $h_x = h_z \lesssim 0.34$ if $h_x = h_z$ and $h_x (h_z) \lesssim 0.33$ if $h_z (h_x) = 0$ [38]. Training data are produced by exactly diagonalizing the Hamiltonian Eq. (13) on a periodic system with 18 qubits.

We first investigate the pure toric code limit $h_x = h_z \rightarrow 0$. $N_S = 1000$ Pauli-6 POVM snapshots are collected and discriminated against random samples. We find that, using a cluster of 2×2 unit cells (8 qubits, as in Appendix B) at rank 4, the machine captures two features whose interpretation yields the toric code stabilizers $A_v = \prod_{i \in v} Z_i$ and $B_p = \prod_{i \in p} X_i$, as shown in Fig. 4. The same procedure is carried out at sizable fields $(h_x, h_z) = (0.3, 0.3)$, where we identify the same prime features in contrast to the trivial paramagnetic phase at $(h_x, h_z) = (0, 1.5)$ as can be seen in Fig. 4. The machine can successfully detect the correct stabilizers from local measurements sampled far away from the pure toric code limit. Although stabilizers do not directly characterize the topological phase, their explicit forms indicate the underlying gauge structure and can inspire specialized feature mapping and kernel designs. For instance, one may generate training sets by sampling closed loops [39,40] or define the kernel of a gauge symmetry [41,42], which are strategies employed in previous studies of machine-learning intrinsic topological orders.

IV. SUMMARY

Machine learning techniques exhibit growing abilities of analyzing complex classical and quantum data. In this work, we demonstrated the potential of TK-SVM as a *first-principle* method to detect entanglement structures in many-body qubit systems. We constructed the phase diagram of the cluster model, without supervision, from experimentally accessible POVM snapshots. Furthermore we extracted the respective order parameters of the phases systematically and, in particular, discussed the local entanglement patterns and string order parameters of its SPT phase. Finally, we examined an intrinsic topological phase subject to magnetic fields and showed the machine's ability to detect the explicit stabilizers with data far

away from the pure limit. Our results pave the way to investigate membranelike order parameters in higher dimensional SPT phases [43] and analyze local measurements of general topological models such as lattice gauge theories [44] and fracton models [45].

ACKNOWLEDGMENTS

We acknowledge support from FP7/ERC Consolidator Grant QSIMCORR, No. 771891, and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2111—Grant No. 390814868. The research is part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. Our simulations make use of the ν -SVM formulation [46], the LIBSVM library [47,48], and the iTensor library [49].

APPENDIX A: IC-POVM AND CLASSICAL MAP

Over a Hilbert space of dimension d , a general POVM is defined as a set of positive-semidefinite operators $\mathbf{M} = \{M^i\}$, called “outcomes,” which sum to identity $\sum_i M_i = I$. To be informationally complete (IC), the outcomes of a POVM must span the whole space of self-adjoint operators acting on the Hilbert space. This can only be fulfilled if the POVM has at least d^2 outcomes; if it has exactly d^2 outcomes it is called minimal IC-POVM. Moreover, a POVM is symmetric if the overlap between any two different outcomes is constant $\text{Tr}(M^i M^j) = \text{const}$. Informational completeness ensures that the density matrix can be uniquely determined from the probabilities $\text{Tr}(M^i \rho)$ and is therefore a strict necessity for our algorithm. On the other hand, POVM symmetry is completely optional for TK-SVM. Given an IC-POVM, every outcome is mapped onto a classical vector $\mathbf{S} \in \mathbb{R}^{d^2-1}$ serving as representation of a single site quantum state, the density matrix of which is determined by $d^2 - 1$ real parameters. Specifically for the spin 1/2 ($d = 2$) case, the classical one-to-one map $M^i \mapsto S^i \in S^2$ (onto the Bloch sphere) reads

$$M^i = \frac{d}{|\mathbf{M}|} \rho(S^i) = \frac{1}{|\mathbf{M}|} (I + S^i \cdot \boldsymbol{\sigma}), \quad (\text{A1})$$

where $\boldsymbol{\sigma}$ denotes the Pauli vector. A straight-forward choice for an IC-POVM is the Pauli-6 POVM, defined by the set of Bloch vectors $\{(\pm 1, 0, 0)^T, (0, \pm 1, 0)^T, (0, 0, \pm 1)^T\}$. Although it is not minimal, it has the advantage of being easy to implement in experiments by simply measuring each site in a randomly chosen Pauli basis.

Alternatively, one might use the Tetra-POVM, which is minimal and symmetric (SIC-POVM), defined by the set $\{(0, 0, 1)^T, (2\sqrt{2}/3, 0, -1/3)^T, (-\sqrt{2}/3, -\sqrt{2}/3, -1/3)^T, (-\sqrt{2}/3, \sqrt{2}/3, -1/3)^T\}$, spanning a tetrahedron. Given that a certain POVM is IC, if all of its Bloch vectors can be expressed as linear combinations of the vectors defining a second POVM, then this second POVM is IC as well. Using either the Tetra- or the Pauli-6 POVM yields fully consistent results in our application of TK-SVM.

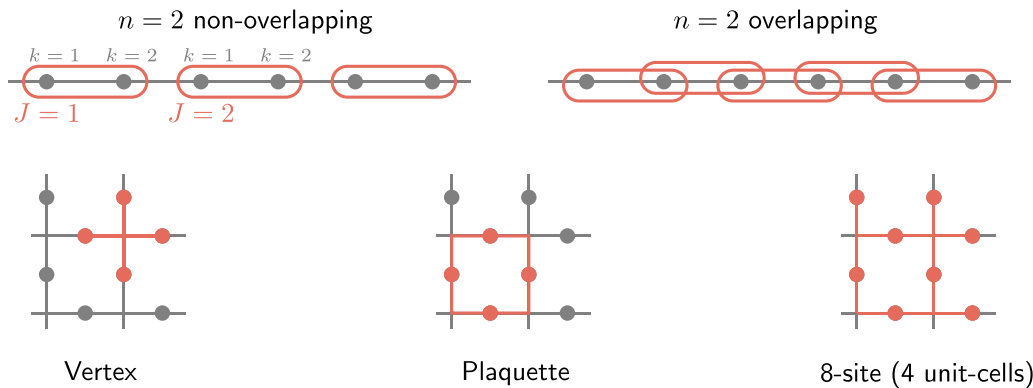


FIG. 5. Different examples of clusters for the chain lattice (top) and square link lattice (bottom). For an unknown quantum state, natural choices for a cluster include multiple lattice cells and the nearest neighbors of a site. The clusters may or may not be overlapping. Nonoverlapping clusters are appropriate for detecting structures with translational symmetry breaking, while overlapping clusters are more suitable for probing structures with translational invariance. In the case of nonoverlapping clusters of the chain, the lattice size L must be divisible by n , such that the number of clusters $N_{\text{cl.}} = L/n$ is meaningfully defined as an integer. Similar conditions arise for other lattices when using nonoverlapping clusters. In the case of the square link lattice, the vertex and plaquette clusters are sufficient to learn the constraints $A_v = 1$ and $B_p = 1$, respectively. The most unbiased cluster choice when no information about the Hamiltonian is available, however, is a multiple of unit cells of the lattice such as the eight-site cluster, which is large enough to capture both local constraints simultaneously.

APPENDIX B: CLUSTERING AND FEATURE DETECTION

For a fixed choice of hyperparameters every Bloch-vector configuration is partitioned into clusters $\{S_i^c\} \mapsto \{S_{J,\alpha}^c\}$ as described in Sec. II for the chain lattice. Figure 5 shows several possible clustering choices for the chain and square link lattice. The feature construction used in TK-SVM makes use of the fact that local order parameters of symmetry-broken phases [17] and entanglement structures of SPT phases [43] can be represented by a finite number of elementary degrees of freedom. The feature mapping provides a high-dimensional vector space to host potential local orders and entanglement structures, while SVM optimizations search for their explicit expressions. The results of the SVM optimization are weights (λ_s) for the feature vectors $\phi^{(s)}$, from which the coefficient matrix $C_{\mu\nu} = \sum_s \lambda_s y_s \phi_\mu^{(s)} \phi_\nu^{(s)}$ [cf. Eq. (5)] is constructed. Each row or column of $C_{\mu\nu}$ encodes pairwise correlations between a feature ϕ_ν and the entire set of features $\{\phi_\mu\}$. Hence, to find the nontrivial features, it typically suffices to construct a few nonvanishing rows or columns of the matrix. Without loss of generality, one can start by examining a single column with strong overall weight $|\sum_s \lambda_s \phi_\nu^{(s)}|$. The number of columns required for information convergence may depend on the nature of an order but is in principle bounded by the size of the unit cell. In case of SPT phase in the cluster-state model and the toric code (Fig. 3 and Fig. 4 in the main text), a single nonvanishing column is enough for extracting the features.

APPENDIX C: GRAPH PARTITIONING

A weighted graph $G = (V, E, w)$ is defined as a tuple of a set of vertices V , a set of edges E , and a set of normalized weights w on the edges. All graphs produced by TK-SVM are of a restricted class of undirected graphs with no self-edges (edges that connect a vertex to itself) and no multiedges (E is not a multiset).

The goal is to find a two-way partition $V_1 \cup V_2 = V \mid V_1 \cap V_2 = \emptyset$, such that $|V_1| \approx |V_2|$ and the connectivity, i.e., the

summed weight of all edges connecting V_1 and V_2 , is minimal. Fiedler's theory [22,23] makes use of the Laplace matrix

$$\Lambda_{i \neq j} = \begin{cases} -w_{ij}, & \text{if } (i, j) \in E, \\ 0, & \text{otherwise,} \end{cases} \quad \Lambda_{ii} = \sum_{j|(i,j) \in E} w_{ij}. \quad (\text{C1})$$

Further define a partition vector as $\mathbf{x} = \{x_i = \pm 1\}$ where a positive (negative) sign indicates that the vertex i belongs to V_1 (V_2). For simplicity we assume that the graph is connected, as the discussion of multicomponent graphs can always be reduced to discussing the disconnected components separately.

For an arbitrary partition vector \mathbf{x} , the quadratic form $\mathbf{x}^T \Lambda \mathbf{x}$ equals four times the connectivity

$$\mathbf{x}^T \Lambda \mathbf{x} = \sum_{(ij) \in E} w_{ij} (x_i - x_j)^2 = 4 \sum_{(ij) \in E_{\text{con}}} w_{ij}, \quad (\text{C2})$$

where $E_{\text{con}} \subset E$ is the set of edges that connect V_1 and V_2 . So the goal is to find a partition vector that minimizes the quadratic form while maintaining (approximately) equally sized parts. By changing from discrete variables $x_i \in \{-1, 1\}$ to continuous ones $z_i \in [-1, 1]$ and diagonalizing Λ , it can be shown that

$$\min(\mathbf{z} \Lambda \mathbf{z}) = |V| \lambda_2, \quad (\text{C3})$$

with λ_2 being the second smallest of all eigenvalues of Λ . Note that any Laplace matrix has non-negative eigenvalues. Thus the argument minimizing the quadratic form is simply the second eigenvector $\mathbf{z}_{(2)}$ of Λ . To recover a discrete solution from the optimal continuous partition vector $\mathbf{z}_{(2)}$, the sign function is applied to each component

$$\mathbf{x}_{\text{min}} = \text{sgn}(\mathbf{z}_{(2)}). \quad (\text{C4})$$

For a general multi- rather than two-way partition of the graph, in place of the sign function applied to the Fiedler vector, clustered appearances of similar entries can indicate more than two parts. In applications of TK-SVM, the final number of parts of the graph is unknown, requiring a multiway

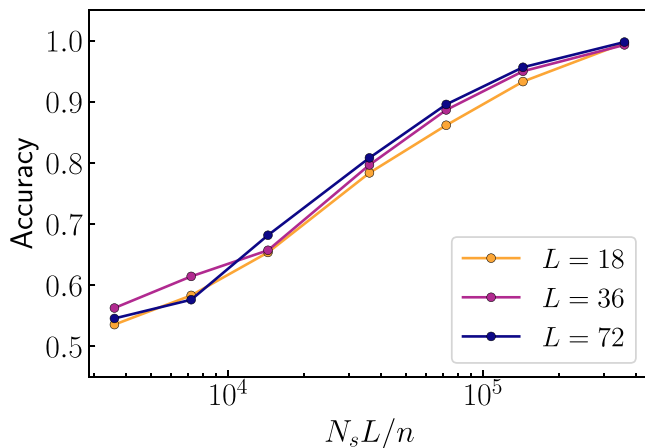


FIG. 6. Dependence of binary prediction accuracy on the number of samples (N_s) and system size (L). The x axis is scaled with the number of local clusters, L/n , in the sample. Samples are collected from the clean cluster state and classified against random samples. A $r = n = 5$ TK-SVM with overlapping clusters is employed.

partition and hence a continuous representation of the Fiedler vector. The lower panel of Fig. 2 displays the histogram of Fiedler vector entries as obtained from partitioning the graph constructed for the cluster model.

In the TK-SVM framework, we relate the learning of an unknown phase diagram to a graph partitioning problem. Specifically, each data set sampled at some physical parameter is represented as a vertex in the graph. Edges connecting two vertices are weighted by the bias parameter b ; see Eq. (6). The weights are normalized using a Lorentzian function

$$w(e_b) = 1 - \frac{b_c^2}{(|b| - 1)^2 + b_c^2} \in [0, 1), \quad (\text{C5})$$

where the parameter b_c sets a characteristic scale quantifying “ $\gg 1$.” As we consider a fully connected graph, edges between data sets located deep in phases are also included. This redundancy makes the choice of b_c uncritical: the partitioning is typically robust against varying b_c over several orders of magnitude [19].

APPENDIX D: BENCHMARKING PREDICTION ACCURACY

Here we investigate the learning efficiency of our machine as a function of system size and number of samples. For simplicity, we restrict ourselves to train the cluster state of

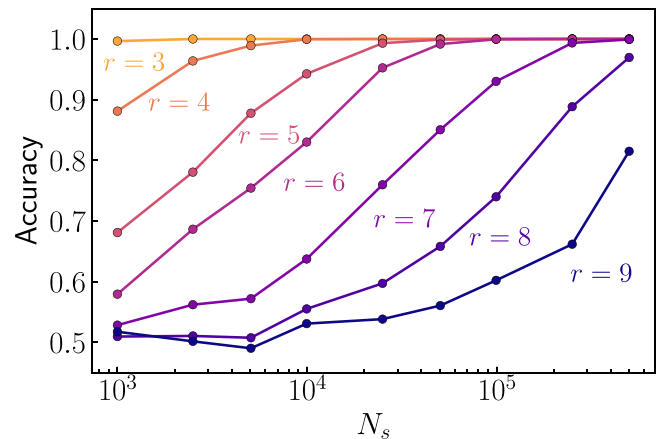


FIG. 7. Dependence of binary prediction accuracy on the ranks and number of samples for a system of size $L = 72$, using overlapping clusters. An accuracy of ~ 0.8 is typically enough to learn a clear structure for the underlying order.

length L against random samples. The learning success is determined by the accuracy with which the machine correctly classifies data from an unseen test set. The benchmarks in this section aim to understand how many samples are sufficient for our machine to learn the characteristic observables of an entangled quantum state.

We first discuss the dependence on system size with fixed rank r and cluster size n . As displayed from the coinciding data points for $L = 18, 36, 72$ in Fig. 6, our machine shows a (nearly) constant complexity over the total number of clusters $N_s L/n$: fewer samples are needed for larger systems to reach a certain accuracy. This is understood from the feature construction Eq. (1): larger systems have more local clusters to average over, yielding smaller errors of the corresponding feature.

Next, we discuss the effects of the rank r , which detects r -body quantities. We choose $r = n$ for simplicity and analyze the prediction accuracy up to rank $r = 9$. As shown in Fig. 7, the required number of samples to reach good accuracy scales exponentially with r as expected, since the dimension of the feature space is $\propto 3^r$. This has a clear physical implication: in general, the efficiency of an unbiased machine learning algorithm strongly depends on the nature of the phase and the complexity of the target variable. A few samples are typically enough for learning the simplest orders, but complicated orders and entanglement patterns require more. It also represents an intrinsic difficulty for unbiased algorithms to learn arbitrarily high-rank quantities and long-range entanglement structures.

- [1] K. Banaszek, M. Cramer, and D. Gross, Focus on quantum tomography, *New J. Phys.* **15**, 125020 (2013).
- [2] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu, *Sample-Optimal Tomography of Quantum States* (Association for Computing Machinery, New York, 2016), pp. 913–925.
- [3] C. Song, K. Xu, W. Liu, C.-p. Yang, S.-B. Zheng, H. Deng, Q. Xie, K. Huang, Q. Guo, L. Zhang, P. Zhang, D. Xu, D. Zheng,

- X. Zhu, H. Wang, Y.-A. Chen, C.-Y. Lu, S. Han, and J.-W. Pan, 10-Qubit Entanglement and Parallel Logic Operations with a Superconducting Circuit, *Phys. Rev. Lett.* **119**, 180511 (2017).
- [4] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, Efficient quantum state tomography, *Nat. Commun.* **1**, 149 (2010).

- [5] B. P. Lanyon, C. Maier, M. Holzäpfel, T. Baumgratz, C. Hempel, P. Jurcevic, I. Dhand, A. S. Buyskikh, A. J. Daley, M. Cramer, M. B. Plenio, R. Blatt, and C. F. Roos, Efficient tomography of a quantum many-body system, *Nat. Phys.* **13**, 1158 (2017).
- [6] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [7] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita, Reconstructing quantum states with generative models, *Nat. Mach. Intell.* **1**, 155 (2019).
- [8] B. Vermersch, A. Elben, L. M. Sieberer, N. Y. Yao, and P. Zoller, Probing Scrambling Using Statistical Correlations between Randomized Measurements, *Phys. Rev. X* **9**, 021061 (2019).
- [9] T. Brydges, A. Elben, P. Jurcevic, B. Vermersch, C. Maier, B. P. Lanyon, P. Zoller, R. Blatt, and C. F. Roos, Probing Rényi entanglement entropy via randomized measurements, *Science* **364**, 260 (2019).
- [10] A. Elben, J. Yu, G. Zhu, M. Hafezi, F. Pollmann, P. Zoller, and B. Vermersch, Many-body topological invariants from randomized measurements in synthetic quantum matter, *Sci. Adv.* **6**, eaaz3666 (2020).
- [11] S. Aaronson, Shadow tomography of quantum states, in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018 (Association for Computing Machinery, New York, 2018), pp. 325–338.
- [12] S. Aaronson and G. N. Rothblum, Gentle measurement of quantum states and differential privacy, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, 2019), pp. 322–333.
- [13] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, *Nat. Phys.* **16**, 1050 (2020).
- [14] H.-Y. Huang, R. Kueng, and J. Preskill, Efficient Estimation of Pauli Observables by Derandomization, *Phys. Rev. Lett.* **127**, 030503 (2021).
- [15] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, Provably efficient machine learning for quantum many-body problems, *Science* **377** (2022).
- [16] S. Shivam, C. W. von Keyserlingk, and S. L. Sondhi, On classical and hybrid shadows of quantum states, [arXiv:2206.06616](https://arxiv.org/abs/2206.06616).
- [17] J. Greitemann, K. Liu, and L. Pollet, Probing hidden spin order with interpretable machine learning, *Phys. Rev. B* **99**, 060404(R) (2019).
- [18] Ke Liu, J. Greitemann, and L. Pollet, Learning multiple order parameters with interpretable machines, *Phys. Rev. B* **99**, 104410 (2019).
- [19] J. Greitemann, K. Liu, L. D. C. Jaubert, H. Yan, N. Shannon, and L. Pollet, Identification of emergent constraints and hidden order in frustrated magnets using tensorial kernel methods of machine learning, *Phys. Rev. B* **100**, 174408 (2019).
- [20] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, *Ann. Phys. (NY)* **303**, 2 (2003).
- [21] J. Greitemann, Ph.D. thesis, Ludwig Maximilians University Munich, 2019, available at <http://nbn-resolving.de/urn:nbn:de:bvb:19-250579>.
- [22] M. Fiedler, Algebraic connectivity of graphs, *Czech. Math. J.* **23**, 298 (1973).
- [23] M. Fiedler, A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory, *Czech. Math. J.* **25**, 619 (1975).
- [24] B. Olshoorn, J. Hellsvik, and A. V. Balatsky, Finding hidden order in spin models with persistent homology, *Phys. Rev. Res.* **2**, 043308 (2020).
- [25] R. Raussendorf and H. J. Briegel, A One-Way Quantum Computer, *Phys. Rev. Lett.* **86**, 5188 (2001).
- [26] M. A. Nielsen, Optical Quantum Computation Using Cluster States, *Phys. Rev. Lett.* **93**, 040503 (2004).
- [27] F. Verstraete and J. I. Cirac, Valence-bond states for quantum computation, *Phys. Rev. A* **70**, 060302(R) (2004).
- [28] M. A. Nielsen, Cluster-state quantum computation, *Rep. Math. Phys.* **57**, 147 (2006).
- [29] U. Schollwck, The density-matrix renormalization group in the age of matrix product states, *Ann. Phys. (NY)* **326**, 96 (2011), January 2011 Special Issue.
- [30] A. Smith, B. Jobst, A. G. Green, and F. Pollmann, Crossing a topological phase transition with a quantum computer, *Phys. Rev. Res.* **4**, L022020 (2022).
- [31] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, *Nat. Phys.* **15**, 1273 (2019).
- [32] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap, Classifying snapshots of the doped Hubbard model with machine learning, *Nat. Phys.* **15**, 921 (2019).
- [33] K. Kottmann, P. Huembeli, M. Lewenstein, and A. Acín, Unsupervised Phase Discovery with Deep Anomaly Detection, *Phys. Rev. Lett.* **125**, 170603 (2020).
- [34] A. Kitaev, Anyons in an exactly solved model and beyond, *Ann. Phys. (NY)* **321**, 2 (2006), January Special Issue.
- [35] T. Lan, L. Kong, and X.-G. Wen, Classification of (3 + 1)D Bosonic Topological Orders: The Case When Pointlike Excitations Are All Bosons, *Phys. Rev. X* **8**, 021074 (2018).
- [36] T. Lan and X.-G. Wen, Classification of 3 + 1D Bosonic Topological Orders (II): The Case When Some Pointlike Excitations Are Fermions, *Phys. Rev. X* **9**, 021005 (2019).
- [37] We note that, in the machine-learning literature, the term “classifying phases” typically refers to discriminating the phases in given data sets instead of a topological or symmetry classification.
- [38] F. Wu, Y. Deng, and N. Prokof'ev, Phase diagram of the toric code model in a parallel magnetic field, *Phys. Rev. B* **85**, 195104 (2012).
- [39] Y. Zhang, R. G. Melko, and E.-A. Kim, Machine learning \mathbb{Z}_2 quantum spin liquids with quasiparticle statistics, *Phys. Rev. B* **96**, 245119 (2017).
- [40] D. Sehayek and R. G. Melko, Persistent homology of \mathbb{Z}_2 gauge theories, *Phys. Rev. B* **106**, 085111 (2022).
- [41] J. F. Rodriguez-Nieva and M. S. Scheurer, Identifying topological order through unsupervised machine learning, *Nat. Phys.* **15**, 790 (2019).
- [42] M. S. Scheurer and R.-J. Slager, Unsupervised Machine Learning and Band Topology, *Phys. Rev. Lett.* **124**, 226401 (2020).
- [43] X. Chen, Z.-C. Gu, Z.-X. Liu, and X.-G. Wen, Symmetry protected topological orders and the group cohomology of their symmetry group, *Phys. Rev. B* **87**, 155114 (2013).

- [44] J. B. Kogut, An introduction to lattice gauge theory and spin systems, *Rev. Mod. Phys.* **51**, 659 (1979).
- [45] S. Vijay, J. Haah, and L. Fu, Fracton topological order, generalized lattice gauge theory, and duality, *Phys. Rev. B* **94**, 235157 (2016).
- [46] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, New support vector algorithms, *Neural Comput.* **12**, 1207 (2000).
- [47] C.-C. Chang and C.-J. Lin, Training ν -support vector classifiers: Theory and algorithms, *Neural Comput.* **13**, 2119 (2001).
- [48] C.-C. Chang and C.-J. Lin, Libsvm: A library for support vector machines, *ACM Trans. Intell. Syst. Technol.* **2**, 1 (2011).
- [49] M. Fishman, S. R. White, and E. M. Stoudenmire, Codebase release 0.3 for ITensor, *SciPost Phys. Codebases* 4-r0.3 (2022).