Provably efficient machine learning for quantum many-body problems

Hsin-Yuan Huang,[1] Richard Kueng,[2] Giacomo Torlai,[3] Victor V. Albert,[4] and John Preskill[1,3]

[1]*Institute for Quantum Information and Matter and Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA*
[2]*Institute for Integrated Circuits, Johannes Kepler University Linz, Austria*
[3]*AWS Center for Quantum Computing, Pasadena, CA, USA*
[4]*Joint Center for Quantum Information and Computer Science, National Institute of Standards and Technology and University of Maryland, College Park, MD, USA*

(Dated: June 25, 2021)

Classical machine learning (ML) provides a potentially powerful approach to solving challenging quantum many-body problems in physics and chemistry. However, the advantages of ML over more traditional methods have not been firmly established. In this work, we prove that classical ML algorithms can efficiently predict ground state properties of gapped Hamiltonians in finite spatial dimensions, after learning from data obtained by measuring other Hamiltonians in the same quantum phase of matter. In contrast, under widely accepted complexity theory assumptions, classical algorithms that do not learn from data cannot achieve the same guarantee. We also prove that classical ML algorithms can efficiently classify a wide range of quantum phases of matter. Our arguments are based on the concept of a classical shadow, a succinct classical description of a many-body quantum state that can be constructed in feasible quantum experiments and be used to predict many properties of the state. Extensive numerical experiments corroborate our theoretical results in a variety of scenarios, including Rydberg atom systems, 2D random Heisenberg models, symmetry-protected topological phases, and topologically ordered phases.

I. INTRODUCTION

Solving quantum many-body problems, such as finding ground states of quantum systems, has far-reaching consequences for physics, materials science, and chemistry. While classical computers have facilitated many profound advances in science and technology, they often struggle to solve such problems. Powerful methods, such as density functional theory [84, 105], quantum Monte Carlo [18, 34, 144] and density-matrix renormalization group [181, 182], have enabled solutions to certain restricted instances of many-body problems, but many general classes of problems remain outside the reach of even the most advanced classical algorithms.

Scalable fault-tolerant quantum computers will be able to solve a broad array of quantum problems, but are unlikely to be available for years to come. Meanwhile, how can we best exploit our powerful classical computers to advance our understanding of complex quantum systems? Recently, classical machine learning (ML) techniques have been adapted to investigate problems in quantum many-body physics [29, 31], with promising results [30, 32, 68, 69, 143, 163, 164, 171, 178]. So far these approaches are mostly heuristic, reflecting the general paucity of rigorous theory in ML. While shown to be effective in some intermediate-size experiments [21, 140, 165], these methods are generally not backed by convincing theoretical arguments to ensure good performance, particularly for problem instances where traditional classical algorithms falter.

In general, simulating quantum many-body physics is hard for classical computers, because accurately describing an n-qubit quantum system may require an amount of classical data that is exponential in n. In prior work, some of us addressed this bottleneck using *classical shadows* — succinct classical descriptions of quantum many-body states that can be used to accurately predict a wide range of properties with rigorous performance guarantees [88, 129]. Furthermore, this quantum-to-classical conversion technique can be readily implemented in various existing quantum experiments [39, 54, 157]. Classical shadows open new opportunities for addressing quantum problems using classical methods such as ML. In this paper, we build on the classical shadow formalism and devise efficient classical ML algorithms for quantum many-body problems which are supported by rigorous theory.

We consider two applications of classical ML, indicated in Figure 1. The first application we examine is learning to predict classical representations of quantum many-body ground states. We consider a family of Hamiltonians, where the Hamiltonian $H(x)$ depends smoothly on m real parameters (denoted by x). The ML algorithm is trained on a set of training data consisting of sampled values of x, each accompanied by the corresponding classical shadow for the ground state $\rho(x)$ of $H(x)$. This training data could be obtained from either classical simulations or quantum experiments. During the prediction phase, the ML algorithm predicts a classical representation of $\rho(x)$ for new values of x different from those in the training data. Ground state properties can then be estimated using the predicted classical representation.

This learning algorithm is efficient, provided that the ground state properties to be predicted do not vary too rapidly as a function of x. Indeed, sufficient upper bounds on the gradient can be derived for any family of gapped geometrically-local Hamiltonians in any finite spatial dimension, if the property of interest is the

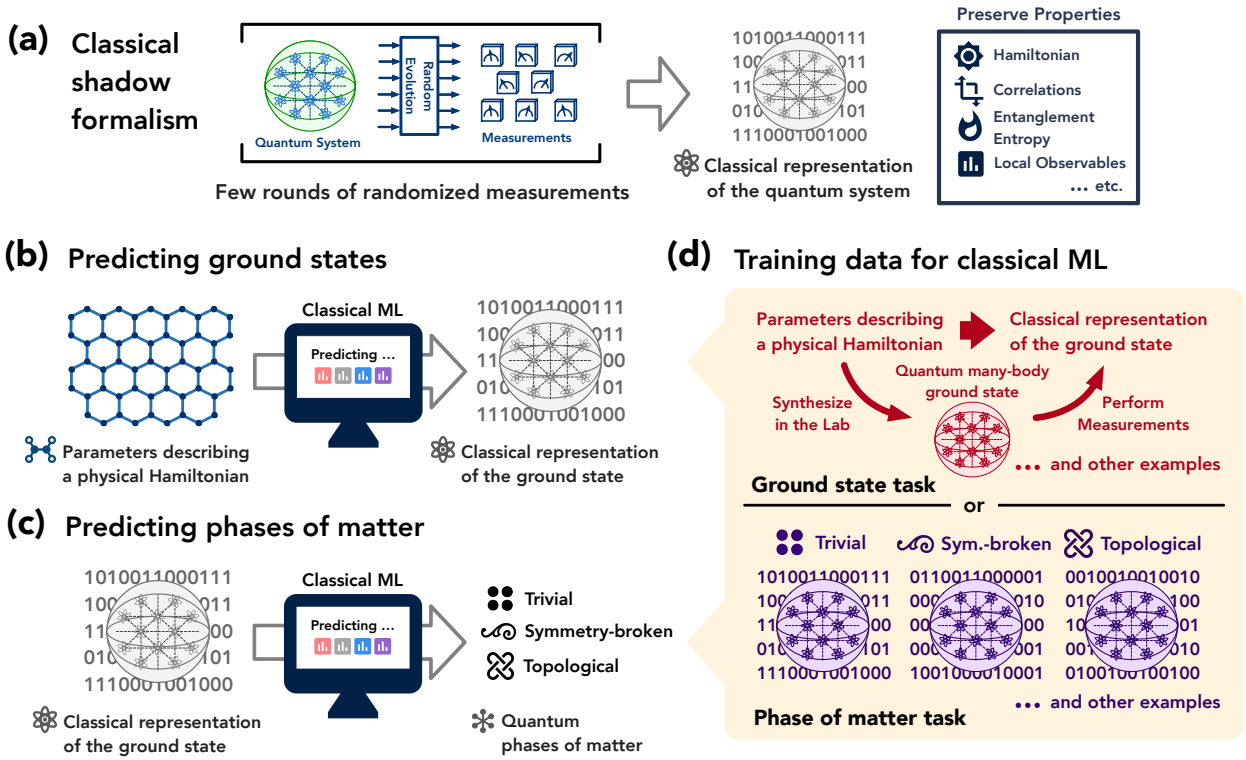

Figure 1: (a) Efficient quantum-to-classical conversion. The classical shadow of a quantum state, constructed by measuring very few copies of the state, can be used to predict many properties of the state with a rigorous performance guarantee. (b) Predicting ground state properties. After training on data obtained in quantum experiments, a classical ML model predicts a classical representation of the ground state $\rho(x)$ of the Hamiltonian $H(x)$ for parameters x spanning the entire phase. This representation yields estimates of the properties of $\rho(x)$, avoiding the need to run exhaustive classical computations or quantum experiments. (c) Classifying quantum phases. After training, a classical ML receives a classical representation of a quantum state, and predicts the phase from which the state was drawn. (d) Training data. For predicting ground states, the classical ML receives a classical representation of $\rho(x)$ for each value of x sampled during training. For predicting quantum phases of matter, the training data consists of classical representations of quantum states accompanied by labels identifying the phase to which each state belongs.

expectation value of a sum of few-body observables. The conclusion is that any such property can be predicted with a small average error, where the amount of training data and the classical computation time are polynomial in m and linear in the system size. Furthermore, we show that classical algorithms that do not learn from data cannot provide the same rigorous guarantee without violating widely accepted complexity-theoretic conjectures. This is a manifestation of the advantage of ML algorithms with data over those without data [87].

In the second application we examine, the goal is to classify quantum states of matter into phases [139] in a supervised learning scenario. Suppose that during training we are provided with sample quantum states which carry labels indicating whether each state belongs to phase A or phase B. Our goal is to predict the phase label for new quantum states that were not encountered during training. We assume that, during both the learning and prediction stages, each quantum state is represented by its classical shadow, which could be obtained either from a classical computation or from an experiment on a quantum device. The classical ML, then, trains on labeled classical shadows, and learns to predict labels for new classical shadows.

We assume that the A and B phases can be distinguished by a nonlinear function of marginal density operators of subsystems of constant size. This assumption is reasonable because we expect the phase to be revealed in subsystems which are larger than the correlation length, but independent of the total system size. We show that if such a function exists, a classical ML can learn to distinguish the phases using an amount of training data and classical processing which are polynomial in the system size. We do not need to know anything about this nonlinear function in advance, apart from its existence.

In what follows, we briefly review the classical shadow formalism [88], and then use this formalism to derive rigorous guarantees for ML algorithms in predicting ground state properties and classifying quantum phases of matter. We also describe numerical experiments in a wide range of physical systems to support our theoretical results.

II. CONSTRUCTING EFFICIENT CLASSICAL REPRESENTATIONS OF QUANTUM SYSTEMS

We begin with an overview of the randomized measurement toolbox [55, 58, 88, 126, 129, 170], relegating further details to Appendix A. We approximate an n-qubit quantum state ρ by performing randomized single-qubit Pauli measurements on T copies of ρ. That is, we measure every qubit of the unknown quantum state ρ in a random Pauli basis X, Y or Z to yield a measurement outcome of ± 1. Collapse of the wavefunction implies that this measurement procedure transforms ρ into a random pure product state $|s^{(t)}\rangle = \bigotimes_{i=1}^{n} |s_i^{(t)}\rangle$, where $|s_i^{(t)}\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\}$ are eigenstates of the selected Pauli matrices. Performing one randomized measurement grants us classical access to one such snapshot. Performing a total of T randomized measurements grants us access to an entire collection $S_T(\rho) = \big\{|s_i^{(t)}\rangle : \ i \in \{1, \ldots, n\}, \ t \in \{1, \ldots, T\}\big\}$. Each element is a highly structured single-qubit pure state, and there are nT of them in total. So, $3nT$ bits suffice to store the entire collection in classical memory. The randomized measurements can be performed in actual physical experiments or through classical simulations. Resulting data can then be used to approximate the underlying n-qubit state ρ:

$$\rho \approx \sigma_T(\rho) = \frac{1}{T} \sum_{t=1}^{T} \sigma_1^{(t)} \otimes \cdots \otimes \sigma_n^{(t)} \quad \text{where} \quad \sigma_i^{(t)} = 3|s_i^{(t)}\rangle\langle s_i^{(t)}| - \mathbb{I}, \tag{1}$$

and $\mathbb{I}$ denotes the 2×2 identity matrix. This *classical shadow* representation [88, 129] exactly reproduces the global density matrix in the limit $T \to \infty$, but $T = \mathcal{O}(\mathrm{const}^r \log(n)/\epsilon^2)$ already provides an ϵ-accurate approximation of *all* reduced r-body density matrices (in trace distance). This, in turn, implies that we can use $\sigma_T(\rho)$ to predict any function that depends on only reduced density matrices, such as expectation values of (sums of) local observables and (sums of) entanglement entropies of small subsystems. Classical storage and postprocessing cost also remain tractable in this regime. To summarize, the classical shadow formalism equips us with an efficient quantum-to-classical converter that allows classical machines to efficiently and reliably estimate subsystem properties of any quantum state ρ.

III. PREDICTING GROUND STATES OF QUANTUM MANY-BODY SYSTEMS

We consider the task of predicting ground state representations of quantum many-body Hamiltonians in finite spatial dimensions. Suppose that a family of geometrically local, n-qubit Hamiltonians $\{H(x) : \ x \in [-1, 1]^m\}$ is parametrized by a classical variable x. That is, $H(x)$ smoothly maps a bounded m-dimensional vector x (parametrization) to a Hermitian matrix of size $2^n \times 2^n$ (n-qubit Hamiltonian). We do not impose any additional structure on this mapping; in particular, we do not assume knowledge about how the physical Hamiltonian depends on the parameterization. The goal is to learn a model $\hat{\sigma}(x)$ that can predict properties of the ground state $\rho(x)$ associated with Hamiltonian. This problem arises in many practical scenarios. Suppose diligent experimental effort has produced experimental data for ground state properties of various physical systems. We would like to use this data to train an ML model that predicts ground state representations of hitherto unexplored physical systems.

A. An ML algorithm with rigorous guarantee

We will prove that a classical ML algorithm can predict classical representations of ground states after training on data belonging to the same quantum phase of matter. Formally, we consider a smooth family of Hamiltonians $H(x)$ with a constant spectral gap. During the training phase of the ML algorithm, many values of x are randomly sampled, and for each sampled x, the classical shadow of the corresponding ground state $\rho(x)$ of $H(x)$ is provided, either by classical simulations or quantum experiments. The full training data of size N is given by $\big\{x_\ell \to \sigma_T(\rho(x_\ell))\big\}_{\ell=1}^{N}$, where T is the number of randomized measurements in the construction of the classical shadows at each value of x_ℓ.

We train classical ML models using the size-N training data, such that when given the input x_ℓ, the ML model can produce a classical representation $\hat{\sigma}(x)$ that approximates $\sigma_T(\rho(x_\ell))$. During prediction, the classical ML produces $\hat{\sigma}(x)$ for new values of x different from those in the training data. While $\hat{\sigma}(x)$ and $\sigma_T(\rho(x_\ell))$ classically represent exponentially large density matrices, the training and prediction can be done efficiently on a classical computer using various existing classical ML models, such as neural networks with large hidden layers [52, 92, 113, 125] and kernel methods [35, 46]. In particular, the predicted output of the trained classical

ML models can be written as the extrapolation of the training data using a learned metric $\kappa(x, x_\ell) \in \mathbb{R}$,

$$\hat{\sigma}(x) = \frac{1}{N}\sum_{\ell=1}^{N} \kappa(x, x_\ell)\sigma_T(\rho(x_\ell)). \tag{2}$$

For example, prediction using a trained neural network with large hidden layers [92] is equivalent to using the metric $\kappa(x, x_\ell) = \sum_{\ell'=1}^{N} f^{(\mathrm{NTK})}(x, x_{\ell'})(F^{-1})_{\ell'\ell}$, where $f^{(\mathrm{NTK})}(x, x')$ is the neural tangent kernel [92] corresponding to the neural network and $F_{\ell'\ell} = f^{(\mathrm{NTK})}(x_{\ell'}, x_\ell)$; see Appendix C for more discussion. The ground state properties are then estimated using these predicted classical representations $\hat{\sigma}(x)$. Specifically, $f_O(x) = \mathrm{tr}\,(O\rho(x))$ can be predicted efficiently whenever O is a sum of few-body operators.

To derive a provable guarantee, we consider the simple metric $\kappa(x, x_\ell) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \cos(\pi k \cdot (x - x_\ell))$ with cutoff Λ, which we refer to as the l_2-Dirichlet kernel. We prove that the prediction will be accurate and efficient if the function $f_O(x)$ does not vary too rapidly when x changes in any direction. Indeed, sufficient upper bounds on the gradient magnitude of $f_O(x)$ can be derived using quasi-adiabatic continuation [13, 81].

Under the l_2-Dirichlet kernel, the classical ML model is equivalent to learning a truncated Fourier series to approximate the function $f_O(x)$. The parameter Λ is a cutoff for the wavenumber k that depends on (upper bounds on) the gradient of $f_O(x)$. Using statistical analysis, one can guarantee that $\mathbb{E}_x\,|\,\mathrm{tr}(O\hat{\sigma}(x)) - f_O(x)|^2 \leq \epsilon$ as long as the amount of training data obeys $N = m^{\mathcal{O}(1/\epsilon)}$ in the $m \to \infty$ limit. The conclusion is that any such $f_O(x)$ can be predicted with a small *constant* average error, where the amount of training data and the classical computation time are polynomial in m and at most linear in the system size n. Moreover, the training data need only contain a *single* classical shadow snapshot at each point x_ℓ in the parameter space (i.e., $T = 1$). An informal statement of the theorem is given below; we explain the proof strategy in Appendix E, and provide more details in Appendix F.

Theorem 1 (Learning to predict ground state representations; informal)**.** *For any smooth family of Hamiltonians $\{H(x) : x \in [-1, 1]^m\}$ in a finite spatial dimension with a constant spectral gap, a classical machine learning algorithm can learn to predict a classical representation of the ground state $\rho(x)$ of $H(x)$ that approximates few-body reduced density matrices up to a constant error ϵ when averaged over x. The required training data size N and computation time are polynomial in m and linear in the system size n.*

Though formally "efficient" in the sense that N scales polynomially with m for any fixed approximation error ϵ, the required amount of training data scales badly with ϵ. This unfortunate scaling is not a shortcoming of the considered ML algorithm, but a necessary feature. In Appendix G, we show that the data size and time complexity cannot be improved further without making stronger assumptions about the class of gapped local Hamiltonians. However, in cases of practical interest, the Hamiltonian may obey restrictions such as translational invariance or graph structure that can be exploited to obtain better results. Incorporating these restrictions can be achieved by using a suitable $\kappa(x, x_\ell)$, such as one that corresponds to a large-width convolutional neural network [113] or a graph neural network [52]. Rigorously establishing that neural-network-based ML algorithms can achieve improved prediction performance and efficiency for particular classes of Hamiltonians is a goal for future work.

B. Computational hardness for non-ML algorithms

In the following proposition, we show that a classical algorithm that does not learn from data cannot achieve the same guarantee in estimating ground state properties without violating the widely believed conjecture that NP-complete problems cannot be solved in randomized polynomial time. This proposition is a corollary of standard complexity-theoretic results [114, 169]. See Appendix H for the detailed statement and proof.

Proposition 1 (Informal; A variant of Lemma 1.4 in [3])**.** *Suppose a randomized polynomial-time classical algorithm that does not learn from data can efficiently compute expectation values of one-body observables up to a constant error in the ground states of any smooth class of two-dimensional Hamiltonians with a constant spectral gap. Then, the randomized classical algorithm can solve NP-complete problems in polynomial time.*

It is instructive to observe that a classical ML algorithm with access to data can perform tasks that cannot be achieved by classical algorithms which do not have access to data. This phenomenon is studied in [87], where it is shown that the complexity class defined by classical algorithms that can learn from data is strictly larger than the class of classical algorithms that do not learn from data. (The data can be regarded as a restricted form of randomized advice string.) We caution that obtaining the data to train the classical ML model could be challenging. However, if we focus only on data that could be efficiently generated by quantum-mechanical processes, it is still possible that a classical ML that learns from data could be more powerful than classical computers. In Appendix H we present a contrived class of Hamiltonians that establishes this claim, based on the (classical) computational hardness of factoring.

IV. CLASSIFYING QUANTUM PHASES OF MATTER

Classifying quantum phases of matter is another important application of machine learning to physics. We will consider this classification problem in the case where quantum states are succinctly represented by their classical shadows. For simplicity, we consider the classification of two phases (denoted A and B), but the analysis naturally generalizes to classifying any number of phases.

A. ML algorithms

We envision training a classical ML with classical shadows, where each classical shadow carries a label y indicating whether it represents a quantum state ρ from phase A ($y(\rho) = 1$) or phase B ($y(\rho) = -1$). We want to show that a suitably chosen classical ML can learn to efficiently predict the phase for new classical shadows beyond those encountered during training. Following a strategy which is standard in learning theory, we consider a classical ML that maps each classical shadow to a corresponding feature vector in a high-dimensional feature space, and then attempts to find a hyperplane that separates feature vectors in the A phase from feature vectors in the B phase. The learning is efficient if the geometry of the feature space is efficiently computable, and if the feature map is sufficiently expressive. Thus, our task is to construct a feature map with the desired properties.

In the simpler task of classifying symmetry-breaking phases, there is typically a local order parameter $O = \sum_i O_i$ given as a sum of r-body observables for some $r > 0$ that satisfies

$$\mathrm{tr}(O\rho) \geq 1, \forall \rho \in \text{phase } A, \qquad \mathrm{tr}(O\rho) \leq -1, \forall \rho \in \text{phase } B. \tag{3}$$

Under this criterion, the classification function may be chosen to be $y(\rho) = \mathrm{sign}(\mathrm{tr}(O\rho))$. Hence, classifying symmetry-breaking phases can be achieved by finding a hyperplane that separates the two phases in the high-dimensional feature space that subsumes all r-body reduced density matrices of the quantum state ρ. The feature vector consisting of all r-body reduced density matrices of the quantum state ρ can be accurately reconstructed from the classical shadow representation $S_T(\rho)$ when T is sufficiently large.

Finding a suitable choice of hyperplane in the feature space can be cast as a convex optimization problem known as the soft-margin support vector machine, discussed in more detail in Appendix J.1. With a sufficient amount of training data, the hyperplane found by the classical ML model will generalize so the phase $y(\rho)$ can be predicted accurately for a previously unseen quantum state ρ. The classical ML is not merely a black box; it exhibits the order parameter (encoded by the hyperplane), guiding physicists toward a deeper understanding of the phase structure.

For more exotic quantum phases of matter, such as topologically ordered phases, the above classical ML model no longer suffices. The topological phase of a state is invariant under a constant-depth quantum circuit, and a phase containing the product state $|0\rangle^{\otimes n}$ is called the trivial phase. Using these notions, we can prove that no observable — not even one that acts on the entire system — can be used to distinguish between two topological phases. The proof, given in Appendix I, uses the observation that random single-qubit unitaries can confuse any global or local order parameter.

Proposition 2. *Consider two distinct topological phases A and B (one of the phases could be the trivial phase). No observable O exists such that*

$$\mathrm{tr}(O\rho) > 0, \forall \rho \in \textit{phase } A, \qquad \mathrm{tr}(O\rho) \leq 0, \forall \rho \in \textit{phase } B. \tag{4}$$

While this proposition implies that no linear function $\mathrm{tr}(O\rho)$ can be used to classify topologically ordered phases, it does not exclude nonlinear functions, such as quadratic functions $\mathrm{tr}(O\rho \otimes \rho)$, degree-$d$ polynomials $\mathrm{tr}(O\rho^{\otimes d})$ and more general analytic functions. For example, it is known that the topological entanglement entropy [102, 110], a nonlinear function of ρ, can be used to classify a wide variety of topologically ordered phases. For this purpose, it suffices to consider a subsystem whose size is large compared to the correlation length of the state, but is independent of the total size of the system. The correlation length in the ground state of a local Hamiltonian increases when the spectral gap between the ground state and the first excited state becomes smaller [79]. On the other hand, a linear function on the full system will fail even with constant correlation length.

To learn nonlinear functions, we need a more expressive ML model. For this purpose we devise a powerful feature map that takes the classical shadow $S_T(\rho)$ of the quantum state ρ to a feature vector that includes arbitrarily-large r-body reduced density matrices, as well as an arbitrarily-high-degree polynomial expansion,

$$\phi^{(\text{shadow})}(S_T(\rho)) = \lim_{D,R\to\infty} \bigoplus_{d=0}^{D} \sqrt{\frac{\tau^d}{d!}} \left(\bigoplus_{r=0}^{R} \sqrt{\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r} \bigoplus_{i_1=1}^{n} \cdots \bigoplus_{i_r=1}^{n} \mathrm{vec}\left[\frac{1}{T} \sum_{t=1}^{T} \bigotimes_{\ell=1}^{r} \sigma_{i_\ell}^{(t)} \right] \right)^{\otimes d}, \tag{5}$$

where $\tau, \gamma > 0$ are hyper-parameters. The direct sum $\bigoplus_{r=0}^{R}$ is a concatenation of all r-body reduced density matrices, and the other direct sum $\bigoplus_{d=0}^{D}$ subsumes all degree-d polynomial expansions. The computational cost of finding a hyperplane in feature space that separates the training data into two classes is dominated by the cost of computing inner products between feature vectors. The inner product $\langle \phi^{(\mathrm{shadow})}(S_T(\rho)), \phi^{(\mathrm{shadow})}(S_T(\tilde{\rho}))\rangle$ can be analytically computed by reorganizing the direct sums, writing it as a double series, and wrapping both series into an exponential, which gives

$$k^{(\mathrm{shadow})}\left(S_T(\rho), S_T(\tilde{\rho})\right) = \exp\left(\frac{\tau}{T^2}\sum_{t,t'=1}^{T}\exp\left(\frac{\gamma}{n}\sum_{i=1}^{n}\mathrm{tr}\left(\sigma_i^{(t)}\tilde{\sigma}_i^{(t')}\right)\right)\right), \tag{6}$$

where $S_T(\rho)$ and $S_T(\tilde{\rho})$ are classical shadow representations of ρ and $\tilde{\rho}$, respectively. The computation time for the inner product is $\mathcal{O}(nT^2)$, linear in the system size n and quadratic in T, the number of copies of each quantum state which are measured to construct the classical shadow.

B. Rigorous guarantee

By statistical analysis, we can establish a rigorous guarantee for the classical ML model $\langle \alpha, \phi^{(\mathrm{shadow})}(S_T(\rho))\rangle$, where α is the trainable vector defining the classifying hyperplane. The result is the following theorem proven in Appendix J.

Theorem 2 (Classifying quantum phases of matter; informal)**.** *If there is a nonlinear function of few-body reduced density matrices that classifies phases, then the classical algorithm can learn to classify these phases accurately. The required amount of training data and computation time scale polynomially in system size.*

If there is an efficient procedure based on *few-body reduced density matrices* for classifying phases, the proposed ML algorithm is guaranteed to find the procedure efficiently. This includes local order parameters for classifying symmetry breaking phases, and topological entanglement entropy in a sufficiently large local region for partially classifying topological phases [102, 110]. We expect that, to classify topological phases accurately, the classical ML will need access to local regions that are sufficiently large compared to the correlation length, and as we approach the phase boundary, the correlation length increases. As a result, the classifying function for topological phases may depend on r-body subsystems with a larger r, and the amount of training data and computation time required would increase accordingly. Note that the classical ML not only classifies phases accurately, but also constructs a classifying function explicitly.

Our classical ML model may also be useful for classifying and understanding symmetry-protected topological (SPT) phases. SPT phases are characterized much like topological phases, but with the additional constraint that all structures involved (states, Hamiltonians, and quantum circuits) respect a particular symmetry. It is reasonable to expect that an SPT phase can be identified by examining reduced density matrices on constant-size regions [49, 75, 112, 136, 137, 155], where the size of the region is large compared to the correlation length. The existence of classifying functions based on reduced matrices have been rigorously established in some cases [12, 14, 80, 96, 101, 161, 162, 188]. In Appendix K, we prove that the ML algorithm is guaranteed to efficiently classify a class of gapped spin-1 chains in one dimension. For more general SPT phases, the ML algorithm should be able to corroborate known classification schemes, determine new and potentially more compact classifiers, and shed light on interacting SPT phases in two or more dimensions for which complete classification schemes have not yet been firmly established.

V. NUMERICAL EXPERIMENTS

We have conducted numerical experiments assessing the performance of classical ML algorithms in some practical settings. The results demonstrate that our theoretical claims carry over to practice, with the results sometimes turning out even better than our guarantees suggest.

A. Predicting ground state properties

For predicting ground states, we consider classical ML models encompassed by Eq. (2). We examine various metrics $\kappa(x, x_\ell)$ equivalent to training neural networks with large hidden layers [92, 125] or training kernel methods [46, 122]. We find the best ML model and the hyperparameters using a validation set to minimize root-mean-square error (RMSE) and report the predictions on a test set. The full details of the models and hyperparameters, as well as their comparison, are given in Appendix D.

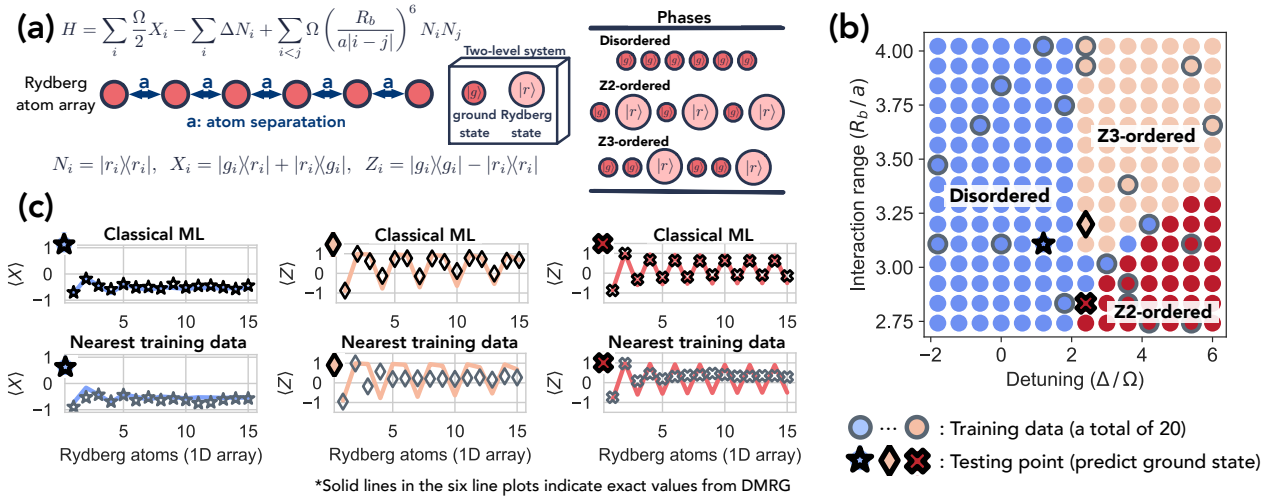

Figure 2: Numerical experiment for predicting ground-state properties in a 1D Rydberg atom system with 51 atoms. (a) Hamiltonian. Illustration of the Rydberg array geometry, Hamiltonian, and phases. (b) Phase diagram. The system's three distinct phases [20] are characterized by two order parameters (for Z_2 and Z_3 orders). Training data are enclosed by gray circles, and three specific testing points are indicated by the star, diamond, and cross, respectively. (c) Local expectation values. We use classical ML (the best model is selected from a set of ML models) to predict the expectation values of Pauli operators X_i and Z_i for each atom at the three testing points. We compare with "predictions" obtained from the training data nearest to the testing points. The markers denote predicted values, while the solid lines denote exact values obtained from DMRG. Additional predictions are shown in Appendix D.1.

Rydberg atom chain — Our first example is a system of trapped Rydberg atoms [26, 61], a programmable and highly-controlled platform for Ising-type quantum simulations [20, 53, 57, 107, 145, 151]. Following [20], we consider a one-dimensional array of $n = 51$ atoms, with each atom effectively described as a two-level system composed of a ground state $|g\rangle$ and a highly-excited Rydberg state $|r\rangle$. The atomic chain is characterized by a Hamiltonian $H(x)$ (given in Figure 2(a)) whose parameters are the laser detuning $x_1 = \Delta/\Omega$ and the interaction range $x_2 = R_b/a$. The phase diagram (shown in Figure 2(b)) features a disordered phase and several broken-symmetry phases, stemming from the competition between the detuning and the Rydberg blockade (arising from the repulsive Van der Waals interactions).

We trained a classical ML model using 20 randomly chosen values of the parameter $x = (x_1, x_2)$; these values are indicated by gray circles in Figure 2(b). For each such x, an approximation to the exact ground state was found using DMRG [181] based on the formalism of matrix product states (MPS) [152]. For each MPS, we performed $T = 500$ randomized Pauli measurements to construct a classical shadow. The classical ML then predicted classical representations at the testing points in the parameter space, and these predicted classical representations were used to estimate expectation values of local observables at the testing points.

Predictions for expectation values of Pauli operators Z_i and X_i at the testing points are shown in Figure 2(c), and found to agree well with exact values obtained from the DMRG computation of the ground state at the testing points. Additional predictions can be found in Appendix D.1. Also shown are results from a more naive procedure, in which properties are predicted using only the data at the point in the training set which is closest to the testing point. The naive procedure predicts poorly, illustrating that the considered classical ML model effectively leverages the data from multiple points in the training set.

This example corroborates our expectation that classical machines can learn to efficiently predict ground state representations. An important caveat is that the rigorous guarantee in Theorem 1 applies only when the training points and the testing points are sampled from the same phase, while in this example the training data includes values of x from three different phases. Nevertheless, our numerics show that classical machines can still learn to predict well.

2D antiferromagnetic Heisenberg model — Our next example is the two-dimensional antiferromagnetic Heisenberg model. Spin-$\frac{1}{2}$ particles (i.e. qubits) occupy sites on a square lattice, and for each pair (ij) of neighboring sites the Hamiltonian contains a term $J_{ij}\,(X_iX_j + Y_iY_j + Z_iZ_j)$ where the couplings $\{J_{ij}\}$ are uniformly sampled from the unit interval $[0, 2]$. The parameter x is a list of all J_{ij} couplings; hence in this case the dimension of the parameter space is $m = O(n)$, where n is the number of qubits. The Hamiltonian $H(x)$ on a 5×5 lattice is shown in Figure 3(a).

We trained a classical ML model using 90 randomly chosen values of the parameter $x = \{J_{ij}\}$. For each such x, the exact ground state was found using DMRG, and we simulated $T = 500$ randomized Pauli measurements to construct a classical shadow. The classical ML predicted the classical representation at new values of x,

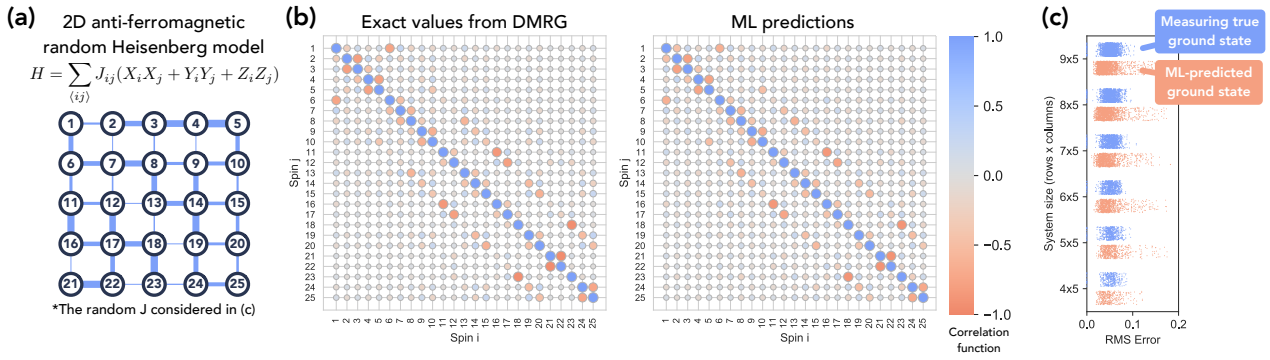

Figure 3: Numerical experiment for predicting ground state properties in the 2D antiferromagnetic Heisenberg model. (a) Hamiltonian. Illustration of the Heisenberg model geometry and Hamiltonian. We consider random couplings J_{ij}, sampled uniformly from $[0, 2]$. A particular instance is shown, with coupling strength indicated by the thickness of the edges connecting lattice points. (b) Two-point correlator. Exact values and ML predictions of the expectation value of the correlation function $C_{ij} = \frac{1}{3}(X_iX_j + Y_iY_j + Z_iZ_j)$ for all spin pairs (ij) in the lattice, for the Hamiltonian instance shown in (a). The absolute value of C_{ij} is represented by the size of each circle, while the circle's color indicates the actual value. (c) Prediction error. Each blue point indicates the root-mean-square error (averaged over Heisenberg model instances) of the correlation function for a particular pair (ij), where the estimate of C_{ij} is obtained using a classical shadow with $T = 500$ randomized Pauli measurements of the true ground state. Red points indicate errors in ML predictions for C_{ij}.

and we used the predicted classical representation to estimate a two-body correlation function, the expectation value of $C_{ij} = \frac{1}{3}(X_iX_j + Y_iY_j + Z_iZ_j)$, for each pair of qubits (ij). In Figure 3(b), the predicted and actual values of the correlation function are displayed for a particular value of x, showing reasonable agreement.

Figure 3(c) shows the prediction performance for all pairs of spins and for variable system size. Each red point in the plot represents the RMSE in the correlation function estimated using our predicted classical representation, for a particular pair of spins and averaged over sampled values of x. For comparison, each blue point is the RMSE when the correlation function is predicted using the classical shadow obtained by measuring the actual ground state $T = 500$ times. For most correlation functions, the prediction error achieved by the best classical ML model is comparable to the error achieved by measuring the actual ground state.

B. Classifying quantum phases of matter

For classifying quantum phases of matter, we consider an unsupervised classical ML model that constructs a high-dimensional feature vector for each quantum state ρ by applying the map $\phi^{(\text{shadow})}$ given in Eq. (5) with $\tau, \gamma = 1$ to the classical shadow $S_T(\rho)$ of the quantum state ρ. We then perform a principal component analysis (PCA) [131] in the high-dimensional feature space. This can be done efficiently using the shadow kernel $k^{(\text{shadow})}$ given in Eq. (6) and the kernel PCA procedure [150]. Details are given in Appendix D.4.

Bond-alternating XXZ model — We begin by considering the bond-alternating XXZ model with $n = 300$ spins. The Hamiltonian is given in Figure 4(a); it encompasses the bond-alternating Heisenberg model ($\delta = 1$) and the bosonic version of the Su-Schrieffer-Heeger model ($\delta = 0$) [158]. The phase diagram in Figure 4(b) is obtained by evaluating the partial reflection many-body topological invariant [56, 136]. There are three different phases: trivial, symmetry-protected topological, and symmetry broken.

For each value of J and δ considered, we construct the exact ground state using DMRG, and find its classical shadow by performing randomized Pauli measurement $T = 500$ times. We then consider a two-dimensional principal subspace of the high-dimensional feature space found by the unsupervised ML based on the shadow kernel, which is visualized in Figure 4(c, d). We can clearly see that the different phases are well separated in the principal subspace. This shows that even without any phase labels on the training data, the ML model can already classify the phases accurately. Hence, when trained with only a small amount of labeled data, the ML model will be able to correctly classify the phases as guaranteed by Theorem 2.

Distinguishing a topological phase from a trivial phase — We consider the task of distinguishing the toric code topological phase from the trivial phase in a system of $n = 200$ qubits. Figure 5(a) illustrates the sampled topological and trivial states. We generate representatives of the nontrival topological phase by applying low-depth geometrically local random quantum circuits to Kitaev's toric code state [100] with code distance 10, and we generate representatives of the trivial phase by applying random circuits to a product state.

Randomized Pauli measurements are performed $T = 500$ times to convert the states to their classical shadows, and these classical shadows are mapped to feature vectors in the high-dimensional feature space using the feature

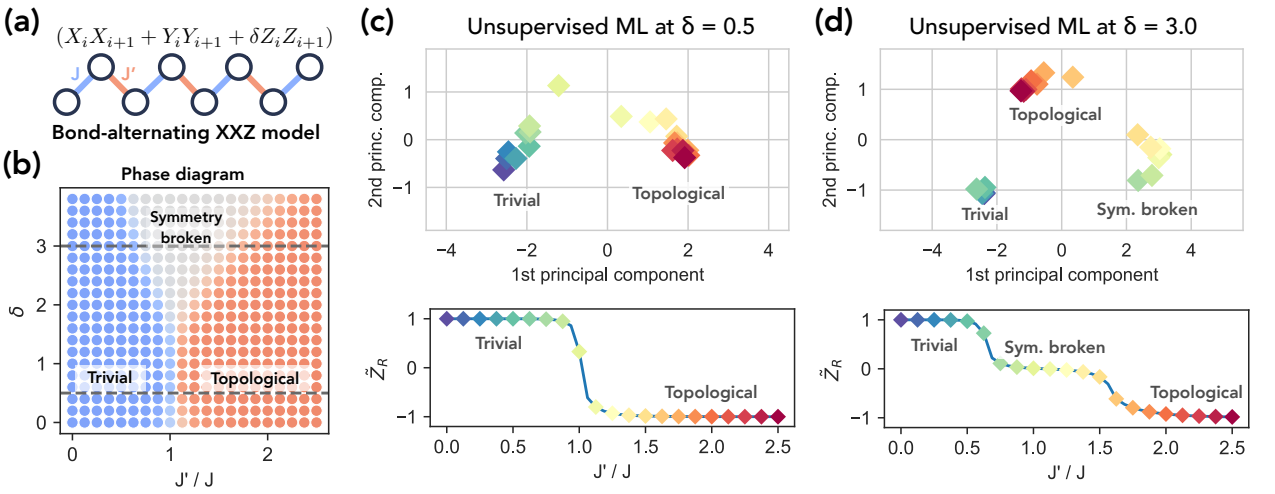

Figure 4: Numerical experiments for classifying quantum phases in the bond-alternating XXZ model. (a) Hamiltonian. Illustration of the model — a one-dimensional qubit chain, where the coefficient of $(X_iX_{i+1}+Y_iY_{i+1}+\delta Z_iZ_{i+1})$ alternates between J and J'. (b) Phase diagram. The system's three distinct phases are characterized by the many-body topological invariant $\tilde{Z}_R$ discussed in Refs. [56, 136]. Blue denotes $\tilde{Z}_R = 1$, red denotes $\tilde{Z}_R = -1$, and gray denotes $\tilde{Z}_R \approx 0$. (c, d) Unsupervised phase classification. Bottom panels: $\tilde{Z}_R$ vs. J'/J at cross sections (c) $\delta = 0.5$ and (d) $\delta = 3.0$ of the phase diagram. Top panels: visualization of the quantum states projected to two dimensions using the unsupervised ML (PCA with shadow kernel). In all panels, colors of the points indicate the value of J'/J, indicating that the two phases naturally cluster in the expressive feature space.

map $\phi^{(\text{shadow})}$. Figure 5(b) displays a one-dimensional projection of the feature space using the unsupervised classical ML for various values of the circuit depth, indicating that the phases become harder to distinguish as the circuit depth increases. In Figure 5(c), we show the classification accuracy of the unsupervised classical ML model. We also compare to training convolutional neural networks (CNN) that use measurement outcomes from the Pauli-6 POVM [33] as input to learn an observable for classifying the phases. Since Proposition 2 establishes that no observable (even a global one) can classify topological phases, this CNN approach is doomed to fail. On the other hand, if the CNN takes classical shadow representations as input, then it can learn nonlinear functions and successfully classify the phases.

VI. OUTLOOK

We have rigorously established that classical machine learning (ML) algorithms, informed by data collected in physical experiments, can effectively address quantum many-body problems. These results boost our hopes that classical ML trained on experimental data can solve practical problems in chemistry and materials science that would be too hard to solve using classical processing alone.

Our arguments build on the concept of a classical shadow derived from randomized Pauli measurements. We expect, though, that other succinct classical representations of quantum states could be exploited by classical ML with similarly powerful results. For example, some currently available quantum simulators are highly programmable, but lack the local control needed to perform arbitrary single-qubit Pauli measurements. Instead, after preparing a many-body quantum state of interest, one might switch rapidly to a different Hamiltonian and then allow the state to evolve for a short time before performing a computational basis measurement. How can we make use of that measurement data to predict properties reliably? Answering such questions, and thereby expanding the reach of near-term programmable quantum platforms, will be an important goal for future research [47, 86].

Viewed from a broader perspective, by illustrating how experimental data can be exploited to make accurate predictions about features of quantum systems that have never been studied directly, our work exemplifies a potentially powerful methodology for advancing the physical sciences. With further theoretical developments, perhaps we can learn how to use experimental data that is already routinely available to accelerate the discovery of new chemical compounds and materials with remarkable properties that could benefit humanity.

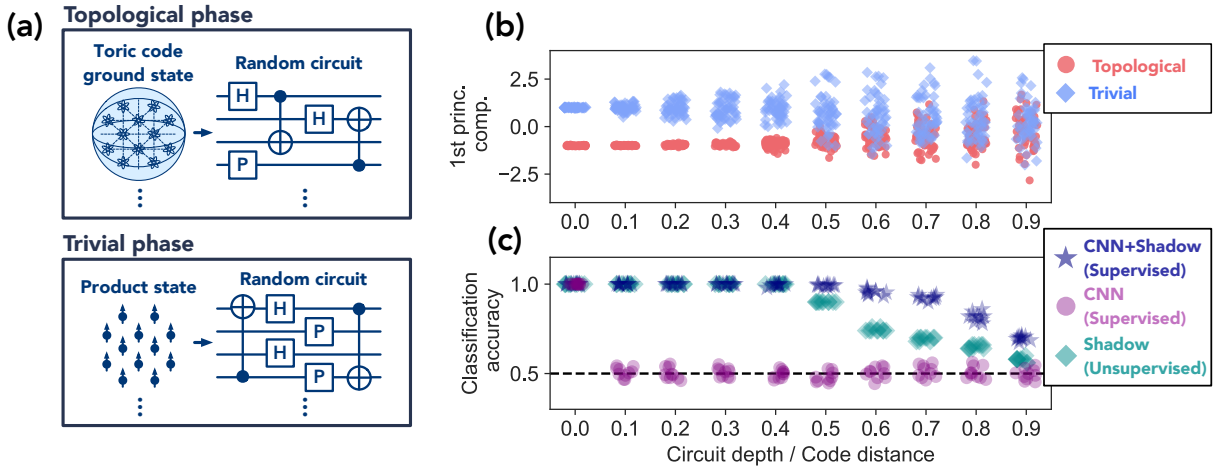

Figure 5: Numerical experiments for distinguishing between trivial and topological phases. (a) State generation. Trivial or topological states are generated by applying local random quantum circuits of some circuit depth to a product state or exactly-solved topological state, respectively. (b) Unsupervised phase classification. visualization of the quantum states projected to one dimension using the unsupervised ML (PCA with shadow kernel), shown for varying circuit depth (divided by the "code distance" 10, which quantifies the depth at which the topological properties are washed out). The feature space is sufficiently expressive to resolve the phases for a small enough depth without training, with classification becoming more difficult as the depth increases. (c) Classification accuracy for three ML algorithms described in Section V B.

Acknowledgments:

The authors thank Nir Bar-Gill, Juan Carrasquilla, Sitan Chen, Yifan Chen, Matthew Fishman, Scott Glancy, Jeongwan Haah, Felix Kueng, Jarrod McClean, Spiros Michalakis, Jacob Taylor, Yuan Su, and Thomas Vidick for valuable input and inspiring discussions. In particular, HH would like to thank Andreas Elben for providing the code on bond-alternating XXZ model. The numerical simulations were performed on AWS EC2 computing infra-structure, using the software packages ITensors [65] and PastaQ [64]. HH is supported by the J. Yang & Family Foundation. JP acknowledges funding from the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, (DE-NA0003525, DE-SC0020290), and the National Science Foundation (PHY-1733907). The Institute for Quantum Information and Matter is an NSF Physics Frontiers Center. Contributions to this work by NIST, an agency of the US government, are not subject to US copyright. Any mention of commercial products does not indicate endorsement by NIST. V.V.A. thanks Olga Albert, Halina and Ryhor Kandratsenia, as well as Tatyana and Thomas Albert for providing daycare support throughout this work.

[1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(5):052328, 2004.

[2] N. Abrahamsen. A polynomial-time algorithm for ground states of spin trees. *arXiv preprint arXiv:1907.04862*, 2019.

[3] N. Abrahamsen. Sub-exponential algorithm for 2d frustration-free spin systems with gapped subsystems. *arXiv preprint arXiv:2004.02850*, 2020.

[4] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki. Valence bond ground states in isotropic quantum antiferromagnets. *Commun. Math. Phys.*, 115(3):477–528, 1988.

[5] I. Affleck and E. H. Lieb. A proof of part of Haldane's conjecture on spin chains. *Lett. Math. Phys.*, 12(1):57–69, 1986.

[6] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM review*, 50(4):755–787, 2008.

[7] N. Andrejevic, J. Andrejevic, C. H. Rycroft, and M. Li. Machine learning spectral indicators of topology. *arXiv preprint arXiv:2003.00994*, 2020.

[8] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar. Sample-efficient learning of interacting quantum systems. *Nat. Phys.*, 2021.

[9] I. Arad, Z. Landau, U. Vazirani, and T. Vidick. Rigorous rg algorithms and area laws for low energy eigenstates in 1d. *Commun. Math. Phys.*, 356(1):65–105, 2017.

[10] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *NeurIPS*, pages 8139–8148, 2019.

[11] D. P. Arovas. *Lecture Notes on Group Theory in Physics*. online notes. Available at https://courses.physics.ucsd.edu/2016/Spring/physics220/LECTURES/GROUP_THEORY.pdf.

[12] S. Bachmann, A. Bols, W. De Roeck, and M. Fraas. A Many-Body Index for Quantum Charge Transport. *Commun. Math. Phys.*, 375(2):1249–1272, 2020.

[13] S. Bachmann, S. Michalakis, B. Nachtergaele, and R. Sims. Automorphic equivalence within gapped phases of quantum lattice systems. *Commun. Math. Phys.*, 309(3):835–871, 2012.

[14] S. Bachmann and B. Nachtergaele. On Gapped Phases with a Continuous Symmetry and Boundary Operators. *J. Stat. Phys.*, 154(1-2):91–112, 2014.

[15] H. Bacry, L. Michel, and J. Zak. Symmetry and classification of energy bands in crystals. In *Gr. Theor. Methods Phys.*, pages 289–308. Springer-Verlag, Berlin/Heidelberg.

[16] O. Balabanov and M. Granath. Unsupervised interpretable learning of topological indices invariant under permutations of atomic bands. *Mach. Learn. Sci. Technol.*, 2(2):025008, 2021.

[17] M. J. S. Beach, A. Golubeva, and R. G. Melko. Machine learning vortices at the kosterlitz-thouless transition. *Phys. Rev. B*, 97:045207, 2018.

[18] F. Becca and S. Sorella. *Quantum Monte Carlo Approaches for Correlated Systems*. Cambridge University Press, 2017.

[19] B. A. Bernevig and T. L. Hughes. *Topological Insulators and Topological Superconductors*. Princeton University Press, Princeton and Oxford, 2013.

[20] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature*, 551(7682):579–584, 2017.

[21] 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(9):921–924, 2019.

[22] S. Boyd, S. P. Boyd, and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[23] B. Bradlyn, L. Elcoro, J. Cano, M. G. Vergniory, Z. Wang, C. Felser, M. I. Aroyo, and B. A. Bernevig. Topological quantum chemistry. *Nature*, 547(7663):298–305, 2017.

[24] S. Bravyi, D. P. DiVincenzo, R. I. Oliveira, and B. M. Terhal. The Complexity of Stoquastic Local Hamiltonian Problems. *arXiv e-prints*, pages quant–ph/0606140, 2006.

[25] S. Bravyi, M. B. Hastings, and F. Verstraete. Lieb-robinson bounds and the generation of correlations and topological quantum order. *Phys. Rev. Lett.*, 97(5):050401, 2006.

[26] A. Browaeys and T. Lahaye. Many-body physics with individually controlled Rydberg atoms. *Nat. Phys.*, 16(2):132–142, 2020.

[27] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[28] J. Cano and B. Bradlyn. Band Representations and Topological Quantum Chemistry. *Annu. Rev. Condens. Matter Phys.*, 12(1):225–246, 2021.

[29] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, 2019.

[30] G. Carleo and M. Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[31] J. Carrasquilla. Machine learning for quantum matter. *Adv. Phys.: X*, 5(1):1797528, 2020.

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

[33] J. Carrasquilla, G. Torlai, R. G. Melko, and L. Aolita. Reconstructing quantum states with generative models. *Nat. Mach. Intell.*, 1(3):155, 2019.

[34] D. Ceperley and B. Alder. Quantum Monte Carlo. *Science*, 231(4738):555–560, 1986.

[35] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[36] S. Chen, W. Yu, P. Zeng, and S. T. Flammia. Robust shadow estimation. *arXiv preprint arXiv:2011.09636*, 2020.

[37] 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(15):155114, 2013.

[38] X. Chen, Z.-C. Gu, and X.-G. Wen. Classification of gapped symmetric phases in one-dimensional spin systems. *Phys. Rev. B*, 83(3):035107, 2011.

[39] J. Choi, A. L. Shaw, I. S. Madjarov, X. Xie, J. P. Covey, J. S. Cotler, D. K. Mark, H.-Y. Huang, A. Kale, H. Pichler, et al. Emergent randomness and benchmarking from many-body quantum chaos. *arXiv preprint arXiv:2103.03535*, 2021.

[40] F. Chollet et al. Keras. 2015. Available at https://github.com/fchollet/keras.

[41] K. Choo, G. Carleo, N. Regnault, and T. Neupert. Symmetries and many-body excitations with neural-network quantum states. *Phys. Rev. Lett.*, 121:167204, 2018.

[42] K. Choo, A. Mezzacapo, and G. Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nat. Commun.*, 11(1):2368, May 2020.

[43] N. Claussen, B. A. Bernevig, and N. Regnault. Detection of topological materials with machine learning. *Phys. Rev. B*, 101(24):245117, 2020.

[44] I. Cong, S. Choi, and M. D. Lukin. Quantum convolutional neural networks. *Nat. Phys.*, 15(12):1273–1278, 2019.

[45] P. Corboz. Variational optimization with infinite projected entangled-pair states. *Phys. Rev. B*, 94:035133, 2016.

[46] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

[47] J. S. Cotler, D. K. Mark, H.-Y. Huang, F. Hernandez, J. Choi, A. L. Shaw, M. Endres, and S. Choi. Emergent quantum state designs from individual many-body wavefunctions. *arXiv preprint arXiv:2103.03536*, 2021.

[48] J. S. Cotler and F. Wilczek. Quantum overlapping tomography. *Phys. Rev. Lett.*, 124(10):100401, 2020.

[49] H. Dehghani, Z.-P. Cian, M. Hafezi, and M. Barkeshli. Extraction of the many-body Chern number from a single wave function. *Phys. Rev. B*, 103(7):075102, 2021.

[50] D.-L. Deng, X. Li, and S. Das Sarma. Machine learning topological states. *Phys. Rev. B*, 96:195145, 2017.

[51] D.-L. Deng, X. Li, and S. Das Sarma. Quantum entanglement in neural network states. *Phys. Rev. X*, 7:021021, 2017.

[52] S. S. Du, K. Hou, B. Póczos, R. Salakhutdinov, R. Wang, and K. Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *arXiv preprint arXiv:1905.13192*, 2019.

[53] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, S. Choi, S. Sachdev, M. Greiner, V. Vuletic, and M. D. Lukin. Quantum Phases of Matter on a 256-Atom Programmable Quantum Simulator. *arXiv e-prints*, page arXiv:2012.12281, 2020.

[54] A. Elben, R. Kueng, H.-Y. Huang, R. van Bijnen, C. Kokail, M. Dalmonte, P. Calabrese, B. Kraus, J. Preskill, P. Zoller, and B. Vermersch. Mixed-state entanglement from local randomized measurements. *Phys. Rev. Lett.*, 125:200501, 2020.

[55] A. Elben, B. Vermersch, C. F. Roos, and P. Zoller. Statistical correlations between locally randomized measurements: A toolbox for probing entanglement in many-body quantum states. *Phys. Rev. A*, 99(5):052323, 2019.

[56] 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. *Science advances*, 6(15):eaaz3666, 2020.

[57] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin. Atom-by-atom assembly of defect-free one-dimensional cold atom arrays. *Science*, 354(6315):1024–1027, 2016.

[58] T. J. Evans, R. Harper, and S. T. Flammia. Scalable bayesian hamiltonian learning. *arXiv preprint arXiv:1912.07636*, 2019.

[59] H. G. Evertz. The loop algorithm. *Adv. Phys.*, 52(1):1–66, 2003.

[60] H. Fawzi, J. Saunderson, and P. A. Parrilo. Semidefinite Approximations of the Matrix Logarithm. *Found. Comput. Math.*, 19(2):259–296, apr 2019.

[61] P. Fendley, K. Sengupta, and S. Sachdev. Competing density-wave orders in a one-dimensional hard-boson model. *Phys. Rev. B*, 69:075106, 2004.

[62] F. Ferrari, F. Becca, and J. Carrasquilla. Neural gutzwiller-projected variational wave functions. *Phys. Rev. B*, 100:125131, 2019.

[63] A. J. Ferris and G. Vidal. Perfect sampling with unitary tensor networks. *Phys. Rev. B*, 85:165146, 2012.

[64] M. Fishman and G. Torlai. PastaQ: A package for simulation, tomography and analysis of quantum computers. 2020. Available at https://github.com/GTorlai/PastaQ.jl.

[65] M. Fishman, S. R. White, and E. Miles Stoudenmire. The ITensor Software Library for Tensor Network Calculations. *arXiv e-prints*, page arXiv:2007.14822, 2020.

[66] S. T. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. *New J. Phys.*, 14(9):095022, 2012.

[67] E. N. Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

[68] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[69] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac. Neural-network quantum states, string-bond

states, and chiral topological states. *Phys. Rev. X*, 8:011006, 2018.
[70] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber. Unsupervised identification of topological phase transitions using predictive models. *New J. Phys.*, 22(4):045003, 2020.
[71] Z.-C. Gu and X.-G. Wen. Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order. *Phys. Rev. B*, 80(15):155131, 2009.
[72] M. Guţă, J. Kahn, R. Kueng, and J. A. Tropp. Fast state tomography with optimal error bounds. *Journal of Physics A: Mathematical and Theoretical*, 53(20):204001, 2020.
[73] J. Haah, A. W. Harrow, Z. Ji, X. Wu, and N. Yu. Sample-optimal tomography of quantum states. *IEEE Trans. Inf. Theory*, 63(9):5628–5641, 2017.
[74] C. Hadfield, S. Bravyi, R. Raymond, and A. Mezzacapo. Measurements of quantum hamiltonians with locally-biased classical shadows. *arXiv:2006.15788*, 2020.
[75] J. Haegeman, D. Pérez-García, I. Cirac, and N. Schuch. Order Parameter for Symmetry-Protected Phases in One Dimension. *Phys. Rev. Lett.*, 109(5):050402, 2012.
[76] R. Haghshenas, M. J. O'Rourke, and G. K.-L. Chan. Conversion of projected entangled pair states into a canonical form. *Phys. Rev. B*, 100:054404, 2019.
[77] F. Haldane. Continuum dynamics of the 1-D Heisenberg antiferromagnet: Identification with the O(3) nonlinear sigma model. *Phys. Lett. A*, 93(9):464–468, 1983.
[78] M. B. Hastings. Locality in quantum systems. *arXiv:1008.5137*.
[79] M. B. Hastings and T. Koma. Spectral gap and exponential decay of correlations. *Communications in mathematical physics*, 265(3):781–804, 2006.
[80] M. B. Hastings and S. Michalakis. Quantization of Hall Conductance for Interacting Electrons on a Torus. *Commun. Math. Phys.*, 334(1):433–471, 2015.
[81] M. B. Hastings and X.-G. Wen. Quasiadiabatic continuation of quantum states: The stability of topological ground-state degeneracy and emergent gauge invariance. *Phys. Rev. B*, 72(4):045141, 2005.
[82] E. Hazan, T. Koren, and N. Srebro. Beating sgd: Learning svms in sublinear time. In *NIPS*, pages 1233–1241. Citeseer, 2011.
[83] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla. Recurrent neural network wave functions. *Physical Review Research*, 2(2):023358, 2020.
[84] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136:B864–B871, 1964.
[85] A. S. Holevo. Some estimates of the information transmitted by quantum communication channels. *Probl. Inf. Transm.*, 9(3):177–183, 1973.
[86] H.-Y. Hu and Y.-Z. You. Hamiltonian-driven shadow tomography of quantum states. *arXiv preprint arXiv:2102.10132*, 2021.
[87] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean. Power of data in quantum machine learning. *Nat. Commun.*, 12(1):1–9, 2021.
[88] H.-Y. Huang, R. Kueng, and J. Preskill. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.*, 16:1050—1057, 2020.
[89] H.-Y. Huang, R. Kueng, and J. Preskill. Efficient estimation of Pauli observables by derandomization. *arXiv preprint arXiv:2103.07510*, 2021.
[90] H.-Y. Huang, R. Kueng, and J. Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Phys. Rev. Lett.*, 126:190505, 2021.
[91] K. Hyatt and E. M. Stoudenmire. DMRG Approach to Optimizing Two-Dimensional Tensor Networks. *arXiv e-prints*, page arXiv:1908.08833, 2019.
[92] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, pages 8571–8580, 2018.
[93] S. Jiang, S. Lu, and D.-L. Deng. Vulnerability of machine learning phases of matter. *arXiv preprint arXiv:1910.13453*, 2019.
[94] T. Joachims. *Making Large-Scale Support Vector Machine Learning Practical*, page 169–184. MIT Press, Cambridge, MA, USA, 1999.
[95] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac. Classical simulation of infinite-size quantum lattice systems in two spatial dimensions. *Phys. Rev. Lett.*, 101:250602, 2008.
[96] A. Kapustin and N. Sopenko. Hall conductance and the statistics of flux insertions in gapped interacting lattice systems. *J. Math. Phys.*, 61(10):101901, 2020.
[97] Z. Karnin, E. Liberty, S. Lovett, R. Schwartz, and O. Weinstein. Unsupervised svms: On the complexity of the furthest hyperplane problem. In S. Mannor, N. Srebro, and R. C. Williamson, editors, *Proceedings of the 25th Annual Conference on Learning Theory*, volume 23 of *Proceedings of Machine Learning Research*, pages 2.1–2.17, Edinburgh, Scotland, 2012. JMLR Workshop and Conference Proceedings.
[98] R. K. Kaul, R. G. Melko, and A. W. Sandvik. Bridging lattice-scale physics and continuum field theory with quantum monte carlo simulations. *Annual Review of Condensed Matter Physics*, 4(1):179–215, 2013.
[99] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[100] A. Y. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.
[101] A. Y. Kitaev. Anyons in an exactly solved model and beyond. *Ann. Phys.*, 321(1):2–111, 2006.
[102] A. Y. Kitaev and J. Preskill. Topological entanglement entropy. *Phys. Rev. Lett.*, 96(11):110404, 2006.
[103] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.
[104] D. E. Koh and S. Grewal. Classical shadows with noise. *arXiv preprint arXiv:2011.11580*, 2020.

[105] W. Kohn. Nobel lecture: Electronic structure of matter—wave functions and density functionals. *Rev. Mod. Phys.*, 71:1253–1266, 1999.

[106] D. C. Kozen. *The design and analysis of algorithms*. Springer Science & Business Media, 1992.

[107] H. Labuhn, D. Barredo, S. Ravets, S. de Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys. Tunable two-dimensional arrays of single rydberg atoms for realizing quantum ising models. *Nature*, 534:667 EP –, 2016.

[108] Z. Landau, U. Vazirani, and T. Vidick. A polynomial time algorithm for the ground state of one-dimensional gapped local hamiltonians. *Nat. Phys.*, 11(7):566–569, 2015.

[109] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[110] M. Levin and X.-G. Wen. Detecting topological order in a ground state wave function. *Phys. Rev. Lett.*, 96(11):110405, 2006.

[111] M. Lewin, E. H. Lieb, and R. Seiringer. The local density approximation in density functional theory. *Pure Appl. Anal.*, 2(1):35–73, 2020.

[112] H. Li and F. D. M. Haldane. Entanglement Spectrum as a Generalization of Entanglement Entropy: Identification of Topological Order in Non-Abelian Fractional Quantum Hall Effect States. *Phys. Rev. Lett.*, 101(1):010504, 2008.

[113] Z. Li, R. Wang, D. Yu, S. S. Du, W. Hu, R. Salakhutdinov, and S. Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

[114] D. Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11:329–343, 1982.

[115] E. H. Lieb and D. W. Robinson. The finite group velocity of quantum spin systems. *Commun. Math. Phys.*, 28(3):251–257, sep 1972.

[116] D. Luo, Z. Chen, K. Hu, Z. Zhao, V. Mikyoung Hur, and B. K. Clark. Gauge Invariant Autoregressive Neural Networks for Quantum Lattice Models. *arXiv e-prints*, page arXiv:2101.07243, 2021.

[117] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.*, 18(2):023023, 2016.

[118] J. Mercer. Functions of positive and negative type, and their connection the theory of integral equations. *Philos. T. Roy. Soc. A*, 209(441-458):415–446, jan 1909.

[119] Y. Ming, C.-T. Lin, S. D. Bartlett, and W.-W. Zhang. Quantum topology identification with deep neural networks and quantum walks. *npj Comput. Mater.*, 5(1):88, 2019.

[120] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. The MIT Press, 2018.

[121] S. Morawetz, I. J. S. De Vlugt, J. Carrasquilla, and R. G. Melko. U(1) symmetric recurrent neural networks for quantum state reconstruction. *arXiv e-prints*, page arXiv:2010.14514, 2020.

[122] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[123] M. Nakamura and S. Todo. Order Parameter to Characterize Valence-Bond-Solid States in Quantum Spin Chains. *Phys. Rev. Lett.*, 89(7):077204, 2002.

[124] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B*, 96:205152, 2017.

[125] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020.

[126] M. Ohliger, V. Nesme, and J. Eisert. Efficient and feasible state tomography of quantum many-body systems. *New Journal of Physics*, 15(1):015024, Jan 2013.

[127] R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Ann. Phys.*, 349:117–158, 2014.

[128] T. J. Osborne. Simulating adiabatic evolution of gapped spin systems. *Phys. Rev. A*, 75(3):032321, 2007.

[129] M. Paini and A. Kalev. An approximate description of quantum states. *arXiv preprint arXiv:1910.10543*, 2019.

[130] V. Peano, F. Sapper, and F. Marquardt. Rapid Exploration of Topological Band Structures Using Deep Learning. *Phys. Rev. X*, 11(2):021052, 2021.

[131] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[132] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011.

[133] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, 2007.

[134] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, 5:4213, 2014.

[135] H. C. Po, A. Vishwanath, and H. Watanabe. Symmetry-based indicators of band topology in the 230 space groups. *Nat. Commun.*, 8(1):50, 2017.

[136] F. Pollmann and A. M. Turner. Detection of symmetry-protected topological phases in one dimension. *Phys. Rev. B*, 86(12):125441, 2012.

[137] F. Pollmann, A. M. Turner, E. Berg, and M. Oshikawa. Entanglement spectrum of a topological phase in one dimension. *Phys. Rev. B*, 81(6):064439, 2010.

[138] Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller III. Orbnet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.*, 153(12):124111, 2020.

[139] N. Read. Topological phases and quasiparticle braiding. *Phys. Today*, 65(7):38, 2012.

[140] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg. Identifying quantum phase transitions using artificial neural networks on experimental data. *Nat. Phys.*, 15(9):917–

920, 2019.
[141] P. Rigollet and J.-C. Hütter. High dimensional statistics. *Lecture notes for course 18S997*, 813:814, 2015. Available at http://www-math.mit.edu/~rigollet/PDFs/RigNotes17.pdf.
[142] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
[143] J. F. Rodriguez-Nieva and M. S. Scheurer. Identifying topological order through unsupervised machine learning. *Nat. Phys.*, 15(8):790–795, 2019.
[144] A. W. Sandvik. Stochastic series expansion method with operator-loop update. *Phys. Rev. B*, 59:R14157–R14160, 1999.
[145] P. Schauß, J. Zeiher, T. Fukuhara, S. Hild, M. Cheneau, T. Macrì, T. Pohl, I. Bloch, and C. Gross. Crystallization in ising quantum magnets. *Science*, 347(6229):1455–1458, 2015.
[146] M. S. Scheurer and R.-J. Slager. Unsupervised Machine Learning and Band Topology. *Phys. Rev. Lett.*, 124(22):226401, 2020.
[147] F. Schindler, N. Regnault, and T. Neupert. Probing many-body localization with neural networks. *Phys. Rev. B*, 95:245134, 2017.
[148] G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio. From DFT to machine learning: recent approaches to materials science–a review. *J. Phys. Mater.*, 2(3):032001, 2019.
[149] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.
[150] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.
[151] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys. Programmable quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms. *arXiv e-prints*, page arXiv:2012.12268, 2020.
[152] U. Schollwoeck. The density-matrix renormalization group in the age of matrix product states. *Ann. Phys.*, 326(1):96 – 192, 2011. January 2011 Special Issue.
[153] N. Schuch and F. Verstraete. Computational complexity of interacting electrons and fundamental limitations of density functional theory. *Nat. Phys.*, 5(10):732–735, 2009.
[154] K. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer. Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nat. Commun.*, 10(1):1–10, 2019.
[155] H. Shapourian, K. Shiozaki, and S. Ryu. Many-Body Topological Invariants for Fermionic Symmetry-Protected Topological Phases. *Phys. Rev. Lett.*, 118(21):216402, 2017.
[156] E. Stoudenmire and S. R. White. Studying two-dimensional systems with the density matrix renormalization group. *Annu. Rev. Condens.*, 3(1):111–128, 2012.
[157] G. Struchalin, Y. A. Zagorovskii, E. Kovlakov, S. Straupe, and S. Kulik. Experimental estimation of quantum state properties from classical shadows. *PRX Quantum*, 2:010307, 2021.
[158] W. Su, J. Schrieffer, and A. J. Heeger. Solitons in polyacetylene. *Phys. Rev. Lett.*, 42(25):1698, 1979.
[159] N. Sun, J. Yi, P. Zhang, H. Shen, and H. Zhai. Deep learning topological invariants of band insulators. *Phys. Rev. B*, 98(8):085402, 2018.
[160] A. Szasz, J. Motruk, M. P. Zaletel, and J. E. Moore. Chiral spin liquid phase of the triangular lattice hubbard model: A density matrix renormalization group study. *Phys. Rev. X*, 10:021042, 2020.
[161] H. Tasaki. Topological phase transition and z2 index for s=1 quantum spin chains. *Phys. Rev. Lett.*, 121:140604, 2018.
[162] H. Tasaki. *Physics and Mathematics of Quantum Many-Body Systems*. Graduate Texts in Physics. Springer International Publishing, Cham, 2020.
[163] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo. Neural-network quantum state tomography. *Nat. Phys.*, 14(5):447–450, 2018.
[164] G. Torlai and R. G. Melko. Learning thermodynamics with Boltzmann machines. *Physical Review B*, 94(16):165134, 2016.
[165] G. Torlai, B. Timar, E. P. L. van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres. Integrating neural networks with a quantum simulator for state reconstruction. *Phys. Rev. Lett.*, 123:230504, 2019.
[166] K. Totsuka and M. Suzuki. Matrix formalism for the VBS-type models and hidden order. *J. Phys. Condens. Matter*, 7(8):1639–1662, 1995.
[167] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math*, 12(4):389–434, 2012.
[168] M. Troyer and U.-J. Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Phys. Rev. Lett.*, 94:170201, 2005.
[169] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.
[170] S. J. van Enk and C. W. J. Beenakker. Measuring $\mathrm{Tr}\rho^n$ on single copies of ρ using random measurements. *Phys. Rev. Lett.*, 108:110503, 2012.
[171] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber. Learning phase transitions by confusion. *Nat. Phys.*, 13:435, 2017.
[172] L. Vanderstraeten, J. Haegeman, P. Corboz, and F. Verstraete. Gradient methods for variational optimization of projected entangled-pair states. *Phys. Rev. B*, 94:155123, 2016.
[173] B. Vermersch, A. Elben, M. Dalmonte, J. I. Cirac, and P. Zoller. Unitary n -designs via random quenches in atomic

Hubbard and spin models: Application to the measurement of Rényi entropies. *Phys. Rev. A*, 97(2):023604, 2018.
[174] R. Vershynin. *High-dimensional probability*, volume 47 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2018.
[175] F. Verstraete, V. Murg, and J. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Adv. Phys.*, 57(2):143–224, 2008.
[176] T. Vieijra, C. Casert, J. Nys, W. De Neve, J. Haegeman, J. Ryckebusch, and F. Verstraete. Restricted boltzmann machines for quantum states with non-abelian or anyonic symmetries. *Phys. Rev. Lett.*, 124:097201, 2020.
[177] K. Wan and I. Kim. Fast digital methods for adiabatic state preparation. *arXiv preprint arXiv:2004.04164*, 2020.
[178] L. Wang. Discovering phase transitions with unsupervised learning. *Phys. Rev. B*, 94:195105, 2016.
[179] F. Weisz. Summability of multi-dimensional trigonometric fourier series. *arXiv preprint arXiv:1206.1789*, 2012.
[180] S. J. Wetzel. Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Phys. Rev. E*, 96:022140, 2017.
[181] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, 1992.
[182] S. R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48:10345–10356, 1993.
[183] M. M. Wilde. *Quantum Information Theory*. Cambridge University Press, Cambridge, 2nd edition, 2013.
[184] H.-Q. Wu, S.-S. Gong, and D. N. Sheng. Randomness-induced spin-liquid-like phase in the spin-$\frac{1}{2}$ J_1-J_2 triangular heisenberg model. *Phys. Rev. B*, 99:085141, 2019.
[185] B. Yu. Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.
[186] M. P. Zaletel and F. Pollmann. Isometric tensor network states in two dimensions. *Phys. Rev. Lett.*, 124:037201, 2020.
[187] B. Zeng, X. Chen, D.-L. Zhou, and X.-G. Wen. *Quantum information meets quantum matter*. Springer, 2019.
[188] Y. Zhang and E.-A. Kim. Quantum loop topography for machine learning. *Phys. Rev. Lett.*, 118(21):216401, 2017.
[189] A. Zhao, N. C. Rubin, and A. Miyake. Fermionic partial tomography via classical shadows. *arXiv preprint arXiv:2010.16094*, 2020.

Appendices: contents & navigation guide

For those unfamiliar with the classical shadow formalism [88], Appendix A provides a concise introduction and contains all the necessary information on classical shadows to follow this paper. Discussion of related literature on methods in many-body physics, in particular works for training machine learning models to solve quantum many-body problems, is given in Appendix B.

Readers interested in the numerical experiments can jump directly to Appendix C, which provides a detailed discussion on practical approaches for combining neural networks with classical shadows. The reader could then continue to Appendix D, which gives additional numerical experiments, details of numerical experiments, and example codes (in Python) for using the machine learning models.

The rest of the appendices are dedicated to providing a rigorous understanding in using classical machine learning models to solve quantum many-body problems.

a. Predicting ground states : We recommend that readers start with Appendix E, which provides the idea for why classical machine learning models can be trained to predict ground state representations of quantum systems. The detailed proof is given in Appendix F. In Appendix G, we give a fundamental lower bound in the required data size for learning to predict ground state properties for general classes of Hamiltonians. In Appendix H, we show why non-ML algorithms cannot achieve a similar guarantee as ML algorithms in predicting ground state representations.

b. Classifying phases of matter : The reader could begin with the basic proposition given in Appendix I, which shows that no (local or global) observable $\mathrm{tr}(O\rho)$ can be used to classify topological phases. This motivates the need to consider stronger machine learning models that can learn nonlinear functions in the quantum state ρ. The readers could then proceed to Appendix J, which provides a general theory for establishing provable guarantees in training machine learning models based on classical shadows to classify quantum phases of matter. In Appendix K, we briefly introduce symmetry-protected topological phases and prove that the proposed machine learning model can classify a particular subset of such phases.

APPENDIX A: Background on classical shadows

The classical shadows formalism uses randomized (single-shot) measurements to predict many properties of an unknown quantum state ρ at once [88], see also [58, 129]. The underlying idea dates back to [126] and also features prominently in [55, 170, 173]. In particular, the classical shadows formalism comes with rigorous performance guarantees in terms of approximation accuracy, classical storage, as well as data processing. Here, we focus on randomized single-qubit Pauli measurements and repeat the following procedure a total of T times: (i) prepare an independent copy of ρ; (ii) select n single qubit Pauli measurements uniformly at random (Z, X and Y occur with probability 1/3 each) and (iii), perform the associated measurement to obtain n classical bits ($+1$ if we measure 'up' and -1 if we measure 'down'). Subsequently, we store the associated post-measurement state

$$|s_1^{(t)}\rangle \otimes \cdots \otimes |s_n^{(t)}\rangle \quad \text{with} \quad |s_1^{(t)}\rangle, \ldots, |s_n^{(t)}\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\} \subset \mathbb{C}^2 \tag{A1}$$

in classical memory. This is very cheap, because there are only six possibilities for each qubit. After T repetitions, we obtain an entire collection of nT single-qubit states that we arrange in a two-dimensional array:

$$S_T(\rho) = \left\{ |s_i^{(t)}\rangle : \, i \in \{1, \ldots, n\}, t \in \{1, \ldots, T\} \right\} \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\}^{n \times T} \tag{A2}$$

The distribution of product states contains valuable information about the underlying n-qubit density matrix ρ. In fact, we can use $S_T(\rho)$ to approximate ρ via

$$\sigma_T(\rho) = \frac{1}{T} \sum_{t=1}^{T} \left(3|s_1^{(t)}\rangle\langle s_1^{(t)}| - \mathbb{I} \right) \otimes \cdots \otimes \left(3|s_n^{(t)}\rangle\langle s_n^{(t)}| - \mathbb{I} \right), \tag{A3}$$

where $\mathbb{I}$ denotes the identity matrix (here, a 2-by-2 identity). It is instructive to view this as the empirical average of T independent and identically (*iid*) random matrices. Each random matrix is an *iid* copy of $\sigma_1(\rho) = \big(3|s_1\rangle\langle s_1| - \mathbb{I}\big) \otimes \cdots \otimes \big(3|s_n\rangle\langle s_n| - \mathbb{I}\big)$. Each tensor factor is guaranteed to have eigenvalues $\lambda_+ = 2$ and $\lambda_- = -1$. This ensures that

$$\mathrm{tr}\left(\sigma_1(\rho)\right) = \mathrm{tr}\left(|s_1\rangle\langle s_1| - \mathbb{I}\right) \cdots \mathrm{tr}\left(|s_n\rangle\langle s_n| - \mathbb{I}\right) = 1 \quad \text{and} \tag{A4a}$$

$$\left\|\sigma_1(\rho)\right\|_p = \left\|3|s_1\rangle\langle s_1| - \mathbb{I}\right\|_p \cdots \left\|3|s_n\rangle\langle s_n| - \mathbb{I}\right\|_p = \left(|\lambda_+|^p + |\lambda_-|^p\right)^{n/p} = \left(2^p + 1^p\right)^{n/p}, \tag{A4b}$$

regardless of the concrete realization (and the underlying quantum state ρ). The most relevant Schatten-p norms are $\|\sigma_1(\rho)\|_1 = 3^n$, $\|\sigma_1(\rho)\|_2 = 5^{n/2}$ and $\|\sigma_1(\rho)\|_\infty = 2^n$. Note, however, that the matrix $\sigma_1(\rho)$ is never positive semidefinite.

The random matrix $\sigma_1(\rho)$ is a highly structured tensor product that can assume a total of 6^n values. Each of them reflects the outcome of performing randomly selected single-qubit Pauli measurements on the n-qubit state ρ. Let us denote these Pauli matrices by $W_1, \ldots, W_n \in \{X, Y, Z\}$ and let $o_1, \ldots, o_n \in \{\pm 1\}$ be the observed outcomes ($+1$ if we measure 'spin up' and -1 if we measure 'spin down'). Elementary reformulations and Born's rule then imply

$$\sigma_1(\rho) = \frac{1}{2}\left(\mathbb{I} + 3o_1W_1\right) \otimes \cdots \otimes \frac{1}{2}\left(\mathbb{I} + 3o_nW_n\right) \quad \text{with prob.} \quad \frac{1}{3^n}\mathrm{tr}\left(\frac{1}{2}(\mathbb{I} + o_1W_1) \otimes \cdots \otimes \frac{1}{2}(\mathbb{I} + o_nW_n)\rho\right). \tag{A5}$$

This construction ensures that $\sigma_1(\rho)$ exactly reproduces the underlying quantum state ρ in expectation. That is, if we average over all 3^n choices of Pauli measurements and the associated (single-shot) outcomes $o_i \in \{\pm 1\}$, we obtain

$$\mathop{\mathbb{E}}_{s_1,\ldots,s_n}\left[\sigma_1(\rho)\right] = \mathop{\mathbb{E}}_{s_1,\ldots,s_n}\left[\frac{1}{2}\left(\mathbb{I} + 3o_1W_1\right) \otimes \cdots \otimes \frac{1}{2}\left(\mathbb{I} + 3o_n3W_n\right)\right] \tag{A6a}$$

$$= \sum_{W_1,\ldots,W_n=X,Y,Z} \sum_{o_1,\ldots,o_n=\pm 1} \frac{1}{3^n}\mathrm{tr}\left(\frac{1}{2}(\mathbb{I} + o_1W_1) \otimes \cdots \otimes \frac{1}{2}(\mathbb{I} + o_nW_n)\rho\right) \tag{A6b}$$

$$\times \frac{1}{2}\left(\mathbb{I} + o_13W_1\right) \otimes \cdots \otimes \frac{1}{2}\left(\mathbb{I} + o_n3W_n\right) \tag{A6c}$$

$$= \rho. \tag{A6d}$$

We refer to Ref. [88] for a more detailed derivation and context.

The classical shadow (A3) attempts to approximate this expectation value by an empirical average over T independent samples, much like Monte Carlo sampling approximates an integral. The accuracy of the approximation increases with T, but insisting on accurate approximations of the global state ρ is prohibitively expensive. Known fundamental lower bounds [66, 73] state that classical shadows of exponential size (at least) $T = \Omega\left(2^n/\epsilon^2\right)$ are required to ϵ-approximate ρ in trace distance. This quickly becomes intractable in terms of both measurement budget, as well as classical storage and processing.

This bleak picture lightens up considerably if we restrict our attention to subsystem approximations. The classical shadow size required to accurately approximate *all* reduced r-body density matrices scales exponentially in subsystem size r, but is independent of the total number of qubits n.

Lemma 1. *Fix $\epsilon, \delta \in (0,1)$, a subsystem size $r \leq n$ and let $\sigma_T(\rho)$ be a classical shadow (A3) of an n-qubit quantum state ρ with size*

$$T = (8/3)12^r \left(r\left(\log(n) + \log(12)\right) + \log(1/\delta)\right)/\epsilon^2 = \mathcal{O}\left(r12^r \log(n/\delta)/\epsilon^2\right). \tag{A7}$$

Then, with probability at least $1-\delta$,

$$\left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}\left(\rho\right)\right\|_1 \leq \epsilon \quad \textit{for all subsystems } A \subset \{1,\ldots,n\} \textit{ with size } |A| \leq r. \tag{A8}$$

Proof. Let us start by considering a fixed subsystem $A = \{i_1, \ldots, i_r\}$ comprised of (at most) r qubits. Use linearity to exchange partial trace with expectation value to obtain

$$\mathop{\mathbb{E}}_{s_{i_1}^{(t)},\ldots,s_{i_r}^{(t)}} \left[\left(3|s_{i_1}^{(t)}\rangle\langle s_{i_1}^{(t)}| - \mathbb{I}\right) \otimes \cdots \otimes \left(3|s_{i_r}^{(t)}\rangle\langle s_{i_r}^{(t)}| - \mathbb{I}\right)\right] \tag{A9a}$$

$$= \mathrm{tr}_{\neg A}\left(\mathop{\mathbb{E}}_{s_1^{(t)},\ldots,s_n^{(t)}} \left[\left(3|s_1^{(t)}\rangle\langle s_1^{(t)}| - \mathbb{I}\right) \otimes \cdots \otimes \left(3|s_n^{(t)}\rangle\langle s_n^{(t)}| - \mathbb{I}\right)\right]\right) \tag{A9b}$$

$$= \mathrm{tr}_{\neg A}(\rho), \tag{A9c}$$

according to Eq. (A6). In words, each reduced tensor product is an independent random matrix that reproduces the r-qubit state $\mathrm{tr}_{\neg A}(\rho)$ exactly in expectation. Empirical averages of T such independent and identically distributed (*iid*) random matrices tend to concentrate sharply around this expectation value. The matrix Bernstein inequality, see e.g. [167], provides powerful tail bounds in terms of operator norm deviation. Let $X_1, \ldots, X_T$ be *iid* random D-dimensional matrices that obey $\|X_t - \mathbb{E}\, X_t\|_\infty \leq R$ almost surely. Then, for $\tilde{\epsilon} > 0$

$$\Pr\left[\left\|\tfrac{1}{T}\sum_{t=1}^{T}\left(X_t - \mathbb{E}\, X_t\right)\right\|_\infty \geq \tilde{\epsilon}\right] \leq 2D\exp\left(-\frac{T\tilde{\epsilon}^2/2}{\sigma^2 + R\tilde{\epsilon}/3}\right) \quad \text{where} \quad \sigma^2 = \left\|\tfrac{1}{T}\sum_t \mathbb{E}\, X_t^2\right\|_\infty. \tag{A10}$$

Let us apply this tail bound to classical shadow concentration. We have $D \leq 2^r$ (at most r qubits) and set $X_t = \left(3|s_{i_1}^{(t)}\rangle\langle s_{i_1}^{(t)}| - \mathbb{I}\right) \otimes \cdots \otimes \left(3|s_{i_r}^{(t)}\rangle\langle s_{i_r}^{(t)}| - \mathbb{I}\right)$, such that $\mathbb{E}\, X_t = \mathrm{tr}_{\neg A}(\rho)$. Eq. (A4) then implies $\|X_t - \mathbb{E}\, X_t\|_\infty \leq \|X_t\| + \|\mathbb{E}\, X_t\|_\infty \leq 2^r + 1 =: R$. Accurately bounding σ^2 is somewhat more involved, and we turn to existing literature. A computation detailed in [72, Appendix C.3] yields $\sigma^2 = 3^r$. We are now ready to apply the matrix Bernstein inequality. For $\tilde{\epsilon} > 0$,

$$\Pr\left[\left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_\infty \geq \tilde{\epsilon}\right] \leq 2^{r+1}\exp\left(-\frac{T\tilde{\epsilon}^2/2}{3^r + (2^r+1)\tilde{\epsilon}/3}\right) \leq 2^{r+1}\exp\left(-\frac{3T\tilde{\epsilon}^2}{8\times 3^r}\right), \tag{A11}$$

for $\tilde{\epsilon} \in (0,1)$. This is a powerful concentration statement in operator norm. We can use the equivalence relation between trace- and operator norm, $\|X\|_\infty \leq \|X\|_1 \leq D\|X\|_\infty$, to obtain a tail bound for trace norm deviations:

$$\Pr\left[\left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_\infty \geq \epsilon\right] \leq \Pr\left[\left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_1 \geq \epsilon/2^r\right] \leq 2^{r+1}\exp\left(-\frac{3T\epsilon^2}{8\times 12^r}\right). \tag{A12}$$

We see that, for a fixed subsystem $A = \{i_1, \ldots, i_r\}$, the probability of an ϵ-deviation in trace distance is exponentially suppressed in the size T of the classical shadow. A union bound allows us to extend this assertion to *all* subsystems comprised of (at most) r qubits:

$$\Pr\left[\max_{A\subset\{1,\ldots,n\},|A|\leq r} \left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_1 \geq \epsilon\right] \leq \sum_{A\subset\{1,\ldots,n\},|A|\leq r} \Pr\left[\left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_1 \geq \epsilon\right] \tag{A13a}$$

$$\leq n^r 2^{r+1}\exp\left(-\frac{3T\epsilon^2}{8\times 12^r}\right). \tag{A13b}$$

Setting $T = (8/3)12^r\left(\log(n^r 12^r) + \log(1/\delta)\right)/\epsilon^2 = (8/3)12^r\left(r\left(\log(n) + \log(12)\right) + \log(1/\delta)\right)/\epsilon^2$ ensures that this upper bound on failure probability does not exceed δ. □

APPENDIX B: Related work

B.1. Estimating ground state properties

Determining the ground state of a system governed by a known many-body Hamiltonian is a long-standing problem in quantum science. Despite having several well-established and practically successful algorithms at our disposal, we are typically faced with either a runtime that scales exponentially with system size, and/or a lack of rigorous performance guarantees. The literature is vast, and surveying it is beyond the scope of this article. Instead, we review a few families of established algorithms in order to put our work into proper context.

Density functional theory (DFT) has been a workhorse for determining properties of interacting electronic systems in quantum chemistry and solid-state physics. DFT recasts the problem of finding the many-body state with minimal energy into finding a few-body energy functional. While the "true" functional corresponding to the ground state is known to exist in theory [84, 105], determining it to polynomial accuracy in the number of electrons is QMA-hard [153]. Various efficient approximations to the true functional have seen much practical success, but they are difficult to justify rigorously (except so far for some special cases [111]). These limitations present an opportunity for ML approaches to be used instead of or to supplement DFT methods [148].

The family of algorithms known as Quantum Monte Carlo (QMC) [34, 144] utilizes probabilistic sampling techniques to estimate observable properties at either finite or zero temperature. For ground states, expectation values can be obtained using an imaginary-time evolution projector or a high-power of the model Hamiltonian [98]. The efficiency of QMC methods depends on the structure of the Hamiltonian, specifically on whether all of its off-diagonal matrix elements are negative (i.e. it is *stoquastic* [24]). In this case, the ground state wavefunction is real-valued and positive, and the algorithmic complexity of the QMC estimators scales polynomially with the number of particles. For non-stoquastic Hamiltonians, the QMC suffers from the so-called *sign problem*, which makes evaluation of statistical properties of the system NP-hard and renders QMC intractable for large systems or low temperatures [168]. It is important to note that, even for stoquastic Hamiltonians with polynomial computational complexity, the success of QMC simulations heavily relies on the existence of efficient sampling schemes (e.g. cluster updates) which are sufficiently ergodic, and leading to small auto-correlation time [59]. In general, it is not possible to prove the existence of such update algorithms, nor their ergodicity.

An alternative approach to solve for the ground state properties of a many-body Hamiltonian is based on the variational principle in quantum mechanics, which states that the expectation value of the energy on any valid wavefunction is always greater or equal than the ground state energy. It is then possible to design classical parametric representations of the many-body wavefunction, and update their parameters to minimize the corresponding energy estimator. A notable example is the density-matrix renormalization group [181, 182] (DMRG). This algorithm can be interpreted as a variational optimization of a Matrix Product State (MPS) [133, 152, 175], which is a local decomposition of a wavefunction as a one-dimensional tensor network. These parametrized wavefunctions display area law of entanglement and exponentially-decaying correlations [127], which lend themselves most effective for systems described by one-dimensional gapped Hamiltonians. Furthermore, a standout feature of DMRG is that modifications of the original procedure, such as rigorous renormalization group algorithms [9], are guaranteed to find the ground state of one-dimensional geometrically-local gapped Hamiltonians in polynomial time [2, 9, 108]. In two spatial dimensions, MPS-based DMRG can still be applied to solve for ground states [156, 160, 184], though it suffers an exponential scaling in one of the two linear dimensions of the system. Projected entangled pair states (PEPS) [45, 95, 172], the two-dimensional generalization of MPSs, are instead a more suitable ansatz in this context. However, while improved algorithms for PEPS optimization are routinely put forward [76, 91, 186], the same level of performance achieved by DMRG in 1d systems is still out of reach.

Another class of variational wavefunctions that has recently received a lot of attention are *neural-network quantum states* [30]. In this framework, a neural network is used as a parametric function approximator of a many-body wavefunction $\psi_\lambda(\sigma) = \langle\sigma|\psi\rangle$, where the classical state σ is interpreted as the neural-network input, and λ is a set of neural-network parameters (i.e. weights and biases). In a variational setting, these parameters are iteratively optimized to lower the total energy [18], or additionally the energy variance. Neural-network quantum states have been explored in a variety of setups, including topological phases [50], Fermi-Hubbard models [124], molecular ground states [42], frustrated magnetism [62], and more [41, 69, 116, 121, 176]. The auto-regressive property of some types of neural networks (e.g. recurrent neural networks, transformers, etc.) has also been leveraged to improve convergence of variational Monte Carlo [83]. In contrast to tensor-network states, this class of wavefunctions can more easily display non-local correlations, allowing in principle to capture quantum states with higher entanglement [51].

Another class of machine learning methods [68, 138, 154] train neural networks to predict the ground state properties directly. The input to the neural network is a description of the Hamiltonian, and the output is a ground state property of interest. The training data is a set of different Hamiltonians (inputs) and their corresponding ground state properties (outputs). This class of ML methods is closest to the setting considered

in this paper. Methods in this class lack rigorous guarantees, so it is not clear when such approaches could outperform non-ML algorithms. In one of our main contributions, given in Theorem 1 and Proposition 1, we introduce an ML model that, when trained with experimental data, can accurately predict ground state representations better than any classical algorithm that does not learn from data. Our model is relatively basic, utilizing the well-known l_2-Dirichlet kernel, but it is already enough to establish a rigorous guarantee. Similarly determining when other ML models yield an advantage over non-ML algorithms is an interesting topic for future work.

B.2. Classifying quantum phases of matter

Proposals for classifying quantum phases of matter abound. These include quantum neural networks [44], classical neural networks [17, 32, 70, 147, 171], or other classical machine learning models [143, 178, 180]. Since these models do not come with rigorous guarantees, relying on them too much can lead one astray. For example, some deep neural networks can misclassify the original phase if the corresponding state is distorted by noise, even if the distortion is very slight [93].

In this work, we provide rigorous machine learning approaches that are guaranteed to classify accurately under the specified conditions given in Theorem 2. We believe similar analyses can be performed on other machine learning models to understand their limitation and potential, which will be an important future direction. The ML model used in [143], which is based on defining diffusion maps over classical spins systems, is the ML approach that seems most similar to that used in our work. Hence, it is very likely that the models considered in [143] can be rigorously analyzed via similar techniques. Neural network approaches for classifying phases of matter will be harder to analyze, but one should be able to study neural network with large hidden layers using the theory developed in this work and the theory of neural tangent kernels [92, 125].

B.3. Classical representations of quantum systems

One of the most important ingredients in designing classical ML procedures for understanding quantum spin systems is the construction of efficient classical representations of the underlying quantum systems. The properties of the quantum system retained by the classical representation directly determine the set of functions the classical ML procedure can learn. The classical shadow formalism [88, 129], developed by some of us and others and used throughout this work, is a versatile framework for this purpose. It has been extended to fermionic systems [74, 189], suggesting that our ML approaches may be extendable as well. Classical shadows have also been shown to allow sample-efficient reconstruction of Hamiltonian from thermal states [8], although such an algorithm is not yet time efficient. However, classical shadows provide only one of many promising and actively studied approaches for efficient representation [36, 48, 58, 88, 89, 104, 129].

While the curse of dimensionality prevents one from representing general quantum spin systems both exactly and efficiently, simplifying assumptions can lift the curse and drastically reduce both the overhead and complexity of representing and characterizing the system. A prime example is a classical system, whose Hamiltonian is diagonal in the computational basis. ML methods for such systems, such as those in Ref. [143] (discussed above), do not require an additional quantum-to-classical compression. Another example, relevant to electronic material characterization, is the family of solid-state band insulators [19] — gapped two-dimensional non-interacing fermionic systems with various crystalline symmetries. Their myriad topological phases can typically be characterized by data at a discrete set of high-symmetry points in the Brillouin zone [15, 23, 28, 135]. The techniques developed here should pave the way for certifying accuracy of current ML methods for band insulator characterization (e.g., [7, 16, 43, 119, 130, 146, 159]) as well as developing new ones.

APPENDIX C: Neural networks with classical shadow for quantum many-body problems

Imposing inductive biases in the ML model is a common technique for boosting the prediction performance of ML models. One approach is to enhance the proposed ML algorithms with neural networks, such as convolutional or graph neural networks. These neural networks could better capture structure of the underlying function we are trying to learn and hence may require significantly less data than the very expressive ML model given in the main text. We leave the proof that neural network enhancements can lead to better prediction performance as a goal for future work.

There are multiple ways of combining classical shadows and neural networks. Here, we will only showcase one such approach by utilizing the theory of neural tangent kernels [92]. Remarkably, this theory allows us to efficiently train various types of neural networks (convolutional/graph/etc.) with an infinite number of neurons

in each hidden layer (*infinite width*). As such, this line of work has gained a lot of attention [10, 52, 125] in recent years. In the limit of infinite width, one can analytically solve for the neural network after training on a set of data $\{x_\ell, y_\ell\}_{\ell=1}^N$, where x_ℓ and y_ℓ are vectors of some size. For example, consider training a neural network that takes in a vector x and produces a vector $f_\theta^{\rm NN}(x)$ through the following optimization problem using gradient descent,

$$\min_\theta \sum_{\ell=1}^N \left\| f_\theta^{\rm NN}(x_\ell) - y_\ell \right\|_2^2, \tag{C1}$$

where we begin on a randomly initialized θ. Note that due to the infinite number of neurons, θ is a vector of infinite dimension. The trained neural network $f_{\theta^*}^{\rm NN}(x)$ can always be written in the following form

$$f_{\theta^*}^{\rm NN}(x) = \sum_{\ell=1}^N \sum_{\ell'=1}^N k^{(\rm NTK)}(x, x_\ell)(K^{-1})_{\ell\ell'} y_{\ell'}, \tag{C2}$$

where $k^{(\rm NTK)}(x, x')$ is a function called the neural tangent kernel [92], and $K_{\ell,\ell'} = k^{(\rm NTK)}(x_\ell, x_{\ell'})$ is the kernel matrix of the neural tangent kernel. One can see that the infinite-dimensional vector θ^* does not appear on the right hand side of Eq. (C2). And as long as we can efficiently evaluate the neural tangent kernel $k^{(\rm NTK)}(x, x')$, we can evaluate the infinite-dimensional neural network in polynomial time. This is the main contribution of [92], which enables one to efficiently train infinite-width neural networks. For a given neural network architecture, one can compute $k^{(\rm NTK)}(x, x')$ efficiently using open-source software, such as [125]. In Appendix D.2, we give the code for training infinite-width neural networks using the open-source software: Neural Tangents [125].

C.1. Predicting ground state representation

For the task of predicting ground state representation, we consider the training data to be

$$\left\{ x_\ell \to \sigma_T(\rho(x_\ell)) \right\}_{\ell=1}^N, \tag{C3}$$

where $\sigma_T(\rho(x_\ell))$ is the classical shadow representation of $\rho(x_\ell)$ given in Eqs. (1) and (A3) based on T randomized Pauli measurements. Recall that $\sigma_T(\rho(x_\ell))$ is a $2^n \times 2^n$ matrix that reproduces $\rho(x_\ell)$ in expectation over the randomized Pauli measurements. Suppose we now train an infinite-width neural network parameterized by θ that takes in an input x and produces an exponential-size matrix $\sigma_\theta^{\rm NN}(x)$, by solving the optimization problem

$$\min_\theta \quad \sum_{\ell=1}^N \left\| \sigma_\theta^{\rm NN}(x_\ell) - \sigma_T(\rho(x_\ell)) \right\|_F^2. \tag{C4}$$

The squared Frobenius difference between two matrices is equal to the squared Euclidean norm of their vectorizations (flattenings). In turn, the theory of infinite-width neural networks [92] shows that the trained neural network $\sigma_{\theta^*}^{\rm NN}(x)$ could be written in the form

$$\sigma_{\theta^*}^{\rm NN}(x) = \sum_{\ell=1}^N \sum_{\ell'=1}^N k^{(\rm NTK)}(x, x_\ell)(K^{-1})_{\ell\ell'} \sigma_T(\rho(x_{\ell'})). \tag{C5}$$

The kernel function $k^{(\rm NTK)}(x, x')$ depends on the neural network architecture and could be calculated utilizing existing open-source software [125]. This also falls into the general form shown in the main text; see Eq. (2). Hence, training an infinite-width neural networks to predict an exponentially large density matrix can be done efficiently on a classical computer. For a given neural network architecture, all one has to do is compute the kernel function $k^{(\rm NTK)}(x, x')$. Then the neural network optimized using the training data could be analytically solved as given in Eq. (C5). To estimate a property on the predicted ground state using the neural network is as simple as evaluating

$$\mathrm{tr}(O\sigma_{\theta^*}^{\rm NN}(x)) = \sum_{\ell=1}^N \sum_{\ell'=1}^N k^{(\rm NTK)}(x, x_\ell)(K^{-1})_{\ell\ell'} \,\mathrm{tr}(O\sigma_T(\rho(x_{\ell'}))), \tag{C6}$$

which can be done by first computing $\mathrm{tr}(O\sigma_T(\rho(x_\ell))), \forall \ell = 1, \ldots, N$ and compute the linear interpolation.

C.2. Classifying phases of matter

We want to learn how to classify two phases of n-qubit states. A fully classical training set would simply consist of N labeled classical representations of quantum states $\{\rho_\ell \to y_\ell\}_{\ell=1}^N$, where $y_\ell = +1$ (-1) if ρ_ℓ belongs to phase A (B). However, insisting on perfect knowledge of each ρ_ℓ is impractical for a variety of reasons. Instead, we assume that we have access to classical shadows of ρ_ℓ. The raw data $S_T(\rho_\ell)$ behind each classical shadow is a 2-dimensional array,

$$S_T(\rho_\ell) = \left\{ |s_i^{(t)}\rangle : \ i \in \{1,\ldots,n\}, t \in \{1,\ldots,T\} \right\} \quad \text{where} \quad |s_i^{(t)}\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\}. \tag{C7}$$

In the main text, we propose to use this data to train a support vector machine based on the shadow kernel

$$k^{(\mathrm{shadow})}\left(S_T(\rho_\ell), \tilde{S}_T(\rho_{\ell'})\right) = \exp\left(\frac{\tau}{T^2}\sum_{t,t'=1}^{T} \exp\left(\frac{\gamma}{n}\sum_{i=1}^{n} \mathrm{tr}\left(\left(3|s_i^{(t)}\rangle\langle s_i^{(t)}| - \mathbb{I}\right)\left(3|\tilde{s}_i^{(t)}\rangle\langle \tilde{s}_i^{(t)}| - \mathbb{I}\right)\right)\right)\right). \tag{C8}$$

This specific choice of (deterministic) kernel function allows us to carry out a thorough theoretical analysis of the entire learning procedure; see Appendix J.

But there are other sensible kernels that may perform even better in practice. For instance, we could feed the two-dimensional data array (C7) into a neural network architecture, e.g. a convolutional neural network. In the limit of an infinite number of neurons in each hidden layer, this produces the neural tangent kernel $k^{(\mathrm{NTK})}\left(S_T(\rho_\ell), \tilde{S}_T(\rho_{\ell'})\right)$ [92]. This kernel is positive-semidefinite and should be viewed as a measure of similarity induced by the trained neural network. Mercer's theorem [118] allows us to make this intuition precise by reformulating the neural tangent kernel as a Gram matrix in feature space:

$$k^{(\mathrm{NTK})}\left(S_T(\rho_\ell), \tilde{S}_T(\rho_{\ell'})\right) = \left\langle \phi^{(\mathrm{NTK})}\left(S_T(\rho_\ell)\right), \phi^{(\mathrm{NTK})}\left(\tilde{S}_T(\rho_{\ell'})\right)\right\rangle . \tag{C9}$$

Hence, any infinite-width neural network with input array $S_T(\rho)$ induces a feature map $\phi^{(\mathrm{NTK})}$ that can be used instead of the doubly-infinite feature map $\phi^{(\mathrm{shadow})}$ (5) that is associated with the shadow kernel (C8).

APPENDIX D: Details regarding numerical experiments

In this appendix, we provide additional numerical experiments as well as more details about the numerical experiments described in the main text.

D.1. Additional numerical experiments

Rydberg atom chain — In the main text, we have provided partial prediction outcomes for a one-dimensional chain of $n = 51$ Rydberg atoms; see Figure 2. Here, we supply predictions of expectation values of Pauli operators Z_i and X_i on all 51 atoms at the testing points marked in Figure 2(b). These are shown in Figure 6 and Figure 7, respectively. These extend the more restricted presentation in the main text to all qubits.

Distinguishing an SPT phase from a trivial phase — We consider a one-dimensional chain of $n = 50$ qubits with $Z_2 \times Z_2$ symmetry. The 1D cluster state is in the nontrivial SPT phase. We generate other representatives of the nontrivial SPT phase by applying symmetric depth-3 geometrically local random quantum circuits to the cluster state, and we generate representatives of the trivial phase by applying symmetric depth-3 random circuits to a product state.

Randomized Pauli measurements are performed $T = 500$ times to convert the states to their classical shadows, and these classical shadows are mapped to feature vectors in the high-dimensional feature space using the feature map $\phi^{(\mathrm{shadow})}$ (5). In Figure 8(a), inner products of feature vectors (matrix elements of the shadow kernel) are displayed. Figure 8(b) shows the feature vectors projected onto a two-dimensional subspace using principal component analysis (PCA). Both figures show that feature vectors representing distinct phases can be distinguished easily. Correspondingly, the classical ML efficiently learns how to classify phases accurately, even if the training data is unlabeled.

Distinguishing a topologically-ordered phase from a trivial phase — We consider the task of distinguishing the toric code [100] topologically-ordered phase from the trivial phase in a system of $n = 200$ qubits. We generate other representatives of the topologically-ordered phase by applying two-dimensional depth-3 geometrically local random quantum circuits to the toric code state, and we generate representatives of the trivial phase by applying two-dimensional depth-3 random circuits to a product state.

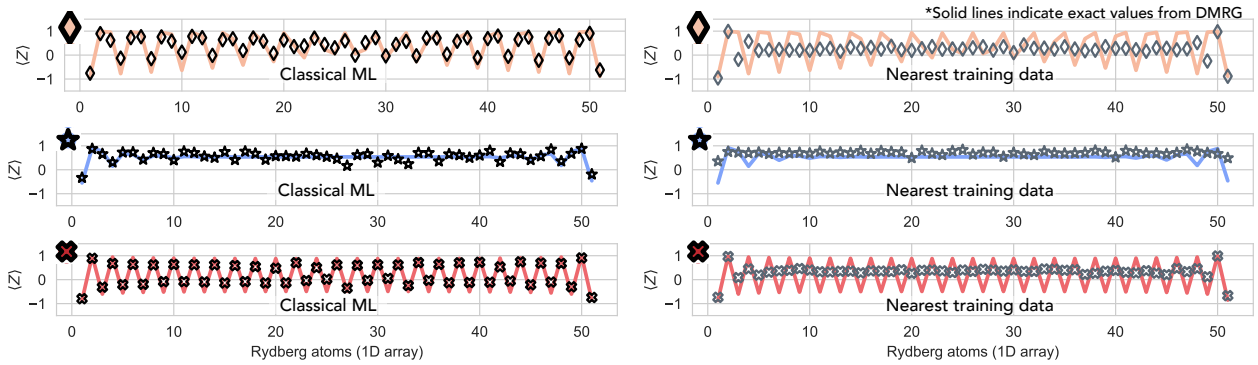

Figure 6: Numerical experiment for predicting ground state properties (Pauli-Z in each atom) in a 1D Rydberg atom system with 51 atoms. We use the classical ML to predict the ground state properties at the three testing points. Also shown are "predictions" obtained from the training data nearest to the testing points. The markers denote predicted values, while the solid lines denote exact values obtained from DMRG.

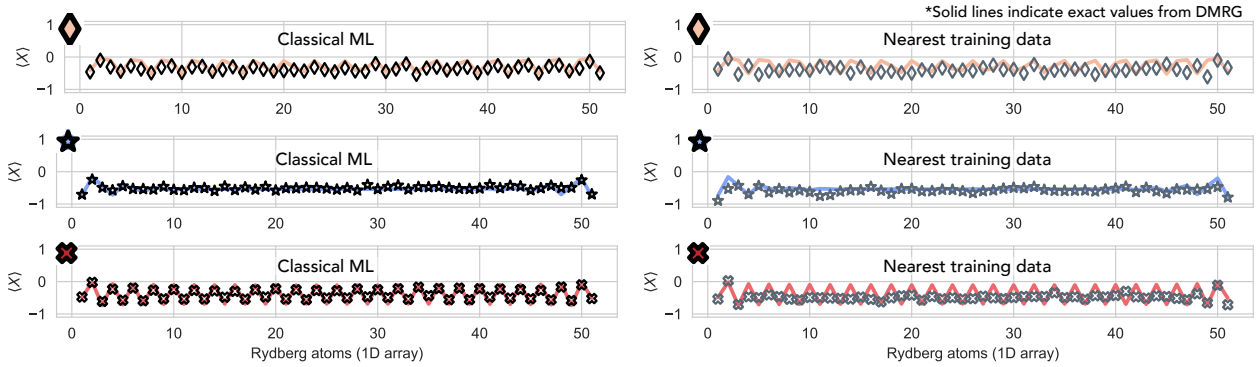

Figure 7: Numerical experiment for predicting ground state properties (Pauli-X in each atom) in a 1D Rydberg atom system with 51 atoms. We use the classical ML to predict the ground state properties at the three testing points. Also shown are "predictions" obtained from the training data nearest to the testing points. The markers denote predicted values, while the solid lines denote exact values obtained from DMRG.

Randomized Pauli measurements are performed $T = 500$ times to convert the states to their classical shadows, and these classical shadows are mapped to feature vectors in the high-dimensional feature space using the feature map $\phi^{(\text{shadow})}$. In Figure 8(c, d), inner products of feature vectors (matrix elements of the shadow kernel) and the projection of feature space data onto the two-dimensional subspace spanned by the largest principal components is shown. Once more, one can clearly see that feature vectors representing distinct phases can be distinguished easily. Correspondingly, the classical ML efficiently learns how to classify phases accurately, even if the training data is unlabeled.

D.2. Ground state properties of the Rydberg atom chain

Our first example is a one-dimensional chain of $n = 51$ Rydberg atoms [20, 26, 61]. Each atom can be in either its ground state or a highly excited Rydberg state. Such systems can effectively be regarded as a qubit, where the basis state $|0\rangle$ is the ground state $|g\rangle$ and the basis state $|1\rangle$ is the Rydberg state $|r\rangle$. The Hamiltonian of the atomic chain is

$$H = \frac{\Omega}{2}\sum_i X_i - \Delta\sum_i N_i + \Omega\sum_{i<j}\left(\frac{R_b}{a|i-j|}\right)^6 N_i N_j \ , \tag{D1}$$

where Ω is the (fixed) Rabi frequency, Δ is the laser detuning, N_i is the Rydberg occupation number operator, a is the separations of the atoms, and R_b is the so called Rydberg blockade radius. For large and negative Δ, the ground state of H is a vacuum state, where all atoms are in the ground state $|g\rangle$. In contrast, for large and positive Δ, different broken-symmetry ground states can be engineered depending on the value of R_b.

Approximations of the exact ground states of the Rydberg chain were found using the density-matrix renormalization group (DMRG) based on matrix product states (MPS). Starting from a random MPS with bond

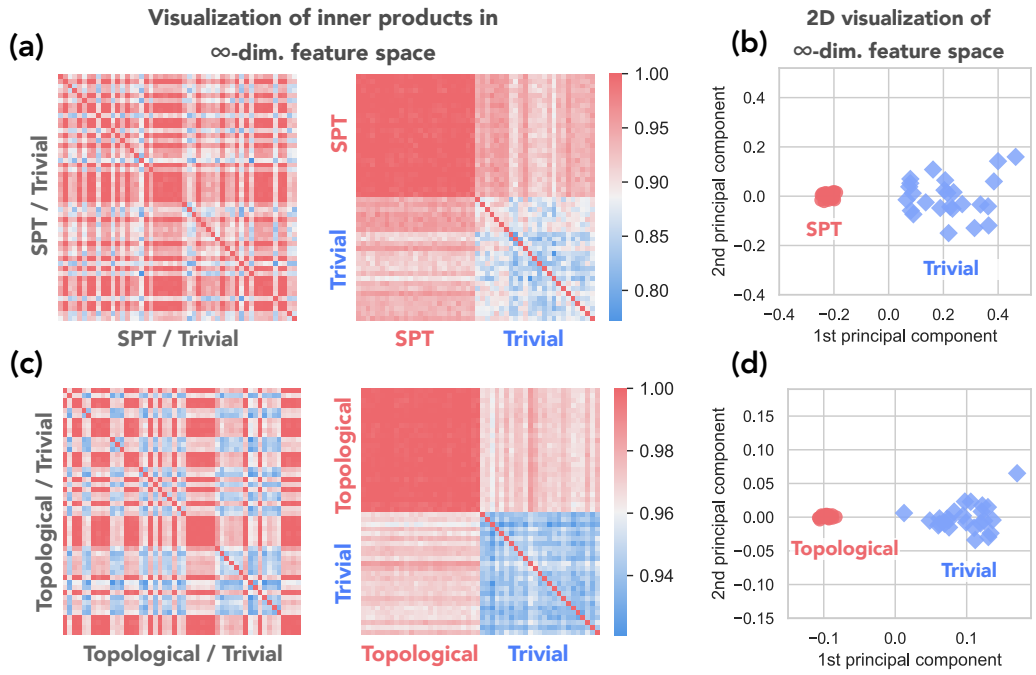

Figure 8: Numerical experiments for distinguishing trivial and topological phases. Trivial or topological states are generated by applying low-depth local random quantum circuits to a product state or exactly solved topological state respectively. (a) KERNEL MATRIX FOR SPT/TRIVIAL PHASES The exactly solved topological state is the cluster state. The (i, j)-entry denotes the inner product of the i-th and j-th feature vectors in the infinite-dimensional feature space defined by the classical shadow representation. To the left, states from the two phases are randomly mixed. To the right, the two phases are ordered. (b) KERNEL MATRIX FOR TOPOLOGICALLY-ORDERED/TRIVIAL PHASES. The exactly solved topological state is the toric code ground state.

dimension $\chi = 10$, we variationally optimize the MPS using a singular value decomposition (SVD) cutoff of 10^{-9}. We perform a number of DMRG sweeps until the change in energy is below $\epsilon = 10^{-6}$. Upon convergence, we perform randomized Pauli measurements simply by performing local rotations into the corresponding Pauli bases, and sampling the resulting state [63].

In Figure 2(b), the color in the phase diagram corresponds to the phase obtained by two order parameters for characterizing Z_2 and Z_3 order. For Z_2 order, where the atoms are in $|rgrgrg\ldots\rangle$ or $|grgrgr\ldots\rangle$, we consider the order parameter,

$$O_{Z_2} = \frac{1}{n-1}\sum_{i=1}^{n-1}\left(|r_i g_{i+1}\rangle\langle r_i g_{i+1}| + |g_i r_{i+1}\rangle\langle g_i r_{i+1}|\right). \tag{D2}$$

For Z_3 order, where the atoms are in $|rggrgg\ldots\rangle$ or $|grggrg\ldots\rangle$ or $|ggrggr\ldots\rangle$, we consider the order parameter,

$$O_{Z_3} = \frac{1}{n-2}\sum_{i=1}^{n-2}\left(|r_i g_{i+1} g_{i+2}\rangle\langle r_i g_{i+1} g_{i+2}| + |g_i r_{i+1} g_{i+2}\rangle\langle g_i r_{i+1} g_{i+2}| + |g_i g_{i+1} r_{i+2}\rangle\langle g_i g_{i+1} r_{i+2}|\right). \tag{D3}$$

We estimate the two order parameters of the ground state ρ. First we check which order parameter (O_{Z_2} or O_{Z_3}) yields a larger expectation value. Then, we check if that expectation value is larger than the threshold value 0.8. If $O_{Z_2} > O_{Z_3}$ and $O_{Z_2} > 0.8$, we associate the state with the Z_2-order phase (red color). Else if $O_{Z_3} > O_{Z_2}$ and $O_{Z_3} > 0.8$, we say that the state is in the Z_3-order phase (vanilla color). If neither of these conditions is satisfied (both expectation values are less than 0.8), we assign the disordered phase (blue color) to this state.

For the Rydberg atom experiment, the input parameter vector x is two-dimensional. We first normalize the values to lie within a square $[-1, 1]^2$. Then we consider classical machine learning models given by

$$\hat{\sigma}_N(x) = \sum_{\ell=1}^{N}\kappa(x, x_\ell)\sigma_T(x_\ell) = \sum_{\ell=1}^{N}\underbrace{\left(\sum_{\ell'=1}^{N}k(x, x_{\ell'})(K+\lambda I)^{-1}_{\ell'\ell}\right)}_{\kappa(x,x_\ell)}\sigma_T(x_\ell), \tag{D4}$$

where $\lambda > 0$ is a parameter to regularize the model when K is not invertible, $\sigma_T(x_\ell)$ is shorthand for $\sigma_T(\rho_\ell)$ and denotes the classical shadow representation of the ground state $\rho_\ell = \rho(x_\ell)$ under T randomized Pauli measurements. Moreover, $K_{ij} = k(x_i, x_j)$ is the kernel matrix, $k(x, x')$ is a kernel function, and $\kappa(x, x_\ell)$ is a function that depends on the kernel function, the kernel matrix K, and λ. We consider a set of different regularization parameters,

$$\lambda \in \{0.0125, 0.025, 0.05, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0\}, \tag{D5}$$

and we also consider a set of different kernel functions $k(x, x') = \tilde{k}(x, x')/\sqrt{\tilde{k}(x, x)\tilde{k}(x', x')}$, where

$$\tilde{k}(x, x') = \exp(-\gamma \|x - x'\|_2^2), \qquad \text{(Gaussian kernel)}, \tag{D6a}$$

$$\tilde{k}(x, x') = \sum_{k_1=-3}^{3} \sum_{k_2=-3}^{3} \cos\left(\pi(k_1(x_1 - x_1') + k_2(x_2 - x_2'))\right), \qquad \text{(Dirichlet kernel)}, \tag{D6b}$$

$$\tilde{k}(x, x') = k^{(\mathrm{NTK})}(x, x'), \qquad \text{(Neural tangent kernel)}. \tag{D6c}$$

The hyperparameter $\gamma > 0$ in the Gaussian kernel is chosen to be equal to $N^2/\sum_{i=1}^N \sum_{j=1}^N \|x_i - x_j\|_2^2$, the inverse of the average distance between x_i and x_j. We consider the neural tangent kernel $k^{(\mathrm{NTK})}(x, x')$ [92, 125] that is equivalent to an infinite-width feed-forward neural network with $2, 3, 4, 5$ hidden layers and that uses the rectified linear unit (ReLU) as the activation function. Computing the neural tangent kernel can be implemented easily using the open-source software Neural Tangents [125]. Suppose that the input data $\{x_\ell\}_{\ell=1}^N$ is stored in a `numpy` array of size $N \times m$, denoted as `dataX` in the following code. We can use then use following code to generate the neural tangent kernel matrix. The imported package `neural_tangents` can be downloaded from https://github.com/google/neural-tangents.

```
import jax
import numpy as np
from neural_tangents import stax

init_fn, apply_fn, kernel_fn = stax.serial(
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(1)
)
kernel_NN2 = kernel_fn(dataX, dataX, 'ntk')

init_fn, apply_fn, kernel_fn = stax.serial(
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(1)
)
kernel_NN3 = kernel_fn(dataX, dataX, 'ntk')

init_fn, apply_fn, kernel_fn = stax.serial(
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(1)
)
kernel_NN4 = kernel_fn(dataX, dataX, 'ntk')

init_fn, apply_fn, kernel_fn = stax.serial(
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(32), stax.Relu(),
    stax.Dense(1)
```

```
)
kernel_NN5 = kernel_fn(dataX, dataX, 'ntk')

list_kernel_NN = [kernel_NN2, kernel_NN3, kernel_NN4, kernel_NN5]

# Normalization of the kernel matrix
for r in range(len(list_kernel_NN)):
    for i in range(len(list_kernel_NN[r])):
        for j in range(len(list_kernel_NN[r])):
            list_kernel_NN[r][i][j] /= (list_kernel_NN[r][i][i] \
                                        * list_kernel_NN[r][j][j]) ** 0.5
```

In order to predict the expectation value $\mathrm{tr}(O\hat{\sigma}_N(x))$ of an observable O for a new ground state $\hat{\sigma}_N(x)$, we utilize the following property of expectation values,

$$\mathrm{tr}(O\hat{\sigma}_N(x)) = \sum_{\ell=1}^{N} \kappa(x, x_\ell)\, \mathrm{tr}(O\sigma_T(x_\ell)). \tag{D7}$$

Hence, we first compute $\mathrm{tr}(O\sigma_T(x_\ell))$, which can be done efficiently for r-body observables that factorize nicely into tensor products. Indeed, an $O = O_{i_1} \otimes \ldots \otimes O_{i_r}$ ensures

$$\mathrm{tr}(O\sigma_T(x_\ell)) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{tr}\left(O\sigma_1^{(t)}(x_\ell) \otimes \cdots \otimes \sigma_n^{(t)}(x_\ell)\right) = \frac{1}{T}\sum_{t=1}^{T} \mathrm{tr}\left(O_{i_1}\sigma_{i_1}^{(t)}(x_\ell)\right) \ldots \mathrm{tr}\left(O_{i_r}\sigma_{i_r}^{(t)}(x_\ell)\right), \tag{D8}$$

and the right hand side can be computed with $\mathcal{O}(Tn)$ arithmetic operations. Then, we can compute $\mathrm{tr}(O\hat{\sigma}_N(x))$ by extrapolating $\mathrm{tr}(O\sigma_T(x_\ell))$ using $\kappa(x, x_\ell)$. We utilize scikit-learn, a Python package [132], for the training of these machine learning models.

Due to the different classical ML models one could consider (corresponding to different regularization parameters λ and kernel functions $k(x, x')$), we have to perform model selection to find an appropriate ML model. Typically, the prediction performance will be quite sensitive to these parameters, so one has to select them carefully. To evaluate the ML models, we consider 100 different points $x \in [-1, 1]^2$ in parameter space. Among these 100 points, we select $N = 20$ to be training data. These are the circled points in Figure 2(b). For each property we would like to predict, we choose one of the the three kernels and the different values of λ such that the prediction error is minimized on a validation set containing $80 - 3$ inputs of x. The validation set is disjoint from the 20 training points and the 3 testing points for evaluating the prediction performances (special markers in Figure 2(a)). Their purpose is to perform model selection. Finally, we test on the three input x's shown by the special markers (cross, diamond and star) in Figure 2(b).

We found that for each property we would like to predict, the prediction performance for different classical ML model varies moderately. When we have sufficiently large training data N sizes, most choices of λ and the kernel function should yield good prediction performance. However, we are using a very small number of training data in our experiments, hence the choice of these options becomes more important. In particular, the best choice of λ can differ quite significantly over the different properties we would like to predict.

For completeness, we include a set of experiments where we vary the training data size N or the classical shadow size T, that is the number of randomized Pauli measurements used to approximate each state. The result in given in Figure 9. For this set of experiments, we consider fixed sets of 70 validation points and 10 testing points in the phase space. We consider the root-mean-square error over the points in the phase space where we predict the ground state representation. We use the predicted ground state representation to estimate all single-atom Pauli-Z expectation values (a total of 51 atoms). We can see that as training set size N increases, the prediction becomes better. On the other hand, increasing shadow size T may not improve the prediction performance beyond certain error (0.21 in this experiment). The intuition is that when the shadow size is large enough, we have very accurate values in the training data. The prediction error then comes from the generalization from the training set to the testing set. While we proved a rigorous result using the basic Dirichlet kernel, we expect other more commonly used ML models to yield better prediction performance in practice. Proving rigorous prediction guarantees and understanding the limitations and strengths for other more commonly used ML models are important directions for future research.

D.3. Ground state properties of the 2D antiferromagnetic Heisenberg model

Our next example is the two-dimensional antiferromagnetic Heisenberg model. Spin-$\frac{1}{2}$ particles (i.e. qubits) occupy sites on a square lattice, and for each pair (ij) of neighboring sites the Hamiltonian contains a term

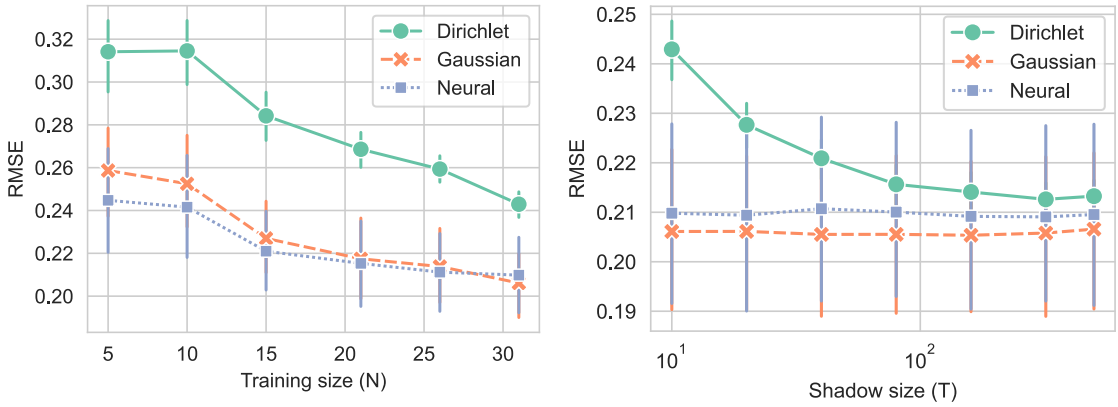

Figure 9: Numerical experiment for predicting ground state properties (Pauli-Z in each atom) in a 1D Rydberg atom system with 51 atoms under different hyperparameters. (Left) The prediction error (root-mean-square error) over different training sizes N with a fixed number $T = 10$ of randomized Pauli measurements , also referred to as the shadow size. (Right) The prediction error over different shadow sizes T with a fixed training data size $N = 31$. In both plots, the error bars show the standard deviation of RMSE over the 51 predictions on each atom.

$J_{ij}\,(X_iX_j + Y_iY_j + Z_iZ_j)$ where the couplings $\{J_{ij}\}$ are uniformly sampled from the interval $[0, 2]$. The parameter x is a list of all J_{ij} couplings; hence in this case the dimension of the parameter space is $m = O(n)$, where n is the number of qubits. The Hamiltonian $H(x)$ on a 5×5 lattice is shown in Figure 3(a). The exact ground state was found using DMRG. Analogously to the Rydberg atoms experiments, we fixed the SVD cutoff to 10^{-8} and stop the DMRG runs when the difference in energy was below 10^{-4}.

The classical ML models we considered are the same as the Rydberg atom chain experiment. The only difference is that we slightly modify the Dirichlet kernel (D6b) to

$$k(x, x') = \sum_{i\neq j} \sum_{k_i=-3}^{3} \sum_{k_j=-3}^{3} \cos\left(\pi(k_i(x_i - x_i') + k_j(x_j - x_j'))\right), \quad \text{(Dirichlet kernel).} \tag{D9}$$

We trained the classical ML model using a training set containing $N = 90$ randomly chosen values of the parameter $x = \{J_{ij}\}$. Then, for each property we would like to predict, we find the top-performing ML model setting (out of all λ parameters and kernel functions $k(x, x')$) on a validation set containing 100 parameters x distinct from the training set. Finally, we test on 10 newly sampled parameters x to estimate the prediction error. Figure 3(b) shows the prediction outcome from one of the input parameter x. Figure 3(c) shows the RMSE from all 10 input parameters.

Similar to the Rydberg atom experiment, the best-performing ML model setting differs across the properties we would like to predict. The three kernels perform similarly at larger training data size N and larger number of randomized Pauli measurements T. But neural networks and Gaussian kernel methods tend to perform better in most cases. The best choice of λ differs substantially across the different properties: there is not single choice of λ that performs uniformly better than the other choices.

To showcase these effects, we also include a set of experiments where we vary the training data size N or the classical shadow size T, that is the number of randomized Pauli measurements used to approximate each state. The numerical results are summarized in Figure 10. For this set of experiments, we consider fixed sets of 100 validation points and 10 testing points in the $m = \mathcal{O}(n)$ dimensional parameter space. We consider the root-mean-square error over the points in the parameter space where we predict the ground state representation. We use the predicted ground state representation to estimate two-point correlation functions (a total of 100 pairs of spins). The observations are similar to the Rydberg atom experiments. As training set size N increases, the prediction becomes better. And increasing shadow size T does not seem to improve the prediction performance beyond certain error (0.07 in this experiment).

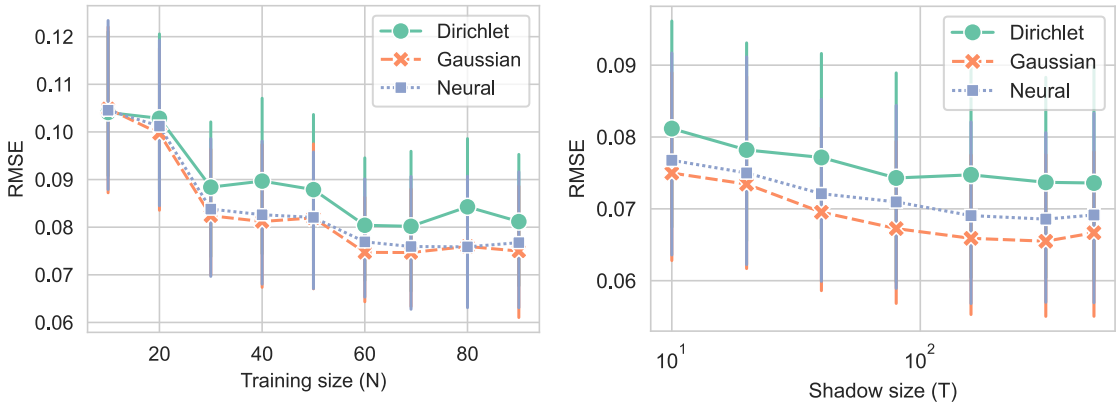

Figure 10: Numerical experiment for predicting ground state properties (two-point correlation functions) in a 2D antiferromagnetic Heisenberg model with 5×5 spins under different hyperparameters. (Left) The predict error (root-mean-square error) over different training size N with a fixed number of randomized Pauli measurements $T = 10$, also referred to as the shadow size. (Right) The prediction error (root-mean-square error) over different shadow size T with a fixed training data size $N = 90$. In both plots, the error bars show the standard deviation of RMSE over 100 predictions on a fixed set of spin pairs.

D.4. Classifying phases of the bond-alternating XXZ model

To illustrate our classical ML for classifying quantum phases of matter, we consider the bond-alternating XXZ model with $n = 300$ spin-$\frac{1}{2}$ particles (i.e. qubits). The Hamiltonian is given by

$$\sum_{i:\text{odd}} J(X_iX_{i+1} + Y_iY_{i+1} + \delta Z_iZ_{i+1}) + \sum_{i:\text{even}} J'(X_iX_{i+1} + Y_iY_{i+1} + \delta Z_iZ_{i+1}), \tag{D10}$$

and encompasses the bond-alternating Heisenberg model ($\delta = 1$), as well as the bosonic version of the Su-Schrieffer-Heeger model [158] ($\delta = 0$). The phase diagram in Figure 4(b) is obtained by evaluating the partial reflection many-body topological invariant [56, 136]. It is given by

$$\tilde{\mathcal{Z}}_{\mathcal{R}} = \frac{\mathcal{Z}_{\mathcal{R}}}{\sqrt{[\text{tr}(\rho_{I_1}^2) + \text{tr}(\rho_{I_2}^2)]/2}}, \quad \text{where} \quad \mathcal{Z}_{\mathcal{R}} = \text{tr}(\rho_{I_1\cup I_2}\mathcal{R}_{I_1\cup I_2}), \tag{D11}$$

and we consider I_1 with 6 spins: the 145-th spin to the 150-th spin. Likewise, we fix I_2 to also contain 6 spins: the 151-th spin to the 156-th spin. Hence, the union $I_1 \cup I_2$ contains 12 spins. The symbols ρ_{I_1}, ρ_{I_2} and $\rho_{I_1\cup I_2}$ denote the reduced density matrices associated with each local region. The reflection operator $\mathcal{R}_{I_1\cup I_2}$ acts on the local region $I_1 \cup I_2$ and is given by

$$\mathcal{R}_{I_1\cup I_2} |s_1, \ldots, s_{|I_1\cup I_2|}\rangle = |s_{|I_1\cup I_2|}, \ldots, s_1\rangle, \quad \forall s_1, \ldots, s_{|I_1\cup I_2|} \in \{0, 1\}. \tag{D12}$$

The partial reflection many body-topological invariant can resolve three phases: trivial ($\tilde{\mathcal{Z}}_{\mathcal{R}} = +1$), symmetry-protected topological (SPT) ($\tilde{\mathcal{Z}}_{\mathcal{R}} = -1$) and symmetry broken ($\tilde{\mathcal{Z}}_{\mathcal{R}} = 0$). In Figure 4(b), we use the colors blue (trivial), red (SPT) and gray (symmetry broken) to visualize these different types of phases.

For each value of J'/J and δ considered, we construct the exact ground state using DMRG, and find its classical shadow by performing randomized single-qubit Pauli measurements a total of $T = 500$ times. To simulate this experiment, we follow the same setting for DMRG used in [56]. We limit the maximum number of sweeps to 100 and set the DMRG cutoff to 10^{-9}. We initialize the state to be the Néel state $|0101\ldots\rangle$. To pin one of the degenerate ground state in the symmetry broken phase, we include a penalty term given by $0.1JZ_1$ in the Hamiltonian.

After obtaining the classical shadow representation $S_T(\rho_\ell)$ for each quantum state ρ_ℓ, we compute the kernel matrix $K \in \mathbb{R}^{N\times N}$, where each entry is given by the shadow kernel $k^{(\text{shadow})}(S_T(\rho_\ell), S_T(\rho_{\ell'}))$. Recall that the shadow kernel is defined as

$$k^{(\text{shadow})}(S_T(\rho), S_T(\tilde{\rho})) = \exp\left(\frac{1}{T^2}\sum_{t,t'=1}^{T}\exp\left(\frac{1}{n}\sum_{i=1}^{n}\text{tr}\left(\sigma_i^{(t)}\tilde{\sigma}_i^{(t')}\right)\right)\right), \text{ where } \sigma_i^{(t)} = 3|s_i^{(t)}\rangle\langle s_i^{(t)}| - \mathbb{I}, \tag{D13}$$

and the classical shadow representation is given by

$$S_T(\rho) = \left\{ |s_i^{(t)}\rangle : \; i \in \{1, \ldots, n\}, \; t \in \{1, \ldots, T\} \right\}, \quad \text{where} \quad |s_i^{(t)}\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\} \; . \tag{D14}$$

Care should be taken when computing diagonal elements of the kernel matrix K. The problem is that for $\rho = \tilde{\rho}$ and $t = t'$, we necessarily have $\mathrm{tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t)}\right) = 5$ for all $1 \leq i \leq n$. And the double exponential will amplify this already substantial contribution enormously. We found that counteracting this blow-up improves the numerical stability of the kernel method substantially. When $\ell = \ell'$, when we compute $k^{(\mathrm{shadow})}(S_T(\rho_\ell), S_T(\rho_\ell))$, we sum over $t \neq t'$ instead of all t, t'. In particular, when $\rho = \tilde{\rho}$, we consider a slight modification to the kernel definition,

$$k^{(\mathrm{shadow})}(S_T(\rho), S_T(\rho)) = \exp\left(\frac{1}{T(T-1)} \sum_{t \neq t'} \exp\left(\frac{1}{n} \sum_{i=1}^{n} \mathrm{tr}\left(\sigma_i^{(t)} \sigma_i^{(t')} \right) \right) \right), \tag{D15}$$

This modification also seems to slightly improve the classification performance.

After evaluating the kernel matrix K, we renormalize the entries to obtain the standardized kernel matrix

$$\overline{K}_{\ell\ell'} = \frac{K_{\ell\ell'}}{\sqrt{K_{\ell\ell} K_{\ell'\ell'}}} \quad \text{for} \quad \ell, \ell' \in \{1, \ldots, N\} . \tag{D16}$$

Subsequently, we perform kernel principal component analysis (PCA) on $\overline{K}$. The implementation we used for PCA is based on scikit-learn [27]. The output of kernel PCA is a list of low-dimensional vectors (the dimension can be chosen arbitrarily, but we choose two dimensions for this experiment). Each low-dimensional vector corresponds to a quantum state. In Figure 4(c, d), we can see that the low-dimensional vectors are clustered into different quantum phases of matter.

Distinguishing an SPT phase from a trivial phase — We consider a one-dimensional chain of $n = 50$ qubits with $Z_2 \times Z_2$ symmetry. The 1D cluster state is in the nontrivial SPT phase. We generate other representatives of the nontrivial SPT phase by applying symmetric low-depth geometrically local random quantum circuits to the cluster state, and we generate representatives of the trivial phase by applying symmetric random circuits to a product state. We simulate the application of symmetric low-depth geometrically local random quantum circuits to the cluster state through matrix product states (MPS). Each circuit layer consists of patterns of random two-qubit gates acting on next-to-nearest neighbors sites. We generate the random gates in a block-sparse structure in the parity symmetry sectors. This choice, together with the choice of connectivity, guarantees that the $Z_2 \times Z_2$ symmetry is conserved during the circuit evolution.

Randomized Pauli measurements are performed $T = 500$ times to convert the states to their classical shadows. We perform kernel PCA to find low-dimensional representation for the quantum states using exactly the same method as the experiment on bond-alternating XXZ model.

D.5. Distinguishing a topological phase from a trivial phase

We consider the task of distinguishing the toric code topological phase from the trivial phase in a system of $n = 200$ qubits. Kitaev's toric code state [100] is in the nontrivial topologically-ordered phase, while a product state represents the trivial phase. To populate both phases, we apply low-depth geometrically local random Clifford circuits [1] to Kitaev's toric code state [100] with code distance 10, and we generate representatives of the trivial phase by applying random Clifford circuits to a product state. We utilize Clifford circuits to ensure efficient simulation of in total $n = 200$ qubits (and with a depth up to 9) by means of the Gottesman-Knill theorem. We again perform kernel PCA to find low-dimensional representations for the quantum states using exactly the same method as the experiment on bond-alternating XXZ model. This is used to generate the plot in Figure 5(b) for a one-dimensional projection of the feature space, as well as the plot in Figure 8(d) for a two-dimensional projection.

For the unsupervised ML model shown in Figure 5(c), we consider a combination of kernel PCA and randomized projections [97]. First we perform kernel PCA to map the data to a six-dimensional subspace of the infinite-dimensional feature space. Then we repeat the following procedure 500 times. We select a one-dimensional subspace uniformly at random in the six-dimensional subspace. We project all the quantum states to the one-dimensional subspace. Then, we find the center point (according to median instead of mean) to split up the quantum states into two phases. We also record the sum of the absolute values from all points to the center point in the one-dimensional subspace. Finally, we consider the classification obtained from the random one-dimensional projection that results in the largest sum of the absolute values.

For the convolutional neural network (CNN) approach shown in Figure 5(c), we consider the following CNN built from Keras [40].

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

model = models.Sequential()
model.add(layers.Conv2D(32, (2, 2), activation='relu', padding='same',
            input_shape=(2*L, L, 6)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (2, 2), activation='relu', padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(32, (2, 2), activation='relu', padding='same'))
model.add(layers.Flatten())
model.add(layers.Dense(32, activation='relu'))
model.add(layers.Dense(2))
```

In the above code, L is the code distance for the toric code and is equal to 10 in this experiment (recall that toric code ground state has $n = 2L^2$ qubits). This CNN model is supervised and requires a training data with a corresponding label for indicating which phase the training data point is in. We first perform the Pauli-6 POVM on each qubit [33] to transform the quantum state into a array of size n where each entry has six outcomes. We perform one-hot encoding to yield a classical vector of size $6n$, where each entry in the classical vector is either 0 or 1. Because the toric code ground state is two-dimensional ($2L \times L$), we restructure the classical vector into a three-dimensional tensor of size $2L \times L \times 6$. The first two dimensions corresponds to the spatial dimension of the toric code ground state. The last dimension corresponds to the one-hot encoded vector for the six-outcome POVM. We then train the above model using the Adam optimizer [99] with the categorical cross entropy as the loss function. The code is given below.

```
model.compile(optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
    metrics=['accuracy'])
```

We train the convolutional neural network using 100 training points (half are topologically-ordered states, and the other half are trivial states). Then we use a validation set of 100 points to perform early stopping. This is because the longer we train, the more likely the neural network is going to overfit. Hence, it is a good practice to perform model selection by choosing which model to use at different time points (during the training process). We choose the model that performs the best on the validation set. Then we test the classification accuracy (the percentage that the prediction of the phases is correct) on a testing set consisting of 100 points.

The performance of the above ML model is not substantially different from random guessing. Hence, we also consider a very simple CNN enhanced with classical shadow under $T = 500$ randomized Pauli measurements. In particular, we compute the local reduced density matrix using the classical shadow. Then for each qubit, we represent it with the local reduced density matrix. For simplicity, we consider the i-th qubit to be represented by a vector of size 16, which includes the 2-body reduced density matrix for the subsystem consisting of the i-th and the $i + 1$-th qubit. Hence, each quantum state is now represented by a classical vector of dimension $2L^2 \times 16$. We reshape the classical vector into a three-dimensional tensor of size $2L \times L \times 16$. The classical vector is feed into the convolutional neural network structured as follows. We also apply the Adam optimizer [99] with the categorical cross entropy as the loss function. The evaluation process is exactly the same as the CNN approach based on the Pauli-6 POVM.

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

model = models.Sequential()
model.add(layers.Conv2D(16, (1, 1), activation='relu',\
            padding='same', input_shape=(2*L, L, 16)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(16, (2, 2), activation='relu',\
            padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(16, (2, 2), activation='relu',\
            padding='same'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(32, activation='relu'))
```

```
model.add(layers.Dense(2))

model.compile(optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(
            from_logits=True),
    metrics=['accuracy'])
```

APPENDIX E: Proof idea for the efficiency in predicting ground states

In order to illustrate the proof of Theorem 1, let us begin by looking at a simpler task: training a machine learning model to predict a specified ground state property instead of the classical representation of the ground state. Consider the property $\mathrm{tr}(O\rho)$, where ρ is the ground state and O is a local observable. In this simpler task, we consider the training data to be

$$\big\{x_1 \to \mathrm{tr}(O\rho(x_1)), \quad \ldots, \quad x_N \to \mathrm{tr}(O\rho(x_N))\big\}, \tag{E1}$$

where $x_\ell \in [-1,1]^m$ is a classical description of the Hamiltonian $H(x_\ell)$ and $\rho(x_\ell)$ is the ground state of $H(x)$. Intuitively, in a quantum phase of matter, the ground state property $\mathrm{tr}(O\rho(x))$ changes smoothly as a function of the input parameter x. The smoothness condition can be rigorously established as an upper bound on the average magnitude of the gradient of $\mathrm{tr}(O\rho(x))$ using quasi-adiabatic evolution [13, 81], assuming that the spectral gap of $H(x)$ is bounded below by a nonzero constant throughout the parameter space. The upper bound on the average gradient magnitude enables us to design a simple classical ML model based on an l_2-Dirichlet kernel for generalizing from the training set to a new input $x \in [-1,1]^m$:

$$\hat{O}_N(x) = \frac{1}{N}\sum_{\ell=1}^{N} \kappa(x,x_\ell)\,\mathrm{tr}(O\rho(x_\ell)) \;\text{ with }\; \kappa(x,x_\ell) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \cos(\pi k\cdot(x-x_\ell)) \in \mathbb{R}. \tag{E2}$$

The l_2-Dirichlet kernel is often used in the study of high-dimensional Fourier series [179] and the proposed ML model is equivalent to learning a truncated Fourier series to approximate the function $\mathrm{tr}(O\rho(x))$, where the parameter Λ is a cutoff on the wavenumber k that depends on the upper bound on the gradient of $\mathrm{tr}(O\rho(x))$. Using statistical analysis, one can guarantee that $\mathbb{E}_x\,|\hat{O}_N(x) - \mathrm{tr}(O\rho(x))|^2 \leq \epsilon$ as long as the amount of training data $N = m^{\mathcal{O}(1/\epsilon)}$ where our big-$\mathcal{O}$ notation is with respect to the $m\to\infty$ limit. Hence, we can achieve a small *constant* prediction error with an amount of training data and computational time that are both polynomial in the number m of input parameters. The training is efficient because the number of modes needed for the truncated Fourier series to provide an accurate approximation to $\mathrm{tr}(O\rho)$ scales polynomially with m.

The key to the statistical analysis is to bound the model complexity of the above machine learning model. In particular, the model complexity depends on the number of wave vectors we consider in the l_2-Dirichlet kernel. The more wave vectors k we include, the higher the model complexity; and we would have to use more data to train the ML model to achieve good generalization performance. Furthermore, one could show that the amount of data is proportional to the number of wave vectors we consider. In order to achieve a prediction error $\mathbb{E}_x\,|\hat{O}_N(x) - \mathrm{tr}(O\rho(x))|^2 \leq \epsilon$, we would need to select Λ to be of order $\sqrt{1/\epsilon}$. Hence, the number of wave vectors is proportional to the number of lattice points in an m-dimensional l_2 ball of radius Λ. The volume of an m-dimensional l_2 ball with radius Λ is proportional to $\Lambda^m = (1/\epsilon)^{m/2}$. If the number of lattices points is proportional to the volume, then this would imply an exponential scaling in the number of parameters m. However, through a proper combinatorial analysis, we show that the number of lattices points is actually proportional to $m^{\mathcal{O}(\Lambda^2)} = m^{\mathcal{O}(1/\epsilon)}$, which is only polynomial in the number of parameters m.

We can build on this idea to address the task of predicting ground state representations. Now instead of predicting $\mathrm{tr}(O\rho)$ for a new input x, the goal is to predict the classical shadow of the ground state $\rho(x)$. We consider the training data to be $\big\{x_\ell \to \sigma_1(\rho(x_\ell))\big\}_{\ell=1}^N$, where $\sigma_1(\rho(x_\ell))$ is the classical shadow representation of $\rho(x_\ell)$ obtained from just a *single* randomized Pauli measurement of the state (the $T=1$ case of Eq. (1)). Following the same approach as outlined above for the case of predicting a single property, the predicted ground state representation is now given by

$$\hat{\sigma}_N(x) = \frac{1}{N}\sum_{\ell=1}^{N} \kappa(x,x_\ell)\sigma_1(\rho(x_\ell)) \;\text{ with }\; \kappa(x,x_\ell) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \cos(\pi k\cdot(x-x_\ell)) \in \mathbb{R}. \tag{E3}$$

One can then guarantee that this representation accurately predicts expectation values for a wide range of observables.

The fact that only a single snapshot σ_1 per parameter point is required for our protocol may be surprising. However, since the snapshots depends on the parameters, sampling over training data indirectly samples over different snapshots, and is thus sufficient for a reasonable estimate of properties of the phase. The estimate can of course be further improved if multiple snapshots are used for each parameter point, and we leave proving such improved bounds as an exciting goal for future work.

APPENDIX F: Proof of efficiency for predicting ground states

This section contains a detailed proof for one of our main contributions. Namely, a rigorous performance guarantee for learning to predict ground state representations.

Theorem 3 (Theorem 1, detailed restatement)**.** *Consider any family of n-qubit geometrically-local Hamiltonians $\{H(x) : x \in [-1,1]^m\}$ in a finite spatial dimension, such that each local term in $H(x)$ depends smoothly on x, and the smallest eigenvalue and the next smallest eigenvalues have a constant gap $\gamma \geq \Omega(1)$ between them. Let $\rho(x)$ be the ground state of $H(x)$, that is*

$$\rho(x) = \lim_{\beta\to\infty} e^{-\beta H(x)} / \operatorname{tr}(e^{-\beta H(x)}) \in (\mathbb{H}_2)^{\otimes n} \qquad \textit{(ground state of Hamiltonian } H(x)\textit{)} \tag{F1}$$

where $\mathbb{H}_2$ is the vector space of 2×2 Hermitian matrices. Suppose that we are interested in learning to predict a sum $O = \sum_{i=1}^L O_i$ of L local observables that satisfies $\sum_{i=1}^L \|O_i\| \leq B$ (bounded norm). Then, classical shadow data $\{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$, with $x_\ell \sim \mathrm{Unif}[-1,1]^m$ and

$$N = B^2 m^{\mathcal{O}(B^2/\epsilon)} \qquad \textit{(training data size)}, \tag{F2}$$

suffices to produce a ground state prediction model

$$\hat{\sigma}_N(x) = \frac{1}{N}\sum_{\ell=1}^N \kappa(x,x_\ell)\operatorname{tr}(O\rho(x_\ell)) \;\; \textit{with} \;\; \kappa(x,x_\ell) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \cos(\pi k\cdot(x-x_\ell)) \in \mathbb{R}, \tag{F3}$$

that achieves

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}_N(x)) - \operatorname{tr}(O\rho(x))|^2 \leq \epsilon, \tag{F4}$$

with high probability. The classical training time for constructing $\hat{\sigma}_N(x)$ and the prediction time for computing $\operatorname{tr}(O\hat{\sigma}(x))$ are both upper bounded by $\mathcal{O}((n+L)B^2 m^{\mathcal{O}(B^2/\epsilon)})$.

Theorem 3 can be generalized to the following statement about learning a family of quantum states. In particular, we will prove the following theorem and use it to derive Theorem 3.

Theorem 4. *Consider a parametrized family of n-qubit states $\{\rho(x) : x \in [-1,1]^m\}$ and a sum $O = \sum_{i=1}^L O_i$ of L local observables that obey*

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \leq C \quad \textit{(smoothness condition)}, \tag{F5a}$$

$$\sum_i \|O_i\| \leq B \quad \textit{(bounded norm)}. \tag{F5b}$$

Then, classical shadow data $\{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$, with $x_\ell \sim \mathrm{Unif}[-1,1]^m$ and

$$N = B^2 m^{\mathcal{O}(C/\epsilon)} \qquad \textit{(training data size)}, \tag{F6}$$

suffices to produce a state prediction model we can learn from classical data $\{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$ to produce a model

$$\hat{\sigma}_N(x) = \frac{1}{N}\sum_{\ell=1}^N \kappa(x,x_\ell)\operatorname{tr}(O\rho(x_\ell)) \;\; \textit{with} \;\; \kappa(x,x_\ell) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \cos(\pi k\cdot(x-x_\ell)) \in \mathbb{R}, \tag{F7}$$

that achieves

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}_N(x)) - \operatorname{tr}(O\rho(x))|^2 \leq \epsilon, \tag{F8}$$

with high probability. The classical training time for constructing $\hat{\sigma}_N(x)$ and the prediction time for computing $\operatorname{tr}(O\hat{\sigma}(x))$ are both upper bounded by $\mathcal{O}((n+L)B^2 m^{\mathcal{O}(C/\epsilon)})$.

The following sections are structured as follows. In Section F.1, we provide an overview to illustrate the proof of the sample complexity upper bound. The first step, given in Section F.2, bounds the truncation error when approximating the quantum state function $\rho(x)$ using a truncated Fourier series. The second step, given in Section F.3, bounds the generalization error for learning the Fourier approximation to the quantum state function $\rho(x)$. Then, in Section F.4, we analyze the training and prediction time of the proposed classical machine learning model. These three sections establish Theorem 4. Finally, in Section F.5, we use Theorem 4 and nice properties about ground states of Hamiltonians to prove Theorem 3.

F.1. Overview for sample complexity upper bound

The key intermediate step is to construct a truncated Fourier series of the quantum state function $\rho(x)$. The Fourier series of the matrix-valued function $\rho(x)$ is given as

$$\rho(x) = \sum_{k\in\mathbb{Z}^m} e^{\mathrm{i}\pi k\cdot x} A_k, \tag{F9}$$

where A_k are matrix-valued Fourier coefficients

$$A_k = \frac{1}{2^m}\int_{[-1,1]^m} e^{-\mathrm{i}\pi k\cdot x}\rho(x)\mathrm{d}^m x. \tag{F10}$$

We define the truncated Fourier series as

$$\rho_\Lambda(x) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} e^{\mathrm{i}\pi k\cdot x} A_k, \tag{F11}$$

where $\Lambda > 0$ is a pre-specified cutoff value. Given an observable O that can be written as a sum of local observables $O = \sum_i O_i$ with $\sum_i \|O_i\|_\infty \leq B$ and $\mathbb{E}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \leq C$, the proof of Theorem 3 consists of two parts.

First, we bound the error between the truncated Fourier series $\rho_\Lambda(x)$ and the true quantum state function $\rho(x)$ in Section F.2 giving

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\rho(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \mathcal{O}\left(\frac{C}{\Lambda^2}\right), \tag{F12}$$

We choose the truncation $\Lambda = \Theta(\sqrt{C/\epsilon})$ such that the error between truncated Fourier series and the true quantum state function obeys

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\rho(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \frac{\epsilon}{4}. \tag{F13}$$

In the second part, we bound the error between the machine learning model $\hat{\sigma}(x)$ and the truncated Fourier series $\rho_\Lambda(x)$ in Section F.3. With high probability over the randomness in generating the training data, we have

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \frac{B^2 m^{\mathcal{O}(\Lambda^2)}}{N}. \tag{F14}$$

The training data contains two sources of randomness, one from the sampling of x_ℓ and the other from the local randomized measurement to construct approximate classical representation for $\rho(x_\ell)$ that could be feed into the classical machine learning model. We choose the training data size

$$N = \frac{2B^2 m^{\mathcal{O}(C/\epsilon)}}{\epsilon} \leq B^2 m^{\mathcal{O}(C/\epsilon)+\log(1/\epsilon)+1} = B^2 m^{\mathcal{O}(C/\epsilon)}, \tag{F15}$$

such that the error between the machine learning model and the truncated Fourier series obeys

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \epsilon/4, \tag{F16}$$

with high probability. The two parts can be combined by a triangle inequality to yield

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho(x))|^2 \tag{F17a}$$

$$\leq \left(\sqrt{\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \left|\mathrm{tr}(O\hat{\sigma}(x)) - \mathrm{tr}\left(O\rho_\Lambda(x)\right)\right|^2} + \sqrt{\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \left|\mathrm{tr}(O\rho(x)) - \mathrm{tr}\left(O\rho_\Lambda(x)\right)\right|^2} \right)^2 = \epsilon, \tag{F17b}$$

with high probability over the randomness in the training data. This establishes the sample complexity upper bound for Theorem 3.

When the Hamiltonians $H(x)$ have spectral gap $\geq \Omega(1)$ in the domain $x \in [-1,1]^m$, for any observable $O = \sum_i O_i$ that can be written as a sum of local observables with $\sum_i \|O_i\|_\infty \leq B$, we have

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x \mathrm{tr}(O\rho(x))\|_2^2 \leq \mathcal{O}(B^2). \tag{F18}$$

Hence, we can prove the sample complexity upper bound in Theorem 4 by utilizing Theorem 3 and the fact that $C = \mathcal{O}(B^2)$.

F.2. Controlling the truncation error

For a fixed observable O, we can define a function

$$f(x) = \mathrm{tr}(O\rho(x)) = \sum_{k\in\mathbb{Z}^m} e^{\mathrm{i}\pi k\cdot x} \mathrm{tr}(OA_k). \tag{F19}$$

And the truncated Fourier series of the function $f(x)$ is given by

$$f_\Lambda(x) = \mathrm{tr}(O\rho_\Lambda(x)) = \rho_\Lambda(x) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} e^{\mathrm{i}\pi k\cdot x} \mathrm{tr}(OA_k). \tag{F20}$$

Lemma 2 (truncation error)**.** *Let* $f(x) = \sum_{k\in\mathbb{Z}^m} \alpha_k e^{\mathrm{i}\pi k\cdot x}$ *and* $f_\Lambda(x) = \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} \alpha_k e^{\mathrm{i}\pi k\cdot x}$*. Then*

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |f(x) - f_\Lambda(x)|^2 \leq \frac{1}{\pi^2\Lambda^2} \mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x f(x)\|_2^2 \quad \textit{for any cutoff} \quad \Lambda > 0. \tag{F21}$$

Proof. The claim follows from standard Harmonic analysis arguments. More precisely, we combine *orthogonality* ($\int_{[-1,1]^m} \mathrm{e}^{\mathrm{i}(\pi(k-k')x} \mathrm{d}^m x = \delta_{(k,k')}$) with the fact that the Fourier transform exchanges differentials ("momentum") with multiplications ("position"):

$$\nabla_x f(x) = \sum_{k\in\mathbb{Z}^m} \alpha_k \nabla_x \mathrm{e}^{\mathrm{i}\pi kx} = \mathrm{i}\pi \sum_{k\in\mathbb{Z}^m} \alpha_k k \mathrm{e}^{\mathrm{i}\pi kx}. \tag{F22}$$

Use orthogonality to rewrite the truncation error as

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |f(x) - f_\Lambda(x)|^2 = \int_{[-1,1]^m} \Big| \sum_{k\in\mathbb{Z}^m:\|k\|>\Lambda} \mathrm{e}^{\mathrm{i}\pi kx} \alpha_k \Big|^2 \mathrm{d}^m x \tag{F23a}$$

$$= \sum_{k:\|k\|_2>\Lambda} \sum_{k':\|k'\|_2>\Lambda} \Big(\int_{[-1,1]^m} \mathrm{e}^{\mathrm{i}\pi(k-k')x} \mathrm{d}^m x \Big) \overline{\alpha_k} \alpha_k \tag{F23b}$$

$$= \sum_{k:\|k\|_2>\Lambda} |\alpha_k|^2 . \tag{F23c}$$

Conversely, we use orthogonality and Rel. (F22) to rephrase this upper bound. Let $\langle k', k\rangle$ be the Euclidean inner product between two vectors $k, k' \in \mathbb{Z}^m$. Then,

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x f(x)\|_2^2 = \int_{[-1,1]^m} \Big\| \sum_{k\in\mathbb{Z}^m} \pi k \mathrm{e}^{\mathrm{i}\pi kx} \alpha_k \Big\|_2^2 \mathrm{d}^m x \tag{F24a}$$

$$= \sum_{k,k'\in\mathbb{Z}^m} \pi^2 \langle k', k\rangle \int_{[-1,1]^m} \mathrm{e}^{\mathrm{i}\pi(k-k')x} \mathrm{d}^m x \overline{\alpha_{k'}} \alpha_k \tag{F24b}$$

$$= \pi^2 \sum_{k\in\mathbb{Z}^m} \langle k, k\rangle |\alpha_k|^2 = \pi^2 \sum_{k\in\mathbb{Z}^m} \|k\|_2^2 |\alpha_k|^2 . \tag{F24c}$$

In words, the upper bound from Eq. (F23c) can be rephrased as the Euclidean norm $\|\nabla_x f(x)\|_2^2$ of the vector $\nabla_x f(x)$. The advertised claim readily follows from comparing these two reformulations:

$$\sum_{k:\|k\|_2>\Lambda} |\alpha_k|^2 \leq \frac{1}{\Lambda^2} \sum_{k:\|k\|_2>\Lambda} \|k\|_2^2 |\alpha_k|^2 \leq \frac{1}{\pi^2\Lambda^2}\Big(\pi^2 \sum_{k\in\mathbb{Z}^m} \|k\|_2^2 |\alpha_k|^2\Big). \tag{F25}$$

□

Using Lemma 2 and the condition that $\mathbb{E}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \leq C$, we can obtain the desired inequality for bounding the truncation error,

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\rho(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \mathcal{O}\left(\frac{C}{\Lambda^2}\right). \tag{F26}$$

F.3. Controlling generalization errors from using the training data

This section is devoted to a practical issue regarding training data based on classical shadows. Each label is obtained by performing a single-shot quantum measurement of a parametrized quantum state $\rho(x_i)$. We can use Eq. (A3) to convert the single-shot outcome into $\sigma_1(\rho) = \bigotimes_{i=1}^n (3|s_i\rangle\langle s_i| - \mathbb{I})$. Such a classical shadow approximation reproduces the underlying state in expectation, i.e., $\mathbb{E}_{s_1,\ldots,s_n}[\sigma_1(\rho)] = \rho$. Recall that the training data $\mathcal{T} = \{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$ consists of such classical shadow approximations. The machine learning model makes predictions based on a truncated Fourier kernel for future predictions. For new input $x \in [-1,1]^n$, we predict

$$\hat{\sigma}(x) = \frac{1}{N}\sum_{\ell=1}^N \kappa(x,x_\ell)\sigma_1(\rho(x_\ell)) \quad \text{with} \tag{F27a}$$

$$\kappa(x,x_\ell) = \sum_{k\in\mathbb{Z}^m,\|k\|_2\leq\Lambda} \mathrm{e}^{\mathrm{i}\pi k\cdot(x-x_\ell)} = \sum_{k\in\mathbb{Z}^m,\|k\|_2\leq\Lambda} \cos(\pi k\cdot(x-x_\ell)). \tag{F27b}$$

In the following, we will show that machine learning model $\hat{\sigma}(x)$ is equal to the truncated Fourier series $\rho_\Lambda(x)$ of the true target quantum state if we take the expectation over the training data, which includes the sampled inputs $x_1,\ldots,x_N$ and the randomized measurement outcomes $S_1(\rho(x_\ell)) = \{s_i\}_{i=1}^n$ for each input x_ℓ. Moreover, statistical flucutations due to shot noise will be small provided that we are interested in predicting an observable that decomposes nicely as a sum of local terms. These observations are the content of the following statement.

Lemma 3 (Statistical properties of the predicted quantum state $\hat{\sigma}(x)$)**.** *Let $\mathcal{T} = \{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$ be a training set featuring uniformly random inputs $x_\ell \overset{unif}{\sim} [-1,1]^m$ and classical shadows of the associated quantum states as labels. Then, the machine learning model obeys*

$$\mathop{\mathbb{E}}_{\mathcal{T}}[\hat{\sigma}(x)] = \rho_\Lambda(x) = \sum_{k\in\mathbb{Z}^m,\|k\|_2\leq\Lambda} e^{\mathrm{i}\pi k\cdot x} A_k. \tag{F28}$$

Moreover, suppose that an observable $O = \sum_i O_i$ decomposes into a sum of q-local terms. Then, with probability at least $1-\delta$, we have

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 \leq \frac{1}{N} 9^q \big(\sum_i \|O_i\|_\infty\big)^2 (2m+1)^{\Lambda^2} \left(\Lambda^2 \log(2m+1) + \log(4/\delta)\right). \tag{F29}$$

The advertised bound can be further streamlined if the observable locality q and confidence level δ are constant. Assuming $q, \delta = \mathcal{O}(1)$ ensures the following simplified scaling:

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 = \mathcal{O}\left(\frac{1}{N}\left(\sum_i \|O_i\|\right)^2 (2m+1)^{\Lambda^2+\log(\Lambda^2)+1}\right) = \frac{\left(\sum_i \|O_i\|\right)^2 m^{\mathcal{O}(\Lambda^2)}}{N}. \tag{F30}$$

Using the condition that $\sum_i \|O_i\| \leq B$, we have

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho_\Lambda(x))|^2 = \frac{B^2 m^{\mathcal{O}(\Lambda^2)}}{N}, \tag{F31}$$

which controls the generalization error from quantum measurements. The argument is based on fundamental properties of classical shadows that have been reviewed in Appendix A.

Proof of Lemma 3. We begin by condensing notation somewhat. Here, we only consider classical shadows of size $T = 1$. Hence, we may replace the superscript (t) by (x_ℓ) to succinctly keep track of classical input parameters. More precisely, we let $|s_i^{(x_\ell)}\rangle$ be the randomized Pauli measurement outcome for the i-th qubit when measuring the quantum state $\rho(x_\ell)$. The training data $\mathcal{T} = \{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$ is determined by the following random variables

$$x_\ell \in [-1,1]^m, \qquad \text{for } \ell \in \{1,\dots,N\}, \tag{F32a}$$

$$s_i^{(x_\ell)} \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |\mathrm{i}+\rangle, |\mathrm{i}-\rangle\}, \qquad \text{for } i \in \{1,\dots,n\} \text{ and } \ell \in \{1,\dots,N\}. \tag{F32b}$$

The first claim is an immediate consequence of Eq. (A6):

$$\mathop{\mathbb{E}}_{\mathcal{T}}[\hat{\sigma}(x)] = \frac{1}{N}\sum_{\ell=1}^N \mathop{\mathbb{E}}_{x_\ell\sim[-1,1]^m}\left[\kappa(x,x_\ell) \mathop{\mathbb{E}}_{s_1^{(x_\ell)},\dots,s_n^{(x_\ell)}}[\sigma_1(\rho(x_\ell))]\right] \tag{F33a}$$

$$= \frac{1}{N}\sum_{\ell=1}^N \mathop{\mathbb{E}}_{x_\ell\sim[-1,1]^m}[\kappa(x,x_\ell)\rho(x_\ell)] \tag{F33b}$$

$$= \mathop{\mathbb{E}}_{x_1\sim[-1,1]^m}[\kappa(x,x_1)\rho(x_1)] \tag{F33c}$$

$$= \sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda} e^{\mathrm{i}\pi k\cdot x} \mathop{\mathbb{E}}_{x_1\sim[-1,1]^m}\left[e^{-\mathrm{i}\pi k\cdot x_1}\rho(x_1)\right] \tag{F33d}$$

$$= \sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda} e^{\mathrm{i}\pi k\cdot x_1}\frac{1}{2^m}\int_{[-1,1]^m} e^{-\mathrm{i}\pi k\cdot x_1}\rho(x)\mathrm{d}^m x_1 \tag{F33e}$$

$$= \sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda} e^{\mathrm{i}\pi k\cdot x} A_k \tag{F33f}$$

$$= \rho_\Lambda(x). \tag{F33g}$$

Here, we have also used the fact that each x_ℓ is sampled independently and uniformly from $[-1,1]^m$.

The second result is contingent on the training data for predicting the ground state representation $\mathcal{T} = \{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$. We begin with using the definitions of $\hat{\sigma}(x)$ (F27) and $\rho_\Lambda(x)$ to rewrite the expression of interest as

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\mathrm{tr}(O\hat{\sigma}(x)) - \mathrm{tr}(O\rho_\Lambda(x))|^2$$

$$= \frac{1}{2^m}\int_{[-1,1]^m}\mathrm{d}^m x \left|\sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda} \mathrm{e}^{\mathrm{i}\pi k\cdot x}\left(\frac{1}{N}\sum_{\ell=1}^N \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right) - \mathrm{tr}\left(OA_k\right)\right)\right|^2 \tag{F34a}$$

$$= \sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda}\left|\frac{1}{N}\sum_{\ell=1}^N \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right) - \mathrm{tr}\left(OA_k\right)\right|^2, \tag{F34b}$$

$$\equiv \sum_{k\in\mathbb{Z}^m,\|k\|_2\le\Lambda} D_{(k)}(\mathcal{T})^2\,, \tag{F34c}$$

where we have evaluated the Fourier integral over x and introduced shorthand notation $D_{(k)}(\mathcal{T})^2$ for each summand.

The next key step is to notice that each A_k is an expectation value over both the parameters and the shadows. Writing out A_k and expressing ρ in terms of shadows using Eq. (A6),

$$A_k = \frac{1}{2^m}\int_{[-1,1]^m} \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\rho(x_\ell)\right)\mathrm{d}^m x_\ell \tag{F35a}$$

$$= \mathop{\mathbb{E}}_{x_\ell\sim[-1,1]^m} \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\rho(x_\ell)\right) \tag{F35b}$$

$$= \mathop{\mathbb{E}}_{x_\ell \text{ and } s_1^{(x_\ell)},\dots,s_n^{(x_\ell)}} \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right)\,. \tag{F35c}$$

Plugging this back into the summand in Eq. (F34c) yields

$$D_{(k)}(\mathcal{T})^2 = \left|\frac{1}{N}\sum_{\ell=1}^N \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right) - \mathop{\mathbb{E}}_{x_\ell \text{ and } s_1^{(x_\ell)},\dots,s_n^{(x_\ell)}} \mathrm{e}^{-\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right)\right|^2. \tag{F36}$$

Therefore, each $D_{(k)}(\mathcal{T})^2$ is the (square-)deviation of an empirical average from the true expectation value A_k. Hence, we can use Hoeffding's inequality to bound it, provided that O is local and bounded. This may come as a surprise, as the empirical average samples only different parameters x_ℓ and not different shadows σ. However, the shadows depend on the parameters, so sampling only over the parameters turns out to be sufficient for a reasonable estimate.

In order to apply Hoeffding's inequality, we first have to make sure the expectation value is bounded. Recall that $O = \sum_i O_i$ decomposes nicely into a sum of q-body terms. More formally, $\mathrm{supp}(O_j) \subset \{1, \ldots, n\}$ contains at most q qubits. We also know trace and trace norm of each single-qubit contribution to $\sigma_1(\rho(x_\ell))$, $\mathrm{tr}\big(3|s_j^{(x_\ell)}\rangle\langle s_j^{(x_\ell)}| - \mathbb{I}\big) = 1$, and Eq. (A4) asserts $\big\|3|s_j^{(x_\ell)}\rangle\langle s_j^{(x_\ell)}| - \mathbb{I}\big\|_1 = 3$. The matrix Hoelder inequality then implies, for every $x_\ell \in [-1, 1]^m$,

$$\begin{aligned}
\left|\mathrm{e}^{\mathrm{i}\pi k\cdot x_\ell}\,\mathrm{tr}\left(O\sigma_1\left(\rho(x_\ell)\right)\right)\right| &\leq \sum_i \left|\mathrm{tr}\left(O_i\sigma_1(\rho(x_\ell))\right)\right| && \text{(F37a)}\\
&= \sum_i \left|\mathrm{tr}\left(O_{A_i}\mathrm{tr}_{\neg A_i}\left(\sigma_1\left(\rho(x_\ell)\right)\right)\right)\right| && \text{(F37b)}\\
&\leq \sum_i \|O_{A_i}\|_\infty \left\|\mathrm{tr}_{\neg A_i}\left(\sigma_1\left(\rho(x_\ell)\right)\right)\right\|_1 && \text{(F37c)}\\
&= \sum_i \|O_i\|_\infty \prod_{j\in\mathrm{supp}(O_i)} \left\|3|s_j^{(x_\ell)}\rangle\langle s_j^{(x_\ell)}| - \mathbb{I}\right\|_1 && \text{(F37d)}\\
&= \sum_i \|O_i\|_\infty 3^{|\mathrm{supp}(O_j)|} \leq 3^q \sum_i \|O_i\|_\infty. && \text{(F37e)}
\end{aligned}$$

Thus, the expectation value is bounded.

We are now ready to bound the likelihood of a large deviation $D_{(k)}(\mathcal{T})^2$. To recap, for each $k \in \mathbb{Z}^m$ obeying $\|k\|_2 \leq \Lambda$, we face a contribution that collects the (square-)deviation of a sum of *iid* and *bounded* random variables around their expectation value. These variables are complex, but one can analyze their real and imaginary parts separately and collect them into a complex version of Hoeffding's inequality:

$$\begin{aligned}
\Pr\left[D_{(k)}(\mathcal{T})^2 \geq \tau^2\right] &= \Pr\left[D_{(k)}(\mathcal{T}) \geq \tau\right] && \text{(F38a)}\\
&\leq 2\exp\left(-\frac{2N\tau^2}{9^q\left(\sum_i \|O_i\|\right)^2}\right) \quad \text{for all} \quad \tau > 0. && \text{(F38b)}
\end{aligned}$$

This concentration bound connects training data size $N = |\mathcal{T}|$ with the size of a (fixed, but arbitrary) contribution to the expected deviation (F34). For fixed magnitude τ and confidence δ, there is always a (finite) training data size $N = N(\tau, \delta)$ that ensures $D_{(k)}(\mathcal{T})^2 \leq \tau$ with probability at least $1 - \delta$. We can extend this reasoning to the entire sum in Eq. (F34) by exploiting that Rel. (F38) is independent of k, and the summation only ranges over finitely many terms. Introduce $K_\Lambda = |\{k \in \mathbb{Z}^m : \|k\|_2 \leq \Lambda\}|$ — the number of wave-vectors $k \in \mathbb{Z}^m$ whose Euclidean norm is bounded by Λ — and apply a union bound to conclude

$$\begin{aligned}
\Pr\left[\sum_{k\in\mathbb{Z}^m,\|k\|_2\leq\Lambda} D_{(k)}(\mathcal{T})^2 \geq K_\Lambda\tau^2\right] &\leq \Pr\left[\exists k \in \mathbb{Z}^m : \|k\|_2 \leq \Lambda,\ \text{s.t. } D_{(k)}(\mathcal{T})^2 \geq \tau^2\right] && \text{(F39a)}\\
&\leq \sum_{k\in\mathbb{Z}^m,\|k\|_2\leq\Lambda} \Pr\left[D_{(k)}(\mathcal{T})^2 \geq \tau^2\right] && \text{(F39b)}\\
&\leq 2K_\Lambda \exp\left(-\frac{2N\tau^2}{9^q\left(\sum_i \|O_i\|\right)^2}\right). && \text{(F39c)}
\end{aligned}$$

for all $\tau > 0$. To finish the argument, we take guidance from Eq. (F38). Fix a confidence level $\delta \in (0, 1)$ and set

$$\tau^2 = \frac{1}{N}9^q\left(\sum_i \|O_i\|\right)^2 \log(2K_\Lambda/\delta) \quad \text{to ensure} \quad \Pr\left[\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \left|\mathrm{tr}(O\hat{\sigma}(x)) - \mathrm{tr}\left(O\rho_\Lambda(x)\right)\right|^2 \geq K_\Lambda\tau^2\right] \leq \delta. \tag{F40}$$

The advertised bound follows from inserting an explicit bound on the number of relevant wavevectors:

$$K_\Lambda = |\{k \in \mathbb{Z}^m :\ \|k\|_2 \leq \Lambda\}| \leq (2m + 1)^{\Lambda^2}. \tag{F41}$$

To see this, note that $\|k\|_2^2 = \sum_{i=1}^n |k_i|^2 \geq \sum_{i=1}^n |k_i| = \|k\|_1$, because $k_i \in \mathbb{Z}$. Conversely, every $k \in \mathbb{Z}^m$ that obeys $\|k\|_2 \leq \Lambda$ also obeys $\|k\|_1 \leq \Lambda^2$. Next, we enumerate all wave-vectors that obey the relaxed condition $\|k\|_1 \leq \Lambda^2$. To this end, we consider a simple process: select an index $i \in [m]$, and update the associated wave number by $+1$ (increment), 0 (do nothing) or -1 (decrement). Repeating this process a total of Λ^2 times allows us to generate no more than $(2m+1)^{\Lambda^2}$ different wavevectors. But, at the same time, every wave vector $k \in \mathbb{Z}^m$ that obeys $\|k\|_2 \leq \Lambda^2$ can be reached in this fashion. Hence, we conclude $K_\Lambda \leq \left|\left\{k \in \mathbb{Z}^m : \|k\|_1 \leq \Lambda^2\right\}\right| \leq (2m+1)^{\Lambda^2}$. □

F.4. Computational time for training and prediction

We have proposed a very simple prediction model that is based on approximating a truncated Fourier series (l_2-Dirichlet kernel). The training time is equivalent to loading the training data $\mathcal{T} = \{x_\ell \to \sigma_1(\rho(x_\ell))\}_{\ell=1}^N$. Only a single snapshot is provided for each sampled parameter x_ℓ (i.e., $T = 1$), so we relabel $s^{(t)} \to s^{(x_\ell)}$. The training data is given by the collection of x_ℓ and shadows $\{s_i^{x_\ell}\}_{i=1}^n$, following Eq. (F32). Therefore, one only needs

$$\mathcal{O}\left((n+m)N\right) = \mathcal{O}\left((n+m)B^2 m^{\mathcal{O}(C/\epsilon)}\right) = \mathcal{O}\left(nB^2 m^{\mathcal{O}(C/\epsilon)}\right) \quad \text{(training time)} \tag{F42}$$

computational time to load the relevant data into a classical memory. Next, suppose that $O = \sum_{i=1}^L O_i$ is comprised of L q-local terms. Then, we can compute the associated expectation value for the predicted quantum state $\hat{\sigma}(x)$ by evaluating

$$\operatorname{tr}(O\hat{\sigma}(x)) = \frac{1}{N}\sum_{\ell=1}^N \sum_{i=1}^L \kappa(x, x_\ell) \operatorname{tr}(O_i \sigma_1(\rho(x_\ell))). \tag{F43}$$

Recall that the kernel function is defined as

$$\kappa(x, x_\ell) = \sum_{k \in \mathbb{Z}^m, \|k\|_2 \leq \Lambda} \mathrm{e}^{\mathrm{i}\pi k \cdot (x - x_\ell)} = \sum_{k \in \mathbb{Z}^m, \|k\|_2 \leq \Lambda} \cos(\pi k \cdot (x - x_\ell)). \tag{F44}$$

This can be computed in time $\mathcal{O}(K_\Lambda)$, where $K_\Lambda = |\{k \in \mathbb{Z}^m : \|k\|_2 \leq \Lambda\}| \leq (2m+1)^{\Lambda^2}$, according to Rel. (F41) above. Because we have chosen $\Lambda = \Theta(\sqrt{C/\epsilon})$, the runtime to evaluate one kernel function is upper bounded by $m^{\mathcal{O}(C/\epsilon)}$.

On the other hand, the computation of each $\operatorname{tr}(O_j \sigma_1(\rho(x_\ell)))$ can be performed in constant time after storing the data in a classical memory. This is a consequence of the tensor product structure of $\sigma_1(\rho(x_\ell)) = \bigotimes_{i=1}^n \left(3|s_i^{(x_\ell)}\rangle\langle s_i^{(x_\ell)}| - \mathbb{I}\right)$ which ensures

$$\operatorname{tr}(O_j \sigma_1(\rho(x_\ell))) = \operatorname{tr}\left(O_j \bigotimes_{i \in \operatorname{supp}(O_j)} \left(3|s_i^{(x_\ell)}\rangle\langle s_i^{(x_\ell)}| - \mathbb{I}\right)\right), \tag{F45}$$

where $\operatorname{supp}(O_j)$ is the set of qubits in $\{1, \ldots, n\}$ the local observable O_j acts on. Because $|\operatorname{supp}(O_j)| \leq q = \mathcal{O}(1)$, computing $\operatorname{tr}(O_j \sigma_1(\rho(x_\ell)))$ takes only constant time. However, the computation time does scale exponentially in $|\operatorname{supp}(O_j)|$. This can become a problem if $|\operatorname{supp}(O_j)|$ ceases to be a *small* constant. Putting everything together implies that $\operatorname{tr}(O\hat{\sigma}(x))$ can be computed in time (at most)

$$\mathcal{O}\left(NLm^{\mathcal{O}(C/\epsilon)}\right) = \mathcal{O}\left(LB^2 m^{\mathcal{O}(C/\epsilon)}\right) \qquad \text{(prediction time)}. \tag{F46}$$

We conclude that both classical training time and prediction time for $\operatorname{tr}(O\hat{\sigma}(x))$ are upper bounded by

$$\mathcal{O}((n+L)B^2 m^{\mathcal{O}(C/\epsilon)}). \tag{F47}$$

This concludes the proof of all statements given in Theorem 4.

F.5. Spectral gap implies smooth parametrizations

We attempt to deduce Theorem 3 from Theorem 4. The key step involves showing that the ground state $\rho(x)$ in a quantum phase of matter satisfies the following condition: For any observable $O = \sum_i O_i$ that can be written as a sum of local observables with $\sum_i \|O_i\|_\infty \leq B$, we have

$$\mathop{\mathbb{E}}_{x \sim [-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \leq \mathcal{O}(B^2). \tag{F48}$$

Then we can apply Theorem 4 with $C = \mathcal{O}(B^2)$ to derive Theorem 3.

The average gradient magnitude $\mathbb{E}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2$ depends on the observable O in question, but also on the parametrization $x \mapsto H(x) \mapsto \rho(x)$. This section provides a useful smoothness bound based on physically meaningful assumptions:

(a) *Physical system:* We consider n finite-dimensional quantum many-body systems that are arranged at locations, or sites, in a d-dimensional space, e.g., a spin chain ($d = 1$), a square lattice ($d = 2$), or a cubic lattice ($d = 3$). Unless specified otherwise, our big-$\mathcal{O}, \Omega, \Theta$ notation will be with respect to the thermodynamic limit $n \to \infty$.

(b) *Hamiltonian:* $H(x)$ decomposes into a sum of geometrically local terms $H(x) = \sum_j h_j(x)$, each of which only acts on an $\mathcal{O}(1)$ number of sites in a ball of $\mathcal{O}(1)$ radius. Individual terms $h_j(x)$ obey $\|h_j(x)\|_\infty \le 1$ and also have bounded directional derivative: $\|\partial h_j/\partial \hat{u}\|_\infty \le 1$, where $\hat{u}$ is a unit vector in parameter space. However, each term $h_j(x)$ can depend on the entire input vector $x \in [-1, 1]^m$.

(c) *Ground-state subspace:* We consider "the" ground state $\rho(x)$ for the Hamiltonian $H(x)$ to be defined as $\rho(x) = \lim_{\beta\to\infty} \mathrm{e}^{-\beta H(x)}/\operatorname{tr}(\mathrm{e}^{-\beta H(x)})$. This is equivalent to a uniform mixture over the eigenspace of $H(x)$ with the minimum eigenvalue.

(d) *Observable:* O decomposes into a sum of few-body observables $O = \sum_i O_i$, each of which only acts on an $\mathcal{O}(1)$ number of sites. Each few-body observables O_i can act on geometrically-nonlocal sites.

Assumptions (a)–(c) should be viewed as mild technical assumptions that are often met in practice. The main result of this section bounds the smoothness condition based on an additional requirement.

Lemma 4 (Spectral gap implies smoothness condition)**.** *Consider a class of local Hamiltonians*

$$\{H(x) : \ x \in [-1, 1]^m\} \tag{F49}$$

and an observable $O = \sum_i O_i$ that obey the technical requirements (a)–(c) above. Moreover, suppose that the spectral *gap of each $H(x)$ is lower bounded by (constant) $\gamma > \Omega(1)$. Then,*

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \le c_{\mathrm{all}} \Big(\sum_i \|O_i\|_\infty \Big)^2. \tag{F50}$$

Here, $c_{\mathrm{all}} > 0$ is a constant that depends on spatial dimension d, spectral gap γ, as well as the Lieb-Robinson velocities.

The proof is based on combining two powerful techniques from quantum many body physics. Namely, Lieb-Robinson bounds [115] to exploit locality and the spectral flow formalism [13], also referred to as quasi-adiabatic evolution or continuation [81, 128], to exploit the spectral gap.

a. Quasi-adiabatic continuation for gapped Hamiltonians [13, 81, 128]: Given a quantum system satisfying the above assumptions (a)-(c), it is reasonable to expect that small changes in x only lead to small changes in the associated ground state $\rho(x)$. Spectral flow makes this intuition precise. Let the spectral gap of $H(x)$ be lower bounded by a constant γ over $[-1, 1]^m$. Then, the directional derivative of an associated ground state, in the direction defined by the parameter unit vector $\hat{u}$, obeys

$$\frac{\partial \rho}{\partial \hat{u}}(x) = \mathrm{i}[D_{\hat{u}}(x), \rho(x)] \quad \text{where} \quad D_{\hat{u}}(x) = \int_{-\infty}^{\infty} W_\gamma(t) \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial H}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \mathrm{d}t. \tag{F51}$$

Here, $W_\gamma(t)$ is a fast decaying weight function that obeys $\sup_t |W_\gamma(t)| = 1/2$ and only depends on the spectral gap. More precisely,

$$|W_\gamma(t)| \le \begin{cases} \frac{1}{2} & 0 \le \gamma|t| \le \theta, \\ 35\mathrm{e}^2(\gamma|t|)^4 \mathrm{e}^{-\frac{2}{7}\frac{\gamma|t|}{\log(\gamma|t|)^2}} & \gamma|t| > \theta. \end{cases} \tag{F52}$$

The constant θ is chosen to be the largest real solution of $35\mathrm{e}^2\theta^4 \exp(-\frac{2}{7}\frac{\theta}{\log(\theta)^2}) = 1/2$.

b. Lieb-Robinson bounds for local Hamiltonians/observables [78, 115]: Let $\operatorname{supp}(X)$ denote the sites on which a many-body operator X acts nontrivially. Furthermore, for any two operators X_1, X_2, we define the distance $\Delta(X_1, X_2)$ to be the minimum distance between all pairs of sites acted on by X_1 and X_2, respectively, in the d-dimensional space. We also consider the number of local terms in a ball of radius r. For any operator X acting on a single site, this ball contains $\mathcal{O}(r^d)$ local terms in d-dimensional space,

$$\sum_{j:\Delta(X,h_j)\le r} 1 \le b_d + c_d r^d, \forall r \ge 0, \tag{F53}$$

where we recall the definition that $H = \sum_j h_j$ is a sum of local terms h_j. The bound on the number of local terms in a ball of radius r implies the existence of a Lieb-Robinson bound [25, 78]. It states that for any two operator X_1, X_2 and any $t \in \mathbb{R}$, we have

$$\|[\exp(\mathrm{i}tH(x))X_1 \exp(-\mathrm{i}tH(x)), X_2]\|_\infty \leq c_{\mathrm{lr}} \|X_1\|_\infty \|X_2\|_\infty |\mathrm{supp}(X_1)| \mathrm{e}^{-a_{\mathrm{lr}}(\Delta(X_1,X_2) - v_{\mathrm{lr}}|t|)}, \tag{F54}$$

for some constants $a_{\mathrm{lr}}, c_{\mathrm{lr}}, v_{\mathrm{lr}} = \Theta(1)$.

Apart from these two concepts, we will also need a bound on integrals of certain fast-decaying functions.

Lemma 5 (Lemma 2.5 in [13]). *Fix $a > 0$ and define the function $u_a(x) = \exp(-ax/\log(x)^2)$ on the domain $x \in (1, \infty)$. Then,*

$$\int_t^\infty x^k u_a(x) \mathrm{d}x \leq \frac{2k+3}{a} t^{2k+2} u_a(t) \quad \textit{for all } t > \mathrm{e}^4 \textit{ and } k \in \mathbb{N} \textit{ that obey } \quad 2k+2 \leq \frac{at}{\log(t)^2}. \tag{F55}$$

Proof of Lemma 4. Fix an input $x \in [-1,1]^n$ and a unit vector $\hat{u} \in \mathbb{R}^n$ (direction). We may then rewrite the associated directional derivative of $\rho(x)$ in two ways, namely

$$\frac{\partial \rho}{\partial \hat{u}}(x) = \hat{u} \cdot \nabla_x \rho(x), \quad \text{and} \tag{F56a}$$

$$\frac{\partial \rho}{\partial \hat{u}}(x) = -\mathrm{i}\,[D_{\hat{u}}(x), \rho(x)] \quad \text{with} \quad D_{\hat{u}}(x) = \int_{-\infty}^\infty \mathrm{d}t W_\gamma(t) \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial H}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)}. \tag{F56b}$$

When evaluated on an observable O, this establishes the following correspondence:

$$\hat{u} \cdot \nabla_x \operatorname{tr}(O\rho(x)) = \operatorname{tr}\left(O\,[D_{\hat{u}}(x), \rho(x)]\right) = \operatorname{tr}\left([O, D_{\hat{u}}(x)]\,\rho(x)\right), \tag{F57}$$

for any $\hat{u}$. Choosing $\hat{u} = \hat{u}(x, O) = \frac{\nabla_x \operatorname{tr}(O\rho(x))}{\|\nabla_x \operatorname{tr}(O\rho(x))\|_2}$ implies

$$\|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 = \left|\operatorname{tr}([O, D_{\hat{u}(x,O)}(x)]\rho(x))\right|^2. \tag{F58}$$

The left hand side is the magnitude of steepest slope in a phase for the particular observable O. The average slope over the entire domain $[-1,1]^m$ is thus given as

$$\mathop{\mathbb{E}}_{x \sim [-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 = \frac{1}{2^m} \int_{[-1,1]^m} \left|\operatorname{tr}([O, D_{\hat{u}(x,O)}(x)]\rho(x))\right|^2 \mathrm{d}^m x. \tag{F59}$$

Intuitively, thermodynamic observables should not change too rapidly within a phase. Making this intuition precise will allow us to upper bound the average slope by a constant C.

We first expand $D_{\hat{u}}(x)$ and apply a triangle inequality to obtain

$$|\operatorname{tr}([O, D_{\hat{u}}(x)]\rho(x))| \leq \sum_i \int_{-\infty}^\infty W_\gamma(t) \sum_j \left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty \mathrm{d}t. \tag{F60}$$

For fixed t, we can separate local Hamiltonian terms into two groups, defines using the constants in the Lieb-Robinson bound (F54). The first group contains all terms h_j that obey $\Delta(O_i, h_j) \leq v_{\mathrm{lr}}|t|$. The second group contains all h_j that obey $\Delta(O_i, h_j) > v_{\mathrm{lr}}|t|$ instead. Equation (F53) above provides a useful bound on the size of the first group. It contains at most $|\mathrm{supp}(O_i)|(b_d + c_d(v_{\mathrm{lr}}|t|)^d) \leq c_O(b_d + c_d(v_{\mathrm{lr}}|t|)^d)$ local terms h_j, for some constant $c_O \leq \mathrm{supp}(O)$. We can bound the summation over these terms using $\|[A,B]\|_\infty \leq 2\|A\|_\infty \|B\|_\infty$ to obtain

$$\sum_{j:\Delta(O_i,h_j) \leq v_{\mathrm{lr}}t} \left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty \leq c_O(b_d + c_d(v_{\mathrm{lr}}|t|)^d) \times 2\|O_i\|_\infty \left\| \frac{\partial h_j}{\partial \hat{u}} \right\|_\infty \tag{F61a}$$

$$\leq 2c_O \|O_i\|_\infty (b_d + c_d(v_{\mathrm{lr}}|t|)^d). \tag{F61b}$$

The second inequality follows from technical assumption (b): $\|\partial h_j/\partial \hat{u}\|_\infty \leq 1$.

The contributions from the second group can be controlled via the Lieb-Robinson bound from Eq. (F54). For every h_j that obeys $\Delta(O_i, h_j) > v_{\mathrm{lr}}|t|$, we have

$$\left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty \leq c_{\mathrm{lr}} \|O_i\|_\infty \|\partial h_j/\partial \hat{u}\|_\infty |\mathrm{supp}(h_j)| \mathrm{e}^{-a_{\mathrm{lr}}(\Delta(O_i,h_j) - v_{\mathrm{lr}}|t|)} \tag{F62a}$$

$$\leq c_{\mathrm{lr}} c_h \left\|O_i\right\|_\infty \mathrm{e}^{-a_{\mathrm{lr}}(\Delta(O_i,h_j)-v_{\mathrm{lr}}|t|)}. \tag{F62b}$$

Reusing Eq. (F53), we conclude that there are at most $|\mathrm{supp}(O_i)|(b_d + c_d(v_{\mathrm{lr}}|t| + r + 1)^d)$ local terms h_j with $\Delta(O_i, h_j) \in [v_{\mathrm{lr}}|t| + r, v_{\mathrm{lr}}|t| + r + 1]$. This ensures

$$\sum_{j:\Delta(O_i,h_j)>v_{\mathrm{lr}}|t|} \left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty$$

$$\leq \sum_{r=0}^{\infty} \sum_{j:\Delta(O_i,h_j)\in[v_{\mathrm{lr}}|t|+r, v_{\mathrm{lr}}|t|+r+1]} \left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty \tag{F63a}$$

$$\leq \int_{r=0}^{\infty} \mathrm{d}r c_{\mathrm{lr}} c_h \left\|O_i\right\|_\infty \mathrm{e}^{-a_{\mathrm{lr}} r} \times \mathrm{supp}(O_i)(b_d + c_d(v_{\mathrm{lr}}|t| + r + 1)^d) \tag{F63b}$$

$$\leq c_{\mathrm{lr}} c_h c_O \left\|O_i\right\|_\infty \int_{r=0}^{\infty} \mathrm{d}r \mathrm{e}^{-a_{\mathrm{lr}} r}(b_d + c_d(v_{\mathrm{lr}}|t| + r + 1)^d) \tag{F63c}$$

$$\leq c_{\mathrm{lr}} c_h c_O \left\|O_i\right\|_\infty \left(\frac{b_d}{a_{\mathrm{lr}}} + c_d \sum_{p=0}^{d} \frac{d!}{p! a_{\mathrm{lr}}^{d-p+1}} (v_{\mathrm{lr}}|t| + 1)^p \right). \tag{F63d}$$

We can now combine the two bounds into a single statement:

$$\sum_j \left\| \left[O_i, \mathrm{e}^{\mathrm{i}tH(x)} \frac{\partial h_j}{\partial \hat{u}}(x) \mathrm{e}^{-\mathrm{i}tH(x)} \right] \right\|_\infty \leq \left\|O_i\right\|_\infty \sum_{p=0}^{d} C_p |t|^p. \tag{F64}$$

Here, we have implicitly defined a new set of constants C_p that depend on the constants $c_O, c_h, c_{\mathrm{lr}}, c_d, a_{\mathrm{lr}}, v_{\mathrm{lr}}, d$ that had already featured before. Plugging the above into Eq. (F60) and substituting the spectral flow weight function W (F52) for its absolute value allows us to bound the maximum slope of $\mathrm{tr}(O\rho(x))$ when the Hamiltonian moves from $H(x)$ to $H(x + \mathrm{d}\hat{u})$. Indeed,

$$|\operatorname{tr}([O, D_{\hat{u}}(x)]\rho(x))| \leq \Big(\sum_i \left\|O_i\right\|_\infty \Big) \sum_{p=0}^{d} C_p \int_{-\infty}^{\infty} |W_\gamma(t)||t|^p \mathrm{d}t \ . \tag{F65}$$

To bound the resulting integral, we recall that $W_\gamma(t)$ obeys $\sup_t |W_\gamma(t)| = 1/2$, define $t^* = \max(\mathrm{e}^4, 7(d+5), \theta)/\gamma$, and split up the integration into two parts, $t \in [-t^*, t^*]$ and $t \notin [-t^*, t^*]$. Symmetry then ensures

$$\int_{-\infty}^{\infty} \mathrm{d}t |W_\gamma(t)||t|^p \leq \frac{1}{2} \int_{-t^*}^{t^*} \mathrm{d}t |t|^p + 2 \int_{t^*}^{\infty} \mathrm{d}t \, 35\mathrm{e}^2 (\gamma t)^4 \mathrm{e}^{-\frac{2}{7}\frac{\gamma t}{\log(\gamma t)^2}} t^p \tag{F66a}$$

$$= \int_0^{t^*} \mathrm{d}t \, t^p + 70\mathrm{e}^2 \gamma^{-p-1} \int_{x=\gamma t^*}^{\infty} \mathrm{d}x \, x^{p+4} \mathrm{e}^{-\frac{2}{7}\frac{x}{\log(x)^2}}. \tag{F66b}$$

The first integral is straightforward, and the second integral can be bounded using Lemma 5. Set $a = 2/7$, $k = p + 4$ and note that we have chosen t^* such that all assumptions are valid. Applying Lemma 5 ensures

$$\int_{-\infty}^{\infty} \mathrm{d}t |W_\gamma(t)||t|^p \mathrm{d}t \leq \frac{|t^*|^{p+1}}{p+1} + 70\mathrm{e}^2 \gamma^{-p-1} \frac{2k+3}{a} (\gamma t^*)^{2k+2} \mathrm{e}^{-\frac{2\gamma t^*}{7\log(\gamma t^*)^2}} \tag{F67a}$$

$$= \frac{|t^*|^{p+1}}{p+1} + 35\mathrm{e}^2 \gamma^{-p-1} 7(2p+11)(\gamma t^*)^{2p+10} \mathrm{e}^{-\frac{2\gamma t^*}{7\log(\gamma t^*)^2}}, \tag{F67b}$$

for any integer $0 \leq p \leq d$. Inserting these bounds into the sum (F64) implies

$$|\operatorname{tr}([O, D_{\hat{u}}(x)]\rho(x))| \leq \Big(\sum_i \left\|O_i\right\|_\infty \Big) \sum_{p=0}^{d} C_p \left(\frac{|t^*|^{p+1}}{p+1} + 35\mathrm{e}^2 \gamma^{-p-1} 7(2p+11)(\gamma t^*)^{2p+10} \mathrm{e}^{-\frac{2\gamma t^*}{7\log(\gamma t^*)^2}} \right). \tag{F68}$$

Recall that $t^* = \max(\mathrm{e}^4, 7(d+5), \theta)/\gamma$ is a constant that only depends on d and γ, and the C_p's are also constants that depend on on $c_O, c_h, c_{\mathrm{lr}}, c_d, a_{\mathrm{lr}}, v_{\mathrm{lr}}, d$. We may subsume all of these constant contributions in a new constant c_{all} and conclude

$$|\operatorname{tr}([O, D_{\hat{u}}(x)]\rho(x))| \leq c_{\mathrm{all}} \Big(\sum_i \left\|O_i\right\|_\infty \Big). \tag{F69}$$

Inserting this uniform upper bound into Eq. (F59) completes the proof of Lemma 4. □

APPENDIX G: Sample complexity lower bound for predicting ground states

This section establishes an information-theoretic lower bound for the task of predicting ground state approximations. It highlights that, without further assumptions on the Hamiltonians, the training data size required in Theorem 4 is essentially tight.

Theorem 5. *Fix a prediction error tolerance ϵ, a number m of parameters, as well as constants $C, B > 0$ such that $C/(9\epsilon) \leq m^{0.99}$. Consider a quantum ML model that learns from quantum data $\{x_\ell \to \rho(x_\ell)\}_{\ell=1}^N$ of size N to generate ground state predictions $\hat{\sigma}(x)$, where $x \in [-1,1]^m$. Suppose the quantum ML model can achieve*

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho(x))|^2 \leq \epsilon, \tag{G1}$$

with high probability, for every class of Hamiltonians $H(x)$ and for every observable O given as a sum of local observables $\sum_i O_i$ that obey

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho(x))\|_2^2 \leq C \qquad \textit{(smoothness condition)}, \tag{G2a}$$

$$\sum_i \|O_i\| \leq B \qquad \textit{(bounded norm)}. \tag{G2b}$$

Then, the (quantum) training data size must obey

$$N \geq B^2 m^{\Omega(C/\epsilon)}/\log(B). \tag{G3}$$

This is also a lower bound on quantum computational time associated with the quantum ML model.

The assumption $C/(9\epsilon) \leq m^{0.99}$ is required for technical reasons outlined below. It is equivalent to demanding that the prediction error tolerance is large enough compared to the inverse of m, i.e., $\epsilon \geq C/(9m^{0.99})$. If the quantum ML model can achieve an even smaller prediction error, such that $\mathbb{E}_{x\sim[-1,1]^m} |\operatorname{tr}(O\hat{\sigma}(x)) - \operatorname{tr}(O\rho(x))|^2 < C/(9m^{0.99})$, then we choose $\epsilon = C/(9m^{0.99})$. For such a choice of ϵ, the training data size lower bound becomes $N \geq B^2 m^{\Omega(m^{0.99})}/\log(B)$, which is exponential in $m^{0.99}$. Hence, in all cases, we need ϵ to be a constant for any (quantum or classical) machine learning algorithm to obtain a sample complexity that scales polynomially in m.

We prove Theorem 5 by means of an information-theoretic analysis. Conceptually, it resembles arguments developed in prior work [90] (sample complexity lower bound for general quantum machine learning models). Section G.1 formulates a learning problem that involves predicting ground state properties of a certain class of Hamiltonians. Subsequently, Section G.2 incorporates a hypothetical (quantum ML) solution to this learning problem as a decoding procedure in a communication protocol. Information-theoretic bottlenecks then beget fundamental restrictions on the sample complexity of any ML model that solves the learning problem, see Section G.3.

G.1. Learning problem formulation

We consider a family of single-qubit Hamiltonians, i.e. $n = 1$, that is parametrized by m degrees of freedom. We first map $x \in [-1,1]^m$ to a real number by evaluating a truncated Fourier series f_a. Fix a cutoff $\Lambda = \sqrt{C/(9\epsilon)}$ and let

$$K_\Lambda = \left|\left\{k \in \mathbb{Z}^m : \|k\|_2 \leq \Lambda = \sqrt{C/(9\epsilon)}\right\}\right| \tag{G4}$$

denote the number of n-dimensional wave-vectors with Euclidean norm at most Λ. We equip each of these wave vectors k with a sign $a_k \in \{\pm 1\}$ and define the function

$$f_a(x) = \sqrt{\frac{9\epsilon}{K_\Lambda}} \sum_{k\in\mathbb{Z}^m, \|k\|_2\leq\Lambda} a_k \cos(\pi k \cdot x), \quad \text{where} \quad a \in \{\pm 1\}^{K_\Lambda}, \tag{G5}$$

subsumes all sign choices involved. We use this function to define a single-qubit Hamiltonian. For Pauli matrices X and Z, we set

$$H_a(x) = \exp\left(+\tfrac{\mathrm{i}}{2} \arcsin\left(f_a(x)/B\right) X\right) (-Z) \exp\left(-\tfrac{\mathrm{i}}{2} \arcsin\left(f_a(x)/B\right) X\right), \tag{G6}$$

where B is a constant that will reflect the size of the target observable, see Eq. (G2b). To summarize, each choice of $a \in \{\pm 1\}^{K_\Lambda}$ yields an entire class of single-qubit Hamiltonians $H_a(x)$ that is parametrized by m-dimensional inputs $x \in [-1,1]^m$. These stylized Hamiltonians are simple enough to compute their (nondegenerate) ground state explicitly:

$$\rho_a(x) = |\psi_a(x)\rangle\langle\psi_a(x)| \quad \text{with} \quad |\psi_a(x)\rangle = \begin{pmatrix} \cos\left(\frac{1}{2}\arcsin(f_a(x)/B)\right) \\ \mathrm{i}\sin\left(\frac{1}{2}\arcsin(f_a(x)/B)\right) \end{pmatrix} \in \mathbb{C}^2. \tag{G7}$$

Finally, we fix the single-qubit observable O to be a scaled version of Pauli Y. Setting $O = BY$ yields a 1-local observable. And, more importantly,

$$\begin{aligned} \operatorname{tr}\left(O\rho_a(x)\right) &= B\langle\psi_a|Y|\psi_a\rangle = B\left(-\mathrm{i}\overline{\langle 0|\psi_a(x)\rangle}\langle 1|\psi_a(x)\rangle + \mathrm{i}\langle 0|\psi_a(x)\rangle\overline{\langle 1|\psi_a(x)\rangle}\right) && \text{(G8a)} \\ &= 2B\cos\left(\tfrac{1}{2}\arcsin\left(f_a(x)/B\right)\right)\sin\left(\tfrac{1}{2}\arcsin\left(f_a(x)/B\right)\right) && \text{(G8b)} \\ &= B\sin\left(\arcsin\left(f_a(x)/B\right)\right) = f_a(x). && \text{(G8c)} \end{aligned}$$

By construction, the expectation value $\operatorname{tr}\left(O\rho_a(x)\right)$ exactly reproduces the function $f_a(x)$ defined in Eq. (G5). Being able to accurately predict it will be equivalent to accurately learning this function – regardless of the underlying sign parameter $a \in \{\pm 1\}^{K_\Lambda}$.

To complete the formulation of the learning problem, we recall that the training parameters are sampled from the uniform distribution over the hypercube, $\operatorname{Unif}[-1,1]^m$, and that we will evaluate the expectation $\mathbb{E}$ over x with respect to this distribution from now on. This choice of distribution implies a nice closed-form expression for the average squared distance of two functions f_a, f_b. For $a, b \in \{\pm 1\}^{K_\Lambda}$,

$$\begin{aligned} \mathop{\mathbb{E}}_{x} |f_a(x) - f_b(x)|^2 &= \frac{9\epsilon}{K_\Lambda} \sum_{k,l\in\mathbb{Z}^m, \|k\|_2, \|l\|_2 \le \Lambda} (a_k - b_k)(a_l - b_l) \int_{[-1,1]^m} \cos\left(\pi k\cdot x\right)\cos\left(\pi l\cdot x\right) \mathrm{d}^m x && \text{(G9a)} \\ &= \frac{9\epsilon}{K_\Lambda} \sum_{k\in\mathbb{Z}^m, \|k\|_2 \le \Lambda} (a_k - b_k)^2 && \text{(G9b)} \\ &= \frac{9\epsilon}{K_\Lambda} \sum_{k\in\mathbb{Z}^m, \|k\|_2 \le \Lambda} 4 \times \mathbf{1}\left\{a_k \neq b_k\right\} && \text{(G9c)} \\ &= \frac{36\epsilon}{K_\Lambda} d_H(a,b)\ , && \text{(G9d)} \end{aligned}$$

where we have used orthonormality of the Fourier basis $\cos(\pi k\cdot x)$, and $d_H(a,b) = \sum_k \mathbf{1}\left\{a_k \neq b_k\right\}$ is the *Hamming distance* on $\{\pm 1\}^{K_\Lambda}$.

We conclude this expository section by examining whether the construction fulfills the requirement stated in the theorem and presenting a technical lemma. First of all, we have

$$\|O\| = B\,\|Y\| = B, \tag{G10}$$

which satisfies the bounded norm constraint in Eq. (G2b). Furthermore, we can use the orthonormality of $\cos(\pi k\cdot t)$ to find that

$$\mathop{\mathbb{E}}_{x\sim[-1,1]^m} \|\nabla_x \operatorname{tr}(O\rho_a(x))\|_2^2 = \frac{9\epsilon}{K_\Lambda} \sum_{k\in\mathbb{Z}^m : \|k\|_2 \le \Lambda} \|k\|_2^2\,|a_k|^2 \le \frac{9\epsilon}{K_\Lambda} K_\Lambda \Lambda^2 = C. \tag{G11}$$

Thus the smoothness condition in Eq. (G2a) is also satisfied. Now, we turn our attention to the ground state (G7). The following technical lemma exposes the function $f_a(x)/B$ directly in the amplitudes of ground states.

Lemma 6. *Let $|\psi_a(x)\rangle$ be the ground state of Hamiltonian H_a defined in Eq. (G7). Then,*

$$\rho_a(x) = |\psi_a(x)\rangle\langle\psi_a(x)| = \tfrac{1}{2}\begin{pmatrix} 1 + \sqrt{1-(f_a(x)/B)^2} & -\mathrm{i}f_a(x)/B \\ \mathrm{i}f_a(x)/B & 1 - \sqrt{1-(f_a(x)/B)^2} \end{pmatrix}. \tag{G12}$$

Proof. The proof is based on double-angle and half-angle trigonometric identities. Suppressing x dependence, the first diagonal entry becomes

$$\langle 0|\rho_a|0\rangle = |\langle 0|\psi_a\rangle|^2 = \cos^2\left(\tfrac{1}{2}\arcsin\left(f_a/B\right)\right) = \tfrac{1}{2}\left(1 + \cos\left(\arcsin\left(f_a/B\right)\right)\right) \tag{G13a}$$

$$= \tfrac{1}{2}\left(1+\sqrt{1-\sin^2\left(\arcsin\left(f_a/B\right)\right)}\right) = \tfrac{1}{2}\left(1+\sqrt{1-\left(f_a/B\right)^2}\right), \tag{G13b}$$

and normalization implies that $\langle 1|\rho_a|1\rangle = 1 - \langle 0|\rho_a|0\rangle$. The off-diagonal entries are

$$\overline{\langle 1|\rho_a|0\rangle} = \langle 0|\rho_a|1\rangle = \langle 0|\psi_a\rangle\langle\psi_a|1\rangle = -\mathrm{i}\cos\left(\tfrac{1}{2}\arcsin\left(f_a/B\right)\right)\sin\left(\tfrac{1}{2}\arcsin\left(f_a/B\right)\right) \tag{G14a}$$

$$= -\tfrac{\mathrm{i}}{2}\sin\left(\arcsin\left(f_a/B\right)\right) = -\tfrac{\mathrm{i}}{2}f_a/B\ . \tag{G14b}$$

□

G.2. Communication protocol

Consider the learning problem introduced in the previous section. Suppose that a quantum ML model can use training data $\mathcal{T} = \{x_\ell, \rho_a(x_\ell)\}_{\ell=1}^N$ to learn a function $f^Q(x)$ that (on average) predicts $\mathrm{tr}\left(O\rho_a(x)\right) = f_a(x)$ for a particular unknown $a \in \{\pm 1\}^{K_\Lambda}$, up to some accuracy ϵ,

$$\mathop{\mathbb{E}}_x \left|f^Q(x) - f_a(x)\right|^2 \leq \epsilon\ . \tag{G15}$$

Such a model will not fare as well in estimating the expectation value associated with $b \neq a$, whenever b is sufficiently far away from a. Using the triangle inequality and Eq. (G9),

$$\mathop{\mathbb{E}}_x \left|f^Q(x) - f_b(x)\right|^2 \geq \mathop{\mathbb{E}}_x \left|f_a(x) - f_b(x)\right|^2 - \mathop{\mathbb{E}}_x \left|f^Q(x) - f_a(x)\right|^2 \geq \frac{36\epsilon}{K_\Lambda} d_H(a,b) - \epsilon\ . \tag{G16}$$

The model's accuracy significantly worsens at $d_H(a,b) > K_\Lambda/18$, where we recall $K_\Lambda = |\{k \in \mathbb{Z}^m :\ \|k\|_2 \leq \Lambda\}|$ from Eq. (G4). In other words, a good quantum ML model would allow us to use training data $\mathcal{T}$ in order to recover the underlying parameter $a \in \{\pm 1\}^{K_\Lambda}$ up to resolution $K_\Lambda/18$ in Hamming distance.

We can use this assertion as an effective decoding procedure in a two-way communication protocol involving Alice and Bob. To accommodate imperfect resolution, Alice and Bob agree on a dictionary of sign vectors $\left\{a^{(1)}, \ldots, a^{(M)}\right\} \subset \{\pm 1\}^{K_\Lambda}$ whose pairwise Hamming distance is large enough: $d_H(a_i, a_j) > K_\Lambda/18$ for all $i \neq j$. Let M denote the cardinality of this dictionary. Alice and Bob use this dictionary and the ML procedure to transmit integers up to size M over a quantum channel. Alice samples an integer $j \in \{1, \ldots, M\}$ and sets $a = a^{(j)} \in \{\pm 1\}^{K_\Lambda}$. Subsequently, she uses a to generate (quantum) training data $\mathcal{T} = \{(x_\ell, \rho_a(x_\ell))\}_{\ell=1}^N$ with $x_1, \ldots, x_N \sim \mathrm{Unif}\left[-1,1\right]^m$ which she passes on to Bob. Subsequently, Bob uses $\mathcal{T}$ to train a quantum ML model to predict the underlying function $\mathrm{tr}\left(O\rho_a(x)\right) = f_a(x)$. By checking $\mathbb{E}_x \left|f_{\bar{a}}(x) - f^Q(x)\right|^2 \leq \epsilon$ for every possible dictionary element $\bar{a}$, he will retrieve the correct message with high probability, i.e., $\bar{a} = a$.

This is a protocol that conveys classical information via a quantum dataset. It is subject to fundamental constraints from information theory. These will allow us to deduce a lower bound on the required training data size $N = |\mathcal{T}|$. An important figure of merit in this argument is the cardinality M of the dictionary. That is, the number of different integers that can be communicated. The larger M, the more powerful the communication protocol, and following result, sometimes attributed to Gilbert and Varshamov [67], is a lower bound on how many bits one can "pack" into the space of L-bit strings while maintaining the required distance.

Lemma 7 (Lemma 5.12 in [141]). *There exists a dictionary $\left\{a^{(1)}, \ldots, a^{(M)}\right\} \in \{\pm 1\}^{K_\Lambda}$ of cardinality $M \geq \lfloor \exp\left(K_\Lambda/32\right) \rfloor$ that achieves $d_H\left(a^{(i)}, a^{(j)}\right) \geq K_\Lambda/4$ whenever $i \neq j$.*

G.3. Information-theoretic analysis

Let us now take a closer look at the communication protocol introduced above by bounding the correlation between Alice's original randomly chosen message a and Bob's decoded signal $\bar{a}$. Up to now, we have stablished the following. Per the bound in Lemma 7, the dictionary of available a's can be chosen to be rather large: $M = \lfloor \exp\left(K_\Lambda/32\right) \rfloor$. Moreover, the existence of a good quantum ML procedure, in the sense of Proposition 5, ensures that $\bar{a} = a$ with high probability.

Correlations between Alice's and Bob's variables are quantified by the (classical) mutual information

$$I(a:\bar{a}) \geq \Omega\left(\log(M)\right) = \Omega(K_\Lambda), \tag{G17}$$

which we have bounded from below using Fano's inequality [185]. Our task now is to provide an upper bound on $I(a:\bar{a})$, in terms of N, B and ϵ, in order to relate those parameters to K_Λ and obtain the desired result in Theorem 5.

Since the parameters $x_1, \ldots, x_N$ are sampled independently from a, we have $I(a : x_1, \ldots, x_N) = 0$ and $a|_{x_1,\ldots,x_N} = a$. Therefore, we can upper bound the mutual information as follows,

$$\begin{aligned} I(a:\bar{a}) &\leq I(a:\bar{a}, x_1, \ldots, x_N) && \text{(G18a)} \\ &= I(a:x_1,\ldots,x_N) + I(a:\bar{a}|x_1,\ldots,x_N) && \text{(G18b)} \\ &= I(a:\bar{a}|x_1,\ldots,x_N) && \text{(G18c)} \\ &= \mathop{\mathbb{E}}_{x_1,\ldots,x_N} I(a|_{x_1,\ldots,x_N} : \bar{a}|_{x_1,\ldots,x_N}) && \text{(G18d)} \\ &= \mathop{\mathbb{E}}_{x_1,\ldots,x_N} I(a:\bar{a}|_{x_1,\ldots,x_N}) \ , && \text{(G18e)} \end{aligned}$$

where $Q|_x$ denotes the random variable Q conditioned on the random variable x.

Next, recall that Bob reconstructs the classical $\bar{a}$ by performing quantum operations on the training data $\mathcal{T} = \{(x_\ell, \rho_a(x_\ell)\}_{\ell=1}^N$. For each instance of randomly chosen parameters $x_1, \ldots, x_N \sim \text{Unif}[-1,1]^m$, Bob performs a quantum measurement on the state $\bigotimes_{\ell=1}^N \rho_a(x_\ell)$ and uses the measurement outcomes to reconstruct $\bar{a}$. Bob's procedure is equivalent to performing the quantum ML algorithm that we have been promised in Sec. G.2. Thus we can use Holevo's theorem [85][183, Sec. 11.6.1] to write

$$I(a:\bar{a}|_{x_1,\ldots,x_N}) \leq \chi\left(a : \bigotimes_{\ell=1}^N \rho_a(x_\ell)\Big|_{x_1,\ldots,x_N}\right) \ , \tag{G19}$$

where the Holevo information χ quantifies correlations between a random variable z and a quantum state ρ_z,

$$\chi(z:\rho_z) = S\left(\mathop{\mathbb{E}}_z \rho_z\right) - \mathop{\mathbb{E}}_z S(\rho_z) \ , \tag{G20}$$

and $S(\rho) = -\operatorname{tr}(\rho \log \rho)$ is the von Neumann entropy. In other words, for each instance of parameters, the correlation between a and $\bar{a}$ is bounded by the Holevo information of Bob's ensemble of quantum states.

Next, we use the subadditivity of von Neumann entropy, $S(\mathbb{E}_z \rho_z \otimes \sigma_z) \leq S(\mathbb{E}_z \rho_z) + S(\mathbb{E}_z \sigma_z)$, and the additivity of entropy for independent systems, $S(\rho \otimes \sigma) = S(\rho) + S(\sigma)$, to obtain

$$\chi\left(a : \bigotimes_{\ell=1}^N \rho_a(x_\ell)\Big|_{x_1,\ldots,x_N}\right) \leq \sum_{\ell=1}^N \chi\left(a : \rho_a(x_\ell)\big|_{x_1,\cdots,x_N}\right) \ . \tag{G21}$$

Plugging Eqs. (G19) and (G21) into Eq. (G18) and using the fact that $\rho_a(x_\ell)$ is independent to $x_{\ell'}$ for any $\ell' \neq \ell$, we obtain

$$\begin{aligned} I(a:\bar{a}) &\leq \sum_{\ell=1}^N \mathop{\mathbb{E}}_{x_1,\cdots,x_N} \chi\left(a : \rho_a(x_\ell)\big|_{x_1,\cdots,x_N}\right) && \text{(G22a)} \\ &= \sum_{\ell=1}^N \mathop{\mathbb{E}}_{x_\ell} \chi\left(a : \rho_a(x_\ell)\big|_{x_\ell}\right) && \text{(G22b)} \\ &= N \mathop{\mathbb{E}}_x \chi(a:\rho_a(x)) \ . && \text{(G22c)} \end{aligned}$$

The last equality follows from the fact that each $(x_\ell, \rho_a(x_\ell))$ is generated independently and in an identical fashion for all $\ell = 1, \ldots, N$.

We have thus reduced the problem of bounding the correlations between classical variables a and $\bar{a}$ to that of bounding the Holevo information of the ensemble of states ρ_a — a much simpler problem because ρ_a is a two-by-two matrix. In Lemma 8 at the end of section, we obtain the bound

$$\mathop{\mathbb{E}}_x \chi(a:\rho_a(x)) \leq \frac{9\epsilon}{4B^2} \log\left(\frac{4eB^2}{9\epsilon}\right) \ . \tag{G23}$$

Using this bound, the first claim in Theorem 5 readily follows, provided that we are allowed to choose

$$K_\Lambda = m^{\Omega(C/\epsilon)} \ . \tag{G24}$$

This assumption, combined with Eqs. (G17-G23) ensures that

$$N \frac{9\epsilon}{4B^2} \log\left(\frac{4eB^2}{9\epsilon}\right) \geq \Omega(K_\Lambda) = m^{\Omega(C/\epsilon)} \quad \text{which implies} \quad N \geq \frac{B^2 m^{\Omega(C/\epsilon)}}{\log(B)}. \tag{G25}$$

Because the quantum ML has to process quantum training data of size $N \geq \frac{B^2 m^{\Omega(C/\epsilon)}}{\log(B)}$, the runtime of the quantum ML has to be lower bounded by that amount as well.

Let us now verify the assumption (G24) on the number of Fourier modes K_Λ available for estimating the quantum state. While we have already determined that $K_\Lambda \leq m^{\mathcal{O}(C/\epsilon)}$ in Eq. (F41), here we need a lower bound. We utilize the assumption that $C/(9\epsilon) \leq m^{0.99}$, which implies $\lfloor C/(9\epsilon) \rfloor \leq m^{0.99}$. To establish Eq. (G24), we restrict our attention to binary wavevectors $k \in \{0,1\}^m$, such that the number of ones is exactly equal to $\lfloor C/(9\epsilon) \rfloor$. Clearly, every such wavevector obeys $\|k\|_2 \leq \sqrt{C/(9\epsilon)}$, so the number of such wavevectors lower bounds K_Λ. This observation, along with some combinatorics, yields

$$K_\Lambda \geq \left| \left\{ k \in \{0,1\}^m : \sum_{j=1}^m k_j = \lfloor C/(9\epsilon) \rfloor \right\} \right| \tag{G26a}$$

$$= \binom{m}{\lfloor C/(9\epsilon) \rfloor} \geq \frac{m^{\lfloor C/(9\epsilon) \rfloor}}{(\lfloor C/9\epsilon \rfloor)^{\lfloor C/(9\epsilon) \rfloor}} \tag{G26b}$$

$$= m^{\lfloor C/(9\epsilon) \rfloor - (\lfloor C/9\epsilon \rfloor) \log(\lfloor C/(9\epsilon) \rfloor)/\log(m)} \geq m^{0.01 \lfloor C/(9\epsilon) \rfloor} = m^{\Omega(C/\epsilon)}. \tag{G26c}$$

We now prove the upper bound (G23) on the mutual information. It follows from analyzing the ground state representations provided by Lemma 6.

Lemma 8. *The learning problem from Section G.1 is set up to obey*

$$\mathop{\mathbb{E}}_{x \sim \mathrm{Unif}[-1,1]^m} \chi\left(a : \rho_a(x)\right) \leq \frac{9\epsilon}{4B^2} \log\left(\frac{4\mathrm{e}B^2}{9\epsilon}\right). \tag{G27}$$

Proof. Using the definition (G20) of the Holevo information and the von Neumann entropy,

$$\mathop{\mathbb{E}}_{x \sim \mathrm{Unif}[-1,1]^m} \chi\left(a : \rho_a(x)\right) = \mathop{\mathbb{E}}_x \left[\mathop{\mathbb{E}}_a [\mathrm{tr}(\rho_a(x) \log \rho_a(x))] - \mathrm{tr}\left(\left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \log \left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \right) \right] \tag{G28a}$$

$$= -\mathop{\mathbb{E}}_x \mathrm{tr}\left[\left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \log \left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \right] . \tag{G28b}$$

The second equality follows from the fact that $\rho_a(x)$ is a pure state, so we have $\mathrm{tr}(\rho_a(x) \log \rho_a(x)) = 0$. We also consider $\mathbb{E}_x$ to be $\mathbb{E}_{x \sim \mathrm{Unif}[-1,1]^m}$. Recalling Lemma 6 yields

$$\mathop{\mathbb{E}}_a \rho_a(x) = \frac{1}{2} \mathop{\mathbb{E}}_a \begin{pmatrix} 1 + \sqrt{1 - (f_a(x)/B)^2} & -\mathrm{i} f_a(x)/B \\ \mathrm{i} f_a(x)/B & 1 - \sqrt{1 - (f_a(x)/B)^2} \end{pmatrix}. \tag{G29}$$

The eigenvalues $\lambda_\pm$ of $\mathbb{E}_a \rho_a(x)$, like those of any two-by-two matrix, can be expressed in terms of the trace and determinant. Using the formula for the eigenvalues and evaluating the trace and determinant yield

$$\lambda_\pm = \frac{1}{2} \mathrm{tr}\left[\mathop{\mathbb{E}}_a \rho_a(x) \right] \pm \frac{1}{2} \sqrt{ \left(\mathrm{tr}\left[\mathop{\mathbb{E}}_a \rho_a(x) \right] \right)^2 - 4 \det\left[\mathop{\mathbb{E}}_a \rho_a(x) \right] } \tag{G30a}$$

$$= \frac{1}{2} \pm \frac{1}{2} \sqrt{ \left(\mathop{\mathbb{E}}_a f_a(x) \right)^2 / B^2 + \left(\mathop{\mathbb{E}}_a \sqrt{1 - f_a(x)^2/B^2} \right)^2 } . \tag{G30b}$$

We will use following lower bound for λ_+

$$\lambda_+ \geq \frac{1}{2} + \frac{1}{2} \mathop{\mathbb{E}}_a \sqrt{1 - f_a(x)^2/B^2} \tag{G31a}$$

$$\geq \frac{1}{2} + \frac{1}{2} (1 - \mathop{\mathbb{E}}_a f_a(x)^2/B^2) \tag{G31b}$$

$$= 1 - \frac{1}{2} \mathop{\mathbb{E}}_a f_a(x)^2/B^2 \geq \frac{1}{2}. \tag{G31c}$$

The first inequality follows from dropping the term $(\mathbb{E}_a f_a(x))^2/B^2$. The second inequality follows from the fact that $\sqrt{1-z} \geq 1 - z$ for all $z \in [0,1]$.

We now proceed to bounding the von Neumann entropy of $\mathbb{E}_a \rho_a(x)$,

$$-\mathrm{tr}\left(\left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \log \left(\mathop{\mathbb{E}}_a \rho_a(x) \right) \right) = -\lambda_+ \log \lambda_+ - \lambda_- \log \lambda_- = H(\lambda_+) \tag{G32a}$$

$$\leq H\left(1 - \frac{1}{2} \mathop{\mathbb{E}}_a f_a(x)^2/B^2 \right) \tag{G32b}$$

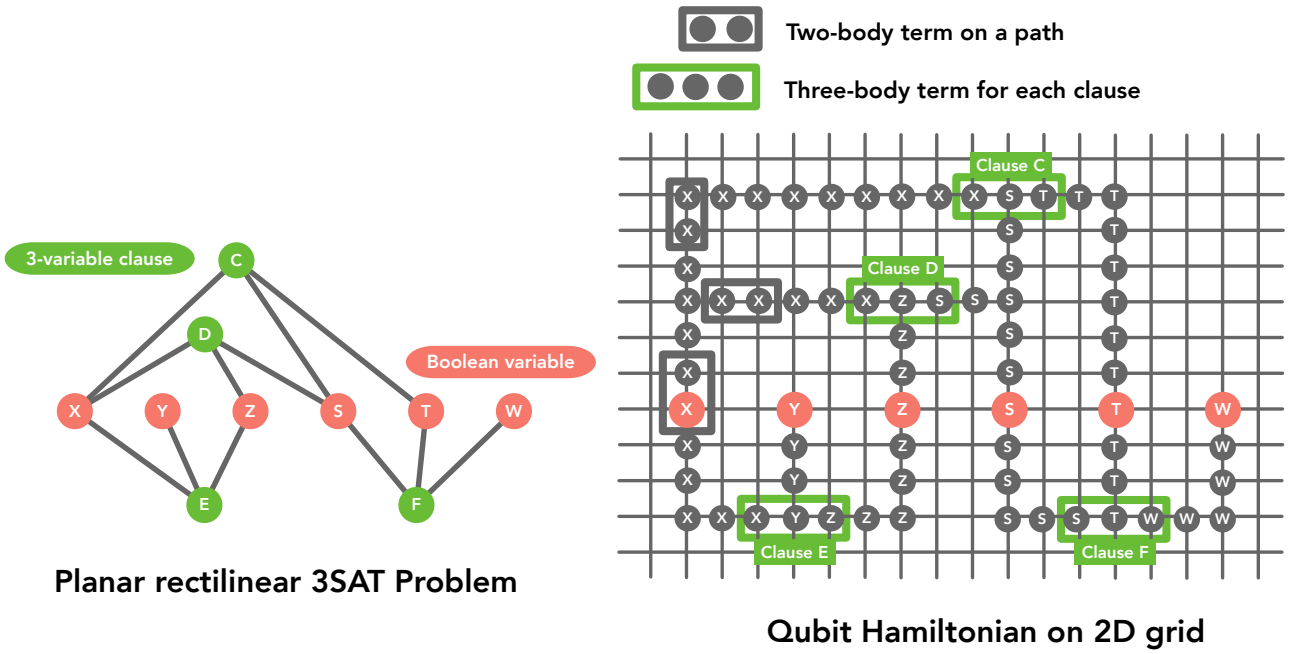

Figure 11: Reduction of planar rectilinear 3SAT (LEFT) to a qubit Hamiltonian on a 2D grid (RIGHT). Each pair (i,j) of nearby grid points on a path (originating from variable X,Y,Z,S,T,W) contains a two-body local term $-Z_iZ_j$ (illustrated by boxes with gray stroke). Each clause (C,D,E,F) corresponds to a three-body local term that imposes the Boolean constraint, e.g., $X \vee Z \vee S$ would correspond to $-\sum_{x,z,s\in\{0,1\}} \mathbb{1}[x \vee z \vee s = 1] \cdot |x\rangle\langle x| \otimes |z\rangle\langle z| \otimes |s\rangle\langle s|$. Every empty grid point (the irrelevant qubits) contain a single body term $-Z_i$.

$$= H\left(\frac{1}{2}\mathop{\mathbb{E}}_a f_a(x)^2/B^2\right) \tag{G32c}$$

$$\equiv H(g(x)) \le g(x)\log(\mathrm{e}/g(x)) \ , \tag{G32d}$$

where $H(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy, and $g(x) = \frac{1}{2}\mathbb{E}_a f_a(x)^2/B^2$. The first inequality follows from the fact that $H(x) \le H(x')$ for all $1/2 \le x' \le x$. Going back to Eq. (G28),

$$\mathop{\mathbb{E}}_x \chi(a:\rho_a(x)|x) = -\mathop{\mathbb{E}}_x \mathrm{tr}\left[\left(\mathop{\mathbb{E}}_a \rho_a(x)\right)\log\left(\mathop{\mathbb{E}}_a \rho_a(x)\right)\right] \tag{G33a}$$

$$\le \mathop{\mathbb{E}}_x [g(x)\log(\mathrm{e}/g(x))] \tag{G33b}$$

$$\le \left(\mathop{\mathbb{E}}_x g(x)\right)\log\left(\frac{\mathrm{e}}{\mathbb{E}_x g(x)}\right) \tag{G33c}$$

$$= \frac{\mathbb{E}_{x,a} f_a(x)^2}{2B^2}\log\left(\frac{2\mathrm{e}B^2}{\mathbb{E}_{x,a} f_a(x)^2}\right). \tag{G33d}$$

The first inequality follows from Eq. (G32). The second inequality follows Jensen's inequality and the fact that $z\log(\mathrm{e}/z)$ is concave for all $z \ge 0$. Orthogonality of the $\cos(\pi k\cdot x)$ terms in f_a (G5) yields

$$\mathop{\mathbb{E}}_{x,a} f_a(x)^2 = \frac{1}{2} \times \frac{9\epsilon}{L} \sum_{k\in\mathbb{Z}^m, \|k\|_2 \le \Lambda} \mathop{\mathbb{E}}_a |a_k|^2 = \frac{9\epsilon}{2}. \tag{G34}$$

Plugging the above into Eq. (G33d), we obtain the advertised bound. □

APPENDIX H: Computational hardness for non-ML algorithms to predict ground state properties

H.1. NP-hardness for estimating one-body observables in the ground state of 2D Hamiltonians

We begin by showing that the task of estimating one-body observables in the ground state of a two-dimensional Hamiltonian with a constant spectral gap is NP-hard.

Proposition 3 (Detailed restatement of Proposition 2; a variant of Lemma 1.4 in [3])**.** *Consider a randomized polynomial-time classical algorithm whose input is the description of a Hamiltonian in any smooth class of qubit Hamiltonians in two dimensional grid with a spectral gap* ≥ 1 *and a unique ground state. The Hamiltonian is denoted by*

$$H = \sum_a h_a, \tag{H1}$$

where h_a *is a three-qubit geometrically-local observable. Let* ρ *be the unique ground state of* H*. Suppose that for each one-body Pauli-Z observable* Z_i*, where* i *enumerates the qubits in the Hamiltonian* H*, the randomized classical algorithm outputs an estimate of* $\mathrm{tr}(Z_i\rho)$ *up to an error* $3/4$ *with probability at least* $2/3$*. Then,* RP = NP*.*

Proof. From standard results in complexity theory [103, 114, 169], it is known that if there is a randomized polynomial-time classical algorithm that can find the solution for any planar rectilinear 3SAT problem with a unique solution with probability at least $1/2$, then RP = NP. (RP, also known as Randomized Polynomial Time, is the class of decision problems such that there is a polynomial-time randomized classical algorithm that outputs YES with probability at least $1/2$ when the correct answer is YES, and outputs NO with probability one when the correct answer is NO. RP is contained in BPP, the class of decision problems that can be solved efficiently by a randomized classical computer.) The planar rectilinear 3SAT problem is a constrained version of 3SAT, where all the Boolean variables $x_1, \ldots, x_n$ are vertices on the x-axis and all the clauses containing three variables are vertices that lie above or below the x-axis. Each clause is connected by an edge to each of the the variables that the clause contains. The graph containing the vertices and the edges form a planar graph; see Figure 11 (left) for an illustration.

We can embed such a planar graph in a two-dimensional grid with a single qubit on each grid point; see Figure 11 for an illustration of the embedding. First, we distinguish between the variable vertices and the clause vertices in the planar graph. Variable vertices lie on the x-axis of the two-dimensional qubit grid, and clause vertices lie above or below the x-axis. Edges of the planar graph become embedded paths on the the 2D grid connecting clause vertices to variable vertices. Because the original graph is planar, we can ensure that the paths corresponding to each edge on the planar graph do not overlap (except when they terminate at the same variable) by choosing a large enough spacing between the variable vertices on the x-axis. For each path on the 2D grid, we add a $-Z_iZ_j$ term to the Hamiltonian for every pair of nearest neighbors along the path. The two body $-Z_iZ_j$ term ensures that in the unique ground state, the qubits on the path must be either all $|0\rangle$'s or all $|1\rangle$'s. Then, for every clause vertex on the planar graph, we add a three-body geometrically-local term (diagonal in the Z-basis) to the Hamiltonian enforcing that in the ground state the endpoints of the three corresponding paths satisfy the Boolean constraint of the corresponding clause. For example, the Boolean clause $X \vee Z \vee S$ would correspond to the three body local term $-\sum_{x,z,s\in\{0,1\}} \mathbb{1}[x \vee z \vee s = 1] \cdot |x\rangle\langle x| \otimes |z\rangle\langle z| \otimes |s\rangle\langle s|$, where $\mathbb{1}[A]$ is 1 if A is true and 0 otherwise. The qubits on paths are called the "relevant" qubits, and the rest of the qubits are called "irrelevant." We add a $-Z_i$ term to the Hamiltonian for all the irrelevant qubits, fixing these qubits to be $|0\rangle$ in the ground state.

Moreover, the eigenstates of the Hamiltonian are computational basis states, because all the local terms are diagonal in the Z-basis. We can also see that there are no terms connecting the relevant and irrelevant qubits, hence the ground state space of the constructed Hamiltonian must be the tensor product of the ground state space for the relevant qubits and the ground state space for the irrelevant qubits. The unique ground state for the irrelevant qubits is the all-zero state $|0\rangle \otimes \cdots \otimes |0\rangle$ due to the $-Z_i$ term. Because the original planar rectilinear 3SAT has an unique solution, the ground state for the relevant qubits is also unique. In this ground state, all variable vertices are fixed at the values that solve the 3SAT problem.

After this reduction from planar rectilinear 3SAT problem to a qubit Hamiltonian H in a two-dimensional consisting of three-body geometrically-local terms, we can apply the randomized classical algorithm to provide an estimate of all the expectation values of Pauli-Z observables in the ground state ρ of H. For each Pauli-Z observable, we repeat the randomized classical algorithm m times, and take the majority vote of the sign of the m estimates for $\mathrm{tr}(Z_i\rho)$. Because $\mathrm{tr}(Z_i\rho)$ is either $+1$ or -1 (since the ground state ρ is a computational basis state), the sign of each estimate from the randomized classical algorithm will be equal to $\mathrm{tr}(Z_i\rho)$ with probability at least $2/3$. Hence, by choosing $m = \Theta(\log(n))$, the majority vote of the sign of the m estimates for $\mathrm{tr}(Z_i\rho)$ will be equal to $\mathrm{tr}(Z_i\rho)$ with probability at least $1 - \frac{1}{2n}$, where n is the total number of qubits in the 2D grid. Using union bound, with probability at least $1/2$, we can obtain $\mathrm{tr}(Z_i\rho)$, for all i. This means that our randomized classical algorithm can find the unique solution for the planar rectilinear 3SAT problem with probability at least $1/2$. Therefore, RP=NP if such an algorithm exists. □

We conclude this section by emphasizing that the Hamiltonian constructed in above proposition is a classical Hamiltonian; all the local terms in the Hamiltonian are diagonal in the computational basis.

H.2. Computational hardness for a class of Hamiltonians based on factoring

Proposition 3 shows that an NP-hard problem could be solved by performing single-qubit measurements on a modest number of copies of the ground state of a two-dimensional local Hamiltonian, and then performing an efficient classical computation with the measurement outcomes as input. We may therefore conclude that, in hard instances, the preparation of the ground state is itself an NP-hard task. Because we do not expect any NP-hard task to be performed efficiently in the physics lab, or in any other physically realizable process, Proposition 3 does not usefully characterize the computational power of data under realistic conditions.

In contrast, it is reasonable to expect that simple measurements performed on quantum states that are efficiently prepared by quantum computers, combined with classical processing, suffice for solving computational problems that are beyond the reach of classical processing alone. Indeed, proposals for using variation quantum eigensolvers to study quantum chemistry and materials [117, 134] are motivated by this expectation. Theorem 1 is of potential practical interest for a class of Hamiltonians $\{H(x)\}$ such that the ground state of $H(x)$ can be prepared efficiently by a feasible quantum process, yet cannot be efficiently prepared classically.

The rest of this subsection outlines a stylized example that illustrates this idea. Leveraging the efficient quantum algorithm for factoring large numbers, and the assumption that factoring is classically hard, we construct a smooth class of local Hamiltonians whose ground states are easy to prepare quantumly, such that expectation values of one-local observables can be learned efficiently from training data, yet are hard to learn by any classical procedure without access to data.

The first step is to construct two-dimensional Hamiltonians such that computing expectation values of one-local observables in the ground state is equivalent to solving a factoring problem. This can be done by noting a series of well-known facts in complexity theory.

1. The following task is expected to be hard for classical computers. Given a n-bit number R guaranteed to be a product of two prime numbers $p < q$, find p, q. When R is large, all known classical algorithms scale superpolynomially with n. Solving this problem suffices to break the RSA encryption [142].

2. We can represent p, q using at most $2n$ binary variables (bits), and we can write down a propositional formula for these $2n$ variables, which corresponds to a logical circuit that computes the multiplication of p, q and checks if the product equals R. The propositional formula can be written without any additional Boolean variable. This yields a SAT problem with $2n$ Boolean variables whose unique solution is equal to the two prime numbers p, q.

3. A SAT problem with a unique solution can be efficiently mapped to a 3SAT problem with a unique solution; see [106].

4. A 3SAT problem with a unique solution can be efficiently mapped to a planar rectilinear 3SAT problem with a unique solution; see [103, 114].

5. A planar rectilinear 3SAT problem with a unique solution can be efficiently mapped to a two-dimensional 3-local Hamiltonian with a spectral gap of one and a unique ground state, such that estimating one-local observables in the ground state of the Hamiltonian to a constant error with a constant probability is sufficient to find the unique solution for the planar rectilinear 3SAT problem; see the proof of Proposition 3.

We now focus on any smooth class of two-dimensional Hamiltonians $H^{\mathrm{RSA}}(x)$ with a constant spectral gap such that there exists $x^{\mathrm{RSA}} \in [-1, 1]^m$ such that $H^{\mathrm{RSA}}(x^{\mathrm{RSA}})$ can be written as a two-dimensional Hamiltonian that is mapped from a factoring problem. We refer to such a class of Hamiltonians as an RSA-based two-dimensional gapped Hamiltonian class.

For any RSA-based Hamiltonian class H^{RSA}, we can efficiently obtain the training data from a quantum experiment. We first prepare the ground state for $H^{\mathrm{RSA}}(x^{\mathrm{RSA}})$ by applying Shor's algorithm. Then we can adiabatically evolve the ground state for $H^{\mathrm{RSA}}(x^{\mathrm{RSA}})$ to obtain the ground state for $H^{\mathrm{RSA}}(x), \forall x \in [-1, 1]$ due to the existence of a constant spectral gap [6, 177]. Hence, according to Theorem 1, for any RSA-based Hamiltonian class, a classical ML algorithm trained from data obtained in quantum experiments can predict efficiently expectation values of one-local observables in the ground state. In contrast, a classical algorithm that does not learn from training data is unable to efficiently estimate 1-body observables in the ground state, assuming that RSA encryption can not be broken by classical computers.

APPENDIX I: No observable can classify topological phases

Recall that ground states of two Hamiltonians are in the same topological phase if there exists a constant-depth geometrically-local quantum circuit that can transform one state to another [187]. The goal of this section is to establish the following proposition.

Proposition 4. *Consider two distinct topological phases A and B (one of the phases could be the trivial phase). No observable O exists such that*

$$\mathrm{tr}(O\rho) > 0, \forall \rho \in \textit{phase } A, \qquad \mathrm{tr}(O\rho) \leq 0, \forall \rho \in \textit{phase } B. \tag{I1}$$

Proof. We consider depth-1 quantum circuits consisting of single-qubit unitaries $U_1, \ldots, U_n$. We let $|\psi_A\rangle, |\psi_B\rangle$ be the signature quantum state for phase A and B. Suppose there is an observable such that

$$\mathrm{tr}(O\rho) > 0, \forall \rho \in \text{phase } A, \qquad \mathrm{tr}(O\rho) \leq 0, \forall \rho \in \text{phase } B. \tag{I2}$$

Then, by definition, we have

$$\langle\psi_A| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_A\rangle > 0, \forall U_1, \ldots, U_n \in U(2), \tag{I3a}$$
$$\langle\psi_B| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_B\rangle \leq 0, \forall U_1, \ldots, U_n \in U(2), \tag{I3b}$$

However, from Lemma 9, no such observable exists. Hence no observable exists that can be used to classify two topologically ordered phases. □

The key lemma utilized in the above proof is the following.

Lemma 9. *For any two n-qubit states $|\psi_1\rangle, |\psi_2\rangle$, no observable O exists such that*

$$\langle\psi_1| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_1\rangle > 0, \forall U_1, \ldots, U_n \in U(2), \tag{I4a}$$
$$\langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle \leq 0, \forall U_1, \ldots, U_n \in U(2), \tag{I4b}$$

where $U(2)$ is the unitary group of 2×2 unitary matrices.

Proof. We will prove this result by contradiction. Assume the existence of an observable O such that Eq. (I4a) and (I4b) both hold. Consider $U_1, \ldots, U_n$ to be independent random matrices that follows the Haar measure on the unitary group $U(2)$. Then using the identity for the first order moment of Haar integration,

$$\mathop{\mathbb{E}}_{U \sim \mathrm{Haar}(U(d))} UXU^\dagger = \mathrm{tr}(X)\frac{\mathbb{I}}{d}, \tag{I5}$$

we can obtain the following identity,

$$\mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \left[(U_1 \otimes \ldots \otimes U_n)|\psi_1\rangle\langle\psi_1|(U_1^\dagger \otimes \ldots \otimes U_n^\dagger)\right] = \mathrm{tr}(|\psi_1\rangle\langle\psi_1|)\frac{\mathbb{I}}{2^n} = \frac{\mathbb{I}}{2^n}. \tag{I6}$$

The key property is the compactness of the unitary group $U(2)$. Consider the following infimum,

$$o_1 = \inf_{U_1,\ldots,U_n \in U(2)} \langle\psi_1| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_1\rangle . \tag{I7}$$

Because the infimum is always attained by an element in the compact set, $\exists U_1^{\inf}, \ldots, U_n^{\inf} \in U(2)$ such that

$$o_1 = \langle\psi_1| ((U_1^{\inf})^\dagger \otimes \ldots \otimes (U_n^{\inf})^\dagger) O (U_1^{\inf} \otimes \ldots \otimes U_n^{\inf}) |\psi_1\rangle . \tag{I8}$$

Therefore, we have $o_1 > 0$ from Eq. (I4a). Using the property of infimum, we have

$$\langle\psi_1| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_1\rangle \geq o_1, \forall U_1, \ldots, U_n \in U(2), \tag{I9}$$

we have the following inequality,

$$\mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \langle\psi_1| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_1\rangle \geq o_1 > 0. \tag{I10}$$

By the linearity of expectation and Eq. (I6), we have

$$\begin{aligned} &\mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \langle\psi_1| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_1\rangle \\ &= \mathrm{tr}\left(O \mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \left[(U_1 \otimes \ldots \otimes U_n)|\psi_1\rangle\langle\psi_1|(U_1^\dagger \otimes \ldots \otimes U_n^\dagger)\right]\right) = \frac{\mathrm{tr}(O)}{2^n}. \end{aligned} \tag{I11}$$

Together, we have

$$\frac{\mathrm{tr}(O)}{2^n} \geq o_1 > 0. \tag{I12}$$

The argument for $|\psi_2\rangle$ is slightly simpler. Consider the following supremum,

$$o_2 = \sup_{U_1,\ldots,U_n \in U(2)} \langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle . \tag{I13}$$

From Eq. (I4b), we have $o_2 \leq 0$. Using the fact that

$$\langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle \leq o_2, \forall U_1, \ldots, U_n \in U(2), \tag{I14}$$

we have the following inequality,

$$\mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle \leq o_2 \leq 0. \tag{I15}$$

By the linearity of expectation and Eq. (I6), we have

$$\begin{aligned} &\mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) O (U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle \\ &= \mathrm{tr}\left(O \mathop{\mathbb{E}}_{U_1,\ldots,U_n \sim \mathrm{Haar}(U(2))} \left[(U_1 \otimes \ldots \otimes U_n) |\psi_2\rangle\langle\psi_2| (U_1^\dagger \otimes \ldots \otimes U_n^\dagger) \right] \right) = \frac{\mathrm{tr}(O)}{2^n}. \end{aligned} \tag{I16}$$

Together, we have

$$\frac{\mathrm{tr}(O)}{2^n} \leq o_2 \leq 0. \tag{I17}$$

From Eq. (I12) and (I17), we have derived the following result

$$\frac{\mathrm{tr}(O)}{2^n} \leq o_2 \leq 0 < o_1 \leq \frac{\mathrm{tr}(O)}{2^n}, \tag{I18}$$

which is a contradiction. Therefore, no such observable O exists. □

APPENDIX J: Proof of efficiency for classifying phases of matter

This section contains a detailed proof for another one of our main contributions. Namely, a rigorous performance guarantee for learning to predict quantum phases of matter.

J.1. Training support vector machines

Let us start by reviewing the textbook framework for reasoning about supervised learning tasks: support vector machines (SVMs). The underlying idea is simple and intuitive. Suppose that we have N data points $\mathbf{x}_\ell \in \mathbb{R}^D$ with binary labels $y_\ell \in \{\pm 1\}$ that form two well separated clusters. Then, we may try to separate these training clusters with a linear hyperplane $\mathsf{H}_{\boldsymbol{\alpha}} = \{\mathbf{x} \in \mathbb{R}^D : \langle \boldsymbol{\alpha}, \mathbf{x}\rangle = 0\} \subset \mathbb{R}^D$, defined using any vector $\boldsymbol{\alpha}$ that is perpendicular to all vectors in the hyperplane. Here, we implicitly assume that the hyperplane $\mathsf{H}_{\boldsymbol{\alpha}}$ must contain the origin $\mathbf{0} \in \mathbb{R}^D$. This simplifies exposition and will suffice for our purposes, but also constitutes an actual restriction (linear SVMs typically also allow for affine shifts). Such a hyperplane divides $\mathbb{R}^D$ up into two half-spaces. For linear classification, we want that these half-spaces perfectly capture the labels of training data: $\langle \boldsymbol{\alpha}, \mathbf{x}_\ell\rangle > 0$ whenever $y_\ell = +1$ and $\langle \boldsymbol{\alpha}, \mathbf{x}_\ell\rangle < 0$ whenever $y_\ell = -1$. The hope is that this simple linear classification strategy generalizes to data we haven't yet seen. When we get a new data point, we simply check which halfspace it belongs to and assign the label accordingly. In the training stage, the main question is: how do we find a suitable hyperplane? Several strategies are known in the literature. One of them is the *soft margin* problem:

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^D}{\text{minimize}} \quad \sum_{\ell=1}^{N} \max\{0, 1 - y_\ell \langle \boldsymbol{\alpha}, \mathbf{x}_\ell\rangle\} \tag{J1a}$$

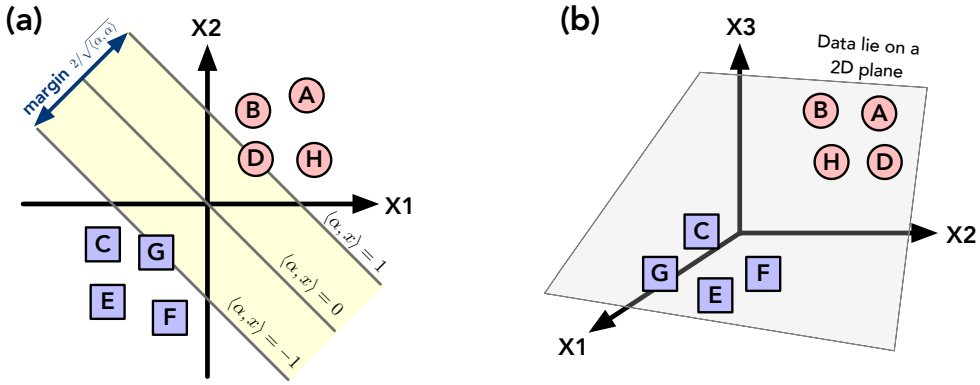

Figure 12: (a) Geometric intuition behind support vector machines (SVMs). The idea is to separate clusters of labeled data with a linear hyperplane. The separation margin (yellow) is inversely proportional to the length $\sqrt{\langle\alpha,\alpha\rangle}$ of the hyperplane normal vector. During the training stage we try to find a hyperplane that separates points with label $+1$ (blue) from points with label -1 (red) such that the margin is as large as possible (left). This hyperplane separates the data space into two halfspaces. In order to predict the label of a new data point, we simply check which halfspace it belongs to. (b) Geometric intuition behind the representer theorem. When trying to find a separating hyperplane, the total dimension of the data space does not matter. We can without loss restrict our attention to the smallest subspace that contains all the data points. This is because orthogonal directions don't matter during training and has two implications: (i) the cost of finding a separating hyperplane depends on the training data size N, not feature space dimension and (ii) we can express the hyperplane vector as a linear combination of training data points.

$$\text{subject to} \quad \langle\boldsymbol{\alpha},\boldsymbol{\alpha}\rangle \leq \Lambda^2. \tag{J1b}$$

For both label values, a positive product $y_\ell\langle\boldsymbol{\alpha},\mathbf{x}_\ell\rangle$ is theoretically sufficient. However, numerical precision considerations warrant a nonzero separation between the clusters, so the product is optimized to be at least as large as a positive number (here, 1). Otherwise, a hyperplane defined by $\boldsymbol{\alpha}$ does not perfectly classify the data, yielding the training error $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}) = \sum_{\ell=1}^{N}\max\{0, 1 - y_\ell\langle\boldsymbol{\alpha},\mathbf{x}_\ell\rangle\}$. The task is to find $\boldsymbol{\alpha}_\sharp$ that achieves the smallest training error: $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) \leq \mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha})$ for all vectors that obey $\langle\boldsymbol{\alpha},\boldsymbol{\alpha}\rangle \leq \Lambda^2$. This is a convex optimization problem that can be solved in polynomial time and we refer to Figure 12 for a visual illustration. The most interesting situation occurs if we manage to achieve an optimal objective value of 0. This corresponds to zero training error. In this case, we have found a hyperplane $\mathsf{H}_{\boldsymbol{\alpha}_\sharp}$ that perfectly separates training data. What is more, the constraint $\langle\boldsymbol{\alpha}_\sharp,\boldsymbol{\alpha}_\sharp\rangle \leq \Lambda^2$ ensures that the margin of separation is strictly positive. Let $\hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}/\|\boldsymbol{\alpha}\|$ be the unit vector that characterizes a hyperplane. Then, zero training error implies $\langle\hat{\boldsymbol{\alpha}},\mathbf{x}_\ell\rangle \geq 1/\|\boldsymbol{\alpha}\| \geq 1/\Lambda$ for all $\mathbf{x}_\ell$ with $y_\ell = +1$ and $\langle\hat{\boldsymbol{\alpha}},\mathbf{x}_\ell\rangle < -1/\Lambda$ for all $\mathbf{x}_\ell$ with $y_\ell = -1$. In turn, the minimal margin amounts to $2/\Lambda$.

However, it should not come as a surprise that such linear classification strategies are often inadequate. Most labeled collections of data simply cannot be separated by a linear hyperplane. However, it has been observed that this drawback can be overcome by first transforming data into a (usually much larger) feature space $\mathbf{x}_\ell \mapsto \phi(\mathbf{x}_\ell)$ and trying to find a separating hyperplane there. This transformation is typically nonlinear and increases the expressiveness of hyperplane classification. Although the separating hyperplane is linear in feature space, it may be highly nonlinear in the original data space. Denote the feature space by $\mathcal{F}$ and suppose that it comes with an inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$ and dual space $\mathcal{F}^*$. We can then formally phrase the search for a linear classifier in feature space as

$$\underset{\boldsymbol{\alpha}\in\mathcal{F}^*}{\text{minimize}} \quad \sum_{\ell=1}^{N}\max\{0, 1 - y_\ell\langle\boldsymbol{\alpha},\phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}}\} \tag{J2a}$$

$$\text{subject to} \quad \langle\boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2. \tag{J2b}$$

This problem looks more daunting than its linear counterpart, especially because the feature space $\mathcal{F}$ may have an exceedingly large – perhaps even infinite – dimension. But we are still interested in identifying a hyperplane that separates a total of N transformed data points $\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_N) \in \mathcal{F}$ in a linear fashion: $\langle\boldsymbol{\alpha},\phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}} > 0$ if $y_\ell = +1$ and $\langle\boldsymbol{\alpha},\phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}} < 0$ else if $y_\ell = -1$. And in order to achieve this, we can without loss restrict ourselves to the N-dimensional subspace $\operatorname{span}\{\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_N)\} \subset \mathcal{F}$ that is spanned by the data points themselves (all other directions are orthogonal to *all* data points and do not play a role for classification). For finite dimensional feature spaces $(\mathcal{F},\langle\cdot,\cdot\rangle_{\mathcal{F}})$, this is an intuitive observation that follows from basic orthogonality arguments. We refer to Figure 12 for a visual illustration. For infinite-dimensional feature spaces it is the content of the celebrated generalized representer theorem [149]. More formally, this

insight allows us to decompose every (relevant) hyperplane normal vector $\boldsymbol{\alpha}$ in the optimization problem (J2a) as $\boldsymbol{\alpha} = \sum_{\ell=1}^{N} \alpha_\ell \phi(\mathbf{x}_\ell)$. Linearity then ensures $\langle \boldsymbol{\alpha}, \phi(\mathbf{x}_{\ell'}) \rangle_{\mathcal{F}} = \sum_{\ell=1}^{N} \alpha_\ell \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_{\ell'}) \rangle_{\mathcal{F}}$ for each $\ell' \in \{1, \dots, N\}$ and also $\langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle_{\mathcal{F}} = \sum_{\ell,\ell'=1}^{N} \alpha_\ell \alpha_{\ell'} \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_{\ell'}) \rangle_{\mathcal{F}}$. These expressions only depend on the elements of a $N \times N$ Gram matrix in feature space:

$$[\mathbf{K}]_{\ell\ell'} = \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_{\ell'}) \rangle_{\mathcal{F}} =: k(\mathbf{x}_\ell, \mathbf{x}_{\ell'}) \quad \text{for } \ell, \ell' \in \{1, \dots, N\}. \tag{J3}$$

The expression $k(\mathbf{x}_\ell, \mathbf{x}_{\ell'})$ is called the *kernel* associated with the feature map ϕ and the matrix $\mathbf{K}$ is the *kernel matrix*. Kernels are a measure of similarity between (training) data points that is often easier to compute than performing the underlying feature map $\phi : \mathbb{R}^D \to \mathcal{F}$. But, for linear classification (in feature space), both contain exactly the same amount of information. Indeed, we may re-express the optimization problem (J2a) as

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^N}{\text{minimize}} \quad \sum_{\ell=1}^{N} \max\left\{0, 1 - y_\ell \boldsymbol{\alpha}^T \mathbf{K} \mathbf{e}_\ell\right\} \tag{J4a}$$

$$\text{subject to} \quad \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \leq \Lambda^2. \tag{J4b}$$

We can also collect the classification labels in a diagonal matrix $\mathbf{Y} = \operatorname{diag}(y_1, \dots, y_N)$ of compatible dimension and linearize the loss function by means of an entry-wise nonnegative slack variable $\boldsymbol{\beta} \geq \mathbf{0}$. Let $\mathbf{1} = (1, \dots, 1)^T$ denote the vector of ones. Then, problem (J4a) is equivalent to solving

$$\underset{\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^N}{\text{minimize}} \quad \langle \mathbf{1}, \boldsymbol{\beta} \rangle \tag{J5a}$$

$$\text{subject to} \quad \boldsymbol{\beta} \geq \mathbf{1} - \mathbf{Y}\mathbf{K}\boldsymbol{\alpha} \tag{J5b}$$

$$\boldsymbol{\beta} \geq \mathbf{0}, \ \boldsymbol{\alpha}^* \mathbf{K} \boldsymbol{\alpha} \leq \Lambda^2. \tag{J5c}$$

Similar to before, the optimal function value denotes the minimal *training error* $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) = \langle \mathbf{1}, \boldsymbol{\beta}_\sharp \rangle$. Apart from a single quadratic constraint ($\boldsymbol{\alpha}^* \mathbf{K} \boldsymbol{\alpha} \leq \Lambda^2$), this optimization problem looks like a linear program in $2N$ dimensions. It is a convex instance of a quadratically constrained quadratic program (QCQP) and can be solved in time at most polynomial in the training data size N [22]. In practice, one could use existing software packages, such as scikit-learn [132] or LIBSVM [35]. If the time to compute the kernel function $k(\mathbf{x}_\ell, \mathbf{x}_{\ell'})$ is t_{kernel}, then the time complexity for training a support vector machine is given by

$$\mathcal{O}(t_{\mathrm{kernel}} N^2 + \mathrm{poly}(N)) \qquad \text{(training time)}. \tag{J6}$$

Hence, for support vector machines with efficiently computable kernel functions $k(\mathbf{x}_\ell, \mathbf{x}_{\ell'})$, small training data sizes N directly ensure a short training time. The polynomial scaling in training data size depends on the type of algorithm. Dedicated solvers for the soft margin problem [35, 82, 94] require (at most) $\mathcal{O}\left(N^3 + \Lambda^2 N / \epsilon^2\right)$ arithmetic operations to produce a solution $\boldsymbol{\alpha}_{\sharp,\epsilon}$ that is ϵ-close to optimal: $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_{\sharp,\epsilon}) \leq \mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) + \epsilon$. For the concrete training problems considered here, such an approximation is good enough and the associated runtime bound simplifies to $\mathcal{O}\left(t_{\mathrm{kernel}} N^2 + N^3\right)$. Interior point methods offer an alternative that scale worse in training data size, but much better in the approximation error ϵ, see e.g. [22].

J.2. Prediction using support vector machines

In the last section, we have explained how feature maps and kernels can considerably boost the expressiveness of initially linear classifiers. We have also explained how to use labeled training data of size N to find a separating hyperplane in feature space by solving a quadratic program (J5a) that depends on the kernel matrix (J3). Ideally, $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) = 0$ (zero training error) and the optimal solution $\boldsymbol{\alpha}_\sharp \in \mathbb{R}^N$ parametrizes a separating hyperplane with minimal margin $2/\Lambda$ in feature space:

$$h_\sharp(\mathbf{x}_{\ell'}) = \sum_{\ell=1}^{N} [\boldsymbol{\alpha}_\sharp]_\ell \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_{\ell'}) \rangle_{\mathcal{F}} = \sum_{\ell=1}^{N} [\boldsymbol{\alpha}_\sharp]_\ell \, k(\mathbf{x}_\ell, \mathbf{x}_{\ell'}) \quad \begin{cases} > +1/\Lambda & \text{if } y_{\ell'} = +1, \\ < -1/\Lambda & \text{else if } y_{\ell'} = -1, \end{cases} \tag{J7}$$

for all (labeled) training data points $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$.The sign of this classifier, in turn, correctly reproduces training labels:

$$y_\sharp(\mathbf{x}_{\ell'}) := \operatorname{sign}(h_\sharp(\mathbf{x}_{\ell'})) = y_{\ell'} \quad \text{for each } \ell \in \{1, \dots, N\}. \tag{J8}$$

In the prediction stage, we use this function to assign a label $y_\sharp(\mathbf{x}) \in \{\pm 1\}$ to a new (and unlabeled) data point $\mathbf{x}$. The cost of evaluating $y_\sharp(\mathbf{x}_{\ell'})$ is dominated by the cost of evaluating N kernel functions. If the time to compute the kernel function is t_{kernel}, then the prediction time for a new input vector $\mathbf{x}$ is bounded by

$$\mathcal{O}(t_{\text{kernel}} N) \qquad \text{(prediction time)}. \tag{J9}$$

Similar to the training time (J6), a small training data size N translates into a fast prediction time.

The hope is that training with an adequate kernel uncovers latent structure that generalizes beyond training data. Typically, larger training data sizes N also increase the chance for learning good classifiers (J8). But generalization beyond training data often only makes sense if the new data point $\mathbf{x}$ is somewhat related to the training data (e.g. training a SVM on labeled cat-vs-dog images does not necessarily produce a classifier that can distinguish apples from oranges). Extra assumptions that address similarity of training and prediction data are important when one aims at establishing rigorous bounds on the probability of making a wrong prediction, i.e. $y_\sharp(\mathbf{x}) = -y(\mathbf{x})$. A common assumption is that both the training data and new data points are generated independently from the same distribution: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N), (\mathbf{x}, y) \sim \mathcal{D}$. The data distribution $\mathcal{D}$ is a joint distribution over both the input vector $\mathbf{x}$ and the label y. Such an assumption encompasses the intuition that the label y is correlated with the input vector $\mathbf{x}$, but is not necessarily a function of $\mathbf{x}$. Flexibility of this form is useful for describing situations where the data points $\mathbf{x}$ are corrupted by noise. This is often the case in quantum mechanics due to the inherent randomness in quantum measurements. The underlying data distribution should be taken into account when reasoning about false predictions, motivating the probability

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[y_\sharp(\mathbf{x}) \neq y\right] \in [0,1] \qquad \text{(average-case prediction error)} \tag{J10}$$

as a good figure of merit. Noting that there are in general many approaches to bounding the prediction error, we present a user-friendly theorem that bounds the average-case prediction error in terms of the training error $\mathrm{E}_{\text{tr}}(\boldsymbol{\alpha}_\sharp)$ and training data size N.

Theorem 6 (Prediction error for support vector machines)**.** *Fix a data distribution $(\mathbf{x}, y) \sim \mathcal{D}$, a kernel function $k(\cdot,\cdot)$, a minimal margin $2/\Lambda$ and a training data size N. Assume $k(\mathbf{x},\mathbf{x}) \leq R^2$ almost surely. Then, with probability (at least) $1-\delta$,*

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[y_\sharp(\mathbf{x}) \neq y\right] \leq \frac{1}{N}\mathrm{E}_{\text{tr}}(\boldsymbol{\alpha}_\sharp) + 7(\Lambda R + 1)\sqrt{\frac{\log(2/\delta)}{N}}, \tag{J11}$$

where $y_\sharp(\mathbf{x})$ is the classifier (J8) obtained from solving the training problem (J5a) on independently sampled training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \sim \mathcal{D}$, and E_{tr} denotes the associated training error.

This rigorous statement bounds the average prediction error in terms of the training error plus an error term that decays as $1/\sqrt{N}$ in training data size. The core assumption is that training and prediction data is sampled in an independent and identically distributed (*iid*) fashion. The proof is based on specializing a standard result from high dimensional probability theory to the task at hand.

Theorem 7 (Theorem 3.3 in [120])**.** *Fix a probability distribution $\mathcal{D}$ over elements in a set X, a family of functions $\mathcal{G}$ from X to the interval $[0, \gamma_{\max}]$, as well as $\delta \in (0,1)$ and $N \in \mathbb{N}$. Then, with probability $1-\delta$, the following bound is valid for* all *functions $g \in \mathcal{G}$ simultaneously:*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[g(x)] \leq \frac{1}{N}\sum_{\ell=1}^{N} g(x_\ell) + 3\gamma_{\max}\sqrt{\frac{\log(2/\delta)}{2N}} + \frac{2}{\sqrt{N}}\mathop{\mathbb{E}}_{\varepsilon_1,\ldots,\varepsilon_N}\left[\sup_{g\in\mathcal{G}}\frac{1}{\sqrt{N}}\sum_{\ell=1}^{N}\varepsilon_\ell g(x_\ell)\right]. \tag{J12}$$

Here, $x_1, \ldots, x_N \overset{iid}{\sim} \mathcal{D}$ are sampled from X and $\varepsilon_1, \ldots, \varepsilon_N \overset{iid}{\sim} \{\pm 1\}$ are Rademacher random variables (the failure probability $\leq \delta$ addresses these random selections).

The right hand side of this upper bound contains three qualitatively different contributions. The first term describes an empirical average over N independent samples. It approximates the true expectation value by Monte Carlo sampling, and can underestimate the true average. As N increases, the approximation accuracy becomes better and, simultaneously, the probability of sampling a poor approximation diminishes exponentially. This is precisely the content of the second term. Larger sampling rates N suppress it and also allow for insisting on ever smaller failure probabilities δ. However, these two terms are still not enough for an upper bound because we would like to have a bound for *all functions* $g \in \mathcal{G}$. This is where the third term comes into play. It contains the empirical width, a statistical summary parameter for the extent of the function set $\mathcal{G}$, see e.g. [174]. Suppose, for instance, that $\mathcal{G} = \{g\}$ contains only a single function. Then, we can ignore the supremum (over a single element) and the contribution vanishes entirely (Rademacher random variables have zero expectation). The empirical width parameter can, however, grow with the size of the function set $g \in \mathcal{G}$.

In the context of bounding the performance of support vector machines, the domain variable x becomes $(\mathbf{x}, y)$, and the function family consists of the training error $g_{\boldsymbol{\alpha}}$ from Eq. (J2a), indexed by $\boldsymbol{\alpha}$. The third term in Theorem 7 can then be bounded by the largest norm of the feature vectors.

Lemma 10. *Fix a feature map* $\phi : \mathbb{R}^D \times \{\pm 1\} \to \mathcal{F}$ *and define* $g_{\boldsymbol{\alpha}}(\mathbf{x}, y) = \max\{0, 1 - y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}}\}$ *for* $\boldsymbol{\alpha} \in \mathcal{F}^*$. *Then,*

$$\mathop{\mathbb{E}}_{\varepsilon_1,\ldots,\varepsilon_N}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \tfrac{1}{\sqrt{N}} \sum_{\ell=1}^{N} \varepsilon_\ell g_{\boldsymbol{\alpha}}\left(\mathbf{x}_\ell, y_\ell\right)\right] \leq \Lambda \max_{1\leq \ell \leq N} \sqrt{\langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}}} \tag{J13}$$

for any collection $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \in \mathbb{R}^D \times \{\pm 1\}$.

Proof. Let us abbreviate the expectation over all N Rademacher random variables by $\mathbb{E}_\varepsilon$. Note that the empirical width is invariant under a constant shift of the hinge loss function: $\max\{0, 1-z\} \mapsto \max\{0, 1-z\} - 1$. In turn, the shifted loss function $L(z) = \max\{0, 1-z\} - 1$ describes a contraction, i.e. $L(0) = 0$ and $|L(z_1) - L(z_2)| \leq |z_1 - z_2|$ for all $z_1, z_2 \in \mathbb{R}$. Such contractions can only decrease the empirical width. More precisely, the Rademacher comparison principle [109, Eq. (4.20)] asserts

$$\mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \tfrac{1}{\sqrt{N}} \sum_{\ell=1}^{N} \varepsilon_\ell g_{\boldsymbol{\alpha}}(\mathbf{x}_\ell, y_\ell)\right] = \mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \tfrac{1}{\sqrt{N}} \sum_{\ell=1}^{N} \varepsilon_\ell \left(\max\{0, 1 - y_\ell\langle \boldsymbol{\alpha}, \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}}\} - 1\right)\right] \tag{J14a}$$

$$\leq \mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \tfrac{1}{\sqrt{N}} \sum_{\ell=1}^{N} \varepsilon_\ell y_\ell \langle \boldsymbol{\alpha}, \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}}\right] \tag{J14b}$$

$$= \mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \langle \boldsymbol{\alpha}, h_\varepsilon\rangle_{\mathcal{F}}\right]. \tag{J14c}$$

In the last step, we have introduced the short-hand notation $h_\varepsilon = \frac{1}{\sqrt{N}} \sum_{\ell=1}^{N} \varepsilon_\ell y_\ell \phi(\mathbf{x}_\ell) \in \mathcal{F}$. Applying a Cauchy-Schwarz inequality in feature space allows us to separate the supremum from the Rademacher randomness:

$$\mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \langle \boldsymbol{\alpha}, h_\varepsilon\rangle_{\mathcal{F}}\right] \leq \sup_{\langle \boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2} \sqrt{\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{F}}}\ \mathop{\mathbb{E}}_{\varepsilon}\left[\sqrt{\langle h_\varepsilon, h_\varepsilon\rangle_{\mathcal{F}}}\right] \leq \Lambda \sqrt{\mathop{\mathbb{E}}_{\varepsilon} \langle h_\varepsilon, h_\varepsilon\rangle_{\mathcal{F}}}. \tag{J15}$$

The last inequality is Jensen's. We complete the argument using $\mathbb{E}_\varepsilon[\varepsilon_\ell \varepsilon_{\ell'}] = \delta_{\ell,\ell'}$ and $y_\ell^2 = 1$:

$$\mathop{\mathbb{E}}_{\varepsilon} \langle h_\varepsilon, h_\varepsilon\rangle_{\mathcal{F}} = \frac{1}{N} \sum_{\ell,\ell'=1}^{N} \mathop{\mathbb{E}}_{\varepsilon}[\varepsilon_\ell \varepsilon_{\ell'}]\, y_\ell y_{\ell'} \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_{\ell'})\rangle_{\mathcal{F}} = \frac{1}{N} \sum_{\ell=1}^{N} \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}} \leq \max_{1\leq \ell \leq N} \langle \phi(\mathbf{x}_\ell), \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}}. \tag{J16}$$

□

We are now ready to prove the general connection between average prediction (J10) and training error.

Proof of Theorem 6. We consider functions $y_{\boldsymbol{\alpha}}(\mathbf{x}) = \operatorname{sign}\left(\langle \boldsymbol{\alpha}, \phi\left(\mathbf{x}\right)\rangle_{\mathcal{F}}\right) \in \{\pm 1\}$, such that $\boldsymbol{\alpha} \in \mathcal{F}^*$ obeys $\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2$. This family of functions includes all classifiers that are feasible points in the training stage (J5a) of our support vector machine. For $\boldsymbol{\alpha}$ fixed, but otherwise arbitrary, we want to compare the corresponding classifier $y_{\boldsymbol{\alpha}}\left(\mathbf{x}\right)$ to the true data label $y \in \{\pm 1\}$. Elementary reformulations then allow us to re-express the failure probability as

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[y_{\boldsymbol{\alpha}}(\mathbf{x}) \neq y\right] = \Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[\operatorname{sign}\left(\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}}\right) \neq y\right] = \Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}} < 0\right], \tag{J17}$$

because the sign is negative if and only if the number itself is. Next, we rewrite this probability as the expectation value of the associated indicator function and use $\mathbf{1}\{z \leq 0\} \leq \max\{0, 1-z\}$ for all $z \in \mathbb{R}$ to obtain

$$\Pr_{(\mathbf{x},y)\sim\mathcal{D}}\left[y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}} < 0\right] = \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbf{1}\left\{y\langle \boldsymbol{\alpha}, \phi(\mathbf{x}_\ell)\rangle_{\mathcal{F}} < 0\right\}\right] \leq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\max\left\{0, 1 - y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}}\right\}\right]. \tag{J18}$$

This upper bound is the expected value of a certain hinge loss function

$$g_{\boldsymbol{\alpha}}\left(\mathbf{x}, y\right) = \max\left\{0, 1 - y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}}\right\} \quad \text{with} \quad \langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2. \tag{J19}$$

The function is a specific element of an entire family, namely

$$\mathcal{G} = \left\{g_{\boldsymbol{\alpha}}(\cdot, \cdot) : \langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{F}} \leq \Lambda^2\right\} : \mathbb{R}^D \times \{\pm 1\} \to [0, \infty). \tag{J20}$$

The associated function values are always nonnegative and bounded. Indeed, the Cauchy-Schwarz inequality in feature space asserts

$$g_{\boldsymbol{\alpha}}(\mathbf{x}, y) \leq |y\langle \boldsymbol{\alpha}, \phi(\mathbf{x})\rangle_{\mathcal{F}}| + 1 \leq \sqrt{\langle \boldsymbol{\alpha}, \boldsymbol{\alpha}\rangle_{\mathcal{F}} \langle \phi(\mathbf{x}), \phi(\mathbf{x})\rangle_{\mathcal{F}}} + 1 \leq \Lambda \sqrt{k(\mathbf{x}, \mathbf{x})} + 1 \leq \Lambda R + 1 =: \gamma_{\max}. \tag{J21}$$

We are now in a position to use Theorem 7. With probability (at least) $1-\delta$,

$$\mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\left[g_{\boldsymbol{\alpha}}(\mathbf{x},y)\right] \leq \frac{1}{N}\sum_{\ell=1}^{N} g_{\boldsymbol{\alpha}}(\mathbf{x}_\ell,y_\ell) + 3\gamma_{\max}\sqrt{\frac{\log(2/\delta)}{2N}} + \frac{2}{\sqrt{N}}\mathop{\mathbb{E}}_{\varepsilon}\left[\sup_{\langle\boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}}\leq\Lambda^2}\frac{1}{\sqrt{N}}\sum_{\ell=1}^{N}\varepsilon_\ell g_{\boldsymbol{\alpha}}(\mathbf{x}_\ell,y_\ell)\right], \tag{J22}$$

is true for *all* dual vectors $\boldsymbol{\alpha}\in\mathcal{F}^*$ that obey $\langle\boldsymbol{\alpha},\boldsymbol{\alpha}\rangle_{\mathcal{F}}\leq\Lambda^2$. Here, $(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_N,y_N)\sim\mathcal{D}$ is a randomly sampled (but fixed) collection of labeled data points. We now use $\sqrt{k(\mathbf{x}_\ell,\mathbf{x}_\ell)}\leq R$ almost surely to apply Lemma 10 and control the empirical width term:

$$\mathop{\mathbb{E}}_{(\mathbf{x},y)\sim\mathcal{D}}\left[g_{\boldsymbol{\alpha}}(\mathbf{x},y)\right] \leq \frac{1}{N}\sum_{\ell=1}^{N} g_{\boldsymbol{\alpha}}(\mathbf{x}_\ell,y_\ell) + 3(\Lambda R+1)\sqrt{\frac{\log(2/\delta)}{2N}} + \frac{2\Lambda R}{\sqrt{N}} \tag{J23a}$$

$$\leq \frac{1}{N}\sum_{\ell=1}^{N} g_{\boldsymbol{\alpha}}(\mathbf{x}_\ell,y_\ell) + 7(\Lambda R+1)\sqrt{\frac{\log(2/\delta)}{N}}. \tag{J23b}$$

With probability (at least) $1-\delta$, this bound is valid for all hyperplane vectors $\boldsymbol{\alpha}\in\mathcal{F}$. The tightest bound is achieved for minimizing the right hand side. This is precisely what training a support vector machine does, as the first term is precisely the training error that is minimized in the training stage (J5a). The optimal solution $\boldsymbol{\alpha}^\sharp$ to this problem simultaneously produces the actual classifier $y_\sharp(\mathbf{x})$ on the left hand side and the (minimal) training error on the right hand side. □

J.3. Kernel functions for classical shadows

We have reviewed the classical shadow formalism in Appendix A. For randomized single-qubit Pauli measurements, a classical shadow approximates a n-qubit state ρ by means of T elementary tensor products. Each shadow raw data corresponds to a two-dimensional array

$$S_T(\rho) = S_T(\rho) = \left\{|s_i^{(t)}\rangle :\ i\in\{1,\ldots,n\}, t\in\{1,\ldots,T\}\right\} \in \{|0\rangle,|1\rangle,|+\rangle,|-\rangle,|\mathrm{i}+\rangle,|\mathrm{i}-\rangle\}^{n\times T} \tag{J24}$$

and is combined into an approximator of the state as

$$\sigma_T(\rho) = \frac{1}{T}\sum_{t=1}^{T}\left(3|s_1^{(t)}\rangle\langle s_1^{(t)}| - \mathbb{I}\right)\otimes\cdots\otimes\left(3|s_n^{(t)}\rangle\langle s_n^{(t)}| - \mathbb{I}\right) = \frac{1}{T}\sum_{t=1}^{T}\sigma_1^{(t)}\otimes\cdots\otimes\sigma_n^{(t)}, \tag{J25}$$

where we have introduced the short-hand notation $\sigma_i^{(t)} = 3|s_i^{(t)}\rangle\langle s_i^{(t)}| - \mathbb{I}$. For these quantum state representations, we fix parameters $\tau,\gamma>0$ and introduce a suggestive, yet finite-dimensional feature map. For large, but finite, integers $D,R>0$ we define

$$\phi^{(\mathrm{finite})}\left(S_T(\rho)\right) = \bigoplus_{d=0}^{D}\sqrt{\frac{\tau^d}{d!}}\Big(\bigoplus_{r=0}^{R}\sqrt{\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r}\bigoplus_{i_1=1}^{r}\cdots\bigoplus_{i_r=1}^{r}\frac{1}{T}\sum_{t=1}^{T}\mathrm{vec}\left(\sigma_{i_1}^{(t)}\right)\otimes\cdots\otimes\mathrm{vec}\left(\sigma_{i_r}^{(t)}\right)\Big)^{\otimes d}, \tag{J26}$$

Here, $\mathrm{vec}(\cdot)$ denotes an appropriate vectorization operation that maps the real-valued vector space $\mathbb{H}_2$ of Hermitian 2×2 matrices to $\mathbb{R}^4$ such that the Hilbert-Schmidt inner product is preserved: $\langle\mathrm{vec}(A),\mathrm{vec}(B)\rangle = \mathrm{tr}(AB)$.

This feature map embeds classical shadows in a very large-dimensional, real-valued feature space $\mathcal{F}^{(\mathrm{finite})}$. This feature space arises from taking direct sums and tensor products of $\mathrm{vec}(\mathbb{H}_2)\simeq\mathbb{R}^4$. We can extend the standard inner product $\langle\cdot,\cdot\rangle$ on $\mathbb{R}^4$ to this feature space by setting $\langle x_1\oplus x_2, y_1\oplus y_2\rangle = \langle x_1,y_1\rangle + \langle x_2,y_2\rangle$ (direct sums), as well as $\langle x_1\otimes x_2, y_1\otimes y_2\rangle = \langle x_1,y_1\rangle\langle x_2,y_2\rangle$ (tensor products) and extend these definitions linearly. Doing so equips the feature space $\mathcal{F}^{(\mathrm{finite})}$ with a well-defined inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}^{(\mathrm{finite})}}$. The inner product and feature map induce a kernel function on pairs of classical shadows of equal size T:

$$k^{(\mathrm{finite})}\left(S_T(\rho_1),\tilde{S}_T(\rho_2)\right) = \left\langle\phi^{(\mathrm{finite})}\left(S_T(\rho_1)\right),\phi^{(\mathrm{finite})}\left(\tilde{S}_T(\rho_2)\right)\right\rangle_{\mathcal{F}^{(\mathrm{finite})}} \tag{J27a}$$

$$= \sum_{d=0}^{D}\frac{\tau^d}{d!}\left(\sum_{r=0}^{R}\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r\sum_{i_1=1}^{n}\cdots\sum_{i_r=1}^{n}\frac{1}{T^2}\sum_{t,t'=1}^{T}\left\langle\mathrm{vec}\left(\sigma_{i_1}^{(t)}\right),\mathrm{vec}\left(\tilde{\sigma}_{i_1}^{(t')}\right)\right\rangle\cdots\left\langle\mathrm{vec}\left(\sigma_{i_r}^{(t)}\right),\mathrm{vec}\left(\tilde{\sigma}_{i_r}^{(t')}\right)\right\rangle\right)^d \tag{J27b}$$

$$= \sum_{d=0}^{D}\frac{\tau^d}{d!}\left(\sum_{r=0}^{R}\frac{1}{r!}\left(\frac{\gamma}{n}\right)^r\sum_{i_1=1}^{n}\cdots\sum_{i_r=1}^{n}\mathrm{tr}\left(\left(\frac{1}{T}\sum_{t=1}^{T}\sigma_{i_1}^{(t)}\otimes\cdots\otimes\sigma_{i_r}^{(t)}\right)\right)\left(\frac{1}{T}\sum_{t'=1}^{T}\tilde{\sigma}_{i_1}^{(t')}\otimes\cdots\otimes\tilde{\sigma}_{i_r}^{(t')}\right)\right)^d \tag{J27c}$$

$$= \sum_{d=0}^{D} \frac{1}{d!} \left(\frac{\tau}{T^2} \sum_{t,t'=1}^{T} \sum_{r=0}^{R} \frac{1}{r!} \left(\frac{\gamma}{n}\right)^r \sum_{i=1}^{n} \cdots \sum_{i_r=1}^{r} \operatorname{tr}\left(\sigma_{i_1}^{(t)} \tilde{\sigma}_{i_1}^{(t')}\right) \cdots \operatorname{tr}\left(\sigma_{i_r}^{(t)} \tilde{\sigma}_{i_r}^{(t')}\right) \right)^d \tag{J27d}$$

$$= \sum_{d=0}^{D} \frac{1}{d!} \left(\frac{\tau}{T^2} \sum_{t,t'=1}^{T} \sum_{r=0}^{R} \frac{1}{r!} \left(\frac{\gamma}{n} \sum_{i=1}^{n} \operatorname{tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t')}\right) \right)^r \right)^d . \tag{J27e}$$

This kernel function sill looks somewhat complicated, but it simplifies considerably if we first take $R \to \infty$ and then $D \to \infty$:

$$k^{(\text{shadow})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2)\right) := \lim_{D\to\infty} \lim_{R\to\infty} k^{(\text{finite})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2)\right) \tag{J28a}$$

$$= \lim_{D\to\infty} \sum_{d=0}^{D} \frac{1}{d!} \left(\frac{\tau}{T^2} \sum_{t,t'=1}^{T} \lim_{R\to\infty} \sum_{r=0}^{R} \frac{1}{r!} \left(\frac{\gamma}{n} \sum_{i=1}^{n} \operatorname{tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t')}\right) \right)^r \right)^d \tag{J28b}$$

$$= \exp\left(\frac{\tau}{T^2} \sum_{t,t'=1}^{T} \exp\left(\frac{\gamma}{n} \sum_{i=1}^{n} \operatorname{tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t')}\right) \right) \right) \tag{J28c}$$

We call this kernel function a *shadow kernel*. In contrast to its finite approximations, this kernel function can be computed very efficiently. Trace inner products between single-qubit shadow constituents assume one out of 3 values only:

$$\operatorname{tr}\left(\sigma_i^{(t)} \tilde{\sigma}_i^{(t)}\right) = \operatorname{tr}\left((3|s_i^{(t)}\rangle\langle s_i^{(t)}| - \mathbb{I})(3|\tilde{s}_i^{(t)}\rangle\langle \tilde{s}_i^{(t)}| - \mathbb{I})\right) = 9\left|\langle s_i^{(t)}|\tilde{s}_i^{(t)}\rangle\right|^2 - 4 \in \{-4, 1/2, 5\}\,. \tag{J29}$$

And we need to compute exactly nT^2 of them to unambiguously characterize the shadow kernel (J28a). The total cost for evaluating shadow kernels also amounts to

$$\mathcal{O}\left(nT^2\right) \qquad \text{(shadow kernel evaluation cost)} \tag{J30}$$

arithmetic operations. As long as T is not too large, this is extremely efficient, given that we combine classical approximations of n-qubit quantum states ρ_1, ρ_2 which way well have $(4^n - 1)$ degrees of freedom. Eq. (J29) also ensures that shadow kernels remain bounded functions:

$$0 \leq k^{(\text{shadow})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2)\right) \leq \exp\left(\tau \exp\left(5\gamma\right)\right), \tag{J31}$$

because exponential functions are nonnegative and monotonic.

While easy to evaluate and conceptually appealing, the shadow kernel does have its downsides. By construction, the associated feature space is not finite-dimensional anymore. This can complicate a thorough analysis of support vector machines substantially. In particular, it is a priori not clear if powerful results, like Theorem 6, cover the shadow kernel as well. Fortunately, we can bypass such mathematical subtleties by approximating $k^{(\text{shadow})}(\cdot,\cdot)$ with $k^{(\text{finite})}(\cdot,\cdot)$, where D and R are large, but finite, numbers. This incurs an additional approximation error, but allows us to formulate theoretical prediction and training guarantees exclusively for finite-dimensional feature spaces. What is more, elementary approximation results from calculus ensure that we can make this additional approximation error arbitrarily small by making the cutoffs sufficiently large. Taylor's approximation theorem, for instance, shows that $D = \mathrm{e}^2\tau\exp(5\gamma) + \log(1/\eta) - 1$, as well as $R = 5\mathrm{e}^2\gamma + \tau\exp(5\gamma) + \log(\tau/\eta) - 1$ ensure

$$\left|k^{(\text{shadow})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2)\right) - k^{(\text{finite})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2)\right)\right| \leq 2\eta \tag{J32}$$

for all pairs of classical shadows with compatible size T. Properly tuning γ and τ would yield better prediction performance in practice. Nevertheless, for simplicity, we will assume $\gamma = \tau = 1$ in the following theoretical analysis.

Finite-dimensional feature space approximations also allow us to highlight the expressiveness behind the shadow kernel (J28a). It describes (the limit of) a feature map that extracts *all* tensor powers of *all* subsystem operators $X_A = \operatorname{tr}_{\neg A}(X) \in \mathbb{H}_2^{\otimes|A|}$, where $A \subset [n] = \{1, \ldots, n\}$. In particular, any function that can be written as a finite power series, of degree at most d_p, in reduced subsystem operators, of size at most r, becomes a *linear* function in feature space, represented by the dual vector $\boldsymbol{\alpha}_f$:

$$f\left(S_T(\rho)\right) = \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1 \ldots A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \operatorname{tr}\left(O_{A_1,\ldots,A_d} \operatorname{tr}_{\neg A_1}\left(\sigma_T(\rho)\right) \otimes \cdots \otimes \operatorname{tr}_{\neg A_d}\left(\sigma_T(\rho)\right)\right) \tag{J33a}$$

$$= \langle \boldsymbol{\alpha}_f, \phi^{(\text{finite})}(S_T(\rho)) \rangle_{\mathcal{F}^{(\text{finite})}}, \tag{J33b}$$

provided that $d_p \leq D, r \leq R$. The (extended) Euclidean norm of $\boldsymbol{\alpha}_f$ is also bounded. Use Eq. (J26) (with tuning parameters $\gamma, \tau = 1$) to compute

$$\langle \boldsymbol{\alpha}_f, \boldsymbol{\alpha}_f \rangle_{\mathcal{F}^{(\text{finite})}} \leq \sum_{d=0}^{d_p} \frac{(r! n^r)^d}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \text{tr}\left(O^2_{A_1,\ldots,A_d}\right) \tag{J34a}$$

$$\leq \sum_{d=0}^{d_p} \frac{(r! n^r)^d}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} 2^{rd} \|O_{A_1,\ldots,A_d}\|^2_\infty \tag{J34b}$$

$$\leq (2nr)^{rd_p} \max_{\substack{d \leq d_p, A_1,\ldots,A_d \\ \subset \{1,\ldots,n\}, |A_i| \leq r}} \|O_{A_1,\ldots,A_d}\|_\infty \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \|O_{A_1,\ldots,A_d}\|_\infty \tag{J34c}$$

$$\leq (2nr)^{rd_p} d_p^{d_p} \left(\sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \|O_{A_1,\ldots,A_d}\|_\infty \right)^2 \tag{J34d}$$

Here, we have used the fundamental Schatten-p norm relation $\|X\|_2 \leq \sqrt{\dim(X)} \|X\|_\infty$, as well as the assumption that each $O_{A_1,\ldots,A_d}$ is supported on a total tensor product space with dimension 2^{rd} (a tensor product of d subsystems comprised of at most r qubits each). The second to last inequality follow from using $\sum_i x_i^2 \leq \max_i |x_i| \sum_i |x_i|$, and Stirling's formula. The final simplifications uses Stirling's formula again as well as the fact that $\sum_i |x_i| \geq \max_i |x_i|$.

J.4. Physical assumptions about classifying quantum phases of matter

We want to learn how to classify two phases of n-qubit states: either ρ belongs to phase A ($y(\rho) = +1$) or ρ belongs to phase B ($y(\rho) = -1$). We assume that we have access to labeled classical shadows: $\{(S_T(\rho_\ell), y(\rho_\ell)) : \ell \in \{1, \ldots, N\}\}$, where each $S_T(\rho_\ell)$ is classical shadow data obtained from performing T randomized single-qubit measurements on independent copies of ρ_ℓ. We can use this raw data to form classical representations $\sigma_T(\rho_\ell)$ of the underlying quantum state ρ_ℓ, see Eq. (1). The number T determines the resolution of these approximations. Note that $\sigma_T(\rho_\ell) \approx \rho_\ell$ can only become exact for $T \geq \exp(\Omega(n))$ [72, 73]. This would be far too costly for experimental implementations and efficient data processing. For instance, recall from Eq. (J30) that a single shadow kernel evaluation scales quadratically in T. In this section, we show that we can choose much coarser resolutions if the underlying phase can be classified by a nice analytic function on reduced density matrices.

Assumption 1 (well-conditioned phase separation)**.** *Consider two phases among n-qubit states. For $\epsilon > 0$, we assume that there exists a function f on reduced r-body density matrices $\rho_A = \text{tr}_{\neg A}(\rho)$ that can distinguish the two phases in question. In particular,*

$$f(\rho) = f\left(\{\rho_A : A \subset \{1, \ldots, n\}, |A| \leq r\}\right) \quad \textit{satisfies} \tag{J35a}$$

$$f(\rho) \quad \begin{cases} > +1 & \textit{for all } \rho \textit{ that belong to phase } A \ (y(\rho) = +1), \\ < -1 & \textit{for all } \rho \textit{ that belong to phase } B \ (y(\rho) = -1). \end{cases} \tag{J35b}$$

Moreover, we assume that $f(\rho)$ can be approximated by a truncated power series

$$f^{(d_p)}(\rho) = \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \text{tr}\left(O_{A_1,\ldots,A_d} \rho_{A_1} \otimes \cdots \otimes \rho_{A_d}\right), \tag{J36}$$

up to constant accuracy: $\left|f(\rho) - f^{(d_p)}(\rho)\right| \leq 0.25$ for all n-qubit quantum states ρ. We refer to d_p as the truncation degree and define the normalization constant

$$C = \sum_{d=0}^{d_p} \frac{1}{d!} \Big(\sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \|O_{A_1,\ldots,A_d}\|_\infty \Big). \tag{J37}$$

We don't need to know the normalization constant exactly. An upper bound is fully adequate for the theoretical analysis presented in this section.

Morally, the second part of Assumption 1 requires that the phase classication function can be well-approximated by a degree-d_p-polynomial in reduced density matrices. The actual formulation is general enough to encompass most physically relevant functions. Let us illustrate this by means of three popular examples.

a. Subsystem purity: Fix a subsystem $A \subset \{1, \dots, n\}$ comprised of $|A| = r$ qubits and let $\rho_A = \mathrm{tr}_{\neg A}(\rho)$ be the associated r-body density matrix. The subsystem purity $f(\rho) = \mathrm{tr}(\rho_A^2)$ is a quadratic polynomial in this reduced density matrix. We can rewrite this as $f^{(2)}(\rho) = \mathrm{tr}(S_A \rho_A \otimes \rho_A)$, where S_A denotes the swap operator between two copies of the subsystem A. This reformulation is also an *exact* approximation of $f(\rho)$ with degree $d_p = 2$ and normalization constant $C = \frac{1}{2!}\|S_A\|_\infty = \frac{1}{2}$. These arguments readily extend to averages of multiple subsystem purities.

b. Subsystem Rényi entropy: Let us consider the subsystem Rényi entropy of order two $H_2(\rho_A) = -\log\left(\mathrm{tr}(\rho_A^2)\right)$ (the argument will generalize straightforwardly to higher order entropies). This function is closely related to the subsystem purity, but also features a logarithm. And, although the logarithm is *not* a polynomial, $-\log(1-x)$ can be accurately approximated by the truncated Mercator series. A crude, but sufficient, bound ensures

$$l^{(d_p)}(x) = \sum_{d=1}^{d_p} \frac{1}{d} x^d \quad \text{obeys} \quad \left| l^{(d_p)}(x) - \log(1-x) \right| \leq x^{d_p} \log\left(1/(1-x)\right) \quad \text{for } x \in (-1, 1). \tag{J38}$$

We can now approximate $H_2(\rho_A) = -\log\left(1 - (1 - \mathrm{tr}(\rho_A^2))\right)$ by $l^{(d_p)}(1 - \mathrm{tr}(\rho_A^2))$. Subsystem purities necessarily obey $\mathrm{tr}(\rho_A^2) \geq 2^{-|A|} = 2^{-r}$. This allows us to conclude

$$\left| l^{(d_p)}\left(1 - \mathrm{tr}(\rho_A^2)\right) - H_2(\rho_A) \right| \leq \left(1 - 2^{-r}\right)^{d_p} r \log(2) \leq \log(2) r \exp\left(-d_p / 2^r\right) \tag{J39}$$

which drops beneath 0.25 if we set $d_p = \log(4\log(2)r)2^r = \mathcal{O}\left(\log(r)2^r\right)$. This degree scales exponentially in the subsystem size r, but is independent of total dimension. We can also use $1 = \mathrm{tr}(\rho_A)^2 = \mathrm{tr}\left(\mathbb{I}_A^{\otimes 2} \rho_A^{\otimes 2}\right)$ and $\mathrm{tr}(X)\mathrm{tr}(Y) = \mathrm{tr}(X \otimes Y)$ to bring this polynomial approximation onto the form advertised in Eq. (J36). Indeed,

$$l^{(d_p)}\left(1 - \mathrm{tr}(\rho_A^2)\right) = l^{(d_p)}\left(\mathrm{tr}\left((\mathbb{I}_A^{\otimes 2} - S_A)\rho_A^{\otimes 2}\right)\right) = \sum_{d=1}^{d_p} \frac{1}{d!} \mathrm{tr}\left((d-1)!\left(\mathbb{I}_A^{\otimes 2} - S_A\right)^{\otimes d} \rho_A^{\otimes 2d}\right) \quad \text{and} \tag{J40a}$$

$$C = \sum_{d=1}^{d_p} \frac{1}{d!} \left\|(d-1)!(\mathbb{I}_A^{\otimes 2} - S_A)^{\otimes d}\right\|_\infty = \sum_{d=1}^{d_p} \frac{1}{d} \left\|\mathbb{I}_A^{\otimes 2} - S_A\right\|_\infty^d = \sum_{d=1}^{d_p} \frac{1}{d} \approx \log(d_p). \tag{J40b}$$

This analysis readily extends to higher order Rényi entropies, as well as averages over multiple subsystems.

c. Entanglement entropy: This is where things start to get somewhat interesting, because the entanglement (von Neumann) entropy $H(\rho_A) = -\mathrm{tr}\left(\rho_A \log(\rho_A)\right) \in [0, r\log(2)]$ of a r-body subsystem is notoriously difficult to accurately approximate with a polynomial [60]. Fortunately, Assumption 1 does not require an accurate approximation – a constant error of size 1/4 is fine. To achieve this goal, we make the following polynomial ansatz in the reduced density matrix ρ_A:

$$H^{(d_p)}(\rho_A) = -\mathrm{tr}\left((\rho_A - \mathbb{I}_A) + \sum_{k=2}^{d_p} \frac{(\mathbb{I}_A - \rho_A)^k}{k(k-1)}\right) \tag{J41}$$

Let λ_i denote the eigenvalues of a subsystem density matrix ρ_A and note that there are 2^r eigenvalues in ρ_A. We can rewrite the entanglement entropy and the polynomial ansatz as

$$H(\rho_A) = -\sum_{i=1}^{2^r} \lambda_i \log(\lambda_i) \quad \text{and} \tag{J42a}$$

$$H^{(d_p)}(\rho_A) = -\sum_{i=1}^{2^r} \left((\lambda_i - 1) + \sum_{k=2}^{d_p} \frac{(1-\lambda_i)^k}{k(k-1)}\right), \tag{J42b}$$

respectively. Using Taylor's theorem in the interval $[0, 1]$, we have

$$x\log(x) = (x-1) + \left(\sum_{k=2}^{\infty} \frac{(1-x)^k}{k(k-1)}\right). \tag{J43}$$

Note that at $x = 0$, $x \log x = 0$ and the infinite sum comprising the second term on the right hand side also converges to 1. This ensures that the above equality is valid for the closed interval $[0, 1]$. We shall also use the following identity

$$\sum_{k=2}^{n} \frac{1}{k(k-1)} = 1 - \frac{1}{n}, \tag{J44}$$

which remains valid even in the limit $n \to \infty$. We can combine Eq. (J43) and (J44) to obtain an approximation error for our polynomial ansatz function. For all $x \in [0, 1]$, we have

$$\left| x \log(x) - \left((x-1) + \left(\sum_{k=2}^{d_p} \frac{(1-x)^k}{k(k-1)} \right) \right) \right| \leq \sum_{k=d_p+1}^{\infty} \frac{(1-x)^k}{k(k-1)} \leq \sum_{k=d_p+1}^{\infty} \frac{1}{k(k-1)} = \frac{1}{d_p}. \tag{J45}$$

This allows us to bound the approximation error for each individual eigenvalue $\lambda_i \in [0, 1]$ of ρ_A. There are in total 2^r eigenvalues and a triangle inequality asserts

$$|H(\rho_A) - H^{(d_p)}(\rho_A)| \leq \sum_{i=1}^{2^r} \left| \lambda_i \log(\lambda_i) - \left((\lambda_i - 1) + \left(\sum_{k=2}^{d_p} \frac{(1-\lambda_i)^k}{k(k-1)} \right) \right) \right| \leq \frac{2^d}{d_p}. \tag{J46}$$

By choosing $d_p = 2^{r+2}$, we can approximate the entanglement entropy in r-body subsystem by a polynomial function. As long as the subsystem size r is a constant independent of total system size n, the polynomial approximation degree d_p is also a constant. And it is not hard to check that the same is true for the normalization constant C. This analysis readily extends to averages of multiple entanglement entropies.

J.5. Training with shadow kernels

We are now ready to dive into the main results of this section: converting Assumption 1 into a statement about classical shadows and their expressiveness when it comes to training a support vector machine. Our measure of similarity is the *shadow kernel* (J28a) evaluated on classical shadows. The kernel matrix is

$$[\mathbf{K}]_{\ell\ell'} = k^{(\text{shadow})}\left(S_T(\rho_\ell), S_T(\rho_{\ell'})\right) \quad \text{for } \ell, \ell' \in \{1, \ldots, N\}, \tag{J47}$$

and implicitly specifies the feature map, as well as the nonlinear geometry with respect to which we want to find classifiers for phases. We begin by approximating the true classifier, given as a nonlinear function $f(\rho)$ in Assumption 1, by a finite power series $f^{(d_p)}(\rho)$ with degree-d_p. We will then use $f^{(d_p)}(\rho)$ as an approximate phase classifier. Recalling Eq. (J33b), a finite power series $f^{(d_p)}(S_T(\rho))$ is linear in feature space, with its corresponding dual vector $\boldsymbol{\alpha}_f$ defining a candidate hyperplane for separating the two phases. To complete the connection to the support vector machines from Section J.1, we must ensure that $f^{(d_p)}(S_T(\rho))$ does not differ substantially from the approximate phase classifier $f^{(d_p)}(\rho)$ from Assumption 1. This is the content of the following auxiliary statement.

Lemma 11. *Suppose that Assumption 1 is valid for a function on reduced r-body density matrices with the two constants $C \geq 1$ and $d_p \in \mathbb{N}$. For any $0 < \epsilon < 1$, classical shadows of size*

$$T = (32/3) d_p^2 C^2 12^r \left(r \left(\log(n) + \log(12) \right) + \log(1/\delta) \right) / \epsilon^2 \tag{J48}$$

suffice to ϵ-approximate $f^{(d_p)}(\rho)$ with high probability. In particular, for any density matrix $\rho \in \mathbb{H}_2^{\otimes n}$,

$$\left| f^{(d_p)}(S_T(\rho)) - f^{(d_p)}(\rho) \right| \leq \epsilon \tag{J49}$$

with probability at least $1 - \delta$ (over the randomized measurement settings and outcomes producing $S_T(\rho)$).

A proof can be found at the end of this subsection. With high probability, this statement ensures that existence of a well-conditioned phase separation implies the existence of a separating hyperplane in shadow feature space. This, in turn, is enough to ensure that the SVM training stage can be executed perfectly: solving the training problem (J5a) efficiently yields a separating hyperplane parametrization $\boldsymbol{\alpha}_\sharp$ that (1) lies in the subspace $\mathbb{R}^N$ of $\mathcal{F}^{(\text{shadow})}$ spanned by the N training vectors, and (2) performs at least as well as $\boldsymbol{\alpha}_f$. Since we are guaranteed that $\boldsymbol{\alpha}_f$ separates training data perfectly and achieves zero training error, $\boldsymbol{\alpha}_\sharp$ must be at least as good: $\mathrm{E}_{\text{tr}}(\boldsymbol{\alpha}_\sharp) \leq \mathrm{E}_{\text{tr}}(\boldsymbol{\alpha}_f) = 0$ with high probability. The main result of this section formalizes this observation.

Proposition 5. *Suppose that Assumption 1 is valid for some function on reduced r-body density matrices with normalization constant C and truncation degree d_p. Then, for $\delta \in (0,1)$, a (joint) classical shadow size $T = (512/3)d_p^2 C^2 12^r \left(r\left(\log(n)+\log(12)\right)+\log(N/\delta)\right)$ ensures that we can achieve zero training error when solving (J5a) with squared margin constant $\Lambda^2 = 4\left(2rn\right)^{rd_p} d_p^{d_p} C^2$.*

The extra constraint $\Lambda^2 \geq \langle \boldsymbol{\alpha}_f, \boldsymbol{\alpha}_f \rangle_{\mathcal{F}^{(\mathrm{finite})}}$ ensures that the ideal separating hyperplane is a feasible point of the training problem (J5a).

Proof of Proposition 5. We establish the claim not for the shadow kernel itself ($k^{(\mathrm{shadow})}(\cdot,\cdot)$), but for large finite-dimensional approximations ($k^{(\mathrm{finite})}(\cdot,\cdot)$) thereof. We begin by utilizing Eq. (J36) that approximates the nonlinear function $f(\rho)$ by a finite power series $f^{(d_p)}(\rho)$ with the approximation error,

$$|f(\rho) - f^{(d_p)}(\rho)| \leq 0.25. \tag{J50}$$

For each $\ell \in \{1,\ldots,N\}$, we invoke Lemma 11 using the truncated Taylor series to conclude

$$\Pr\left[\left|f^{(d_p)}(\rho_\ell) - f^{(d_p)}\left(S_T(\rho_\ell)\right)\right| \geq 0.25\right] \leq \delta/N, \tag{J51}$$

provided that $T = (512/3)d_p^2 C^2 12^r \left(r\left(\log(n)+\log(12)\right)+\log(N/\delta)\right)$. Triangle inequality and a union bound allows us to combine these approximation guarantees into a single statement:

$$\max_{1\leq \ell \leq N} \left|f(\rho_\ell) - f^{(d_p)}\left(S_T(\rho_\ell)\right)\right| \leq 0.5 \quad \text{with probability (at least) } 1-\delta. \tag{J52}$$

Let us condition on this desirable event and also assume hat the cutoff values of our finite kernel approximation are large enough, i.e. $D \geq d_p$, $R \geq r$). Then, the function

$$2f^{(d_p)}(S_T(\rho_\ell)) = \langle \boldsymbol{\alpha}_f, \phi^{(\mathrm{finite})}(S_T(\rho_\ell))\rangle \tag{J53}$$

describes a linear function in feature space $\mathcal{F}^{(\mathrm{finite})}$ that is guaranteed to achieve zero training error. Indeed, combine Eq. (J35) and Eq. (J52) to ensure $\left|2f^{(d_p)}(S_T(\rho_\ell))\right| \geq 2(|f(\rho_\ell)| - \left|f(\rho_\ell) - f^{(d_p)}(S_T(\rho_\ell))\right|) \geq 2(1-0.5) = 1$ and, moreover, $\mathrm{sign}\left(f^{(d_p)}(S_T(\rho_\ell))\right) = \mathrm{sign}\left(f(\rho_\ell)\right) = y(\rho_\ell) \in \{\pm 1\}$ for all $\ell \in \{1,\ldots,N\}$. In turn,

$$\sum_{\ell=1}^{N} \max\left\{0, 1 - y(\rho_\ell)\langle \boldsymbol{\alpha}_f, \phi^{(\mathrm{finite})}\left(S_T(\rho_\ell)\right)\rangle\right\} = \sum_{\ell=1}^{N} \max\left\{0, 1 - \mathrm{sign}\left(f^{(d_p)}(S_T(\rho_\ell))\right) 2f^{(d_p)}\left(S_T(\rho_\ell)\right)\right\} \tag{J54a}$$

$$= \sum_{\ell=1}^{N} \max\left\{0, 1 - \left|f^{(d_p)}\left(S_T(\rho_\ell)\right)\right|\right\} = 0. \tag{J54b}$$

Since zero is the smallest possible training error, the minimizer of the original training problem (J5a) must also achieve zero, provided that $\boldsymbol{\alpha}_f$ is actually a feasible point of this optimization. We can, however, ensure this by choosing the squared margin constant large enough. Eq. (J34) and Assumption 1 ensures

$$\langle \boldsymbol{\alpha}_f, \boldsymbol{\alpha}_f \rangle_{\mathcal{F}^{(\mathrm{finite})}} \leq 4\left(2rn\right)^{rd_p} d_p^{d_p} C^2. \tag{J55}$$

Choosing a squared margin size Λ^2 that exceeds this bound ensures that $\boldsymbol{\alpha}_f$ is indeed a feasible point of the training problem (J5a) and the claim follows. □

We conclude our discussion on training with shadow kernels by providing a rigorous proof of the auxiliary statement.

Proof of Lemma 11. It suffices to analyze implications of Lemma 1: for $\eta, \delta \in (0,1)$

$$T \geq (8/3)12^r \left(r\left(\log(n)+\log(12)\right)+\log(1/\delta)\right)/\eta^2 \;\Rightarrow \max_{A\subset\{1,\ldots,n\},|A|\leq r} \left\|\mathrm{tr}_{\neg A}\left(\sigma_T(\rho)\right) - \mathrm{tr}_{\neg A}(\rho)\right\|_1 \leq \eta \tag{J56}$$

with probability at least $1-\delta$. Here, $\|\cdot\|_1$ denotes the trace norm. Abbreviate $\mathrm{tr}_{\neg A_i}\left(\sigma_T(\rho)\right)$ and $\mathrm{tr}_{\neg A_i}(\rho)$ as σ_{A_i} and ρ_{A_i}, respectively. A combination of triangle inequalities and Matrix Hoelder ($\mathrm{tr}(XY) \leq \|X\|_\infty \|Y\|_1$) asserts

$$\left|f^{(d_p)}(\rho) - f^{(d_p)}\left(S_T(\rho)\right)\right| \tag{J57a}$$

$$\leq \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \left| \mathrm{tr}\left(O_{A_1,\ldots,A_r} \left(\rho_{A_1} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right) \right) \right| \tag{J57b}$$

$$\leq \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \left\| O_{A_1,\ldots,A_d} \right\|_\infty \left\| \rho_{A_1} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right\|_1 . \tag{J57c}$$

Next, we fix a trace norm contribution and use a telescoping trick ($A_1 \otimes A_2 - B_1 \otimes B_2 = (A_1 - B_1) \otimes A_2 + B_1 \otimes (A_2 - B_2)$), as well as a triangle inequality and $\|\rho_{A_i}\|_1 = \mathrm{tr}(\rho_{A_i}) = 1$ to infer

$$\left\| \rho_{A_1} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right\|_1 \tag{J58a}$$

$$= \left\| (\rho_{A_1} - \sigma_{A_1}) \otimes \rho_{A_2} \otimes \cdots \otimes \rho_{A_d} + \sigma_{A_1} \otimes (\rho_{A_2} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_2} \otimes \cdots \otimes \sigma_{A_d}) \right\|_1 \tag{J58b}$$

$$\leq \|\rho_{A_1} - \sigma_{A_1}\|_1 \|\rho_{A_2}\|_1 \cdots \|\rho_{A_d}\|_1 + \|\sigma_{A_1}\|_1 \left\| \rho_{A_2} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right\|_1 \tag{J58c}$$

$$\leq \|\rho_{A_1} - \sigma_{A_1}\|_1 + (1 + \|\rho_{A_1} - \sigma_{A_1}\|_1) \left\| \rho_{A_2} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right\|_1 \tag{J58d}$$

$$\leq \eta + (1 + \eta) \|\rho_{A_2} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \sigma_{A_d}\|_1. \tag{J58e}$$

The last line follows from Rel. (J56). Iterating this simplification procedure ensures

$$\left\| \rho_{A_1} \otimes \cdots \otimes \rho_{A_d} - \sigma_{A_1} \otimes \cdots \otimes \sigma_{A_d} \right\|_1 \leq \eta \sum_{k=0}^{d-1} (1 + \eta)^k = (1 + \eta)^d - 1. \tag{J59}$$

According to Rel. (J56), such an upper bound is valid for every trace norm contribution in Eq. (J57). This allows us to obtain

$$\left| f^{(d_p)}(\rho) - f^{(d_p)}\left(S_T(\rho)\right) \right| \leq \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \left\| O_{A_1,\ldots,A_d} \right\|_\infty \left[(1 + \eta)^d - 1 \right] \tag{J60a}$$

$$\leq \left[(1 + \eta)^{d_p} - 1 \right] \sum_{d=0}^{d_p} \frac{1}{d!} \sum_{A_1,\ldots,A_d \subset \{1,\ldots,n\}, |A_i| \leq r} \left\| O_{A_1,\ldots,A_d} \right\|_\infty \tag{J60b}$$

$$= C \left[(1 + \eta)^{d_p} - 1 \right]. \tag{J60c}$$

Here, we have used Assumption 1. Finally, by choosing $\eta = \epsilon/(2Cd_p)$, we can see that

$$\left| f^{(d_p)}(\rho) - f^{(d_p)}\left(S_T(\rho)\right) \right| \leq C \left[\left(1 + \frac{\epsilon}{2Cd_p} \right)^{d_p} - 1 \right] \leq C[\exp(\epsilon/2C) - 1] \leq \epsilon. \tag{J61}$$

The second inequality follows from $(1 + x/n)^n \leq \exp(x), \forall |x| \leq n, n \geq 1$. The third inequality utilizes $\exp(x) \leq 1 + 2x, \forall x \in [0, 1]$. The claim of Lemma 11 now follows from inserting this specific choice of η into Rel. (J56). □

J.6. Prediction based on shadow kernels

We now have all pieces in place to prove strong bounds on the prediction error of a SVM based on shadow kernels. The main result of this section will be a consequence of Theorem 6. For fixed parameters $\tau, \gamma = 1$, the shadow kernel (J28a) (and finite approximations thereof) is always bounded when applied to classical shadows. Eq. (J31) (under $\tau = \gamma = 1$) asserts

$$k^{(\mathrm{shadow})}\left(S_T(\rho_1), \tilde{S}_T(\rho_2) \right) \leq \exp\left(\exp\left(5\right)\right) \tag{J62}$$

for any T and quantum states ρ_1, ρ_2. This bound readily extends to finite dimensional approximations $k^{(\mathrm{finite})}(\cdot, \cdot)$. Next, we need to specify a distribution. We assume that $\tilde{\mathcal{D}}$ is a distribution over n-qubit quantum states ρ that either belong to phase A or phase B. We sample quantum states $\rho_\ell \sim \tilde{\mathcal{D}}$ accordingly, but are not permitted to process them directly. Instead, we obtain a (randomly generated) classical shadow of size T. Denote the raw data by $S_T(\rho_\ell)$ which allows us to produce a state approximation $\sigma_T(\rho_\ell)$. We do, however, require that we have direct access to the label $y(\rho_\ell) \in \{\pm 1\}$ associated with the phase of ρ_ℓ. This produces a joint distribution over input data $S_T(\rho_\ell)$ and the label $y(\rho_\ell)$ which we call $\mathcal{D}$. In summary, we assume that training data and new data are generated independently from this data distribution: $(S_T(\rho_1), y(\rho_1)), \ldots, (S_T(\rho_N), y(\rho_N)), (S_T(\rho), y) \sim \mathcal{D}$. We are now ready to combine Theorem 6 (the prediction error is bounded by the training error) and Proposition 5 (the training error vanishes if a good phase classifier exists) to obtain a powerful result about generalization.

Corollary 1. *Fix $\delta, \epsilon \in (0,1)$ and suppose there exists an analytic function on reduced r-body density matrices that can distinguish phases: $f(\rho) > 1$ if $\rho \in$ phase A and $f(\rho) < -1$ else if $\rho \in$ phase B. Let C be the normalization constant and d_p be the truncation degree given in Assumption 1. Suppose that we obtain identically distributed training data $(S_T(\rho_1), y(\rho_1)), \ldots, (S_T(\rho_N), y(\rho_N)) \sim \mathcal{D}$ such that*

$$T \geq (512/3) d_p^2 C^2 12^r \left(r\left(\log(n) + \log(12)\right) + \log(N/\delta)\right) \quad \text{and} \tag{J63a}$$

$$N \geq 256\,(2rn)^{rd_p}\, d_p^{d_p} C^2 \exp(\exp(5)) \log(4/\delta)/\epsilon^2. \tag{J63b}$$

Then, solving the training problem (J5a) for the shadow kernel with squared margin constant $\Lambda^2 = 4\,(2rn)^{rd_p}\, d_p^{d_p} C^2$ will produce a hyperplane $\boldsymbol{\alpha}_\sharp \in \mathbb{R}^N$ in shadow feature space that achieves zero training error with probability (at least) $1 - \delta/2$. Conditioned on perfect training, the resulting classifier

$$y_\sharp\left(S_T(\rho)\right) = \operatorname{sign}\Big(\sum_{\ell=1}^{N} [\boldsymbol{\alpha}_\sharp]_\ell\, k^{(\mathrm{shadow})}\left(S_T(\rho_\ell), S_T(\rho)\right)\Big) \in \{\pm 1\} \tag{J64}$$

achieves, with probability (at least) $1 - \delta/2$,

$$\Pr_{(S_T(\rho), y(\rho))}\left[y_\sharp\left(S_T(\rho)\right) \neq y(\rho)\right] \leq \epsilon\ . \tag{J65}$$

The total probability of success is (at least) $1 - \delta$ and follows from a union bound over either desirable event failing. Theorem 1 is contingent on four core assumptions:

1. It must be possible to distinguish phases A and B by evaluating a well-conditioned analytical function on reduced r-body density matrices. The coefficients in the power series of the analytical function should also be bounded, but explicit knowledge is *not* necessary. This is the content of Assumption 1.

2. We use classical shadow raw data to read-in training data $(\rho_\ell \mapsto S_T(\rho_\ell))$ and process new states in the prediction phase $(\rho \mapsto S_T(\rho))$. We assume that each classical shadow arises from T randomized single-qubit Pauli measurements on independent state copies. The larger T, the more accurate these representations become. Theorem 1 requires $T \geq (512/3) d_p^2 C^2 12^r \left(r\left(\log(n) + \log(12)\right) + \log(N/\delta)\right) = \mathcal{O}\left(r 12^r d_p^2 C^2 \log(nN/\delta)\right)$. If r, C, d_p are constants, this resolution only scales polylogarithmically in system size n because N scales polynomially in n; see the next bullet point.

3. The training data size must not be too small either. We need to have a training data size N of order at least $(2rn)^{rd_p}\, d_p^{d_p} C^2 \exp(\exp(5)) \log(4/\delta)/\epsilon^2$. As long as r, C, d_p are constants (independent of system size n), this requirement simplifies to $N = \mathcal{O}\left(n^{rd_p} \log(1/\delta)/\epsilon^2\right)$. Hence, the number scales polynomially in system size n.

4. The squared margin constant also scales polynomially with system size n: $\Lambda^2 = 4\,(2rn)^{rd_p}\, d_p^{d_p} C^2 = \mathcal{O}\left(n^{rd_p}\right)$ if $r, C, d_p = \text{const}$. This is equivalent to demanding that the minimal margin $2/\Lambda$ scales inverse polynomially in system size n.

Corollary 1 does not only bound a hypothetical training error. The required shadow size T and training data size N both scale favorably in the number of qubits n. This also ensures that the numerical costs behind this procedure remain tractable for a wide range of system sizes. The costs associated with storage (classical shadows are sums of T elementary tensor products), training (can be reduced to a QCQP in N dimensions per Section J.1) and prediction (execute Formula (J8)) all scale polynomially in system size n, shadow size T, and training data size N.

Proof of Corollary 1. Again, we establish the claim for large, but finite-dimensional, approximations to the shadow kernel ($1 \leq d_p \ll D < \infty$ and $1 \leq r \ll R < \infty$). Fix $\delta \in (0,1)$ (probability of failure) and $\epsilon \in (0,1)$ (bound on average prediction error). Consider the data distribution $(S_T(\rho), y(\rho)) \sim \mathcal{D}$, the kernel $k^{(\mathrm{finite})}(\cdot,\cdot)$ – which obeys $k^{(\mathrm{finite})}\left(S_T(\rho), S_T(\rho)\right) \leq \exp\left(\exp(5)\right)$ – and a squared margin constant Λ^2 to be specified later. Assume $\Lambda^2 \exp(\exp(5)) \geq 1$ for simplicity (the other case is similar). Then, for training data size N, Theorem 6 asserts

$$\Pr_{(S_T(\rho), y(\rho)) \sim \mathcal{D}}\left[y_\sharp\left(S_T(\rho)\right) \neq y(\rho)\right] \leq \frac{1}{N} \mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) + 8\sqrt{\Lambda^2 \exp(\exp(5)) \frac{\log(4/\delta)}{N}}, \tag{J66}$$

with probability (at least) $1 - \delta/2$. Choosing N large enough allows us to suppress the second contribution beneath the desired approximation error bound:

$$N \geq 64 \Lambda^2 \exp(\exp(5)) \log(4/\delta)/\epsilon^2 \quad \Rightarrow \quad \Pr_{(S_T(\rho), y(\rho)) \sim \mathcal{D}}\left[y_\sharp\left(S_T(\rho)\right) \neq y(\rho)\right] \leq \frac{1}{N} \mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) + \epsilon, \tag{J67}$$

with probability (at least) $1-\delta/2$. Here, $\mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp)$ is the training error obtained from solving problem (J5a) for N independently sampled training data points $(S_T(\rho_1), y(\rho_1)), \ldots, (S_T(\rho_N), y(\rho_N)) \sim \mathcal{D}$. Proposition 5 asserts that this training error can vanish with high probability, provided that a well-conditioned analytical function on reduced r-body density matrices exists that can distinguish the phases (see Assumption 1). The classical shadow size T and the squared margin constant Λ^2 depend on the number of body r, the normalization constant C, and the truncation degree d_p of this classifier:

$$\left.\begin{array}{rcl} T & \geq & (512/3) d_p^2 C^2 12^r \left(r\left(\log(n)+\log(12)\right)+\log(N/\delta)\right) \\ \Lambda & \geq & 4\,(2rn)^{rd_p}\, d_p^{d_p} C^2 \end{array}\right\} \quad \Rightarrow \mathrm{E}_{\mathrm{tr}}(\boldsymbol{\alpha}_\sharp) = 0 \tag{J68}$$

with probability (at least) $1-\delta/2$. The claim now follows from inserting this squared margin size into the expression (J67) for training data size. □

APPENDIX K: Classifying SPT phases with $O(2)$ symmetry using a few-body observable

K.1. Symmetry-protected topological phases

We consider a scenario similar to that of Section F.5, namely, a family of Hamiltonians $H(x)$ parameterized by x. We additionally enforce that $H(x)$ be invariant under certain symmetry transformations, which can include tensor products of on-site rotations, "spatial" transformations permuting the sites, or antiunitary maps characterizing time-reversal. These additional symmetry constraints allow for a fine-grained characterization of $H(x)$ into various symmetry-protected topological (SPT) phases. Removing said constraints reduces this characterization to the coarser one involving purely topological phases. Similar to the coarser characterization, ground states of $H(x)$ remain in a particular SPT when the parameters x are varied continuously, as long as the spectral gap of the Hamiltonian remains finite. In other words, the gap has to close at some x in order for the ground states to transition into another phase. When there is a constant spectral gap, it is expected that an operator acting on a local region larger than some constant size independent of the full system size n can classify different SPT phases. The existence of a classifying function of local density matrices has been rigorously established for a handful of cases: $U(1)$-symmetric systems in two dimensions (either noninteracting fermionic [101, 188] or interacting [12, 80, 96]), and certain spin-1 chains in one dimension [14, 161, 162].

SPT phases of one-dimensional spin chains with unique ground states, symmetric under tensor-product unitaries forming a symmetry group G, are in one-to-one correspondence with the various projective representations realized by G [38]. Projective representations are those in which the group's multiplication table is decorated with phases in a way that is consistent with associativity [11]. A genuine (i.e., linear) representation corresponds to the unique trivial projective representation.

Consider, for example, spin chains symmetric under $G = SO(3)$. This group admits two distinct classes of projective representations: one class corresponds to integer spin, and one corresponds to half-integer spin. Thus, there are two different phases for such chains — the trivial phase and the "Haldane phase" [38, 77].

Relaxing the symmetry group down to its $O(2)$ subgroup maintains the two-phase classification, because $O(2)$ also admits two projective representations [37]. In fact, one can relax the symmetry all the way down to the simplest dihedral subgroup $Z_2 \times Z_2$ [71, 137]; such a classification is similar to that of the model in Appendix D.4. We investigate systems admitting the larger $O(2)$ symmetry below, noting that the work we rely on [161, 162] also studies symmetry groups that include spatial inversion and time reversal.

K.2. $O(2)$-symmetric qutrit spin chains

The representative states for each of the two $O(2)$-symmetric phases for qutrit spin chains are the product state, representing the trivial phase, and the valence-bond-solid (VBS) state [4], admitting a projective representation of the symmetry [162] and thus representing the Haldane phase. It has long been known that the expectation value of a nonlocal "twist" operator O_L [123, 166] distinguishes these two representative states: $\mathrm{sign}(\langle O_L \rangle)$ is $+1$ for the product state, and -1 for the VBS state. We will see later that, by continuity arguments, this sign will stay constant for other states within the same phase.

In order to work efficiently, our phase classification algorithms require a *local* operator whose expectation value (a) has the same sign as that of O_L; and (b) is above or below a margin (here, $1/2$), in order to determine the required accuracy of the classical shadows. Recently, criterion (a) was explicitly demonstrated by Tasaki [161, 162] using a local version O_ℓ of the twist, see Eq. (K5) below. We collect relevant parts of his results to prove both criteria in the theorem below. Due to the existence of a local operator for classifying the SPT phases, our ML algorithms are guaranteed to predict the SPT phases accurately based on the proof given in Appendix J.

Theorem 8. *Consider the triple* $\{H(x), |\psi(x)\rangle, \Delta(x)\}$ *containing* $(2L+2)$*-site spin-one chains with periodic boundary conditions*

$$H(x) = \sum_{j=-(L-r)}^{L-r+1} h_j(x) + h_{-L}(x) + h_{L+1}(x) \tag{K1}$$

that admit corresponding unique ground states $|\psi(x)\rangle$ *and spectral gaps* $\Delta(x) \geq \gamma = \Omega(1)$*, bounded interaction strength* $\|h_j(x)\|_\infty \leq R = \mathcal{O}(1)$*, and whose terms* $h_j(x)$ *are supported on sites* k *such that* $|j-k| \leq r = \mathcal{O}(1)$*. Assume that* $H(x)$ *is* $O(2)$*-symmetric, with the symmetry group generated by*

1. *a collective* z*-axis rotation by any angle, and*
2. *an* x*-axis rotation by* π*.*

There exists a few-body observable A*, such that for all* x*, we have*

$$\operatorname{sign}(\langle\psi(x)| A |\psi(x)\rangle) = \operatorname{sign}(\langle\psi(x)| O_L |\psi(x)\rangle), \quad \text{as well as} \tag{K2a}$$

$$|\langle\psi(x)| A |\psi(x)\rangle| \geq 1/2\ . \tag{K2b}$$

Proof. We use spin-one operators $S^{(\alpha)}$ with $\alpha \in \{x, y, z\}$ that have eigenvalues $\{0, \pm 1\}$ and satisfy angular-momentum commutation relations $[S^{(x)}, S^{(y)}] = \mathrm{i}S^{(z)}$. Eigenstates of $S^{(z)}$ are denoted by $|\sigma\rangle$ with $\sigma \in \{0, \pm 1\}$. A rotation around axis α is a unitary operator generated by the corresponding $S^{(\alpha)}$. The two symmetry group generators are, for $\theta \in [0, 2\pi)$,

$$U(\theta) = \bigotimes_{j=-L}^{L+1} e^{-\mathrm{i}\theta S_j^{(z)}} \qquad \text{and} \qquad V = \bigotimes_{j=-L}^{L+1} e^{-\mathrm{i}\pi S_j^{(x)}}\ . \tag{K3}$$

By assumption, both symmetries commute with each Hamiltonian term h_j; we will explicitly use both to prove the theorem. We will also need superimposed versions $S^{(\pm)} = S^{(x)} \pm \mathrm{i}S^{(y)}$, which satisfy

$$e^{\mathrm{i}\phi S^{(z)}} S^{(\pm)} e^{-\mathrm{i}\phi S^{(z)}} = S^{(\pm)} e^{\pm i\phi}\,. \tag{K4}$$

The family of unitary twist operators [5], acting on an interval of 2ℓ spins centered at the origin, is

$$O_\ell = \bigotimes_{k\,,\,\left|k-\frac{1}{2}\right| \leq \ell+\frac{1}{2}} \exp\left(-\mathrm{i}2\pi \frac{k+\ell}{2\ell+1} S_k^{(z)}\right)\ . \tag{K5}$$

Each site's rotation is by a multiple of $2\pi/(2\ell+1)$ that is proportional to the site index, forming the namesake twist pattern. The $\ell = L$ case reduces to the aforementioned nonlocal twist operator O_L, while $\ell \ll L$ are its local versions.

Suppressing x dependence, the key property is that the twisted ground state $O_\ell|\psi\rangle$ has energy close to that of the ground state. In particular, there exists $C_0, C_1 > 0$, such that for all $\ell \geq C_0$, Lemma 12 below yields

$$\langle\psi|O_\ell H O_\ell^\dagger|\psi\rangle - \langle\psi|H|\psi\rangle \leq \frac{C_1}{\ell}\,. \tag{K6}$$

The ground state is unique by our assumption of a gap, so the twisted ground state must then become proportional to the ground state as $\ell \to \infty$. In other words, the magnitude of their overlap must be close to one as long as $\ell \geq C_0$,

$$|\langle\psi|O_\ell|\psi\rangle|^2 \geq 1 - \frac{C_1}{\Delta\ell}\,; \tag{K7}$$

see Lemma 13 below. The phase of this overlap is either 0 or π because the π-rotation V leaves the ground state invariant:

$$\langle\psi|O_\ell|\psi\rangle = \langle\psi|V^\dagger O_\ell V|\psi\rangle = \langle\psi|O_\ell^\dagger|\psi\rangle = \overline{\langle\psi|O_\ell|\psi\rangle} \in \mathbb{R}\,. \tag{K8}$$

Hence, the few-body Hermitian observable $A = (O_\ell + O_\ell^\dagger)/2$ with $\ell = \max(4\gamma/(3C_1), C_0)$ satisfies

$$|\langle\psi|A|\psi\rangle| = |\langle\psi|O_\ell|\psi\rangle| \geq \sqrt{1 - \frac{C_1}{\Delta\ell}} \geq \frac{1}{2}\ , \tag{K9}$$

proving Eq. (K2b). Note that the required value of ℓ depends on the gap, and thus also on x.

To prove Eq. (K2a), we need to show that the sign of the twist's expectation value remains the same for any $\ell \geq \max(4\gamma/(3C_1), C_0)$. To do this, first notice that, when ℓ is relaxed to be a nonnegative real, the twist (K5) is *continuous* in ℓ. (This can be verified, e.g., by studying the twist's eigenvalues.) Continuity implies that the expectation value cannot change sign; otherwise, it would have to cross zero, thus violating Eq. (K9). Therefore, the sign remains the same, confirming Eq. (K2a). Similarly, by continuity in ℓ and x, the expectation value maintains its sign within each phase. □

The above argument is contingent on two auxiliary statements, which we now prove.

Lemma 12 (Vanishing energy difference [161]; Eq. (K6))**.** *For constants C_0, C_1, as long as $\ell \geq C_0$, we have*

$$\langle\psi|O_\ell H O_\ell^\dagger|\psi\rangle - \langle\psi|H|\psi\rangle \leq \frac{C_1}{\ell}. \tag{K10}$$

Proof. Using the variational principle (which says that the difference in energy between any state and the ground state is nonnegative), plugging in O_ℓ and H, applying $\langle\psi|O|\psi\rangle \leq \|O\|_\infty$, and distributing the norm over the sum yields

$$\begin{aligned}
\langle\psi|O_\ell H O_\ell^\dagger|\psi\rangle - \langle\psi|H|\psi\rangle &\leq \langle\psi|\left(O_\ell H O_\ell^\dagger + O_\ell^\dagger H O_\ell - 2H\right)|\psi\rangle && \text{(K11a)}\\
&= \sum_{j=-(\ell+r)}^{\ell+r+1} \langle\psi|\left(O_\ell h_j O_\ell^\dagger + O_\ell^\dagger h_j O_\ell - 2h_j\right)|\psi\rangle && \text{(K11b)}\\
&\leq \sum_{j=-(\ell+r)}^{\ell+r+1} \left\|O_\ell h_j O_\ell^\dagger + O_\ell^\dagger h_j O_\ell - 2h_j\right\|_\infty && \text{(K11c)}
\end{aligned}$$

Next, we use the finite support and rotational invariance of h_j from Eq. (K3) to rotate the twist O_ℓ,

$$\begin{aligned}
O_\ell h_j O_\ell^\dagger &= O_\ell U(\theta_j)\, h_j U^\dagger(\theta_j)\, O_\ell^\dagger && \text{(K12a)}\\
&= \left(\bigotimes_{|k-j|\leq r} e^{-\mathrm{i}\left(\frac{2\pi}{2\ell+1}[k+\ell]+\theta_j\right)S_k^{(z)}}\right) h_j \left(\bigotimes_{|k-j|\leq r} e^{\mathrm{i}\left(\frac{2\pi}{2\ell+1}[k+\ell]+\theta_j\right)S_k^{(z)}}\right), && \text{(K12b)}
\end{aligned}$$

where we pick $\theta_j = -\frac{2\pi}{2\ell+1}(j+\ell)$ for each j. That way, the twist does not affect site j, with

$$O_\ell h_j O_\ell^\dagger = e^{\mathrm{i}\frac{2\pi}{2\ell+1}M_j} h_j e^{-\mathrm{i}\frac{2\pi}{2\ell+1}M_j}, \qquad \text{and} \qquad M_j = \sum_{|k-j|\leq r} (j-k)\, S_k^{(z)}. \tag{K13}$$

We now expand h_j as a polynomial in $\{S_k^{(z)}, S_k^{(\pm)}\}$. This can be done because products of powers of these operators form a matrix basis for any operator on the chain. For a single site, the set $\{S^{(z)}S^{(\pm)}, (S^{(+)})^2\}$, along with their complex conjugates and some powers of $S^{(z)}$, form the basis of nine matrix units for all 3×3 operators on the site. Tensor products of these operators therefore form a matrix-unit basis for all sites. The conjugation property (K4) and Eq. (K13) imply that each term in the expansion of h_j, upon conjugation by O_ℓ, will be imparted with a phase that is some multiple μ of $2\pi/(2\ell+1)$. Combining all terms with the same phase into $h_{j,\mu}$, we have

$$e^{\mathrm{i}\frac{2\pi}{2\ell+1}M_j} h_{j,\mu} e^{-\mathrm{i}\frac{2\pi}{2\ell+1}M_j} = h_{j,\mu} e^{\mathrm{i}\frac{2\pi}{2\ell+1}\mu}. \tag{K14}$$

Moreover, $|\mu| \leq 2\mu_{\max}$, where $\mu_{\max} = \sum_{|k-j|\leq r} |j-k| = r(r+1)$ is the largest eigenvalue of M_j. Plugging this in and expanding the resulting cosine yields

$$\begin{aligned}
\left\|O_\ell h_j O_\ell^\dagger + O_\ell^\dagger h_j O_\ell - 2h_j\right\|_\infty &= 2\left\|\sum_{|\mu|\leq 2r(r+1)} \left[\cos\left(\frac{2\pi}{2\ell+1}\mu\right) - 1\right] h_{j,\mu}\right\|_\infty && \text{(K15a)}\\
&\leq \left(\frac{2\pi}{2\ell+1}\right)^2 \sum_{|\mu|\leq 2r(r+1)} \mu^2 \|h_{j,\mu}\|_\infty. && \text{(K15b)}
\end{aligned}$$

Since the spin operators form a matrix-unit basis, each $h_{j,\mu}$ is simply h_j with some entries removed. Therefore, the norm of $h_{j,\mu}$ is bounded by R. Applying that and performing the remaining sum (K11c) over j yields

$$\langle\psi|O_\ell H O_\ell^\dagger|\psi\rangle - \langle\psi|H|\psi\rangle \leq \frac{\ell+r+1}{(2\ell+1)^2} 4\pi^2 R \left(\sum_{|\mu|\leq 2r(r+1)} \mu^2 \right) . \tag{K16}$$

Thus, for $\ell \geq C_0$, the difference in energies between the ground state and twisted ground state will be bounded by C_1/ℓ, where C_0, C_1 are two constants depending on the interaction range r and norm bound R of the Hamiltonian terms. □

Lemma 13 (High overlap [162]; Eq. (K7))**.** *For constants C_0, C_1, as long as $\ell \geq C_0$, we have*

$$|\langle\psi|O_\ell|\psi\rangle|^2 \geq 1 - \frac{C_1}{\Delta\ell} . \tag{K17}$$

Proof. All eigenvalues of H are bounded below by the sum of the ground state energy $E_{\text{gnd}} = \langle\psi|H|\psi\rangle$ and spectral gap Δ,

$$H \geq E_{\text{gnd}}|\psi\rangle\langle\psi| + (E_{\text{gnd}} + \Delta)(\mathbb{I} - |\psi\rangle\langle\psi|) = E_{\text{gnd}}\mathbb{I} + \Delta(\mathbb{I} - |\psi\rangle\langle\psi|) . \tag{K18}$$

Conjugating by O_ℓ and evaluating the result in the ground state yields

$$\langle\psi|O_\ell H O_\ell^\dagger|\psi\rangle \geq E_{\text{gnd}} + \Delta\left(1 - |\langle\psi|O_\ell|\psi\rangle|^2\right) . \tag{K19}$$

Rearranging this and plugging in Lemma 12 yields the desired result. □