Quantum kernels to learn the phases of quantum matter

Teresa Sancho-Lorente, Juan Román-Roche, and David Zueco
Instituto de Nanociencia y Materiales de Aragn (INMA), CSIC–Universidad de Zaragoza, Zaragoza 50009, Spain

(Received 4 October 2021; accepted 6 April 2022; published 25 April 2022)

Classical machine learning has succeeded in the prediction of both classical and quantum phases of matter. Notably, kernel methods stand out for their ability to provide interpretable results, relating the learning process with the physical order parameter explicitly. Here we exploit quantum kernels instead. They are naturally related to the *fidelity*, and thus it is possible to interpret the learning process with the help of quantum information tools. In particular, we use a support vector machine (with a quantum kernel) to predict and characterize second-order quantum phase transitions. We explain and understand the process of learning when the fidelity per site (rather than the fidelity) is used. The general theory is tested in the Ising chain in transverse field. We show that for small-sized systems, the algorithm gives accurate results, even when trained away from criticality. Besides, for larger sizes we confirm the success of the technique by extracting the correct critical exponent ν. Finally, we present two algorithms, one based on fidelity and one based on the fidelity per site, to classify the phases of matter in a quantum processor.

DOI: 10.1103/PhysRevA.105.042432

I. INTRODUCTION

It is suggestive to merge quantum computing and machine learning (ML), looking for their constructive combination in hopes of increasing the number of problems that can be solved in the near future. Both are disruptive technologies that cross the boundaries of current computational capabilities. *Classical* ML has found several applications for optimizing tasks in quantum information processing. Some examples are quantum state tomography [1,2], quantum gate optimization [3–5], and ground-state estimation [6], among others. See the reviews [7,8]. *Quantum* machine learning (QML), instead, seeks to extend the algorithms of classical ML to be run in a quantum computer [9–11]. An incomplete list of reported examples includes the quantum versions of neural networks [12], principal component analysis [13], classification [14–18], or support vector machines [19]. The key question here is whether they have any kind of advantage over their classical counterparts [20–23].

Both in classical and quantum ML, input data is encoded in M vectors $\boldsymbol{x}_j$, $j = 1, \dots, M$ of dimension d. They are processed in different ways, depending on the chosen algorithm and/or type of learning. In this work we are interested in *kernel* methods [24]. Here the learning is based on the kernel, which is the inner product of these vectors $K_{ij} = \boldsymbol{x}_i \cdot \boldsymbol{x}_j$ or, more generally, the inner product defined in a *feature space*. The latter is given by a nonlinear map $\boldsymbol{x}_j \to \boldsymbol{\Phi}(\boldsymbol{x}_j)$. Thus $K_{ij} = \boldsymbol{\Phi}(\boldsymbol{x}_i) \cdot \boldsymbol{\Phi}(\boldsymbol{x}_j)$. To understand this, we can think of a classification task where the data is split into two classes. The feature map should clump data belonging to the same class while dispersing data from different classes, so that the resulting distribution is separable. The kernel defines distances on the feature space on which classification takes place.

In QML, data is loaded in a quantum computer; thus the first step is *encoding* it onto a quantum state:

$$\mathbf{x}_j \to |\psi(\mathbf{x}_j)\rangle = U_\theta(\mathbf{x}_j)|0\rangle. \tag{1}$$

Here $|0\rangle$ is the initial state [25] and θ a set of parameters that, eventually, can be optimized. Thus the quantum kernel is given by [19,26–28]

$$K_{ij}^{(\theta)} = |\langle\psi_\theta(\mathbf{x}_j)|\psi_\theta(\mathbf{x}_{j'})\rangle| = |\langle 0|U_\theta^\dagger(\mathbf{x}_i)\, U_\theta(\mathbf{x}_j)|0\rangle|. \tag{2}$$

Mapping (1) offers a quantum advantage if the quantum circuit is difficult to simulate on a classical computer and provides a better performance than classical maps. It is not trivial to obtain instances of quantum advantage. A heuristic approach, implementing entangled maps that are shown to be classically hard, has been followed in [29,30]. From these seminal works, attempts to (rigorously) determine under what conditions q kernels are superior have been discussed [31,32]. Finally, a quantum speed-up has been shown for the task of classifying integers according to the so-called discrete logarithm problem [33]. This is quite a formal problem, so the challenge posed in the first section of this article persists—the identification of tasks of *practical* use where QML is advantageous.

One possible shortcut to achieving the goal is to consider quantum data. The idea is simple: generating the data already requires a quantum computer, and the step of loading classical data onto a quantum RAM is skipped. The task we propose is classifying the phases of matter. For classical models, classical ML techniques have been discussed with both kernel methods [34–36] and beyond [37–44]. For quantum models, neural networks trained with different observables [10,45–47], among other ML techniques [48–50] or, even experimentally, with a quantum simulator [51], have been used to classify phases in strongly correlated electron systems. Here we propose to use quantum kernels as (2) [52,53]. Then the classification is done with a support vector machine (SVM). Borrowing from quantum information, we argue that by employing the fidelity (or related measures) as a kernel, the classification can be interpreted to extract the phase boundary [54–57]. We show that the machine predicts the critical point and that it is capable of learning the critical exponents. Remarkably, it does so despite being trained with samples far from the critical point. Here, by way of illustration we use the one-dimensional Ising model in a transverse field. This is an exactly solvable model, but our arguments are pretty general. In fact, we present two algorithms to classify the quantum states of matter in the general scenario where the ground states are computed in a quantum processor. To be concrete, we discuss the use of a variational quantum eigensolver to find the ground state [58,59].

We overview the rest of the paper here. In the next section, Sec. II, the theory of fidelity-based characterization of quantum phase transitions (QPTs) is summarized following the seminal works of Zanardi. Then, in Sec. III we sketch the idea behind SVMs and the kernel trick. Quantum kernels are introduced. Importantly, in Sec. III we discuss the process of learning and how the machine learns to characterize a second-order QPT, which explains the results of the rest of the paper. Our general theory is tested in the quantum Ising model in one dimension in Sec. IV. We also present two algorithms to implement the ideas of this paper in a quantum computer, see Sec. V. The paper ends, as usual, with the conclusions in Sec. VI. Some identities for quantum states, used throughout the paper, can be found in the Appendix.

II. QUANTUM PHASE TRANSITIONS AND FIDELITIES

Consider a Hamiltonian $H(J)$ such that at $J = J_c$ the system undergoes a quantum phase transition. Whether first, second order, or topological, the QPT can be studied from the *distinguishability* between ground states. Following the original idea of Zanardi and Paunković, a measure of this distinguishability is the fidelity between two ground states,

$$F(J, J') := |\langle\psi_0(J)|\psi_0(J')\rangle|. \tag{3}$$

In a nutshell, and considering $F(J, J+\epsilon)$ with ϵ small enough, only at the critical point (or close enough) is F expected to deviate from 1. Thus, an abrupt change in the fidelity signals criticality [54,60] (see [61] for a review). Following this idea, Zhou and co-workers argue in terms of renormalization [55–57,62]. This is specially useful for continuous QPTs, occurring at the thermodynamic limit. The rest of this article assumes this type of QPT.

In what follows we find it convenient to discuss general expressions for the fidelity between two quantum states using the matrix product state (MPS) formalism. This allows us to anticipate the N dependence of the fidelity and to introduce a new distance measure between ground states that will be used throughout this work. In the case of a translational invariant lattice with local dimension l, the ground state can be written in its MPS form [63], $|\psi_0(J)\rangle = \mathrm{Tr}[A_{i_1}...A_{i_N}]|i_1,\dots,i_N\rangle$. Here, $\{A_{i_j}\}$ are $D\times D$ matrices that depend on the local quantum number $i_j = 1,\dots,l$. D is the bond dimension, which is related to the amount of entanglement contained in the state. Using the MPS representation, the fidelity takes the convenient form [60] (see also the Appendix):

$$F(J, J') = \sum_{k=1}^{D^2} \lambda_k(J, J')^N \overset{N\gg 1}{\cong} \lambda_1(J, J')^N. \tag{4}$$

Here λ_k are the eigenvalues of the transfer matrix $E(J, J') = \sum_{i=1}^{l} A_i(J)\otimes A_i(J')$. The state is normalized so $|\lambda_k| \leqslant 1$. If we denote λ_1 the largest (in absolute value) of these eigenvalues, the second equality is obvious and motivates the definition of the *fidelity per site*:

$$\ln[\lambda(J, J')] := \ln F[(J, J')]/N. \tag{5}$$

Importantly, it inherits the properties of F, thus being a distance measure fully characterizing the QPT. It also has an important advantage (over F). In (4) the orthogonality catastrophe is explicit. For N large enough, two ground states with $J \neq J'$ have exponentially small fidelity, regardless of whether or not they belong to the same phase. This points to the failure of F as a distance measure to resolve the transition. Using (4) and (5) we note that $\lambda_1 = \lim_{N\to\infty}\lambda$, i.e., λ_1 is a scale factor independent of size. The use of the fidelity per site as a measure prevents the orthogonality catastrophe.

III. PHASE CLASSIFICATION AND SVMS

Identifying the phases in quantum many-body systems can be formulated as a classification task. To simplify the discussion, let us assume that the system has two phases separated at $J = J_c$. Classifying means assigning a label $y_J = \pm 1$ to every ground state $|\psi_0(J)\rangle$ depending on $J \gtrless J_c$, respectively. In this work, the training data are the ground states themselves, i.e., $\mathbf{x}_j = |\psi_0(J_j)\rangle$. In other words, we choose a set $\{J_j\}$ and compute the corresponding ground state, presumably in a quantum computer. This is the training set, used in a supervised learning algorithm for classifying the data. In this paper we use a support vector machine. This algorithm finds the hyperplane that optimally splits the data in two, given a training set [24]. It turns out that the hyperplane is found by minimizing a Lagrangian, $L(\alpha) = \sum\alpha_j - \frac{1}{2}\sum\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$, that depends on the kernel, with constraints $\sum\alpha_j y_j = 0$. The α_j's are the Lagrange multipliers found in the Lagrangian minimization. Only a subset of them will be non-null, attending to Eq. (6). These determine the classification, i.e., define the separating hyperplane. They are termed support vectors (SVs). Then, given a ground state $|\psi_0(J)\rangle$, its signed "distance" to the hyperplane is given by

$$d(J) = \sum_{j=1}^{M} \alpha_j y_j K(J, J_j) + b, \tag{6}$$

where b is the offset parameter (we use here the standard notation). It is given by

$$b = \frac{1}{M}\sum_{i,j} \alpha_i y_i K(x_i, x_j) - y_j. \tag{7}$$

If the optimization is successful, the separating hyperplane will lie at the phase boundary between the two phases and new data points will be classified attending to the sign of their "distance" to the hyperplane.

The crux of the matter is the kernel matrix. The better it measures the similarity between the data points the more it facilitates classification. Based on our fidelity discussion, we are going to consider two kernels that, as argued above, can measure the distance between quantum states and thus are useful to discriminate different phases. Using Eqs. (2), (3), and (5) we can define the following:

$$K^{(F)}(J_i, J_j) := F(J_i, J_j), \tag{8}$$

$$K^{(\lambda)}(J_i, J_j) := \lambda(J_i, J_j). \tag{9}$$

We expect that the fidelity-based kernel $K^{(F)}$ will fail for a sufficiently large system size N (orthogonality catastrophe). In this case the kernel is "overfitted" (see also Refs. [52,64]). Therefore it seems that it cannot fully characterize QPTs at the thermodynamic limit. However, below we show that $K^{(\lambda)}$ learns the QPT. This will be complemented (in Sec. IV) with numerical simulations where we will also test the failure of $K^{(F)}$ as the system size grows.

What does the SVM learn?

In this section we argue that, under reasonable conditions (to be explained), *the SVM learns the critical J_c in a second-order QPT using the kernel* $K^{(\lambda)}(J_i, J_j) := \lambda(J_i, J_j)$, Eq. (9). However, some remarks are necessary. We assume that the data is separable. As one sweeps (in values of J) across the phase transition, for every new J the ground state $|\phi(J)\rangle$ is increasingly further (fidelity closer to zero) from any given ground state in the other phase. In other words, there is a direction in the Hilbert space endowed with distance $\lambda(J_i, J_j)$ such that the projection of the ground states $|\phi(J)\rangle$ along this direction is a strictly monotonic function of J. This is (by definition) what happens in a phase transition when we look at it in parameter space: phases are separated by a clear boundary, i.e., the critical point. What we are assuming is that this intuitive property applies as well to the Hilbert space when the distance $\lambda(J_i, J_j)$ is used. This view is supported by the results in [54,56,61]. A separable set allows us to train the SVM with a hard margin, which will have consequences for the result of the learning process, mainly that, barring a failure of the learning process due to a deficient kernel, we will find only two SVs—one on each side of the separating hyperplane. Let us denote by $|\psi(J_\pm)\rangle$ the SVs on the right and left side of the transition (respectively). In addition, the ground state located at the separating hyperplane, $d(\widetilde{J}_c) = 0$, is $|\psi(\widetilde{J}_c)\rangle$. Consequently, if $\widetilde{J}_c = J_c$, the SVM has learned the critical point.

Let us first prove that for QPTs and using $K^{(\lambda)}$, $b = 0$ in (6). We use that only SVs have a non-null multiplier, $\alpha_j \neq 0$, and that they are equidistant to the separating hyperplane, $\lambda(J_+, \widetilde{J}_c) = \lambda(J_-, \widetilde{J}_c)$. Therefore

$$b = \frac{\lambda(J_+, \widetilde{J}_c)}{2}(\alpha_+ - \alpha_-) = 0.$$

Here we have used the constraint $\sum \alpha_j y_j = 0$, which implies $\alpha_+ = \alpha_-$. If $b = 0$, then $d(\widetilde{J}_c) \propto \lambda(J_+, \widetilde{J}_c) - \lambda(J_+, \widetilde{J}_c) = 0$. To show how the intersection of the two curves $\lambda(J_\pm, J)$ relates to the QPT, we plot Fig. 1. It shows a generic situation of how two fidelity curves intersect. It is a calculation using the quantum Ising model, to be discussed below, but other models with a second-order QPT show the same phenomenology. Starting from $\lambda(J_+, J_+) = 1$, the fidelity remains close to 1. As $J \to J_c$, see Sec. II, there is a nonanalyticity in $\lambda(J_+, J)$, with a sudden increase in the slope. This is the signature of the transition. The same occurs for $\lambda(J_-, J)$. For our purposes here, it means the intersection of the two curves occurs in the vicinity of J_c. The nonanalyticity corresponds to the point where $\partial_x \lambda(J_\pm, x)$ is maximum. In fact, a way of finding QPTs is by looking at this maximum, i.e., defining $J_c(N)$ as the point where the derivative is maximum, then $J_c = \lim_{N\to\infty} J_c(N)$. Since the slope $\partial_{x=J_c(N)}\lambda(J_-, x)$ grows with N, we find that the larger the N the closer the intersection moves towards $J_c(N)$, i.e., the closer $\widetilde{J}_c(N)$ to $J_c(N)$. In that case, the SVM learns the transition point as predicted by the fidelity theory.

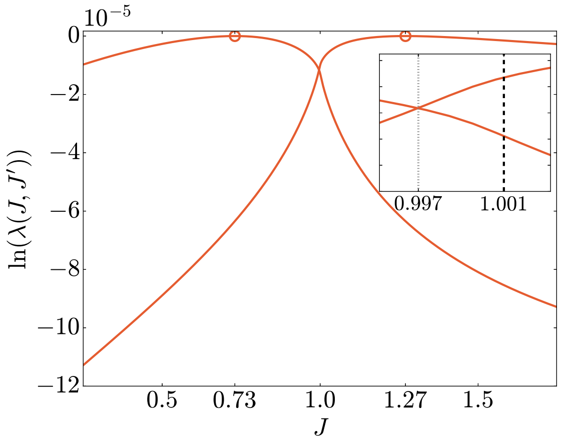


FIG. 1. Critical point from the SVM. $\lambda(J_\pm, J)$ in the one-dimensional quantum Ising model, see Eq. (10), with $J_- = 0.73$ and $J_+ = 1.27$ for $N = 1000$ spins. J has units of energy ($\hbar = 1$). $J_\pm$ are represented by the open circles. Inset: Zoom of the intersection. The gray dotted line points out the intersection between the two curves, i.e., the J at which $d = 0$, and so the boundary predicted by the SVM, $\widetilde{J}_c(N)$. The maximum of the derivative $\partial_x \lambda(J_+, x)$ is marked with the black dashed line, $J_c(N)$.

IV. APPLICATION: THE QUANTUM ISING MODEL

We consider the one-dimensional quantum Ising model in a transverse field with Hamiltonian ($\hbar = 1$ through this paper)

$$H(J) = \sum_{i=1}^{N} \sigma_j^z - J \sum_{i=1}^{N} \sigma_j^x \sigma_{j+1}^x. \tag{10}$$

σ_j^α are the Pauli matrices acting on the j-lattice site. The lattice size is N. Periodic boundary conditions (PBCs) are considered. This Ising model has a second-order phase transition occurring at $J_c = 1(-1)$ in the $N \to \infty$ limit. For $J_c > 1(J_c < -1)$ the $\mathbb{Z}_2$ (parity) symmetry is spontaneously broken and the ground state is ferromagnetically (antiferromagnetically) ordered. Without loss of generality, we fix our attention on the paramagnetic-ferromagnetic transition occurring at $J_c = 1$.

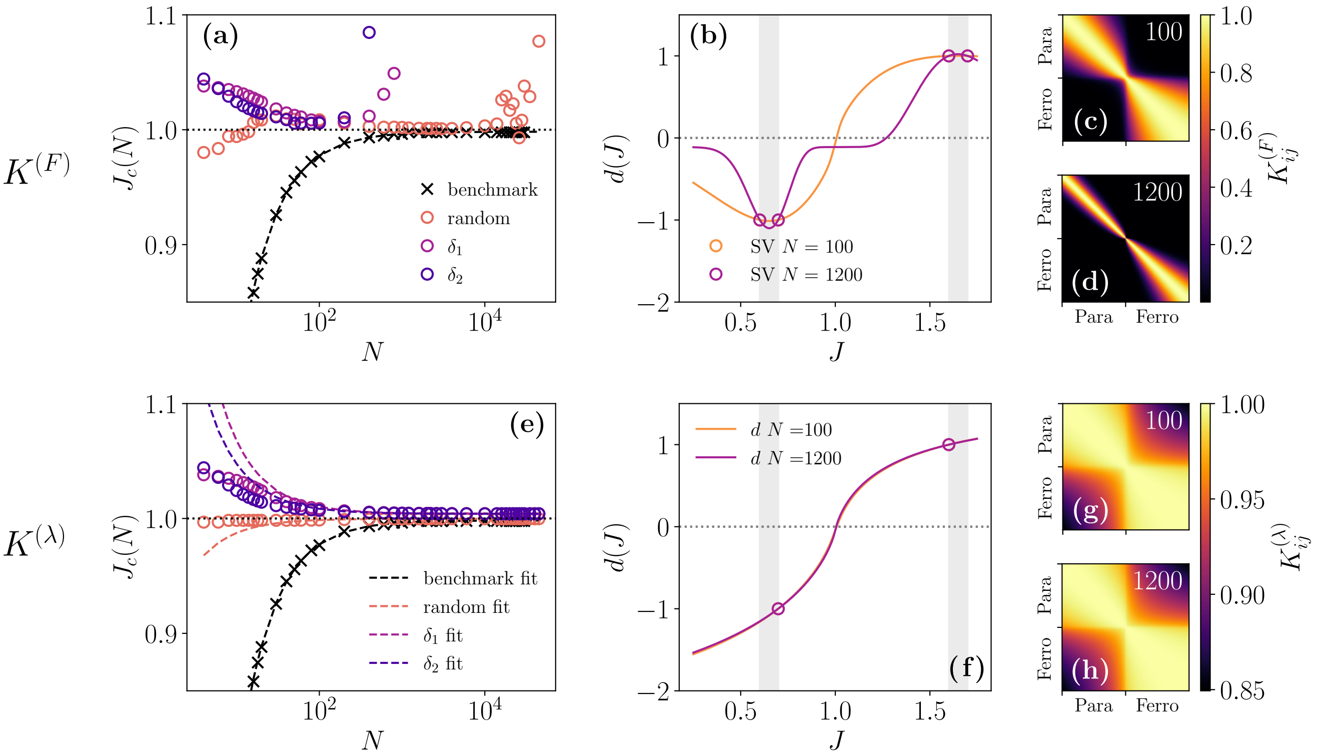


FIG. 2. Quantum kernels learning phases of matter. (a) $J_c(N)$ using the kernel $K^{(F)}$ for the three trainings discussed in this work. Namely, by intervals taking points failing in the Δ subsets $\delta_1 = [0.8, 0.9] \cup [1.2, 1.3]$, $\delta_2 = [0.6, 0.7] \cup [1.6, 1.7]$, and taking $M = 133$ points randomly distributed over Δ (random). The set Δ are 1000 ground states taking Js equally spaced in $\Delta = [0.25, 1.75]$. J has units of energy ($\hbar = 1$). Open circles are SVM results using the class `sklearn.svm.SVC` from Sklearn [65]. The crosses stand for a finite-size scaling analysis following [56]. The corresponding dashed lines are fittings (see Table I). (b) Distance function, Eq. (6), for two sizes N. The open circles are the SVs and the shaded (gray) zones mark the training interval, δ_2. (c, d) Contour plots of the matrix $K_{ij}^{(F)}$ where i, j run on the total Δ. The labels mark the (theoretical) phases of the model for J_i (J_j). Lower row, panels (e), (f), (g), and (h), are the counterparts but using the kernel $K^{(\lambda)}$.

For our interest here, the QPT has been discussed in terms of the ground-state fidelity $F(J, J')$ and the fidelity per site $\lambda(J, J')$, Eqs. (3) and (5) [54,56,61]. Thus it is an ideal testbed for our proposal. Besides, it is exactly solvable via the Jordan Wigner (JW) transformation, so an explicit formula for F between two ground states can be found:

$$F(J_i, J_j) = \prod_k |\cos\left(\theta_k(J_i) - \theta_k(J_j)\right)|. \tag{11}$$

Here,

$$\cos\left(2\theta_k(J_j)\right) = \frac{1 + 2J_j \cos k}{\sqrt{1 + 4J_j \cos k + 4J_j^2}}. \tag{12}$$

Considering even N and PBC, $k = (2n - 1)\pi/N$ with $n = 1, \ldots, N/2$.

With (11) at hand, kernels (8) and (9) can be computed. In this paper, the data sets are obtained by taking 1000 J's equally spaced in the range $\Delta = [0.25, 1.75]$. For the training set we explore three possibilities. Two consist of training with J's belonging only to specific intervals. From Δ we take the points falling in the subsets $\delta_1 = [0.8, 0.9] \cup [1.2, 1.3]$ and $\delta_2 = [0.6, 0.7] \cup [1.6, 1.7]$. Notice that they are nonsymmetric (around $J_c = 1$) in order to challenge our algorithm abilities in the classifying task. Besides, and importantly enough, they are far away from the critical value. The third training set is formed by taking $M = 133$ points randomly distributed over the whole set Δ [66].

Our main results are summarized in the different panels of Fig. 2. The top row of panels contains results for the fidelity kernel $K^{(F)}$ (8). They are compared to the results from the λ kernel $K^{(\lambda)}$ (9) (bottom row). In panel (a) we plot the predicted $\widetilde{J}_c(N)$, using the kernel $K^{(F)}$. This is obtained from the distance function (6) after interpolating the point that fulfills $d(\widetilde{J}_c(N)) \overset{!}{=} 0$. We do it for the three training sets (empty

TABLE I. J_c and ν critical exponent results. The numbers are obtained by fitting the points in Fig. 2(e) as a function of N to the function $J_c(N) = J_c + a N^{-\nu}$ with fitting parameters a, J_c, and ν. $J_c(N)$ are given by the SVM algorithm. They are compared to the procedure developed in [56] based on the fidelity per site, λ (see main text). The $(*)$ means that the error given by the fitting is smaller than 10^{-8}. Other error sources, such as the number of training data or the Δ discretization, limit the accuracy.

	J_c	ν
λ as [56]	0.99827(12)	0.966(17)
SVM (δ_1)	1.00414 $(*)$	0.974(60)
SVM (δ_2)	1.00404 $(*)$	1.003(75)
SVM (random)	0.99975 $(*)$	0.966(46)

circles) as a function of N. For small sizes the tendency is expected. The estimation of $\widetilde{J}_c(N)$ improves with the system size N. Going to larger sizes, $K^{(F)}$ fails. To understand why, in Fig. 2(b) we plot the distance to the separating hyperplane (6). We only show the δ_1-interval training. The other training sets behave similarly and can be found in [67]. For small N the distance is a smooth function, which confirms that the SVM is able to generalize well to the test data and provides a good estimation of $\widetilde{J}_c(N)$. Increasing N, the distance function flattens in the critical region, hindering the extrapolation of $\widetilde{J}_c(N)$. The explanation for this failure, as we have anticipated, is the orthogonality catastrophe. In Figs. 2(c) and 2(d) we show further proof of this by plotting the kernel matrix $K^{(F)}(J_i, J_j)$ for the whole set Δ. For small sizes ($N = 100$) there is a block structure, marking the ability to distinguish between the two phases. On the other hand, for $N = 1200$, all entries are close to zero (bar the diagonal), i.e., any two states are orthogonal. This is in accordance with the fact that the training using the interval δ_1 breaks down earlier, as it is the furthest away from the critical point, while random training has points closest to the transition point and breaks down last. The corresponding kernels trained with intervals can be found in [67].

One way to get around the catastrophe is by using $K^{(\lambda)}$ instead. For small sizes, both kernels give comparable predictions. However, as shown in panel 2(e) the prediction always improves with N, approaching $\widetilde{J}_c(N) \to J_c \cong 1$ in the limit $N \to \infty$, see Table I. This confirms what was said in Sec. III. In Fig. 2(f) we plot the distance, confirming convergence in the thermodynamic limit [cf. Fig. 2(b)]. Finally, the kernel matrix shows a marked block diagonal structure at any lattice size. See Figs. 2(g) and 2(h) and compare them to their counterparts 2(d) and 2(e), respectively.

To benchmark the SVM predictions, after our last discussion in Sec. III and following [56] we can define $J_c(N)$ as the point at which the function $\partial_J \log \lambda(J, J' = 1.75)$ is maximal. For a fair comparison, we do it on the same set of J's (Δ) as the SVM training. The SVM works better at small lattice sizes. A *tentative* explanation is that our SVM training sets δ_1 and δ_2 are comprised of ground states far from criticality (less so for the random set), whereas the benchmark uses the full dataset Δ, which includes ground states close to the transition. This is relevant because far from criticality the correlation length is finite, and not-too-large systems seem to be sufficient for learning. This is good news for medium-sized quantum processors.

In addition, for the SVM to characterize the QPT, we must check whether it is capable of learning the critical exponents. For thermal transitions, the critical exponents are learned when the distance (6) can be related to the order parameter [34,36] such that the distance inherits the scaling exponents of the latter. In our case, the distance to the hyperplane is a linear combination of the fidelity per site. Thus we expect to have the same finite scaling as λ, from which the corresponding critical parameters can be extracted. This is plotted in Fig. 2(e). Dashed lines are the best fittings to the scaling formula,

$$|J_c - J_c(N)| \sim N^{-1/\nu}. \tag{13}$$

The fitted ν are summarized in Table I. For the Ising model, Hamiltonian (10), $\nu = 1$. Thus the SVM with $K^{(\lambda)}$ is able to learn the critical exponent.

Algorithm 1. Classification using $K^{(F)}$.

```
Require: Training set $\{J_j, y_j\}_{j=1}^{M}$, i.e., parameter values of the
    parameterized Hamiltonian $H(J)$ along with the labels of the
    corresponding phase, $y_j = \pm 1$.
1: Compute the corresponding ground states. Here we are thinking
    of a visual question answering (VQA) algorithm, where the
    circuit depends on some variational parameters $\theta_j$:
        $|\psi(J_j)\rangle = U_{\theta_j}|0\rangle.$
    $U_{\theta_j}$ is the quantum circuit.
2: Store the classical parameters $\theta_j$ (in a classical memory).
3: Prepare the circuit
        $U^{\dagger}_{\theta_i} U_{\theta_j}|0^N\rangle \ (\equiv |\psi_{J_i,J_j}\rangle)$
4: if $i = j$ then
        $K^{(F)}_{ii} = 1$
5: else Measure all bits for the state $|\psi_{J_i,J_j}\rangle$ in the
    computational basis. The frequency $p_{0^N}$ of the all-zero
    outcome corresponds to the state overlap, i.e., the kernel
    entrance
        $(K^{(F)}_{ij})^2 = p_{0^N},$
    see Fig. 2(a).
6: end if
7: Use SVM (hard margin).
```

V. QUANTUM ALGORITHMS

The Ising model allowed us to demonstrate the usefulness of quantum kernels and their performance in the classification of quantum phases. Our arguments are both interpretable and based on the wave function; thus they are exportable to other Hamiltonians, cf. Sec. III, particularly to those where the ground states can be obtained within a quantum processor. For those cases we introduce here two algorithms for computing $K^{(F)}(J_i, J_j)$ and $K^{(\lambda)}(J_i, J_j)$, respectively (see Fig. 3).

While $K^{(F)}(J_i, J_j)$ fails for a sufficiently large system, it gives a very good estimate of J_c for medium sizes, which is the realistic situation within the NISQ era. See the following, Algorithm 1: Notice that the depth of the circuit to calculate any kernel entrance K_{ij} is the sum of the depths to obtain the corresponding $|\psi(J_i)\rangle$ and $|\psi(J_j)\rangle$. The complexity scales as $\mathcal{O}(\epsilon^{-2}M^4)$. Here ϵ is the largest sampling error $\epsilon \sim \mathcal{O}(R^{-1/2})$. R is the number of shots to estimate p_{0^N} [29].

In principle, the most demanding part is obtaining ground states (step 1). It is hard even for a quantum computer. This task is within the QMA complexity class [68], roughly speaking, the NP-complete analog for quantum computers. Nevertheless, quantum computers can be better than classical methods such as density-functional theory [69], density normalization group [70], tensor networks [63], quantum Monte Carlo [71], or even ML-inspired techniques [6] in certain cases. See a recent discussion in [72]. In particular, heuristic quantum algorithms such as adiabatic [73] or variational ones [74] can be efficient for some Hamiltonians. This has been shown, for example, for noncritical spin systems [75]. This justifies the combined use of SVM and quantum computing.

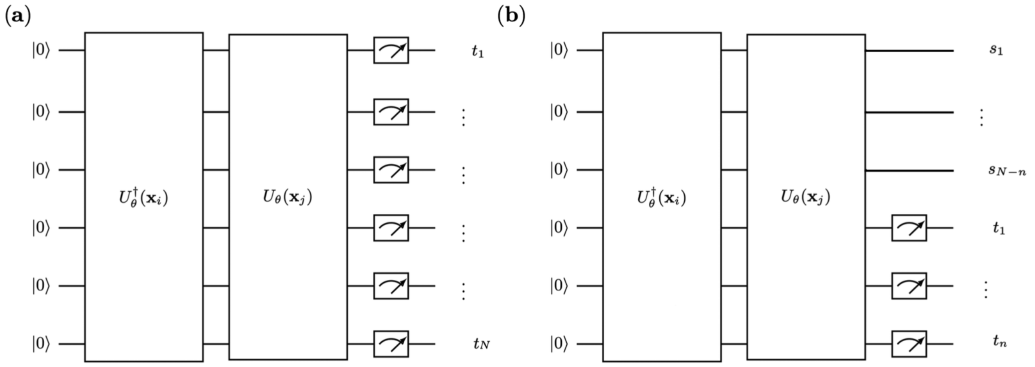


FIG. 3. Classification algorithms. (a) Trial quantum circuit constructed to estimate the fidelity between quantum stated. This measurements are the key of the classification of the phases in the system of study. (b) Our proposal of a quantum algorithm to estimate the fidelity per site in the chain, minimizing the scaling errors.

We have found that classification is successful even if trained away from criticality.

Fidelity per site algorithm

A complete characterization of a QPT requires a scaling analysis of the fidelity per site, so an algorithm to compute this quantity must be devised. An option would be to compute the Nth root of each element $K_{ij}^{(\lambda)} = [K_{ij}^{(F)}]^{1/N}$. It does not work. The inevitable error in $K_{ij}^{(F)}$ explodes (with N) when performing the Nth root.

This can be fixed by modifying Algorithm 1 as follows. After step 3 there, if only $n < N$ qubits are measured [cf. Figs. 2(a) and 2(b)], the frequency, p_{0^n}, of all-zero outcome corresponds to

$$f_n^2 := \langle O_n\rangle_{J,J'} \sim (\lambda_1^n)^2. \tag{14}$$

Here $O_n = |0^n\rangle\langle 0^n|$ and $\langle\ \rangle_{J,J'}$ is the expectation using the state $|\psi_{J_i,J_j}\rangle\rangle$ (cf. Algorithm 1). For the second relation we use (A5). The latter follows from further identities summarized in the Appendix. Furthermore, using (4) and (5), $\lambda_1 \to \lambda$ in the thermodynamic limit. Therefore λ can be inferred by repeating the protocol for different Ns and fitting the value of the scaling parameter. This completes our second algorithm:

Like in Algorithm 1, the complexity scales as $\mathcal{O}(\epsilon^{-2}M^4)$. Now, $\epsilon \sim \mathcal{O}((RR')^{-1/2})$.

VI. CONCLUSIONS

In this work we have discussed the capabilities of quantum kernels to characterize QPTs. For second-order QPTs, we have shown that, using the fidelity per site as kernel, a SVM characterizes a QPT, for both the transition point and critical exponents. Let us emphasize that the kernels rely on the fidelity and thus solely on the wave function. Accordingly, the classification does not need any previous knowledge of the order parameter or the symmetries of the Hamiltonian.

The theory has been applied in the quantum Ising model. This is an exactly solvable model. For nonsolvable models we have presented two quantum algorithms, one based on the fidelity and the other on the fidelity per site. In both of them the most expensive part is obtaining the ground states. Since the SVM training can done out of criticality, we argued that this alleviates this task.

We believe that this work shows a fruitful synergy between quantum information processing (in this work for obtaining the quantum data, i.e., the ground states) and ML (here, for learning the phases of matter) in a classically hard problem [76,77].

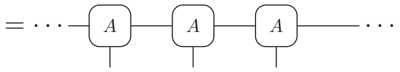
ACKNOWLEDGMENTS

The authors acknowledge enlightening discussions with Javier Molina-Vilaplana, Juanjo Garca-Ripoll, and Eduardo Snchez-Burillo. Funding from the EU (COST Action 15128 MOLSPIN, QUANTERA SUMO, and FET-OPEN Grant No. 862893 FATMOLS), the Spanish MICINN, PID2020-115221GBC41/AEI/10.13039/501100011033, the Gobierno de Aragón (Grant E09-17R Q-MAD), and the CSIC Quantum Technologies Platform PTI-001.

APPENDIX: UNIFORM MATRIX PRODUCT STATES IDENTITIES

Let us follow [78]. Any translational invariant state can be written as

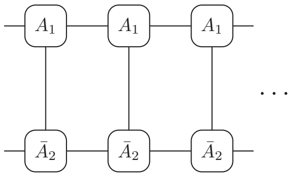
$$|\psi\rangle = \sum_{\{s\}} \boldsymbol{v}_L^\dagger \left[\prod_{m\in\mathbb{Z}} A^{s_m}\right] \boldsymbol{v}_R |\{s\}\rangle \tag{A1}$$



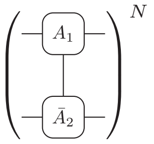
The boundary vectors $\boldsymbol{v}_L$ and $\boldsymbol{v}_R$ must be irrelevant for larger sizes. In the thermodynamical limit, physical states cannot depend on the boundary conditions. Those where the boundary conditions do matter have measure zero in the space of all possible MPS tensors.

The fidelity can be written

$$\langle\Psi(\bar{A}_2)|\Psi(A_1)\rangle = \dots$$



$$= \lim_{N\to\infty}$$

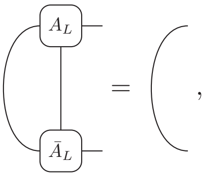


(A2)

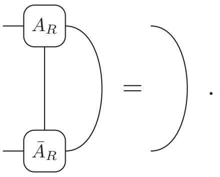
$$= \lim_{N\to\infty} E^N = \lim_{N\to\infty} \lambda_1^N$$

Here $\bar{A}$ is the complex conjugate of A, $E = A_1 \otimes \bar{A}_2$ is the so-called transfer matrix, and λ_1 is the leading E eigenvalue. This formula is used in the main text, Sec. II.

The state is uniquely defined by the tensor A. The opposite is not true. Different tensors can yield the same state. In fact, the *gauge transform* $A \to XAX^{-1}$ leaves the state (A1) invariant. This being said, it is convenient to introduce left and right canonical forms $A_L = LAL^{-1}$ ($A_R = RAR^{-1}$) such that the identities are



(A3a)



(A3b)

Algorithm 2. Classification using $K^{(\lambda)}$.

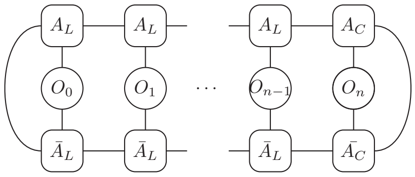
```
Require: Same as in Algorithm 1.
1: Steps 1, 2, and 3 are as in Algorithm 1
2: if $i = j$ then
        $K_{ii}^{(\lambda)} = 1$
3: else
4:    Initialize $n$ such that $1 \ll n \ll N$.
5:    for $l = 0, 1, \ldots, R'$ do
         Measure $n$ qubits $(t_1, t_2 \ldots t_n)$, see Fig. 2(b).
         Obtain $f_n^2$, Eq. (14).
         $n = n + l$
6:    end for
7:    Fit $K_{ij}^{(\lambda)} := \lambda_{ij} \sim (f_n)^{1/n}$.
8: end if
9: Use SVM (hard margin).
```

Canonical forms are useful for computing observables. For our purposes it is sufficient consider observables of the form

$$O = O_0 \otimes \cdots \otimes O_n, \tag{A4}$$

where O_0 acts on a single site that without loss of generality we can label as an 0 site, then 0_i acts on site i with respect to this 0 site. Using the canonical forms (A3a) and (A3b), the expectation value can be computed as

$$\langle O \rangle =$$



(A5)

Here, $A_C = L^{-1}AR$. This formula is used in Algorithm 2.

[1] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, Nat. Phys. **14**, 447 (2018).

[2] S. Ahmed, C. Sánchez Muñoz, F. Nori, and A. F. Kockum, Quantum State Tomography with Conditional Generative Adversarial Networks, Phys. Rev. Lett. **127**, 140502 (2021).

[3] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, Universal quantum control through deep reinforcement learning, npj Quantum Inf. **5**, 33 (2019).

[4] Z. An and D. L. Zhou, Deep reinforcement learning for quantum gate control, Europhys. Lett. **126**, 60002 (2019).

[5] X.-M. Zhang, Z. Wei, and R. Asad, S.-C. Yang, and X. Wang, When does reinforcement learning stand out in quantum control? A comparative study on state preparation, npj Quantum Inf. **5**, 85 (2019).

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

[7] 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).

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

[9] M. Schuld, I. Sinayskiy, and F. Petruccione, An introduction to quantum machine learning, Contemp. Phys. **56**, 172 (2014).

[10] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[11] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers, Quantum Sci. Technol. **3**, 030502 (2018).

[12] M. Altaisky, Quantum neural network, arXiv:quant-ph/0107012.

[13] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nat. Phys. **10**, 631 (2014).

[14] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[15] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[16] T. Dutta, A. Prez-Salinas, J. P. S. Cheng, J. I. Latorre, and M. Mukherjee, Realization of an ion trap quantum classifier, arXiv:2106.14059.

[17] K. Bartkiewicz, C. Gneiting, A. Černoch, K. Jiráková, K. Lemr, and F. Nori, Experimental kernel-based quantum machine learning in finite feature space, Sci. Rep. **10**, 12356 (2020).

[18] V. Belis, S. Gonzlez-Castillo, C. Reissel, S. Vallecorsa, E. F. Combarro, G. Dissertori, and F. Reiter, Higgs analysis with quantum classifiers, arXiv:2104.07692.

[19] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[20] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (ACM, 2019), p. 12.

[21] E. Tang, Quantum-Inspired Classical Algorithms for Principal Component Analysis and Supervised Clustering, Phys. Rev. Lett. **127**, 060503 (2019).

[22] J. M. Arrazola, A. Delgado, B. R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, Quantum **4**, 307 (2020).

[23] H.-Y. Huang, R. Kueng, and J. Preskill, Information-Theoretic Bounds on Quantum Advantage in Machine Learning, Phys. Rev. Lett. **126**, 190505 (2021).

[24] B. Schölkopf, B. Smola, D. Schölkopf, A. Smola, F. Bach, M. Press, and M. Scholkopf, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Adaptive computation and machine learning (MIT Press, Cambridge, MA, 2002).

[25] Encoding may be generalized to a map onto density matrices $\mathbf{x}_j \to \varrho(\mathbf{x}_j)$.

[26] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[27] R. LaRose and B. Coyle, Robust data encodings for quantum classifiers, Phys. Rev. A **102**, 032420 (2020).

[28] M. Schuld, Supervised quantum machine learning models are kernel methods, arXiv:2101.11020.

[29] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[30] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, Machine learning of high dimensional data on a noisy quantum processor, arXiv:2101.09581.

[31] 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**, 2631 (2021).

[32] X. Wang, Y. Du, Y. Luo, and D. Tao, Towards understanding the power of quantum kernels in the NISQ era, Quantum **5**, 531 (2021).

[33] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, arXiv:2010.02174.

[34] P. Ponte and R. G. Melko, Kernel methods for interpretable machine learning of order parameters, Phys. Rev. B **96**, 205146 (2017).

[35] K. Liu, J. Greitemann, and L. Pollet, Learning multiple order parameters with interpretable machines, Phys. Rev. B **99**, 104410 (2019).

[36] C. Giannetti, B. Lucini, and D. Vadacchino, Machine learning as a universal tool for quantitative investigations of phase transitions, Nucl. Phys. B **944**, 114639 (2019).

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

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

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

[40] S. J. Wetzel and M. Scherzer, Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory, Phys. Rev. B **96**, 184410 (2017).

[41] M. J. S. Beach, A. Golubeva, and R. G. Melko, Machine learning vortices at the Kosterlitz-Thouless transition, Phys. Rev. B **97**, 045207 (2018).

[42] F. Schäfer and N. Lörch, Vector field divergence of predictive model output as indication of phase transitions, Phys. Rev. E **99**, 062107 (2019).

[43] T. Mendes-Santos, X. Turkeshi, M. Dalmonte, and A. Rodriguez, Unsupervised Learning Universal Critical Behavior via the Intrinsic Dimension, Phys. Rev. X **11**, 011040 (2021).

[44] N. Maskara, M. Buchhold, M. Endres, and E. van Nieuwenburg, A learning algorithm with emergent scaling behavior for classifying phase transitions, arXiv:2103.15855.

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

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

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

[48] Y. Che, C. Gneiting, T. Liu, and F. Nori, Topological quantum phase transitions retrieved through unsupervised machine learning, Phys. Rev. B **102**, 134213 (2020).

[49] A. Lidiak and Z. Gong, Unsupervised Machine Learning of Quantum Phase Transitions Using Diffusion Maps, Phys. Rev. Lett. **125**, 225701 (2020).

[50] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, Provably efficient machine learning for quantum many-body problems, arXiv:2106.12627.

[51] A. Bohrdt, C. S. Chiu, G. Ji, M. Xu, D. Greif, M. Greiner, E. Demler, F. Grusdt, and M. Knap, Classifying snapshots of the doped Hubbard model with machine learning, Nat. Phys. **15**, 921 (2019).

[52] L. Banchi, J. Pereira, and S. Pirandola, Generalization in quantum machine learning: A quantum information standpoint, PRX Quantum **2**, 040321(2021).

[53] Y. Wu, B. Wu, J. Wang, and X. Yuan, Provable advantage in quantum phase learning via quantum kernel alphatron, arXiv:2111.07553.

[54] P. Zanardi and N. Paunković, Ground state overlap and quantum phase transitions, Phys. Rev. E **74**, 031123 (2006).

[55] H.-Q. Zhou and J. P. Barjaktarevič, Fidelity and quantum phase transitions, J. Phys. A: Math. Theor. **41**, 412001 (2008).

[56] H.-Q. Zhou, J.-H. Zhao, and B. Li, Fidelity approach to quantum phase transitions: Finite-size scaling for the quantum Ising model in a transverse field, J. Phys. A: Math. Theor. **41**, 492002 (2008).

[57] H.-Q. Zhou, Renormalization group flows and quantum phase transitions: Fidelity versus entanglement, arXiv:0704.2945.

[58] 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).

[59] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[60] M. Cozzini, R. Ionicioiu, and P. Zanardi, Quantum fidelity and quantum phase transitions in matrix product states, Phys. Rev. B **76**, 104420 (2007).

[61] S.-J. Gu, Fidelity approach to quantum phase transitions, Int. J. Mod. Phys. B **24**, 4371 (2010).

[62] H.-Q. Zhou, R. Orús, and G. Vidal, Ground State Fidelity from Tensor Network Representations, Phys. Rev. Lett. **100**, 080601 (2008).

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

[64] R. Shaydulin and S. M. Wild, Importance of kernel bandwidth in quantum machine learning, arXiv:2111.05451.

[65] 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 (2011).

[66] The number $M = 133$ is fixed to match the number of points entering in both δ_1 and δ_2.

[67] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevA.105.042432 for additional figures describing other trainings that complete Fig. 2 in the main text.

[68] J. Kempe, A. Kitaev, and O. Regev, The complexity of the local Hamiltonian problem, in *FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science* (Springer, Berlin, Heidelberg, 2004), pp. 372–383.

[69] N. Argaman and G. Makov, Density functional theory: An introduction, Am. J. Phys. **68**, 69 (2000).

[70] U. Schollwöck, The density-matrix renormalization group, Rev. Mod. Phys. **77**, 259 (2005).

[71] M. Troyer and P. Werner, Quantum Monte Carlo simulations, in *Lectures on the Physics of Strongly Correlated Systems XIII: Thirteenth Training Course in the Physics of Strongly Correlated Systems*, edited by A. Avella and F. Mancini, AIP Conf. Proc. No. 1162 (AIP, New York, 2009), p. 98.

[72] B. F. Schiffer, J. Tura, and J. I. Cirac, Adiabatic spectroscopy and a variational quantum adiabatic algorithm, arXiv:2103.01226.

[73] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[74] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[75] C. Bravo-Prieto, J. Lumbreras-Zarapico, L. Tagliacozzo, and J. I. Latorre, Scaling of variational quantum circuit depth for condensed matter systems, Quantum **4**, 272 (2020).

[76] J. I. Cirac and P. Zoller, Goals and opportunities in quantum simulation, Nat. Phys. **8**, 264 (2012).

[77] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

[78] L. Vanderstraeten, J. Haegeman, and F. Verstraete, Tangent-space methods for uniform matrix product states, SciPost Phys. Lect. Notes, 7 (2019).