Extrapolating Quantum Observables with Machine Learning: Inferring Multiple Phase Transitions from Properties of a Single Phase

Rodrigo A. Vargas-Hernández,[1] John Sous,[1,2,3] Mona Berciu,[2,3] and Roman V. Krems[1]
[1]Department of Chemistry, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z1
[2]Department of Physics and Astronomy, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z1
[3]Stewart Blusson Quantum Matter Institute, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Z4

(Received 21 March 2018; revised manuscript received 24 June 2018; published 17 December 2018)

We present a machine-learning method for predicting sharp transitions in a Hamiltonian phase diagram by extrapolating the properties of quantum systems. The method is based on Gaussian process regression with a combination of kernels chosen through an iterative procedure maximizing the predicting power of the kernels. The method is capable of extrapolating across the transition lines. The calculations within a given phase can be used to predict not only the closest sharp transition but also a transition removed from the available data by a separate phase. This makes the present method particularly valuable for searching phase transitions in the parts of the parameter space that cannot be probed experimentally or theoretically.

DOI: 10.1103/PhysRevLett.121.255702

It is very common in quantum physics to encounter a problem with the Hamiltonian

$$H = H_0 + \alpha H_1 + \beta H_2 \tag{1}$$

whose eigenspectrum can be readily computed or measured in certain limits of α and β, e.g., at $\alpha = 0$ or at $\alpha \gg \beta$, but not at arbitrary values of α and β. For such problems, it is necessary to interpolate the properties of the quantum system between the known limits or extrapolate from a known limit. Both the interpolation and the extrapolation become exceedingly complex if the system properties undergo sharp transitions at some values of α and/or β. Such sharp transitions separate the phases of the Hamiltonian (1). Because the wave functions of the quantum system are drastically different in the different phases [1], an extrapolation of quantum properties across phase transition lines is generally considered unfeasible.

Here, we challenge this premise. We note that, while certain properties of quantum systems undergo a sharp change at a phase transition, other properties evolve smoothly through the transition. Using the example of three different lattice models, we show that the evolution of such properties within a given phase contains information about the transitions and the same properties beyond the transitions. We present a machine-learning method that can be trained by *the evolution* of such properties in a given phase to predict the sharp transitions and the properties of the quantum system in other phases by extrapolation. The importance of this result is clear. Characterizing quantum phase transitions embodied in model Hamiltonians is one of the foremost goals of quantum condensed-matter physics. Our work illustrates the possibility of predicting transitions at Hamiltonian parameters, where obtaining the solutions of the Schrödinger equation may not be feasible.

The application of machine-learning (ML) tools for solving problems in condensed-matter physics has recently become popular [2–34]. In all of these applications, ML is used as an efficient method to solve one of three problems: interpolation, classification, or clustering. Interpolation amounts to fitting multidimensional functions or functionals, whereas classification and clustering are used to separate physical data by properties. For example, ML can be used to identify quantum phases of lattice spin Hamiltonians [5,6,12,16,19,23,24]. However, in order to identify a quantum phase transition by interpolation and/or classification, the aforementioned ML models must be trained (fed on input) by the data describing both phases on both sides of the transition. The distinct feature of the present work is a ML method that requires information from only one phase and *extrapolates* the properties of lattice models to and across the transitions. To illustrate the method, we consider four different problems: lattice polaron models with zero, one, and two sharp transitions and the mean-field Heisenberg model with a critical temperature. In all cases, we show that the phase transitions (or lack thereof) can be accurately identified.

We first consider a generalized lattice polaron model describing an electron in a one-dimensional lattice with $N \to \infty$ sites coupled to a phonon field:

$$\mathcal{H} = \sum_k \epsilon_k c_k^\dagger c_k + \sum_q \omega_q b_q^\dagger b_q + V_{e\text{-ph}}, \tag{2}$$

where c_k and b_q are the annihilation operators for the electron with momentum k and phonons with momentum q, $\epsilon_k = 2t\cos(k)$ and $\omega_q = \omega = \text{const}$ are the electron and

phonon dispersions, respectively, and $V_{e\text{-ph}}$ is the electron-phonon coupling. We choose $V_{e\text{-ph}}$ to be a combination of two qualitatively different terms $V_{e\text{-ph}} = \alpha H_1 + \beta H_2$, where

$$H_1 = \sum_{k,q} \frac{2i}{\sqrt{N}} [\sin(k+q) - \sin(k)] c^\dagger_{k+q} c_k (b^\dagger_{-q} + b_q) \tag{3}$$

describes the Su-Schrieffer-Heeger (SSH) [35] electron-phonon coupling and

$$H_2 = \sum_{k,q} \frac{2i}{\sqrt{N}} \sin(q) c^\dagger_{k+q} c_k (b^\dagger_{-q} + b_q) \tag{4}$$

is the breathing-mode model [36]. The lowest energy eigenstate of the model (2) represents polarons known to exhibit two sharp transitions as the ratio α/β increases from zero to large values [37]. At $\alpha = 0$, the model (2) describes breathing-mode polarons, which have no sharp transitions [38]. At $\beta = 0$, the model (2) describes SSH polarons, which exhibit one sharp transition in the polaron phase diagram [35]. At these transitions, the ground state momentum and the effective mass of the polaron change sharply.

Method.—We use Gaussian process (GP) regression as the prediction method [39], described in detail in Supplemental Material [40]. The goal of the prediction is to infer an unknown function $f(\cdot)$ given n inputs $\mathbf{x}_i$ and outputs y_i. The assumption is that $y_i = f(\mathbf{x}_i)$. The function f is generally multidimensional, so $\mathbf{x}_i$ is a vector.

GPs do not infer a single function $f(\cdot)$ but rather a distribution over functions, $p(f|\mathbf{X}, \mathbf{y})$, where $\mathbf{X}$ is a vector of all known $\mathbf{x}_i$ and $\mathbf{y}$ is a vector of the corresponding values y_i. This distribution is assumed to be normal. The joint Gaussian distribution of random variables $f(\mathbf{x}_i)$ is characterized by a mean $\mu(\mathbf{x})$ and a covariance matrix $K(\cdot,\cdot)$. The matrix elements of the covariance $K_{i,j}$ are specified by a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ that quantifies the similarity relation between the properties of the system at two points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the multidimensional space.

Prediction at $\mathbf{x}_*$ is done by computing the conditional distribution of $f(\mathbf{x}_*)$ given $\mathbf{y}$ and $\mathbf{X}$. The mean of the conditional distribution is [39]

$$\boldsymbol{\mu}(\mathbf{x}_*) = \sum_i^n d(\mathbf{x}_*, \mathbf{x}_i) y_i = \sum_i^n \alpha_i k(\mathbf{x}_*, \mathbf{x}_i), \tag{5}$$

where $\alpha = K^{-1}\mathbf{y}$ and $\mathbf{d} = K(\mathbf{x}_*, \mathbf{X})^\top K(\mathbf{X}, \mathbf{X})^{-1}$. The predicted mean $\mu(\mathbf{x}_*)$ can be viewed as a linear combination of the training data y_i or as a linear combination of the kernels connecting all training points $\mathbf{x}_i$ and the point $\mathbf{x}_*$, where the prediction is made. In order to train a GP model, one must choose an analytical representation for the kernel function.

To solve the *interpolation* problem, one typically uses a simple form for the kernel. In the limit of large n, any simple kernel function produces accurate interpolation results [39]. For example, k can be approximated by any of the following functions:

$$k_{\text{LIN}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j + \alpha, \tag{6}$$

$$k_{\text{RBF}}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\frac{1}{2} r^2(\mathbf{x}_i, \mathbf{x}_j)\right], \tag{7}$$

$$k_{\text{MAT}}(\mathbf{x}_i, \mathbf{x}_j) = \left[1 + \sqrt{5} r(\mathbf{x}_i, \mathbf{x}_j) + \frac{5}{3} r^2(\mathbf{x}_i, \mathbf{x}_j)\right] \times \exp\left[-\sqrt{5} r(\mathbf{x}_i, \mathbf{x}_j)\right], \tag{8}$$

$$k_{\text{RQ}}(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \frac{|\mathbf{x}_i - \mathbf{x}_j|^2}{2\alpha \ell^2}\right)^{-\alpha}, \tag{9}$$

where $r^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^\top \times M \times (\mathbf{x}_i - \mathbf{x}_j)$ and M is a diagonal matrix with different length scales ℓ_d for each dimension of $\mathbf{x}_i$. The length-scale parameters ℓ_d, ℓ, and α are the free parameters. We describe them collectively by θ. A GP is trained by finding the estimate of θ (denoted by $\hat{\theta}$) that maximizes the logarithm of the *marginal likelihood* function:

$$\log p(\mathbf{y}|\mathbf{X}, \theta, \mathcal{M}_i) = -\frac{1}{2}\mathbf{y}^\top K^{-1}\mathbf{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi. \tag{10}$$

For the *extrapolation* problem, the prediction produced by Eq. (5) is clearly sensitive to the particular choice of the kernel function. While different interpolation problems can be solved with the same mathematical form of the kernel function, different extrapolation problems generally require different kernels. The key for successful extrapolation is thus to find the appropriate kernel function. Because we aim to solve a variety of different problems with varying underlying physics, the procedure for constructing the kernel must be fully automated and independent of the particular problem under consideration.

Here, we follow Refs. [47,48] to build a prediction method based on a *combination* of products of different kernels (6)–(9). To select the best combination, we use the Bayesian information criterion (BIC) [49],

$$\text{BIC}(\mathcal{M}_i) = \log p(\mathbf{y}|\mathbf{X}, \hat{\theta}, \mathcal{M}_i) - \frac{1}{2}|\mathcal{M}_i| \log n, \tag{11}$$

where $|\mathcal{M}_i|$ is the number of kernel parameters of kernel $\mathcal{M}_i$. Here, $p(\mathbf{y}|\mathbf{X}, \hat{\theta}, \mathcal{M}_i)$ is the marginal likelihood for an optimized kernel $\hat{\theta}$. It is impossible to train and try models with all possible combinations of kernels. We use an iterative procedure schematically depicted in Fig. 1.

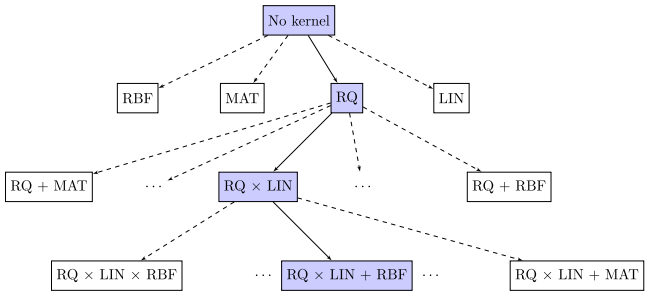

FIG. 1. Schematic diagram of the kernel construction method employed to develop a Gaussian process model with extrapolation power. At each iteration, the kernel with the highest Bayesian information criterion (11) is selected. The labels in the boxes correspond to the kernel functions defined in (6)–(9).

We begin by training a GP model with *each* of the kernels (6)–(9). These kernels have one (LIN), d (RBF and MAT), and two (RQ) free parameters [50]. The algorithm then selects the kernel—denoted k_0—that leads to the model with the highest BIC and combines $k_0(\cdot,\cdot)$ with each of the original kernels k_i defined by Eqs. (6)–(9). The kernels are combined as products $k_0(\cdot,\cdot) \times k_i(\cdot,\cdot)$ and additions $k_0(\cdot,\cdot) + k_i(\cdot,\cdot)$. Each kernel in the combination is scaled by a constant factor, which introduces another free parameter for the product or two parameters for the sum. For each of the possible combinations, a new GP model is constructed and a BIC is computed. The kernel yielding the highest BIC is then used as a new base kernel k_0, and the procedure is iterated. This fully automated algorithm is applied here to four different problems, yielding physical extrapolation results, thus showing that Eq. (11) can be used as a criterion for building prediction models capable of physical extrapolation.

Results.—All GP models are trained by the dispersions $E(K)$, where E is the polaron energy and K is the polaron momentum. These dispersions are calculated for infinite lattices using the momentum average (MA) approach from previous work [37,51–55]. The models are trained to predict the polaron energy as a function of K and the Hamiltonian parameters α, β, and ω. The vectors $\mathbf{x}_i$ are thus $\mathbf{x}_i \Rightarrow \{K, \omega, \alpha, \beta\}$, while $f(\cdot)$ is the polaron energy. Once the models are trained, we numerically compute the ground state momentum K_{GS} and the polaron effective mass from the predicted dispersions [56]. Note that we always train all models by the polaron dispersions in one phase and the models have no *a priori* information about the existence of another phase(s). The transition is encoded in the evolution of the polaron band as a function of $\mathbf{x}$. All results are in units of t.

Figure 2 shows the predictions for the pure SSH polaron model ($\beta = 0$, one sharp transition in the polaron phase diagram). The vertical lines show where the training points end and the extrapolation begins. As can be seen, the GP models predict accurately the location of the transition and can be used for quantitative extrapolation in a wide range of the Hamiltonian parameters to strong electron-phonon coupling. All models, including the ones trained by quantum calculations far removed from the transition point, predict accurately the location of the transition. As the coupling to phonons increases, the polaron develops a phonon-mediated next-nearest-neighbor hopping term: $E(K) = -2t\cos(K) + 2t_2(\lambda_{\mathrm{SSH}})\cos(2K)$, where $t_2(\lambda_{\mathrm{SSH}})$ is a function of λ_{SSH} [35]. The transition occurs when the

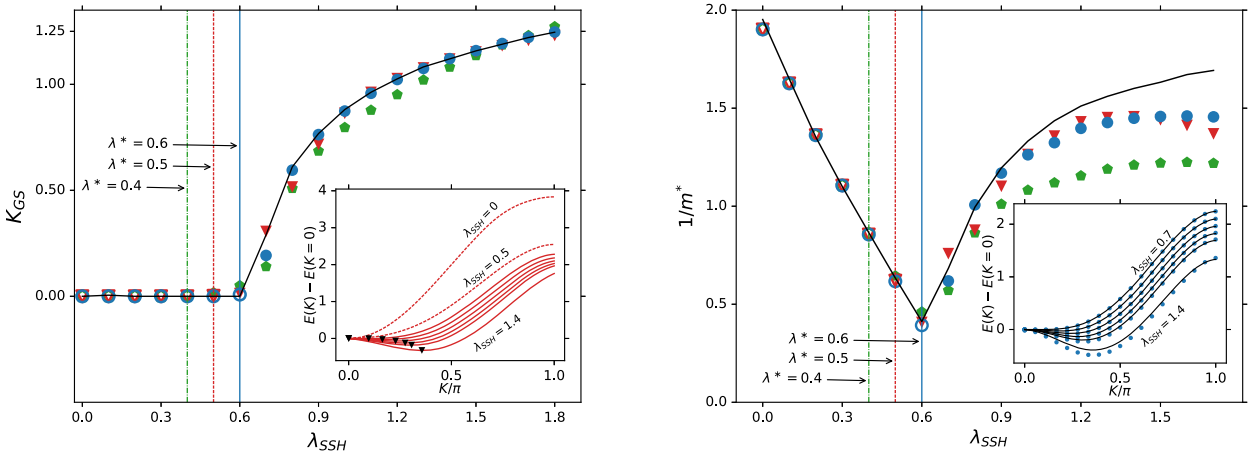

FIG. 2. Extrapolation of the polaron ground state momentum K_{GS} (left) and effective mass m^* (right) across the sharp transition at $\lambda_{\mathrm{SSH}} = 2\alpha^2/t\hbar\omega \approx 0.6$ for the model (2) with $\beta = 0$. The black solid curves are the accurate quantum calculations. The symbols are the predictions of the GP models trained by the full polaron dispersions $E(K)$ at values of $\lambda_{\mathrm{SSH}} \leq \lambda^*$, where λ^* is shown by the vertical lines (solid for circles, dashed for triangles, and dot-dashed for pentagons). The GP models are used for interpolation (open symbols) and extrapolation (full symbols). The algorithm of Fig. 1 yields the kernel $k_{\mathrm{RQ}} \times k_{\mathrm{LIN}} + k_{\mathrm{RBF}}$ for the GP models represented by the triangles and pentagons and $k_{\mathrm{RQ}} \times k_{\mathrm{LIN}} \times k_{\mathrm{MAT}}$ for the circles. Left inset: The polaron dispersions used as input (dashed curves) and predicted by the GP model (solid curves) with $\lambda^* = 0.5$ with the triangles showing the position of the dispersion minimum. Right inset: The polaron dispersions predicted by the GP model trained with $\lambda^* = 0.6$ (solid curves) in comparison with the quantum calculations (symbols).

second term dominates. Figure 2 shows that the GP models trained using the algorithm of Fig. 1 extrapolate accurately this evolution of the polaron energy.

The power of this method is better illustrated with the example of the mixed breathing-mode–SSH model ($\alpha \neq 0$, $\beta \neq 0$) with three phases [37]. The dots in Fig. 3 represent the points of the phase diagram used for training the GP model with the optimized kernels. Remarkably, the model trained by the polaron dispersions all entirely in one phase predicts *both* transitions. The location of the first transition is predicted quantitatively. The second transition is predicted qualitatively. If the model is trained by the polaron properties in two side phases and the prediction is made by extrapolation to low values of λ_{SSH} (lower panel in Fig. 3), both transition lines are predicted quantitatively.

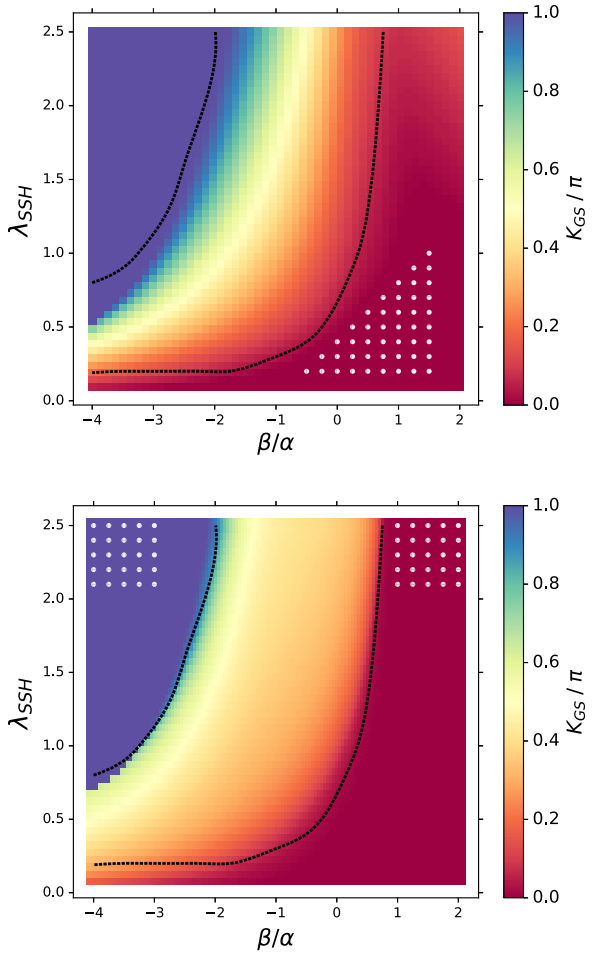

FIG. 3. The polaron ground state momentum K_{GS} for the mixed model (2) as a function of β/α for $\lambda_{\mathrm{SSH}} = 2\alpha^2/t\hbar\omega$. The color map is the prediction of the GP models. The curves are the quantum calculations from Ref. [37]. The models are trained by the polaron dispersions at the parameter values indicated by the white dots. No other information is used. The optimized kernel combination is $(k_{\mathrm{MAT}} + k_{\mathrm{RBF}}) \times k_{\mathrm{LIN}}$ (upper panel) and $(k_{\mathrm{MAT}} \times k_{\mathrm{LIN}} + k_{\mathrm{RBF}}) \times k_{\mathrm{LIN}}$ (lower panel).

As a third independent test, we applied the method to the Holstein polaron model defined by Eq. (2) with $V_{e\text{-ph}} = \mathrm{const}\sum_{k,q} c^{\dagger}_{k+q} c_k (b^{\dagger}_{-q} + b_q)$. Such a model is known to have no transitions [38]. We find that the method presented here can extrapolate accurately the polaron dispersions to a wide range of the Hamiltonian parameters and yields predictions that exhibit no sign of transitions. Since it is often not feasible to explore the entire phase diagram with rigorous quantum calculations, especially for models with many independent parameters, predicting the absence of transitions is as important as locating different phases.

Finally, we demonstrate the method on an analytically soluble model. We consider the Heisenberg model $H = -(J/2)\sum_{i,j} \vec{S}_i . \vec{S}_j$ in the nearest-neighbor approximation. Employing a mean-field description, the resulting free energy density at temperature T is [1,57,58]

$$f(T,m) \approx \frac{1}{2}\left(1 - \frac{T_c}{T}\right) m^2 + \frac{1}{12}\left(\frac{T_c}{T}\right)^3 m^4, \qquad (12)$$

where m is the magnetization and $T_c = 1.25J$ the critical temperature of the phase transition. $T > T_c$ corresponds to the paramagnetic phase, while $T < T_c$ is the ferromagnetic phase.

We train GP models by the results of Eq. (12) in the paramagnetic phase far away from T_c (shaded region in the inset in Fig. 4). We then extrapolate the function $f(T,m)$ across the critical temperature and compute the order parameter m_0 which minimizes $f(T,m)$. Figure 4 demonstrates that m_0 thus predicted can be accurately extrapolated across T_c and far into a different phase. This demonstrates again the general idea behind the technique developed here:

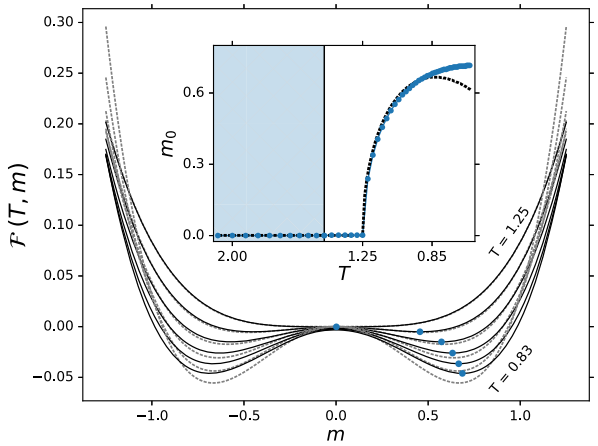

FIG. 4. GP prediction (solid curves) of the free energy density $f(T,m)$ of the mean-field Heisenberg model produced by Eq. (12) (dashed curves). Inset: The order parameter m_0 that minimizes $f(T,m)$: symbols, GP predictions; dashed curve, from Eq. (12). The GP models are trained with 330 points at $1.47 < T < 2.08$ (shaded area) and $-1.25 < m < 1.25$.

use ML to predict the *evolution* of continuous functions that encodes phase transitions.

It is important to point out that the iterative kernel selection algorithm of Fig. 1 must be analyzed before the present method is used for the *quantitative* extrapolation. As the iterations continue, the kernels become more complex, more prone to overfitting, and more difficult to optimize. The quantitative accuracy of the prediction may, therefore, decrease. Supplemental Material [40] illustrates the convergence to Figs. 2–4 with the kernel optimization levels and also the increase of the prediction error after a certain number of levels. To prevent this problem, we stop the kernel optimization when the prediction error is minimal, as explained in Supplemental Material [40]. We emphasize that this does not affect the prediction of the transitions: Once a certain level of Fig. 1 is reached, kernels from the subsequent optimization levels predict the transitions. We have confirmed this for all the results (Figs. 2–4) presented here. Thus, if the goal is to predict the presence or absence of transitions, this method can be used without validation. It is sufficient to check that subsequent levels of the kernel optimization do not produce or eliminate transitions. In order to predict quantitatively the quantum properties by extrapolation, the training data must be divided into the training and validation sets. The models must then be trained with the training set and the error calculated with the validation set. The kernel optimization must then be stopped, when the error is minimal. This is a common approach to prevent the overfitting problem in ML with artificial neural networks.

Summary.—We have presented a powerful method for predicting sharp transitions in Hamiltonian phase diagrams by extrapolating the properties of quantum systems. The method is based on Gaussian process regression with a combination of kernels chosen through an iterative procedure maximizing the predicting power of the kernel. The model thus obtained captures the change of the quantum properties as the system approaches the transition, allowing the extrapolation of the physical properties, even across sharp transition lines.

We believe that the present work is the first example of the application of ML for the extrapolation of physical observables for quantum systems. We have demonstrated that the method is capable of using the properties of the quantum system within a given phase to predict not only the closest sharp transition but also a transition removed from the training points by a separate phase. This makes the present method particularly valuable for searching phase transitions in the parts of the parameter space that cannot be probed experimentally or theoretically. Given that the training of the models and the predictions do not present any numerical difficulty [59], the present method can also be used to guide rigorous theory or experiments in search for phase transitions. Finally, we must note that, although the present extrapolation method works well for all four problems considered, we cannot prove that it is accurate for an arbitrary system so the predictions must always be validated, as is common in machine learning.

We acknowledge useful discussions with Dries Sels and Mathias Sheurer. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by a visiting student fellowship at the Institute for Theoretical Atomic, Molecular, and Optical Physics (ITAMP) at Harvard University and the Smithsonian Astrophysical Observatory (J. S.).

[1] S. Sachdev, *Quantum Phase Transitions* (Cambridge University Press, Cambridge, England, 1999).
[2] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, Phys. Rev. B **90**, 155136 (2014).
[3] L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.
[4] T. Ohtsuki and T. Ohtsuki, J. Phys. Soc. Jpn. **85**, 123706 (2016).
[5] L. Wang, Phys. Rev. B **94**, 195105 (2016).
[6] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017).
[7] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017).
[8] P. Broecker, F. Assaad, and S. Trebst, arXiv:1707.00663.
[9] S. J. Wetzel and M. Scherzer, Phys. Rev. B **96**, 184410 (2017).
[10] S. J. Wetzel, Phys. Rev. E **96**, 022140 (2017).
[11] Y.-H. Liu and E. P. L. van Nieuwenburg, Phys. Rev. Lett. **120**, 176401 (2018).
[12] K. Ch'ng, J. Carrasquilla, R. G. Melko, and E. Khatami, Phys. Rev. X **7**, 031038 (2017).
[13] P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, Sci. Rep. **7**, 8823 (2017).
[14] F. Schindler, N. Regnault, and T. Neupert, Phys. Rev. B **95**, 245134 (2017).
[15] M. J. S. Beach, A. Golubeva, and R. G. Melko, Phys. Rev. B **97**, 045207 (2018).
[16] E. van Nieuwenburg, E. Bairey, and G. Refael, Phys. Rev. B **98**, 060301(R) (2018).
[17] N. Yoshioka, Y. Akagi, and H. Katsura, Phys. Rev. B **97**, 205110 (2018).
[18] J. Venderley, V. Khemani, and E.-A. Kim, Phys. Rev. Lett. **120**, 257204 (2018).
[19] G. Carleo and M. Troyer, Science **355**, 602 (2017).
[20] M. Schmitt and M. Heyl, SciPost Phys. **4**, 013 (2018).
[21] Z. Cai and J. Liu, Phys. Rev. B **97**, 035116 (2018).
[22] Y. Huang and J. E. Moore, arXiv:1701.06246.
[23] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. B **96**, 195145 (2017).
[24] Y. Nomura, A. S. Darmawan, Y. Yamaji, and M. Imada, Phys. Rev. B **96**, 205152 (2017).
[25] D.-L. Deng, X. Li, and S. Das Sarma, Phys. Rev. X **7**, 021021 (2017).
[26] X. Gao and L.-M. Duan, Nat. Commun. **8**, 662 (2017).
[27] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Nat. Phys. **14**, 447 (2018).
[28] T. Hazan and T. Jaakkola, arXiv:1508.05133.

[29] A. Daniely, R. Frostig, and Y. Singer, NIPS **29**, 2253 (2016).
[30] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, in Deep Neural Networks as Gaussian Processes (International Conference on Machine Learning, Vancouver, 2018).
[31] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Müller, and E. K. U. Gross, Phys. Rev. B **89**, 205118 (2014).
[32] L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl, and M. Scheffler, Phys. Rev. Lett. **114**, 105503 (2015).
[33] F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armient, Int. J. Quantum Chem. **115**, 1094 (2015).
[34] F. A. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armient, Phys. Rev. Lett. **117**, 135502 (2016).
[35] D. J. J. Marchand, G. De Filippis, V. Cataudella, M. Berciu, N. Nagaosa, N. V. Prokof'ev, A. S. Mishchenko, and P. C. E. Stamp, Phys. Rev. Lett. **105**, 266605 (2010).
[36] B. Lau, M. Berciu, and G. A. Sawatzky, Phys. Rev. B **76**, 174305 (2007).
[37] F. Herrera, K. W. Madison, R. V. Krems, and M. Berciu, Phys. Rev. Lett. **110**, 223002 (2013).
[38] B. Gerlach and H. Löwen, Rev. Mod. Phys. **63**, 63 (1991).
[39] C. E. Rasmussen and C. K. I. Williams, *Gaussian Process for Machine Learning* (MIT Press, Cambridge, 2006).
[40] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for details of the ML methods, convergence, and the quantum calculations used for training the ML models. Supplemental Material includes Refs. [41–46].
[41] R. S. Sutton and A. G. Barto, *Reinforcement Learning, An Introduction* (MIT Press, Cambridge, 2016).
[42] M. Berciu, Phys. Rev. Lett. **97**, 036402 (2006).
[43] M. Berciu and G. L. Goodvin, Phys. Rev. B **76**, 165109 (2007).
[44] G. L. Goodvin and M. Berciu, Phys. Rev. B **78**, 235120 (2008).
[45] C. P. J. Adolphs and M. Berciu, Phys. Rev. B **90**, 085149 (2014).
[46] M. Berciu and H. Fehske, Phys. Rev. B **82**, 085116 (2010).
[47] D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, Adv. Neural Inf. Process. Syst. **24**, 226 (2011).
[48] D. K. Duvenaud, J. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, Proc. Mach. Learn. Res. **28**, 1166 (2013).
[49] G. Schwarz, Ann. Stat. **6**, 461 (1978).
[50] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for more details on combining kernels in Sec. I. A.
[51] M. Berciu, Phys. Rev. Lett. **97**, 036402 (2006).
[52] J. Sous, M. Chakraborty, C. P. J. Adolphs, R. V. Krems, and M. Berciu, Sci. Rep. **7**, 1169 (2017).
[53] J. Sous, M. Berciu, and R. V. Krems, Phys. Rev. A **96**, 063619 (2017).
[54] J. Sous, M. Chakraborty, R. V. Krems, and M. Berciu, arXiv:1805.06109 [Phys. Rev. Lett. (to be published)].
[55] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for more details of the MA approach in Sec. III.
[56] See Supplemental Material http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for more details in Sec. II. A.
[57] P. M. Chaikin and T. C. Lubensky, *Principles of Condensed Matter Physics* (Cambridge University Press, Cambridge, England, 1998).
[58] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for more details in Sec. III. B.
[59] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.121.255702 for the description of the computational scaling for the ML method adopted here in Sec. I. B.