Extrapolation of quantum observables with Gaussian processes

R. A. Vargas-Hernández and R. V. Krems[1]

Department of Chemistry, University of British Columbia, Vancouver, BC V6T 1Z1, Canada

(Dated: 4 January 2019)

For applications in chemistry and physics, machine learning is generally used to solve one of three problems: interpolation, classification or clustering. These problems use information about physical systems in a certain range of parameters or variables in order to make predictions at unknown values of these variables within the same range. The present work considers the application of machine learning to *extrapolation* of physical properties beyond the range of the training parameters. We show that Gaussian processes can be used to build machine learning models capable of extrapolating the quantum properties of complex systems across quantum phase transitions. The approach is based on training Gaussian process models of variable complexity by the evolution of the physical functions. We show that, as the complexity of the models increases, they become capable of predicting new transitions. We also show that, where the evolution of the physical functions is analytic and relatively simple (the function considered here is $a + b/x + c/x^3$), Gaussian process models with simple kernels (such as a simple Gaussian) yield accurate extrapolation results. We thus argue that Gaussian processes can be used as a meaningful extrapolation tool for a wide variety of problems in physics and chemistry. We discuss strategies to prevent overfitting and obtain meaningful extrapolation results without validation.

I. INTRODUCTION

As described throughout this book, machine learning has in recent years become a powerful tool for physics research. A large number of machine learning applications in physics can be classified as supervised learning, which aims to build a model $\mathcal{F}(\cdot)$ of the $\boldsymbol{x} \mapsto y$ relation, given a finite number of $\boldsymbol{x}_i \mapsto y_i$ pairs. Here, $\boldsymbol{x}$ is a vector of (generally multiple) parameters determining the physical problem of interest and y is a physics result of relevance.

For example, $\boldsymbol{x}$ could be a vector of coordinates specifying the positions of atoms in a polyatomic molecule and y the potential energy of the molecule calculated by means of a quantum chemistry method[1–11]. In this case, $\mathcal{F}(\boldsymbol{x})$ is a model of the potential energy surface constructed based on n energy calculations $\boldsymbol{y} = (y_1, ..., y_n)^\top$ at n points $\boldsymbol{x}_i$ in the configuration space of the molecule. To give another example, $\boldsymbol{x}$ could represent the parameters entering the Hamiltonian of a complex quantum system (e.g., the tunnelling amplitude, the on-site interaction strength and/or the inter-site interaction strength of an extended Hubbard model) and y some observable such as the free energy of the system. Trained by a series of calculations of the observable at different values of the Hamiltonian parameters, $\mathcal{F}(\boldsymbol{x})$ models the dependence of the observable on the Hamiltonian parameters[12–44], which could be used to map out the phase diagram of the corresponding system.

The model $\mathcal{F}$ can be any of the commonly used machine-learning models, including artificial neural networks of various depth or models based on kernel methods such as ridge regression or Gaussian process regression. These standard machine-learning methods can be readily used to obtain the models $\mathcal{F}(\boldsymbol{x})$ accurate within the range of the training data. In other words, if the $\boldsymbol{x}_i \mapsto y_i$ pairs are sampled from the range $\boldsymbol{x}_{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{\max}$, the standard machine-learning algorithms provide models accurate for $\boldsymbol{x} \in [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]$.

In the present work, we discuss the methods for building machine-learning models suitable for *extrapolation*, i.e. models that could make accurate predictions at $\boldsymbol{x} < \boldsymbol{x}_{\min}$ and $\boldsymbol{x} > \boldsymbol{x}_{\max}$, outside the range of the training data. In the absence of knowledge of the analytic mathematical functions underlying physics results, extrapolation is not a well-defined operation. Different models may provide widely different extrapolation results. Therefore, when building an extrapolation method, it is necessary to construct a model that captures the physical evolution of the system. One can imagine this to be an easy task for smooth and simply varying functions: extrapolating a straight line does not require much sophis-

tication. Our goal here is, however, to extrapolate complex physical behaviour without *a priori* knowledge of the physical laws governing the evolution of the system.

For this purpose, we consider a rather challenging problem: extrapolation of physical properties of complex quantum systems with multiple phases across the phase transition lines. The main goal of this work is schematically illustrated in Figure 1. We aim to construct the machine-learning models that, when trained by the calculations or experimental measurements within one of the Hamiltonian phases (encircled region in Figure 1), are capable of predicting the physical properties of the system in the other phases. Of particular interest is the prediction of the phase transitions, which are often challenging to find with rigorous quantum calculations.

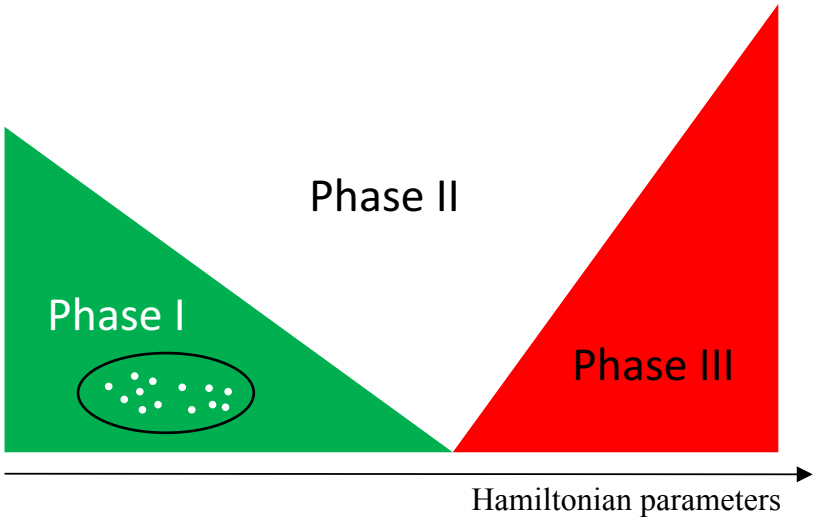

FIG. 1. Schematic diagram of a quantum system with three phases. The goal of the present work is to predict both of the phase transitions based on information about the properties of the system in the encircled region of phase I.

This problem is challenging because the wave functions of the quantum systems – as well as the physical observables characterizing the phases – undergo sharp changes at the phase transitions. Most of the machine-learning models used for interpolation/fitting are, however, smooth functions of $\boldsymbol{x}$. So, how can one construct a machine-learning model that would capture the sharp and/or discontinuous variation of the physical properties by extrapolation? The method discussed here is based on the idea put forward in our earlier work[45].

We assume that the properties of a physical system within a given phase contain information about multiple transitions and that, when a system approaches a phase transition, the properties must change in a way that is affected by the presence of the transition as well as the properties in the other phase(s). In addition, every physical system is always characterized by some properties that vary smoothly through the transition. The goal is then to build a ML model that could be trained by such properties within a given phase, make a prediction in a different phase by extrapolation and predict the properties that change abruptly at the transition from the *extrapolated* models. We will use Gaussian processes to build models capable of extrapolation.

The present chapter is an extension of our work published in Ref.[45], where the main ideas were proposed. Here, we describe the extrapolation method in detail, illustrate its robustness, and discuss the general applicability of ML for extrapolation of physics results. We illustrate the problems arising when the complexity of extrapolation models increases and discuss two approaches to overcome these problems. Finally, we also attempt to address the question of how much information on the physical properties is needed for meaningful extrapolation.

A. Organization of this chapter

The remainder of this chapter is organized as follows. The following section describes the quantum problems considered here. Understanding the physics of these problems is not essential for understanding the contents of this chapter. The main purpose of Section II is to introduce the notation for the physical problems discussed here. These problems are used merely as examples. Section III briefly discusses the application of Gaussian process regression for interpolation in multi-dimensional spaces, mainly to set the stage and define the notation for the subsequent discussion of the extrapolation models. Section IV describes the extension of Gaussian process models to the extrapolation problem. Section V presents the results and Section VI concludes the present chapter. We will abbreviate 'Gaussian process' as GP, 'Artifical Neural Networks' as NN and 'machine-learning' as ML throughout this chapter.

II. QUANTUM SYSTEMS

In this section, we describe the quantum systems considered in the present work. In general, we consider a system described by the Hamiltonian $\hat{H} = \hat{H}(\Gamma)$ that depends on a finite number of free parameters $\Gamma = \{\alpha, \beta, \dots\}$. The observables depend on these Hamiltonian parameters as well as the intrinsic variables $V = \{v_1, v_2, \dots\}$ such as the total linear momentum for few-body systems or thermodynamic variables for systems with a large number of particles. The set $\Gamma + V$ comprises the independent variables of the problems considered here. The ML models $\mathcal{F}$ will be functions of $\Gamma + V$.

More specifically, we will illustrate the extrapolation method using two completely different quantum models: the lattice polaron model and the mean-field Heisenberg model.

A. Lattice polarons

The lattice polaron model describes low-energy excitations of a quantum particle hopping on a lattice coupled to the bosonic field provided by lattice phonons. We consider a quantum particle (often referred to as the 'bare' particle) in a one-dimensional lattice with $N \to \infty$ sites coupled to a phonon field:

$$\mathcal{H} = \sum_k \epsilon_k c_k^\dagger c_k + \sum_q \omega_q b_q^\dagger b_q + V_{\mathrm{e-ph}}, \tag{1}$$

where c_k and b_q are the annihilation operators for the bare particle with momentum k and phonons with momentum q, $\epsilon_k = 2t\cos(k)$ is the energy of the bare particle and $\omega_q = \omega = \text{const}$ is the phonon frequency. The particle-phonon coupling is chosen to represent a combination of two qualitatively different polaron models:

$$V_{\mathrm{e-ph}} = \alpha H_1 + \beta H_2, \tag{2}$$

where

$$H_1 = \sum_{k,q} \frac{2i}{\sqrt{N}} \left[\sin(k+q) - \sin(k)\right] c_{k+q}^\dagger c_k \left(b_{-q}^\dagger + b_q\right) \tag{3}$$

describes the Su-Schrieffer-Heeger (SSH)[47] particle-phonon coupling, and

$$H_2 = \sum_{k,q} \frac{2i}{\sqrt{N}} \sin(q) c_{k+q}^\dagger c_k \left(b_{-q}^\dagger + b_q\right) \tag{4}$$

is the breathing-mode model[48]. We will focus on two specific properties of the polaron in the ground state: the polaron momentum and the polaron effective mass. The ground state band of the model (1) represents polarons whose effective mass and ground-state momentum are known to exhibit two sharp transitions as the ratio α/β increases from zero to large values[49]. At $\alpha = 0$, the model (1) describes breathing-mode polarons, which have no sharp transitions[50]. At $\beta = 0$, the model (1) describes SSH polarons, whose effective mass and ground-state momentum exhibit one sharp transition in the polaron phase diagram[47]. At these transitions, the ground state momentum and the effective mass of the polaron change abruptly.

B. The Heisenberg model

The second model we consider here is the Heisenberg model

$$H = -\frac{J}{2} \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j. \tag{5}$$

This model describes a lattice of interacting quantum spins S_i, which – depending on the strength of the interaction J – can be either aligned in the same direction (ferromagnetic phase) or oriented in various directions leading to zero net magnetization (paramagnetic phase). The parameter J is the amplitude of the interaction and the $\langle .. \rangle$ brackets indicate that the interaction is non-zero only between nearest neighbour spins.

Within a mean-field description, this many-body quantum system has free energy density[51,52]

$$f(T, m) \approx \frac{1}{2} \left(1 - \frac{T_c}{T}\right) m^2 + \frac{1}{12} \left(\frac{T_c}{T}\right)^3 m^4, \tag{6}$$

where m is the magnetization, T is the temperature and T_c is the critical temperature of the phase transition. At temperatures $T > T_c$, the model yields the paramagnetic phase, while $T < T_c$ corresponds to the ferromagnetic phase. The main property of interest here will be the magnetization m. This property undergoes a sharp change at the critical temperature of the paramagnetic - ferromagnetic phase transitions.

III. GAUSSIAN PROCESS REGRESSION FOR INTERPOLATION

The main purpose of GP models, as of any other supervised ML approach, is to make a prediction of some quantity y at an arbitrary point $\boldsymbol{x} \in [\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]$ of a d-dimensional space, given a finite number of values $\boldsymbol{y} = (y_1, ..., y_n)^\top$, where y_i is the value of y at $\boldsymbol{x}_i$. Here, $\boldsymbol{x}_i$ is a d-dimensional vector specifying a particular position in the parameter space and it is assumed that the values $\boldsymbol{x}_i$ sample the entire range $[\boldsymbol{x}_{\min}, \boldsymbol{x}_{\max}]$. It is assumed that y is represented by a continuous function f that passes through the points y_i, so the vector of given results is $\boldsymbol{y} = (f(\boldsymbol{x}_1), ..., f(\boldsymbol{x}_n))^\top$. The goal is thus to infer the function $f(\boldsymbol{x})$ that interpolates the points $y_i \equiv f(x_i)$. The values y_i in the vector $\boldsymbol{y}$ represent the 'training data', since the model is 'trained' by these values.

One of the main differences between GP models and other supervised learning algorithms, such as ones based on NNs, is that GPs infer a *distribution over functions* given the training data $p(f|\boldsymbol{X}, \boldsymbol{y})$, where $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)^\top$. The notation '...$|\boldsymbol{X}, \boldsymbol{y}$' means that the distribution is conditioned by $\boldsymbol{X}$ and $\boldsymbol{y}$. In other words, GP regression produces distributions that depend on the values in $\boldsymbol{X}$ and $\boldsymbol{y}$, as illustrated in Figure 2. The left panel of Figure 2 shows the unrestricted GP (grey curves) before the training. The right panel shows the GP (grey curves) conditioned by the training data (red dots). Obviously, changing the positions of the red dots (i.e. changing $\boldsymbol{X}$) or the values represented by the red dots (i.e. changing $\boldsymbol{y}$) must change the distributions represented by the grey curves in the right panel.

Within the framework of GP, it is assumed that $p(f(\boldsymbol{x}_1), \cdots, f(\boldsymbol{x}_n))$ is jointly Gaussian with a mean $\boldsymbol{\mu}(\boldsymbol{x})$ and covariance $\Sigma(\boldsymbol{x})$. The matrix elements of the covariance are defined as $\Sigma_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $k(\cdot, \cdot)$ is a positively defined kernel function. The kernel function plays a key role since it describes the *similarity* relation between two points.

It is possible to derive the closed-form equations for the conditional mean and variance of a GP[53], yielding

$$\mu(\boldsymbol{x}_*) = K(\boldsymbol{x}_*, \boldsymbol{x})^\top \left[K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 I\right]^{-1} \boldsymbol{y} \tag{7}$$

$$\sigma(\boldsymbol{x}_*) = K(\boldsymbol{x}_*, \boldsymbol{x}_*) - K(\boldsymbol{x}_*, \boldsymbol{x})^\top \left[K(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 I\right]^{-1} K(\boldsymbol{x}_*, \boldsymbol{x}), \tag{8}$$

where $\boldsymbol{x}_*$ is a point in the parameter space where the prediction $\boldsymbol{y}_*$ is to be made; $K(\boldsymbol{x}, \boldsymbol{x})$ is the $n \times n$ square matrix with the elements $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ representing the covariances between $y(\boldsymbol{x}_i)$ and $y(\boldsymbol{x}_j)$. The elements $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are represented by the kernel function.

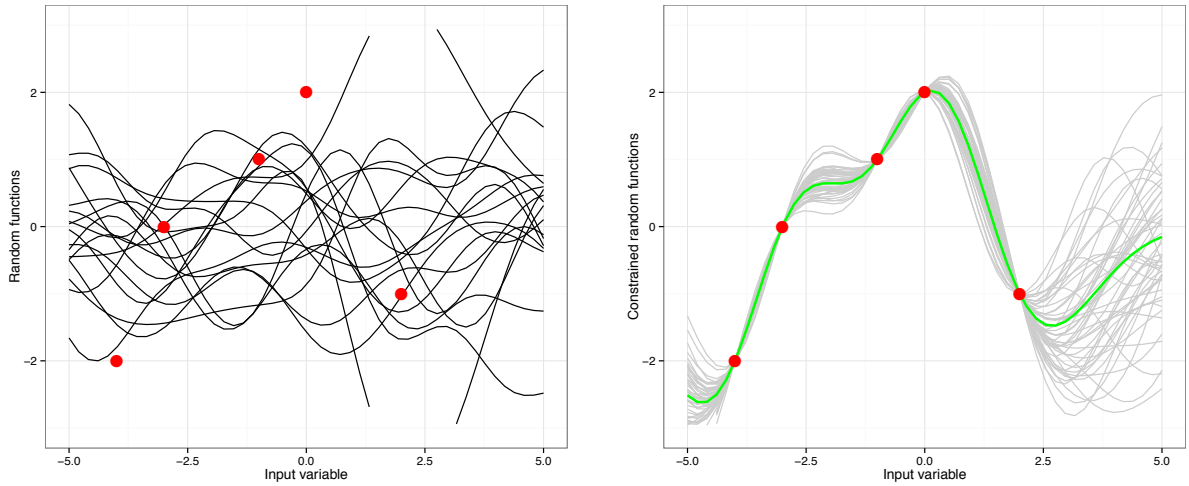

FIG. 2. Left: unconditional (unconstrained) Gaussian distribution represented by the grey lines. Right: multi-variate Gaussian distribution (grey curves) conditioned by the training data (red dots). The green curve represents the mean of the conditional Gaussian distribution.

Eq. (7) can then be used to make the prediction of the quantity y at point $\boldsymbol{x}^*$, while Eq. (8) can be used as the error of the prediction.

In this work, the GP models are trained by the results of quantum mechanical calculations. For the case of the polaron models considered here,

$$\boldsymbol{x}_i \Rightarrow \{\text{polaron momentum } K,\ \text{Hamiltonian parameter } \alpha,$$
$$\text{Hamiltonian parameter } \beta, \text{phonon frequency } \omega\}.$$

For the case of the Heisenberg model considered here,

$$\boldsymbol{x}_i \Rightarrow \{\text{Temperature } T,\ \text{magnetization } m\}$$

As already mentioned, $\boldsymbol{y} \Rightarrow f(\boldsymbol{x})$ is a vector of quantum mechanics results at the values of the parameters specified by $\boldsymbol{x}_i$. For the case of the polaron models considered here, $\boldsymbol{y} \Rightarrow$ polaron energy E. For the case of the Heisenberg model considered here, $\boldsymbol{y} \Rightarrow$ free energy density.

To train a GP model, it is necessary to assume some analytic form for the kernel function $k(\cdot,\cdot)$. In the present work, we will use the following analytic forms for the kernel functions:

$$k_{\rm LIN}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{x}_i^\top \boldsymbol{x}_j + \ell \tag{9}$$

$$k_{\rm RBF}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \exp\left(-\frac{1}{2} r^2(\boldsymbol{x}_i,\boldsymbol{x}_j)\right) \tag{10}$$

$$k_{\rm MAT}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \left(1+\sqrt{5}\, r^2(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{5}{3}\, r^2(\boldsymbol{x}_i,\boldsymbol{x}_j)\right) \times \exp\left(-\sqrt{5}\, r^2(\boldsymbol{x}_i,\boldsymbol{x}_j)\right) \tag{11}$$

$$k_{\rm RQ}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \left(1+\frac{|\boldsymbol{x}_i-\boldsymbol{x}_j|^2}{2\alpha\ell^2}\right)^{-\alpha} \tag{12}$$

where $r^2(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i-\boldsymbol{x}_j)^\top \times M \times (\boldsymbol{x}_i-\boldsymbol{x}_j)$ and M is a diagonal matrix with different length-scales ℓ_d for each dimension of $\boldsymbol{x}_i$. The unknown parameters of these functions are found by maximizing the log *marginal likelihood* function,

$$\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top K^{-1}\boldsymbol{y} - \frac{1}{2}\log|K| - \frac{n}{2}\log(2\pi), \tag{13}$$

where θ denotes collectively the parameters of the analytical function for $k(\cdot,\cdot)$ and $|K|$ is the determinant of the matrix K. Given the kernel functions thus found, Eq. (7) is a GP model, which can be used to make a prediction by interpolation.

A. Overfitting with GP models

A special note should be made about the possibility of overfitting with GP models. Overfitting occurs when the fitting model becomes so complex that it aims to reproduce the training points, while yielding highly inaccurate results between the training points. The most common example of overfitting is the fit of a series of points all on a straight line by a sum of high-order polynomials. The polynomials may pass through the given points but do not reproduce the straight line between the points. Overfitting is a common problem in applications of NNs, because NNs are complex analytic fits of the training data.

Overfitting is generally *not* a problem for GP models. This is owing to the fact that GP models depend on a very small number of independent kernel parameters and the prediction is not an analytic fit but a probability distribution, which becomes more accurate as the number of training points increases. In other words, the conditional distributions become more accurate when restricted by more conditions. However, overfitting *can* occur for GP

models with very complex kernels. In such cases, it will manifest itself as a decrease in the prediction accuracy with increasing complexity of the kernels.

The interpolation problems of relevance to physics can generally be described quite accurately by GP models with simple kernels. For a variety of examples, see Refs.[7–11,54,55] In all of these examples, the complexity of the kernels is similar to that of kernels in Eqs. (9) - (13). However, as will be discussed below, the extrapolation problem may require rather complex kernels and overfitting may become an important factor.

B. Model selection criteria

As Eq. (7) clearly shows, the GP models with different kernel functions will generally have a different predictive power. Consider two models with two different kernels. Which of the two models is better? In the context of physics, which of the two models is more physical?

Before we consider this question, it is necessary to discuss the difference between *interpolation* and *extrapolation*. To make a prediction by interpolation, the model does not need to be physical. Given enough training data, any GP model should be accurate. In other words, if the training points are close enough in the parameter space, the value between these points can be accurately predicted with any GP model with any kernel. For completeness, we note that the efficiency of the GP model (defined here as the number of training points required for a prediction with desired accuracy) and the differentiability of the GP process (i.e. the number of derivatives that the function in Eq. (7) can permit) do depend on the kernel function. However, in the limit of a large number of training points, any GP model should yield accurate interpolation results.

This is, of course, not the case for *extrapolation*. As Eq. (7) shows, the result of extrapolation must be sensitive to the particular choice of the kernel function. Thus, in the context of the present work, 'better' or 'more physical' model refers to models that yield more accurate extrapolation results. Ideally, the most reliable model will capture the analytic behaviour, if any, of the physical data to be extrapolated. So, how to quantify the model quality?

In principle, one could use the marginal likelihood as a metric to compare models with different kernels. However, different kernels have different numbers of free parameters and the second term of Eq. (13) directly depends on the number of parameters in the kernel. This

makes the log marginal likelihood undesirable to compare kernels of different complexity.

As shown in Ref.[56], a better metric could be the Bayesian information criterion (BIC) defined as

$$\mathrm{BIC}(\mathcal{M}_i) = \log p(\boldsymbol{y}|\boldsymbol{x}, \hat{\theta}, \mathcal{M}_i) - \frac{1}{2}|\mathcal{M}_i| \log n \tag{14}$$

where $|\mathcal{M}_i|$ is the number of kernel parameters of the kernel $\mathcal{M}_i$. In this equation, $p(\boldsymbol{y}|\boldsymbol{x}, \hat{\theta}, \mathcal{M}_i)$ is the marginal likelihood for the optimized kernel $\hat{\theta}$ which maximizes the logarithmic part. The assumption – one that will be tested in the present work for physics applications – is that better models have a higher BIC. The last term in Eq. (14) penalizes kernels with larger number of parameters. The optimal BIC will thus correspond to the kernel yielding the largest value of the marginal log-likelihood function with the fewest number of free parameters.

IV. EXTRAPOLATION WITH GAUSSIAN PROCESSES

If the BIC is a good metric to quantify the model quality, one can design a general and powerful algorithm for building accurate extrapolation models[57,58]. The algorithm proposed in Refs.[57,58] aims to build up the complexity of kernels, starting from the simple kernels (9) - (13), in an interative procedure guided by the values of the BIC. Here, we employ this procedure to extrapolate the quantum properties embodied in lattice models across the phase transition lines.

A. Learning with kernel combinations

When designing a GP model capable of extrapolation, the main question is, how to increase the complexity of the kernel functions in a systematic way that prevents overfitting and results in a model that captures the physical behaviour of the training results? The approach adopted here starts with the simple kernels (9) - (13). For each of the kernels, a GP model is constructed and the BIC is calculated. The kernel corresponding to the highest BIC is then selected as the best kernel. We will refer to such kernel as the 'base' kernel and denote it by k_0.

The base kernel is then combined with each of the kernels (9) - (13). The new 'combined'

kernels are chosen to be either of the sum form

$$c_0 k_0 + c_i k_i \tag{15}$$

or of the product form

$$c_i \times k_0 \times k_i, \tag{16}$$

where c_0 and c_i are treated as independent constants to be found by the maximization of the marginal log-likelihood. The GP models with each of the new kernels are constructed and the BIC values are calculated. The kernel of the model with the highest BIC is then chosen as k_0 and the process is iterated. We thus have a 'greedy' search algorithm that is an 'optimal policy' algorithm[59] that selects the kernel assumed optimal based on the BIC at every step in the search.

We note that a similar procedure could be used to improve the quality of the kernels for the interpolation problems. In this case, it may also be possible to use the prediction error on the validation set of data for the kernel selection. We have done this in one of our recent articles[60], where GP was used to construct a six-dimensional potential energy surface for a chemically reactive complex with a very complex landscape. A similar idea could be applied to the extrapolation problem. One might design a method, where the training data are divided into a training set and a validation set and the kernel selection in the algorithm described above is guided by the error on the validation set. We have not attempted to do this in the present work. We will compare the relative performance of the validation error and the BIC as the kernel selection metric in a future work.

V. EXTRAPOLATION OF QUANTUM PROPERTIES

In this section, we present the results illustrating the performance of the extrapolation algorithm described above for the prediction of the quantum properties of complex systems outside the range of the training data. Our particular focus is on predicting properties that undergo a sharp variation or discontinuity at certain values of the Hamiltonian parameters. Such properties cannot be directly modelled by GPs because the mean of a GP is a smooth, differentiable function.

The main idea proposed in our previous work[45] is to train a GP model with functions (obtained from the solutions to the Schrödinger equation) that vary smoothly across phase

transitions and derive the properties undergoing sharp changes from such smoothly varying function. We thus expect this procedure to be generally applicable to extrapolation across second-order phase transitions. Here, we present two examples to illustrate this. The particular focus of the discussion presented below is on how the extrapolation method converges to the accurate predictions as the complexity of the kernels increases.

A. Extrapolation across sharp polaron transitions

As discussed in section II, the Hamiltonian describing a quantum particle coupled to optical phonons through a combination of two couplings defined by Eq. (2) yields polarons with unusual properties. In particular, it was previously shown[49] that the ground-state momentum of such polarons undergoes two sharp transitions as the ratio α/β in Eq. (2) as well as the parameter $\lambda = 2\alpha^2/t\hbar\omega$ are varied. The dimensionless parameter λ is defined in terms of the bare particle hopping amplitude t and the phonon frequency ω. It quantifies the strength of coupling between the bare particle and the phonons. One can thus calculate the ground-state momentum or the effective mass of the polaron as a function of λ and α/β. The values of λ and α/β, where the polaron momentum and effective mass undergo sharp changes, separate the 'phases' of the Hamiltonian (1).

The GP models are trained by the *polaron energy dispersions* (i.e. the full curves of the dependence of the polaron energy on the polaron momentum) at different values of λ, α and β. These models are then used to extrapolate the full energy dispersions to values of λ, α and β outside the range of the training data and the momentum of the polaron with the lowest energy is calculated from these dispersion curves. The results are shown in Figure 3. Each of the white dots in the phase diagrams depicted specifies the values of α, β and λ, for which the polaron dispersions were calculated and used as the training data. One can thus view the resulting GP models as four-dimensional, i.e. depending on α, β, λ and the polaron momentum.

Figure 3 illustrates two remarkable results:

- The upper panel illustrates that the GP models are capable of predicting *multiple* new phase transitions by using the training data *entirely* in one single phase. This proves our conjecture[45] that the evolution of physical properties with the Hamiltonian

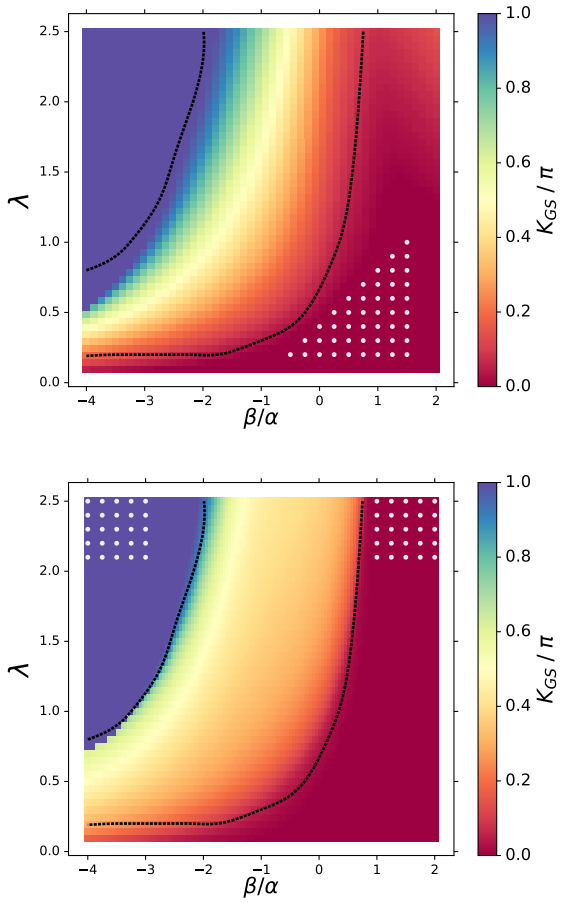

FIG. 3. Adapted with permission from Ref.[45], Copyright © APS, 2018. The polaron ground state momentum K_{GS} for the mixed model (1) as a function of β/α for $\lambda = 2\alpha^2/t\hbar\omega$. The color map is the prediction of the GP models. The curves are the quantum calculations from Ref.[49]. The models are trained by the polaron dispersions at the parameter values indicated by the white dots. The optimized kernel combination is $(k_{MAT} + k_{RBF}) \times k_{LIN}$ (upper panel) and $(k_{MAT} \times k_{LIN} + k_{RBF}) \times k_{LIN}$ (lower panel).

parameters in a single phase contains information about multiple phases and multiple phase transitions.

- While perhaps less surprising, the lower panel illustrates that the accuracy of the predictions increases significantly and the predictions of the phase transitions become quantitative if the models are trained by data in two phases. The model illustrated in this panel extrapolates the polaron properties from high values of λ to low values of λ. Thus, the extrapolation becomes much more accurate if the models are trained by

data in multiple phases.

In the following section we analyze how the kernel selection algorithm described in Ref.[45] and briefly above arrives at the models used for the predictions in Figure 3.

B. Effect of kernel complexity

Figure 4 illustrates the performance of the models with kernels represented by a simple addition of two simple kernels, when trained by the data in two phases, as in the lower panel of Figure 3. The examination of this figure shows that the extrapolation accuracy, including the prediction of the number of the phase transitions, is sensitive to the kernel combination. For example, the models with the combination of the RBF and LIN kernels do not predict any phase transitions. Most of the other kernel combinations predict only one of the two transitions. Remarkably, the combination of two RBF kernels already leads to the appearance of the second phase transition, and allows the model to predict the location of the first transition quite accurately. The combination of Figures 3 and 4 thus illustrates that the BIC is a meaningful metric to guide the kernel selection algorithm, as it rules out many of the kernles leading to incorrect phase diagrams shown in Figure 4. The results in Figure 4 also raise the question, how many combinations are required for kernels to allow quantitative predictions?

To answer this question, we show in Figures 5 and 6 the convergence of the phase diagrams to the results in Figure 3 with the number of iterations in the kernel selection algorithm. We use the following notation to label the figure panels: GPL-X, where X is the number of iteration. Thus, $X = 0$ corresponds to level zero of the kernel selection algorithm, i.e. GPL-0 is the phase diagram predicted by the model with the single simple kernel leading to the highest BIC. Level $X = 1$ corresponds to kernels constructed as the simple combinations (15) or (16). Level $X = 2$ corresponds to kernels of the form (15) or (16), where k_i is a combination of two kernels. As can be seen from Figures 5 and 6, leveld $X = 2$ and $X = 3$ produce kernels with sufficient complexity for accurate predictions.

It must be noted that increasing the complexity of the kernels further (by increasing X) often decreases the accuracy of the predictions. This is illustrated in Figure 7. We assume that this happens either due to overfitting or because the kernels become so complex that it is difficult to optimize them and the maximization of the marginal log-likelihood gets stuck

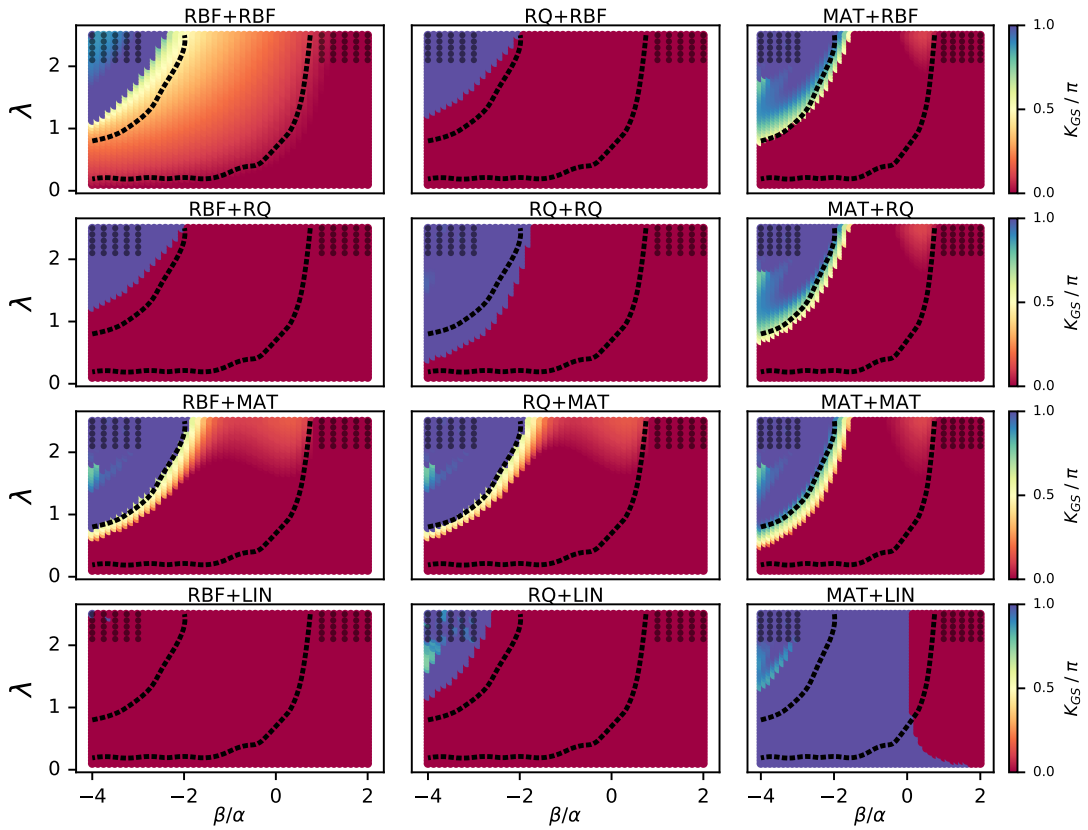

FIG. 4. The polaron ground state momentum K_{GS} for the mixed model (1) as a function of β/α for $\lambda = 2\alpha^2/t\hbar\omega$. The black dashed curves are the calculations from Ref.[49]. The color map is the prediction of the GP models with the fully optimized kernels. The models are trained by the polaron dispersions at the parameter values indicated by the black dots. The different kernels considered here are all possible pairwise additions (15) of two simple kernels from the family of kernels (k_{MAT} , k_{RQ} and k_{RBF}).

in a local maximum. To overcome this problem, one needs to optimize kernels multiple times starting from different conditions (either different sets of training data or different initial kernel parameters) and stop increasing the complexity of kernels when the optimization produces widely different results. Alternatively, the models could be validated by a part of the training data and the complexity of the kernels must be stopped at level X that corresponds to the minimal validation error, as often done to prevent overfitting with NNs[46].

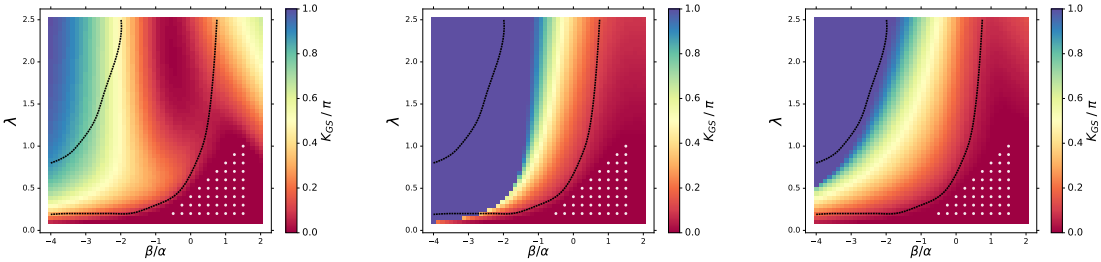

FIG. 5. Adapted from the supplementary material of Ref.[45]. Improvement of the phase diagram shown in Figure 3 (upper panel) with the kernel complexity increasing as determined by the algorithm described in Section IV A. The panels correspond to the optimized kernels GPL-0 (left), GPL-1 (center), GPL-2 (right), where "GPL-X" denotes the optimal kernel obtained after X depth levels.

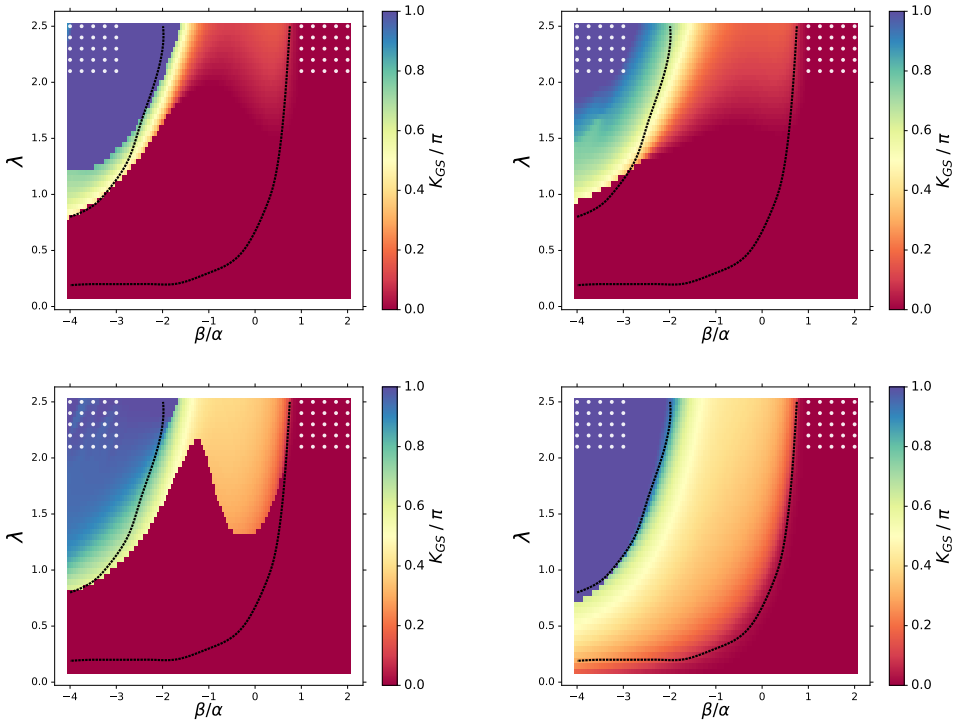

FIG. 6. Adapted from the supplementary material of Ref.[45]. Improvement of the phase diagram shown in Figure 3 (lower panel) with the kernel complexity increasing as determined by the algorithm depicted in Section IV A. The panels correspond to the optimized kernels GPL-0 (upper left), GPL-1 (upper right), GPL-2 (lower left), GPL-3 (lower right), where "GPL-X" denotes the optimal kernel obtained after X depth levels.

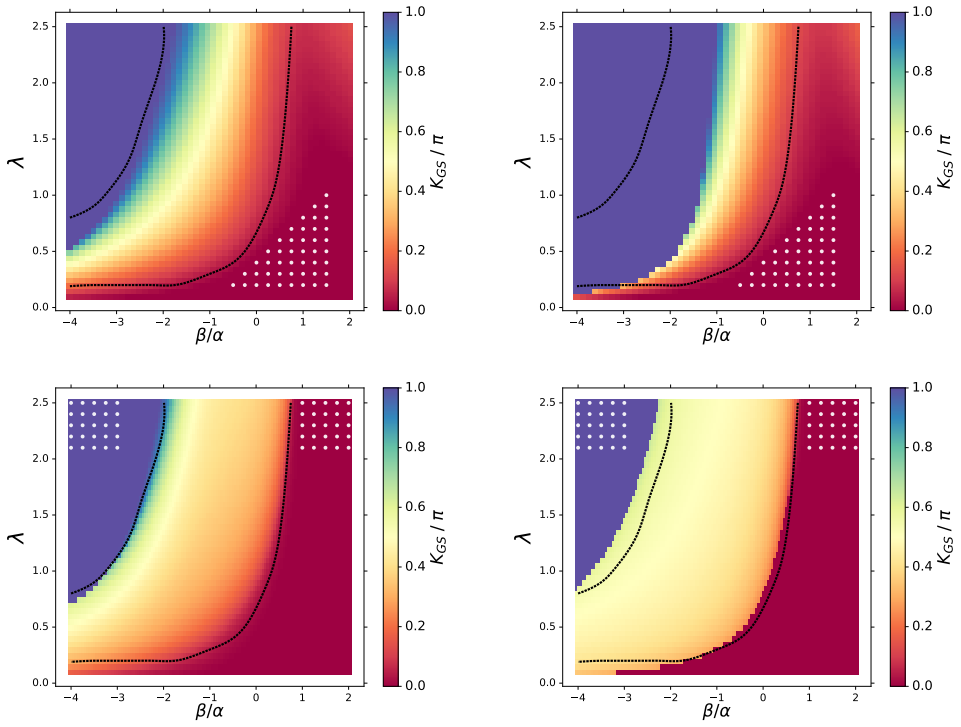

FIG. 7. Decrease of the prediction accuracy with increasing kernel complexity. Upper panels: left – GPL-2 (same as the right panel of Figure 5), right – GPL-3. Lower panels: left – GPL-3 (same as the lower right panel of Figure 6), right – GPL-4.

C. Extrapolation across paramagnetic - diamagnetic transition

In this section, we discuss the Heisenberg spin model described by the lattice Hamiltonian

$$\mathcal{H} = -\frac{J}{2} \sum_{\langle i,j \rangle} \bar{S}_i \cdot \bar{S}_j, \tag{17}$$

where $\langle i, j \rangle$ only account for nearest-neighbour interactions between different spins $\bar{S}_i$. The free energy of the system can be calculated within the mean-field approximation to yield

$$f(T, m) \approx \frac{1}{2} \left(1 - \frac{T_c}{T}\right) m^2 + \frac{1}{12} \left(\frac{T_c}{T}\right)^3 m^4, \tag{18}$$

where m is the magnetization and $T_c = 1.25$ is the critical temperature of the phase transition between the paramagnetic ($T > T_c$) and diamagnetic ($T < T_c$) phase.

We train GP models by the entire free-energy curves at temperatures far above T_c. The free energy curves are then predicted by the extrapolation models at temperatures decreasing

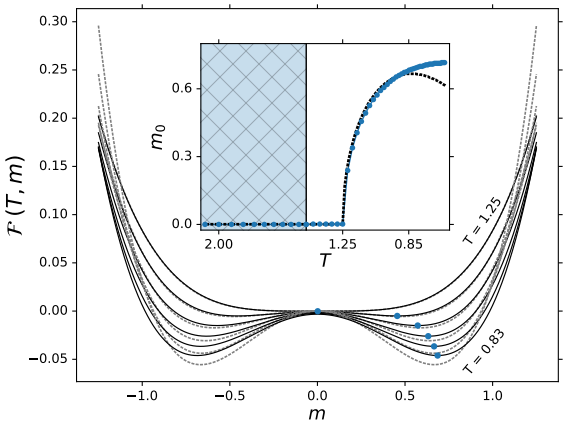

FIG. 8. Adapted with permission from Ref.[45], Copyright © APS, 2018. GP prediction (solid curves) of the free energy density $f(T, m)$ of the mean-field Heisenberg model produced by Eq. (18) (dashed curves). Inset: the order parameter m_0 that minimizes $f(T, m)$: symbols – GP predictions, dashed curve – from Eq. (18). The GP models are trained with 330 points at $1.47 < T < 2.08$ (shaded area) and $-1.25 < m < 1.25$.

to the other side of the transition. The oder parameter m_0 – defined as the value of magnetization that minimizes free energy – is then computed from the extrapolated predictions. The results are shown in Figure 8.

As evident from Eq. (18), the free-energy curves have an analytic dependence on temperature T so this is a particularly interesting case for testing the extrapolation models. Can the kernel selection algorithm adopted here converge to a model that will describe accurately the analytic dependence of the free energy (18) as well as the order parameter derived from it? We find that the temperature dependence of Eq. (18) can be rather well captured and accurately extrapolated by a model already with one simple kernel! However, this kernel must be carefully selected. As Figure 9 illustrates, the accuracy of the free-energy prediction varies widely with the kernel. This translates directly into the accuracy of the order-parameter prediction illustrated by Figure 10. Figure 10 illustrates that the RBF and RQ kernels capture the evolution of the order parameter quantitatively, while the LIN, MAT and quadratic kernels produce incorrect results.

Table I lists the BIC values for the models used to obtained the results depicted in Figure 10, clearly demonstrating that the higher value of the BIC corresponds to the model with

Kernel type	BIC
RQ	8667.10
RBF	8657.13
MAT	7635.20
LIN	- 104437128213.0
LIN×LIN	-10397873744.9

TABLE I. The numerical values of the BIC (14) for the models with different simple kernels (9) - (13) used for the predictions of the order parameter depicted in Figure 10

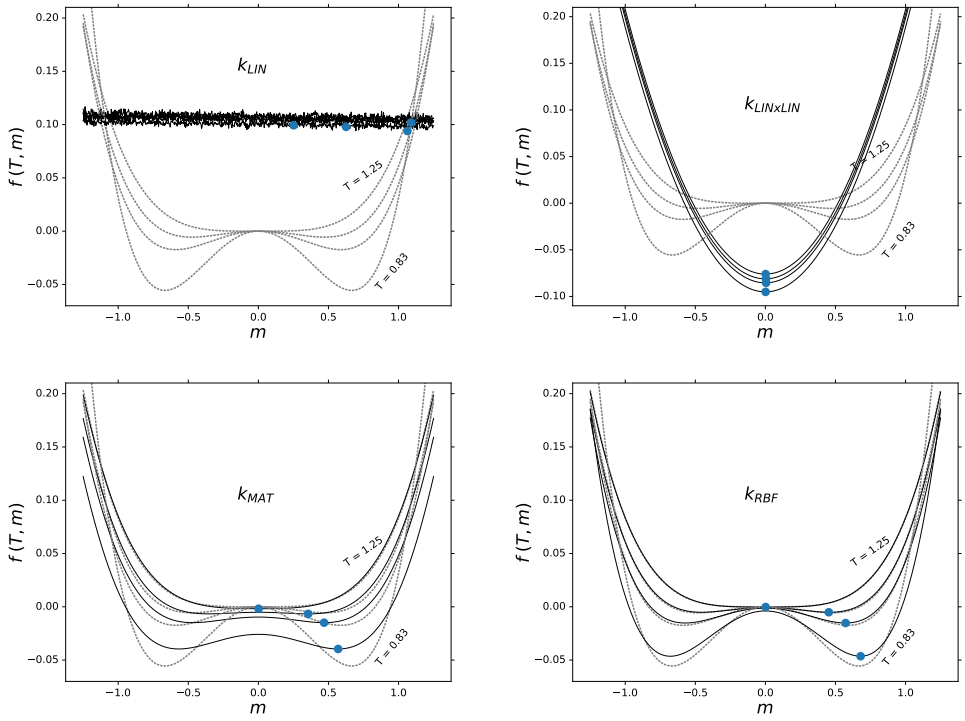

FIG. 9. GP prediction (solid curves) of the free energy density $f(T, m)$ of the mean-field Heisenberg model produced by Eq. (18) (dashed curves). All GP models are trained with 330 points at $1.47 < T < 2.08$ (shaded area) and $-1.25 < m < 1.25$. The kernel function used in the GP models is indicated in each panel.

the better prediction power.

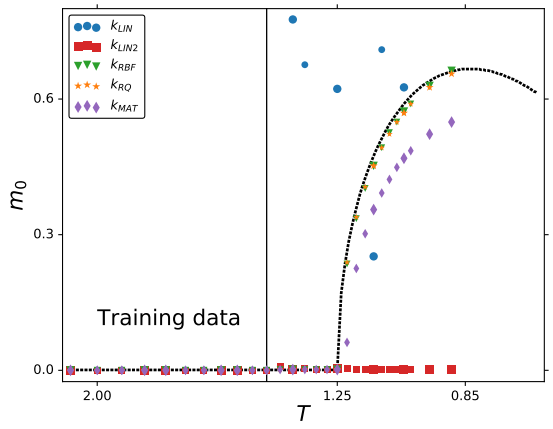

FIG. 10. The order parameter m_0 that minimizes $f(T, m)$: symbols – GP predictions, dashed curve – from Eq. (18). The order parameter m_0 is computed with the GP predictions using different kernels, illustrated in Figure (9).

D. Effect of the number and positions of the training points

In this section, we reexamine the predictions of the extrapolation models of the phase diagram depicted in Figure 3 with a variable number of training points. Figure 11 shows the results of the extrapolation predictions obtained with models trained by the quantum calculations at different values of λ and α/β. The figure illustrates the following:

- The extrapolation models capture both transitions even when trained by the quantum calculations far removed from the transition line and with a random distribution of training points.
- The predictions of the transitions become more accurate as the distribution of the training points approaches the first transition line.

This suggests that the following algorithm can be used to make stable predictions of unknown phase transitions:

(1) Sample the phase diagram with a cluster of training points at random.

(2) Identify the phase transitions by extrapolation in all directions.

(3) Move the cluster of the training points towards any predicted transition.

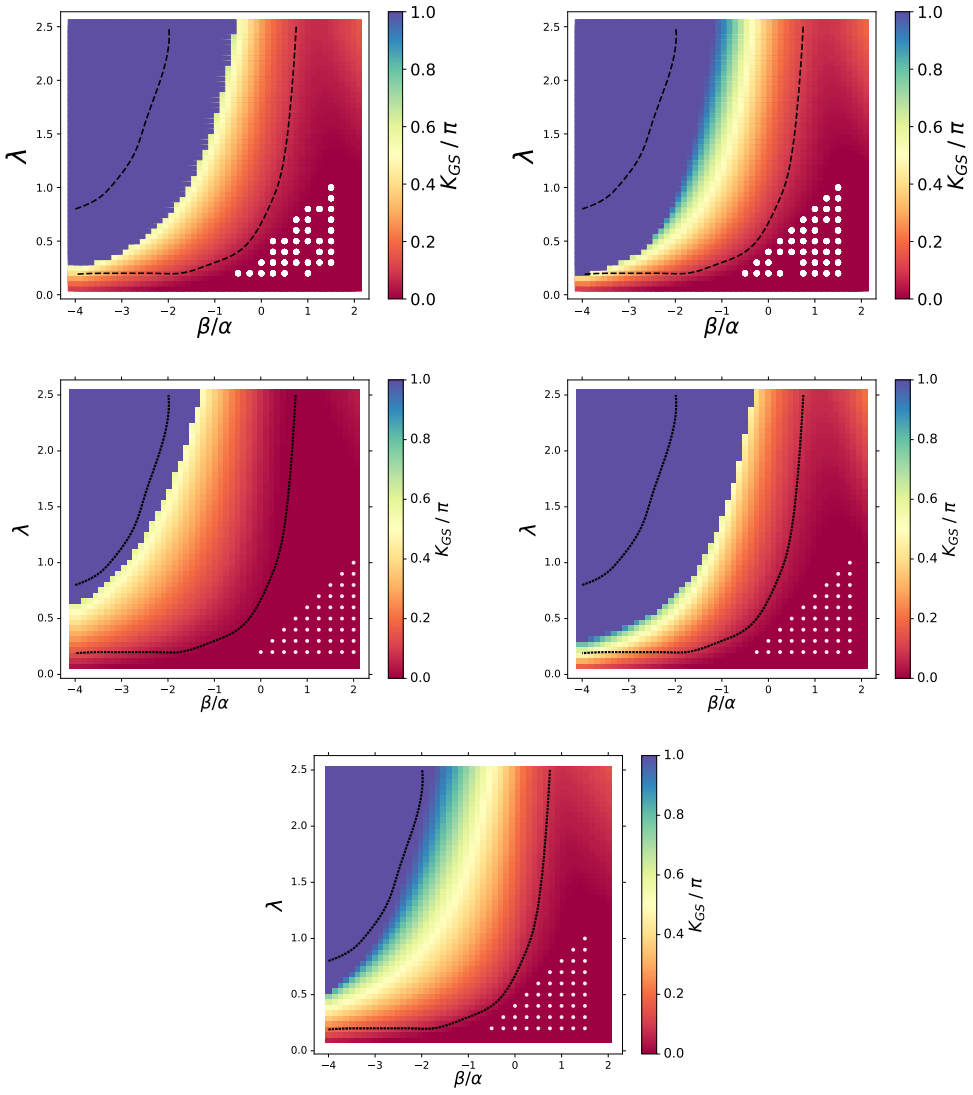

FIG. 11. Dependence of the prediction accuracy on the number and positions of training points. The white dots indicate the values of the parameters λ and α/β, at which the quantum properties were calculated for training the GP models. All results are computed with optimal kernels with the same complexity level GPL-2.

(4) Repeat the calculations until the predictions do not change with the change of the training point distributions.

VI. CONCLUSION

In the present work, we show that Gaussian process models can be designed to extrapolate the physical properties of complex quantum systems for predictions outside the range of the Hamiltonian parameters used for training the models. The particular focus of this work has been to predict (multiple) quantum phase transitions based on system properties in a given phase. The previous ML approaches to identifying phase transitions based on NNs rely on the knowledge that there are multiple phases. The present method is different from the previous approaches in that it does not require any information about the other phases and can be used to search for phase transitions without a priori knowledge of the existence of the phase transitions. This is particularly important for models that do not permit theoretical solutions or experimental measurements for a certain range of the Hamiltonian parameters. The approach presented here can be used to describe the physical properties of the system in this range.

The present method can also be used to guide rigorous quantum calculations in search of phase transitions and/or particular properties of quantum systems. Generating the phase diagram, such as the one depicted in Figure 3, presents no computational difficulty (taking essentially minutes of CPU time), while numerical quantum calculations, even if feasible, may take hours, to days, to months of CPU time. One can thus envision the following efficient approach for the generation of the full phase diagrams based on a combination of the present method with rigorous calculations:

(1) Start with a small number of rigorous quantum calculations.

(2) Generate the full phase diagram with the present extrapolation method. This diagram is likely to be inaccurate at the system parameters far away from the initial training points.

(3) Use the rigorous quantum calculations to add training points in the parts of the parameter space, where (a) the system exhibits desired properties of interest; and (b) where the system properties undergo the most rapid change.

(4) Repeat the calculations until the extrapolation predictions do not change with the change of the training point distributions.

With this approach, one can envision generating complete $\mathcal{D}$-dimensional phase diagrams with about $10 \times \mathcal{D}$ rigorous quantum calculations. Training the extrapolation models and making the extrapolation predictions in step (2) will generally take a negligibly small fraction of the total computation time.

Finally, the results presented in this work suggest algorithms to construct complex GP models capable of meaningful extrapolation without validation. To do this, one can examine the sensitivity of the predictions to the distribution of the training points for models with the same level of kernel complexity as well as models with different complexity. Increase of the sensitivity to the training points with the kernel complexity would suggest overfitting or insufficient optimization of the kernel parameters. In such cases, the iterative process building up the kernel complexity should be stopped or the process of optimizing the kernel parameters revised. Constructing algorithms for physical extrapolation without the need for validation should be the ultimate goal of the effort aimed at designing ML models for physics and chemistry. Such models could then use all available chemistry and physics information to make new discoveries.

ACKNOWLEDGMENTS

We thank Mona Berciu for the quantum results used for training and verifying the ML models for the polaron problem. We thank John Sous and Mona Berciu for the ideas that have led to work published in Ref.[45] and for enlightening discussions.

REFERENCES

[1]J. N. Murrell, S. Carter, S. C. Farantos, P. Huxley, and A. J. C. Varandas, *Molecular Potential Energy Functions*, Wiley, Chichester, England, 1984.

[2]T. Hollebeek, T. -S. Ho, and H. Rabitz, *Annu. Rev. Phys. Chem.***50**, 537 (1999).

[3]B. J. Braams, and J. M. Bowman, *Int. Rev. Phys. Chem.***28**, 577 (2009).

[4]M. A. Collins, *Theor. Chem. Acc.* **108**, 313 (2002).

[5]C. M. Handley, and P. L. A. Popelier, *J. Phys. Chem. A*. **114**, 3371 (2010).

[6]S. Manzhos, and T. Carrington Jr. *J. Chem. Phys.* **125**, 194105 (2006).

[7]J. Cui, and R. V. Krems, *Phys. Rev. Lett.* **115**, 073202 (2015).

[8]J. Cui, and R. V. Krems, *J. Phys. B* **49** , 224001 (2016).

[9]R. A. Vargas-Hernández, Y. Guan, D.H. Zhang, and R. V. Krems, arXiv preprint arXiv:1711.06376

[10]A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington Jr, S. Manzhos *J. Chem. Phys.* **148**, 241702 (2018).

[11]C. Qu, Q. Yu, B. L. Van Hoozen Jr., J. M. Bowman, and R. A. Vargas-Hernández, *J. Chem. Theory Comp.* **14**, 3381 (2018).

[12]L. Wang, *Phys. Rev. B* **94**, 195105 (2016).

[13]J. Carrasquilla, and R. G. Melko, *Nat. Phys.* **13**, 431 (2017).

[14]E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, *Nat. Phys.* **13**, 435 (2017).

[15]P. Broecker, F. Assaad, and S. Trebst, arXiv:1707.00663.

[16]S. J. Wetzel, and M. Scherzer, *Phys. Rev. B* **96**, 184410 (2017).

[17]S. J. Wetzel, *Phys. Rev. E* **96**, 022140 (2017).

[18]Y.-H. Liu, and E. P. L. van Nieuwenburg, *Phys. Rev. Lett.* **120**, 176401 (2018).

[19]K. Chang, J. Carrasquilla, R. G. Melko, and E. Khatami, *Phys. Rev. X* **7**, 031038 (2017).

[20]P. Broecker, J. Carrasquilla, R. G. Melko, and S. Trebst, *Sci. Rep.* **7**, 8823 (2017).

[21]F. Schindler, N. Regnault, and T. Neupert, *Phys. Rev. B* **95**, 245134 (2017).

[22]T. Ohtsuki, and T. Ohtsuki, *J. Phys. Soc. Japan* **85**, 123706 (2016).

[23]L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, *Phys. Rev. B* **90**, 155136 (2014).

[24]L.-F. Arsenault, O. A. von Lilienfeld, and A. J. Millis, arXiv:1506.08858.

[25]M. J. Beach, A. Golubeva, and R. G. Melko, *Phys. Rev. B* **97**, 045207 (2018).

[26]E. van Nieuwenburg, E. Bairey, and G. Refael, *Phys. Rev. B* **98**, 060301(R) (2018).

[27]N. Yoshioka, Y. Akagi, and H. Katsura, *Phys. Rev. B* **97**, 205110 (2018).

[28]J. Venderley, V. Khemani, and E.-A. Kim, *Phys. Rev. Lett.* **120**, 257204 (2018).

[29]G. Carleo, and M. Troyer, *Science* **355**, 602 (2017).

[30]M. Schmitt, and M. Heyl, *SciPost Phys.* **4**, 013 (2018)

[31]Z. Cai, and J. Liu, *Phys. Rev. B* **97**, 035116 (2017).

[32]Y. Huang, and J. E. Moore, arXiv:1701.06246.

[33]D.-L. Deng, X. Li, and S. D. Sarma, *Phys. Rev. B* **96**, 195145 (2017).

[34]Y. Nomura, A. Darmawan, Y. Yamaji, and M. Imada, *Phys. Rev. B* **96**, 205152 (2017).

[35]D.-L. Deng, X. Li, and S. D. Sarma, *Phys. Rev. X* **7**, 021021 (2017).

[36]X. Gao, and L.-M. Duan, *Nat. Commun.* **8**, 662 (2017).

[37]G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, *Nat. Phys.* **14**, 447 (2018).

[38]T. Hazan, and T. Jaakkola, arXiv:1508.05133.

[39]A. Daniely, R. Frostig, and Y. Singer, *NIPS* **29**, 2253 (2016).

[40]J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, Deep Neural Networks as Gaussian Processes *ICLR* (2018).

[41]K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K. R. Mller, and E. K. U. Gross, *Phys. Rev. B* **89**, 205118 (2014).

[42]L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, . Draxl, and M. Scheffler, *Phys. Rev. Lett.* **114**, 105503 (2015).

[43]F. A. Faber, A. Lindmaa, O. A, von Lilienfeld, and R. Armient *Int. J. Quantum Chem.* **115**, 1094 (2015).

[44]F. A. Faber, A. Lindmaa, O. A, von Lilienfeld, and R. Armient *Phys. Rev. Lett.* **117**, 135502 (2016).

[45]R. A. Vargas-Hernández, J. Sous, M. Berciu, R. V. Krems, arXiv:1803.08195 (to be published in *Phys. Rev. Lett.*)

[46]N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *J. Mach. Learn. Res.* **15**, 1929 (2014).

[47]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).

[48]B. Lau, M. Berciu, and G. A. Sawatzky, *Phys. Rev. B* **76**, 174305 (2007).

[49]F. Herrera, K. W. Madison, R. V. Krems, and M. Berciu, *Phys. Rev. Lett.* **110**, 223002 (2013).

[50]B. Gerlach, and H. Löwen, *Rev. Mod. Phys.* **63**, 63 (1991).

[51]P. M. Chaikin, and T. C. Lubensky *Principles of condensed matter physics*, Cambridge University Press, Cambridge (1998).

[52]S. Sachdev, *Quantum phase transitions*, Cambridge University Press, Cambridge (1999).

[53]C. E. Rasmussen, and C. K. I. Williams, *Gaussian Process for Machine Learning*. MIT Press, Cambridge (2006).

[54]J. Cui, Z. Li, and R. V. Krems, *J. Chem. Phys.* **143**, 154101 (2015).

[55]D. Vieira, and R. V. Krems, *Ap. J.* **835**, 255 (2017).

[56]G. Schwarz, *The Annals of Statistics* **6**(2), 461 (1978).

[57]D. K. Duvenaud, H. Nickisch, and C. E. Rasmussen, *Advances in Neural Information Processing Systems* **24**, 226 (2011).

[58]D. K. Duvenaud, J. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, *Proceedings of the 30th International Conference on Machine Learning Research* **28**, 1166 (2013).

[59]R. S. Sutton, and A. G. Barto, *Reinforcement Learning, An Introduction*. MIT Press, Cambridge (2016).

[60]A. Christianen, T. Karman, R. A. Vargas-Hernández, G. C. Groenenboom, and R. V. Krems, *submitted to J. Chem. Phys.* (2018).

[61]J. Snoek, H. Larochelle, and R. P. Adams, *NIPS*, pages 2951-2959, (2012).

[62]B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, *Proceedings of the IEEE* **104** (1), 148-175, (2016).