Accurate Machine Learning of Rate Coefficients for State-to-State Transitions in Molecular Collisions

Darin E. Mihalik,[1] R. Wang, B. H. Yang, P. C. Stancil,[1] T. J. Price, R. C. Forrey,[2] N. Balakrishnan,[3] and R. V. Krems[4]

[1)]*Department of Physics and Astronomy and the Center for Simulational Physics, University of Georgia, Athens, GA 30602*

[2)]*Department of Physics, Penn State University, Berks Campus, Reading, PA 19610*

[3)]*Department of Chemistry and Biochemistry, University of Nevada, Las Vegas, NV 89154*

[4)]*Department of Chemistry, University of British Columbia, Vancouver, BC Canada, V6T 1Z1*

(*Electronic mail: mihalid@sunysuffolk.edu)

(Dated: 2 October 2024)

We present an algorithm that combines quantum scattering calculations with probabilistic machine-learning models to predict quantum dynamics rate coefficients for a large number of state-to-state transitions in molecule - molecule collisions much faster than with direct solutions of the Schrödinger equation. By utilizing the predictive power of Gaussian process regression with kernels, optimized to make accurate predictions outside of the input parameter space, the present strategy reduces computational cost by about 75%, with an accuracy within 5%. Our method uses temperature dependences of rate coefficients for transitions from isolated states of initial rotational angular momentum j, determined via explicit calculations, to predict temperature dependences of rate coefficients for other values of j. The approach, demonstrated here for ro-vibrational transitions of SiO due to thermal collisions with H_2, uses different prediction models and is thus adaptive to various time and accuracy requirements. The procedure outlined in this work can be used to extend multiple inelastic molecular collision databases without the exponentially large computation resources required for conventional rigorous quantum dynamics calculations.

I. INTRODUCTION

Full-dimensional state-resolved quantum dynamics calculations have been shown to produce highly accurate inelastic molecular collision rates[1–9]. As the kinetic and internal energies of the collision partners increase, so does the computational cost to produce these results. The quantum scattering problem for molecular collisions rapidly becomes more difficult as the rotational quantum number j and the total angular momentum grow. One can reduce the complexity of the problem by decoupling various angular momenta and/or freezing one or more of the internal coordinates. One such approach is known as the coupled-states (CS) approximation, which replaces the centrifugal barrier with an effective average and decouples the orbital angular momentum from the remaining sources of angular momenta. For a diatom-diatom system, the full six dimensional close-coupling (6DCC) formulation may be replaced by a 6DCS approximation[10,11] if the twist-angle dependence of the molecule - molecule potential energy surface (PES) is not strong. It is also possible to introduce an additional approximation which reduces the scattering problem to five dimensions (5DCS)[12]. Although less accurate than the numerically exact 6DCC calculations, the 5DCS method has produced results in good agreement with experimental measurements[7–9,12–15]. While this approximation reduces the computational complexity of the scattering problem, the computation time required for the 5DCS calculations scales with the maximum value of the initial rotational quantum number j in the basis set leading to the computation time scaling[12],

$$t_{\rm CS} \propto j_{\rm max}^4 \quad \text{vs} \quad t_{\rm CC} \propto j_{\rm max}^6. \tag{1}$$

This $j_{\rm max}^4$ scaling still restricts the CS calculations to a limited range of ro-vibrational states of molecules. At the same time, astrophysical models require rate coefficients for state-to-state transitions in molecular collisions for a wide range of initial and final quantum states, extending to high energies. Here, we present a data-generation approach that replaces CS calculations for some state-to-state transitions with surrogate machine learning models using kernel-based, non-parametric, probabilistic regression. Our goal is to determine which (isolated) quantum states must be addressed by the CS calculations in order to build the most optimal machine learning (ML) models that can predict the most accurate set of rate coefficients for state-to-state transitions which span the entire range of accessible quantum states over a given interval of temperatures.

ML algorithms have recently found applications in numerous disciplines of physics including nuclear physics[16–18], particle physics[19–21], astronomy[22–24], cosmology[25–27], condensed matter physics[28–30], and quantum many-body physics[31–33]. In the present work, we use Gaussian process (GP) regression for the prediction of the rate coefficients. In a previous related study[23], it was demonstrated that GP models could be used to correct some of the errors contained in approximate 5DCS cross sections using a training set consisting of a small number of accurate 6DCC cross sections. The utility of this approach depends on the availability of a large amount of approximate data, which as noted above, may be impractical to compute. Here, we employ GPs with variable functional forms of the kernel functions to build models that maximize inference from a given, limited data set of rate coefficients. This is achieved by increasing the complexity of GP models in order to align GP kernels with target functions to be learned[34,35]. A similar approach has been shown to pro-

duce accurate results for physical extrapolation, i.e. by generalization of observables outside the range of input variables, in several applications[36–42]. This prediction power of GPs is exploited here to construct models with variable kernels selected to reduce the number of quantum dynamics calculations required to produce accurate rate coefficients for molecule - molecule collisions over a wide range in j. The resulting models are trained, and validated, by transitions with only a few isolated j values and used to predict temperature dependencies of rate coefficients for transitions with other j values that are not included in the training set. The training sets for such models are also optimized in order to produce an optimal interplay of CS calculations and ML surrogate models for accurate predictions of temperature dependencies of rate coefficients for a wide range of quantum states.

II. METHODS

The purpose of a GP model is to infer a target function $f(\boldsymbol{x})$ given n inputs $\boldsymbol{x}_i$ and corresponding output values y_i. GP regression uses the conditional distribution $p(f|\{\mathbf{X},\boldsymbol{y}\})$ of a normal distribution of functions, where $\mathbf{X}$ is the data matrix collecting the input vectors $\boldsymbol{x}_i$ and $\boldsymbol{y}$ is a vector of the corresponding outputs y_i. For a noiseless dataset, $y_i = f(\boldsymbol{x}_i)$. A GP is defined by a mean function $\mu(\boldsymbol{x})$ and covariance matrix $\mathbf{K}$. The matrix elements K_{ij} of $\mathbf{K}$ are given by the values of a kernel function $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$. Predictions are made by calculating the mean of the conditional distribution at value $\boldsymbol{x}_*$ given, in the absence of noise on y_i, by[43]

$$\mu(\boldsymbol{x}_*) = \mathbf{k}(\boldsymbol{x}_*)^\top \mathbf{K}^{-1}\boldsymbol{y}. \tag{2}$$

where $\mathbf{k}(\boldsymbol{x}_*)$ is a vector of $k(\boldsymbol{x}_*,\boldsymbol{x}_j)$.

The choice of $k(\boldsymbol{x}_i,\boldsymbol{x}_j)$ determines the efficiency of the model. Commonly used analytical forms of the kernel functions include the following:

$$k_{\mathrm{LIN}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 \boldsymbol{x}_i^\top \boldsymbol{x}_j, \tag{3}$$

$$k_{\mathrm{RBF}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 \exp\left[-\frac{r^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{2\ell^2}\right], \tag{4}$$

$$k_{\mathrm{EXP}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 \exp\left[-\frac{r(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\ell}\right], \tag{5}$$

$$k_{\mathrm{MAT}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\ell}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\ell}\right), \tag{6}$$

and

$$k_{\mathrm{RQ}}(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2 \left(1 + \frac{r^2(\boldsymbol{x}_i,\boldsymbol{x}_j)}{2\alpha\ell^2}\right)^{-\alpha}, \tag{7}$$

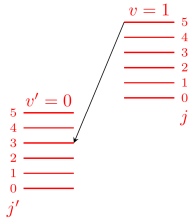

FIG. 1. Energy level diagram illustrating SiO relaxation from an initial state $v = 1, j = 5$ to a final state $v' = 0, j' = 3$. The plot is not scaled vertically in energy and the $v = 1, j = 0$ state energy is not comparable in energy to the $v' = 0, j' = 5$ state.

where $r^2(\boldsymbol{x}_i,\boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^2$ is the squared Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. For k_{MAT}, ν is a positive parameter that defines the kernel with values $\nu = 1/2$, $3/2$, and $5/2$ (which will be denoted k_{12}, k_{32}, k_{52}), K_ν is the modified Bessel function of the second kind, and for k_{RQ}, α is the scale-mixing free parameter[43].

Instead of restricting the choice of the kernel function to a single kernel, in this work we use a combination of Eqs. (3) - (7), constructed as described below. Moreover, we use anisotropic kernel functions, i.e. each kernel function is allowed an independent length-scale parameter per input dimension. Collectively, the free parameters of the chosen kernel functions can be represented as $\boldsymbol{\theta}$. The kernel function parameters $\boldsymbol{\theta}$ are estimated by maximizing the logarithm of the marginal likelihood function[43]

$$\log p(\boldsymbol{y}|\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\top \mathbf{K}^{-1}\boldsymbol{y} - \frac{1}{2}\log|\mathbf{K}| - \frac{n}{2}\log(2\pi), \tag{8}$$

where n is the number of input points and $\mathbf{K}$ and $\boldsymbol{y}$ are defined above.

We consider inelastic state-resolved diatom-diatom scattering between SiO molecules in a ro-vibrationally excited state and para-H_2 molecules in the ground state. For astrophysics applications, the rotational and vibrational state of para-H_2 are fixed to $v = 0$, $j = 0$, as a significant fraction of H_2 in the interstellar medium is expected to populate this state. The same approach can be applied to H_2 in its lowest ortho state, i.e. $j = 1$. The GP models are trained by rate coefficients for collision-induced relaxation, i.e. transitions depicted by the energy level diagram in Figure 1. The rate coefficients are found by thermal averaging the collisional cross sections over a Maxwellian velocity distribution,

$$k(T) = \sqrt{\frac{8k_BT}{\pi\mu}} \frac{1}{(k_BT)^2} \int_0^\infty E_c \sigma(E_c) e^{-\frac{E_c}{k_BT}} dE_c, \tag{9}$$

where E_c is the collision energy, $\sigma(E_c)$ is the cross section for a particular transition at a given E_c, μ is the reduced mass of the SiO+H_2 system, k_B is the Boltzmann constant, and T is the temperature in Kelvin. The collision cross sections, σ, are calculated by the 5DCS method.

We use rate coefficients, instead of the cross sections, for ML input because the integration in Eq. (9) replaces the energy dependence of the cross sections, often exhibiting sharp

resonance features, with a smoother temperature dependence of the rate coefficients. We expect these smoother functions to be easier to learn. This is also important because resonant scattering is often sensitive to small variations of the underlying interaction potentials and details of the calculations. The approximations of the 5DCS method may result in shifts of resonances, leading to large errors in cross sections, which may not manifest themselves in the rate coefficients. Applying ML to rate coefficients rather than cross sections represents a change in strategy over a previous work[23] which aimed to model the energy dependence of cross sections with energy-sparse data. The present approach requires 5DCS cross sections to be computed with sufficient resolution over a large range of energies for all transitions used in the training set. Moreover, it is the rate coefficients that are used in kinetic astrophysical models.

The input vectors for GP models of rate coefficients are built using the following physical parameters that can classify a state-to-state transition:

$$\boldsymbol{x} = \left[T, \Delta v, \Delta j, v, j, v', j', \delta_{\Delta j}, \delta_{\Delta v}\right], \tag{10}$$

where v, j and v', j' represent the vibrational and rotational quantum numbers of the initial and final state of the SiO molecule, respectively; T is the temperature ranging from 10 to 1000 K by decade, or nineteen temperatures in total; Δv and Δj are the differences between the final and initial state vibrational and rotational quantum numbers, $\Delta v = v' - v$ and $\Delta j = j' - j$. The last two features $\delta_{\Delta j}$ and $\delta_{\Delta v}$ reflects the well-known differences in propensities for different types of internal energy transfer and are defined as

$$\delta_{\Delta j} = \begin{cases} 0 & \Delta j \text{ even} \\ 1 & \Delta j \text{ odd}, \end{cases} \tag{11}$$

and

$$\delta_{\Delta v} = \begin{cases} 0 & \Delta v = 0 \\ 1 & \Delta v \neq 0. \end{cases} \tag{12}$$

We have found that these two features play a critically important role for the models of state-to-state transitions.

The goal of the present data generation scheme is two-fold. First, we aim to develop robust models that greatly reduce the need to explicitly perform approximate quantum dynamics calculations for each value of j, j', v, and v'. Second, we aim to limit the amount of training data needed. Due to the computational expense of the explicit calculations, the goal is to build a framework that requires less transitions to build the models than the number of transitions predicted by the models, without compromising the accuracy of the predictions.

To develop this data prediction scheme, it must be assumed that most of the rate cofficient data are unavailable. The current database of 5DCS transitions for SiO(v, j) – H_2 collisions includes rate coefficients for transitions ranging from $j_{\min} = 0$ to $j_{\max} = 20$ with $j'_{\min} = 0$ and $j'_{\max} = 20$ for each value of j as well as, $v = 0, 1, 2$ to all allowed values of v' based on the de-excitation requirement. By dividing the current data into sets of transitions based on j and assuming j = 5 is the highest j calculated explicitly, it can be determined which additional sets higher than $j = 5$ must be calculated explicitly to train an accurate ML prediction model. The accuracy of the models is investigated as a function of the number of different sets of j used to train the model as well as the specific values of j that are included in the training set.

To achieve this, input training sets are created with all possible unique sets of j from the current database of 5DCS results. The training sets have a lower limit of $\mathcal{N} = 2$ and an upper limit $\mathcal{N} = 6$, where $\mathcal{N}$ is the number of different j values in the training set. The upper limit of $\mathcal{N} = 6$ is due to the requirement that more data be predicted than used to train the model. If a particular value of j is included in the training, all of the transitions in the set with that j are used to train the model. Each unique input set is then used to train a GP model with one of seven kernels given by Eqs. (3) - (7) using GPflow[44,45]. The seven base kernels include Eq. (6) with three values of ν for k_{MAT}.

For each $\mathcal{N}$, the input combination that produces the lowest overall test error is chosen as the input for a particular model, $\mathcal{M}_{\mathcal{N}}$. The test sets in this work includes all transitions with values of j that are not included in the training set. This means that the test sets are different for different $\mathcal{N}$, however, the test sets are larger than the training sets. It is important to note that the test error is only used as a guide to determine the spread of the j values that need to be calculated via explicit quantum dynamics for the training of the models. Datasets are not built iteratively upon the optimal combination of the previous dataset as it was found that this does not guarantee the successive iteration to be an optimal one.

It is clear from Eqs. (3)-(7) that GPs are sensitive to the specific choice of the kernel function. As shown in Refs.[34,35,41], given a fixed data set, the model performance can be improved by increasing the complexity of the kernel function. The algorithm of[34,35] begins by testing models with a simple kernel from a given set of simple functions. The kernel complexity is then increased through an iterative process by selecting the best performing kernel from the previous iteration and composing new unique kernels using sums and products of this kernel with the same predetermined set of kernels. For a set of seven functions in Eqs. (3) – (7), the unique kernels at the i-th iteration are $k_i = k_{i-1} + k_{\text{LIN}}$, ..., $k_i = k_{i-1} + k_{\text{RQ}}$, $k_i = k_{i-1} \times k_{\text{LIN}}$, ..., $k_i = k_{i-1} \times k_{\text{RQ}}$. A new GP model is then trained with each of the new fourteen possible kernel combinations, again choosing the best performing kernel for the model. This process is repeated until the best performing kernel of a successive iteration is comparable, or worse, to that of the previous iteration in which case the kernel from the previous successful iteration is chosen as the optimal kernel. In this work, the first step of the kernel construction algorithm is done concurrently with determining the input training sets, as described above. Each of the five models $\mathcal{M}_{\mathcal{N}}$, where $\mathcal{M}_{\mathcal{N}}$ encompasses the optimal input combinations and their associated single kernel, $k_{\mathcal{N}}^{\text{single}}$, are then individually put through the kernel building algorithm until the optimal kernel $k_{\mathcal{N}}^{\text{opt}}$ is determined. This process is shown schematically in Figure 2.

Refs.[34,35,41] used the Bayesian information criterion (BIC)

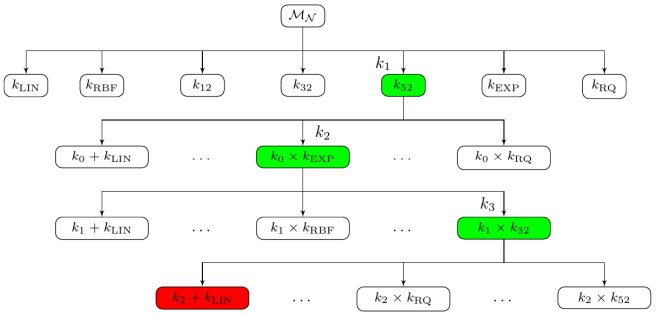

FIG. 2. Schematic of the kernel building algorithm over three iterations. The input from each $\mathcal{M}_\mathcal{N}$ is tested with each of the seven base kernels. The best kernel from the first iteration using only a single kernel, $k_1 = k_\mathcal{N}^{\rm single}$, is chosen using a selected model selection criteria. Fourteen unique kernels are then composed via sums and products of the seven base kernels with the best kernel from the first iteration. This process continues until the model selection criteria of a subsequent run is the same or is worse than the previous iteration in which case the kernel from the previous successful iteration is chosen as the optimal kernel. This figure shows the process for building the optimal kernel after three iterations $k_3 = k_\mathcal{N}^{\rm opt} = k_{52} \times k_{\rm EXP} \times k_{32}$.

as the model selection metric for each iteration of the kernel construction algorithm. In the present work, we find that the model with the largest value of BIC does not always produce the lowest test error. To overcome this issue and develop a method that can be applied to generating new data, we use error over a validation set instead of BIC for model selection. The validation set must be based on transitions with values of j not included in the training set. Given that this adds to the number of quantum calculations required, the validation set should be as small as possible and include lower values of j.

For each $\mathcal{M}_\mathcal{N}$, we use three different validation sets for the kernel selection process depicted in Figure 2. We have found that, if there is a minimum of three units between the values of j in the validation set, then the validation set error correlates with the corresponding test error. To construct the validation sets, we start with the following values of $j = \{6,9\}$, $\{6,12\}$, $\{6,9,12\}$. If one of these values is already included in the training set, the closest higher value is used. For example, $\{6,10,13\}$ is used as the validation set if $j = 9$ and $j = 12$ are used in the training set. The procedure outlined in Figure 2 continues iteratively until the validation error of a subsequent run remains the same or increases, in which case the kernel of the previous iteration is chosen.

All data in this work is reported as the common logarithm of the rate coefficient which allows the reported errors to be given as a relative percentage. With this, the desired prediction accuracy of these models is better than 5% with an upper limit of 10%. This is an accuracy range that is suitable for use in kinetic astrophysical models.

III. RESULTS

The results of optimization of the training sets are shown in Table I. The optimal input combination for each $\mathcal{M}_\mathcal{N}$ is the combination of transitions that produces the lowest test error, while the optimal kernel is determined with the validation sets as described above. Also included in Table I is the number of individual state-to-state transitions used to train each of the models.

$\mathcal{N}$	Values of j in Training	$k_\mathcal{N}^{\rm single}$	Num. of Transitions in Input
2	5, 18	$k_{\rm EXP}$	195
3	5, 16, 19	k_{52}	309
4	5, 9, 14, 19	k_{52}	393
5	5, 8, 12, 17, 19	k_{52}	498
6	5, 8, 11, 14, 17, 19	k_{32}	600

TABLE I. Results for the optimal input combination of j values for each $\mathcal{M}_\mathcal{N}$ with the corresponding optimal single kernel, $k_\mathcal{N}^{\rm single}$. The input combination that returned the lowest test error was chosen to be the input combination for $\mathcal{M}_\mathcal{N}$ which is a guide for selecting which j values should be calculated via explicit quantum dynamics for the input. The optimal kernel was determined via the lowest validation set error using the kernel building algorithm. Only one kernel is reported per $\mathcal{N}$ due to the fact that the models chosen using all three validation sets for a given $\mathcal{N}$ returned the same result for the optimal kernel with slightly different numerical results.

Values of j in $\mathcal{N} = 5$ Training Set	Single Kernel Test Error
5, 8, 12, 17, 19	2.04%
5, 8, 13, 15, 19	2.10%
5, 8, 13, 17, 19	2.21%
5, 8, 11, 17, 19	2.56%
5, 9, 11, 16, 18	2.62%
5, 7, 13, 17, 19	2.89%
5, 8, 12, 16, 19	3.57%

TABLE II. Input combinations for the model $\mathcal{M}_5$ that deviate from the optimal input combination for $\mathcal{N} = 5$ along with the corresponding test error for that model. The first result shown in this table is the optimal combination as shown in Table I. Each of these results were produced with the same optimal kernel, $k_{\mathcal{N}=5}^{\rm single} = k_{52}$.

Table I illustrates that the most optimal input combinations have no adjacent j. This shows the GP models generalize well to transitions with values of j close to those used in the training set. Transitions with $j < 5$ are not included in any of the input sets as they do not improve the results. One should not regard this as a general rule as this could be due in part to this particular situation where these j values are too small to allow the model to make inferences about transitions involving higher values of j. It is also important to note that small deviations from the input values of j for each model do not result in large deviations of the test error compared to the optimal model. This shows that models are not very sensitive to the specific values of j, provided the values of j used in the training set have similar separations as the sets reported in Table I. Some combinations close to our optimal input combination for $\mathcal{M}_5$, but with slightly different j values are shown in Table II with the corresponding test errors.

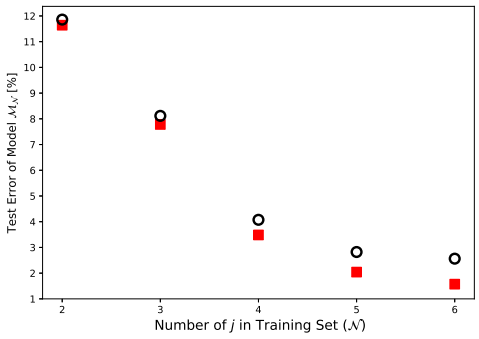

FIG. 3. Test errors for both a single kernel ($k_\mathcal{N}^{\text{single}}$) and the optimal kernel ($k_\mathcal{N}^{\text{opt}}$) after completion of the kernel complexity building algorithm for all $\mathcal{M}_\mathcal{N}$. The hollow black circles represent the test error with $k_\mathcal{N}^{\text{single}}$ and the red squares represent the test error using $k_\mathcal{N}^{\text{opt}}$. The errors reported are the test errors from models chosen using the validation error as model selection metric.

Despite using three different validation sets for each $\mathcal{M}_\mathcal{N}$ only one kernel is reported for each case in Table I. This is because each of the three validation sets used to build each $\mathcal{M}_\mathcal{N}$ returned the same kernel for all iterations of the kernel building algorithm with slightly different numerical results. This indicates that the validation error, as designed here, is a good selection metric. Since we aim to reduce the validation set as much as possible and restrict it to low j, the first validation set, 6 and 9, or the closest higher value of j when necessary, is used in the rest of this work. The composite kernel construction algorithm is then completed for each $\mathcal{M}_\mathcal{N}$ using only the chosen validation set for model selection. Figure 3 shows the test errors for each $\mathcal{M}_\mathcal{N}$ for both $k_\mathcal{N}^{\text{single}}$ and $k_\mathcal{N}^{\text{opt}}$, demonstrating the effect of the kernel complexity on model performance.

Figure 3 shows that increasing from $\mathcal{N}=2$ to $\mathcal{N}=6$ results in a rapid decrease in the test error. However, as the number of j values used to train the models increases the accuracy gained per step decreases. This indicates convergence on the number of j values in the training set as the benefit of performing additional explicit calculations decreases rapidly. Consequently, this is beneficial as the time and computational expense required to train a GP model increases nonlinearly with the number of training points. $\mathcal{N}=6$ was set as the maximum number of j values in the training set to ensure that more data was predicted by the models than was being used to train them. Figure 3 indicates that this is a logical maximum due to the minimal gain in accuracy one would obtain by performing more explicit calculations. The largest improvement in accuracy comes by adding one additional value of j from $\mathcal{N}=3$ to $\mathcal{N}=4$, in which the target accuracy of 5% or less is achieved.

Figure 3 shows that increasing the kernel complexity consistently improves the model accuracy. The increase in model accuracy is even more evident in Figure 4 that shows the predictions for each individual set of j for $k_\mathcal{N}^{\text{single}}$ and $k_\mathcal{N}^{\text{opt}}$ for $\mathcal{N}=3$ and $\mathcal{N}=6$. Since we limit quantum transitions to de-excitation, each increase in j opens more transitions. Figure 4 shows that, for $\mathcal{N}=6$ the optimal model performs better than the single kernel results for every value of j. The dependence on the kernel functional form is more mixed for $\mathcal{N}=3$. This can be attributed to the model having trouble generalizing for the higher j predictions. For this model, the individual test errors for $j=15, 17,$ and 18 actually increase with an increase in kernel complexity which combined with a large improvement in the lower values of j leads to only a small increase in the overall accuracy of the optimal model. We find that for $\mathcal{N}\geq 4$ (not shown) there are no outliers in any of the test errors for any individual values of j. This rules out that any of the results above this point are skewed incidentally. The red horizontal lines in Figure 4 show the lowest accuracy of the prediction of an individual transition for each value of j in the test set.

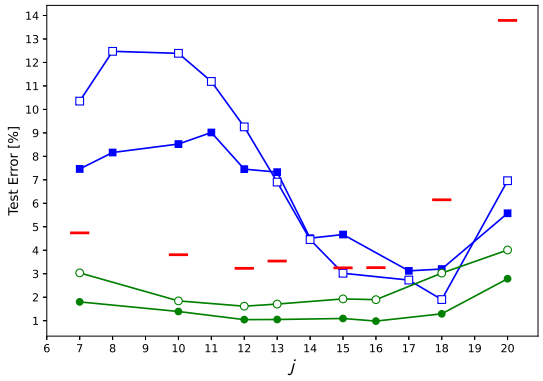

FIG. 4. Test errors for individual values of j representing sums of errors for all inelastic transitions in collisions of SiO(v, j) with H_2 that are not included in the training set. The squares represent the $\mathcal{M}_3$ results, while the circles are the $\mathcal{M}_6$ results. The full symbols show the results obtained with simple kernels and the open symbols – with composite kernels. The red horizontal lines show the lowest accuracy of the prediction of an individual transition for each value of j in the test set for the $\mathcal{M}_6$ model. If a certain data point for a given $\mathcal{N}$ is not shown in the plot, the corresponding value of j is used for training or validating that model.

Table III shows the optimal kernels that were determined via the kernel building algorithm and used to produce the results in Figure 3. It reveals the uniqueness of each kernel as no two optimal kernels were the same. This reinforces that each input combination has a preferred kernel as opposed to one globally applicable kernel. It was found during the single kernel analysis for $\mathcal{M}_4$ that k_{RQ} had the lowest validation error however, the kernel complexity could not be increased as the validation errors in the next iteration of the algorithm were higher for each model. The single kernel cases of $\mathcal{M}_5$ and $\mathcal{M}_6$ were also tested with k_{RQ} as these models had a low, but not the lowest, validation error. Again, these cases showed that the kernel complexity could not be increased beyond the single kernel. The decision to not use this kernel came from the fact that for each of the models tested, the final result using a kernel that was able to become more complex was better than the k_{RQ} case by 1% to 1.5% in all cases. This is not a large difference from the reported final answer and after the full analysis it is clear that one could use this kernel to make

$\mathcal{N}$	Optimal Kernel	Optimal Kernel Test Error
2	$k_{\mathrm{EXP}} + k_{\mathrm{EXP}}$	11.64%
3	$k_{52} \times k_{12} \times k_{\mathrm{EXP}} + k_{12}$	7.78%
4	$k_{52} \times k_{\mathrm{EXP}} + k_{\mathrm{EXP}} * k_{32}$	3.49%
5	$k_{52} \times k_{\mathrm{EXP}} + k_{32}$	2.04%
6	$k_{32} \times k_{\mathrm{EXP}} + k_{\mathrm{RBF}} + k_{12}$	1.57%

TABLE III. Optimal kernels used to make predictions for the results in Figure 3 for all $\mathcal{M}_{\mathcal{N}}$. The first kernel for each $\mathcal{N}$ is the kernel reported in Table I. These kernels were all determined using an RMSProp optimizer with a learning rate of $\eta = 1$ determined via grid search to be the ideal optimizer.

predictions if this decrease in accuracy is acceptable.

Table IV gives a quantitative summary of the number of individual state-to-state transitions that were used to validate the models as well as the number of transitions predicted by the model. Also shown is the number of individual state-to-state transitions, for both the single and optimal kernel results, that exceed the maximum accuracy target of 10% and those transitions that had a better accuracy than the target of 5%. Table IV shows that in order to achieve an overall prediction accuracy of under 10%, 5% and 3% only about a quarter, a third, and a half of the total number of transitions in the test set are needed to train the models, respectively. To validate the models, only between 15% and 18% of the predicted number of transitions were needed for the models which achieved the desired accuracy targets. The reason for the slightly larger increase in validation set size for $\mathcal{N} = 4$ is that $j = 9$ was used in the training so $j = 10$ was reserved for the validation. Additionally, we observe that increasing the kernel complexity from $k_{\mathcal{N}}^{\mathrm{single}}$ to $k_{\mathcal{N}}^{\mathrm{opt}}$ decreases the number of individual state-to-state transitions with test errors larger than 10%, while increasing the number of transitions with errors smaller than 5%. This is the case for all $\mathcal{M}_{\mathcal{N}}$ except $\mathcal{M}_2$ that shows a slight decrease in some transitions below 5%. It is clear that the kernel complexity does improve the overall performance by increasing the number of transitions below 5%. The models with a single optimal kernel, however, already perform well, particularly for $\mathcal{N} \geq 3$.

For $\mathcal{N} \geq 4$, all of the transitions with errors above 10% are due to $(v,20) \rightarrow (v',0)$ transitions because $\Delta j = -20$ transitions are outside the scope of the input parameter space. This is evident from Figure 4 for $j = 20$ in which the highest individual error is roughly 14% which occurs for a $\Delta j = -20$ transition. Neglecting these transitions, the highest error for an individual prediction for $j = 20$ is 6.16%. This shows that these transitions clearly do not have an effect on this data prediction scheme other than contributing some additional errors but should be taken into consideration in future work.

Table V gives an overview of the time required, in hours, to generate the data in the test set using this data generation scheme versus the time required to produce the same data using only explicit calculations. Since this data generation scheme requires some explicit calculations to be performed, the time required to produce the data used to train and validate the models is reported separately from the time to produce the data in the test set. The time required to explicitly generate all of the approximate data used in this analysis is 383 hours. This includes the data used in the training, validation and test sets. Since the data generated from the calculations is completed sequentially it is unknown exactly when the calculation changes from one value of j to the next. To get a rational estimate of the time required to generate the transitions for a particular j, each calculation time of j^4 is normalized to the total time. The reported 5DCS computation times in Table V are from this normalization and are the sum of the times required to explicitly generate data for the individual values of j for the respective sets. The time required to run the GP algorithm is reported as the times needed to complete the analysis for a single kernel and the time to complete the optimal models through the entire kernel construction algorithm. Table V clearly shows that the present approach generates the test set data in a fraction of the time of the explicit calculations of rate coefficients. To illustrate this further, we introduce a "Time Saved Metric", reported as a percentage, which quantifies computation time saved by using the present ML approach. This metric is the ratio of the time required to explicitly generate the data used to train and validate the models plus the time required to complete the kernel building algorithm to the total time to produce the data using the 5DCS method.

Table V additionally shows that the model is able to achieve an accuracy with less than 10% error overall in approximately fifteen minutes, under 5% in about thirty minutes and under, 3% in roughly an hour with only a single kernel for $\mathcal{N} = 3$, $\mathcal{N} = 4$, and $\mathcal{N} = 5$, respectively. While the actual time will fluctuate based on the unique input data and corresponding input features, using the parameters of this method as a guide should not result in any significant escalation in time. Increasing the kernel complexity increases the total computation time, while resulting in improved model accuracy. However, this increase is insignificant compared to the total time of the quantum scattering calculations required for building the models.

IV. CONCLUSION

We have developed an approach that can be used to generate a large amount of rate coefficients for state-to-state transitions in molecular collisions within acceptable accuracy targets. The present adaptive strategy is demonstrated to accurately predict rate coefficients for state-to-state transitions that are not included in the training set. Thus, the models are shown to generalize the temperature dependence of rate coefficients to different state-to-state transitions. Overall, we show that high accuracy of rate coefficients can be achieved with a strategy that employs ML predictions for more inelastic transitions than the number of transitions computed with a quantum scattering method.

The present approach is based on the interplay of quantum scattering calculations and ML using GP regression. To maximize the efficiency of ML models, we have employed GP

[redacted]	$\mathcal{N}=2$	$\mathcal{N}=3$	$\mathcal{N}=4$	$\mathcal{N}=5$	$\mathcal{N}=6$
Transitions for Training	195	309	393	498	600
Transitions for Validation	171	171	179	171	171
Predicted Transitions	1242	1128	1041	939	837
[redacted]					
$k_{\mathcal{N}}^{\text{single}}$, Over 10%	449	220	15	19	12
$k_{\mathcal{N}}^{\text{single}}$, Below 5%	404	606	865	888	790
[redacted]					
$k_{\mathcal{N}}^{\text{opt}}$, Over 10%	447	100	6	6	4
$k_{\mathcal{N}}^{\text{opt}}$, Below 5%	400	615	972	922	829

TABLE IV. Quantitative synopsis of the number of individual state-to-state transitions that are used to validation the models and that are predicted by the models. Also shown is the number of predicted individual state-to-state transitions, for both the single and optimal kernel results, that exceed the maximum accuracy target of 10% and those transitions that had a better accuracy than 5%. For $\mathcal{N} \geq 4$, all single transitions with an error above 10% are accounted for by $(v,20) \rightarrow (v',0)$ transitions.

Time Required	$\mathcal{N}=2$	$\mathcal{N}=3$	$\mathcal{N}=4$	$\mathcal{N}=5$	$\mathcal{N}=6$
5DCS, Input/Validation	52.8	95.4	87.4	115.5	130.6
5DCS, Predicted	285.2	242.6	250.6	222.5	207.4
GP, Single Kernel	.17	.29	.53	1.08	1.73
GP, Optimal Kernel	.52	3.18	4.34	6.90	10.66
Time Saved Metric	86.08%	74.26%	76.05%	68.04%	63.12%

TABLE V. The time, in hours, required to generate the data using this data generation scheme versus the time required to produce the same data using only explicit calculations. The time to produce the results using a single kernel and the optimal kernel are given as well as the time required to produce these results explicitly via quantum dynamics. Also shown is the time required to produce the data that was used for training and validation. In total, all of the data used in this analysis was completed in 383 hours. The "Time Saved Metric", reported as a percent, gives a quantification of how much time is saved by generating data in our method as compared to the explicit calculations.

models with optimized functional forms of the model kernels, using validation error as a kernel selection metric. We observe that increasing kernel complexity improves most models, though not by a significant factor. For most applications, it may be sufficient to employ GP models with simple, but optimally chosen, kernels, i.e. one of the functions in Eqs. (3) – (7) chosen to minimize a validation error for a particular training/validation set of rate coefficients. We have examined the effect of distributions of rate coefficients in the space of quantum numbers on the accuracy of the resulting models and observe that the training data with large gaps between the rotational angular momentum j of the initial state produce optimal results. The large gaps between j are filled by the ML predictions, which results in significant saving of the computation time.

The current work used the rate coefficients from the cross sections computed with the coupled-states approximation since a large number of rate coefficients was required to train, validate, and test the present approach. The 5DCS approximation itself was developed to reduce the computational time compared to the rigorous 6DCC approach. The present approach can be combined directly with the 6DCC calculations. Since each of the 6DCC calculations is much more computationally demanding, the present strategy would result in an enhancement of the time-saved metric as compared to the value estimated for the 5DCS data in Table V, by approximately a factor of 25.

ACKNOWLEDGMENTS

This work was partially supported by NASA grants 80NSSC20K0360 and 80NSSC22K1167 and in part by resources and technical expertise from the Georgia Advanced Computing Resource Center, a partnership between the University of Georgia's Office of the Vice President for Research and Office of the Vice President for Information Technology. The work of TJP and RCF was supported by NSF grant PHY-1806180. The work of NB was supported by NSF grant PHY-2409497.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data was created or analyzed in this study.

AUTHORS DECLARATIONS

The authors have no conflicts to disclose.

AUTHORS CONTRIBUTIONS

Darin E. Mihalik: Methodology (equal); formal analysis (lead); software (lead); writing- original draft (equal) **R.Wang:** Writing- review and editing (equal) **B.H Yang:** Data Curation (equal); writing- review and editing (equal) **N. Balakrishnan**: Writing- review and editing (equal) **P.C Stancil:** Conceptualization (lead); writing- review and editing (equal) **T.J Price:** Data Curation (equal); writing- review and editing (equal) **R.C Forrey:** Data Curation (equal); writing- review and editing (equal) **R. Krems:** Writing- original draft (equal), methodology (equal)

[1] S. Green and P. Thaddeus, "Rotational excitation of CO by collisions with He, H, and H_2 under conditions in interstellar clouds." Astrophys. J. **205**, 766–785.

[2] S. Green, "Rotational excitation in h2–h2 collisions: Close-coupling calculations," The Journal of Chemical Physics **62**, 2271–2277 (1975), https://doi.org/10.1063/1.430752.

[3] N. Balakrishnan, G. Quéméner, R. C. Forrey, R. J. Hinde, and P. C. Stancil, "Full-dimensional quantum dynamics calculations of h2–h2 collisions," The Journal of Chemical Physics **134**, 014301 (2011), https://doi.org/10.1063/1.3511699.

[4] B. Yang, P. Zhang, C. Qu, X. H. Wang, P. C. Stancil, J. M. Bowman, N. Balakrishnan, B. M. McLaughlin, and R. C. Forrey, "Full-dimensional quantum dynamics of sio in collision with h2," The Journal of Physical Chemistry A **122**, 1511–1520 (2018), pMID: 29365271, https://doi.org/10.1021/acs.jpca.7b09762.

[5] B. Yang, P. Zhang, X. Wang, P. C. Stancil, J. M. Bowman, N. Balakrishnan, and R. C. Forrey, "Quantum dynamics of co–h2 in full dimensionality," Nature Communications , 6629 (2015).

[6] B. Yang, X. H. Wang, P. C. Stancil, J. M. Bowman, N. Balakrishnan, and R. C. Forrey, "Full-dimensional quantum dynamics of rovibrationally inelastic scattering between cn and h2," The Journal of Chemical Physics **145**, 224307 (2016), https://doi.org/10.1063/1.4971322.

[7] B. Yang, P. C. Stancil, N. Balakrishnan, and R. C. Forrey, "ROTATIONAL QUENCHING OF CO DUE TO H_2 COLLISIONS," **718**, 1062–1069 (2010).

[8] B. Yang, M. Nagao, W. Satomi, M. Kimura, and P. C. Stancil, "ROTATIONAL QUENCHING OF ROTATIONALLY EXCITED h_2o IN COLLISIONS WITH he," The Astrophysical Journal **765**, 77 (2013).

[9] A. Bohr, S. Paolini, R. C. Forrey, N. Balakrishnan, and P. C. Stancil, "A full-dimensional quantum dynamical study of h2+h2 collisions: Coupled-states versus close-coupling formulation," The Journal of Chemical Physics **140**, 064308 (2014), https://doi.org/10.1063/1.4864357.

[10] T. G. Heil, S. Green, and D. J. Kouri, "The coupled states approximation for scattering of two diatoms," The Journal of Chemical Physics **68**, 2562–2583 (1978), https://aip.scitation.org/doi/pdf/10.1063/1.436115.

[11] M. H. Alexander and A. E. DePristo, "Symmetry considerations in the quantum treatment of collisions between two diatomic molecules," The Journal of Chemical Physics **66**, 2166–2172 (1977), https://doi.org/10.1063/1.434132.

[12] R. C. Forrey, B. Yang, P. Stancil, and N. Balakrishnan, "Mutual vibrational quenching in co+h2 collisions," Chemical Physics **462**, 71–78 (2015), inelastic Processes in Atomic, Molecular and Chemical Physics.

[13] T. J. Price, R. C. Forrey, B. Yang, and P. C. Stancil, "Fine-structure resolved rovibrational transitions for so + h2 collisions," The Journal of Chemical Physics **154**, 034301 (2021), https://doi.org/10.1063/5.0036964.

[14] H. Burton, R. Mysliwiec, R. C. Forrey, B. Yang, P. Stancil, and N. Balakrishnan, "Fine-structure resolved rotational transitions and database for cn+h2 collisions," Molecular Astrophysics **11**, 23–32 (2018).

[15] C. Castro, K. Doan, M. Klemka, R. C. Forrey, B. Yang, P. C. Stancil, and N. Balakrishnan, "Inelastic cross sections and rate coefficients for collisions between co and h2," Molecular Astrophysics **6**, 47–58 (2017).

[16] N.-N. Ma, T.-L. Zhao, W.-X. Wang, and H.-F. Zhang, "Simple deep-learning approach for α-decay half-life studies," Phys. Rev. C **107**, 014310 (2023).

[17] G. P. A. Nobre, D. A. Brown, S. J. Hollick, S. Scoville, and P. Rodríguez, "Novel machine-learning method for spin classification of neutron resonances," Phys. Rev. C **107**, 034612 (2023).

[18] N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo, and G. G. Barnaföldi, "Deep learning predicted elliptic flow of identified particles in heavy-ion collisions at the rhic and lhc energies," Phys. Rev. D **107**, 094001 (2023).

[19] P. Ilten, T. Menzo, A. Youssef, and J. Zupan, "Modeling hadronization using machine learning," SciPost Phys. **14**, 027 (2023).

[20] A. Butter, T. Plehn, S. Schumann, S. Badger, S. Caron, K. Cranmer, F. A. D. Bello, E. Dreyer, S. Forte, S. Ganguly, D. Gonçalves, E. Gross, T. Heimel, G. Heinrich, L. Heinrich, A. Held, S. Höche, J. N. Howard, P. Ilten, J. Isaacson, T. Janßen, S. Jones, M. Kado, M. Kagan, G. Kasieczka, F. Kling, S. Kraml, C. Krause, F. Krauss, K. Kröninger, R. K. Barman, M. Luchmann, V. Magerya, D. Maitre, B. Malaescu, F. Maltoni, T. Martini, O. Mattelaer, B. Nachman, S. Pitz, J. Rojo, M. Schwartz, D. Shih, F. Siegert, R. Stegeman, B. Stienen, J. Thaler, R. Verheyen, D. Whiteson, R. Winterhalder, and J. Zupan, "Machine learning and LHC event generation," SciPost Phys. **14**, 079 (2023).

[21] P. Eller, A. T. Fienberg, J. Weldert, G. Wendel, S. Böser, and D. Cowen, "A flexible event reconstruction based on machine learning and likelihood principles," Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **1048**, 168011 (2023).

[22] G. Fang, S. Ba, Y. Gu, Z. Lin, Y. Hou, C. Qin, C. Zhou, J. Xu, Y. Dai, J. Song, and X. Kong, "Automatic classification of galaxy morphology: A rotationally-invariant supervised machine-learning method based on the unsupervised machine-learning data set," The Astronomical Journal **165**, 35 (2023).

[23] A. Jasinski, J. Montaner, R. C. Forrey, B. H. Yang, P. C. Stancil, N. Balakrishnan, J. Dai, R. A. Vargas-Hernández, and R. V. Krems, "Machine learning corrected quantum dynamics calculations," Phys. Rev. Research **2**, 032051 (2020).

[24] M. Lochner, L. Rudnick, I. Heywood, K. Knowles, and S. S. Shabala, "A unique, ring-like radio source with quadrilateral structure detected with machine learning," Monthly Notices of the Royal Astronomical Society **520**, 1439–1446 (2023), https://academic.oup.com/mnras/article-pdf/520/1/1439/49116741/stad074.pdf.

[25] P. Lemos, M. Cranmer, M. Abidi, C. Hahn, M. Eickenberg, E. Massara, D. Yallup, and S. Ho, "Robust simulation-based inference in cosmology with bayesian neural networks," Machine Learning: Science and Technology **4**, 01LT01 (2023).

[26] D. de Andres, G. Yepes, F. Sembolini, G. Martínez-Muñoz, W. Cui, F. Robledo, C.-H. Chuang, and E. Rasia, "Machine learning methods to estimate observational properties of galaxy clusters in large volume cosmological N-body simulations," Monthly Notices of the Royal Astronomical Society **518**, 111–129 (2022), https://academic.oup.com/mnras/article-pdf/518/1/111/47014393/stac3009.pdf.

[27] I. Gómez-Vargas, J. Briones Andrade, and J. A. Vázquez, "Neural networks optimized by genetic algorithms in cosmology," Phys. Rev. D **107**, 043509 (2023).

[28] H. Moustafa, P. M. Lyngby, J. J. Mortensen, K. S. Thygesen, and K. W. Jacobsen, "Hundreds of new, stable, one-dimensional materials from a generative machine learning model," Phys. Rev. Mater. **7**, 014007 (2023).

[29] S. S. Rahaman, S. Haldar, and M. Kumar, "Machine learning approach to study quantum phase transitions of a frustrated one dimensional spin-1/2 system," Journal of Physics: Condensed Matter **35**, 115603 (2023).

[30] S. Ciarella, M. Chiappini, E. Boattini, M. Dijkstra, and L. M. C. Janssen, "Dynamics of supercooled liquids from static averaged quantities using machine learning," Machine Learning: Science and Technology **4**, 025010 (2023).

[31] D. M. Anstine and O. Isayev, "Machine learning interatomic potentials and long-range physics," *The Journal of Physical Chemistry A*, The Journal of Physical Chemistry A **127**, 2417–2431 (2023).

[32]C. Miles, R. Samajdar, S. Ebadi, T. T. Wang, H. Pichler, S. Sachdev, M. D. Lukin, M. Greiner, K. Q. Weinberger, and E.-A. Kim, "Machine learning discovery of new phases in programmable quantum simulator snapshots," Phys. Rev. Res. **5**, 013026 (2023).

[33]Y.-H. Zhang and M. Di Ventra, "Transformer quantum state: A multipurpose model for quantum many-body problems," Phys. Rev. B **107**, 075147 (2023).

[34]D. Duvenaud, H. Nickisch, and C. E. Rasmussen, "Additive gaussian processes," (2011).

[35]D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," (2013).

[36]J. Dai and R. V. Krems, "Quantum gaussian process model of potential energy surface for a polyatomic molecule," The Journal of Chemical Physics **156**, 184802 (2022), https://doi.org/10.1063/5.0088821.

[37]J. Dai and R. V. Krems, "Interpolation and extrapolation of global potential energy surfaces for polyatomic systems by gaussian processes with composite kernels," Journal of Chemical Theory and Computation **16**, 1386–1395 (2020).

[38]J. Gao, J. Wang, Z. Xu, C. Wang, and S. Yan, "Multiaxial fatigue prediction and uncertainty quantification based on back propagation neural network and gaussian process regression," International Journal of Fatigue **168**, 107361 (2023).

[39]H. Sugisawa, T. Ida, and R. V. Krems, "Gaussian process model of 51-dimensional potential energy surface for protonated imidazole dimer," The Journal of Chemical Physics **153**, 114101 (2020), https://doi.org/10.1063/5.0023492.

[40]D. Poloni, D. Oboe, C. Sbarufatti, and M. Giglio, "Towards a stochastic inverse finite element method: A gaussian process strain extrapolation," Mechanical Systems and Signal Processing **189**, 110056 (2023).

[41]R. A. Vargas-Hernández, J. Sous, M. Berciu, and R. V. Krems, "Extrapolating quantum observables with machine learning: Inferring multiple phase transitions from properties of a single phase," Phys. Rev. Lett. **121**, 255702 (2018).

[42]A. G. Wilson and R. P. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13 (JMLR.org, 2013) p. III–1067–III–1075.

[43]C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, 2005).

[44]A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman, " GPflow: A Gaussian process library using TensorFlow," Journal of Machine Learning Research **18**, 1–6 (2017).

[45]M. van der Wilk, V. Dutordoir, S. John, A. Artemev, V. Adam, and J. Hensman, "A framework for interdomain and multioutput Gaussian processes," arXiv:2003.01115 (2020).

[46]G. Schwarz, "Estimating the Dimension of a Model," The Annals of Statistics **6**, 461 – 464 (1978).