Interpolation of intermolecular potentials using Gaussian processes.

Elena Uteva [a], Richard S. Graham [b], Richard D. Wilkinson [c] and Richard J. Wheatley[a].[1]

[a] *School of Chemistry, University of Nottingham, Nottingham NG7 2RD, UK.*

[b] *School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, UK.*

[c] *School of Mathematics and Statistics, University of Sheffield, Sheffield, S10 2TN, UK.*

(Dated: 23 May 2017)

A procedure is proposed to produce intermolecular potential energy surfaces from limited data. The procedure involves generation of geometrical configurations using a Latin hypercube design, with a maximin criterion, based on inverse internuclear distances. Gaussian processes are used to interpolate the data, using over-specified inverse molecular distances as covariates, greatly improving the interpolation. Symmetric covariance functions are specified so that the interpolation surface obeys all relevant symmetries, reducing prediction errors. The interpolation scheme can be applied to many important molecular interactions with trivial modifications. Results are presented for three systems involving CO_2, a system with a deep energy minimum (HF−HF) and a system with 48 symmetries (CH_4−N_2). In each case the procedure accurately predicts an independent test set. Training this method with high-precision *ab initio* evaluations of the CO_2−CO interaction enables a parameter-free, first-principles prediction of the CO_2−CO cross virial coefficient that agree very well with experiments.

PACS numbers: XXX, XXX, XXX

I. INTRODUCTION

This article concerns a central problem in molecular physics: quantitatively predicting macroscopic material properties from first-principles molecular physics. There are numerous practical applications to this problem, including the molecular design of materials, manufacture and industrial processing. Such applications would be better understood and addressed by exploiting predictions, from molecular first principles, of properties and processes such as solubility, phase separation, osmosis, gas diffusion, crystallisation and nucleation.

Intermolecular potential energies can often be calculated accurately enough to be useful. Thus the problem above can be solved, in principle, by computing molecular interactions from *ab initio* quantum mechanics and macroscopic properties from statistical mechanics. However, in practice, the computational cost of directly combining these two techniques is hopelessly expensive in all but exceptional cases. The cost of evaluating the energy at a single point is significant (often minutes or hours of time). Thus, it is necessary to fit or interpolate calculated energy data at a limited number of points to produce a potential energy surface.

In this article a machine-learning approach is applied to the problem. Machine-learning algorithms have recently produced very rapid progress in computer mastery of complex games, such as Go[1], and they have enormous potential to surmount recalcitrant problems in the physical sciences. The approach herein produces a flexible and widely applicable algorithm to interpolate calculations of intermolecular potential energies, leading to a fast method to compute the potential energy surface. The algorithm's significant advantages include high accuracy, the need for only comparatively small input data sets, and the generality to capture new molecular systems with little or no modification, leading to broad applicability and significant savings in researcher time.

The efficacy of the method is shown by comparison with current best practice in representing potential energy surfaces. Examples of careful, elaborate fits of calculated data include the potential energy surface of CO_2−Ne[2], where a root mean square error (RMSE) $\sim$0.15 μE_h was quoted (1 $E_h \approx$ 27.211 eV$\approx$ 2625.5 kJ mol^{-1}); a RMSE of about 0.6 μE_h in the well region (energy $E < 0$) of CO_2−H_2[3]; and a maximum error of about 2% of the well depth in the well region of CH_4−N_2[4]. Fits with much larger errors are commonplace in the literature, even when RMSE scores are based only on the training data, rather than indepen-

dent data, a procedure which is prone to over-estimating predictive accuracy. Interpolations of intermolecular potential data are less common. Cubic splines are the most popular interpolation method, for example in work on CO_2-Ar[5]. In contrast, Gaussian process (GP) interpolation, of which cubic splines are a special case[6,7], has been relatively little used[8,9], despite its promise in other applications. Applications include solid-state potentials[10,11], conformational energies[12], and the difference between calculated intermolecular potentials of water[13]. The accuracy and speed of GPs compared to other machine learning techniques was demonstrated for the electronic polarisation of water clusters[14]. The same group later applied GPs to the atomic multipole moment surface of ethanol[15], the electrostatic interaction energy between Na^+ and water molecules[16] and the electrostatic energy of cholesterol[17]. It is demonstrated here that with a small set of training points and a carefully chosen coordinate system, a general symmetric Gaussian process interpolation scheme can achieve high predictive accuracy and that the resulting PES can accurately predict experimental data.

II. GAUSSIAN PROCESS MODELLING

The approach involves two sets of data. A set of training data (between 20-1000 points) is used to train the model, and a larger set of grid data is used to test the model's predictive performance. Both datasets are described below. No knowledge of the test data is used during training.

Data sets of the intermolecular interaction energy of the bimolecular complexes CO_2-Ne, CO_2-H_2, CO_2-CO, $HF-HF$ and CH_4-N_2 are calculated as a function of their configurational geometry. These complexes are chosen to cover a range of intermolecular bond strengths and symmetries, and to be small enough that an extensive set of data can be produced for testing the interpolations. For the same reason, all molecules are approximated as linear rigid rotors in their vibrational ground state, with fixed bond lengths, although the interpolation method can be extended straightforwardly to non-rigid molecules. Energy calculations are carried out in Molpro[18] using second-order Møller-Plesset perturbation theory (MP2) and augmented correlation-consistent triple-zeta (aug-cc-pVTZ) basis sets. Basis set superposition errors are corrected using the full counterpoise correction procedure.

Jacobi coordinates are used to describe the multi-dimensional potential energy hypersurfaces (see Table I). In all cases r is the distance between the molecular centres. For

CO_2-Ne, θ is the angle between r and the CO_2 axis. For CO_2-H_2, θ_1 is the angle between r and the CO_2 axis, θ_2 is the angle between r and the H_2 axis, and ϕ is the torsional angle of the H_2 axis. Analogous coordinates are used for HF−HF and for CO_2-CO, where $\theta_2 = 0$ corresponds to the O from the CO molecule being closest to the CO_2. For CH_4-N_2, the N_2 molecule is placed at a position relative to the C of CH_4 at position (r, θ, ϕ) in polar coordinates, and the N-N axis is rotated to orientation (α, β), also in polar coordinates. The C-O (for CO_2), C-O (for CO), H-H, H-F, C-H and N-N bond lengths are taken to be 1.1632 Å, 1.128 Å, 0.77 Å, 0.92 Å, 1.09 Å and 1.098 Å, respectively. An energy cutoff of $E_{\text{cut}} = 0.005\ E_{\text{h}}$ is imposed (0.02 E_{h} for HF-HF due to its much larger well depth), and configurations with intermolecular potentials above this cutoff are excluded from the data sets. Configurations are also excluded if any interatomic distance is below 1.5 Å or if all interatomic distances are above 8.5 Å. Separations below this would also be excluded by the energy cutoff, but this criterion saves time that would be spent in calculating unhelpfully large energies, and beyond 8.5 Å it is more efficient to use an asymptotic expansion of the energy, as discussed later. Details of the test data used for model assessment are given in Table I.

Gaussian processes (GPs) are used extensively in machine learning and statistics as regression models. The book of Rasmussen and Williams[7] contains a detailed introduction to GPs, including several different but equivalent ways of understanding GPs. Like neural networks (NNs), GPs are non-parametric models which have proved successful in creating theory-free models of complex datasets. The advantage of GPs over NNs is that they are mathematically tractable and interpretable, and allow prior information to be built in to the model (such as symmetry, differentiability, and conditioning on derivative information). The prior specification of a GP consists of a mean function (often taken as zero) and a covariance function $k(\mathbf{x}, \mathbf{x}')$, expressing the covariance between $f(\mathbf{x})$ and $f(\mathbf{x}')$, where f is the function being interpolated. Training data, consisting of observations of the value of f at various locations, update the mean and covariance functions to give a posterior model which predicts the function at any location.

Properties of the GP model are inherited from the covariance function, for example, differentiability, continuity and stationarity. The intermolecular energy is a non-stationary function of distance, as it varies rapidly at small interatomic separations, but more gently at larger separation. Although it is possible to specify a non-stationary covariance function,

TABLE I. Coordinates for the test (grid or LHC) data for each system.

System	Test Grid or Latin Hypercube			Test
	Coordinate	Range	Spacing	points
CO_2-Ne	r	1.5-10 Å	0.116 Å	1122
	$\cos\theta$	0-1	0.05	
CO_2-H_2	r	1.5-10 Å	0.5 Å	12844
	$\cos\theta_1$	0-1	0.111	
	$\cos\theta_2$	0-1	0.111	
	ϕ	0-180°	20°	
CO_2-CO	r	1.5-10 Å	Latin	60000
	$\cos\theta_1$	0-1	hypercube	
	$\cos\theta_2$	-1-1	sampling	
	ϕ	0-180°		
$HF-HF$	Latin hypercube			2158
CH_4-N_2	Latin hypercube			1182

in practice it can be challenging to find a flexible form with the correct non-stationary behaviour. It is simpler to transform either the inputs or outputs to achieve approximate stationarity, which is addressed here by using the inverse interatomic distances as covariates in the GP. Thus the GP coordinates are $\mathbf{x} = (1/r_1, ..., 1/r_{N_D})$ where r_i is the interatomic distance, running over all pairs of nuclei on different molecules. This results in an over-specified system, for example with $N_D = 6$ dimensions for CO_2-H_2. It is shown later that this change in variables leads to a dramatic improvement in performance.

The training data should ideally cover evenly a single symmetry-distinct sub-region of $\mathbf{x}$ space, and respect the geometric constraint. The general strategy is to generate many candidate data sets (coordinates only, not energies), exclude points outside the symmetric and geometric constraints, and select the candidate data set with the best distribution of points. The placement of candidate data sets is based on Latin hypercube (LHC) sampling, as explained in appendix B. Specifically, for CO_2-Ne, CO_2-H_2 and CO_2-CO, candidate data sets are generated from LHC sampling of $1/r$ and the angular LHC coordinates in

Table I. For HF−HF, three LHCs are generated and combined into one dataset: one uses the F-F distance as the radial coordinate r, and keeps only those data points within the LHC for which the F-F distance is the shortest of the four internuclear distances; the other two LHCs are generated in the same way but with F-F replaced by H-H and H-F in turn. The LHC for CH_4-N_2 is generated based on an H-N distance as the radial coordinate, and uses only the data points for which the same H-N distance is the shortest of the ten internuclear distances. For all cases, after generating the LHCs, deleting data points based on the symmetric and geometric constraints, and combining the sets of points into one (for HF-HF), the minimum separation of the remaining points is calculated in $\mathbf{x}$ space. The candidate data set with the largest minimum separation is used as the training set. This 'maximin' approach aims to cover evenly the relevant region of $\mathbf{x}$ space. See appendix B for further details of this algorithm.

The Gaussian process has a zero mean function and a squared-exponential covariance function

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \prod_{i=1}^{N_D} \exp\left[-\frac{(x_i - x_i')^2}{2l_i^2}\right] \tag{1}$$

where σ_f^2 is the signal variance and l_i is the correlation length for each dimension. This results in a stationary and infinitely differentiable model, called the 'non-symmetric model'. Here, symmetric refers to the permutations the interatomic distances under which the PES should be invariant, because of molecular symmetries (see appendix C for further details). In the non-symmetric model neither the kernel (covariance function) nor the training data have these symmetries. Introduced later is a model in which the kernel has the appropriate symmetries.

The potential energy surfaces obey various symmetries in $\mathbf{x}$ space. For example, for CO_2-Ne, the energy is invariant under the interchange of the two coordinates corresponding to distances between Ne and each of the O atoms. Let G represent the permutation group containing permutations of elements of $\mathbf{x}$ under which the energy surface is unchanged. If it is assumed that $l_i = l_j$ when coordinates x_i and x_j swap for some permutation in G, then a covariance function of the form

$$k_{\text{sym}}(\mathbf{x}, \mathbf{x}') = \sum_{g \in G} \kappa(g\mathbf{x}, \mathbf{x}'). \tag{2}$$

results in a GP which shares the symmetries of the energy surface (see appendix C). The

'symmetric model' based on this covariance function gives predictions that respect the relevant symmetries, and usually significantly improves the performance, even within the symmetry-distinct region covered by the test data, as shown below.

Results are obtained using the GPy package[19], modified to include symmetric covariance functions. Zero-mean Gaussian observation error[7] is assumed on the function outputs (referred to as a *nugget* in geostatistics), with standard deviation σ_n. Thus the model's hyperparameters are σ_f, σ_n and $\{l_i\}$. These hyperparameters are estimated by optimising the log-likelihood over ≈ 30 random restarts, which typically is sufficient to find the optimal values multiple times.

The choice of inverse internuclear distances to transform to stationarity is important. To illustrate this a 'basic model' GP is created, which uses internuclear distances r as coordinates rather than $1/r$, but is otherwise identical to the non-symmetric GP above, having the same test and training data and a covariance function of the same form as equation (1).

III. RESULTS

Predictive performance is measured using the root mean square error (RMSE) of the GP predictions of the test data. Since the test data extensively cover the potential energy surface, the RMSE is a reasonable guide to the expected 'accuracy' of the interpolation. The methodology used here represents a demanding test of accuracy, since the GP has no advance knowledge of the test data, only the far more limited training data, and equal weighting is used for all test data, including positive interaction energies up to the potential energy cutoff.

The results for CO_2-Ne are shown in Figure 1. Models based on inverse intermolecular distances dramatically outperform the basic model, being typically 2-3 orders of magnitude more accurate for a given LHC size. Furthermore, even though CO_2-Ne has only one symmetry, the symmetric model is typically twice as accurate as the non-symmetric model. Figure 2 shows similar results for CO_2-H_2. Here, the inverse distance models again strongly outperform the basic model, achieving RMSEs $< 10^{-6}$ E_h for a reasonable number of training points. The symmetric kernel typically gives a factor of 2-10 improvement, with the greater improvement compared to CO_2-Ne probably resulting from the greater number

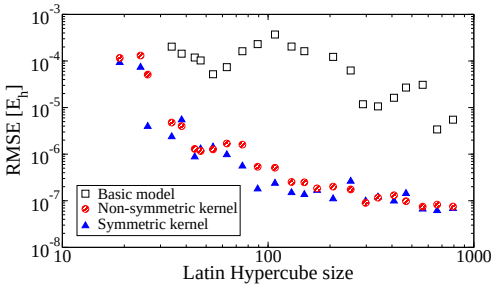

FIG. 1. RMSE against LHC size for CO_2-Ne. The lowest energy in the grid data is -2.90×10^{-4} E_h.

of symmetries.

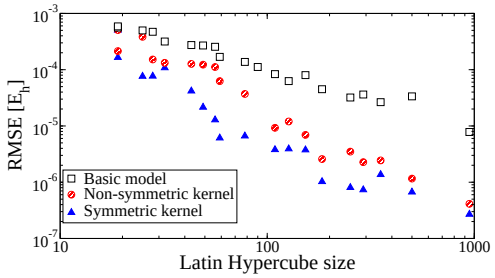

FIG. 2. RMSE against LHC size for CO_2-H_2. The lowest energy in the grid data is -8.25×10^{-4} E_h.

For HF−HF, the minimum energy in the test data is -6.17×10^{-3} E_h, which is about an order of magnitude larger than for the other interactions. Probably as a consequence of this, it is necessary to include training points up to at least 10^{-2} E_h, otherwise the prediction of points on the repulsive wall is poor. Using a cutoff of 2×10^{-2} E_h gives an RMSE of 1.6×10^{-4} E_h for a symmetric GP with 59 training points, and the RMSE generally decreases with increasing numbers of training points, to 1.8×10^{-5} E_h for 327 training points. The RMSE in the negative-energy region is about 5×10^{-6} E_h for the latter GP; one or two high-energy points dominate the overall RMSE. The inclusion of symmetry in the GP has little effect on the RMSE for this interaction.

For CH_4-N_2, all 48 symmetry elements are included in the GP. Consideration of symmetry is important for this interaction, even though all the training and test data are confined within a single symmetry-distinct region of space. The minimum energy in the test data is -6.98×10^{-4} E_h. With a training set of 106 points, the RMSE is 51×10^{-6} E_h for the non-symmetric GP and 6.8×10^{-6} E_h for the symmetric GP. Using 326 training points reduces

these to 17×10^{-6} E_h and 1.3×10^{-6} E_h respectively.

For CO_2-CO, with the symmetric GP, the RMSE is 6.7×10^{-5} E_h for 132 training points, reducing to 1.1×10^{-5} E_h for 345 training points. The RMSE resulting from the non-symmetric GP is generally a factor of ~ 2 larger. Generating CO_2-CO training data by replacing LHC sampling of $1/r$ with $1/r^2$ improves performance, leading to RMSEs of 2.4×10^{-5} and 3.0×10^{-6} for 135 and 366 training points respectively, with the RMSEs computed using the same test set as the above results.

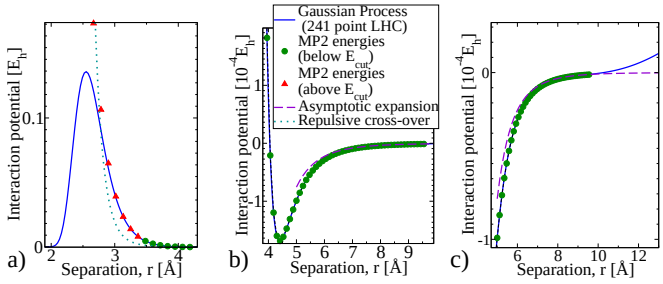

FIG. 3. CO_2-Ne Molpro calculations and the GP model, at $\cos\theta = 1$ (linear geometry), in the repulsive (a), attractive (b) and long-range (c) regions. The long-range asymptotic expansion is $E = -(0.570 + 0.182\cos^2\theta)r^{-6} - (1.704 + 7.266\cos^2\theta + 1.785\cos^4\theta)r^{-8}$.

The performance of the GP outside the training region ($E > 0.005E_h$ and $r > 8.5$Å) is shown for CO_2-Ne in Figure 3; results for other geometries and interactions are qualitatively similar. The extrapolation errors for short-range geometries satisfying the geometric constraint, but with $E > E_{\rm cut}$, are mostly a few percent or less. Even though the maximum energy in the training set is limited to 0.005 $E_{\rm h}$, the predicted repulsive wall continues up to about 0.1 $E_{\rm h}$. However, for small values of r the GP returns to its mean value of zero. This unphysical behaviour can be corrected by crossing over to a strongly repulsive function outside the geometric constraint. One example of the many possible choices is plotted in Figure 3(a), namely

$$E = E_{\rm max}\frac{1}{N}\sum_{i=1}^{N}(x_i/x_{\rm max})^{12}, \tag{3}$$

where $E_{\rm max}$ is an estimate of the typical energy at the small-r edge of the geometric constraint[20] and $x_{\rm max}$ is the maximum inverse distance allowed by the geometric constraint (0.67 Å^{-1} in this case). For large separations the GP tends to a small, but non-zero constant. This can be corrected for points beyond the geometric constraint, by crossing over to

the long-range asymptotic expansion obtained from a truncated multipole expansion of the interaction energy (to second order) from intermolecular perturbation theory. Figure 3(c) shows that smooth interpolation between the GP and this function will be straightforward.

IV. THE CO_2−CO POTENTIAL ENERGY SURFACE AND SECOND VIRIAL COEFFICIENT

Using the LHC training set for CO_2−CO, based on sampling of $1/r^2$ and containing 135 training points, higher quality evaluations of the potential energy were produced for each configuration. This involved complete basis set (CBS) extrapolation of the counterpoise-corrected CCSD(T) interaction energy from the aVQZ and aVTZ basis sets. Training a GP to this set gave a PES designed to be valid within the geometric constraint for energies $< 0.005 E_h$. For large separations outside the geometric constraint, an asymptotic expression consisting of atomic charges, dipoles, quadrupoles, static polarizabilities and C_6 dispersion coefficients is used instead of the GP. For separations smaller than the geometric constraint the PES uses equation (3) with $x_{\max} = 0.5/\text{Å}$ and $E_{\max} = 0.456\ E_h$, which is the largest energy in the LHC training data before applying the energy cut off. The code for this potential energy surface is available in the Supplementary Material.

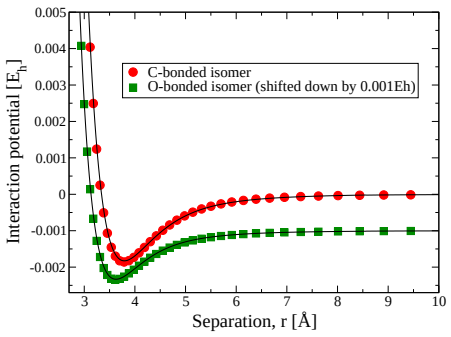

FIG. 4. Comparison of calculated interaction energies (symbols) with the GP-based PES (lines) for CO_2−CO in the T-shaped configuration ($\theta_1 = \pi/2$) for the C-bonded ($\theta_2 = \pi$) and O-bonded ($\theta_2 = 0$) isomers:. The C-bonded isomer includes the minimum energy.

The minimum energy for this PES, computed by numerical minimisation, is $-1.827 \times 10^{-3} E_h$, which occurs at $r = 3.791$ Å, $\theta_1 = \pi/2$ and $\theta_2 = \pi$, corresponding to a T-shaped orientation with the carbon of the CO closest to the CO_2 molecule. This agrees with previous

experimental[21] and theoretical[22] work in which this C-bonded isomer was confirmed to be the most stable. The C-C distance of 3.23 Å from the PES is 0.01 Å larger than the minimum distance taken from the best literature calculation[22]; both are lower than the experimental distance of 3.277 Å[21], mainly as a result of non-rigidity of the intermolecular bond. A direct calculation of the interaction energy for this geometry gives -1.859×10^{-3} E_h, which is within 0.01×10^{-3} E_h of the best calculated value[22]. The difference of $\sim 0.03 \times 10^{-3} E_h$ between the PES and calculation is similar to the RMSE of $0.024 \times 10^{-3} E_h$ reported in section III. A second, O-bonded, isomer has been observed spectroscopically[23], and found to be also T-shaped with the CO reversed relative to the C-bonded isomer. The equilibrium separation predicted from the PES is 3.06 Å between the C of CO_2 and the O of CO, which is about 0.04 Å less than the experimental value. The difference between the two is in the expected direction, and reasonably consistent with the results for the C-bonded isomer. Apparently there have been no other high-quality calculations on the O-bonded isomer; the current PES gives an interaction energy of -1.332×10^{-3} E_h, which, again, agrees well with a direct calculation at the same geometry; the calculated interaction energy is -1.345×10^{-3} E_h. Figure 4 shows how the GP-PES predicts accurately the entire well region for both the C and O-bonded isomers.

The GP's ability to predict these structures and energies is impressive when considering that the training data are sparsely positioned and have essentially no *a priori* information about the position of the minimum. Indeed the lowest energy in the training set is -1.15×10^{-3} E_h, which is positioned at $r = 3.75$ Å, $\theta_1 = 0.36\pi$, $\theta_2 = 0.76\pi$ and $\phi = 0.71\pi$. This configuration is broadly in the vicinity of the minimum but has an energy that is only 62% of the minimum energy. Nevertheless the GP captures the minimum energy with an error of less than 2%. This illustrates the considerable ability of the GP to reproduce quantitative details of the PES even when interpolating features that are a significant distance from the training data.

The CO_2−CO PES is then used to compute the cross virial coefficient, taking the classical contribution plus the (small) first-order quantum corrections; the calculation procedure has been described previously[28]. A comparison with experimental data is shown in figure 5. The PES produces accurate, first-principles, parameter-free prediction of this quantity, over the range of temperatures studied experimentally. Also shown are the PES predictions with the long-range asymptotic correction set to zero. This shows that the virial prediction is

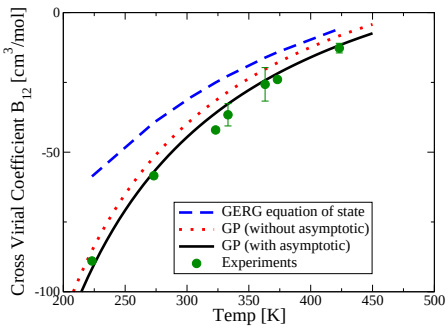

FIG. 5. The CO_2-CO cross virial coefficient: comparison of the GP calculations with experimental data[24–26] and the GERG equation of state[27].

mostly determined by the PES region controlled by the GP, with the long-range asymptotic calculation making a small but significant further contribution. Also in figure 5 are the results from the GERG model[27], a widely used empirical equation of state. Despite the extensive fitting involved in the development of the GERG equation of state, the predictions of the present first-principles approach gives significantly better agreement with experiment.

V. FUTURE WORK

There are numerous extensions that follow from this approach. Application to many other chemical systems is straightforward. Furthermore, the model's performance for small training set sizes could be optimised by sequentially adding training points through active learning methods[29]. This could be achieved either with or without *a priori* knowledge of the test data, depending on the potential energy data to be modelled. Another promising application is the interpolation of non-additive potentials, which are known to be difficult to fit[30]. Such data are usually high-dimensional, vary strongly and rather unpredictably with geometry, and can have many symmetries. Finally, existing high-precision calculations could be used as training and test data for interpolation by the algorithm. Here a sparse GP[31] could select a subset of the preexisting data to train the model, leading to numerically cheap, yet highly accurate, potential energy surfaces.

VI. CONCLUSIONS

The procedure described here produces intermolecular potential energy surfaces efficiently from relatively few input data points. The algorithm is straightforward to generalise to new molecular pairs. It uses a symmetric Gaussian process, with the inverse interatomic distances as input variables. The GP is trained using data on a Latin hypercube design, with a maximin criterion for the inverse internuclear distances.

The general utility and robustness of the approach have been demonstrated by testing against three systems involving CO_2, a system with a deep energy minimum (HF−HF) and a system with 48 symmetries (CH_4−N_2). In all cases the approach accurately predicts an extensive test data set, with no *a priori* knowledge, and gives RMSE values that are similar to, or better than, the best fits in the literature, which were generally based on thousands of training points. Furthermore, the interpolation method can be readily and directly applied to any pairwise interaction, at least for simple molecules, with no bespoke work beyond identifying the relevant symmetries. The approach contains two key features: the use of inverse interatomic distances as the GP input variables, and a strategy for positioning training data on a Latin hypercube design with a maximin criterion on the inverse intermolecular distances.

As the method requires only a relatively small number of training points it can be used with more precise and computationally demanding potential energy calculations. An exemplar is CO_2−CO, for which a four-dimensional PES was produced from 135 data points. This CO_2−CO PES was used to compute the cross virial coefficient, leading to parameter-free predictions of this quantity that agree very well with experimental data.

SUPPLEMENTARY MATERIAL

See the supplementary material for fortran code for the CO_2−CO PES used in section IV.

ACKNOWLEDGEMENTS

The authors are grateful to the EPSRC for the award of a studentship to EU, and to the University of Nottingham for the use of the 'Minerva' high-performance computing facility. We thank Prof. Roland Span for making available to us the literature experimental data

compiled during the IMPACTS project.

Appendix A: Latin hypercube sampling

Latin hypercube (LHC) sampling is a method of choosing sampling points in a multidimensional space. LHC sampling aims to spread the sample points more evenly across all possible values than random sampling. In 2D a latin square is defined as a set of samples with exactly one sample in each row and column. For a given number of rows and columns there are many permutations that satisfy this. A LHC generalises this to an arbitrary number of dimensions by requiring that each hyperplane normal to one of the co-ordinates contains exactly one sample.

We generate an M dimensional unit LHC of N sample points as follows.

- Place all N points on the diagonal of the hypercube to create an initial LHC. We do this by placing the nth point so that each co-ordinate value lies in the range $(n-1)/N...n/N$, with the value within the range being chosen randonly, uniformly and independently for each co-ordinate.

- Choose two points at random and swap one of their co-ordinate values, chosen at random. This permutes the sampling configuration, while ensuring we still have a LHC, as defined above.

- Repeat the co-ordinate swaps until the desired number of swaps has been completed.

Appendix B: Latin hypercube generation

We wish to generate a dataset of model evaluations, $\{\mathbf{x}_i, f(\mathbf{x}_i)\}_{i=1}^{N}$, that can be used to train the Gaussian process, where the $\mathbf{x}_i$ represent N distinct molecular geometries. Each element of $\mathbf{x}_i$ is the inverse distance between two atoms, one from each of the molecules under consideration. The design only needs to contain points in a symmetry-distinct subspace. For example, in CO_2-Ne the O nuclei are denoted O1 and O2, and the symmetry-distinct subspace is defined such that Ne is always nearer to O1 than to O2. Space filling designs are held to be good choices for Gaussian process models, and so we will use a maxi-min criterion to evaluate candidate designs. In other words, we seek designs which maximise

the minimum distance between any two design points. Latin hypercube (LHC) designs are used as candidate designs, as they naturally fill space to some extent, and we then choose a preferred design from a large number of candidates. We define the effective distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the design to be

$$|\mathbf{x}|_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{x}_i - \mathbf{x}_j) \tag{B1}$$

and we generate a training design using the following algorithm:

- Generate a LHC in $1/r$ and rigid-body rotation angles. (For non-rigid molecules, intramolecular coordinates would also be used.)
- Convert the LHC data to atomic positions and compute all interatomic distances for pairs of atoms on separate molecules.
- Reject the geometries that don't obey the geometric constraint or lie outside the symmetry-distinct region of coordinate space.
- Reject the entire LHC if it does not contain at least the target number of geometries (usually the mean number of remaining points after the geometric constraint is applied).
- Find the minimum $|\mathbf{x}|_{ij}^2$ within the current LHC.
- Repeat for as many new LHCs as desired and return the LHC with the largest minimum $|\mathbf{x}|_{ij}^2$.

Appendix C: Symmetric covariance function

The motivating problem is modelling the H_2 - CO_2 system, which we parameterise by 6 distances:

- $r_1 = \mathrm{H}^1 \rightarrow \mathrm{C}$
- $r_2 = \mathrm{H}^2 \rightarrow \mathrm{C}$
- $r_3 = \mathrm{H}^1 \rightarrow \mathrm{O}^1$
- $r_4 = \mathrm{H}^2 \rightarrow \mathrm{O}^1$

- $r_5 = \mathrm{H}^1 \to \mathrm{O}^2$
- $r_6 = \mathrm{H}^2 \to \mathrm{O}^2$

The potential function f between the two molecules obeys the following symmetry relations

$$f(123456) = f(214365) = f(125634) = f(216543)$$

where $f(123456)$ denotes $f(r_1, r_2, r_3, r_4, r_5, r_6)$.

In other words, the function

$$f(x) = f(\sigma x) \forall \sigma \in K_4$$

where K_4 is the permutation group consisting of the permutations

$$\sigma_1 = (12)(34)(56), \quad \sigma_2 = (35)(46) \quad \sigma_3 = (12)(36)(45)$$

where we are using cyclic notation for the permutations. Note that along with the identity e, these four permutations form an abelian group that is isomorphic to the Klein-4 group K_4 ($\equiv Z_2 \times Z_2$), i.e., $\sigma_i^2 = e$ and $\sigma_1\sigma_2 = \sigma_3$ etc.

1. A single symmetry

To illustrate the procedure, suppose we want to model f where f is invariant under the single permutation σ, where $\sigma^2 = e$. If we assume

$$f(x) = g(x) + g(\sigma x)$$

for some arbitrary function g, then f has the required symmetry. If we model $g(\cdot) \sim GP(0, k(\cdot, \cdot))$, then the covariance function for f is

$$\begin{aligned} k_f &= \mathbb{C}\mathrm{ov}(f(x), f(x')) \\ &= k(x, x') + k(\sigma x, x') + k(x, \sigma x') + k(\sigma x, \sigma x') \end{aligned}$$

If k is an isotropic kernel (we only actually require isotropy for each pair of vertices that swap in σ), then $k(x, x') = k(\sigma x, \sigma x')$ and $k(x, \sigma x') = k(\sigma x, x')$ as swaps only occur in pairs ($\sigma^2 = e$). So we can use

$$k_f(x, x') = k(x, x') + k(\sigma x, x')$$

saving half the computation.

2. Invariance under permutations in K_4

Now consider functions that are invariant to permutations in K_4. If we write

$$f(x) = g(x) + g(\sigma_1 x) + g(\sigma_2 x) + g(\sigma_3 x)$$

then if $g(\cdot) \sim GP(0, k(\cdot, \cdot))$

$$\begin{aligned} k_f(x, x') =& k(x, x') + k(\sigma_1 x, x') + k(\sigma_2 x, x') + k(\sigma_3 x, x') \\ &+ k(x, \sigma_1 x') + k(\sigma_1 x, \sigma_1 x') + \ldots k(\sigma_3 x, \sigma_3 x') \end{aligned} \tag{C1}$$

If k is isotropic, then $k(x, \sigma_i x') = k(\sigma_i^{-1} x, x')$. Thus $k(x, x') = k(\sigma_i x, \sigma_i x')$, $k(x, \sigma_i x') = k(\sigma_i x, x')$ and $k(\sigma_i x, \sigma_j x') = k(\sigma_k x, x')$ for $i \neq j \neq k$. Thus we can use

$$k_f(x, x') = k(x, x') + k(\sigma_1 x, x') + k(\sigma_2 x, x') + k(\sigma_3 x, x')$$

as a covariance function for f instead of Equation (C1). This reduces the amount of computation needed to calculate the covariance functions by 75%.

Note that we don't need k to be completely isotropic for this simplification to hold, only that the covariance function is isotropic for any pair of inputs that swap in any of the permutations. So in the $H_2 - CO_2$ system, we require the length-scales to be the same for inputs 1 and 2, and the same for inputs 3, 4, 5 and 6.

REFERENCES

[1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).

[2] R. Chen, E. Jiao, H. Zhu, and D. Xie, Journal of Chemical Physics **133**, 104302 (2010).

[3] H. Li, P.-N. Roy, and R. J. Le Roy, Journal of Chemical Physics **132**, 214309 (2010).

[4]R. Hellmann, E. Bich, E. Vogel, and V. Vesovic, Journal of Chemical Physics **141**, 224301 (2014).

[5]Y. Cui, H. Ran, and D. Xie, Journal of Chemical Physics **130**, 224311 (2009).

[6]G. S. Kimeldorf and G. Wahba, Ann. Math. Statist. **41**, 495 (1970).

[7]C. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, 2006).

[8]T.-S. Ho and H. Rabitz, Journal of Chemical Physics **113**, 3960 (2000).

[9]A. P. Bartok and G. Csanyi, Int. J. Quant. Chem. **115**, 1051 (2015).

[10]A. P. Bartok, M. C. Payne, R. Kondor, and G. Csanyi, Phys. Rev. Letts. **104**, 136403 (2010).

[11]W. J. Szlachta, A. P. Bartok, and G. Csanyi, Phys. Rev. B **90**, 104108 (2014).

[12]M. Rupp, M. R. Bauer, R. Wilcken, A. Lange, M. Reutlinger, F. M. Boeckler, and G. Schneider, J. Chem. Phys. **136**, 074102 (2012).

[13]A. P. Bartok, M. J. Gillan, F. R. Manby, and G. Csanyi, Phys. Rev. B **88**, 054104 (2013).

[14]C. M. Handley, G. I. Hawe, D. B. Kell, and P. L. A. Popelier, Phys. Chem. Chem. Phys. **11**, 6365 (2009).

[15]M. J. Mills and P. L. Popelier, Computational and Theoretical Chemistry **975**, 42 (2011).

[16]M. J. L. Mills, G. I. Hawe, C. M. Handley, and P. L. A. Popelier, Phys. Chem. Chem. Phys. **15**, 18249 (2013).

[17]T. L. Fletcher and P. L. Popelier, Chemical Physics Letters **659**, 10 (2016).

[18]H. J. Werner *et al.*, "MOLPRO version 2012.1: A package of ab initio programs," http://www.molpro.net (2012).

[19]"GPy: A Gaussian Process framework in python," http://github.com/SheffieldML/GPy (2012–2015).

[20]This can be obtained from the maximum energy, before applying the energy cut-off, over the test data (if available) or training data.

[21]A. C. Legon and A. P. Suckley, J. Chem. Phys. **91**, 4440 (1989).

[22]K. deLange and J. R. Lane, J. Chem. Phys. **134**, 034301 (2011).

[23]S. Sheybani-Deloui, A. J. Barclay, K. H. Michaelian, A. R. W. McKellar, and N. Moazzen-Ahmadi, J. Chem. Phys. **143**, 121101 (2015).

[24]T. L. Cottrell, R. A. Hamilton, and R. P. Taubinger, Trans. Faraday Soc. **52**, 1310 (1956).

[25]J. Brewer and G. W. Vaughn, J. Chem. Phys. **50**, 2960 (1969).

[26]B. Mallu, G. Natarajan, and D. Viswanath, The Journal of Chemical Thermodynamics **21**, 989 (1989).

[27]O. Kunz, R. Klimeck, W. Wagner, and M. Jaeschke, *The GERG-2004 wide-range reference equation of state for natural gases* (Fortschr.-Ber. VDI, VDI-Verlag, Düsseldorf, 2007).

[28]M. P. Hodges, R. J. Wheatley, G. K. Schenter, and A. H. Harvey, J. Chem. Phys. **120**, 710 (2004).

[29]J. Beck and S. Guillas, arXiv (2014), 1410.0215v2.

[30]M. T. Oakley and R. J. Wheatley, J Chem Phys **130**, 034110 (2009).

[31]E. Snelson and Z. Ghahramani, in *Advances in Neural Information Processing Systems 18* (MIT press, 2006) pp. 1257–1264.