Gaussian Process Models of Potential Energy Surfaces with Boundary Optimisation

Jack Broad,[1, a)] Simon Preston,[1] Richard J. Wheatley,[2, b)] and Richard S. Graham[1]
[1)] School of Mathematical Sciences, University of Nottingham, Nottingham NG7 2RD, United Kingdom
[2)] School of Chemistry, University of Nottingham, Nottingham NG7 2RD, United Kingdom

(Dated: 27 September 2021)

A strategy is outlined to reduce the number of training points required to model intermolecular potentials using Gaussian processes, without reducing accuracy. An asymptotic function is used at long range and the cross-over distance between this model and the Gaussian process is learnt from the training data. Results are presented for different implementations of this procedure, known as boundary optimisation, across the following dimer systems: CO-Ne, HF-Ne, HF-Na$^+$, CO_2-Ne and $(CO_2)_2$. The technique reduces the number of training points, at fixed accuracy, by up to $\sim$ 49 %, compared to our previous work based on a sequential learning technique. The approach is readily transferable to other statistical methods of prediction or modelling problems.

I. INTRODUCTION

In any molecular simulation, approximations of the potential energy surfaces (PESs) that describe the relevant interactions are a pre-requisite. Traditionally, these approximations have been made using empirical potentials (force-fields)[1,2]. However, such force-fields have closed functional forms, which limit their capacity to capture the complicated topography of the PES. Furthermore, they are laborious to produce and may fail to capture accurately even the fitted data. As the accuracy of the simulation depends on the potential employed, much work has been devoted to developing force-fields that provide approximations of PESs with quantum-mechanical accuracy.

Such work includes methods to generate *ab initio* force fields[3] and attempts to 'learn' the potential via a machine learning algorithm[4,5]. This article concerns the latter approach, which has also been applied in other fields in chemistry and materials science[6–8]. This approach proceeds by training a statistical technique on a relatively small set of data from *ab initio* calculations on the PES of interest, known as the training set. Many such techniques have been employed to predict the energy in these algorithms, including neural networks[9–13], moment tensors[14–16] and Gaussian processes[17–34] (GPs).

Herein, GPs are employed as the statistical technique. An existing example of a force-field that uses GPs is FFLUX[35], which has been used to approximate the energies of weakly bound complexes[36] and water clusters[37], among other applications[38–40]. Furthermore, in the field of materials science, GP models that invoke a smooth overlap of atomic positions (SOAP) kernel[41] have seen successful applications to many systems[29–31,42–45].

GP models have also produced promising results in applications to intermolecular interactions[17–19]. Initially[17], the training sets for these models were constructed with Latin hypercube sampling[46–48] though sequential design strategies[49,50], which achieve a prescribed accuracy with a smaller training set, have been shown to outperform such methods[19]. Regardless of training set design, a fixed cross-over distance, R_{cross}, was imposed prior to training. This parameter defines a boundary beyond which a simple, long-range asymptotic function takes over prediction from the GP[17,19]. A similar approach is used in materials science, in which the contribution by one atom to the neighbour density of another is assumed to be zero if the two are separated by a distance in excess of R_{cross}[31,44]. Cross-over distances have been applied alongside other statistical methods of prediction too[5,51], with neural network[9–12] and moment tensor[15] models of PESs employing fixed, pre-determined cross-over distances.

Use of a cross-over distance limits the portion of the PES approximated by the statistical method to the region around the potential well. Consequently, fewer training points are required to develop an accurate model. Previously these distances were all fixed and specified *a priori*, however, they can be learnt from the training data. Herein the work of Uteva *et al*.[17,19] is extended into a sequential design strategy in which R_{cross} varies as a function of the number of training points and the species of the interacting atoms. This produces models of the same accuracy with fewer training points than the same design strategy with a fixed R_{cross}. The process by which R_{cross} is optimised is referred to here as boundary optimisation.

A. GP Regression

All GP models herein make predictions via GP regression, with detailed descriptions of GP regression theory available elsewhere[52–54]. This section briefly introduces the key concepts in the context of intermolecular potentials. For any intermolecular interaction the PES can be thought of as a multivariate function $f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^Z$, where $\mathbf{x}$ is a vector of inputs and Z is the number of elements in $\mathbf{x}$. Here the inputs are inverse interatomic separations, though promising results have also been obtained with Morse variables[24,55]. Only pair intermolecular interactions are considered, although GP models have been

[a)] Electronic mail: jack.broad@nottingham.ac.uk (author to whom correspondence should be addressed)
[b)] Electronic mail: richard.wheatley@nottingham.ac.uk

applied successfully to non-additive interactions[17,19,40] and the methodology outlined here can be extended to such cases straightforwardly.

When the outputs $f(\mathbf{x}_i)$ = Y_i are available at N values of i, the set $\{\mathbf{x}_i, \mathrm{Y}_i\}_{i=1}^{N}$ forms the training set. Letting $\mathbf{Y}$ be a vector of the observed energies from this set, GP regression can be used to approximate the value of $f(\mathbf{x})$ at a new point $\mathbf{x}_*$ as

$$f(\mathbf{x}_*) = \mathbf{K}_*^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{Y}, \tag{1}$$

where $\mathbf{K}_*$ is a vector of the covariances between $\mathbf{x}_*$ and all $\mathbf{x}_i$,

$$\mathbf{K}_* = \begin{bmatrix} k(\mathbf{x}_*,\mathbf{x}_1) & k(\mathbf{x}_*,\mathbf{x}_2) & \cdots & k(\mathbf{x}_*,\mathbf{x}_N)\end{bmatrix}, \tag{2}$$

and $\mathbf{K}$ is the positive-definite covariance matrix,

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1,\mathbf{x}_1)+\sigma_{\mathrm{n}}^2 & k(\mathbf{x}_1,\mathbf{x}_2) & \cdots & k(\mathbf{x}_1,\mathbf{x}_N) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2)+\sigma_{\mathrm{n}}^2 & \cdots & k(\mathbf{x}_2,\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N,\mathbf{x}_1) & k(\mathbf{x}_N,\mathbf{x}_2) & \cdots & k(\mathbf{x}_N,\mathbf{x}_N)+\sigma_{\mathrm{n}}^2 \end{bmatrix}.$$

Here σ_{n}^2 is the Gaussian noise variance, which accounts for the noise in the training data.

All entries in $\mathbf{K}$ are found by evaluating a covariance function, $k(\mathbf{x},\mathbf{x}')$, where $\mathbf{x}$ and $\mathbf{x}'$ are configurations from the training set. The same holds for $\mathbf{K}_*$, only with one training configuration replaced by the new point $\mathbf{x}_*$. A common example is the squared exponential covariance function, for which

$$k(\mathbf{x},\mathbf{x}') = \sigma_{\mathrm{f}}^2 \prod_{d=1}^{D} \exp\Big(-\frac{(x_d - x'_d)^2}{2l_d^2}\Big). \tag{3}$$

Here D is the total number of interatomic distances which comprise unique atomic pairs, σ_{f}^2 is the signal variance, x_d and x'_d are the inverse separations in $\mathbf{x}$ and $\mathbf{x}'$ which contain the dth atomic pair, and l_d is the lengthscale for the interaction between this pair of atoms. The hyperparameters of the covariance function are σ_{n}^2, σ_{f}^2 and the set of lengthscales, l_d.

This covariance function can be modified to account for the fact that two configurations can have equivalent energies due to symmetry. The result is the symmetric squared exponential covariance function, $k_{\mathrm{sym}}(\mathbf{x},\mathbf{x}')$, investigated by Uteva *et al*.[17], which is used here. The set of all permutations of the interatomic distances in $\mathbf{x}$ under which the energy surface is unchanged is denoted as P, while p is a single permutation within this set. Assuming that l_d is invariant under interchange of $\mathbf{x}_d$ and $\mathbf{x}'_d$,

$$k_{\mathrm{sym}}(\mathbf{x},\mathbf{x}') = \sum_{\mathrm{p}\in\mathrm{P}} k(\mathrm{p}\mathbf{x},\mathbf{x}'), \tag{4}$$

where $k(\mathbf{x},\mathbf{x}')$ is the squared exponential covariance function from equation 3. GPs also employ a mean function, which is taken here to be zero everywhere.

To specify the hyperparameters of $k_{\mathrm{sym}}(\mathbf{x},\mathbf{x}')$ that produce a model that achieves the best fit to the training data, the log of the marginal likelihood function, $\log(\mathscr{L})$, is maximised[52,54]

$$\log(\mathscr{L}) = -\frac{1}{2}\mathbf{Y}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{Y} - \frac{1}{2}\log|\mathbf{K}| - \frac{N}{2}\log(2\pi). \tag{5}$$

Equation 5 shows that this optimisation entails inversion of the N x N covariance matrix. This incurs a sizable computational cost, which translates to scaling of order $\mathscr{O}(N^3)$ for hyperparameter optimisation[56]. Consequently, though they achieve higher predictive accuracies than other statistical methods when modelling PESs[51,57,58], GPs require greater computational effort to train[51]. For prediction, meanwhile, $\mathbf{K}^{-1}\mathbf{Y}$ in equation 1 needs to be calculated once only, and as a result the cost scales linearly with N. Even so, molecular simulations that employ a GP model for which N is large are more computationally intensive than those that use a traditional forcefield.

These issues have led to attempts to minimise the training set size required by a GP model to achieve a given error. For two training strategies that achieve the same error, that which does so with fewer training points is more computationally efficient. Attempts to develop more computationally efficient training strategies have involved active learning or sequential design methods[19,29,59], composite kernels[24] and new sampling methods[27,60].

However, no attempt was made in these previous works to increase training efficiency by learning the optimal crossover distance from the data used in training. Our boundary optimisation approach has the potential to increase the gains in efficiency from these prior methods further still with little associated increase in computational time, provided a viable long-range approximation is available. Such an improvement is shown here, building upon the sequential design method of Uteva *et al*.[19].

B. Training Set Design

It was found by Uteva *et al*.[17] that the predictive performance of a GP for a PES is enhanced by using inverse interatomic separations rather than non-inverse separations to describe intermolecular configurations. This is because $k_{\mathrm{sym}}(\mathbf{x},\mathbf{x}')$ is a stationary kernel (*i.e.* it relies only on the distance between the two configurations being compared), meaning it assumes that the rate of change of the output (*i.e.* the interaction energy) with the input (the interatomic distances) is constant. This is not the case for a PES, where the energy varies rapidly with separation in the short-range repulsive wall but barely at all in the long-range asymptotic region. The $r \rightarrow r^{-1}$ conversion addresses this issue: for low r the rate of change in r^{-1} is faster than in r, meaning the change in the input more closely matches the change in the output under the former; meanwhile, for large r, the rate of change in r^{-1} is far smaller, as is that of the energy.

Previously, Latin hypercube (LHC) sampling, which is a space-filling design, has been employed to build data sets when modelling PESs with GPs[17–19,61]. The approach of Uteva *et al*.[17] entailed designing LHCs in inverse separations over a range of angles that specified the symmetry-distinct region[17]. The algorithm generates a large number of candidate LHCs and finds the minimum separation in each. The LHC with the largest such separation is selected under what is referred to here as the 'maximin' criterion. Quantum chemical calculations are undertaken on the selected LHC only. This LHC is then subject to a high energy cut-off, E_{cut}, and any configuration for which the interaction energy exceeds E_{cut} is discarded from the LHC[17]. In addition, a geometric constraint of 8.5 Å was placed on all LHCs to ensure no configurations with minimum separations above this threshold were included[17]. It was observed that at separations greater than the geometric constraint the GP predictions tended towards a small, non-zero constant[17]. As these predictions should tend towards zero, a long-range function derived from the multipole expansion[62] was introduced to approximate energies at large separations.

In a later work, Uteva *et al*.[19] presented active and sequential learning approaches, in which the model of the PES is progressively refined by adding new points to the GP. In this approach, training and validation of the GP models involves three data sets: a training, a reference and a test set. The training set is used in the GP regression. The reference set provides a pool of configurations from which new training points can be selected, while the test set determines the GP's accuracy against an independent data set.

One sequential design strategy presented by Uteva *et al*.[19] is the highest error search. This approach selects new training points based on the configuration in the reference set with the largest GP prediction error. We adopt this approach herein. However, Uteva et al.[19] applied this strategy only at separations below the fixed, pre-determined cross-over distance[19], meaning no new training data could be added at separations above R_{cross}.

The highest error search is best described as a sequential design rather than an active learning[63,64] method. This is because active learning methods are a subset of sequential design methods that compute the output corresponding to an input only at the point it is added to the training set. This means, in the context of modelling PESs, an active learning method would use a reference set comprising configurations for which the energies were not calculated. Active learning strategies have, however, been successfully used to develop GP[19,29,59], moment tensor[15] and neural network[13] models of PESs, and the methodology used here for boundary optimisation could be altered for use with the two set method of Uteva *et al*.[19], which is an active learning method.

When using a sequential design method, meanwhile, the energies in the reference and test sets are pre-computed using a relatively computationally inexpensive *ab initio* technique. Once a model with the requisite predictive accuracy is obtained, the energies in the minimal training set can be recalculated using a more accurate, and costly, *ab initio* method before use in applications. This process is known as transfer learning[65,66] and it allows a GP model of a PES to be produced with the accuracy of a high-level *ab initio* technique with relatively few computationally expensive calculations. For example, MP2[67] energies have been upgraded to CCSD(T)[68] energies for calculation of the CO_2-CO second virial coefficient[17].

TABLE I: The co-ordinates for the reference and test LHCs for each system. N_{ref} and N_{test} are the number of points in the reference and test sets respectively after application of the high-energy cut-off, while the maximum number of training points for models of each potential are given in the text. Also shown is the minimum energy across the reference and test sets, E_{min}, in Hartrees (E_{h}), though no attempt was made to approximate the global minimum energy for any potential.

System	Coordinate	Range	N_{ref}	N_{test}	E_{min}
CO-Ne	r^{-1}	0.01 to 0.67 Å^{-1}	1914	5718	-1.502 x 10^{-4}
	$\cos(\theta)$	-1 to 1			
HF-Ne	r^{-1}	0.01 to 0.67 Å^{-1}	2148	6468	-2.633 x 10^{-4}
	$\cos(\theta)$	-1 to 1			
HF-Na^+	r^{-1}	0.01 to 0.67 Å^{-1}	2760	8416	-2.518 x 10^{-2}
	$\cos(\theta)$	-1 to 1			
CO_2-Ne	r^{-1}	0.01 to 0.67 Å^{-1}	5057	5072	-2.895 x 10^{-3}
	$\cos(\theta)$	0 to 1			
$(CO_2)_2$	r^{-1}	0.01 to 0.67 Å^{-1}	5810	5837	-1.975 x 10^{-3}
	$\cos(\theta_1)$	0 to 1			
	$\cos(\theta_2)$	0 to 1			
	ϕ	0 to 180$^{\text{o}}$			

II. METHODOLOGY

A. Overview

In previous applications of both LHC learning[17] and sequential design methods[19] a fixed value of the cross-over distance, R_{cross}, was determined *a priori*. Rather than fix R_{cross} prior to training, however, it is possible to determine its optimal value from the reference data. This is boundary optimisation and is possible because a sequential design method permits R_{cross} to be varied each time the GP is updated. Consequently, the size of the region over which GP regression is used for prediction will grow with the predictive accuracy of the GP.

Boundary optimisation may lead to increased accuracy for the following reason. When the number of training points, N_{TP}, is low the predictive accuracy of the long-range function at the outer edge of the potential well will exceed that of the GP. Consequently, it is anticipated that allowing the long-range function to approximate the energies of configurations in this region at low N_{TP} will increase the predictive accuracy of the overall model and facilitate more efficient model development.

B. Data Set Generation

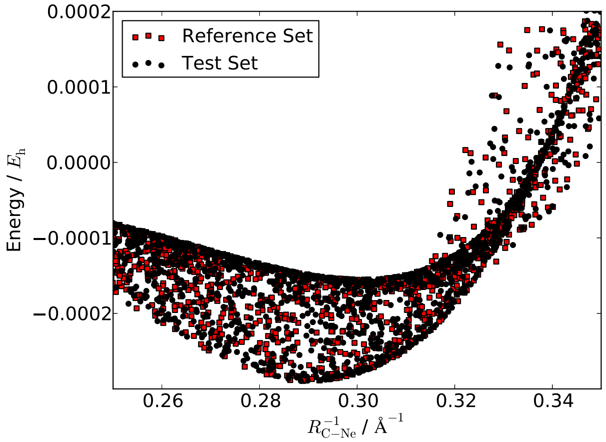


FIG. 1: The calculated energies in the reference set (red) and the test set (black) of the CO_2-Ne potential for separations between 2.85 Å and 4 Å. Both sets contain ∼ 5000 configurations (see table I).

Both the reference and test sets for each chemical system were generated under the LHC design strategy discussed in I B. All energies were calculated in Molpro[69] using MP2 with an aug-cc-pVTZ basis set and the counterpoise correction with the only exception being HF-Na^+, which used an aug-cc-pwCVTZ basis set instead. The specifications of the LHCs for each system are given in table I. Therein r is the distance between the bond centres (not centres of mass), θ is the angle between r and the bond axis of the molecule, θ_1 and θ_2 are the angles between r and the bond axis of the first and second CO_2 molecules respectively, and ϕ is the torsional angle between the two CO_2 molecules. The molecules were kept rigid, with r_{CO} = 1.1283 Å for CO, r_{CO} = 1.1632 Å for CO_2 and r_{HF} = 0.9170 Å. However, boundary optimisation could be applied to non-rigid systems straightforwardly, with the only additional requirement being generalisation of the long-range function to non-rigid molecules. A larger geometric constraint of 100 Å (instead of 8.5 Å[17]) was employed to probe the long range behaviour of the system. The maximum value of N_{TP} was 100 for all systems apart from $(CO_2)_2$, which used 300 training points at most.

A high energy cut-off, E_{cut}, was applied to the reference and test sets to remove configurations with interaction energies which exceeded its value. E_{cut} = 0.005 E_{h} for all systems apart from HF-Na^+, for which E_{cut} = 0.05 E_{h} because the charge-dipole interaction increases the well-depth. (1 E_{h} ≈ 27.211 eV ≈ 2625.5 kJ mol^{-1}.)

For systems with $N_{\mathrm{test}} \gg N_{\mathrm{ref}}$ the independence of the test set is self-evident. However, for systems where $N_{\mathrm{test}} \approx N_{\mathrm{ref}}$ it is also true that the test set is independent. This follows because, although the reference and test sets for each system were designed using the same 'maximin' strategy, the stochastic nature of the LHC algorithm means that separate LHCs contain completely independent sets of configurations. Furthermore, the 'maximin' criterion is based on just one separation in the whole data set, meaning that two LHCs with similar maximin will still be dissimilar. This is demonstrated in figure 1, which shows the energies against the inverse C-Ne separation for the reference and test sets used in training models of the CO_2-Ne potential.

TABLE II: The properties included in the multipolar long-range functions of each system, as well as the *ab initio* methods used in their calculation. All calculations were carried out using an aug-cc-pVQZ basis set apart from those for the dispersion coefficients, which used an aug-cc-pVTZ basis set.

System	Dipole	Quadrupole	Polarizability	Dispersion
CO-Ne	✓	X	✓	✓
HF-Ne	✓	X	✓	✓
HF-Na^+	✓	✓	✓	✓
Level of Theory	MRCI[71,72]	MRCI	MP2	CCSD

C. Long-range asymptotic functions

For the long-range energy model, multipole series were employed for all systems apart from HF-Na^+. The contributions included in the multipolar long-range functions are shown in table II for each system apart from CO_2-Ne, for which the function is already described in previous work[17], and $(CO_2)_2$. The latter was developed prior to the other multipolar long-range functions and uses atomistic charge, dipole, quadrupole and polarizability contributions from Hartree-Fock[70] theory, which were scaled to give the known total molecular properties. For HF-Na^+ a fitted long-range function was used, which was derived by fitting a sum of two power laws between two points where the energy was predicted by GP regression. More information on the motivation for and derivation of this function is found in appendix A.

D. Classification of Phase Space using a Boundary

When modelling PESs using GPs it is often necessary to classify configurations as suitable for prediction via the GP or via a long-range asymptotic function. In the work of Uteva *et al.*[17,19] the classifier formed a boundary from the superposition of atom-centred spheres defined by a single constant, R_{cross}. Specifically, if any interatomic distance was less than R_{cross} the GP was used. As R_{cross} was fixed, this classifier is referred to here as $\mathrm{C_{fixed}}$. Denoting the region in which GP regression was used for prediction as $\mathrm{A_{GP}}$ and the region which employed the long-range function as $\mathrm{A_{LR}}$, under $\mathrm{C_{fixed}}$ these regions were

$$\mathrm{A_{GP}} = \{\mathbf{r} : \min(\mathbf{r}) \leq R_{\mathrm{cross}}\} \tag{6}$$

and

$$\mathrm{A_{LR}} = \{\mathbf{r} : \min(\mathbf{r}) > R_{\mathrm{cross}}\}, \tag{7}$$

where $\mathbf{r}$ is a set of intermolecular atom-atom distances, and $\min(\mathbf{r})$ is the smallest separation in $\mathbf{r}$.

In boundary optimisation, R_{cross} varies according to the GP accuracy. Thus R_{cross} is not constant but a model parameter, which is learnt from the reference data. As this classifier is still parameterised by a single value, R_{cross}, it is referred to here as C_{single}.

More elaborate classifiers are possible by using more detailed parametric forms to define the boundary region. A simple way of defining a more complex classifier is for the value of R_{cross} to depend on the atom types that comprise the interatomic distance. The resulting classifier is referred to here as C_{multi}. For a system of molecules with D interatomic pairs of chemically different atoms, using C_{multi} requires the vector of cross-over distances $\mathbf{R}_{\text{cross}} = (R_1, \ldots, R_d, \ldots, R_D)$. This defines a multiple-parameter boundary region, as follows,

$$\text{C}_{\text{multi}}(\mathbf{r}) = \begin{cases} A_{\text{GP}}, & \text{if } \min_d(\mathbf{r}) \leq R_d \text{ for any } d \\ A_{\text{LR}}, & \text{if } \min_d(\mathbf{r}) > R_d \text{ for all } d, \end{cases} \quad (8)$$

where $\min_d(\mathbf{r})$ is the minimum separation in $\mathbf{r}$ that involves an atomic pair of type d.

The optimum value of the classifier parameters are determined by minimising the error between the model and the reference set (*i.e.* by minimising the training error), meaning the sizes of A_{GP} and A_{LR} vary with the GP. The sum of squared errors, SSE_{tot}, over the two regions is

$$\text{SSE}_{\text{tot}} = \text{SSE}_{\text{GP}} + \text{SSE}_{\text{LR}}, \quad (9)$$

where

$$\text{SSE}_{\text{method}} = \sum_{i=1}^{N_{\text{method}}} (\hat{\text{Y}}_i - \text{Y}_i)^2. \quad (10)$$

Here, "method" denotes either GP or LR, N_{method} the number of points in A_{method}, $\hat{\text{Y}}_i$ the prediction of the energy for the ith configuration from the desired method and Y_i the calculated energy of the same configuration. The RMSE against the test set, $\text{RMSE}_{\text{test}}$, is given by

$$\text{RMSE}_{\text{test}} = \left(\frac{\text{SSE}_{\text{test}}}{N_{\text{test}}}\right)^{\frac{1}{2}}, \quad (11)$$

where SSE_{test} is SSE_{tot} over the test set and N_{test} is the number of configurations in the test set. $\text{RMSE}_{\text{test}}$ is therefore a function of R_{cross} with discrete steps, as the RMSE changes only when a change in R_{cross} causes a configuration in the test set to be re-classified. Both C_{single} and C_{multi} are simple, parametric classifiers that pre-impose a mathematical form on the classification. Hence, neither is expected to be optimal with respect to the RMSE against the reference or test sets. That is, a more complicated boundary than that described by these classifiers will likely produce a lower RMSE against a given data set. However, it is shown later that an artificial 'ideal' classifier produces only very marginal improvements over C_{multi}, suggesting this classifier provides a very good balance of simplicity and accuracy.

E. Overall algorithm

Using the data and calculation methods above, the algorithm generates a GP model sequentially as follows: train the GP to the current training set; select the classifier parameters by minimising the RMSE against the reference set; move a new point from the reference to the training set based on the largest error. Below are further details of each step, along with how the choice of classifier and placement strategy affects the algorithm.

1. GP training

All GPs described herein were trained using the GPy[73] package in Python 2.7. Optimisation of the hyperparameters was carried out by maximising $\log(\mathscr{L})$ using 20 independent restarts whenever a configuration was added to the training set. Moreover, a gamma distribution with an expectation of one and a variance of two was used as a prior on all hyperparameters for all systems. This was to weakly penalise large hyperparameter values, given that the expected values are typically of order 0.1 or below.

2. Direct Search Algorithm

When using C_{single}, R_{cross} can be optimised via a direct search that exploits how the RMSE varies as a piecewise constant function of R_{cross}. Because of this feature, all possible values of SSE_{tot} (and hence the RMSE) can be computed readily against the reference set. A direct search of these values is guaranteed to find the global minimum. Full details of the direct search are given in algorithm 1.

Algorithm 1 Direct Search Algorithm

```
Compute the squared errors for the GP and long-range function and $\min(\mathbf{r})$ at each configuration in the reference set.
Order the configurations from smallest to largest in terms of $\min(\mathbf{r})$.
Approximate $\text{SSE}_{\text{tot}}$ initially from the squared errors of the long-range function alone.
Iterate through the $\min(\mathbf{r})$ from the lowest to the highest:
    a: set $R_{\text{cross}} = \min(\mathbf{r})$, which moves a single configuration from $\text{A}_{\text{LR}}$ to $\text{A}_{\text{GP}}$;
    b: update $\text{SSE}_{\text{tot}}$ by deducting the squared long range error of the moved point from $\text{SSE}_{\text{LR}}$ and adding its squared GP error to $\text{SSE}_{\text{GP}}$;
    c: store the new $\text{SSE}_{\text{tot}}$.
Select whichever value of $R_{\text{cross}}$ corresponds to the smallest value of $\text{SSE}_{\text{tot}}$.
```

The direct search algorithm is computationally cheap for the following reasons. Each instance of the direct search requires only a single set of predictions from each of the long range function and GP regression. As the long range function is fixed throughout the sequential design process, these predictions need only be calculated once at the start. The GP predictions need to be re-calculated whenever the GP is updated, which occurs once per step of the sequential design. However, these predictions are already required by a sequential design step when choosing the new training point. Furthermore, calculation of all possible squared error values (step 4) is cheap because calculating each value in order requires only a simple update of $\mathrm{SSE}_{\mathrm{LR}}$ and $\mathrm{SSE}_{\mathrm{GP}}$. Consequently, the direct search is fast compared to the other steps of the sequential design algorithm and can therefore be undertaken at each design step with negligible additional computational effort.

3. Orthogonal direct search algorithm

An adaption of the direct search algorithm is required for $\mathrm{C}_{\mathrm{multi}}$, as this requires a multidimensional optimisation of multiple classifier parameters. This is called the orthogonal direct search as it optimises one element of $\mathbf{R}_{\mathrm{cross}}$ while keeping the others fixed. The single element that varies is optimised using the direct search algorithm, as this is guaranteed to return the best minimum along that 1D slice of $\mathbf{R}_{\mathrm{cross}}$. Although orthogonal optimisations can be time-consuming, the speed of the one-dimensional direct search means that repeating it multiple times for all cross-over distances is feasible.

Algorithm 2 Orthogonal Direct Search Algorithm

```
1: Choose limits $NI_{\max}$ and $NR_{\max}$ on the number of iterations and restarts respectively (see the text for details).
2: Assemble the array of minimum distances $\mathbf{M}$, as follows:
   a: For every configuration, collect all interatomic distances which comprise the same atoms types into group $d$ and find $\min_d(\mathbf{r})$ for each $d$.
   b: Arrange the lists of $\min_d(\mathbf{r})$ values in an $N$ x $D$ array, $\mathbf{M}$, where $N$ is the number of configurations in the data set and $D$ is the number of unique atomic pairs in $\mathbf{r}$.
   c: For each column in $\mathbf{M}$, order the values from smallest to largest to produce an ordered list in each column.
3: For each restart, select a row from $\mathbf{M}$ at random to be the initial guess at $\mathbf{R}_{\mathrm{cross}}$.
4: For each $d$ in turn, fix all cross-over distances apart from $R_d$ and find the optimal value of $R_d$ using a direct search.
5: Repeat step 4 until $NI = NI_{\max}$ or the elements of $\mathbf{R}_{\mathrm{cross}}$ remain unchanged; save this $\mathbf{R}_{\mathrm{cross}}$ and its corresponding $\mathrm{SSE}_{\mathrm{tot}}$.
6: Repeat steps 3-5 until $NR = NR_{\max}$.
7: Select the $\mathbf{R}_{\mathrm{cross}}$ that corresponds to the lowest value of $\mathrm{SSE}_{\mathrm{tot}}$.
```

The orthogonal direct search proceeds via algorithm 2. As in the one-dimensional direct search, the square errors of the long-range and GP predictions at each configuration are precomputed for this algorithm. The orthogonal search algorithm is not guaranteed to find the global minimum, as local minima may exist in the RMSE landscape. Hence, $NR_{\max}$ restarts are performed with randomly selected starting points. Each restart involves $NI_{\max}$ optimisations of each R_d. Here $NI_{\max}$ = 15 and $NR_{\max}$ = 5 for all systems. If the values in $\mathbf{R}_{\mathrm{cross}}$ converge such that further one-dimensional searches in any orthogonal direction do not change its elements prior to $NI = NI_{\max}$, $\mathbf{R}_{\mathrm{cross}}$ is saved and the next restart undertaken. In fact, it was rare that NI reached $NI_{\max}$ for any of the systems explored here. Moreover, despite the low $NR_{\max}$ employed, the same minimum was usually found across multiple restarts.

The direct search offers a method that is at once very fast, designed for the discrete-stepped surface and perfect in a single dimension. This latter property means it skips over any local minima in a given orthogonal direction. As such, optimal cross-over distances are obtained quickly and reproducibly under this method. Typically, about 80 % of random restarts return the same set of cross-over distances for a given system and a given training set.

4. Training Point Placement Methods

Once the classifier parameters are optimized, the PES model is specified and the next training point determined from the highest error method. The point with the greatest error can be selected either from A_{GP} alone or from the union of A_{GP} and A_{LR}. Using A_{GP} alone is referred to as the constrained placement method, which proceeds via algorithm 3. Steps 2-5 of this algorithm comprise a stage of training, with the $\mathrm{RMSE}_{\mathrm{test}}$ also calculated at each such stage.

Algorithm 3 Constrained Placement Algorithm

```
1: Select the configuration in the reference set with the highest energy; add this configuration to the training set and remove it from the reference set.
2: Retrain the GP to the updated training set.
3: Determine the boundary that minimises the RMSE against the reference set (see subsection II E 2).
4: Find the configuration in $\mathrm{A}_{\mathrm{GP}}$ for which the GP error is highest, where $\mathrm{A}_{\mathrm{GP}}$ is defined by the boundary from the previous step.
5: Add this configuration to the training set and remove it from the reference set.
6: Repeat steps 2-5 until the desired RMSE or number of training points is reached.
```

Choosing new points from the union of A_{GP} and A_{LR} is named the open placement method. This proceeded identically to the constrained placement method (algorithm 3) except the highest error point was from either A_{GP} or A_{LR}. In each region this point is found using the corresponding method of prediction (*i.e.* the error of the long-range function was used in A_{LR}).

Models were trained under both point placement methods because each held potential advantages over the other. The constrained placement method ensures all training points are added in the GP region and so are of immediate use in prediction. Meanwhile, the open placement method is capable of immediately placing points in regions of the PES where the long-range function performed poorly, potentially transferring these regions to A_{GP} more rapidly than under the constrained placement approach.

5. *Closest Model Training Strategy*

In the present work, all training strategies used are combinations of a classifier and a point placement method. A further strategy, which is intended for comparison only, is the closest model training strategy. This employs the open placement method to select training points and classifies a configuration using C_{optimal}, where

$$C_{\text{optimal}}(\mathbf{r}) = \begin{cases} A_{\text{GP}}, & \text{if } \text{SE}_{\text{GP}} \leq \text{SE}_{\text{LR}} \\ A_{\text{LR}}, & \text{otherwise} \end{cases}. \qquad (12)$$

Here, $\text{SE}_{\text{method}}$ is the squared error in the prediction from "method" at $\mathbf{r}$. C_{optimal} is so named because it classifies a configuration based on whether GP regression or the long-range function best approximate its energy.

Equation 12 shows that C_{optimal} employs no boundary and so requires prior knowledge of the energy of a configuration in order to classify it. Consequently, models obtained from the closest model strategy are unsuitable for prediction. However, this method represents an 'ideal' classifier that is guaranteed to find the optimal $\text{RMSE}_{\text{test}}$ for a given GP model. Hence C_{optimal} is useful for estimating how inaccuracies in the parametric classifiers affect the training efficiency. If models from C_{optimal} significantly outperform the other classifiers, this suggests the other classifiers are too simple to properly approximate the true boundary.

There are circumstances where C_{optimal} may not give an optimal RMSE against the test set. There may be short-range hypersurfaces where the interaction energy is predicted near-exactly by the long-range function due to chance. Points in the reference set that are near these hypersurfaces will be classified by C_{optimal} as part of A_{LR} instead of A_{GP}. If the configuration density of the reference set around such a hypersurface is insufficiently high, no training points will be added in its vicinity. Consequently, points in the test set near to the hypersurface will be inadequately approximated by either method of prediction; the GP will perform poorly due to a lack of nearby training points and the long-range function will be inaccurate for short-range points that are not extremely close to the hypersurface. This problem was avoided by using dense reference sets and by not using the constrained placement strategy with C_{optimal}.

6. *Combining Classifiers and Placement Strategies*

The sequential design method, or training strategy, requires a choice of classifier and placement strategy. The combinations of these examined in this work are given in table III. Methods that do not involve C_{optimal} can make predictions and so are suitable for applications. The method involving C_{optimal} can only classify points if the true energy is already known, and so is only useful to estimate the loss in performance due to inaccuracies in the parametric classifiers. The fixed boundary method corresponds to the method of Uteva *et al.*[19] and is included to allow comparison with this prior method, on which the new methods build. The method of Uteva *et al.*[19] was previously shown to significantly reduce the number of training points compared to LHC design. It is demonstrated below that these new boundary optimisation methods improve further the already efficient methods of Uteva *et al.*[19].

TABLE III: The classifier and training point placement method for all training strategies examined here.

Training Strategy	Classifier	Point Placement Method
Single-Constrained	C_{single}	Constrained Placement
Multi-Constrained	C_{multi}	Constrained Placement
Single-Open	C_{single}	Open Placement
Multi-Open	C_{multi}	Open Placement
Closest Model	C_{optimal}	Open Placement
Fixed Boundary	C_{fixed}	Constrained Placement

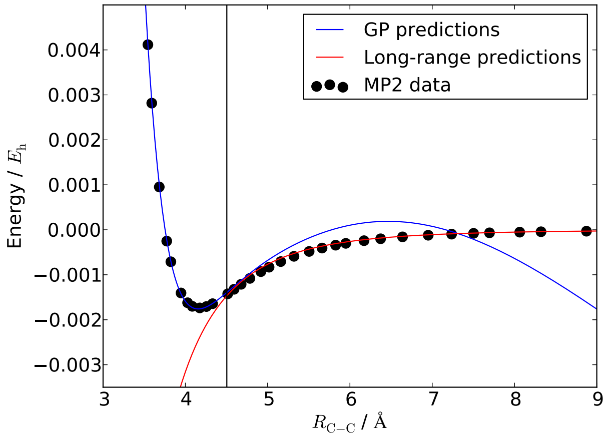


FIG. 2: The predictions of the GP (blue) and the long-range function (red) for a slice through the PES of the $(CO_2)_2$ potential in which the two molecules are in a T-shaped configuration. The cross-over distance for this model is shown by the black line at 4.505 Å. Data points (black circles) are independent of the GP training data.

III. RESULTS AND DISCUSSION

Comparisons of the performances of the different training strategies are made using the HF-Ne, $HF\text{-}Na^+$, CO-Ne, CO_2-Ne and $(CO_2)_2$ potentials. These were selected as they provide a range of interaction types and well depths to test the robustness of the new training strategies.

The number of training points is N_{TP}, which in each system was less than 10 % of the number of configurations in the corresponding reference set (see table I and the related text). Consequently, the training sets for the models discussed are candidates for transfer learning because their small size makes, for example, CCSD(T) calculations possible for the whole training set. Boundary optimisation is anticipated to improve training efficiency as, for small N_{TP}, the long-range function will outperform the GP at the outer edge of the potential well. This is illustrated in figure 2 for the $(CO_2)_2$ system,

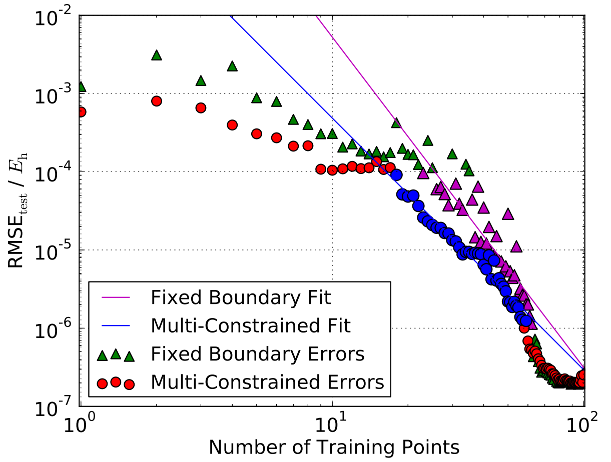


FIG. 3: Plot of RMSE$_{\mathrm{test}}$ / E_{h} against N_{TP} on a $\log_{10}$ scale for models of the HF-Ne PES trained via the multi-constrained (circles) and fixed boundary (triangles) training strategies. The blue circles and pink triangles are points which were included in the fitting of the lines shown.

using a GP model trained under the single-constrained strategy up to $N_{\mathrm{TP}} = 21$.

Figure 2 shows that the predictive accuracy of the long-range function exceeds that of GP regression for configurations with C-C separations above the cross-over distance. Despite this, in the fixed boundary method these configurations would be predicted with the GP. This not only reduces overall accuracy, but also means new training points must be placed at long range to address this, when allowing the long-range function to predict these energies would give adequate accuracy.

The impact of the above on training efficiency compared to fixed boundary training is shown for HF-Ne in figure 3. For C_{multi} the RMSE falls faster with N_{TP} compared to C_{fixed}, indicating improved training efficiency. This improvement is most pronounced when N_{TP} is low and so RMSE$_{\mathrm{test}}$ is high. This is expected because increasing the size of the training set increases the size of the GP region, meaning R_{cross} in the boundary-optimised strategies will approach the fixed value of 8.5 Å. Consequently, the difference between the fixed boundary and boundary-optimised models will close as N_{TP} increases. Equivalent plots to figure 3 for all training strategies for all systems are found in the supplementary material. The RMSE data are somewhat noisy. Possible causes of this noise are minor variations in the hyperparameters upon retraining, the discrete nature and stochastic design of the reference set and the low values of N_{TP} meaning that addition of a single point has a non-smooth effect on the RMSE. The new methods herein have considerably lower noise than the prior fixed boundary method.

To compare the efficiency gain over fixed boundary training across all new strategies from subsections II E 5 and II E 6, a metric, E, is used. This compares the N_{TP} required by two different strategies to achieve a given RMSE$_{\mathrm{test}}$. When comparing a new training strategy with fixed boundary training,

TABLE IV: The equation of the lines of best fit for RMSE$_{\mathrm{test}}$ of models from all training strategies for CO-Ne as power laws in the number of training points, N_{TP}. Also shown are the R^2 values of each fit on the data. All fits were over points in the range 1 x 10^{-6} $E_{\mathrm{h}} \leq$ RMSE$_{\mathrm{test}} \leq$ 1 x 10^{-4} E_{h}.

Training Strategy	Line of best fit to RMSE$_{\mathrm{test}}$	R^2
Single-Constrained Placement	$4.694N_{\mathrm{TP}}^{-3.842}$	0.929
Multi-Constrained Placement	$1.551N_{\mathrm{TP}}^{-3.647}$	0.944
Single-Open Placement	$2.901N_{\mathrm{TP}}^{-3.718}$	0.921
Multi-Open Placement	$3.113N_{\mathrm{TP}}^{-3.830}$	0.958
Closest Model	$0.865N_{\mathrm{TP}}^{-3.490}$	0.944
Fixed Boundary	$546.0N_{\mathrm{TP}}^{-5.069}$	0.914

$$E(\mathrm{RMSE}_{\mathrm{test}}) = \frac{N_{\mathrm{TP}}(\mathrm{RMSE}_{\mathrm{test}})}{N_{\mathrm{TP,fixed}}(\mathrm{RMSE}_{\mathrm{test}})}\mathrm{x}100\%, \qquad (13)$$

where N_{TP} and $N_{\mathrm{TP,fixed}}$ are the numbers of training points required by the new strategy and fixed boundary training, respectively. These quantities and E are shown as functions of RMSE$_{\mathrm{test}}$ as they vary with its value.

For a given RMSE$_{\mathrm{test}}$, the values of N_{TP} and $N_{\mathrm{TP,fixed}}$ are determined from least squares fits of $\log_{10}$(RMSE$_{\mathrm{test}}$) versus $\log_{10}(N_{\mathrm{TP}})$, with examples of such fits shown in figure 3 for the HF-Ne potential. Fits are made in the region where the RMSE decays as a power law of N_{TP}. For all systems this corresponds to 1 x 10^{-6} $E_{\mathrm{h}} \leq$ RMSE$_{\mathrm{text}} \leq$ 1 x 10^{-4} E_{h}, apart from HF-Na$^+$ where the region is 1 x 10^{-5} $E_{\mathrm{h}} \leq$ RMSE$_{\mathrm{text}}$ $\leq$ 1 x 10^{-3} E_{h} due to the larger high energy cut-off for this potential. The fits provide continuous lines to interpolate to any RMSE$_{\mathrm{test}}$ within the above range. This enables a comparison of N_{TP} between training strategies at fixed RMSE$_{\mathrm{test}}$ that accounts for the somewhat noisy data. Moreover, this is the range of errors in which the models are useful for applications and the decrease in log(RMSE$_{\mathrm{test}}$) with log(N_{TP}) is linear.

The fitted equations for each training strategy for the CO-Ne system are given in table IV as power laws in N_{TP}. Equivalent tables for all other systems are found in the supplementary material. As the data are noisy, these tables also show the R^2 value of each fit. In the case of CO-Ne, these evidence the high quality of the fits. In fact, no fit from any training strategy for any system achieves an R^2 value lower than 0.8907 (from the fixed boundary training strategy on the HF-Na$^+$ system). Furthermore, figure 3 and the corresponding plots in the supplementary material show that the RMSE data follow a straight line (in a log-log plot). This implies that the R^2 arises from scatter in the data rather than unsuitability of the fitting function.

Re-arranging the equations in table IV provides expressions for N_{TP} in terms of RMSE$_{\mathrm{test}}$ for the CO-Ne potential. These give E as a function of RMSE$_{\mathrm{test}}$, via equation 13, for all new training strategies herein. This was done for all other potentials as well, with plots of E against RMSE$_{\mathrm{test}}$ for all training strategies for each system given in figure 4.

As observed earlier, the training efficiency gains in figure 4

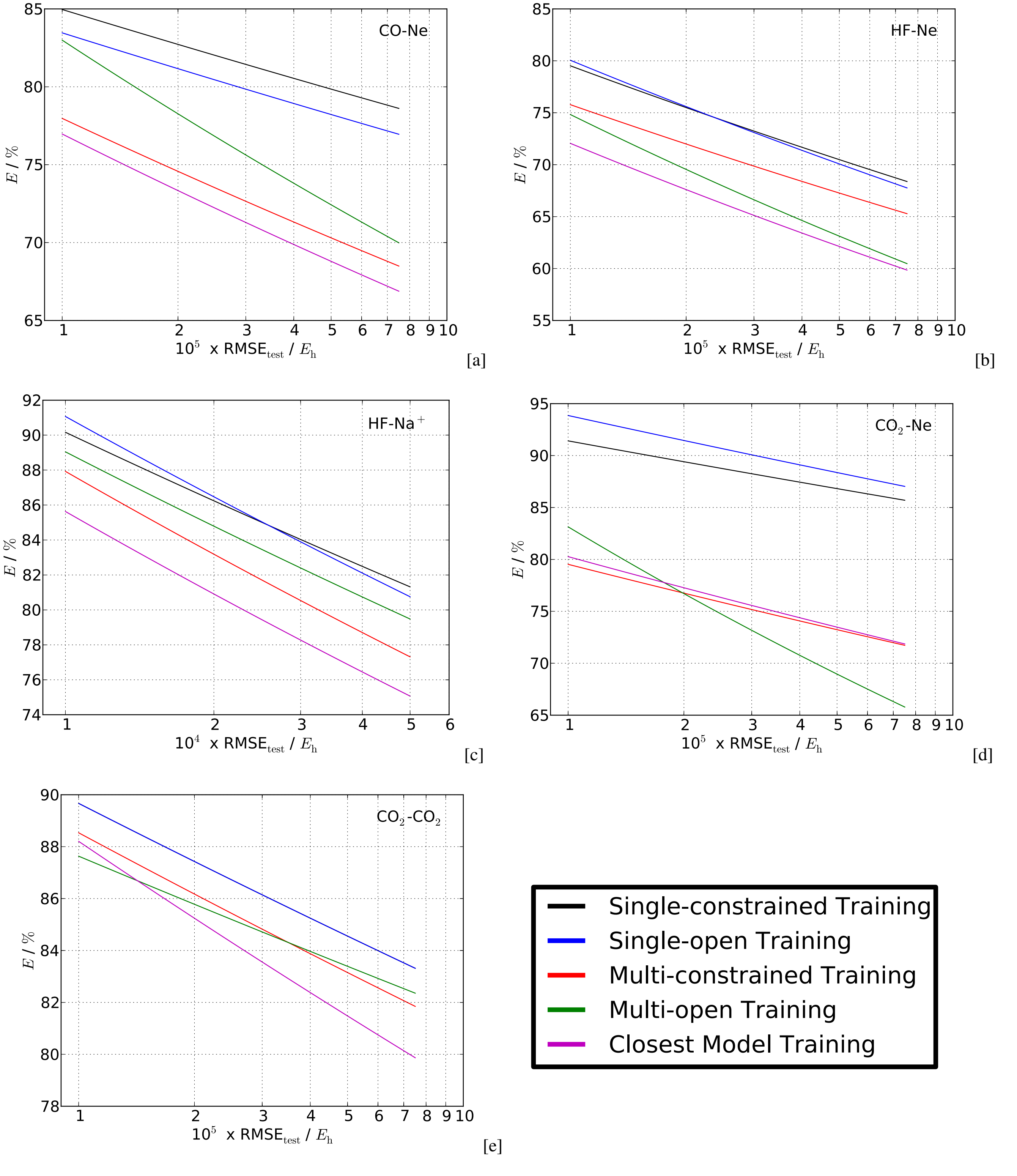


FIG. 4: Plots of E against RMSE$_{\text{test}}$ for all potentials are shown in parts (a) to (e). The potential referred to in each frame is shown in the upper right corner.

are more pronounced at high $RMSE_{test}$ for all training strategies across all systems. This suggests that boundary optimisation is most effective when the training set is small, making this technique ideal for applications where a computationally cheap but less accurate potential is required. Nevertheless, significant reductions in the number of training points are also obtained in the RMSE range where PESs become useful for first principles predictions. For example, a GP potential with an RMSE of 3 x 10^{-4} eV per atom (1.1 x 10^{-5} E_h per atom) was employed in a recent simulation of the thermal properties of β-Ga_2O_3[31]. Furthermore, Uteva *et al*. successfully determined the CO_2-CO second virial coefficient using a GP PES with an RMSE of 2.4 x 10^{-5} E_h[17]. In this RMSE range the boundary optimisation methods typically reduce the required number of training points by 15-33% (see figure 4).

Generally, the closest model strategy generates the largest efficiency gain, while the smallest improvement comes from strategies involving C_{single}. Efficiency gains only slightly below those from the closest model strategy are achieved by the strategies that use C_{multi}. This hierarchy of improvement implies that the choice of classifier, rather than point placement strategy, is most important because strategies with the same classifier perform more similarly than those with the same point placement method. The closest model strategy is included only to illustrate the total training efficiency gain possible from an 'ideal' classifier, and it is encouraging that the best boundary optimisation methods are close to this ideal case. Indeed, the difference in E between this ideal method and the closest-performing C_{multi} strategy never exceeds $\sim$ 3 % for any system. This implies that C_{multi} captures the true shape of the boundary region for the systems explored sufficiently. Thus introducing a more detailed classifier would not be worth the increased cost of evaluating the PES.

C_{multi} always outperforms C_{single}. However, for the CO_2-Ne potential (figure 4d), C_{multi} also outperforms the closest model strategy. This is because the long-range function is nearly exact for a group of short-range configurations in the reference set. This is shown in figure 5 by the thin 'peninsula' of points that are best estimated by the long range function (in red) which encroaches deep into the GP region (in blue). This 'peninsula' exists because the long-range function is of higher energy than the MP2 data in some regions of the PES and of lower energy in others, meaning there must be some hypersurface in between where the two are equal. Prediction for test configurations close to the 'peninsula' is problematic under $C_{optimal}$ unless the reference set is very dense in this region. This is because the exact predictions of the long-range function on the 'peninsula' mean no training points are added there, leading the closest model strategy to perform relatively poorly for the CO_2-Ne potential.

The total gain in training efficiency achieved by boundary optimisation varies somewhat between systems. While the best-performing training strategy for HF-Ne improved training efficiency by between 25-39 % (*i.e.* E = 61-75 %), for $(CO_2)_2$ the gain was only 12-18 % (E = 82-88 %). The more limited gains for $(CO_2)_2$ may be because the *a priori* choice of R_{cross} = 8.5 Å, required for the fixed boundary method, is reasonable for this system. Nevertheless, even for the $(CO_2)_2$ potential, use of multi-constrained training confers an efficiency gain of $\sim$ 18 % over fixed boundary training. This means that for all of the potentials explored, use of a training strategy that employs C_{multi} confers a useful improvement over fixed boundary training.

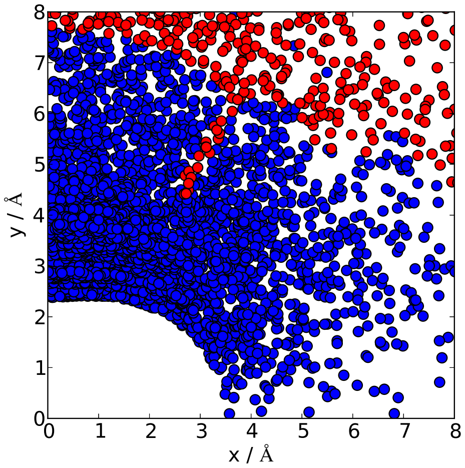


FIG. 5: Plot of the x and y coordinates of the Ne in the CO_2-Ne system for configurations in the reference set. The CO_2 molecule is aligned along the x-axis with the C at the origin. Points are classified as GP points (blue) or asymptotic points (red) by $C_{optimal}$ using a model from the closest model strategy trained up to N_{TP} = 100.

Boundary optimisation improves the training efficiency due to more effective placement of training points. This is illustrated in figure 6, which shows, for HF-Ne, the differences in training point placement for three training methods: fixed boundary, single-constrained and closest model. While all place most points at separations below 3 Å, indicating that the repulsive wall is the hardest region to model, the placement of points at larger separations diverges between methods. For the first 20 training points few configurations are placed beyond 3 Å for the single-constrained and closest model strategies. Thereafter, the distance at which training points are added slowly increases, even for the closest model strategy, which does not employ a boundary directly. In fact, single-constrained training adds training points beyond 8 Å only after $\sim$ 90 training points have been placed and the closest model strategy does not add any training points above 7 Å at all.

In contrast, the fixed boundary training method adds its fifth training point at a minimum separation above 5 Å and its eighth at 8.5 Å. This demonstrates that a fixed boundary strategy switches between placing training points in the repulsive wall and at the boundary from the onset of training. This difference is because fixed boundary training requires the GP to predict energies at separations up to 8.5 Å from the start of training. Consequently, training points must be added at separations near the 8.5 Å boundary from the onset, even though the energies at these separations are very small and generally well-approximated by the long-range function. This contrasts with the boundary-optimised and closest model strategies, which allow the long-range function to approximate con-

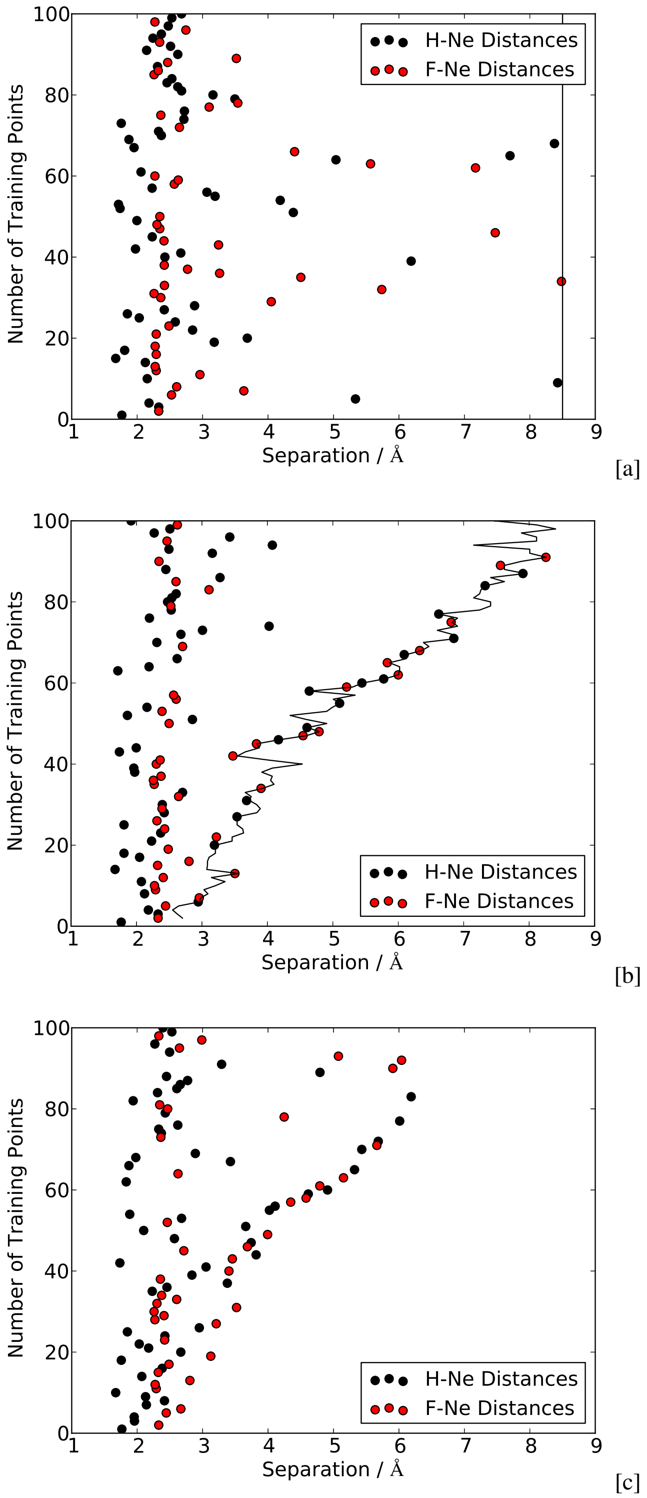


FIG. 6: Plots showing training point placement for the first 100 training points for models of the HF-Ne PES trained using the fixed boundary (a), single-constrained placement (b) and closest model (c) strategies. These points are coloured based on the shortest interatomic distance in the configuration, with only this shortest distance shown for each. Boundary values are represented by black lines where applicable.

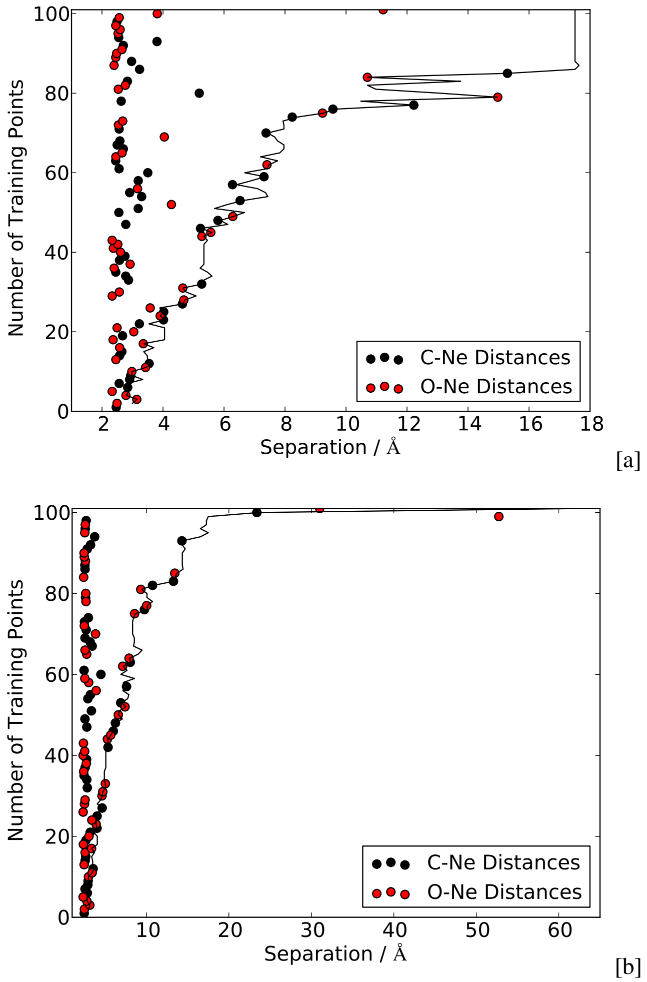


FIG. 7: Plots showing training point placement for the first 100 training points for models of the CO-Ne PES trained using the single-constrained placement (a) and single-open placement (b) strategies. These points are coloured based on the shortest interatomic distance in the configuration, with only this shortest distance shown for each. Boundary values are represented by black lines.

figurations at separations around 8.5 Å when the number of training points is low. This produces more efficient model development under boundary-optimised and closest model training because point placement can be focused on the short-range region of the PES, where energy varies more rapidly with configuration.

Training point plots for the CO-Ne potential, given in figure 7, show that the single-open strategy extends A_{GP} much further than single-constrained training. This suggests that the capacity to place points in A_{LR} facilitates faster expansion of the boundary for this system. However, these plots also indicate that this discrepancy is most noticeable when the number of training points is large; specifically, rapid expansion of the boundary under single-open training was instigated by placement of a single training point at long range, which facilitated

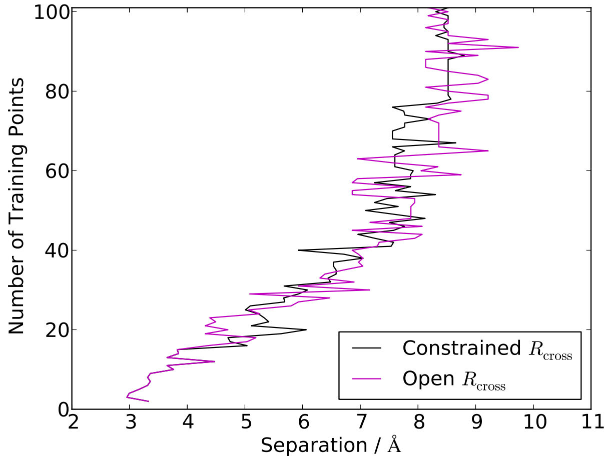


FIG. 8: Plot showing the values of R_{cross} achieved for the first 100 training points by single-open and single-constrained training for models of the CO_2-Ne potential.

the increase in R_{cross} to 63.0 Å. Prior to this, the cross-over value from single-open training was quite similar to that from single-constrained training (both were $\sim$ 17 Å). Hence the two training strategies differ significantly only when the predictions from GP regression are already highly accurate, as at this stage the predictions of the long-range function are poor enough by comparison to merit placement of a training point at long-range. The other systems also show close similarity between the cross-over distance for the single-constrained and single-open strategies, with equivalent plots to figure 8 found in the supplementary material. Also, Figure 8 shows that for the CO_2-Ne potential the values of R_{cross} from the two strategies are similar throughout training. Such a trend suggests that the choice of point placement strategy does not significantly change the value of R_{cross}, implying once more that the choice of classifier is of greater import.

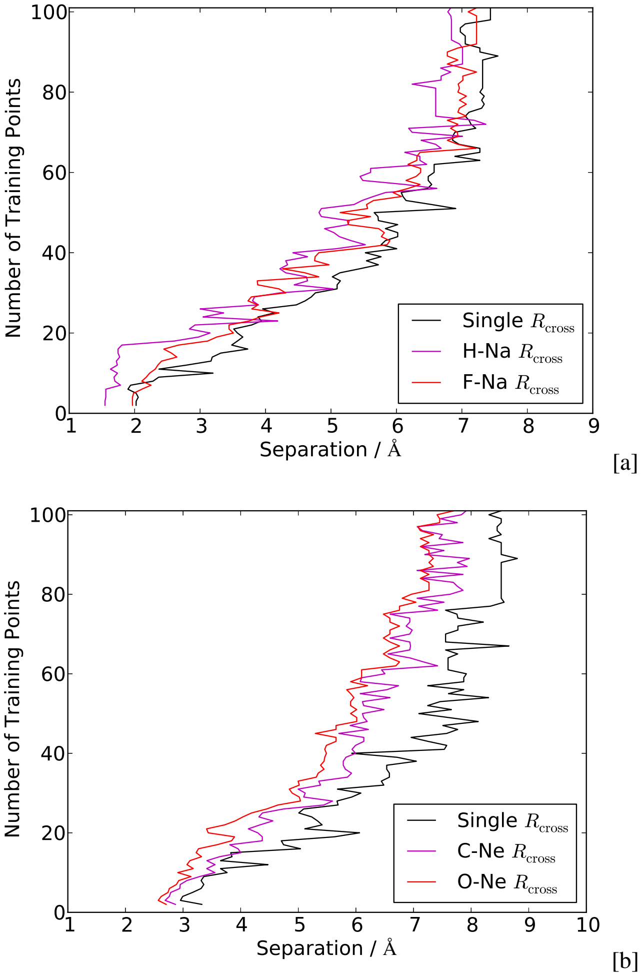


FIG. 9: Plots showing the value of R_{cross} and entries in $\mathbf{R}_{cross}$ for the first 100 training points for models of the HF-Na^+ potential from single/multi-open training (a) and the CO_2-Ne potential from single/multi-constrained training (b).

Similarities in the evolution of the boundary are seen when the cross-over distances achieved under C_{multi} and C_{single} are compared for a given point placement method. Figure 9 shows the results of such a comparison for the HF-Na^+ and CO_2-Ne systems. The cross-over distances generally grow at a similar rate, but the value of R_{cross} is consistently larger than all values in $\mathbf{R}_{cross}$. This suggests that there are differences between C_{single} and C_{multi}, which manifest in both the training efficiency and boundary placement. The larger GP region under C_{single} compared with C_{multi} is explained by noting that both are approximations of an 'ideal' classifier. When N_{TP} is low (*i.e.* one or two), such a classifier will attempt to make A_{GP} as small as possible because the training set comprises configurations from the repulsive wall only. Consequently, given the same training set, C_{single} and C_{multi} also try to minimise the size of A_{GP}. As C_{multi} is more flexible its approximation of the 'ideal' classifier will be closer than that of C_{single}, meaning A_{GP} under C_{single} will be initially larger. Other than the repulsive wall, the largest errors tend to occur at separations around the boundary. Thus, due to its larger A_{GP}, a C_{single} strategy will place its first long-range training point at a larger separation than a C_{multi} strategy. This facilitates faster expansion of the boundary under C_{single} than C_{multi} regardless of which point placement method is used.

Figure 9a shows that $R_{H\text{-}Na^+} < R_{F\text{-}Na^+}$ throughout most of training for the HF-Na^+ potential, meaning that the interaction involving the larger atom (F) obeys a larger cross-over distance. Such ordering of the cross-over distances also applies to the HF-Ne system for most of training. Moreover, figure 9b shows that the ordering of the cross-over distances for the CO_2-Ne potential is consistent, with $R_{O\text{-}Ne} < R_{C\text{-}Ne}$ throughout training. In fact, for the $(CO_2)_2$ and CO-Ne potentials it holds generally throughout training that $R_{O\text{-}O} < R_{O\text{-}C} < R_{C\text{-}C}$ and $R_{O\text{-}Ne} < R_{C\text{-}Ne}$ respectively for both point placement methods. This implies that the ordering of the cross-over distances under C_{multi} is consistent between systems as well as being physically reasonable.

Additionally, figures 6-9 show that the cross-over distance increases with the size of the training set. This is expected

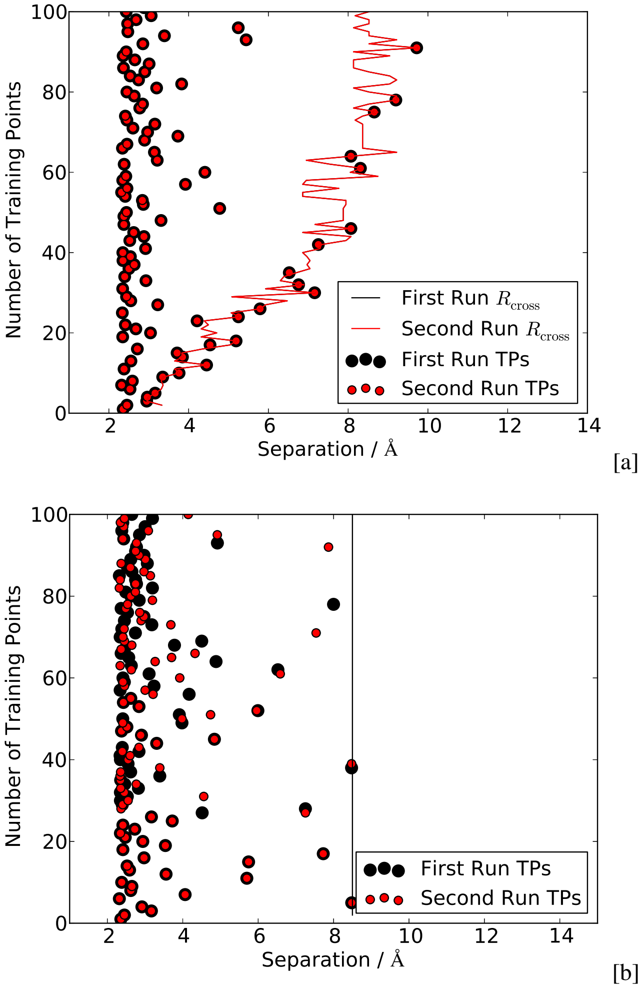


FIG. 10: Plots showing the first 100 training points and values of $R_{\rm cross}$ from two runs of single-open training (a) and fixed boundary training (b) for models of the CO_2-Ne potential.

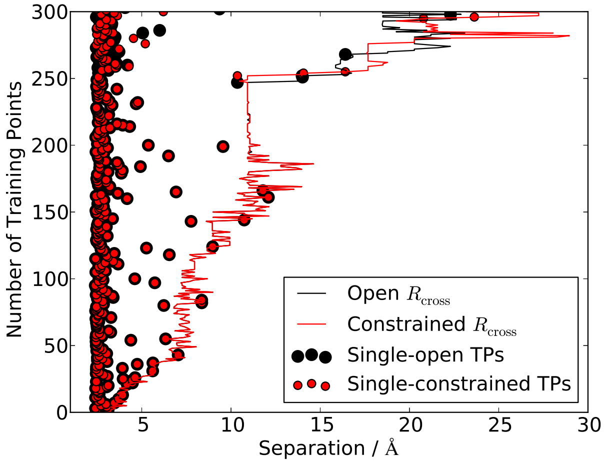


FIG. 11: The first 300 training points and values of $R_{\rm cross}$ from single-open and single-constrained training for the $(CO_2)_2$ potential.

because a larger training set means the GP has more information with which to infer the function describing the PES. The resultant more accurate GP enables larger cross-over distance(s). This is the case for all training strategies across all systems.

Boundary optimisation methods decrease the cost of using PES, by reducing the number of training points and the size of the GP region. Fewer training points means GP evaluations are cheaper, as the GP cost is proportional to $N_{\rm TP}$. Additionally, the lower $R_{\rm cross}$ or $\mathbf{R}_{\rm cross}$ of the boundary-optimised model means that in any application the GP would be used less often, in favour of the much cheaper asymptotic function. Hence boundary optimisation produces models that are more efficient to implement in applications than fixed boundary training, with no reduction in overall accuracy.

Boundary optimisation also increases the reproducibility of training. That is, for two separate models from the same training strategy and reference set, the results will be more similar for a boundary-optimised strategy compared to using a fixed boundary. For example, when modelling the CO_2-Ne potential with single-open training the results from two separate training runs were identical. This is shown in figure 10. This is noteworthy because the values of the hyperparameters selected when maximising $\log(\mathscr{L})$ can vary slightly, even for the same training set. Such variations can alter the predictions of the GP, leading to different values for $R_{\rm cross}$ or $\mathbf{R}_{\rm cross}$ and the selection of different training points. Thus, that two separate runs of the same training strategy are totally identical is encouraging.

Furthermore, figure 10 shows that for fixed boundary training separate runs were identical only up to $N_{\rm TP}$ = 27. While the exact reproducibility in figure 10a is not present for the other potentials, there is generally significantly less difference between independent runs of the boundary-optimised training methods than for fixed boundary training. For example, models of the HF-Ne potential from single-constrained and single-open training are reproducible up to $N_{\rm TP}$ = 30 and $N_{\rm TP}$ = 41 respectively, compared with $N_{\rm TP}$ = 16 for fixed boundary training. Equivalent plots to figure 10 for this system under these strategies are given in the supplementary material. This reproducibility increase compared to fixed boundary training does not transfer to the use of the $C_{\rm multi}$ strategies, likely because a direct search is not as reproducible in multiple dimensions as in a single dimension. However, from the R^2 values in table IV it can be seen that use of either $C_{\rm multi}$ or $C_{\rm single}$ reduces the scatter in the $\mathrm{RMSE}_{\rm test}$ data relative to use of $C_{\rm fixed}$, as evidenced by the larger R^2 values achieved when training with the former two. This trend is repeated across all other potentials examined here, which suggests that use of boundary optimisation leads to more stability in selection of the hyperparameters and hence more consistent predictions.

For the $(CO_2)_2$ system, reproducibility is seen not just for repeat runs of identical training strategies but between the training strategies that use $C_{\rm single}$. This is illustrated in figure

11, which shows that the single-constrained and single-open training strategies choose identical training points until $N_{\rm TP}$ = 210. Such an observation explains why the two methods have identical E in figure 4e. This is because when $N_{\rm TP}$ = 210 the RMSE$_{\rm test}$ values for single-open and single-constrained placement were 2.817 x 10^{-7} $E_{\rm h}$ and 2.784 x 10^{-7} $E_{\rm h}$ respectively. Consequently the error was too low at this $N_{\rm TP}$ to be included in the fit, meaning the two strategies were identical over the RMSE$_{\rm test}$ values used in fitting.

IV. CONCLUSION

It has been shown that boundary optimisation produces GP PESs of the same accuracy using fewer training points than fixing $R_{\rm cross}$ *a priori*. This improvement in efficiency is hierarchical, with a boundary defined by a single, variable cross-over distance offering a modest improvement and a boundary defined by multiple such distances facilitating a further gain.

The results presented imply that the classifier is more important to the training strategy than the point placement method. In the RMSE range that is suitable for first principles calculations ($\sim$2 x 10^{-5} $E_{\rm h}$) the boundary optimisation methods typically reduce the required number of training points by 15-33% relative to a training strategy that is already established as efficient[19]. Because of their reduced training set size, the resulting boundary-optimised PESs are strong candidates for transfer learning, in which the existing *ab initio* calculations are upgraded to a higher level of theory. Furthermore, as the size of the GP region increases with the size of the training set, only as needed, the resulting GPs are also less computationally intensive in applications than fixed boundary methods, as they employ the GP over a smaller region of phase space.

The classifier C$_{\rm multi}$, which uses different cross-over distances for difference atomic pairs, performed almost as well as an 'ideal' classifier. Across all systems, the largest difference in performance between the closest model strategy, which uses an 'ideal' classifier, and nearest boundary-optimised strategy was $\sim$ 3 %. In all cases the best-performing boundary-optimised strategy employed C$_{\rm multi}$, implying that a classifier comprising of a spherical boundary with a unique radius on each unique atomic pair, captures the true boundary effectively. Further refinement of the classifier would not result in a sufficient reduction in training points to justify the extra classifier expense.

The cross-over distance(s) are learned from the reference data under boundary optimisation using a direct search. This is sufficiently fast to be used at every stage of training whether one or many cross-over distances are employed and in the multi-dimensional case returns cross-over distances in a physically reasonable order. Moreover, both direct search algorithms can be used easily in conjunction with another machine learning technique or on another chemical interaction. In fact, a direct search can be applied to any problem whereby a boundary is sought between a good method of approximation in one region of phase space (the long-range function here) and a machine learning technique in another. This means that the methodology which underpins boundary optimisation is both fast and flexible, in addition to being effective in solving the problem of reducing the computational expense associated with training a GP model of an intermolecular PES.

Physical systems in which the behaviour crosses over from a simple asymptotic function to more complicated behaviour are common in many fields. A prominent example is the transition from ideal to non-ideal gas behaviour. As the boundary optimization techniques herein exploit this cross-over in behaviour, there are many potential applications of this technique to physical problems beyond intermolecular potentials.

V. SUPPLEMENTARY MATERIAL

See supplementary material for more plots of results from the explored systems.

VI. ACKNOWLEDGEMENTS

The computations of all reference and test sets described in this paper were performed using the University of Nottingham's Augusta HPC service, which provides a High Performance Computing service to the University's research community. See https://www.nottingham.ac.uk/it-services/research/hpc/ for more details. The authors are also thankful for funding provided by the Leverhulme Trust Doctoral Scholarship through Modelling and Analytics for a Sustainable Society (MASS) at the University of Nottingham.

VII. DATA AVAILABILITY

All test and reference data sets used in this work are available in the Boundary Optimisation Data repository at https://doi.org/10.6084/m9.figshare.16610713.v1, reference number [74].

VIII. AUTHOR DECLARATION

The authors have no conflicts to disclose.

Appendix A: Derivation of an Empirical Long-range Function

HF-Na^{+} was the only intermolecular potential examined above for which a multipolar long-range function was not used. Instead a fitted, empirical long-range function was employed. Although determining an optimal long-range function was not the goal of this work, one that was accurate enough that all configurations were not rapidly transferred to the GP region was needed for proof of concept. The multipolar long-range function was unsuitable for this because of the discrepancy between its predictions and the MP2 energies.

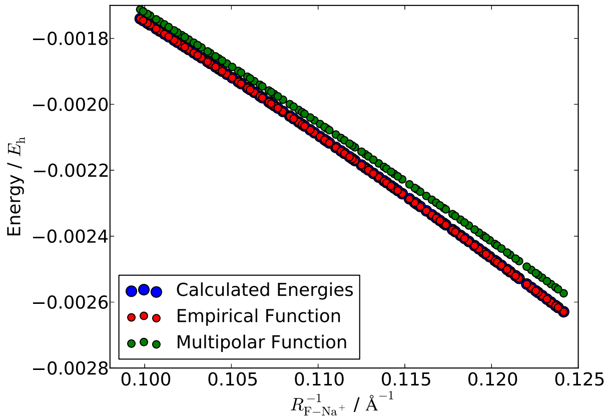


FIG. 12: Plots of the predicted energies of the long-range functions derived by the empirical method (red) and the multipole method (green) for the HF-Na$^+$ potential in a linear configuration with the F proximal to the Na$^+$ (*i.e.* with cos(θ) = 1). Also shown for comparison are MP2 energies (blue) for this configuration.

Evidence of this is given in figure 12, which also highlights the superiority of the predictions of the empirical long-range function. The RMSEs of each method of prediction were separated by two orders of magnitude, with the empirical function achieving an RMSE of 3.07 x 10^{-7} E_h versus 4.15 x 10^{-5} E_h for the multipolar function against the MP2 data shown in the figure.

Figure 12 evidences that the multipolar long-range function captured the power law of the interaction energy for this system in this configuration, with the source of its error being that it was offset from the calculated energies. This offset was a product of the energies being calculated using MP2 while many of the properties used to derive the multipolar long-range function were calculated at higher levels of theory (see table II). In the other systems, the small magnitude of the long-range energies meant that this disagreement was negligible and the multipolar long-range function was usable. However, in the case of HF-Na$^+$, long-range energies with larger magnitudes were commonplace due to the strong repulsive and attractive interactions between the H-F dipole and the Na$^+$ cation. This exacerbated the difference between the predictions of the multipolar long-range function and the MP2 energies.

To approximate accurately the long-range data without upgrading the reference and test data to a higher level of theory, a long-range function was produced by fitting directly to these data. This was the empirical long-range function. Taking r to be the distance between the centre of the H-F bond (not centre of mass) and the Na$^+$, this function estimated the energy, E, as a sum of power laws,

$$E = \mathrm{A}r^{-2} + \mathrm{B}r^{-3}. \tag{A1}$$

In doing so, the empirical long-range function exploits that the dominant powers of r in the HF-Na$^+$ interaction are known to be -2 and -3 but assumes that the coefficients of these terms are unknown.

As the energy varies differently with r when θ changes, a sum such as in equation A1 must be fitted for every configuration in a given data set for which θ is unique. For a given θ the coefficients in equation A1 can be found using simultaneous equations, which are set up by following algorithm 4. In all cases, $r_{\min}$ = 8.5 Å and $r_{\max}$ = 10.5 Å, though the power laws which resulted were accurate up to 100 Å. Furthermore, both GPs in the algorithm were trained on ∼ 300 training points from a LHC design.

Algorithm 4 Coefficients for the Empirical Long-range Function

```
1: Train a GP, GP$_{\min}$, on a range of $\theta$ values at separation $r = r_{\min}$;
   do the same for another GP, GP$_{\max}$, at $r = r_{\max}$.
2: For a given $\theta$, predict $E_{\min}$ and $E_{\max}$ using GP$_{\min}$ and GP$_{\max}$
   respectively.
3: Set up simultaneous equations of the form shown in equation
   A1: one with $E = E_{\min}$ and $r = r_{\min}$, and another with $E = E_{\max}$
   and $r = r_{\max}$.
4: Solve the equations from step three for the coefficients for the
   current $\theta$.
```

By repeating steps 2-4 in algorithm 4, a sum of power laws was determined for every θ value in the reference set and test sets. Fitting to the latter was possible as knowledge of the energies in the set over which fitting was undertaken was unnecessary. This was because the GP predictions were based on their respective training sets, and r and θ were found from the inverse interatomic separations at each configuration alone.

Deriving a long-range function from the predictions of two GPs could be problematic if a transfer learning approach were to be invoked as the data used to train these GPs would itself need to be upgraded. For the HF-Na$^+$ potential this would not be an issue because increasing the quality of the training data would increase the quality of the fit from the multipolar long-range function.

However, when an empirical function is the only option for modelling the long-range data, upgrading the data used in training GP$_{\min}$ and GP$_{\max}$ would be of considerable computational expense. This is because, currently, each comprise ∼ 300 configurations. Such an issue could be circumvented by using a sequential design strategy to build minimal training sets for these GPs, which could then be upgraded instead. Furthermore, the training sets used for GP$_{\min}$ and GP$_{\max}$ are independent of that used to train a GP on the wider PES. Letting the number of training points in the training sets of GP$_{\min}$ and GP$_{\max}$ be Y and that in the training set of the other GP be Z, the cost of short-range predictions in any simulation would scale linearly with Z and the cost of any long-range predictions would scale linearly with 2Y. Given that the training of GP$_{\min}$ and GP$_{\max}$ take place at a fixed separations, it is likely that 2Y < Z if all GPs were trained under a sequential design strategy. This means that the predictions of the empirical long-range function would not be the computational bottleneck and that such a function is suitable for use in simulations. Finally, the method is sufficiently flexible that the GP predictions can easily be replaced by those of another statistical method.

[1] T. A. Halgren and W. Damm, "Polarizable force fields," Curr. Opin. Struct. Biol. **11**, 236–242 (2001).

[2] A. Warshel, M. Kato, and A. V. Pisliakov, "Polarizable force fields: history, test cases, and prospects," Journal of Chemical Theory and Computation **3**, 2034–2045 (2007).

[3] P. Xu, E. B. Guidez, C. Bertoni, and M. S. Gordon, "Perspective: Ab initio force field methods derived from quantum mechanics," J. Chem. Phys. **148**, 090901 (2018).

[4] J. Behler, "Perspective: Machine learning potentials for atomistic simulations," J. Chem. Phys **145**, 170901 (2016).

[5] T. Mueller, A. Hernandez, and C. Wang, "Machine learning for interatomic potential models," J. Chem. Phys. **152**, 050902 (2020).

[6] B. Selvaratnam and R. T. Koodali, "Machine Learning in Experimental Materials Chemistry," Catal. Today (2020).

[7] J. Noh, G. H. Gu, S. Kim, and Y. Jung, "Machine-enabled inverse design of inorganic solid materials: promises and challenges," Chem. Sci. **11**, 4871–4881 (2020).

[8] P. O. Dral, "Quantum chemistry in the age of machine learning," J. Phys. Chem. Lett. **11**, 2336–2347 (2020).

[9] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).

[10] N. Artrith, T. Morawietz, and J. Behler, "High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide," Phys. Rev. B **83**, 153101 (2011).

[11] N. Artrith, B. Hiller, and J. Behler, "Neural network potentials for metals and oxides - First applications to copper clusters at zinc oxide," Phys. Status Solidi B **250**, 1191–1203 (2013).

[12] J. Behler, "Representing potential energy surfaces by high-dimensional neural network potentials," J. Phys. Condens. Matt. **26**, 183001 (2014).

[13] Q. Lin, Y. Zhang, B. Zhao, and B. Jiang, "Automatically growing global reactive neural network potential energy surfaces: a trajectory free active learning strategy," J. Chem. Phys. (2020).

[14] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," MMS **14**, 1153–1173 (2016).

[15] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," Comput. Mater. Sci. **140**, 171–180 (2017).

[16] I. I. Novoselov, A. V. Yanilkin, A. V. Shapeev, and E. V. Podryabinkin, "Moment tensor potentials as a promising tool to study diffusion processes," Comput. Mater. Sci. **164**, 46–56 (2019).

[17] E. Uteva, R. J. Wheatley, R. D. Wilkinson, and R. S. Graham, "Interpolation of intermolecular potentials using Gaussian processes," J. Chem. Phys. **147**, 161706 (2017).

[18] A. J. Cresswell, R. J. Wheatley, R. D. Wilkinson, and R. S. Graham, "Molecular simulation of the thermophysical properties and phase behaviour of impure CO_2 relevant to CCS," Faraday Discussions **192**, 415–436 (2016).

[19] E. Uteva, R. J. Wheatley, R. D. Wilkinson, and R. S. Graham, "Active learning in Gaussian process interpolation of potential energy surfaces," J. Chem. Phys. **149**, 174114 (2018).

[20] C. M. Handley, G. I. Hawe, D. B. Kell, and P. L. A. Popelier, "Optimal construction of a fast and accurate polarisable water potential based on multipole moments trained by machine learning," Phys. Chem. Chem. Phys. **11**, 6365–6376 (2009).

[21] M. J. L. Mills and P. L. A. Popelier, "Intramolecular polarisable multipolar electrostatics from the machine learning method kriging," Comput. Theor. Chem. **975**, 42–51 (2011).

[22] M. J. L. Mills and P. L. A. Popelier, "Polarisable multipolar electrostatics from the machine learning method Kriging: an application to alanine," Theor. Chem. Acc. **131**, 1137 (2012).

[23] S. M. Kandathil, T. L. Fletcher, Y. Yuan, J. Knowles, and P. L. A. Popelier, "Accuracy and tractability of a Kriging model of intramolecular polarizable multipolar electrostatics and its application to histidine," J. Comput. Chem. **34**, 1850–1861 (2013).

[24] J. Dai and R. V. Krems, "Interpolation and extrapolation of global potential energy surfaces for polyatomic systems by Gaussian processes with composite kernels," J. Chem. Theory Comput. **16**, 1386–1395 (2020).

[25] H. Sugisawa, T. Ida, and R. V. Krems, "Gaussian process model of 51-dimensional potential energy surface for protonated imidazole dimer," J. Chem. Phys. **153**, 114101 (2020).

[26] G. Schmitz, E. L. Klinting, and O. Christiansen, "A Gaussian process regression adaptive density guided approach for potential energy surface construction," J. Chem. Phys. **153**, 064105 (2020).

[27] M. J. Burn and P. L. A. Popelier, "Creating Gaussian process regression models for molecular simulations using adaptive sampling," J. Chem. Phys. **153**, 054111 (2020).

[28] M. A. Boussaidi, O. Ren, D. Voytsekhovsky, and N. Manzhos, "Random Sampling High Dimensional Model Representation Gaussian Process Regression (RS-HDMR-GPR) for multivariate function representation: application to molecular potential energy surfaces," J. Phys. Chem. A **124**, 7598–7607 (2020).

[29] G. Sivaraman, A. N. Krishnamoorthy, M. Baur, C. Holm, M. Stan, G. Csányi, C. Benmore, and Á. Vázquez-Mayagoitia, "Machine Learning Inter-Atomic Potentials Generation Driven by Active Learning: A Case Study for Amorphous and Liquid Hafnium dioxide," arXiv preprint arXiv:1910.10254 (2019).

[30] M. A. Caro, G. Csanyi, T. Laurila, and V. L. Deringer, "Machine learning driven simulated deposition of carbon films," Phys. Rev. B. **102**, 21 (2020).

[31] Y. Liu, J. Yang, G. Xin, L. Liu, G. Csányi, and B. Cao, "Machine learning interatomic potential developed for molecular simulations on thermal properties of β-Ga_2O_3," J. Chem. Phys. **153**, 144501 (2020).

[32] A. Glielmo, P. Sollich, and A. De Vita, "Accurate interatomic force fields via machine learning with covariant kernels," Phys. Rev. B **95**, 214302 (2017).

[33] A. Glielmo, C. Zeni, and A. De Vita, "Efficient nonparametric n-body force fields from machine learning," Phys. Rev. B **97**, 184307 (2018).

[34] C. Zeni, K. Rossi, A. Glielmo, Á. Fekete, N. Gaston, F. Baletto, and A. De Vita, "Building machine learning force fields for nanoclusters," J. Chem. Phys. **148**, 241739 (2018).

[35] P. I. Maxwell, N. di Pasquale, S. Cardamone, and P. L. A. Popelier, "The prediction of topologically partitioned intra-atomic and inter-atomic energies by the machine learning method kriging," Theor. Chem. Acc. **135**, 195 (2016).

[36] P. I. Maxwell and P. L. A. Popelier, "Accurate prediction of the energetics of weakly bound complexes using the machine learning method kriging," Struct. Chem. **28**, 1513–1523 (2017).

[37] Z. E. Hughes, E. Ren, J. Thacker, B. Symons, A. F. Silva, and P. L. A. Popelier, "A FFLUX water model: Flexible, Polarizable and with a Multipolar Description of Electrostatics," J. Comput. Chem. **41**, 619–628 (2020).

[38] T. L. Fletcher and P. L. A. Popelier, "FFLUX: Transferability of polarizable machine-learned electrostatics in peptide chains," J. Comput. Chem. **38**, 1005–1014 (2017).

[39] J. C. R. Thacker, A. L. Wilson, Z. E. Hughes, M. J. Burn, P. I. Maxwell, and P. L. A. Popelier, "Towards the simulation of biomolecules: optimisation of peptide-capped glycine using FFLUX," Mol. Simul. **44**, 881–890 (2018).

[40] A. Konovalov, B. Symons, and P. L. A. Popelier, "On the many-body nature of intramolecular forces in fflux and its implications," J. Comput. Chem. **42**, 107–116 (2020).

[41] K. R. Bartók, A P and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).

[42] F. C. Mocanu, K. Konstantinou, T. H. Lee, N. Bernstein, V. L. Deringer, G. Csányi, and S. R. Elliott, "Modeling the phase-change memory material, $Ge_2Sb_2Te_5$, with a machine-learned interatomic potential," J. Phys. Chem. B **122**, 8998–9006 (2018).

[43] D. Dragoni, T. D. Daff, G. Csányi, and N. Marzari, "Achieving DFT accuracy with a machine-learning interatomic potential: Thermomechanics and defects in BCC ferromagnetic iron," Phys. Rev. Mater. **2**, 013808 (2018).

[44] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, "Machine learning a general-purpose interatomic potential for silicon," Phys. Rev. X **8**, 041048 (2018).

[45] V. L. Deringer, M. A. Caro, and G. Csányi, "A general-purpose machine-learning force field for bulk and nanostructured phosphorus," Nat. Commun. **11**, 1–11 (2020).

[46] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," Technometrics **42**, 55–61 (2000).

[47] M. Stein, "Large sample properties of simulations using latin hypercube sampling," Technometrics **29**, 143–151 (1987).

[48] W.-L. Loh *et al.*, "On latin hypercube sampling," Annals of statistics **24**,

2058–2080 (1996).

[49] H. Robbins, "Some aspects of the sequential design of experiments," Bulletin of the American Mathematical Society **58**, 527–535 (1952).

[50] H. Chernoff, "Sequential design of experiments," The Annals of Mathematical Statistics **30**, 755–770 (1959).

[51] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, *et al.*, "Performance and Cost Assessment of Machine Learning Interatomic Potentials," J. Phys. Chem. A **124**, 731–745 (2020).

[52] D. J. MacKay, "Introduction to Gaussian processes," NATO ASI series F computer and systems sciences **168**, 133–166 (1998).

[53] M. N. Gibbs and D. J. MacKay, "Variational Gaussian process classifiers," IEEE Transactions on Neural Networks **11**, 1458–1464 (2000).

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

[55] C. Qu, Q. Yu, B. L. Van Hoozen Jr, J. M. Bowman, and R. A. Vargas-Hernández, "Assessing Gaussian process regression and permutationally invariant polynomial approaches to represent high-dimensional potential energy surfaces," J. Chem. Theory Comput. **14**, 3381–3396 (2018).

[56] J. W. Ng and M. P. Deisenroth, "Hierarchical mixture-of-experts model for large-scale gaussian process regression," arXiv preprint arXiv:1412.3078 (2014).

[57] T. T. Nguyen, E. Székely, G. Imbalzano, J. Behler, G. Csányi, M. Ceriotti, A. W. Götz, and F. Paesani, "Comparison of permutationally invariant polynomials, neural networks, and gaussian approximation potentials in representing water interactions through many-body expansions," J. Chem. Phys. **148**, 241725 (2018).

[58] A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington Jr, and S. Manzhos, "Neural networks vs Gaussian process regression for representing potential energy surfaces: A comparative study of fit quality and vibrational spectrum accuracy," J. Chem. Phys. **148**, 241702 (2018).

[59] R. V. Krems, "Bayesian machine learning for quantum molecular dynamics," Phys. Chem. Chem. Phys. **21**, 13392–13410 (2019).

[60] M. P. Metz and K. Szalewicz, "A statistically guided grid generation method and its application to intermolecular potential energy surfaces," J. Chem. Phys. **152**, 134111 (2020).

[61] J. Cui and R. V. Krems, "Efficient non-parametric fitting of potential energy surfaces for polyatomic molecules with Gaussian processes," J. Phys. B At. Mol. Opt. **49**, 224001 (2016).

[62] A. Stone, *The Theory of Intermolecular Forces* (OUP oxford, 2013).

[63] L. E. Atlas, D. A. Cohn, and R. E. Ladner, "Training connectionist networks with queries and selective sampling," in *Advances in neural information processing systems* (Citeseer, 1990) pp. 566–573.

[64] D. J. MacKay, "Information-based objective functions for active data selection," Neural computation **4**, 590–604 (1992).

[65] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (IGI global, 2010) pp. 242–264.

[66] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering **22**, 1345–1359 (2009).

[67] C. Møller and M. S. Plesset, "Note on an approximation treatment for many-electron systems," Phys. Rev. **46**, 618 (1934).

[68] R. J. Bartlett and M. Musiał, "Coupled-cluster theory in quantum chemistry," RMP **79**, 291 (2007).

[69] H.-J. Werner, P. J. Knowles, G. Knizia, F. R. Manby, and M. Schütz, "Molpro: a general-purpose quantum chemistry program package," Wiley Interdiscip. Rev. Comput. Mol. Sci. **2**, 242–253 (2012).

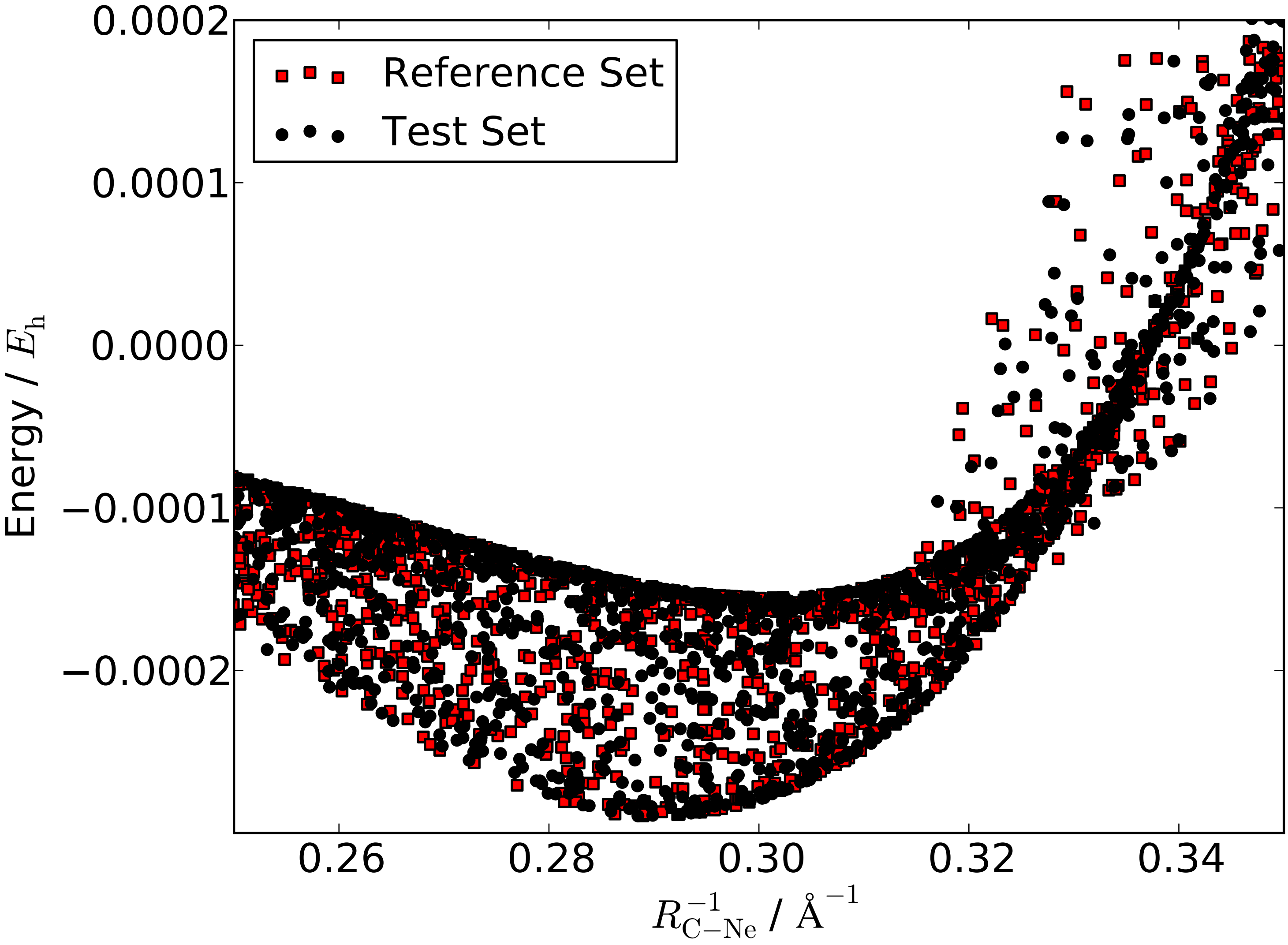
[70] V. Fock, "Näherungsmethode zur lösung des quantenmechanischen mehrkörperproblems," Zeitschrift für Physik **61**, 126–148 (1930).

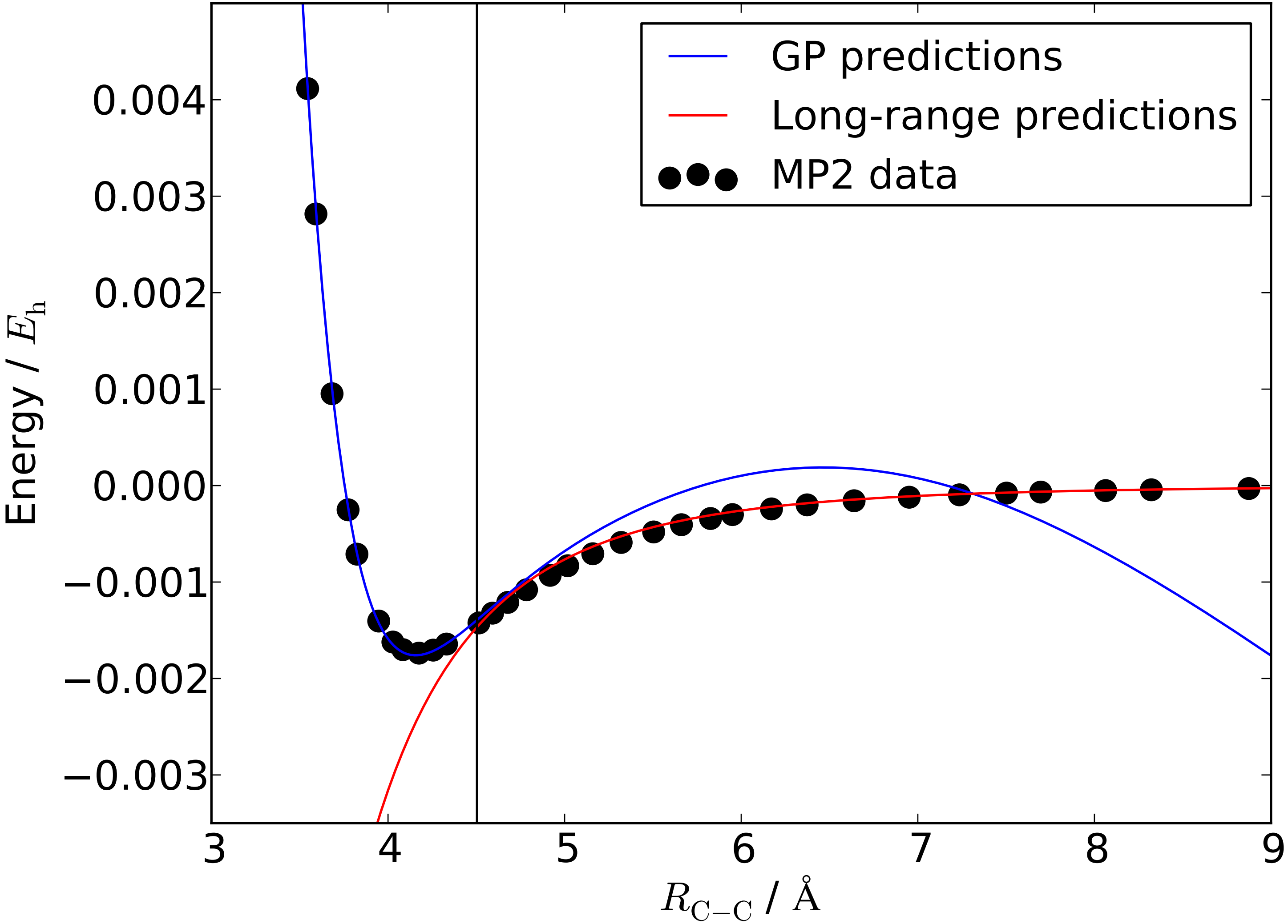
[71] H.-J. Werner and P. J. Knowles, "An efficient internally contracted multiconfiguration–reference configuration interaction method," The Journal of chemical physics **89**, 5803–5814 (1988).

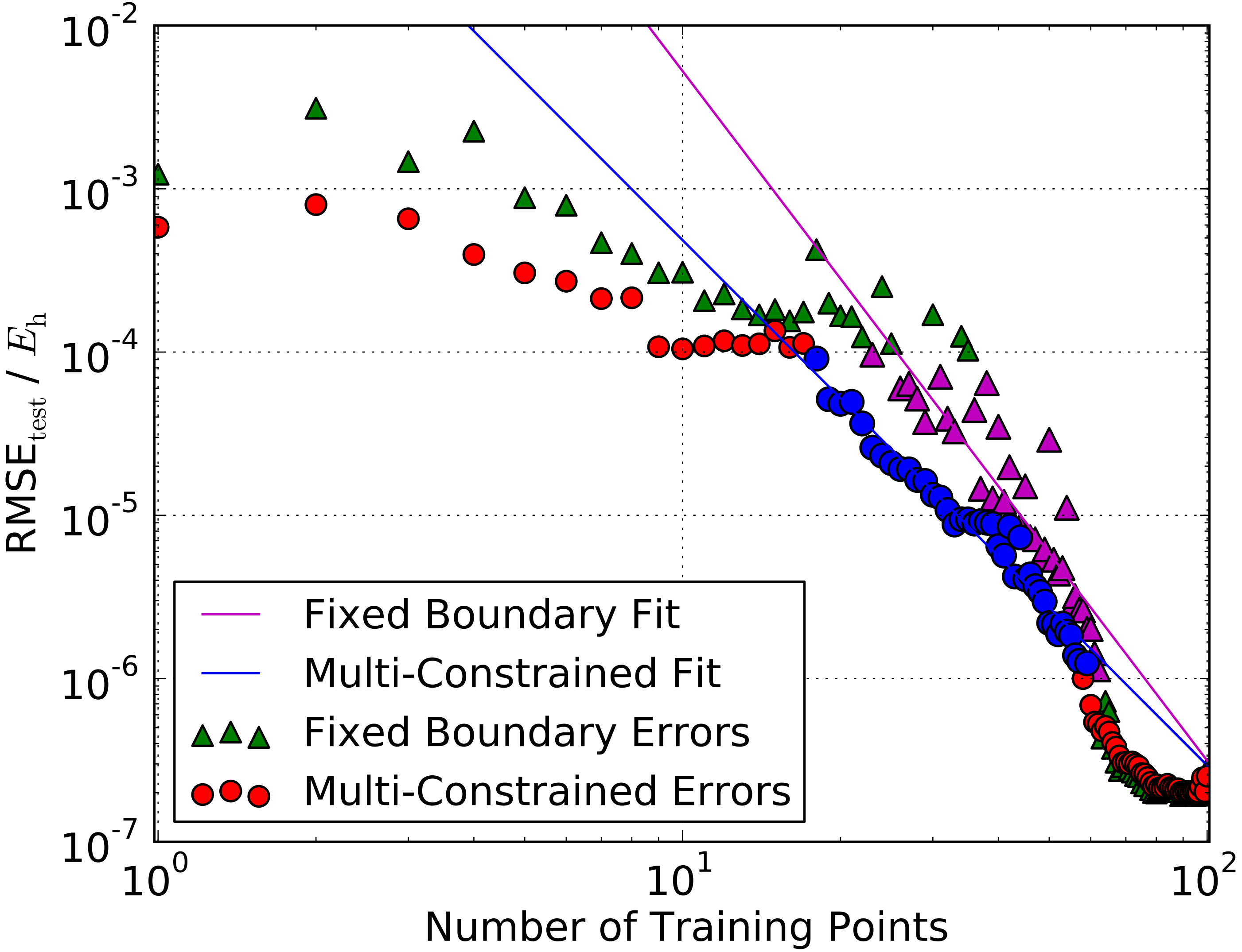
[72] P. J. Knowles and H.-J. Werner, "An efficient method for the evaluation of coupling coefficients in configuration interaction calculations," Chemical physics letters **145**, 514–522 (1988).

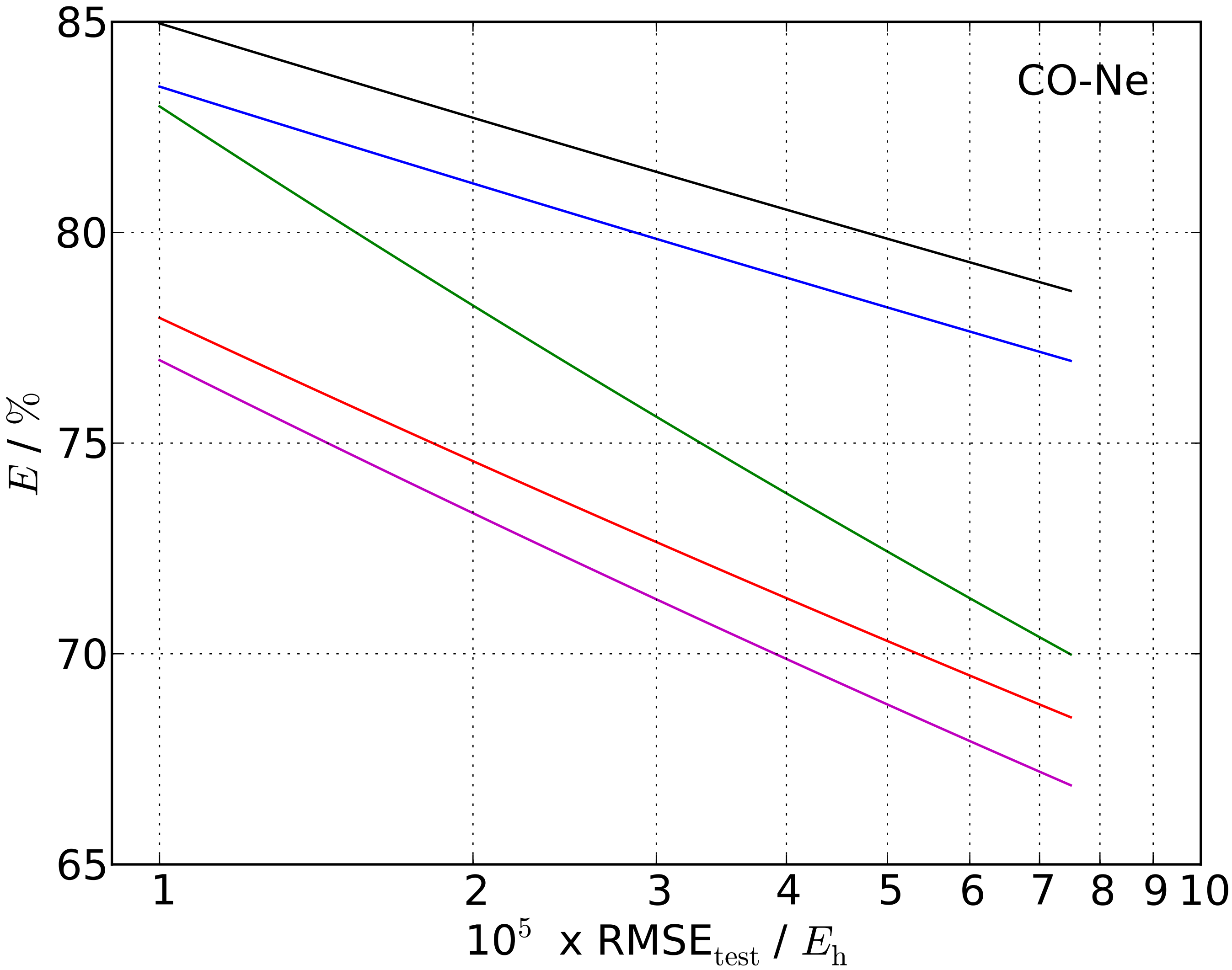
[73] GPy, "GPy: A gaussian process framework in python," `http://github.com/SheffieldML/GPy` (since 2012).

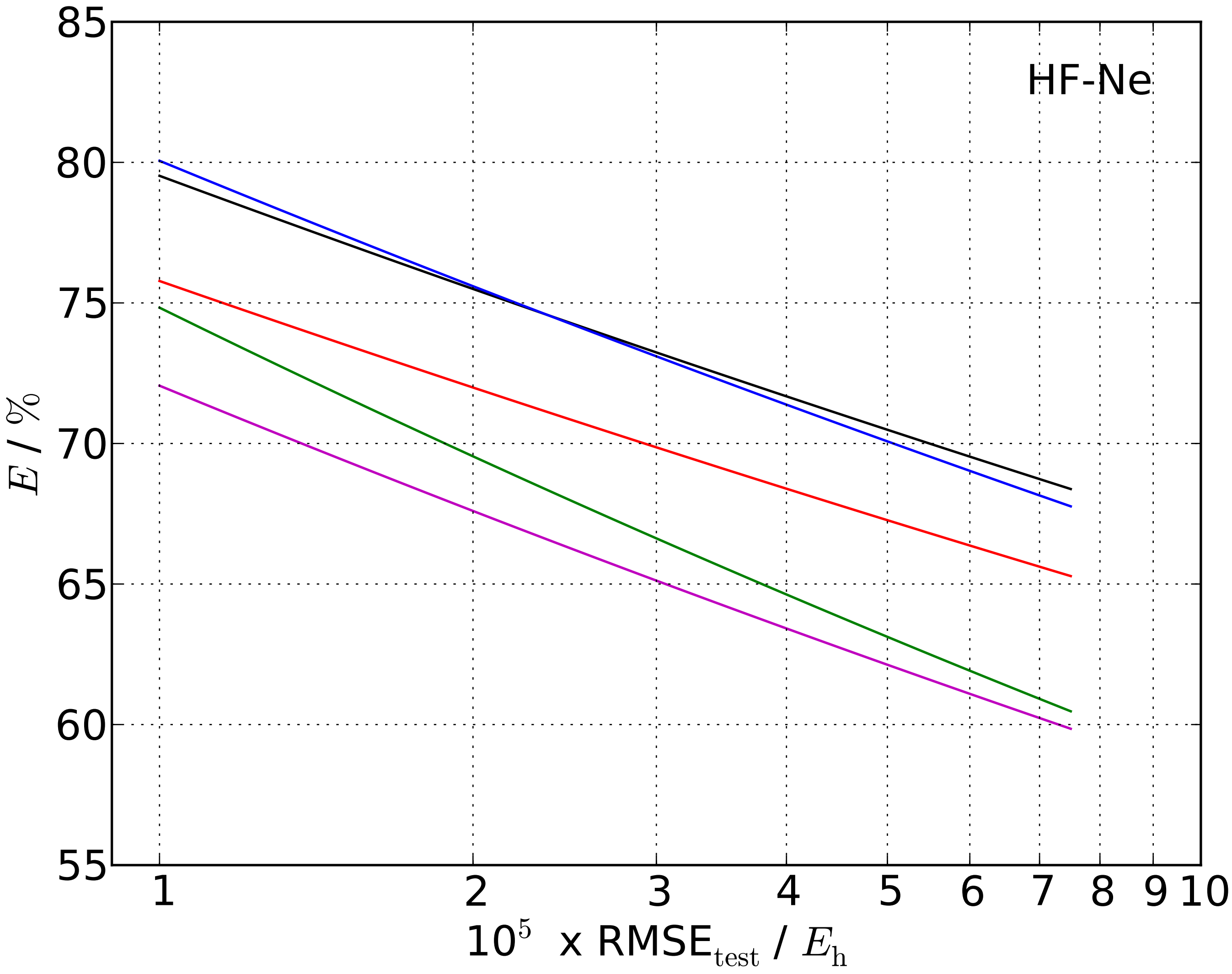
[74] J. Broad, R. J. Wheatley, and R. S. Graham, "Boundary Optimisation Data," `https://doi.org/10.6084/m9.figshare.16610713.v1` (2021).

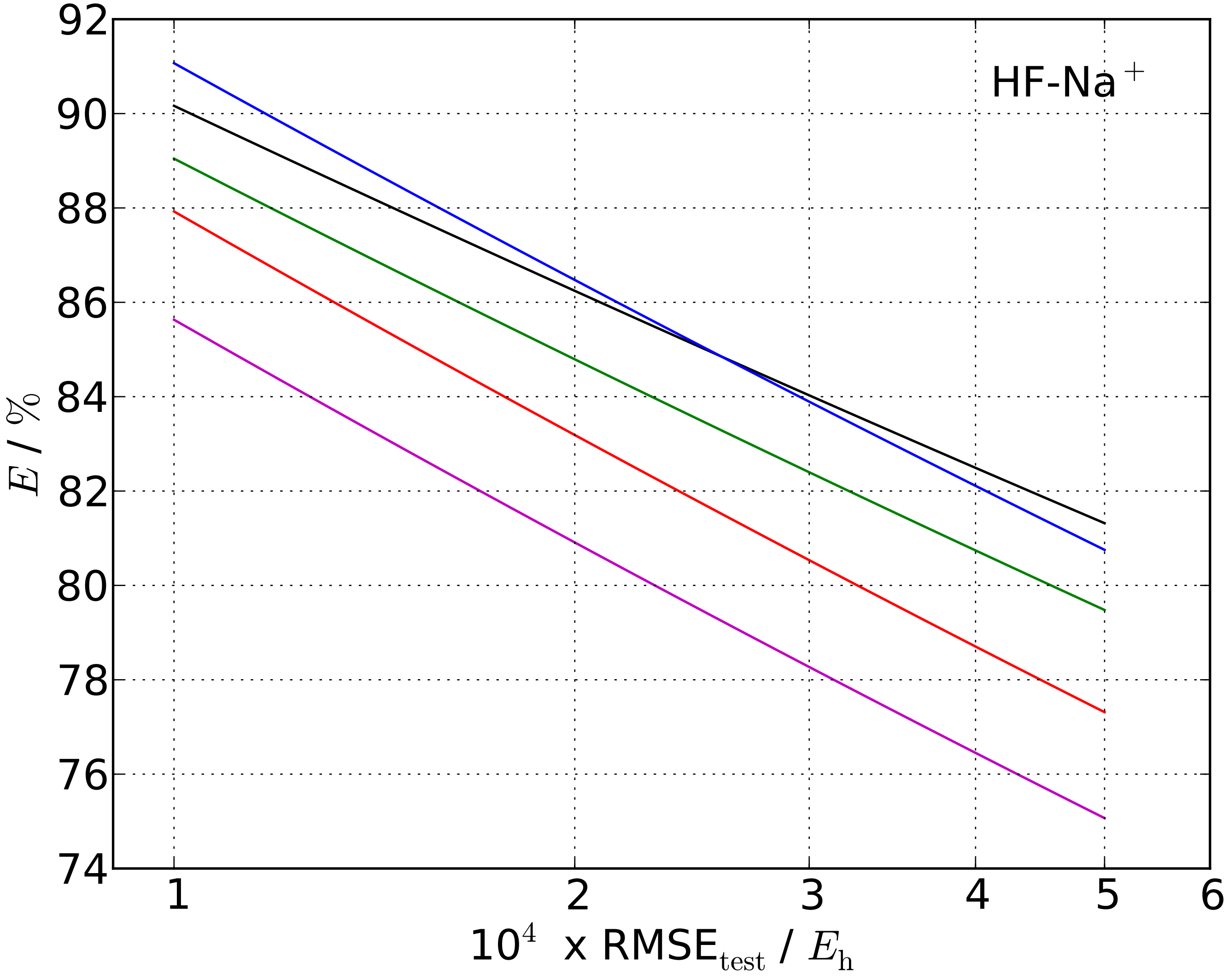

Reference Set
Test Set
Energy / E_h
0.0002
0.0001
0.0000
−0.0001
−0.0002
0.26
0.28
0.30
0.32
0.34
$R_{\mathrm{C-Ne}}^{-1}$ / Å^{-1}

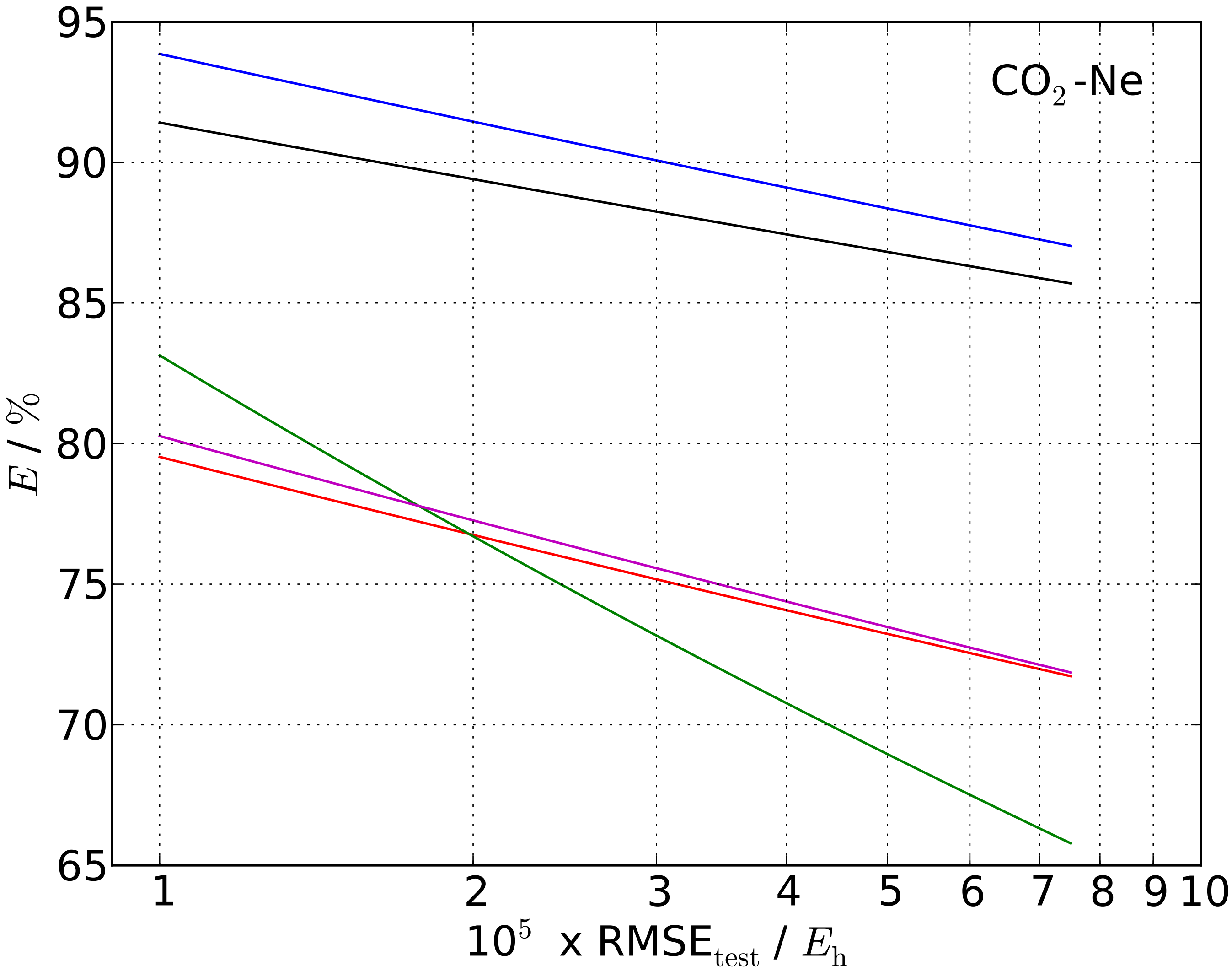

GP predictions
Long-range predictions
MP2 data
Energy / E_h
0.004
0.003
0.002
0.001
0.000
−0.001
−0.002
−0.003
$R_{\mathrm{C-C}}$ / Å
3
4
5
6
7
8
9

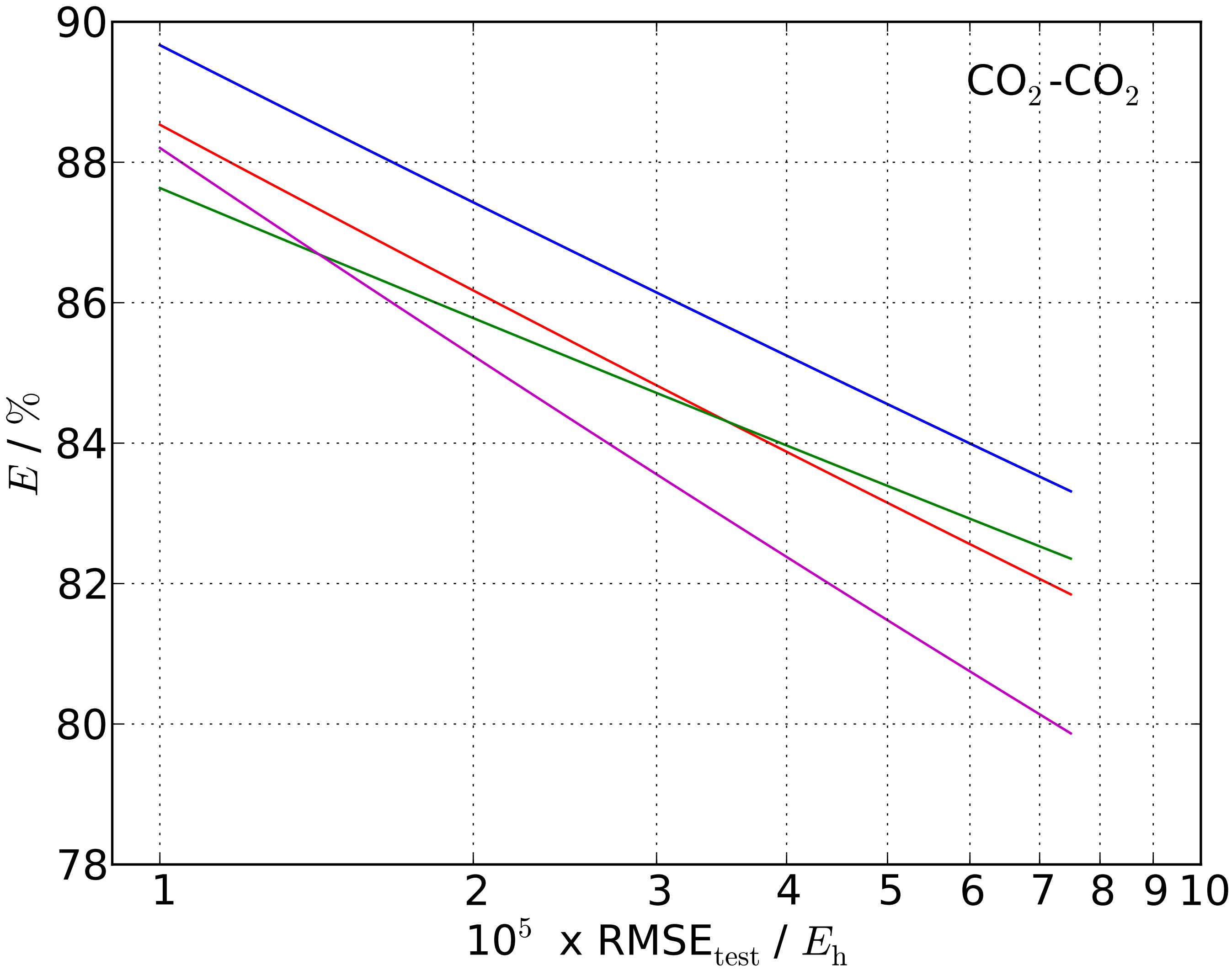

10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
$\mathrm{RMSE}_{\mathrm{test}}$ / E_{h}
10^{0}
10^{1}
10^{2}
Number of Training Points
Fixed Boundary Fit
Multi-Constrained Fit
Fixed Boundary Errors
Multi-Constrained Errors

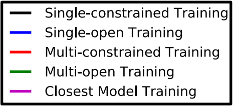

CO-Ne
85
80
75
70
65
E / %
1
2
3
4
5
6
7
8
9
10
10^5 x RMSE$_{\mathrm{test}}$ / E_h

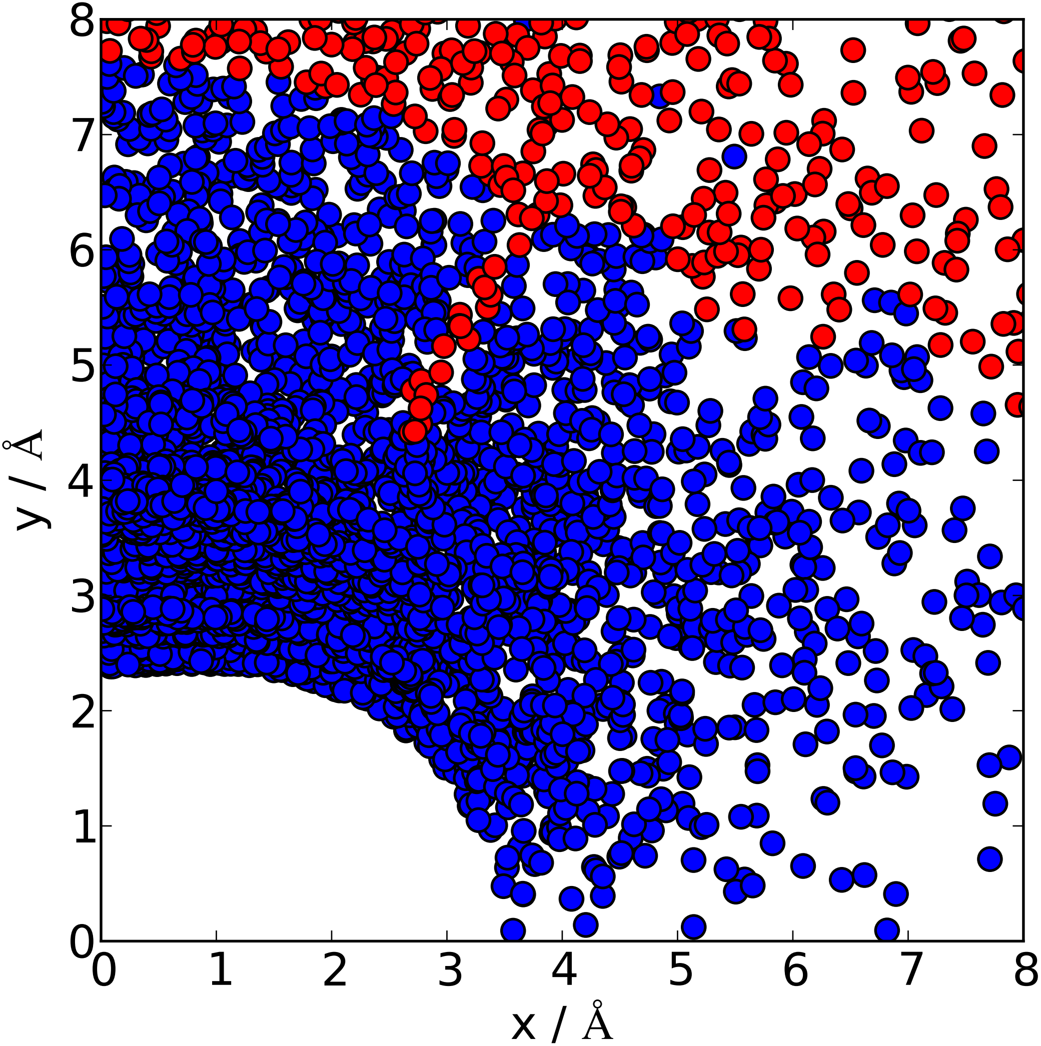

HF-Ne
85
80
75
70
65
60
55
1
2
3
4
5
6
7
8
9
10
E / %
10^5 x RMSE$_{\mathrm{test}}$ / E_{h}

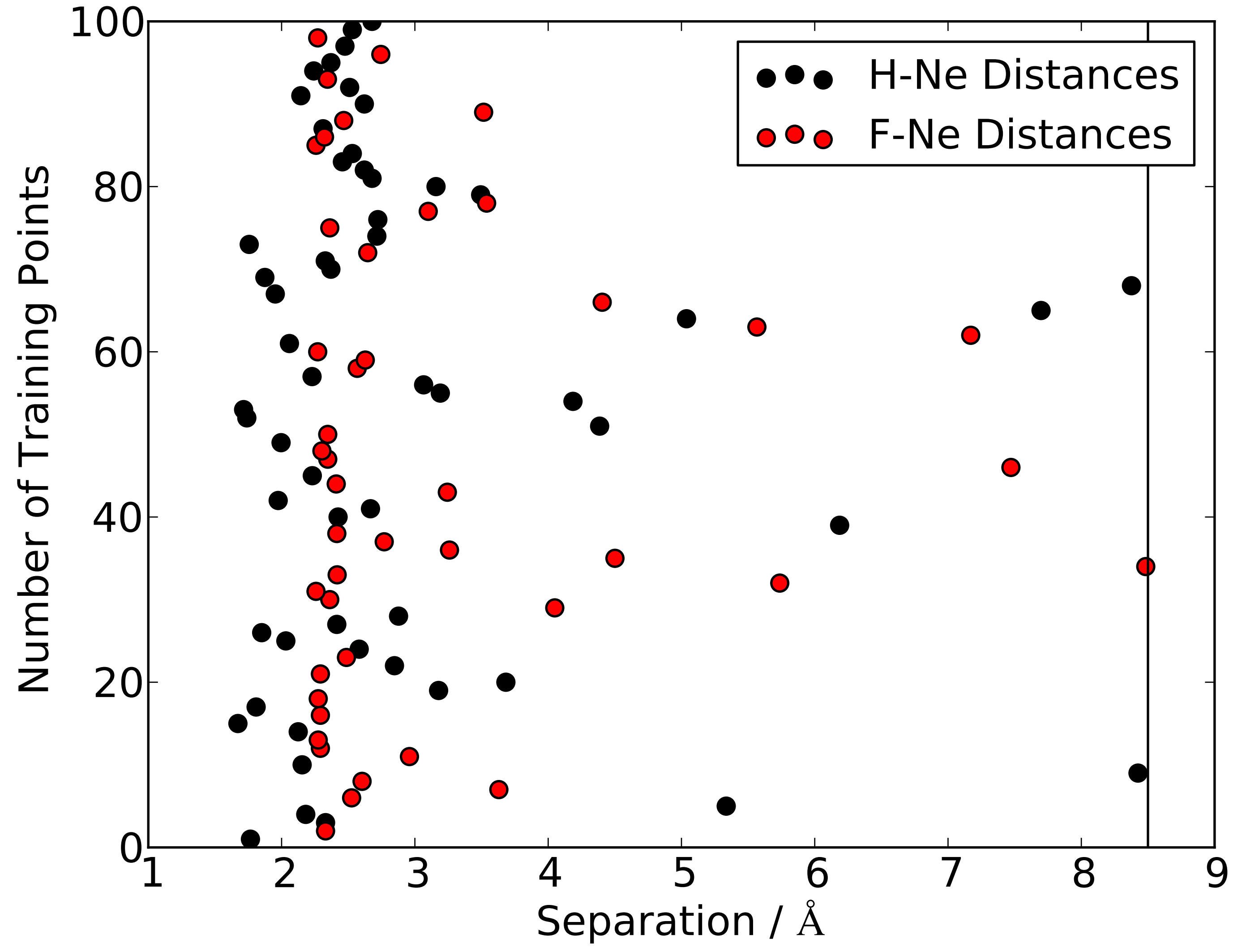

HF-Na$^+$
92
90
88
86
84
82
80
78
76
74
E / %
1
2
3
4
5
6
10^4 x RMSE$_{\mathrm{test}}$ / E_{h}

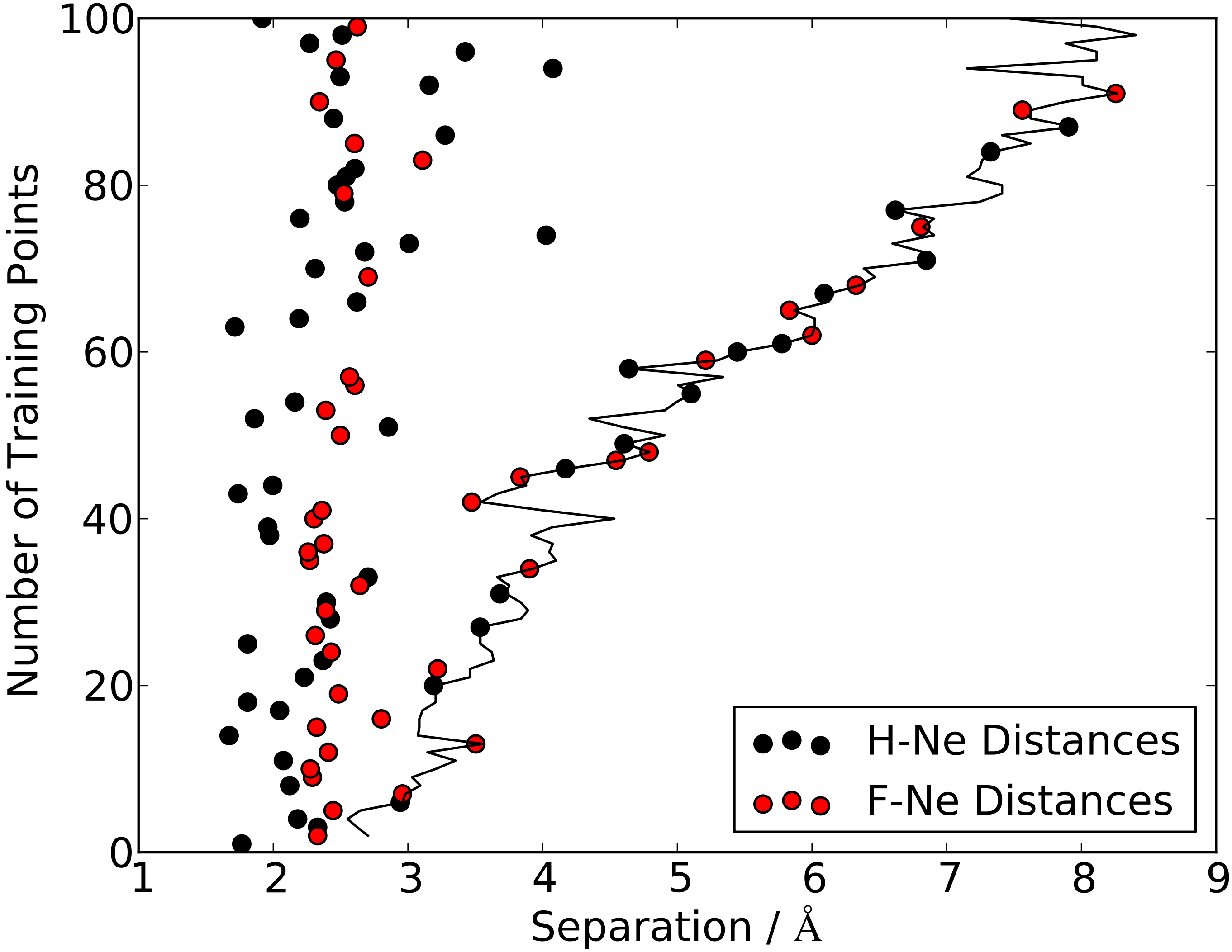

CO$_2$-Ne
95
90
85
80
75
70
65
E / %
1
2
3
4
5
6
7
8
9
10
10^5 x RMSE$_{\mathrm{test}}$ / E_{h}

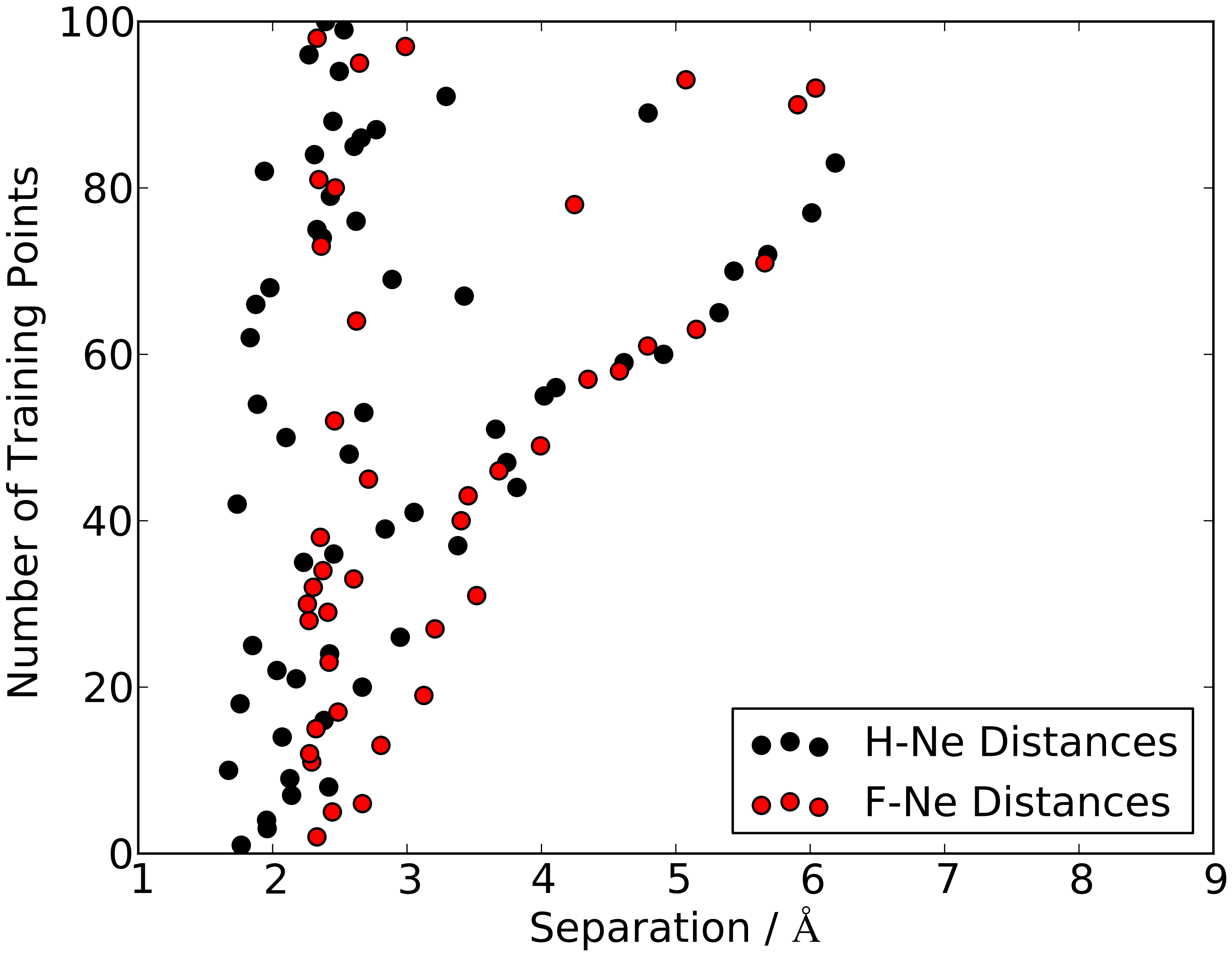

CO_2-CO_2
E / %
90
88
86
84
82
80
78
1
2
3
4
5
6
7
8
9
10
10^5 x RMSE$_{\mathrm{test}}$ / E_{h}


Single-constrained Training
Single-open Training
Multi-constrained Training
Multi-open Training
Closest Model Training

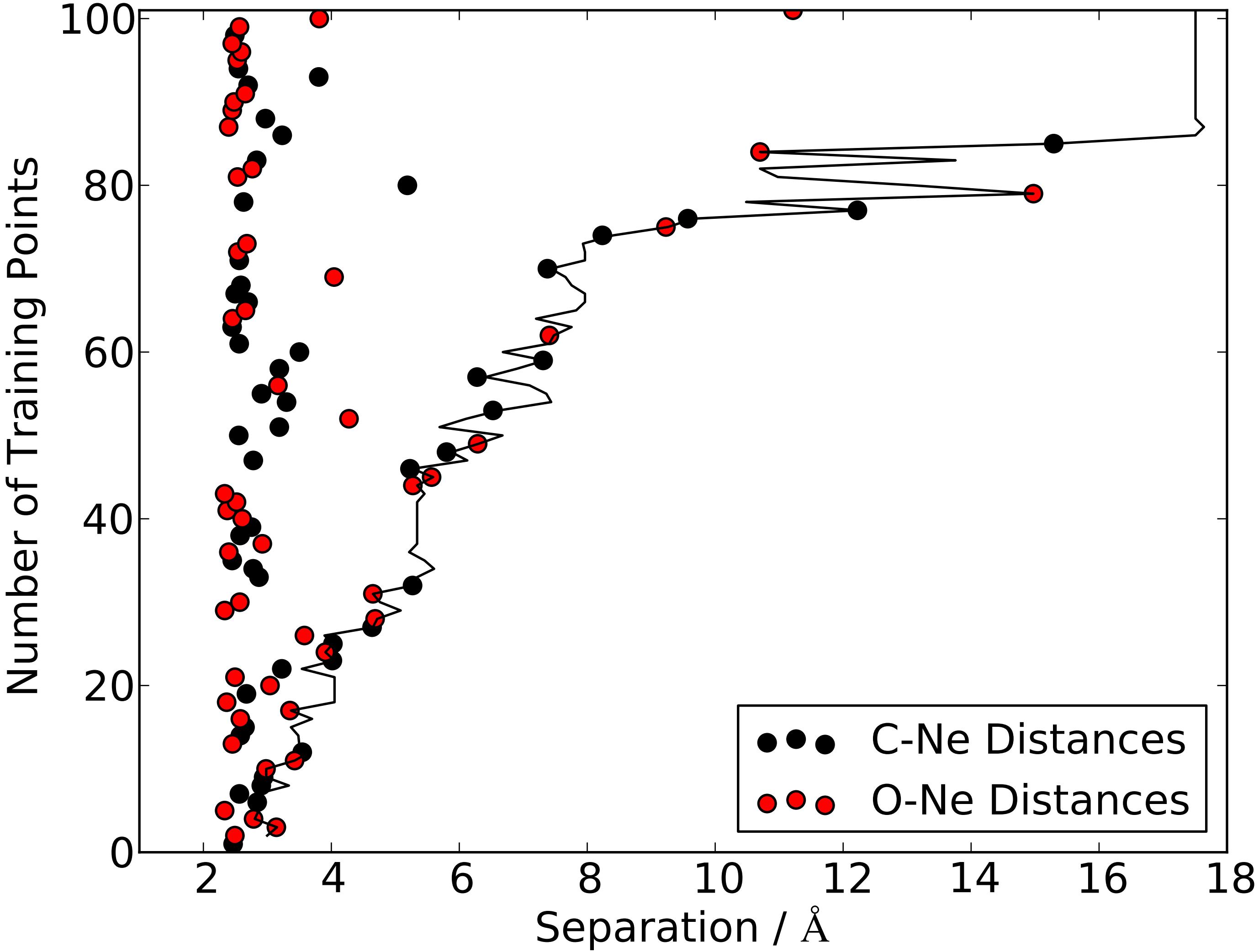

y / Å
x / Å
0
1
2
3
4
5
6
7
8
0
1
2
3
4
5
6
7
8


H-Ne Distances
F-Ne Distances
Number of Training Points
Separation / Å
0
20
40
60
80
100
1
2
3
4
5
6
7
8
9

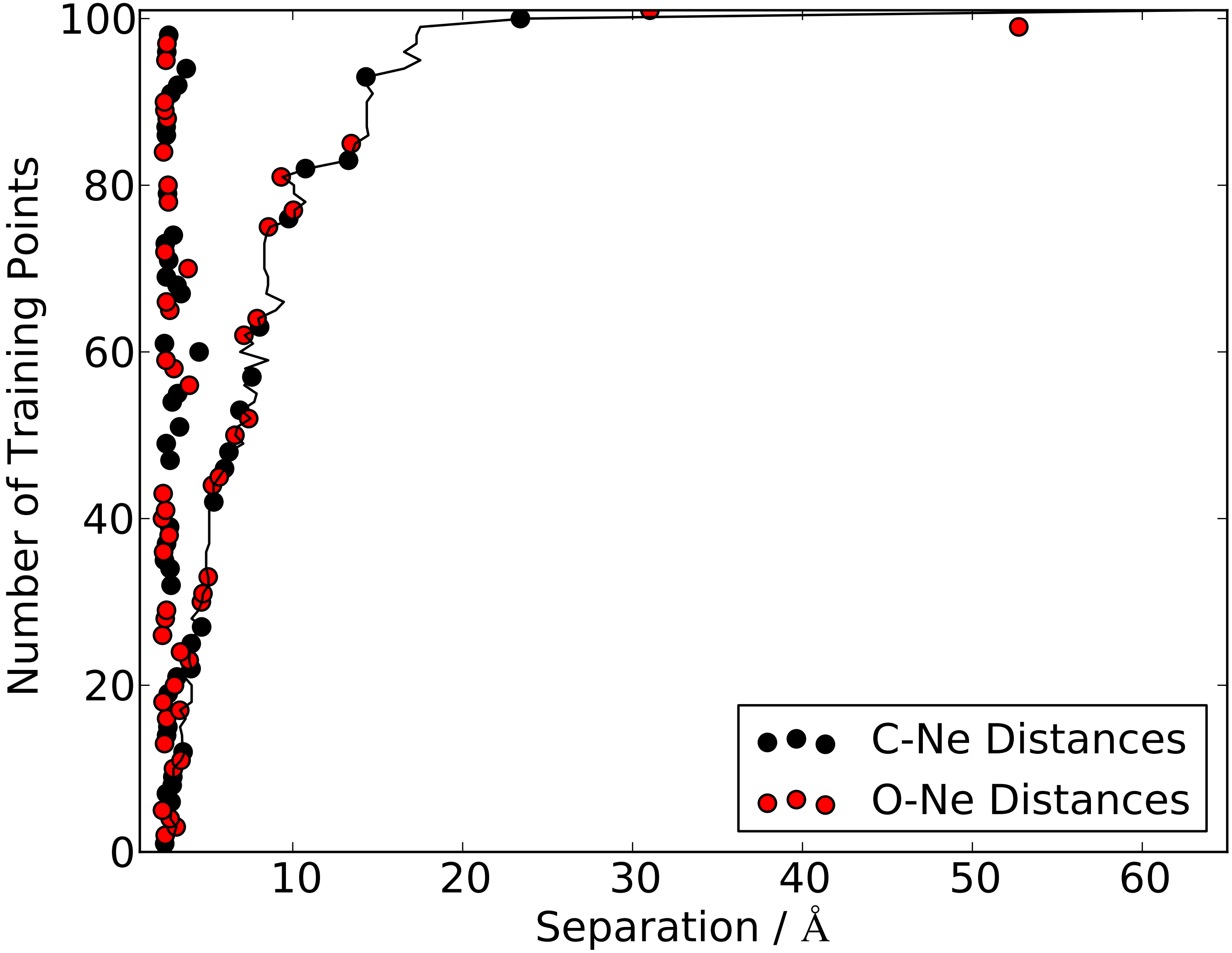

Number of Training Points
0
20
40
60
80
100
Separation / Å
1
2
3
4
5
6
7
8
9
H-Ne Distances
F-Ne Distances



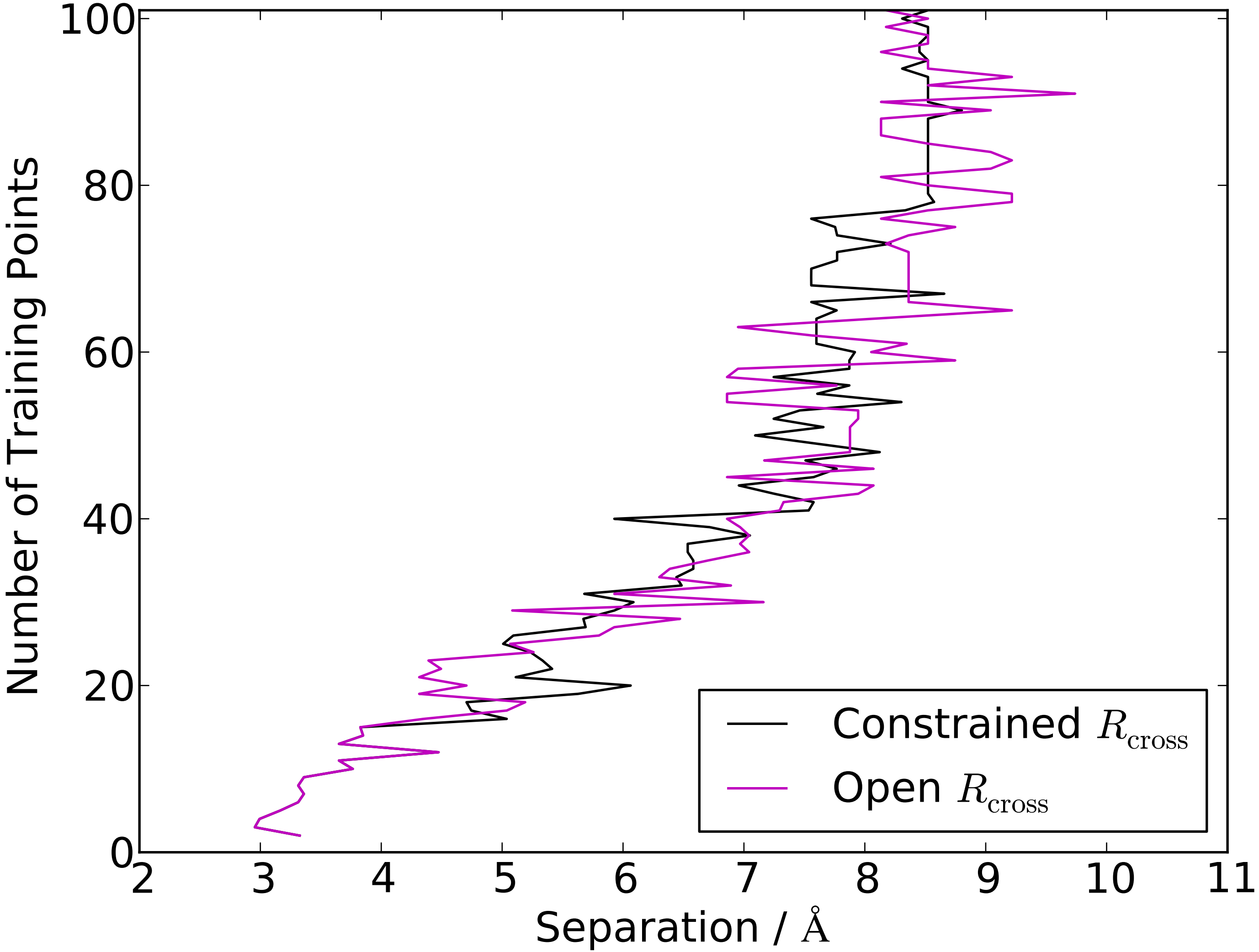
Number of Training Points
0
20
40
60
80
100
Separation / Å
1
2
3
4
5
6
7
8
9
H-Ne Distances
F-Ne Distances

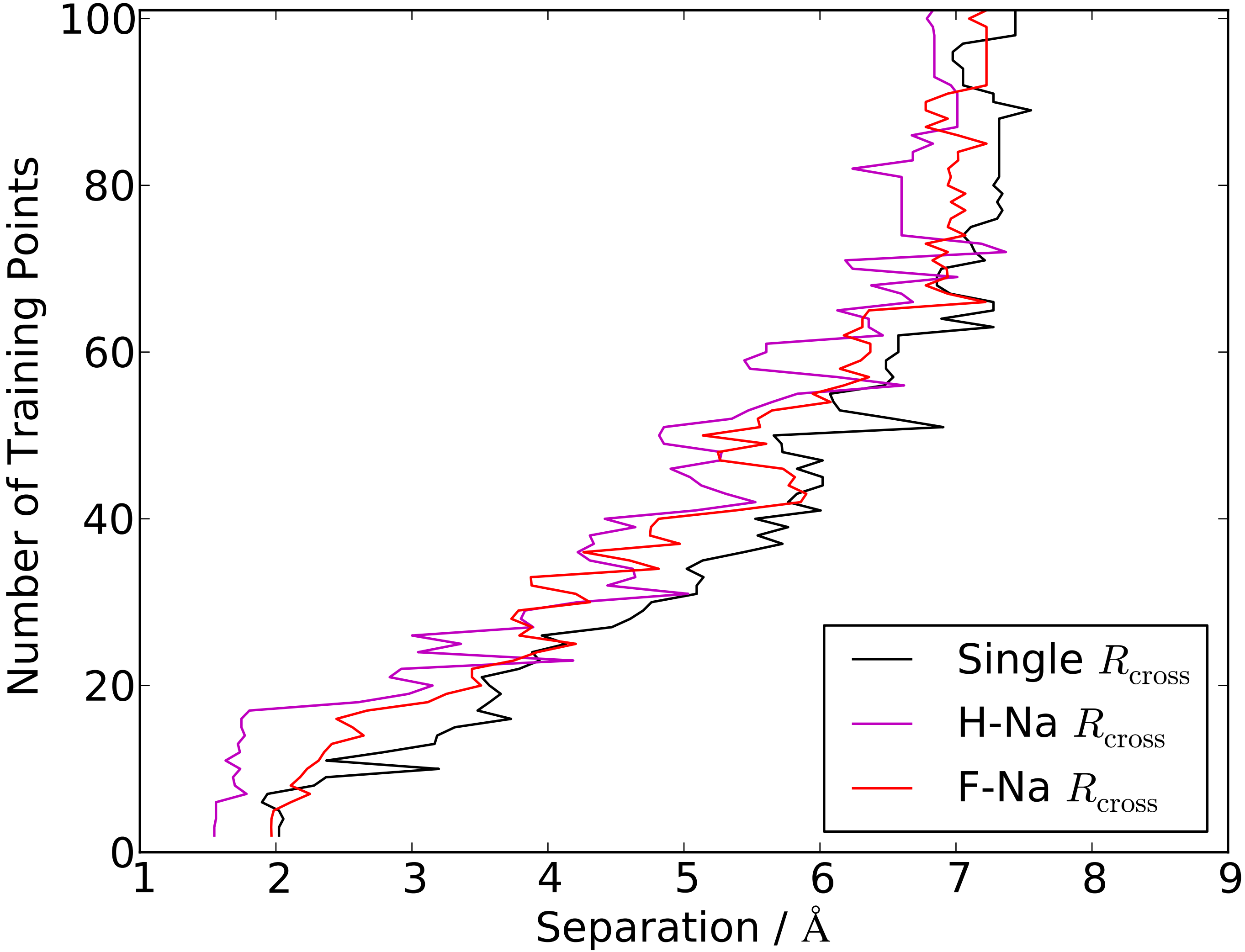


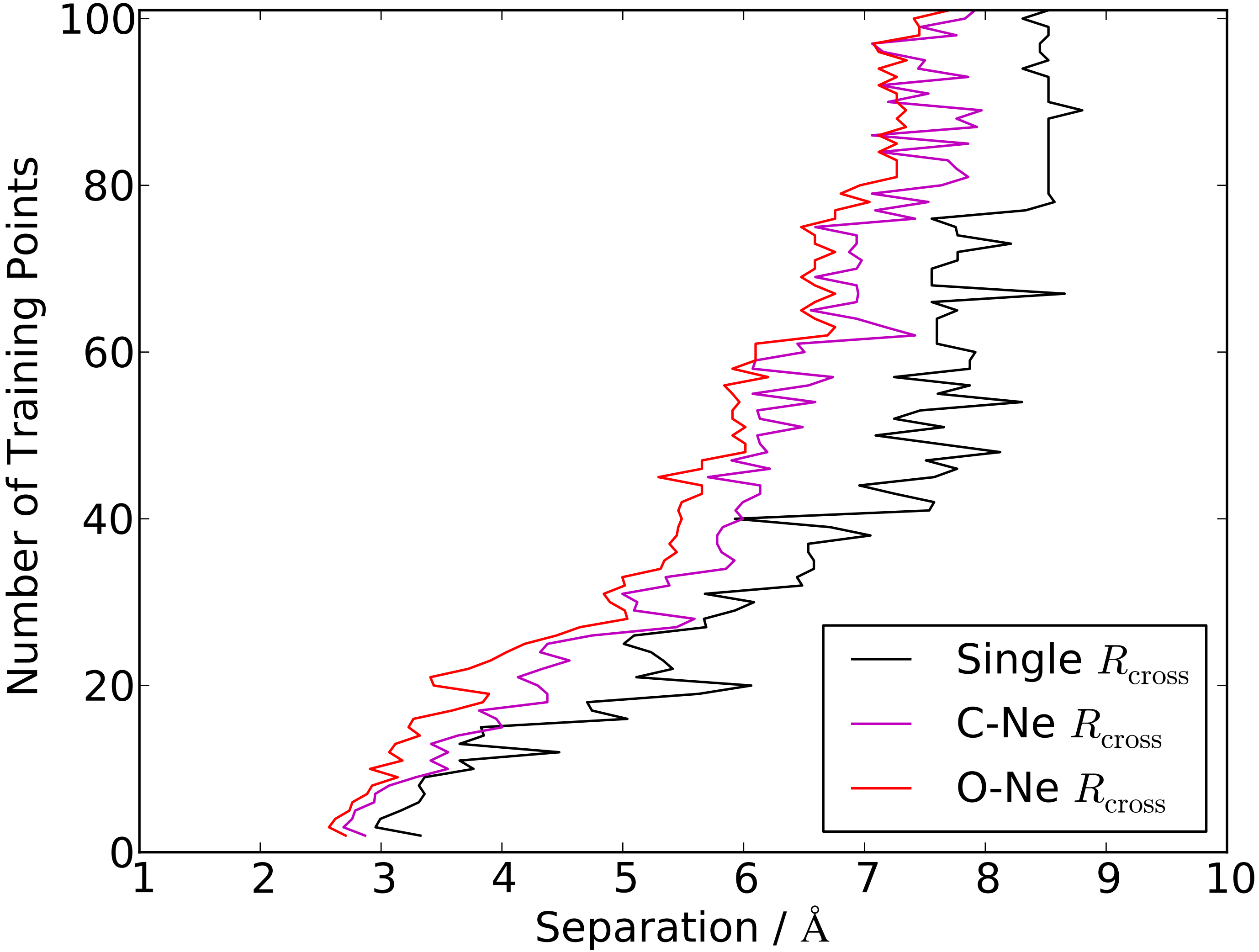
Number of Training Points
0
20
40
60
80
100
Separation / Å
2
4
6
8
10
12
14
16
18
●●● C-Ne Distances
●●● O-Ne Distances

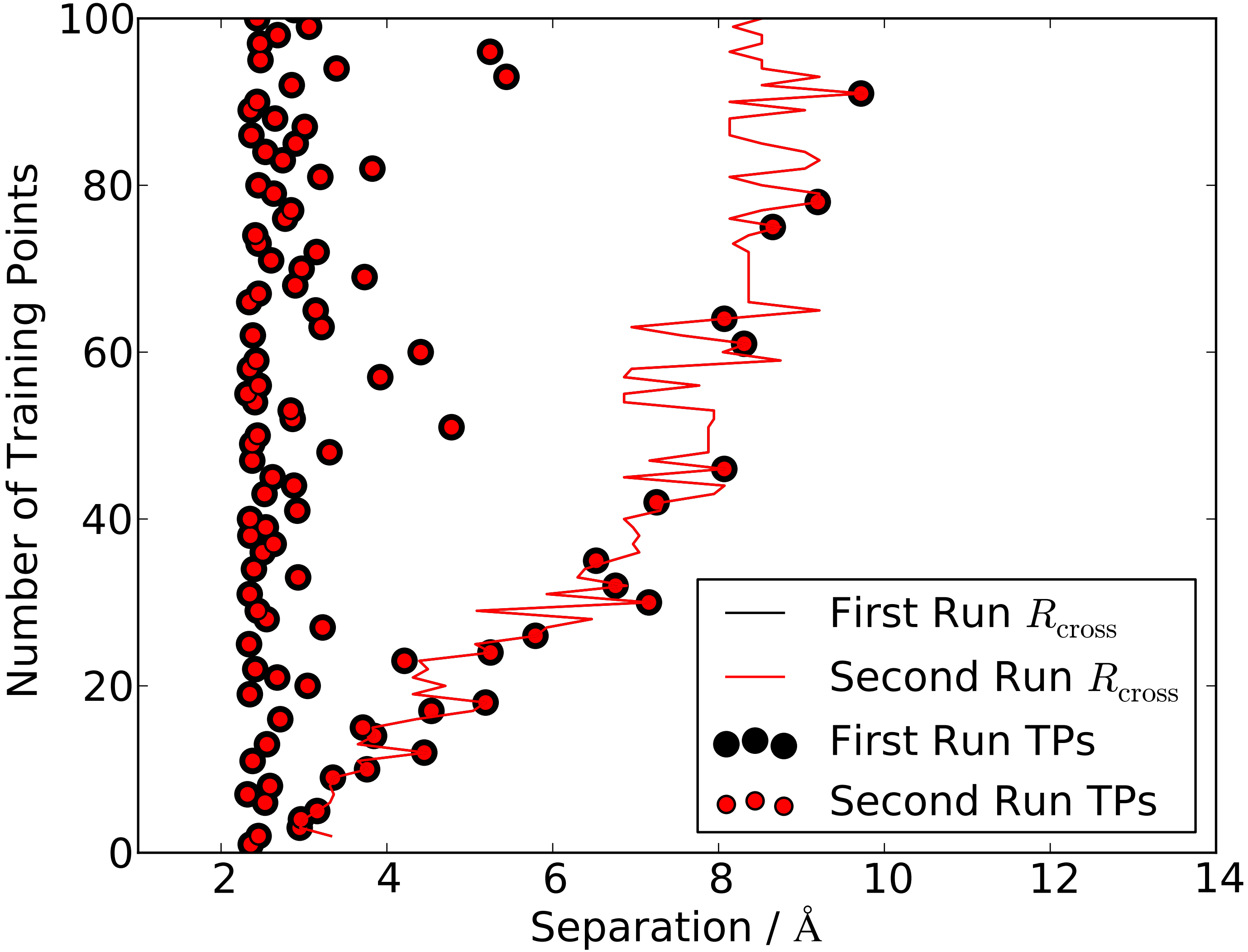


Number of Training Points
0
20
40
60
80
100
Separation / Å
10
20
30
40
50
60
C-Ne Distances
O-Ne Distances

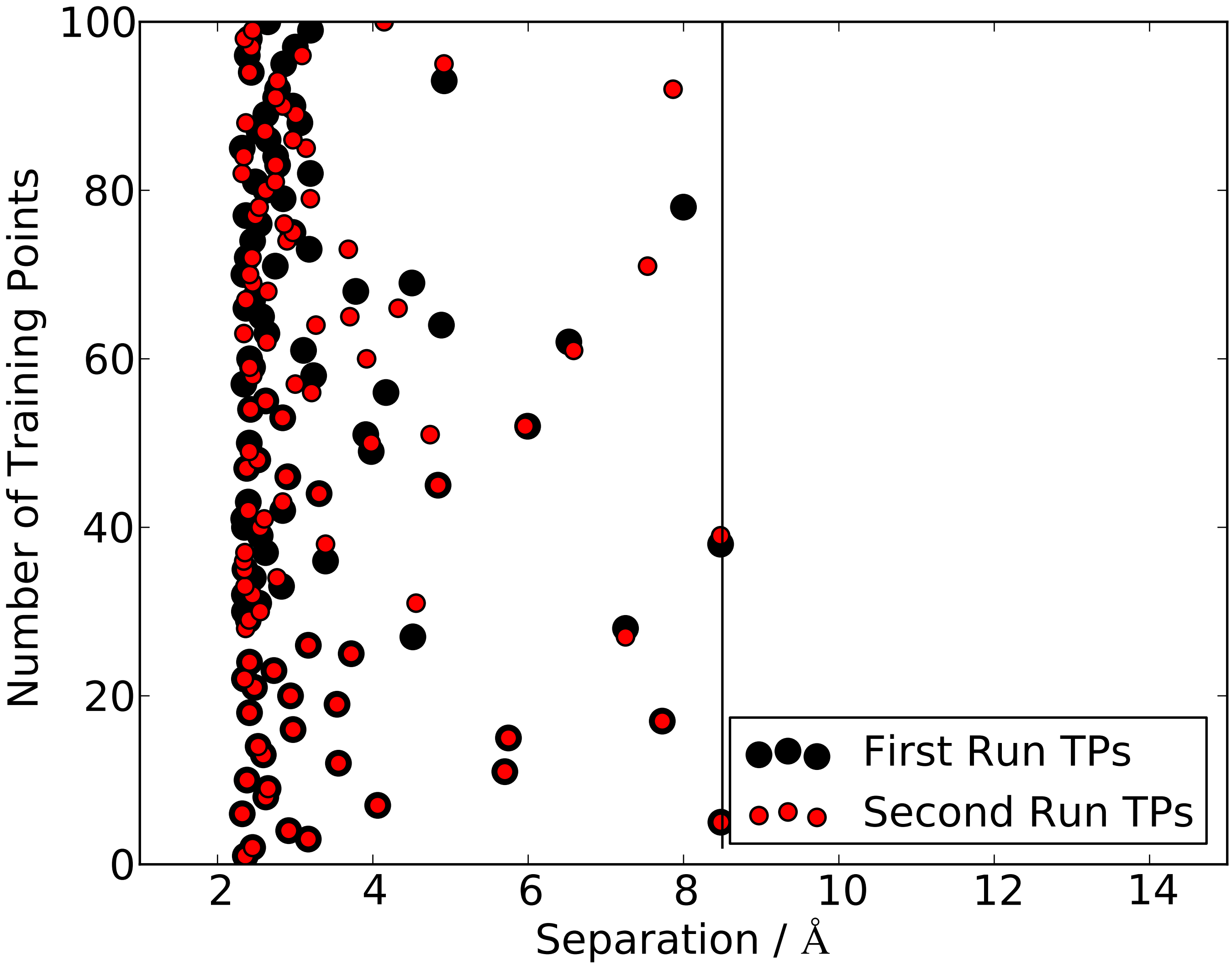

Number of Training Points
0
20
40
60
80
100
2
3
4
5
6
7
8
9
10
11
Separation / Å
Constrained R_{cross}
Open R_{cross}

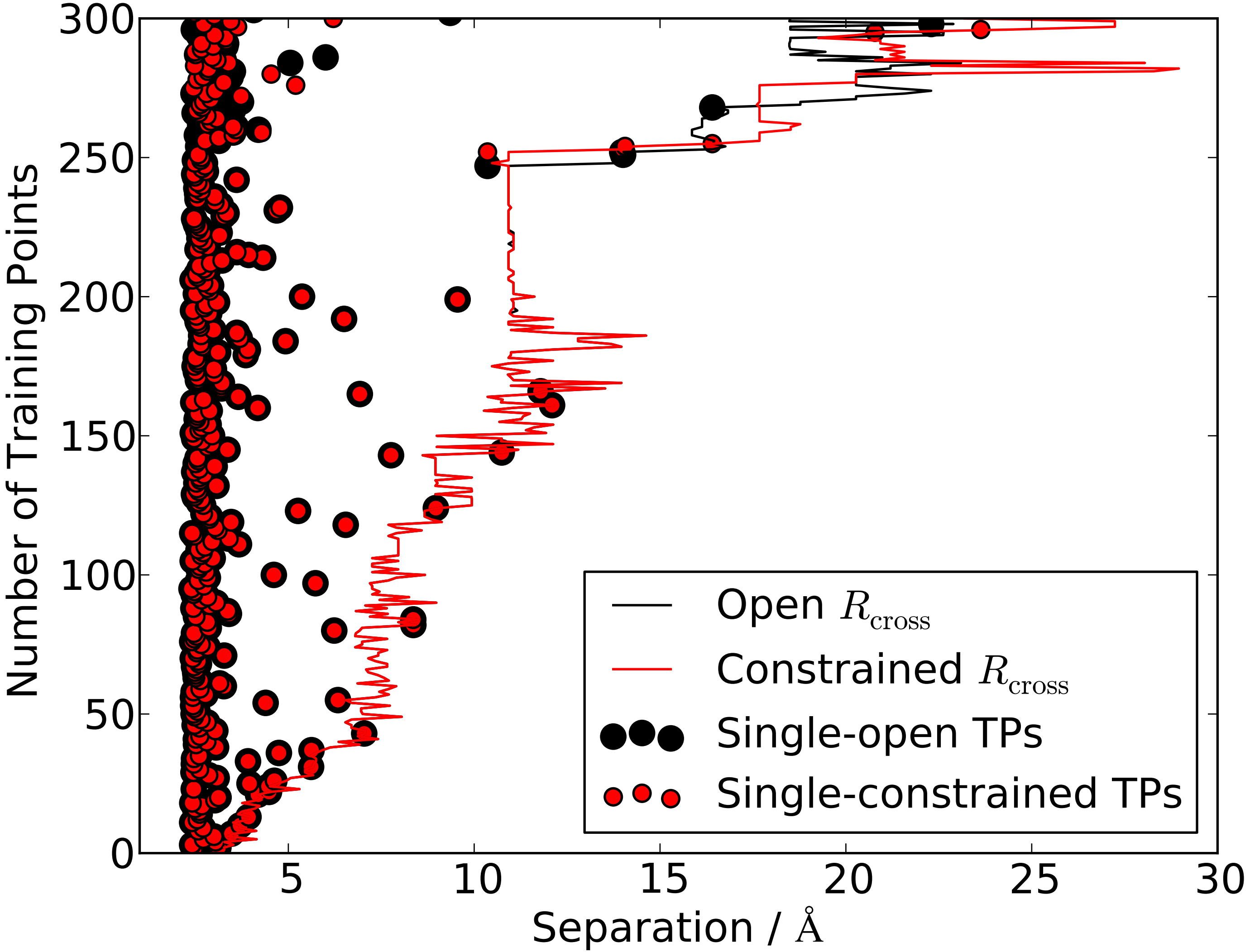

Number of Training Points
0
20
40
60
80
100
Separation / Å
1
2
3
4
5
6
7
8
9
Single R_{cross}
H-Na R_{cross}
F-Na R_{cross}

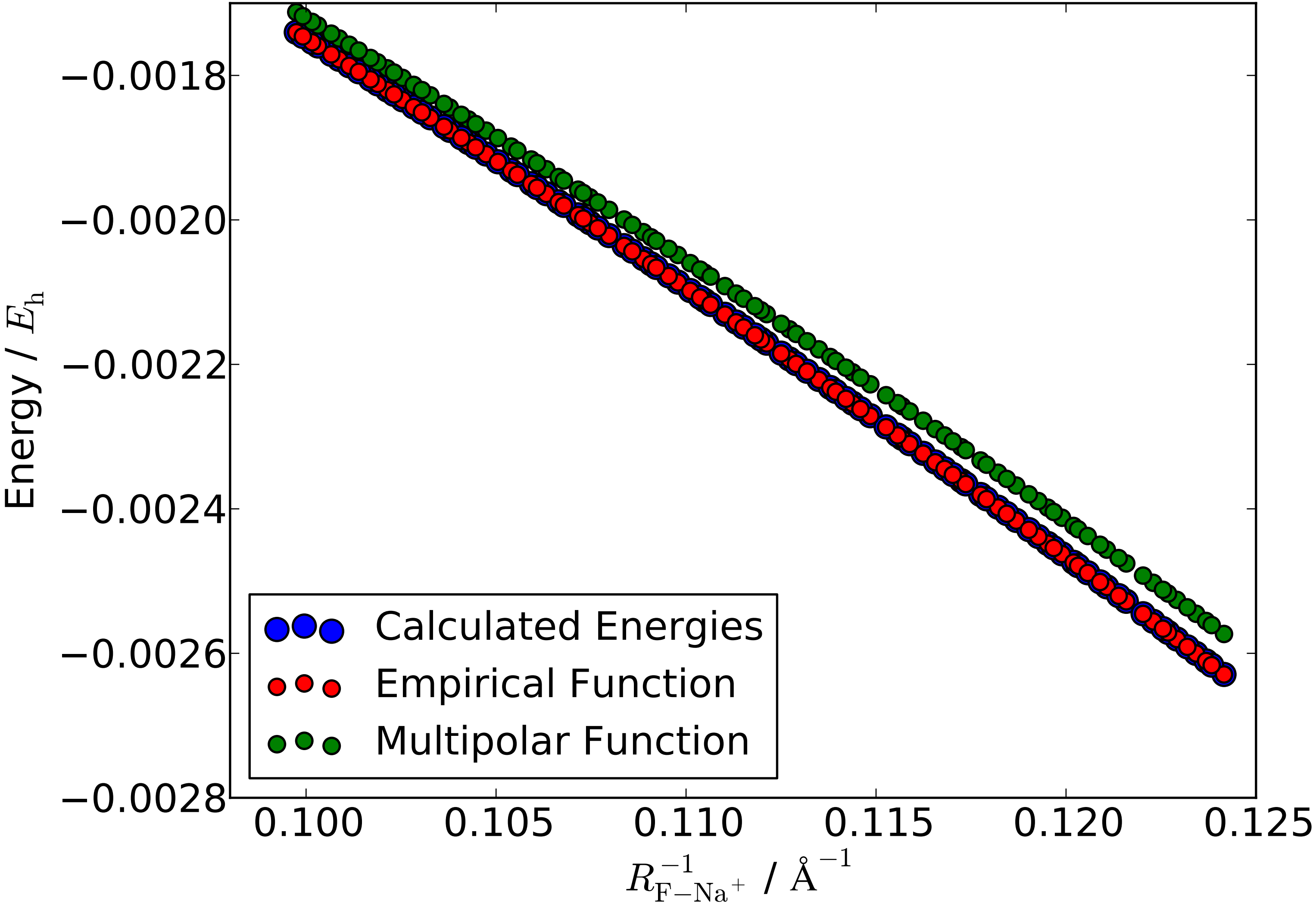

Number of Training Points
0
20
40
60
80
100
1
2
3
4
5
6
7
8
9
10
Separation / Å
Single R_{cross}
C-Ne R_{cross}
O-Ne R_{cross}



100
80
60
40
20
0
Number of Training Points
2
4
6
8
10
12
14
Separation / Å
First Run R_{cross}
Second Run R_{cross}
First Run TPs
Second Run TPs


Number of Training Points
0
20
40
60
80
100
Separation / Å
2
4
6
8
10
12
14
First Run TPs
Second Run TPs


Number of Training Points
300
250
200
150
100
50
0
5
10
15
20
25
30
Separation / Å
Open R_{cross}
Constrained R_{cross}
Single-open TPs
Single-constrained TPs


−0.0018
−0.0020
−0.0022
−0.0024
−0.0026
−0.0028
Energy / E_h
0.100
0.105
0.110
0.115
0.120
0.125
$R_{\mathrm{F-Na^+}}^{-1}$ / Å^{-1}
Calculated Energies
Empirical Function
Multipolar Function