Electronic Supplementary Information

Predicting hydrogen atom transfer energy barriers using Gaussian process regression

Evgeni Ulanov, Ghulam A. Qadir, Kai Riedmiller,
Pascal Friederich, Frauke Gräter

A Gaussian Process Regression

We here provide a more complete description of Gaussian Process Regression (GPR)[1] and the notation used in the main text.

A Gaussian Process (GP), $\{Y(x),\ x \in \mathbb{R}^p\}$, indexed with $p-$dimensional covariates $x \in \mathbb{R}^p$, characterizes the set of normal random variables $Y(\cdot)$, such that for any finite $n \geq 1$, the random vector $\boldsymbol{Y} = (Y(x_1), \ldots, Y(x_n))^\top$ follows a multivariate normal distribution (MVN). Notationally, we write $Y(\cdot) \sim \mathcal{GP}(\mu(\cdot), K(\cdot,\cdot))$, where $\mu(\cdot) : \mathbb{R}^p \to \mathbb{R}$, represents the mean function: $\mathbb{E}(Y(x)) = \mu(x)$, and $K(\cdot,\cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$, represent the non-negative definite covariance function: $\mathrm{Cov}(Y(x), Y(x')) = K(x,x')$. As a consequence, $\boldsymbol{Y} \sim \mathrm{MVN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is a $n \times 1$ vector derived from the mean function as $\boldsymbol{\mu}_i = \mu(x_i),\ i = 1, \ldots, n$, and $\boldsymbol{\Sigma}$ is an $n \times n$ covariance matrix given by $\boldsymbol{\Sigma}_{ij} = K(x_i, x_j),\ i, j = 1, \ldots, n$.

In the context of GP modelling, the covariance function is generally specified with some valid parametric functional form for $K(x_i, x_j)$ as:

$$K_{\boldsymbol{\theta}}(x,x') = \sigma^2 \left(C_{\boldsymbol{\theta}_c}(x,x') + g^2 \delta_{x,x'} \right), \tag{1}$$

where $C_{\boldsymbol{\theta}_c}(\cdot,\cdot) : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ is any valid class of correlation functions which can be characterized by the set of parameters $\boldsymbol{\theta}_c$, $g \geq 0$ is the nugget term that models the potential white noise of the data, $\sigma > 0$ models the process standard deviation, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_c, \sigma, g)$ embodies the complete set of parameters found in the defined covariance function.

Consider $\boldsymbol{Y}_o = (Y(x_1), \ldots, Y(x_n))^\top$ to be a collection of $n \geq 1$ observed values, which constitute the training data. Also, let $\boldsymbol{Y}_u = (Y(x_{n+1}), \ldots, Y(x_{n+m}))^\top$ represent a set of $m \geq 1$ unobserved data points which will serve as the test data. The main objective in GPR is to discover the predictive distribution of $\boldsymbol{Y}_u$. This is essentially equivalent to finding the conditional MVN distribution of $\boldsymbol{Y}_u$ given $\boldsymbol{Y}_o$. Specifically, under the GP specification $Y(\cdot) \sim \mathcal{GP}(\mu(\cdot), K_{\boldsymbol{\theta}}(\cdot,\cdot))$, the corresponding conditional MVN distribution for $\boldsymbol{Y}_u \mid \boldsymbol{Y}_o$ is determined by the following mean and covariance specification:

$$\boldsymbol{\mu}_{u|o} = \boldsymbol{\mu}_u + \boldsymbol{\Sigma}_{uo} \boldsymbol{\Sigma}_{oo}^{-1} (\boldsymbol{Y}_o - \boldsymbol{\mu}_o) \tag{2}$$

$$\boldsymbol{\Sigma}_{u|o} = \boldsymbol{\Sigma}_{uu} - \boldsymbol{\Sigma}_{uo} \boldsymbol{\Sigma}_{oo}^{-1} \boldsymbol{\Sigma}_{uo}^\top, \tag{3}$$

where $\boldsymbol{\mu}_o = (\mu(x_1), \ldots, \mu(x_n))^\top$, $\boldsymbol{\mu}_u = (\mu(x_{n+1}), \ldots, \mu(x_{n+m}))^\top$, $\boldsymbol{\Sigma}_{uo}$ is a $m \times n$ cross-covariance matrix such that its $(i,j)^{th}$ element is given by $K_{\boldsymbol{\theta}}(x_{n+i}, x_j),\ i = 1, \ldots, m,\ j = 1, \ldots, n$, $\boldsymbol{\Sigma}_{uu}$ is a $m \times m$ covariance matrix corresponding to $\boldsymbol{Y}_u$ with its $(i,j)^{th}$ element given by $K_{\boldsymbol{\theta}}(x_{n+i}, x_{n+j}),\ i, j = 1, \ldots, m$, and $\boldsymbol{\Sigma}_{oo}$ is a $n \times n$ covariance matrix corresponding to $\boldsymbol{Y}_o$ with its $(i,j)^{th}$ element given by $K_{\boldsymbol{\theta}}(x_i, x_j),\ i, j = 1, \ldots, n$. Generally, the conditional mean $\boldsymbol{\mu}_{u|o}$ can be considered as the point prediction for $\boldsymbol{Y}_u$, and the diagonal entries of $\boldsymbol{\Sigma}_{u|o}$ provide the corresponding prediction variances.

B Root-Mean-Square Error

For completeness, we also include in table 1 the root-mean-square error (RMSE) of the used models for the trajectory test set.

Table 1 Test Root-Mean-Square Error in kcal/mol for all trajectory data (top row) or only up to a certain maximum transition distance d (bottom two rows).

Test set	$\text{SOAP}_{\text{Full}}$	$\text{SOAP}_{\text{Traj}}$	MGK	$\text{PaiNN}_{\text{Ind}}$	$\text{PaiNN}_{\text{Ens}}$
All (no cut-off)	**4.92**	4.82	4.86	5.43	5.00
Cut-off $d \leq 3$Å	4.49	**4.44**	4.51	4.90	4.49
Cut-off $d \leq 2$Å	3.87	3.91	3.89	3.88	**3.51**

C Effect of node feature ξ

In table 2 we show the effects of excluding the node feature ξ of the marginalized graph kernel based model on the test MAE for the trajectory data. It can be seen that including this information noticeably decreases the MAE.

Table 2 Test MAE in kcal/mol for the marginalized graph kernel with and without node feature ξ.

	All (no cut-off)	Cut-off $d \leq 3$Å	Cut-off $d \leq 2$Å
With ξ	**3.37**	**3.19**	**2.85**
Without ξ	3.76	3.60	3.22

D Additional DFT details

The DFT data and its generation was described in ref.[2]. The barriers were taken as the difference between the single point energy calculations, with no further temperature corrections. The trajectory dataset was constructed using structures obtained from MD simulations solvated in water. The systems used for the DFT calculations, however, included only the reactants in vacuum. We expect solvent effects to be negligible, since the reactions would happen on short distances, in a crowded environment, and with no charge relocation during the process. Furthermore, no catalysis was involved in the reactions.

E Data efficiency fit

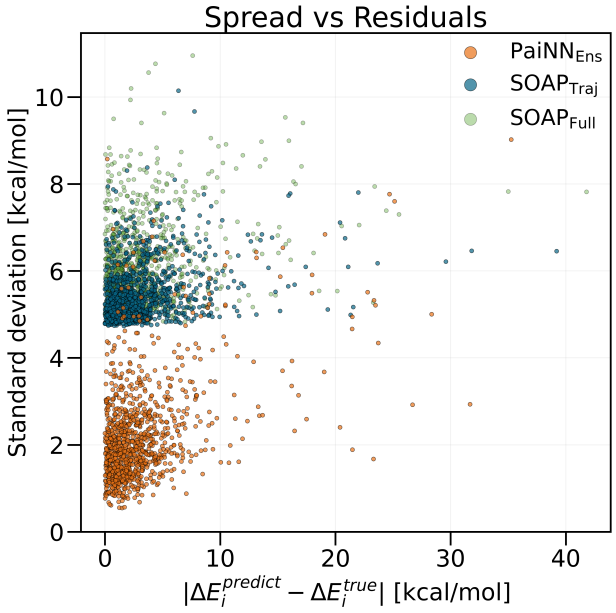
In table 3 we show the fitting metrics for the power-law (18) of table 2 in the main text. To avoid any confusion, the scores shown here refer to the fitted model MAE values for a given number of training points. For example, an MAE score in the table below of 0 would mean that the power law fits perfectly to the means of model test MAE scores for all training set sizes.
The fits were performed on a model's mean test MAE, for a given number of training points, using the *curve_fit* function of SciPy[3]. The standard deviation of a model's MAE across multiple runs, given a certain number of training points, was also included in the optimization for the weight calculation.

Table 3 Power law fit metrics for table 2 in the main text.

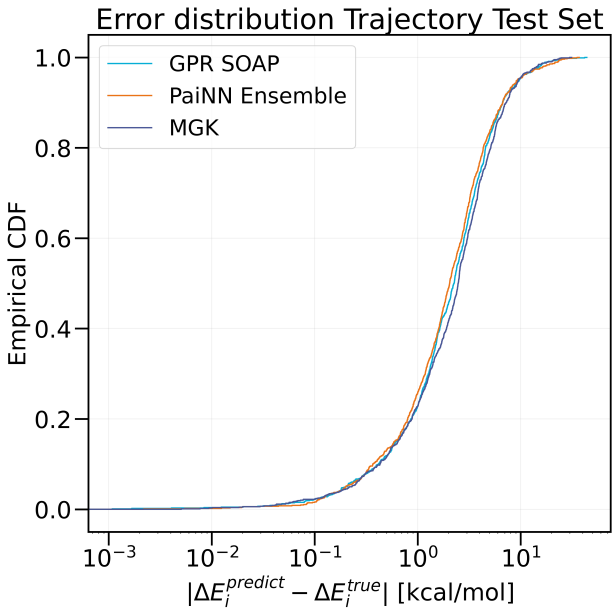
Method	Fit MAE	Fit RMSE	Fit R^2
$\text{SOAP}_{\text{Full}}$	0.08	0.14	0.99
$\text{SOAP}_{\text{Traj}}$	0.03	0.05	1.00
$\text{PaiNN}_{\text{Ind}}$	0.21	0.33	0.99
$\text{PaiNN}_{\text{Ens}}$	0.21	0.36	0.99

F Supplementary Figures

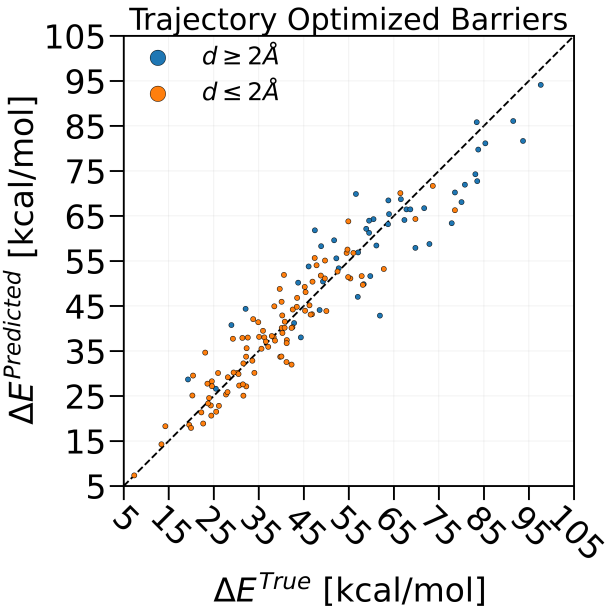
Figure S1 shows additional figures, such as the absolute error vs ensemble spread and GPR predicted standard deviation (a), the empirical distribution of the absolute error (b), the parity plot of the optimized trajectory barriers (c), and the mean absolute error of the optimized barriers as a function of cut-off distance (d).



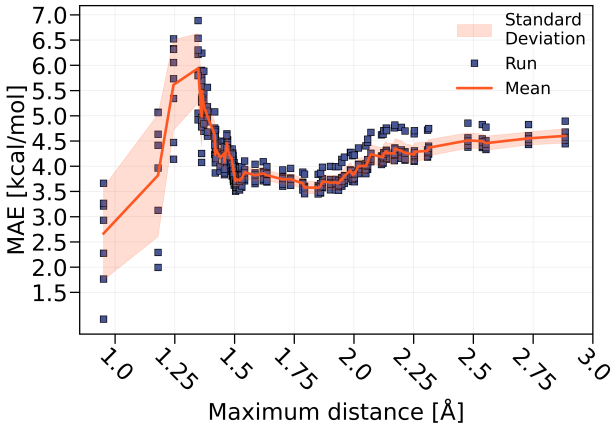
(a) Absolute error of the different models used vs. their derived standard deviation. In the case of GPR, this is the square root of the predicted variance and for the PaiNN ensemble model this is the standard deviation of the student's t-distribution.



(b) Empirical cumulative distribution of the absolute errors on the unoptimized trajectory energy test barriers.



(c) Predictions of the optimized barriers using combination of PaiNN and GPR as described in the main text.



(d) MAE of the optimized trajectory barriers using different distance cut-offs.

Figure S1

References

[1] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, November 2005.

[2] Kai Riedmiller, Patrick Reiser, Elizaveta Bobkova, Kiril Maltsev, Ganna Gryn'ova, Pascal Friederich, and Frauke Gräter. Substituting density functional theory in reaction barrier calculations for hydrogen atom transfer in proteins. *Chemical Science*, 15(7):2518–2527, February 2024.

[3] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau,

Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3):261–272, March 2020.