

EON GPR Test Systems

Under the Auspices of REAXPRO

Alejandro, Hannes, Rohit, Satish, Maxim

March 3, 2021

Brief system descriptions of test cases considered. Also includes developer TODO items.

Many of the systems studied here are taken from Koistinen et al. [9] and the data has been kindly provided by Villi. Before describing the data-sets used, as well as the results; we will give a brief overview of the coupling of eon to the GPR Dimer library. The results of Koistinen [8] established a baseline for studying the effects of accelerating saddle search methods namely:

- Nudged elastic band calculation [10, 11]
- Tailored bespoke covariance functions for atomic data [11, 9]
- Proof of concept studies of accelerating surface walking methods like the dimer [1] using Gaussian Process Regression [9]

However, these proof of concept studies were implemented in MATLAB and were not amenable to large systems due to both theoretical restrictions and computational inefficiency.

Architecture

The structure of the interface is given below:

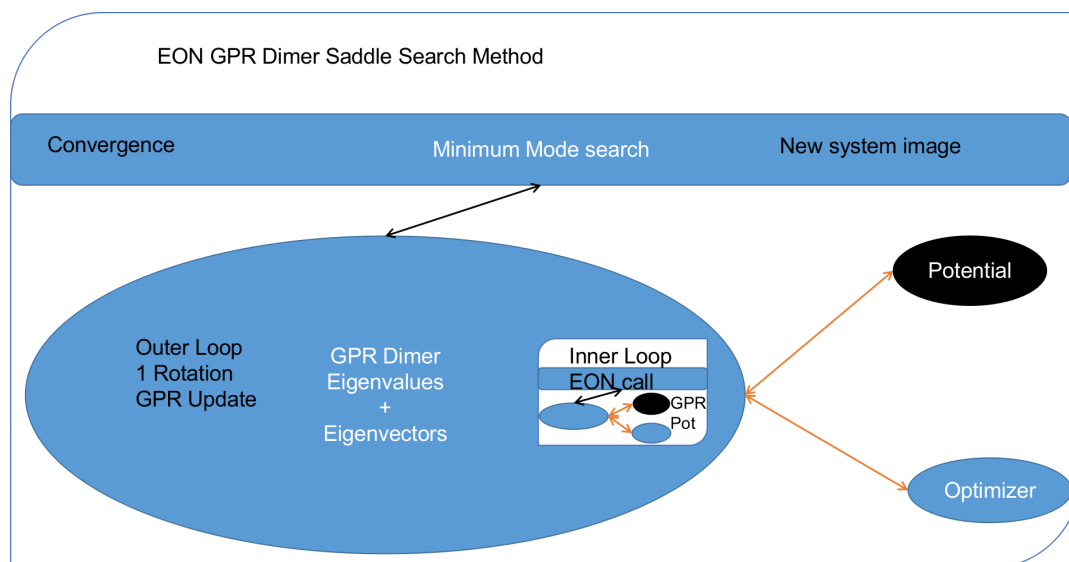


Figure 1: EON to GPR Dimer interface

Programming Methodology

The GPR dimer is implemented as a stand-alone library (Apache2 licensed) with minimally invasive changes to the EON code. This allowed for the use of modern and performance oriented programming practices like:

- Modern build system
 - Allows for cross-platform optimizations
 - Integrates with MKL for improved HPC performance
- Extensive Testing with the googletest framework
 - Both unit tests and integration tests are provided
 - eon does not include a unit testing framework
 - * An extensive set of manual test cases are provided to replicate all existing literature results
- Modular design following OOP practices
 - Separation of GPR hyperparameter optimizations
 - * Allows for updating both the minimizer, and the possibility of implementing monte-carlo methods for hyperparameter optimization
 - Separation of GPR surface and the LBFGS call
 - * This allows for the usage of other minimizers like the Lanczos
 - Separation of covariance functions
 - * A standard interface for the implementation of future covariance functions has been provided
- Only user determined parameters are used internally

We use the numerical linear algebra library, Eigen; which supports Intel MKL optimizations. Eigen is also used by EON.

The library itself has a self contained executable for testing which consumes the same input data as the MATLB code. The interface is seamless, with the addition of input parameters done through the standard eon configuration file.

The structure of the library is such that it accepts a general potential function from EON and then uses it to calculate the GPR approximation to the potential energy surface. For the relaxation phase, the trained GPR potential in EON is called.

A user defined set of pruning parameters have been implemented which ensure optimal and bound scaling of the GPR.

EON Changes

A Cmake build system was implemented for efficient cross-platform builds. We augment the potentials with an interface to AMS. A GPR potential was also written, which is called by the library for the relaxation phase. Additionally, the build system allows for HPC speed-ups which are determined automatically and uses the MKL.

User Inputs

All the configurations follow the standard ini file specification of EON. As the gprdimer is implemented as a Minimum Mode Search method; as a replacement for the existing lanczos or dimer methods. Consider the standard configuration for a saddle search run:

```
1  [Main]
2  job = saddle_search
3  temperature = 300
4  random_seed = 706253457
5
6  [Potential]
7  potential = morse_pt
8
9  [Optimizer]
10 converged_force = 0.001
11 max_iterations = 1000
12
13 [Saddle Search]
14 displace_least_coordinated_weight = 1.0
15 displace_radius = 3.3
16 displace_magnitude = 0.01
17 min_mode_method = dimer
18 max_energy = 10.0
```

- AMS Interface

```
1  [AMS]
2  engine = ADF
3  xc = B3LYP
```

The current implementation uses an I/O pipe and supports any of the engines exposed by AMS; however efforts to use the standard AMSpipe protocol for AMS are ongoing.

- GPR Dimer The interface through EON augments the existing `conf.ini` file with the following GPR specific parameters.

```

1  [GPR Dimer]
2  finite_angle = 0.05
3  converged_angle = 0.0873
4  relaxation_converged_angle = 0.01
5  max_initial_rotation_iterations = 6
6  max_relaxation_rotation_iterations = 10
7  divisor_t_dimer = 10
8  max_outer_iterations = 300
9  max_inner_iterations = 1000
10 max_midpoint_displacement = 0.5
11 rotation_opt_method = "lbfgs"
12 translation_opt_method = "lbfgs"
13 inner_opt = "dimer"
14 active_radius = 10.0
15 dimer_separation = 0.01
16 convex_region_step_size = 0.1
17 max_step_size = 0.1
18 force_threshold = 0.001
19 ratio_at_limit = 0.666666666667
20 nogp_initial_rotations = true
21 has_many_iterations = true
22 hyperparameter_opt_method = "scg"
23 gpr_variance = 1e-7
24 gpr_noise_variance = 1e-5
25 prior_mean = 0.0
26 prior_variance = 1.0
27 prior_degrees_of_freedom = 20
28 # OPT parameters
29 opt_max_iterations = 400
30 opt_tol_sol = 0.001
31 opt_lambda_limit = 1e16
32 opt_lambda_init = 100
33 gpr_jitter_variance = 0
34 # OPT parameters
35 opt_max_iterations = 400
36 opt_tol_sol = 0.001
37 opt_lambda_limit = 1e16
38 opt_lambda_init = 100
39 gpr_jitter_variance = 0

```

Additionally, we have implemented debugging levels in the code for ease of visualization; which are able to generate snapshots of the GPR surface after each call to the true energy and forces (by exhaustively calculating energies over a user defined grid of points around

moving atoms) for one dimensional systems.

GPR Model

The basis.

TODO Toy Systems

These include standard benchmarks for the minimum mode saddle search problems in general.

DONE Hydrogen Adatom on Copper

This system was used for porting the code from MATLAB. The C++ code runs 10x faster than the MATLAB code as can be seen in Table 1

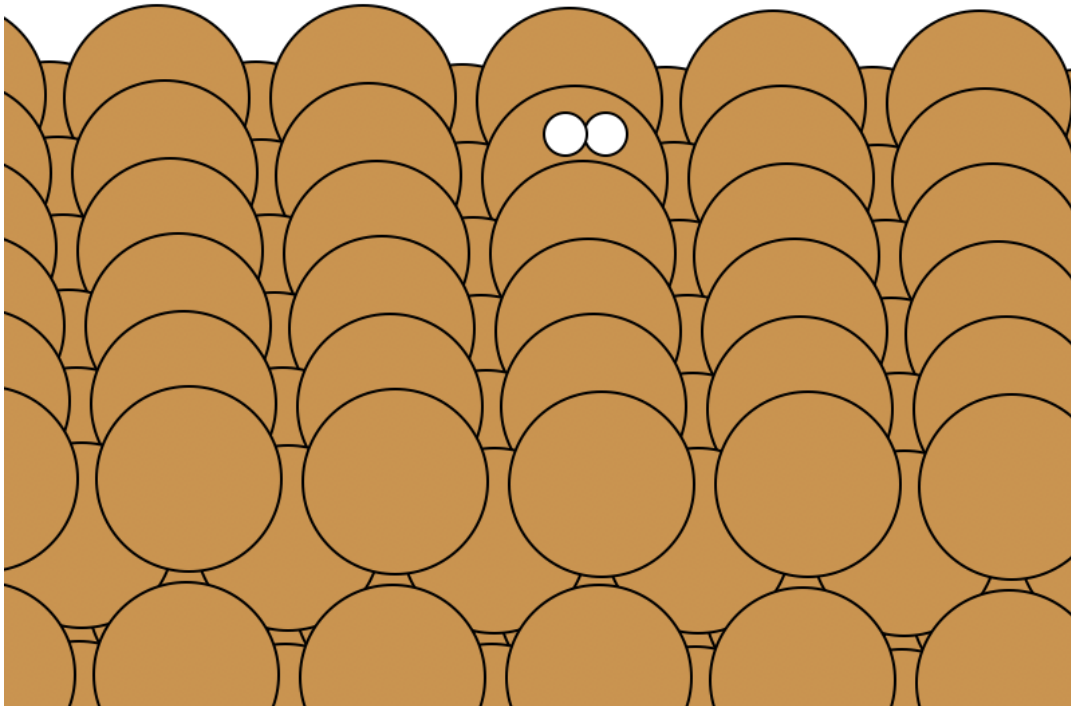


Figure 2: Hydrogen atoms on a Copper surface

For both systems, the results are the same, 5 hyperparameter optimizations, and the number of calls to the true energy and forces (7) is also the same. Each EAM call takes 0.002 seconds so the total PES time is 0.014. (-O3) refers to the compiler optimization level.

Table 1: Hydrogen on Copper. EAM based potential. HyT is the time taken for hyperparameter optimizations. T indicates time.

Algorithm	Overhead	HyT	Total T	Cost
MATLAB GPRD	230.646	81.176	230.66	32.95
CPP GPRD (O3)	21.4993	20.99	21.5133	3.0713
CPP GPRD (Oo)	52.165	50.74	52.1797	7.452

DONE Platinum Adatom on Platinum

Chronologically the first system tested with EON. The system has some pleasant qualities which make it easy to work and test:

- Easy control of the degrees of freedom
 - Achieved by varying the number of moving atoms and the fixed atoms
- Homogeneity meant neither atomic numbers nor atomic masses were required
- Cheap throwaway potential (Morse)

This system takes **0.069s** with the dimer, and **0.109s** with the GPRD in 5 cycles.

Degrees of Freedom	System Size	Active Radius	GPR Time	Dimer Time
48	1008	5	0.109	0.069
12	1008	3.3	0.063	0.069

45 degrees of freedom arise from having a cut off radius of 5, which means the surface atoms enter the covariance matrix, and 3 degrees of freedom are from the movement of the adatom itself. For this system the GPR PES has been mapped out as a function of the planar coordinates in Fig. 5.

In this instance, the GPRD can converge to the saddle point with a lower active radius of 3.3, however, this takes 9 image evaluations; and the potential energy surface suffers, as can be seen in 4.

DONE Literature Extensions

DONE Pruning Rationale

Hyperparameters and Expectations

For inference, even in higher dimensions; it is not the distribution itself which is important, but the **expectation** value of the the distribution:

$$E_{p(\theta|y)}[f(\theta)] = \int f(\theta)p(\theta|y)d\theta,$$

where $p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)d\theta}$

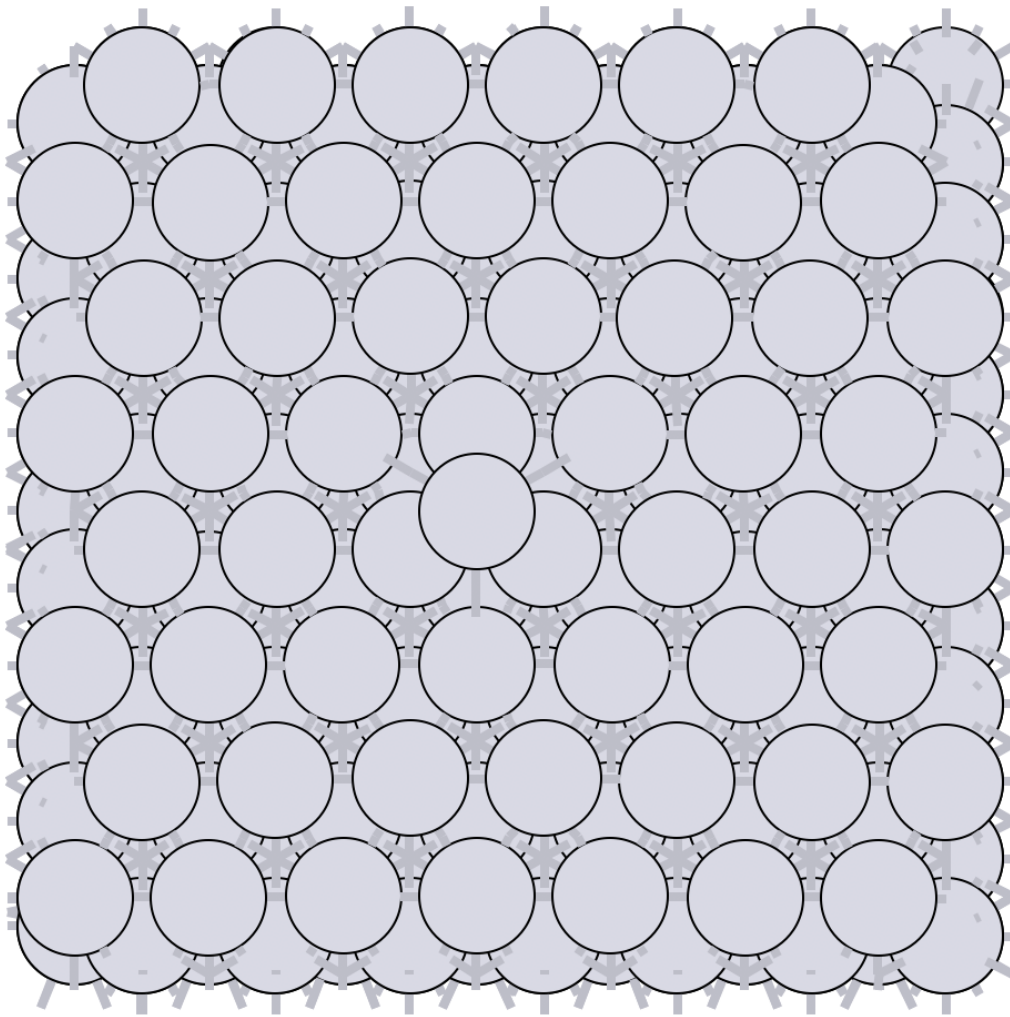


Figure 3: Platinum adatom on a platinum island

Where we note that $p(\theta|y)$ is the posterior probability ($\mathbb{P}(\theta)$); $p(y|\theta)$ is the likelihood ($\mathbb{L}(\theta)$); $p(\theta)$ is the prior ($\pi(\theta)$) and the normalizing constant $\int p(\theta|y)p(\theta)d\theta$ is known as the evidence (Z). The notation of Speagle [16] is indicated.

Visually, this master equation (Bayes theorem) can also be visually represented by Fig. 6.

Where we recognize that the marginal probability, or the evidence is the integral over the parameter space; and describes the independent probability of a single event regardless of the others.

None of the preceding discussion implied any limits on the output range of these expectation values, and in a space of functions [14, 17]; if we consider a regression problem modeled by a Gaussian process we obtain pleasantly simple posterior distribution (a Gaussian) [11, 10, 9, 8].

Having obtained a GPR model from the observations (which are high dimensional points each consisting of the positions of all points considered, R which is of size $N \times 3$, E which is the potential energy of size $N \times 1$ and F which is of size $N \times 9$), we predict additional points on the GPR surface at a fraction of the cost of calculating the energy and forces from our potential.

Adaptive Prior Improvements with Pruning

Due to the high dimensional nature of the relevant space of observations; Koistinen [8] have used a weakly informative (broad) Gaussian prior over the hyperparameters. Empirical studies [4] suggest that the hyperprior distribution; does not have a strong influence on the final predictions; however their studies were in the large data limit and thus are not surprising [13]. Unlike in simulated train-test data sets, as per the algorithm outlined [9], each call for the true energy and surfaces triggers the formation of new GPR surface (see Fig. 5). Indeed, given the high cost and low scalability; it is not feasible to reach the high data limit when the effect of the prior is overcome by the likelihood. Most of the models then, are formed in the low data limit, when the priors are of importance. From a frequentist or machine-learning perspective, the prior naturally gives rise to the “regularization” effect [13], reducing overfitting which is normally explicitly written into models [5, 3].

Since we calculate the fitness of each final point on the GPR surface, which is the maximum force in any direction; we can remove poor points from the data. Note that the effect of this in terms of the posterior can be written out for clarity as:

$$p(\theta|y_1) = p(y_1|\theta)p(\theta)/Z_1 \text{Likelihood from one data point} \quad (1)$$

$$p(\theta|y_1, y_2) = p(y_1, y_2|\theta)p(\theta)/Z_{12} \text{Likelihood from two data points} \quad (2)$$

$$p(\theta|y_1, y_2, y_3) = p(y_1, y_2, y_3|\theta)p(\theta)/Z_{123} \text{Likelihood from two data points} \quad (3)$$

$$p(\theta|y_1, y_3) = p(y_1, y_3|\theta)p'(\theta)/Z_{123} \text{All probabilities considered are Gaussian} \quad (4)$$

$$p(\theta|y_1 \dots y_n) = p(y_1 \dots y_n|\theta)p(\theta)/Z_{1 \dots n} \text{Likelihood from n data points} \quad (5)$$

$$(6)$$

As the product of two Gaussian distributions are also a Gaussian, since we have a Gaussian prior, we can absorb the effect of the second data point into the prior. This means that this iterative modeling could be seen as a case of using a new prior (namely, the posterior of the previous step) with a new data point; which is shown also in Fig. 7.

Normally, there is no need to compute the model posteriors in this manner, since the model is equivalent, irrespective of being trained one point at a time; or on a set of inputs [13]. However, the effect of poor data is more explicit above; if the likelihood is written out as above. More generally, active data set restrictions have been reported to improve efficiency [15].

Perhaps more importantly, the local optimization of the hyperparameters scales poorly; $O(N^3)$ with respect to the number of entries in the covariance matrix [14]; practically, this means that large systems and long timescales are inaccessible in theory as the number of observations increases with each step until convergence. With pruning, we are able to put a bound on the $O(N^3)$ term. Note that overly zealous pruning will lead to too little information being present for the GPR. Adaptive pruning might also help steer the system to saddle points where convergence is difficult. It should be kept in mind that the generation of PES samples (DFT forces and energies) scales worse than $O(N^3)$ [7] typically; and makes the dimer method unfeasible.

By implementing the pruning logic as shown in Fig. 8 using user parameters, we are able to achieve better convergence and in some cases, improved performance as well. Note that since we test for convergence using the maximum forces at the end of each LBFGS translation, we have the results of the fitness function for pruning at a later stage.

Quantifying Overheads

We define the GPR overhead to be the difference between the total time elapsed and the number of samples taken from the true PES times the average time (t_{DFT}^-) for calculating the true forces and energies.

$$O_{\text{method}} = T_{\text{elapsed}} - (t_{DFT}^- \times N_{\text{samples}})$$

Note that this measure includes the time taken by the LBFGS on the GPR surface. We can apply the same formula to the dimer method as well; in order to get an estimate of the normal EON dimer overhead. This means that we can also define a cost function (lower is better) as:

$$C_{\text{method}} = \frac{O_{\text{method}}}{N_{\text{samples}}}$$

The cost function is per unit time taken for evaluating a single sample of the true potential energy, which means lower is better.

DONE Compute Configuration

The toy systems above were studied with resources provided by the Icelandic High Performance Computing Centre at the University of Iceland.

All the systems described below were run on individual thin nodes of the Cartesius supercomputing cluster with resources provided by SURFsara. Each node for a run consists of 2×16 -core 2.6 GHz Intel Xeon E5-2697A v4 (Broadwell) CPUs with 64 GB. All calculations were accelerated with the Intel Math Kernel Library (2020) which provides architecture optimized multi-threaded math functions. The GPR dimer algorithm consists of dense matrix inversions and so distributed memory (MPI) parallelism has not yet been considered for it; however the Amsterdam Modeling Suite was used for the potential energy calculations; which does utilize both the MKL and MPI.

The B₃LYP [2] hybrid functional was used which involves a DFT correlation with a combined DFT and Hartree-Fock exchange and is known to reproduce geometries and binding energies of molecular systems with accuracy comparable to Moller-Plesset calculations at lower costs [6].

For the tables, HyOpt is the number of hyperparameter optimizations and HyT is the time taken for hyperparameter optimizations. T indicates time. For the GPRDP the brackets indicated wastage.

DONE Oxadiazole from N₂O and Ethylene

Described in Koistinen et al. [9].

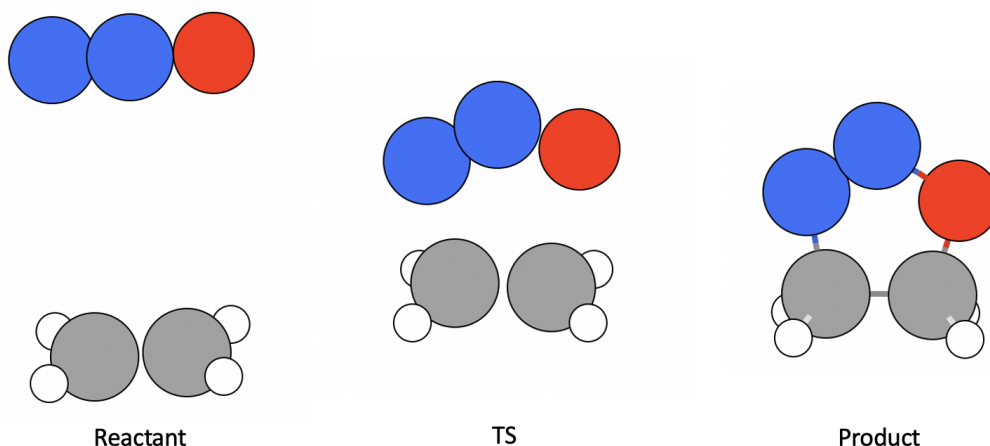


Table 2: B₃LYP-DZ for the Oxadiazole from N₂O and Ethylene. HyOpt is the number of hyperparameter optimizations and HyT is the time taken for hyperparameter optimizations. T indicates time. For the GPRDP the brackets indicated wastage.

Algorithm	Overhead	HyOpt (HyT)	Total T	PES avgT	PES TotT	PES Calls	Cost
Dimer	6512.39	N/A	16449.93	44.966	9937.54	221	144.554
GPR-Dimer	242.266	15 (53.867)	1055.516	45.181	813.25	18	5.456
GPRD-P	209.041	16 (44.875)	1065.221	45.062	856.18	19 (8)	4.672

Clearly, the dimer algorithm is sub-optimal for larger systems, while both the GPRD and the GPRD-P perform well. The GPR-P parameters can be tweaked from the defaults to give better performance. The difference in the GPRD and GPRD-P reflects the relative number of LBFGS rotations taken. Note that the $O(N^3)$ scaling term (HyT) is effectively bound by the GPRD-P algorithm, implying that for larger systems the GPRD-P will outperform the GPRD. The pruning was set to remove a maximum of 4 elements larger than 0.5 every 10 steps.

PM3

To check against the reported values, we also calculated the systems with PM3 [12].

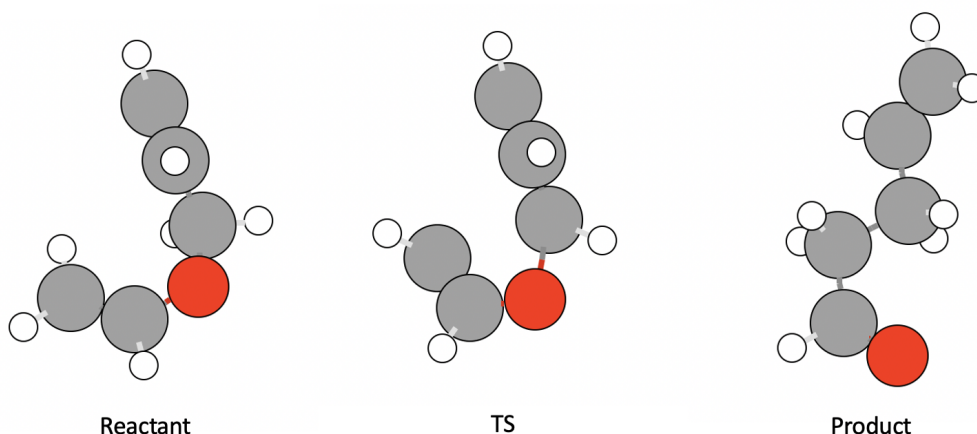
Table 3: PM3 for the Oxadiazole from N₂O and Ethylene. HyOpt is the number of hyperparameter optimizations and HyT is the time taken for hyperparameter optimizations. T indicates time. For the GPRDP the brackets indicated wastage.

Algorithm	Overhead	HyOpt (HyT)	Total T	PES avgT	PES TotT	PES Calls	Cost
Dimer	52.414	N/A	62.394	0.067	9.98	150	787.786
GPR-Dimer	64.379	14 (48.71)	65.599	0.068	1.22	18	949.854
GPRD-P	65.84	15 (48.037)	67.132	0.068	1.292	19 (6)	968.235

Note that now the pruning was set to remove a maximum of 2 elements larger than 0.5 every 10 steps.

Allyl Vinly Ether Rearrangement

Also in Koistinen et al. [9].



The dimer calculation failed to converge after 10 hours and 30 minutes on the cluster; only finding a maximum force of 0.05.

Table 4: B₃LYP-DZ for the Allyl Vinly Ether Rearrangement.

Algorithm	Overhead	HyOpt (HyT)	Total T	PES avgT	PES TotT	PES Calls	Cost
GPRD-Dimer	2324.9	29 (642.244)	5050.9	94	2726.0	29	24.733
GPR-P	10484.867	132 (2604.188)	23268.867	94	12784.0	136 (117)	111.541
GPR-P _{opt}	895.722	37 (390.772)	4749.772	94	3854.0	41 (24)	9.529

In this case, clearly, the pruning caused instabilities; given the complexity of the system; clearly too many values were dropped. The number of samples and the resolution on the GPR

surface are important. The pruning was set to remove a maximum of 4 elements larger than 0.5 every 10 steps.

This was re-run with the pruning set to remove a maximum of 2 elements larger than 0.5 every 10 steps.

PM3

To check against the reported values, we also calculated the systems with PM3 [12].

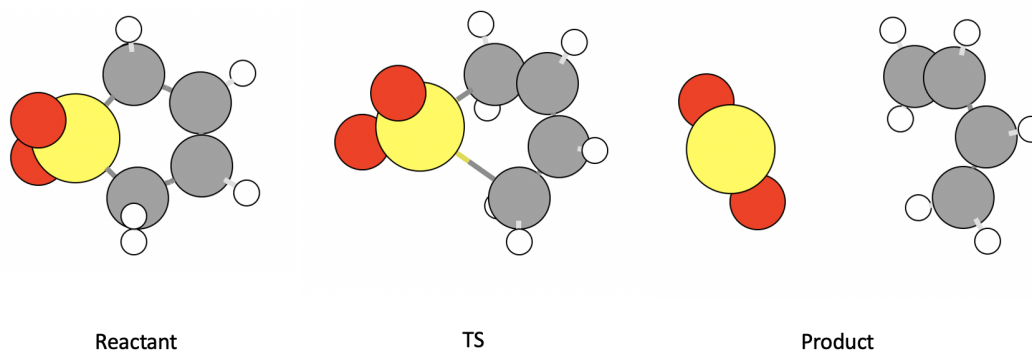
Table 5: PM3 for the Allyl Vinyl Ether Rearrangement. HyOpt is the number of hyperparameter optimizations and HyT is the time taken for hyperparameter optimizations. T indicates time. For the GPRDP the brackets indicated wastage.

Algorithm	Overhead	HyOpt (HyT)	Total T	PES avgT	PES TotT	PES Calls	Cost
Dimer	159.713	N/A	174.993	0.065	15.28	234	2445.867
GPRD-Dimer	348.472	19 (251.507)	349.947	0.065	1.495	23	5361.108
GPR-P	319.302	22 (238.327)	320.992	0.065	1.69	26 (14)	4912.338

Note that now the pruning was set to remove a maximum of 2 elements larger than 0.5 every 10 steps.

Sulfolene from Butadiene and sulfur dioxide

Also in Koistinen et al. [9].



All calculations fail to converge at the B3LYP level in 10 hours and thirty minutes, but the GPR models reach around 0.02 while the dimer reaches around 0.5.

PM3

To check against the reported values, we also calculated the systems with PM3 [12].

Note that now the pruning was set to remove a maximum of 2 elements larger than 0.5 every 10 steps.

Table 6: PM₃ for the Sulfolene from Butadiene and sulfur dioxide. HyOpt is the number of hyperparameter optimizations and HyT is the time taken for hyperparameter optimizations. T indicates time. For the GPRDP the brackets indicated wastage.

Algorithm	Overhead	HyOpt (HyT)	Total T	PES avgT	PES TotT	PES Calls	Cost
Dimer	47.619	N/A	59.419	0.068	11.8	174	702.178
GPRD-Dimer	189.046	14 (152.753)	190.316	0.071	1.27	18	2679.392
GPR-P	198.702	16 (153.707)	200.082	0.069	1.38	20 (8)	2879.739

Analysis

Scaling

The training of the hyperparameters using the methods of Rasmussen and Williams [14] involves the Cholesky decomposition for the calculation of the inverse of a dense covariance matrix. This operation scales as $O(N^3)$. However, we show that there are more considerations, including the time taken by the LBFGS on the GPR dimer; which has a fixed scaling in cost per step (like the EON dimer) for a given system, but is unbound in that the time taken to convergence cannot be estimated.

Pruning

With pruning the covariance matrix size is not allowed to exceed a user-defined size; unless the samples are of a particularly high quality. Pruning is of importance to systems which are larger than those considered here; or for systems which require more iterations.

References

- [1] Vilhjálmur Ásgeirsson and Hannes Jónsson. “Exploring Potential Energy Surfaces with Saddle Point Searches”. In: *Handbook of Materials Modeling*. Ed. by Wanda Andreoni and Sidney Yip. Cham: Springer International Publishing, 2018, pp. 1–26. ISBN: 978-3-319-42913-7. DOI: 10.1007/978-3-319-42913-7_28-1.
- [2] Axel D. Becke. “A New Mixing of Hartree–Fock and Local Density-functional Theories”. In: *The Journal of Chemical Physics* 98.2 (Jan. 15, 1993), pp. 1372–1377. ISSN: 0021-9606, 1089-7690. DOI: 10.1063/1.464304.
- [3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer, 2006. 738 pp. ISBN: 978-0-387-31073-2.
- [4] Zexun Chen and Bo Wang. “How Priors of Initial Hyperparameters Affect Gaussian Process Regression Models”. In: *Neurocomputing* 275 (Jan. 31, 2018), pp. 1702–1710. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.10.028.

- [5] Gareth James et al. *An Introduction to Statistical Learning*. Vol. 103. Springer Texts in Statistics. New York, NY: Springer New York, 2013. ISBN: 978-1-4614-7137-0 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7.
- [6] Jorge Kohanoff. *Electronic Structure Calculations for Solids and Molecules: Theory and Computational Methods*. Cambridge: Cambridge University Press, 2006. ISBN: 978-0-511-75561-3 978-0-521-81591-8 978-1-281-83650-2. URL: <http://www.myilibrary.com?id=183650> (visited on 12/13/2020).
- [7] W. Kohn. “Nobel Lecture: Electronic Structure of Matter—Wave Functions and Density Functionals”. In: *Reviews of Modern Physics* 71.5 (Oct. 1, 1999), pp. 1253–1266. DOI: 10.1103/RevModPhys.71.1253.
- [8] Olli-Pekka Koistinen. “Algorithms for Finding Saddle Points and Minimum Energy Paths Using Gaussian Process Regression”. Aalto University School of Science, 2019. URL: <https://opinvisindi.is/handle/20.500.11815/1460> (visited on 03/02/2020).
- [9] Olli-Pekka Koistinen et al. “Minimum Mode Saddle Point Searches Using Gaussian Process Regression with Inverse-Distance Covariance Function”. In: *Journal of Chemical Theory and Computation* 16.1 (Jan. 14, 2020), pp. 499–509. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b01038.
- [10] Olli-Pekka Koistinen et al. “Nudged Elastic Band Calculations Accelerated with Gaussian Process Regression”. In: *The Journal of Chemical Physics* 147.15 (Sept. 21, 2017), p. 152720. ISSN: 0021-9606. DOI: 10.1063/1.4986787.
- [11] Olli-Pekka Koistinen et al. “Nudged Elastic Band Calculations Accelerated with Gaussian Process Regression Based on Inverse Interatomic Distances”. In: *Journal of Chemical Theory and Computation* 15.12 (Dec. 10, 2019), pp. 6738–6751. ISSN: 1549-9618. DOI: 10.1021/acs.jctc.9b00692.
- [12] Andrew R. Leach. *Molecular Modelling: Principles and Applications*. 2nd ed. Harlow, England ; New York: Prentice Hall, 2001. 744 pp. ISBN: 978-0-582-38210-7.
- [13] Richard McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. 2nd ed. CRC Texts in Statistical Science. Boca Raton: Taylor and Francis, CRC Press, 2020. ISBN: 978-0-367-13991-9.
- [14] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006. 248 pp. ISBN: 978-0-262-18253-9.
- [15] Sambu Seo et al. “Gaussian Process Regression: Active Data Selection and Test Point Rejection”. In: *Mustererkennung 2000*. Ed. by Gerald Sommer, Norbert Krüger, and Christian Perwass. Informatik Aktuell. Berlin, Heidelberg: Springer, 2000, pp. 27–34. ISBN: 978-3-642-59802-9. DOI: 10.1007/978-3-642-59802-9_4.
- [16] Joshua S. Speagle. *A Conceptual Introduction to Markov Chain Monte Carlo Methods*. Mar. 7, 2020. arXiv: 1909.12313 [astro-ph, physics:physics, stat]. URL: <http://arxiv.org/abs/1909.12313> (visited on 06/11/2020).

- [17] Andrew M. Stuart and Aretha L. Teckentrup. “Posterior Consistency for Gaussian Process Approximations of Bayesian Posterior Distributions”. In: *Mathematics of Computation* 87.310 (Aug. 3, 2017), pp. 721–753. ISSN: 0025-5718, 1088-6842. DOI: 10.1090/mcom/3244.

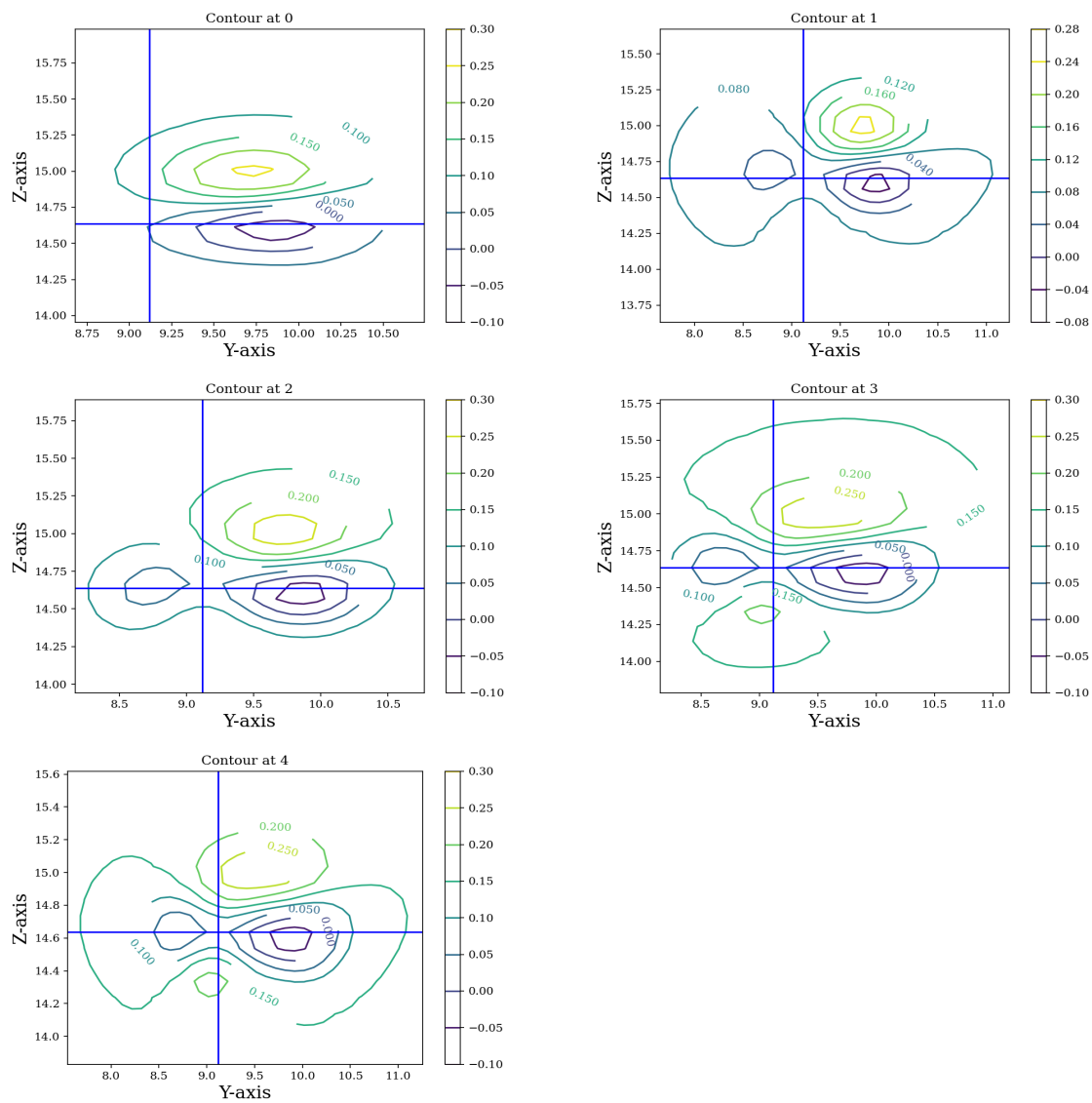


Figure 4: Contour plots representing the GPR PES surfaces predicted for a single Pt adatom on a Pt island with an active radius of 5

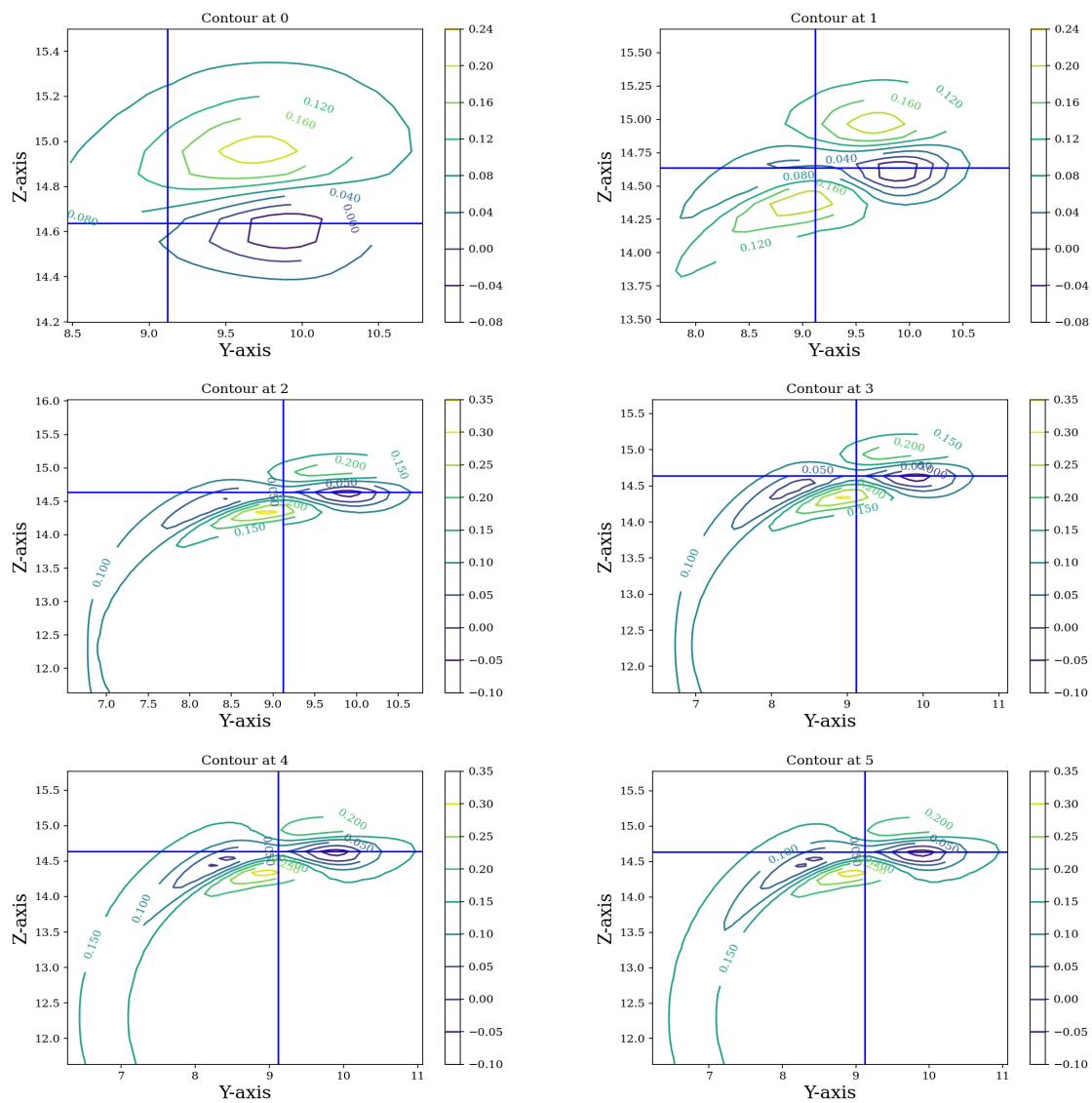


Figure 5: Contour plots representing the GPR PES surfaces predicted for a single Pt adatom on a Pt island with 9 atoms used by the GPR

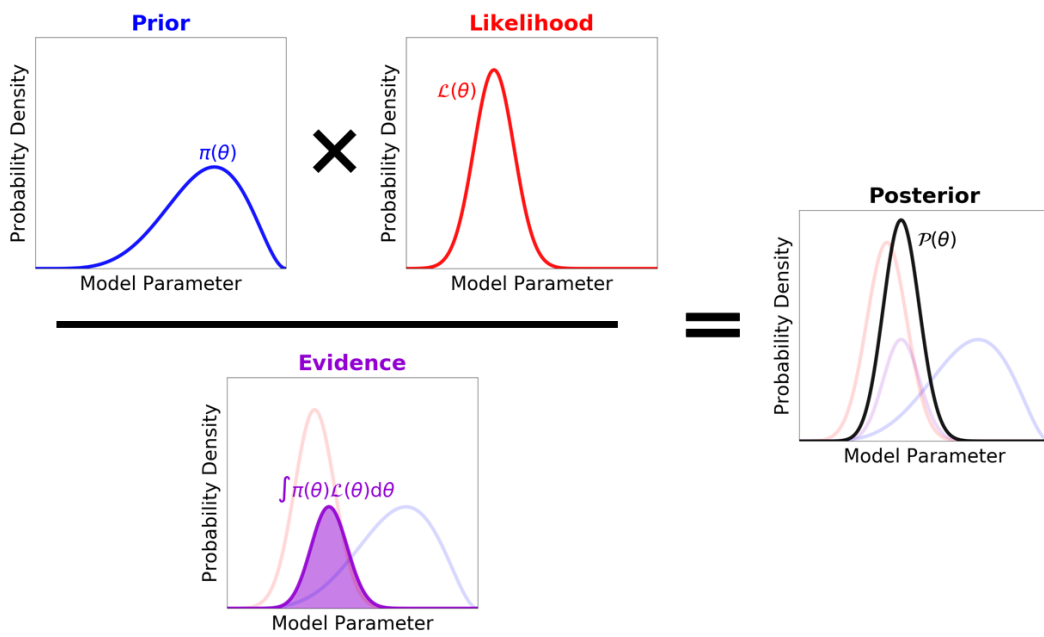


Figure 6: Illustration of Bayes theorem [16]

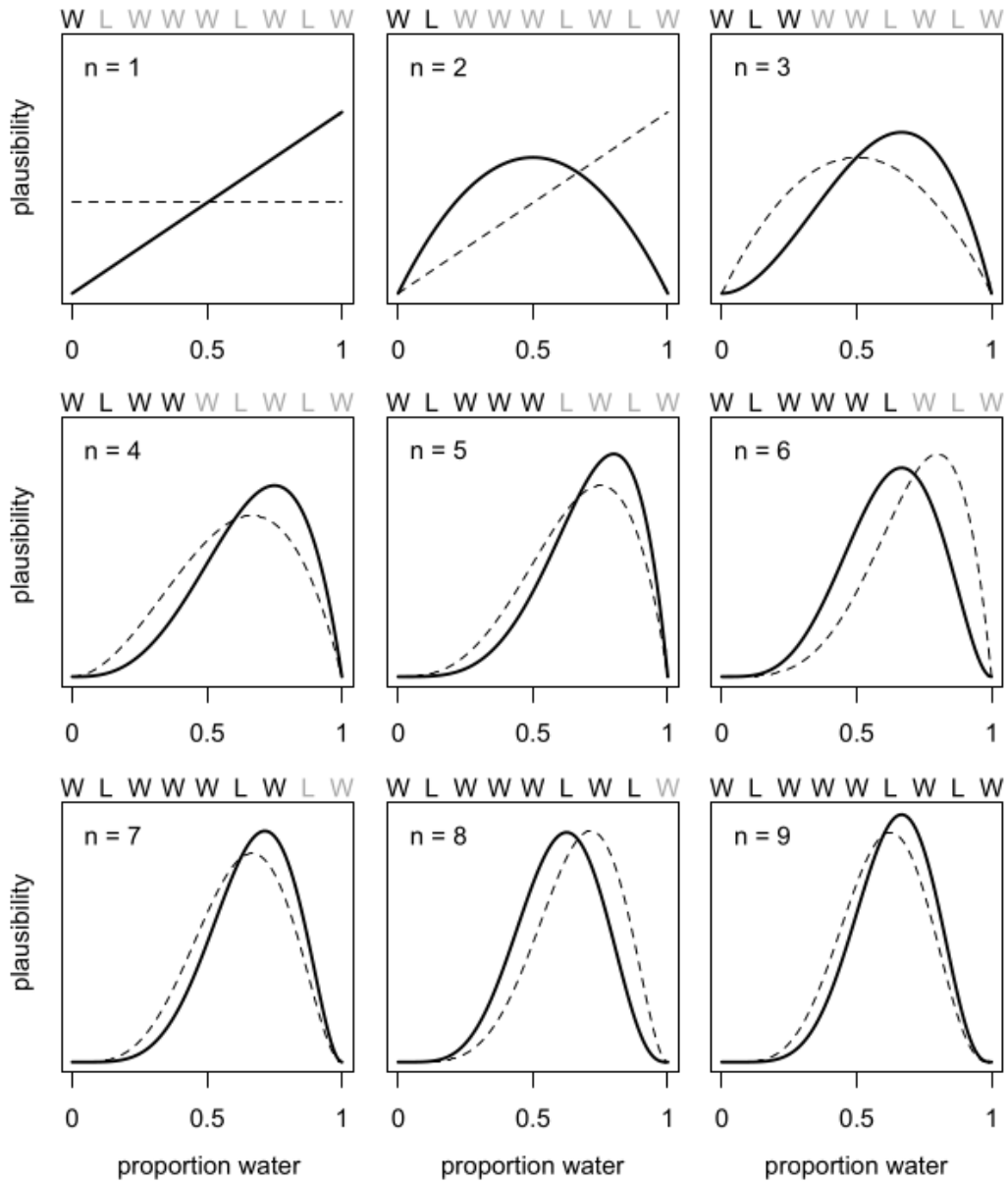


Figure 7: Illustrative effect of iterative modeling from [13]. The lighter dashed line shows the previous model (starting from the uniform prior). Note that importantly, as more data is drawn, the prior becomes less important.

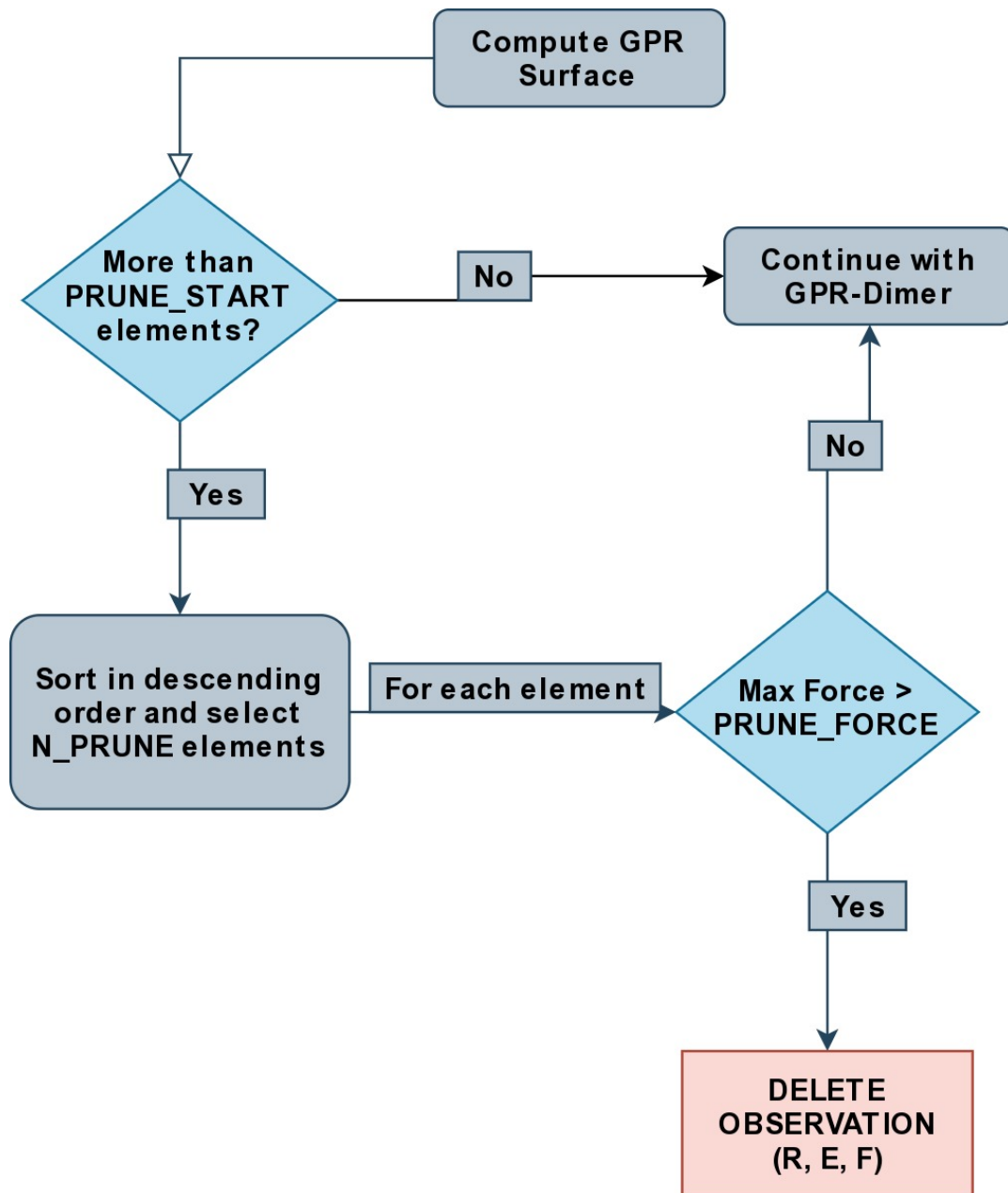


Figure 8: Pruning logic embedded in the GPRD to form GPRD-P; N_{PRUNE} is the (max) number of elements to remove, $\text{PRUNE}_{\text{FORCE}}$ is the force threshold, and $\text{PRUNE}_{\text{START}}$ is the starting number of points;