Efficient Prediction of Quantum Chemical Properties with Multitask Gaussian Process Regression

by

Katharine Fisher

B.S., The University of Texas at Austin (2021)
B.S.A., The University of Texas at Austin (2021)

Submitted to the Center for Computational Science and Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Computational Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Author .....
Center for Computational Science and Engineering
May 24, 2023

Certified by .....
Youssef Marzouk
Professor
Thesis Supervisor

Accepted by .....
Nicolas Hadjiconstantinou
Director, Center for Computational Science and Engineering

Efficient Prediction of Quantum Chemical Properties with Multitask Gaussian Process Regression

by

Katharine Fisher

Submitted to the Center for Computational Science and Engineering
on May 24, 2023, in partial fulfillment of the
requirements for the degree of
Master of Science in Computational Science and Engineering

Abstract

Multitask inference offers an efficient approach to bringing together multiple sources of information to train a surrogate model to predict chemical properties. In this thesis, we explore the task of inferring probability distributions on quantities of interest when we have access to a limited amount of highly accurate CCSD(T) data as well as data obtained using a range of approximations to the exchange-correlation functional in Density Functional Theory (DFT). A CCSD(T) calculation can incur 1000 to one million times the computational cost of a DFT calculation, so an inference model which leverages both types of predictions can benefit from the accuracy of CCSD(T) and the relative efficiency of DFT. We specifically focus on inference methods based on Gaussian process (GP) regression. One example of such an approach, the delta method, uses GP regression to model the difference between two observation data sets, in our case CCSD(T) and DFT. The multitask method, by contrast, models a regression problem for each observation data set and assumes some relationship between the problems so that all relevant data sets can support the primary regression task.

We test the performance of the delta and multitask methods in the tasks of predicting the ionization potential of small organic molecules and the interaction energies of water dimers. The delta method outperforms the multitask approach for data sets where it can be applied, but this approach requires CCSD(T) and DFT data sets to correspond to the same set of molecules and must have access to DFT data for target molecules to make final predictions. The multitask method can use information from CCSD(T) and DFT data sets which correspond to different molecules and can be applied without any DFT insight into the target molecule. For a given training set generation cost, the multitask method produces more accurate predictions than a GP regression model trained only on CCSD(T). The true training set generation cost may be smaller than the listed cost since the flexibility of the multitask method allows it to make use of already existing data sets. Additionally, we find that we can increase accuracy at low computational cost by increasing the number of DFT observation data sets used to inform the model.

Finally, we consider the accuracy of the variances of the distributions predicted by GP inference methods as uncertainty indicators for the models. Though these indicators can capture uncertainty due to limited data set size and extrapolation, they are not designed to capture the impact of the disparity between modeling assumptions and reality. Future work may seek to better understand and represent this reality.

Thesis Supervisor: Youssef Marzouk
Title: Professor

Acknowledgments

Expressing my gratitude with appropriate detail would require at least the space of a chapter. Major thanks to my advisor, Youssef Marzouk, for his guidance and openness to turns this project has taken as well as to Michael Herbst for his significant contributions to this work and insight into the field of chemistry. Thank you to Chenru Duan and Heather Kulik's group for sharing data with me for this work. Additionally, thanks to the National Science Foundation for their support of this work through a Graduate Research Fellowship funded under Grant No. 1745302. Much appreciation is also due to the MIT Uncertainty Quantification group, Jean Sofronas, Kate Nelson, and the MIT libraries.

Thank you to my Mom and Dad for their encouragement, for always being a phone call away, and for all the books. Thanks to my sister. I still want to grow up to be like you. And thanks to my vibrant extended family for their support. This work was written with fond memories of Beatrice Fisher, June and Leonard Meuer, and Jon Miller.

Contents

1 Introduction **17**

2 Application: Quantum Chemistry Predictions **21**

2.1 Kohn Sham Density Functional Theory 21

2.2 The Exchange-Correlation Functional 23

3 Statistical Model: Multitask Gaussian Process Inference **29**

3.1 Bayesian Linear Regression 29

3.2 Gaussian Process Regression 30

3.2.1 The Posterior Distribution 32

3.2.2 Δ Learning 33

3.2.3 Multifidelity Fusion 36

3.2.4 Symmetric Multitasking 39

3.2.5 Asymmetric Multitasking 40

4 Design: Statistical Inference in the Quantum Chemical Setting **45**

4.1 Molecular Features 45

4.2 Kernel Functions 51

4.3 Dataset Construction 55

5 Strengths of the Multitask Approach: Efficient Mean Prediction **61**

5.1 Multitask Method: Organic Molecules Case 62

5.2 Multitask Method: Water Dimers Case 67

5.3 Comparison of the Multitask and Δ Methods 69

6 Challenges for Multitask Approach: Variance Prediction **75**

7 Conclusion **83**

A Feature and Kernel Design **85**

A.1 SOAP Construction 85

A.2 Prediction Calibration of Kernels 86

B Posterior Mean **91**

B.1 Data Set Statistics 91

B.2 Water Dimers Case 93

C Posterior Variance **95**

C.1 Posterior Distribution and Error 95

C.2 Extrapolation 96

List of Figures

2-1 The Jacob's Ladder representation of different classes of density functional approximation [26] 25

3-1 Visualization of observation data obtained by two different prediction methods as well as the Δ between the methods. 34

4-1 Example of inference results for a range of SOAP parameters. The colorbar gives the mean absolute error of predictions made for the ionization potential of small organic molecules. 48

4-2 Correlation between different methods of constructing global features when calculating the distance between molecule pairs. [9] 49

4-3 The performance of GP regression for a polynomial kernel with degree 2 as well as a squared exponential kernel with and without optimized parameters. 54

4-4 Distribution of the number of electrons in molecular configurations used in the organic molecules case study. 56

4-5 **Small Organic Molecules Case.** Example data set structure. The supplemental set, S, contains only DFT predictions, and in this case the S sets for different DFA do not overlap. That is, the predictions for different DFAs are for entirely diferent molecules. The target predictions are on the CCSD(T) level and are highlighted in green. All other predictions are used for training. 57

4-6 **Water Dimers Case.** Example data set structure. Here, the S sets are shown to partially overlap for different levels of theory, indicating that the data set includes predictions by different DFA for some of the same molecules. The testing data set is highlighted in green. 59

5-1 **Organic Molecules Case.** A comparison of different choices of levels of theory as well as the inclusion of different sets (C,S,T) in the training set for the multitask approach. The left plot shows the mean absolute error of an inference model trained with CCSD(T) and PBE produced data, and the right plot shows MAE of a model trained with CCSD(T) and PBE0. The indigo points correspond to inference models trained on only molecules from the core set, the gold points correspond to models trained on both C and S sets, and the teal points correspond to models trained on molecules from the C, S, and T sets. The GP inference results for a training set of CCSD(T) only are plotted as a black line. 64

5-2 **Organic molecules case.** For different numbers of levels of theory (indicated by color), plots show MAE versus cost. The top row corresponds to a fully overlapping S set, the second to a partially overlapping set, and the bottom to a non-overlapping set. 65

5-3 **Water Dimers Case.** Scatter points show the performance of different implementations of the multitask model for a given cost. Color indicates the number of CCSD(T) data points used to train models, and all multitask models are trained with supplemental DFT data. The black line marks the accuracy of a GP model trained only on CCSD(T) for a given cost. All accuracy statistics are based on the average of six tests with different random assignments of dimers to data sets. . . . 68

5-4 **Organic Molecules Case.** Comparison of accuracy of the multitask method and the Δ method applied to the same training data set. The left subfigure compares mean absolute error averaged over three random draws of the C, S, and T sets. Indigo points correspond to training sets with partial overlap of molecules used to train different secondary models in the S set, and gold points correspond to complete overlap in the S set. The right subfigure compares correlation coefficients between predictions and CCSD(T) calculations for the target set. Gold corresponds to Pearson's ρ, indigo to Spearman's ρ, and teal to Kendall's τ. 70

5-5 **Organic Molecules Case.** (a) The scatter points plot the accuracy of a multitask model with a CS secondary training set against the corresponding multitask model which drops the C molecular configurations from the secondary training set. The green line is $x = y$. The black line shows the accuracy of a GP model trained on CCSD(T) at the cost of the multitask model errors reported. (b) The set up of this subfigure is analogous to (a), but a comparison is made between a CST trained model and the corresponding ST trained model. All MAE values are averaged over three random data set assignments. 71

5-6 **Organic Molecules Case.** Comparison of implementations of the Δ method which use a "conventional" model ordering (gold) compared to a scrambled version of the ordering (indigo). The error bars are standard error computed based off three draws of the data set. 73

6-1 **Organic Molecules Case.** Relationship of posterior σ to absolute error of posterior mean. (a) The distribution of Pearson's correlation coefficient between prediction error and posterior σ for various iterations of C, S, and, T. Indigo corresponds to a model trained on CCSD(T) only, gold represents the results of the Δ method, and light blue represents the multitask approach. (b) Posterior σ for target molecular configurations plotted against data. The color scale corresponds to the magnitude of the inner product of the SOAP feature of the molecular configuration. 76

6-2 **Organic Molecules Case.** The left plot shows MAE versus cost for a multitask method, trained on CCSD(T) and PBE. The right plot compares mean posterior σ to cost for the same tests which appear in the left plot. The indigo points correspond to inference models trained on only molecules from the core set, the gold points correspond to models trained on both C and S sets, and the teal points correspond to models trained on molecules from the C, S, and T sets. The GP inference results for a training set of CCSD(T) only are plotted as a black line. 78

6-3 **Water Monomers Case.** Comparison of absolute error and posterior σ for an extrapolation task. Statistical models are trained to predict energy differences between two water monomers and the x axis gives the stretch between monomers for a given prediction task. Training molecules are drawn only from the shaded region of each plot. Indigo points represent the predicted σ, and gold points represent absolute error. 80

A-1 **Organic Molecules Case.** The impact of SOAP parameters on performance of a GP model trained on CCSD(T) data. The color bar reports mean absolute error. 86

A-2 **Organic Molecules Case.** The impact of SOAP parameters on performance of a GP model trained on PBE0 data. The color bar reports mean absolute error. 87
A-3 Comparison of strategies for constructing global SOAP features. The cutoff radius is set to 3 Å. 87
A-4 Comparison of strategies for constructing global SOAP features. The cutoff radius is set to 5 Å. 88
A-5 **Organic Molecules Case.** Comparison between Polynomial and Squared Exponential kernels with respect to the calibration indicator R(p). A result closer to the reference line is better. 89

B-1 **Water Dimers Case.** Scatter points correspond to implementations of the multitask method with a range of data set sizes. The black line shows the accuracy of a GP model trained only on CCSD(T) data. All results are the average MAE values obtained from six random assignments of the data sets. 94

C-1 **Organic Molecules Case.** Mean versus standard deviation predictions of a Gaussian process model. Darker blue points have greater absolute error. 96
C-2 **Water Monomers Case.** GP model error and posterior σ predictions for energy differences of monomer pairs. Training data is drawn from the shaded region. The lowest subfigure shows the true energy differences. 98
C-3 **Water Monomers Case.** Prediction of energy differences of monomer pairs by a GP model trained on pairs exhibiting large monomer stretch. The true energy differences are reported in the bottom subfigure. . . 99

List of Tables

5.1 **Water Dimer Case.** The order of cost for CCSD(T) and DFT computations to generate the training data set 62

B.1 CCSD(T) Data Set Statistics 91

B.2 **Organic Molecules Case.** Correlation coefficients between pairs of observation data sets. 92

B.3 **Water Dimers Case.** Correlation coefficients between pairs of observation data sets. 92

B.4 **Organic Molecules Case.** Summary statistics on the pairwise absolute differences of observation data sets. 92

Chapter 1

Introduction

An abundance of electronics structure methods may be used to calculate properties of a molecule, but it is generally not clear how best to leverage the resulting data for new predictions and to relate these quantities to the truth. Predictions of molecular properties are used in a number of scientific and engineering pursuits including the simulation of molecular dynamics and the screening of new materials for target characteristics. Density Functional Theory (DFT) is an electronics structures method which strikes the right balance between accuracy and computational efficiency for many researchers [34, 26]. To make DFT predictions, an approximation to the exchange-correlation (XC) functional must be selected, but in many cases, the best choice of approximation is unclear and the amount of error that the approximation introduces to subsequent calculations is hard to determine [8, 14]. In complex chemical settings, the ground truth is inaccessible, but practitioners often use DFT in place of ground truth, neglecting uncertainty in the calculations. Additionally, for simulation at the molecular scale and beyond, the cost of DFT becomes prohibitive. This work investigates how surrogate models can draw from DFT and other methods to make accurate predictions under a restricted computational budget and to explore prediction uncertainty.

We focus on using variants of Gaussian process regression to infer probability distributions on quantities of interest (QOI) for target molecular configurations. For each approximation to the XC functional in our data set, as well as for some highly

accurate CCSD(T) calculations, we define a regression task. By imposing structure between these tasks according to a multitask framework, we use the DFT data to support predictions at the level of accuracy of CCSD(T). We demonstrate that the multitask approach outperforms a conventional GP model trained on only CCSD(T) data for the task of predicting ionization potential of small organic molecules as well as the task of predicting the interaction energies of water dimers. Since CCSD(T) calculations can exceed the cost of DFT by factors of 1000 to one million, multitask models can produce significant computational savings. Additionally, because these predictions are made at the level of CCSD(T), they offer an avenue for evaluating or correcting the accuracy of DFT predictions made with various XC approximations. Further, GP regression methods are designed so that in a well-specified setting the variance of a predicted distribution represents the statistical model's certainty. By evaluating how this promise holds up for models trained with quantum chemistry data, we can gain insight into how inference models behave in complicated applications that fall short of the models' underlying assumptions.

Other variants of GP inference have been applied to materials science applications. Bartok et al. introduced Gaussian Approximation Potentials (GAP) which use GP regression to fit potential energy surfaces to energy and force data [3, 1]. Both Pilania et al. and Batra et al. have explored the use of Multifidelity GP models to predict chemical properties based off a training data set informed by high and low fidelity implementations of DFT though they restrict their consideration to two fidelities [28, 5]. Batra et al. also consider the Δ method: using a GP model to predict the difference between two levels of theory, then adding that difference to a lower level prediction for the target to produce a high level prediction. They remark that the data set requirements of the Δ method are rigid, since we must have both low and high level data for each training input and low level data for each target input. By contrast, multitask GP models have the flexibility to relate training data from different levels of theory which do not share the same inputs. It is therefore possible to apply the multitask method to a "dataset of opportunity", by bringing together existing data sets to train the model, rather than expending the computational time

to generate new data.

The outline of this thesis is to explore details of the application before delving into modeling choices and numerical results. Chapter 2 first provides general background for Density Functional Theory before describing the variety of available approximations to the exchange-correlation functional. Chapter 3 focuses on statistical modeling, beginning with conventional frameworks for probabilistic regression, and advancing to variants which incorporate multi-axes data sets. The remaining chapters cover specific applications of statistical models to data from Density Functional Theory approximations. Chapter 4 sets up the main examples of this work by describing how molecules are represented and related to one another and how the training data sets are structured. Finally, in Chapters 5 and 6, we evaluate the accuracy of the means of the distributions predicted by the multitask method and the relationship to model error of the variance of the predicted distributions.

Chapter 2

Application: Quantum Chemistry Predictions

2.1 Kohn Sham Density Functional Theory

Consider a system containing N electrons. The system's ground state energy is the smallest eigenvalue, E, of

$$H\Psi \;=\; E\Psi$$

which is a linear eigenvalue problem. H is the Hamiltonian operator. The solution can be found by minimizing the variational formulation over the space of all 3N dimensional anti-symmetric wave functions, Ψ [34, 32, 13]:

$$E = \min_{\Psi\in\mathbb{R}^{3N},\ \langle\Psi|\Psi\rangle=1} \int \Psi(\boldsymbol{r})\ H\ \Psi(\boldsymbol{r})\ d\boldsymbol{r} \tag{2.1}$$

Unfortunately, the objective of this optimization problem is an integral over $3N$ dimensional space. Even for a relatively small system, this integral can be prohibitively expensive [34, 16]. To borrow an example from [16], consider a system of two silicon atoms, $N = 28$. Even if we restrict to only two quadrature points per dimension and

a leap forward in computing ability makes it possible to evaluate a quadrature point every 1.5 attoseconds, it would still take a year to complete a single computation of the objective function of (2.1). For reference, at peak theoretical performance, the top ranked supercomputer of November 2022 can perform less than two floating point operations every attosecond [18].

By contrast, the ground state energy of a two silicon atom system can be found by Kohn Sham Density Functional Theory (DFT) in under a minute. The Kohn Sham equations are obtained by reformulating the variational principle as the minimization of a functional of the electron density, ρ. Crucially, the problem can be written in terms of N one body wave functions, $\{\psi_i\}_{i=1}^N$, rather than the many body wave function, Ψ [34, 32, 36, 30, 13]. The result is a system of eigenvalue problems

$$\begin{aligned} H^{KS}[\rho]\ \psi_i(r) &= \varepsilon_i\ \psi_i(r), \qquad i = 1, \ldots, N \\ H^{KS}[\rho] &= \Big(-\frac{1}{2}\nabla_r^2 \ +\ V_{ne}[\rho](r) \ +\ V_{Hxc}[\rho](r) \Big) \end{aligned} \tag{2.2}$$

where $H^{KS}[\rho]$ is the Kohn Sham Hamiltonian. Its first additive component is a kinetic energy operator, $V_{ne}[\rho](r)$ represents the nuclei-electron interaction potential, and $V_{Hxc}[\rho](r)$ is the Hartree-exchange-correlation potential. Different approximations to exchange-correlation lead to a wide range of DFT methods.

The set of eigenproblems is nonlinear because the Kohn Sham Hamiltonian depends on the density, ρ, which in turn depends on the eigenfunctions corresponding to occupied orbitals, $\{\psi_i\}_{i=1}^{n_{occ}}$. For simplicity, we assume the Aufbau principle that the occupied orbitals, $1, \ldots, n_{occ}$, correspond to the n_{occ} smallest eigenvalues of the Hamiltonian [34]. The relationship between the eigenfunctions and the electron density is

$$\rho(r) \ = \ 2 \sum_{i=1}^{n_{occ}} |\psi_i(r)|^2 \tag{2.3}$$

where the factor of 2 accounts for spin multiplicity in non-magnetic systems [30].

The nonlinear eigenvalue problem is solved through self-consistent field (SCF)

iteration [34]. We alternate between computing a new approximation for the density from the previous approximation of the potential and obtaining a new approximation of the potential from a previous approximation of the density:

$$V_0 \to \rho_1 \to V_1 \to \rho_2 \to \cdots \to \rho_n \to \ V_n$$

The process ends when the quantities are self consistent—that is, when there is no change in consecutive approximations to the potential. The major computational expense is the solution of a linear eigenvalue problem for a sequence of estimates of the Kohn Sham Hamiltonian. Thus, the computational cost scales cubically with N, the number of electrons in the system. This baseline cost is a major factor in DFT's popularity for electronics structure calculations [34, 36]. By contrast, CCSD(T) calculations—though more accurate—have computational cost $\mathcal{O}(N^7)$ [15]. Recall our example of two silicon atoms. For this system, $N^3 = 21,952$ while $N^7 = 1.3\times10^{10}$.

2.2 The Exchange-Correlation Functional

In this work, we use different implementations of DFT resulting from different approximations to the exchange-correlation functional, $E_{xc}[\rho](r)$. This functional is used to express the Hartree exchange-correlation potential in (2.2):

$$\begin{aligned} V_{Hxc}[\rho](r) &= \frac{\delta E_{Hxc}[\rho](r)}{\delta\rho(r)} \\ E_{Hxc}[\rho](r) &= E_H[\rho](r) + E_{xc}[\rho](r) \end{aligned}$$

The exchange-correlation functional represents the many body effects that one body wave functions are not capable of capturing, but the true form of this functional is unknown [23, 26]. Hundreds of approximations to the functional have been developed, through both empirical fitting techniques and the use of asymptotic physical constraints [32]. For convenience, the density functional approximations (DFAs) are often sorted according to their inputs. This framework has been christened "Ja-

cob's Ladder" because each step up the ladder corresponds to a class of DFAs with an additional input type, and we have hope that once we reach the top of the ladder we can make accurate chemical predictions [26]. That is to say, we will be in Heaven. Figure 2-1 shows the framework—the rungs are labeled with the new input information incorporated on each level, and the arrows label the DFA type.

The next two paragraphs will provide a simplified overview of the functional approximation types represented by Jacob's Ladder. Local-Density Approximations (LDAs) use local information about the electron density, $\rho(r)$, to approximate the exchange-correlation energy density of our system with the same value for a Uniform Electron Gas which also has density $\rho(r)$. These methods have no mechanism to capture dispersion interactions, but favorable error cancellation allows them to perform surprisingly well for solid-state systems [32, 23]. The next two rungs—corresponding to Generalized Gradient Approximations (GGAs) and meta-GGAs—add semi-local information through $\nabla\rho(r)$ and $\nabla^2\rho(r)$, respectively. Both classes improve on LDAs, in general, though GGAs remain vulnerable to self-interaction and static-correlation errors. Meta-GGAs can be designed so that self-interaction errors vanish in the representation of exchange, but these errors remain an issue for the correlation representation [32].

The fourth and fifth rungs both aim to minimize self-interaction error by introducing other electronic structure methods to balance local and non local input from the electron density. On rung four, single hybrid approximations are constructed by mixing Hartree-Fock (HF) exchange energy with GGA or meta GGA exchange representations to increase local emphasis on the $\rho(r)$. This approach is equivalent to adding occupied orbitals, $\{\psi_i^{occupied}\}$, as inputs to our approximation. The fraction of HF exchange energy that can be used in a single hybrid approximations is limited. Thus, on rung five, double hybrid approximations take the additional step of mixing second-order Møller–Plesset (MP2) correlation energy with GGA or meta GGA correlation energy, yielding an approximation in terms of occupied and virtual orbitals. This technique makes it possible to increase the fraction of HF energy mixed with the exchange energy. Both classes of hybrid approximations make large steps in

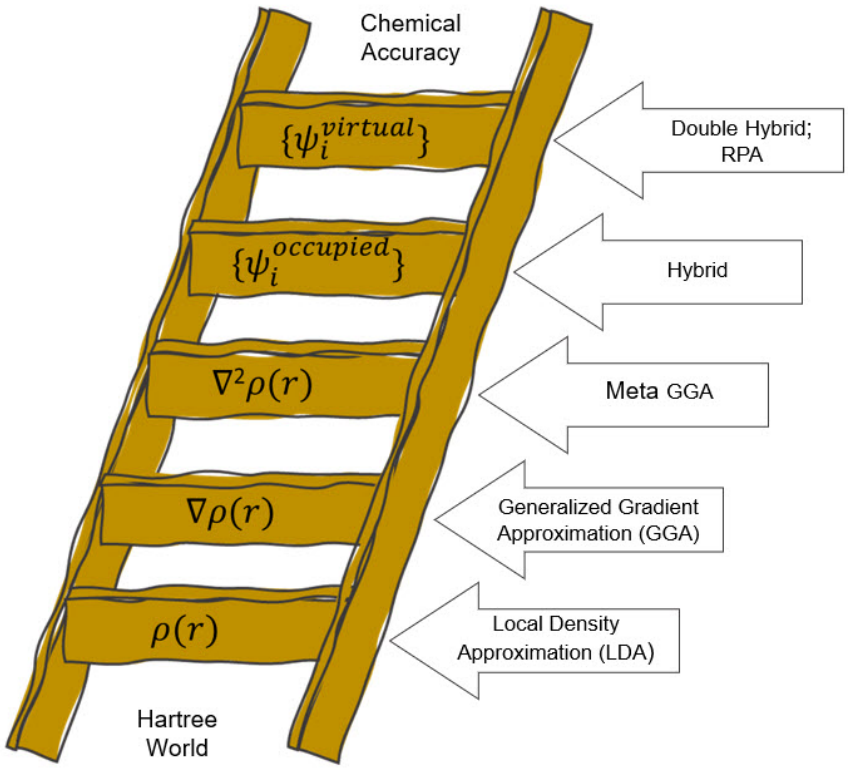


Figure 2-1: The Jacob's Ladder representation of different classes of density functional approximation [26]

reducing self-interaction error at the cost of increased risk of static-correlation errors when representing systems with small HOMO-LUMO gaps [32].

Because approximation formulations become more complex as we move up Jacob's ladder, higher rung DFAs are more computationally expensive than their lower rung counterparts, but accuracy trends are more complicated. It can be hazardous to generalize about the performance of density functional approximations. While for a wide range of benchmarks, the average performance of a class of DFAs is better the higher up that class sits on the ladder, the story for individual DFAs within those classes can be different [14]. As the previous summary of DFA classes demonstrates, trade offs are made when constructing and fitting approximations. DFAs will generally perform well for some categories of molecular systems and poorly for others [14].

Consequently, ranking the applicability of DFAs to a given problem setting is nontrivial. Civalleri et al. consider ten statistics for evaluating the performance of

DFAs and Hartree Fock on bandgap prediction and find ten different recommendations for the best approach. These "best" recommendations include Hartree Fock and representatives from rungs two, four, and five of Jacob's Ladder. Furthermore, some DFAs which give the best result for one statistic demonstrate mediocre performance for another [8]. Researchers commonly select DFAs that are popularly used or that perform well on benchmarks for a test quantity of interest, traditionally atomization energy. Unfortunately, Goerigkab and Grimme report that B3LYP—a popular hybrid DFA—had worse average performance than all 22 other hybrid DFAs considered for a wide range of chemically relevant benchmarks. They also find that the performance of DFAs on atomization energy correlates only weakly with performance on other quantities of interest [14].

There is significant need both for guidance on which DFAs are most appropriate to a given problem and for representation of the uncertainty these approximations introduce to DFT calculations. Lejaeghere reports that error introduced by the choice of density functional approximation typically exceeds error from numerical approximations in DFT by an order of magnitude [23]. Some approaches to reducing error and modeling uncertainty have been proposed. Lejaeghere describes an approach that distinguishes between systematic, correctable error and random error. The relationship between experimental data and DFT predictions is modeled by a linear regression function with stochastic additive noise. The regression coefficients and noise variance are iteratively fit with a weighted least squares algorithm and used to represent systematic and random error, respectively [23]. This approach is limited by the simplicity of the linear regression function and the need for sufficient experimental data.

The Bayesian Error Estimation Functional (BEEF) is another approach to uncertainty estimation which begins by fitting a new DFA through a semi-empirical procedure [24]. In existing work, the structure of the model has been related to a GGA and the parameters are estimated by minimizing a cost function informed by reference datasets. Once the optimal parameters are identified, a probability distribution on the model parameters is constructed to represent the inadequacy of the DFA

model in representing the reference data set. BEEF error estimation aims to assume as little as possible about this distribution and therefore supposes that it is the distribution which maximizes the entropy: the Boltzmann distribution. The temperature of this distribution is chosen so that its variance will be equal to the average deviation of the BEEF functional approximation predictions from the reference data [7, 24, 33]. When using multiple data sets to fit the BEEF DFA, this constraint can be satisfied only approximately. Furthermore, this approach only provides uncertainty information for the BEEF functional approximation, so there remains a need to investigate the uncertainty introduced by other choices of density functional approximation.

The work presented in this thesis is part of an investigation into the use of an ensemble of DFT predictions made using different functional approximations to inform probabilistic models with the goal of improving prediction accuracy and representing uncertainty under a limited budget.

Chapter 3

Statistical Model: Multitask Gaussian Process Inference

3.1 Bayesian Linear Regression

Consider a multivariate linear regression function

$$f(\mathbf{X}) \;=\; \boldsymbol{\beta}^T \mathbf{X}$$

We wish to learn distributions on the model coefficients from data. Noisy observations of $f(\mathbf{X})$ are related to the covariates through the expression

$$\begin{aligned} Y_i \;&=\; \sum_h \beta_h X_{ih} + \varepsilon_i \\ \varepsilon_i \;&\sim\; \mathcal{N}\Big(0,\; \sigma^2\Big) \end{aligned} \tag{3.1}$$

Above, we have assumed that each observation's noise is additive and follows a centered normal distribution with variance σ^2. We also assume that the noise of a given observation is not dependent on the noise of any other. Thus, $\varepsilon_1, \ldots, \varepsilon_n$ are iid, independent and identically distributed. Because we also assume each X_{ih} is a deterministic quantity, the observations are independent of each other, and we write their

likelihood function as

$$\mathbf{Y}\big|\mathbb{X}, \boldsymbol{\beta}, \sigma \sim \mathcal{N}\Big(\sum_h \beta_h \mathbf{X}_h,\ \sigma^2\mathbb{I}\Big) \ \equiv\ \mathbb{P}\big(\mathbf{Y}\big|\mathbb{X}, \boldsymbol{\theta}\big)$$

where $\mathbb{I}$ is the identity matrix, $\mathbb{X}$ is the design matrix with X_{ih} at the i^{th} row and h^{th} column, and $\boldsymbol{\theta} = [\boldsymbol{\beta}, \sigma]^T$ are the model parameters. We can use our existing knowledge of the parameters to choose a prior distribution, $\mathbb{P}(\boldsymbol{\theta})$, and make use of Bayes' Law:

$$\mathbb{P}\big(\boldsymbol{\theta}|\mathbf{Y}, \mathbb{X}\big) \ =\ \frac{\mathbb{P}\big(\mathbf{Y}|\mathbb{X}, \boldsymbol{\theta}\big)\mathbb{P}(\boldsymbol{\theta})}{\mathbb{P}\big(\mathbf{Y}|\mathbb{X}\big)}$$

We obtain a distribution on our model parameters conditioned on our observations. Provided that our assumptions are appropriate to our application, this distribution can be used to construct an approximation $\hat{f}(X; \hat{\boldsymbol{\beta}})$ and to characterize its uncertainty.

3.2 Gaussian Process Regression

Gaussian process (GP) regression is an example of nonparametric regression which shifts the focus from inferring a probability distribution on the finite parameters of a regression function to inferring a distribution on regression functions. As before, we have a likelihood distribution on our observations though we now choose a prior distribution for our regression function. We treat the regression function as infinite realizations of a random variable, $f(\mathbf{X})$. In GP regression, these variables are realizations of a Gaussian process, so every finite sample, $\{f(\mathbf{X}_i)\}_i^n$, is a multivariate Gaussian random variable. Our selection of a prior covariance function for this random variable controls the smoothness of the regression function and encapsulates our assumptions about the relationships between any $f(\mathbf{X}_i)$ and $f(\mathbf{X}_j)$ [29].

We are interested in a latent function, f, which relates each $\mathbf{X}_i$ to a noisy observation Y_i. When we assume independent and homoscedastic Gaussian noise, we have the model

$$
\begin{aligned}
Y_i &= f(\mathbf{X}_i) + \varepsilon_i \\
\varepsilon_i &\sim \mathcal{N}\Big(0,\ \sigma^2\Big)
\end{aligned}
$$

For each observation, Y_i, there is some corresponding function value $f(\mathbf{X}_i)$. We collect these in the vector $\mathbf{f}$. We are interested in inferring the values the function takes on at target $\mathbf{X}_*$, and we collect these values in $\mathbf{f}_*$. Using this notation, we assign a joint Gaussian process prior

$$
f(\mathbf{X}) \sim \mathcal{GP}\Big(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X}')\Big)
$$

The method can be extended to cases with nonzero mean. For the above set up, the joint prior of the function values for the observations and targets is

$$
\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{\mu},\ \begin{bmatrix} K_{ff} & K_{f*} \\ K_{f*}^T & K_{**} \end{bmatrix}\right) \tag{3.2}
$$

and the likelihood distribution is

$$
\mathbf{Y}\big|\mathbf{f} \sim \mathcal{N}\big(\mathbf{f},\ \sigma^2\mathbb{I}\big)
$$

We apply Bayes' law to find the joint posterior distribution

$$
\mathbb{P}(\mathbf{f}, \mathbf{f}_* \,\big|\mathbf{Y}) = \frac{\mathbb{P}(\mathbf{f}, \mathbf{f}_*)\mathbb{P}(\mathbf{Y} \,\big|\mathbf{f})}{\mathbb{P}(\mathbf{Y})} \tag{3.3}
$$

and marginalize to find the distribution for the target function realizations

$$
\begin{aligned}
\mathbb{P}(\mathbf{f}_* \,\big|\mathbf{Y}) &= \int \mathbb{P}(\mathbf{f}, \mathbf{f}_* \,\big|\mathbf{Y}) d\mathbf{f} \\
&= \frac{1}{\mathbb{P}(\mathbf{Y})} \int \mathbb{P}(\mathbf{f}, \mathbf{f}_*)\mathbb{P}(\mathbf{Y} \,\big|\mathbf{f}) d\mathbf{f} \\
&= \mathcal{N}\Big(\boldsymbol{\mu} + K_{f*}^T(K_{ff} + \sigma^2\mathbb{I})^{-1}\big(\mathbf{Y} - \boldsymbol{\mu}\big),\quad K_{**} - K_{f*}^T(K_{ff} + \sigma^2\mathbb{I})^{-1}K_{f*}\Big)
\end{aligned} \tag{3.4}
$$

Thus, GP regression with a Gaussian likelihood has an analytical solution. The next subsection will discuss this distribution in more detail. Then, the following several subsections will describe modifications to this approach which allow us to incorporate observations generated by multiple methods, perhaps with varying levels of accuracy.

3.2.1 The Posterior Distribution

The interpretation of the posterior distribution strongly relies on our assumption that our choice of kernel function and features accurately represent the relationships between observations and targets. For example, to obtain the posterior mean, we derive a correction to the prior mean as the linear combination of each observation's deviance from the prior mean, weighted by how strongly the observation is correlated to the value we want to predict. Note that the deviation of the observations from the prior mean is modeled as

$$\big(\mathbf{Y} - \boldsymbol{\mu}\big) \sim \mathcal{N}\Big(\mathbf{0}, K_{ff} + \sigma^2 \mathbb{I}\Big)$$

Consider the Cholesky factorization of the covariance: $K_{ff} + \sigma^2\mathbb{I} = R^T R$. The deviation of the observations can be rescaled and corrected for correlation

$$R^{-T}\big(\mathbf{Y} - \boldsymbol{\mu}\big) \sim \mathcal{N}\big(0, \mathbb{I}\big)$$

Similarly, the product

$$K_{f*}^T R^{-1}$$

is a rescaling of the estimated covariance between the target and the observations. We can say that these covariances are estimates of the relevance of each observation to our predictions. Once these estimates are in the same coordinate system as the deviation of the observation from the prior mean, we can calculate the correction to the prior mean.

$$\begin{aligned} \boldsymbol{\mu} \ &+ \ K_{f*}^T (K_{ff} + \sigma^2 \mathbb{I})^{-1} \big(\mathbf{Y} - \boldsymbol{\mu}\big) \\ = \boldsymbol{\mu} \ &+ \ \Big[K_{f*}^T R^{-1}\Big] \ \Big[R^{-T} \big(\mathbf{Y} - \boldsymbol{\mu}\big)\Big] \end{aligned}$$

Likewise, we derive a correction to our prior estimate of the target's covariance by determining how informative our observations are about the target values in the context of the spread of the observation data.

The target's covariance is initially estimated as K_{**}. Inference corrects this estimate by subtracting $K_{f*}^T (K_{ff} + \sigma^2 \mathbb{I})^{-1} K_{f*}$ from K_{**}. This correction is the square of the Mahalanobis distance of the point K_{f*} from the origin in a coordinate system defined by the eigendecomposition of $K_{ff} + \sigma^2 \mathbb{I}$.

In one dimension, $K_{ff} + \sigma^2 \mathbb{I}$ is a scalar. The correction to K_{**} reduces to the covariance between $\boldsymbol{f}_*$ and $\mathbf{Y}$ divided by the standard deviation of $\mathbf{Y}$. Analogously, in multiple dimensions, the factor $(K_{ff} + \sigma^2 \mathbb{I})^{-1}$ functions to correct the covariance between the target and the observations for correlation between the observations and to rescale by the standard deviation in each direction. Thus, we obtain a sense of whether K_{f*} is typical or unusual with respect to the spread of the observation data.

In the next subsections, we will investigate different ways of modeling the relationships between observations and targets within a Gaussian process regression approach.

3.2.2 Δ Learning

The profusion of approximations to the exchange-correlation functional beg some consideration over the best choice of observation set $\{Y_i\}_{i=1}^n$ to inform a Gaussian process regression model. For example, we may opt to construct our observation set using the popular and relatively cheap PBE approximation or we may expend the computational resources to predict with a high rung, double hybrid approximation. Ultimately, though, we are interested in a method that can use both the PBE and the double hybrid data, as well as higher level information from CCSD(T).

We must determine some approach to leverage a multi-axis observation set, $\{Y_{ij}\}_{i=1,j=1}^{n,m}$, where we have data for molecular configurations $i = 1, \ldots, n$ and electronics structure methods $j = 1, \ldots, m$. A simple "pooling" approach may put all observations into one vector and proceed with inference via Equation (3.4). If we allow both $\mathbf{Y}_{ij_1}$ and $\mathbf{Y}_{ij_2}$ to share feature $\mathbf{X}_i$, the Gaussian process model will suffer from a low rank covariance matrix. To avoid this problem, the features must have some dependence on j. Thus, if we have a DFT prediction and a CCSD(T) prediction for the same molecule, we must define a feature that reflects both the molecule and the prediction method. This approach would require a model for how differences in electronic structures methods map to differences in their predictions. Alternatively, we may directly model the difference between these predictions.

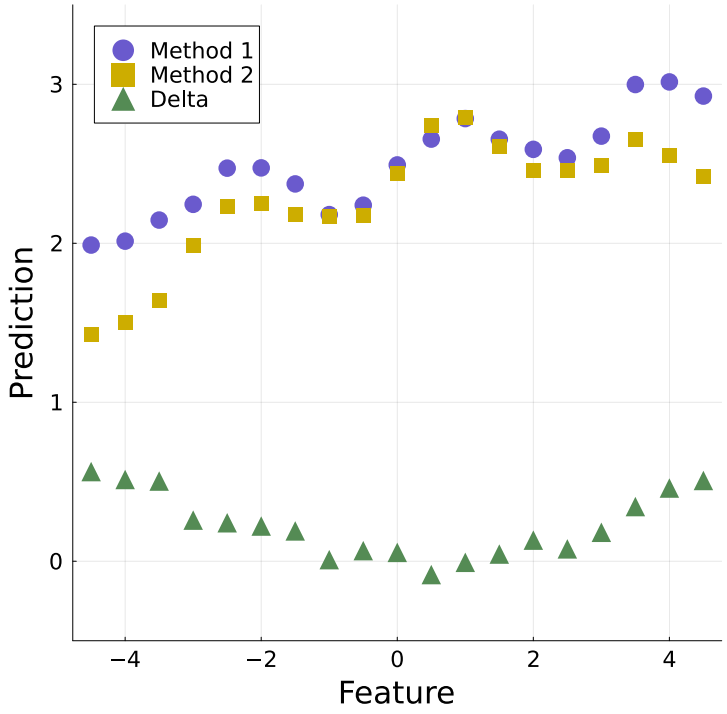


Figure 3-1: Visualization of observation data obtained by two different prediction methods as well as the Δ between the methods.

One approach—the Δ method—performs Gaussian process regression to predict differences between observations, Δ_i, obtained with two different methods for a given $\mathbf{X}_i$. Figure 3-1 provides a visual of observations obtained from two prediction methods

and the Δ_i between each pair. If we retain our assumption that noise follows an i.i.d. Gaussian distribution, we have the model

$$\begin{aligned} Y_{ij_1} - Y_{ij_2} \;\equiv\; \Delta_i \;&=\; f\big(\mathbf{X}_i\big) \;+\; \varepsilon \\ \varepsilon \;&\sim\; \mathcal{N}\big(0, \sigma^2\big) \\ f \;&\sim\; \mathcal{GP}\big(\boldsymbol{\mu}_\Delta,\; k_\Delta(\mathbf{X}, \mathbf{X}')\big) \end{aligned}$$

where j_1 and j_2 indicate two prediction methods. An application of (3.4) yields a distribution on the regression function for Δ. Suppose our ultimate interest is to predict $\mathbf{Y}$ at $\mathbf{X}_{i_*}$. Within the Δ framework, we can supply an observation from one method, say $Y_{i_* j_2}$, in order to make a prediction for the value from the other method:

$$Y_{i_* j_1} \;\approx\; \Delta_i \;+\; Y_{i_* j_2}$$

Thus, the Δ method is suited to settings where calculations by method j_1 are both more accurate and more time consuming than calculations that use j_2. Then, for a given computational budget, we are able to inform our statistical model with a larger data set than would be possible if we applied conventional Gaussian process regression only to observation data from method j_1. Furthermore, this approach has an advantage when the true Δ has smaller average magnitude than the raw observations or the observation data sets have common modes of variation which cancel under subtraction. In these cases, use of the Δ method with relatively little data can demonstrate lower absolute error than some more data rich implementations of Gaussian process regression for a high fidelity prediction method.

For electronics structure calculation methods, there is generally a clear hierarchy in cost, but there is only a clear hierarchy in model fidelity for select chemical and numerical settings. For instance, Batra et al. consider the Δ method when fitting a model for dopant formation energy of hafnia with high and low fidelity data sets constructed from DFT at different numerical settings. Bartók et al. use a version of the Δ method to learn one- and two-body corrections for systems of water molecules

[2]. Their two-body correction is informed by three levels of electronic structure theory. The Δ method can be extended to make use of more than two prediction methods by choosing some baseline method, j_k and fitting multiple Δ models, each stepping between a pair of prediction methods, which sum to a correction for j_k to the highest fidelity level, j_1. The final prediction is

$$Y_{i*j_1} \;\approx\; Y_{i*j_k} \;+\; \Delta_{i*}(j_k \to j_{k-1}) \;+\; \ldots \;+\; \Delta_{i*}(j_2 \to j_1)$$

The approach is most appropriate to cases where there is a clear ordering of the prediction methods.

3.2.3 Multifidelity Fusion

Models of disparity are useful tools for leveraging multiple observation data sets to train a Gaussian process regression model. The version of this approach provided by the Δ method can be limited by its inability to use an observation from one set, Y_{ij_1}, unless we also have an observation from the second set, Y_{ij_2}, which corresponds to the same $\mathbf{X}_i$. Thus, to use a DFT prediction for a molecule $\mathbf{X}_i$ to train a Δ model between CCSD(T) and DFT, we also require a CCSD(T) prediction for the same molecule. We may consider alternative approaches that model disparities but allow us to use all available observations. One such approach is multifidelity Gaussian process inference, introduced by Kennedy and O'Hagan [19].

This approach defines two latent functions: f_p—representing the predictions from some high fidelity model—and f_s—representing a low fidelity model. We assume the relationships

$$\begin{aligned} Y_{ip} \;&=\; f_p\big(\mathbf{X}_i\big) \;+\varepsilon_{pi} \;=\; \rho f_s\big(\mathbf{X}_i\big) \;+\; \delta_{ps} \;+\; \varepsilon_{pi} \\ Y_{is} \;&=\; f_s\big(\mathbf{X}_i\big) \;+\; \varepsilon_{si} \end{aligned} \tag{3.5}$$

The term δ_{ps} captures the difference between the low fidelity model, scaled by parameter ρ, and the high fidelity model. This disparity is endowed with a Gaussian

process prior, as is the latent function, f_s:

$$\begin{aligned}\delta_{ps}(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_\delta(\mathbf{X}),\ k^\delta(\mathbf{X},\mathbf{X}')\Big)\\ f_s(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_s(\mathbf{X}),\ k^s(\mathbf{X},\mathbf{X}')\Big)\end{aligned}$$

As before, we assume the additive noise terms, ε_{ip} or ε_{is}, are iid and are drawn from a centered Gaussian distribution. Now, suppose we want to predict our targets $\mathbf{f}_*$ at the high fidelity level. We can determine that the joint distribution of our observations and targets is

$$\begin{bmatrix}\mathbf{Y}_p\\ \mathbf{Y}_s\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0},\ \begin{bmatrix}\rho^2K^s_{pp}+K^\delta_{pp}+\sigma^2\mathbb{I} & \rho K^s_{ps} & \rho^2K^s_{p*}+K^\delta_{p*}\\ \rho K^s_{sp} & K^s_{ss}+\sigma^2\mathbb{I} & \rho K^s_{s*}\\ \rho^2K^s_{*p}+K^\delta_{*p} & \rho K^s_{*s} & \rho^2K^s_{**}+K^\delta_{**}\end{bmatrix}\right)$$

where σ^2 is the variance of the noise distribution. The superscript of each covariance matrix, K, indicates which kernel function was used in its construction, and the subscripts indicate the pairs of features that are compared. Bayesian inference yields an analytical posterior distribution, analogous to (3.4).

In the realm of DFT predictions, the Pilania et al. apply this multifidelity approach to predict bandgaps of elpasolite compounds [28]. For their high fidelity model, they consider DFT with a hybrid functional approximation, HSE06, while for their low fidelity model, they use a Generalized Gradient Approximation, PBE. They report that the prediction accuracy of the multifidelity approach is comparable to DFT with HSE06, and that performance improves when they add new training data to the high fidelity set as well as when they add new data to the low level set [28]. In their investigation of dopant formation energies in hafnia, Batra et al. find similar results

for the multifidelity approach and recommend it for its flexibility compared to the Δ method [5].

To incorporate more than two levels in our training data, we may consider performing sequential Gaussian process regression to map one observation data set to the next, eventually producing a prediction for the primary observations. This approach is based off the modification to Deep GP introduced by [27] and further discussed in [21]. These works aim to combine predictions from models of increasing fidelity to estimate predictions from the highest fidelity model. Like the Δ method, this approach assumes some ordering on the available observation data sets.

Deep GP methods compose multiple Gaussian process priors. Suppose the available models are indexed $j = 1, \ldots, m$ with 1 indicating the highest fidelity, and the prediction by the j^{th} model corresponding to a feature $\mathbf{X}$ is $f_j(\mathbf{X})$. The Deep GP approach to multifidelity information fusion is

$$\begin{aligned} f_j(\mathbf{X}) &= \rho_{j+1} f_{j+1}(\mathbf{X}) + \delta_{j+1}(\mathbf{X}) \\ \delta_j(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_{\delta_j}(\mathbf{X}),\ k_{\delta_j}(\mathbf{X}, \mathbf{X}')\Big) \\ f_m(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_m(\mathbf{X}),\ k_m(\mathbf{X}, \mathbf{X}')\Big) \end{aligned}$$

Computation of the covariance matrix to make predictions for the highest fidelity requires information from models on all fidelity levels and can be expensive. Perdikaris et al. modify the approach by using the posterior predictions of level $j+1$ to inform the predictions on level j [27]. The revised model is formulated

$$\begin{aligned} f_j(\mathbf{X}) &= g_j\Big(\mathbf{X}, f_{j+1}(\mathbf{X})\Big) \\ g_j &\sim \mathcal{GP}\Big(\mathbf{0}, \quad \gamma_{t_\rho}(\mathbf{X}, \mathbf{X}')\gamma_{t_f}\big(\ f_{j+1}(\mathbf{X}), f_{j+1}(\mathbf{X})'\big) + \gamma_{t_\delta}(\mathbf{X}, \mathbf{X}')\ \Big) \\ f_m(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_m(\mathbf{X}),\ k_m(\mathbf{X}, \mathbf{X}')\Big) \end{aligned}$$

and allows for nonlinear mappings between predictions at level $j+1$ and j. Each kernel function which contributes to the covariance matrix has a unique set of param-

eters. An analytical solution will generally be intractable, and [27] and [21] suggest approximating the posterior with Monte Carlo integration.

3.2.4 Symmetric Multitasking

"Multitasking" refers to a category of methods which consider several Gaussian process regression tasks and assume some relationship between these tasks. As with the Δ method or multifidelity fusion, multitasking can allow us to incorporate a larger dataset into our inference problem without just pooling the data into one observation vector. For instance, we can define a regression problem trained on CCSD(T) data as well as one trained on each DFA we consider. By modeling correlation between regression functions, we can use the DFA regression tasks to support prediction in the CCSD(T) task.

In this subsection, we will describe a "symmetric" approach to multitask regression due to Bonilla et al., and the next section will cover the "asymmetric" approach of Leen et al. [6, 22]. In general, the formulations assume that there are m tasks, and for each task, j, we have observational data, $\mathbf{Y}_j \in \mathbb{R}^n$. In practice, the method can be modified, so that different input sets of different sizes are used for different tasks.

Suppose that the observations for task j can be modeled as

$$\begin{aligned} Y_{ij} &= f_j(\boldsymbol{X}_{ij}) + \varepsilon_{ij} \\ \varepsilon_{ij} &\sim \mathcal{N}\Big(\mathbf{0},\ \sigma_{ij}^2\Big) \end{aligned}$$

We assume that the data for all tasks can be described by a multivariate normal distribution defined by the Gaussian process prior

$$f_j \sim \mathcal{GP}\Big(\boldsymbol{\mu},\ \ k_i\big(\mathbf{X}_i,\ \mathbf{X}_i'\big)\ k_j\big(\mathbf{X}_j, \mathbf{X}_j'\big)\Big)$$

The covariance between functions for two tasks is the product of an input specific component, $k_i(\cdot,\cdot)$, and a task specific component, $k_j(\cdot,\cdot)$. We let $\mathbf{X}_i$ be the entries of $\mathbf{X}_{ij}$ which identify the input and $\mathbf{X}_j$ be the task specific entries. Typically, the input

kernel will be a Squared Exponential kernel or a close variant. Bonilla et al. model the task kernel as the free form matrix $\mathbb{K}^{task}$ so that $\mathbb{K}^{task}_{ab} = k_j(\mathbf{X}_a, \mathbf{X}_b)$ and learn each entry from data [6]. They note that this matrix could also be used to represent known similarity relationships between classes.

Observational noise is taken into account as

$$Y_{ij} \sim \mathcal{N}\Big(f_j(\mathbf{X}_i),\ \sigma^2_{ij}\Big)$$

and the mean of the posterior prediction for a target * and given task j is

$$\begin{aligned} \hat{f}_j(\mathbf{X}_{*j}) &= \big(\mathbf{k}^{task}_j \otimes \mathbf{k}_{i*}\big)^T \Sigma^{-1}\big(\mathbf{Y} - \boldsymbol{\mu}\big) \\ \Sigma &= \mathbb{K}^{task} \otimes k_i(\mathbf{X}, \mathbf{X}') + \mathbb{D} \otimes \mathbb{I} \end{aligned}$$

where $\mathbf{k}^{task}_j$ is the column of $\mathbb{K}^{task}$ that corresponds to task j, and $\mathbb{D}$ is a diagonal matrix containing the variance of the noise for each task. Then, if we assume identical noise variance, σ^2

$$\Sigma = \begin{bmatrix} \mathbb{K}^{task}_{11} k_i(\mathbf{X}, \mathbf{X}') + \sigma^2\mathbb{I} & \dots & \mathbb{K}^{task}_{1m} k_i(\mathbf{X}, \mathbf{X}') \\ \vdots & \ddots & \vdots \\ \mathbb{K}^{task}_{m1} k_i(\mathbf{X}, \mathbf{X}') & \dots & \mathbb{K}^{task}_{mm} k_i(\mathbf{X}, \mathbf{X}') + \sigma^2\mathbb{I} \end{bmatrix}$$

The Kronecker product, $\mathbf{k}^{task}_j \otimes \mathbf{k}_{i*}$ is defined similarly.

3.2.5 Asymmetric Multitasking

In many cases, the goal is to perform regression for a primary task, and all other tasks may be considered secondary tasks which provide data that may be useful in learning the primary task. For our own setting, CCSD(T) is a natural choice of data for the primary regression task because we would like to make predictions which match this method in accuracy. Each set of DFT predictions made using a different DFA could

inform a secondary regression task which supports the primary task. Since it can be challenging to justify a hierarchy of DFAs—particularly when two or more share a rung of Jacob's ladder—we are interested in a model which does not require us to order all observation sets, the way we must in order to use the Δ method or Deep multifidelity GP regression. Rather, we consider a two level hierarchy. At the top is the primary task, p, which is either the task we are most interested in or for which we have the most accurate data. In our setting, this task will be informed by CCSD(T) data. All other tasks, $s_1, \ldots, s_m$, are treated on an equal footing. For us, these will be the tasks informed by different DFAs. With this structure in mind, Leen et al. describe an asymmetrical model in which secondary tasks are related through the primary task [22].

Suppose $\mathbf{Y}_p \in \mathbb{R}^{n_p}$ is the observation data for the primary task and $\mathbf{Y}_{s_j} \in \mathbb{R}^{n_j}$ is the data for the s_j^{th} secondary task. As before, we assume that each task has its own regression function

$$\begin{aligned} \mathbf{Y}_{ip} &= f_p(\mathbf{X}_i) + \varepsilon_{ip} \\ \mathbf{Y}_{is_j} &= f_{s_j}(\mathbf{X}_i) + \varepsilon_{is_j} \qquad \forall j = 1, \ldots, m \end{aligned}$$

where each noise term is Gaussian iid, with some prescribed variance parameter. We now model a shared structure in the regression functions based on the primary function:

$$f_{s_j}(\mathbf{X}_i) = \rho_{s_j} f_p(\mathbf{X}_i) + \delta_{s_j}(\mathbf{X}_i) \qquad \forall j = 1, \ldots, m \tag{3.6}$$

Leen et al. use the specific component, δ_{s_j}, of the secondary function to "explain away" behavior that is not captured by the shared component, f_p [22]. Both the correlation parameter, ρ_{s_j} and the specific component aim to mitigate negative transfer—learning behavior from secondary tasks that is not representative of the primary task. We make the prior assumption that

$$
\begin{aligned}
f_p(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_p(\mathbf{X}),\ k^p(\mathbf{X},\mathbf{X}')\Big) \\
\delta_{s_j}(\mathbf{X}) &\sim \mathcal{GP}\Big(\mu_{\delta_j}(\mathbf{X}),\ k^{\delta_j}(\mathbf{X},\mathbf{X}')\Big) \qquad \forall j = 1,\dots,m
\end{aligned}
$$

For a given secondary function, s_j, the specific component, δ_j, is modeled both as independent from all shared components, f_p, and independent of other functions' specific components. Then, when we write the covariance between two regression functions, the covariance due to the specific component is a block matrix with zeros corresponding to the covariance of specific functions for different tasks. Call this block matrix K_{spec}. We also define $\boldsymbol{\mu_s} = \left[(\rho_{s_1}\boldsymbol{\mu_p} + \boldsymbol{\mu}_{\delta_1})^T \dots (\rho_{s_m}\boldsymbol{\mu_p} + \boldsymbol{\mu}_{\delta_m})^T\right]^T$ as the vector collecting the prior mean for each secondary observation and a diagonal matrix

$$
\boldsymbol{R} = \begin{bmatrix} \boldsymbol{\rho}_1 & & \\ & \ddots & \\ & & \boldsymbol{\rho_m} \end{bmatrix}
$$

where $\boldsymbol{\rho}_j = \rho_{s_j}\mathbb{I} \in \mathbb{R}^{n_j \times n_j}$.This matrix collects the appropriate correlation parameter for each secondary observation.

We can now write the joint distribution on our observations and targets:

$$
\begin{bmatrix} \mathbf{Y}_p \\ \mathbf{Y}_s \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu_p} \\ \boldsymbol{\mu_s} \\ \boldsymbol{\mu}_p \end{bmatrix}, \begin{bmatrix} K^p_{pp} + \sigma^2\mathbb{I} & K^p_{ps}\boldsymbol{R} & K^p_{p*} \\ \boldsymbol{R}K^p_{sp} & \boldsymbol{R}K^p_{ss}\boldsymbol{R} + K_{spec} + \sigma^2\mathbb{I} & \boldsymbol{R}K^p_{s*} \\ K^p_{*p} & K^p_{*s}\boldsymbol{R} & K^p_{**} \end{bmatrix} \right) \tag{3.7}
$$

The superscript, p, indicates that a matrix was created with the primary kernel, and the matrix subscripts indicate the features that are compared. Inference proceeds as

described in Section 3.2.

The assumed relationship between regression functions in asymmetric multitasking—given by (3.6)—is structurally reminiscent of the assumption made in multifidelity fusion—(3.5). Both assume that the regression function of one dataset is related to the regression function of another by a scaling parameter, ρ, and an additive disparity function, δ.

For only one secondary model, s_1, the relationships between the regression functions posited by both methods are mathematically equivalent for both approaches, but computationally, our choice of which quantities to explicitly model may be influential. For example, it may make a difference if we choose a covariance function for f_p rather than f_s.

Asymmetric multitasking diverges more significantly from the multifidelity method when more than two models are considered. In the multifidelity approach, model hierarchy is reflected in a nested relationship between the functions

$$\begin{aligned} f_{s_1}(\mathbf{X}) &= \rho_p f_p(\mathbf{X}) + \delta_p(\mathbf{X}) \\ &\vdots \\ f_{s_j}(\mathbf{X}) &= \rho_{s_{j-1}} f_{s_{j-1}}(\mathbf{X}) + \delta_{s_{j-1}}(\mathbf{X}) \end{aligned}$$

whereas in the asymmetric approach, all functions share the same relationship to the primary function

$$\begin{aligned} f_{s_1}(\boldsymbol{X}) &= \rho_{s_1} f_p(\mathbf{X}) + \delta_{s_1}(\mathbf{X}) \\ &\vdots \\ f_{s_j}(\boldsymbol{X}) &= \rho_{s_j} f_p(\mathbf{X}) + \delta_{s_j}(\mathbf{X}) \end{aligned}$$

Thus, compared to multifidelity fusion, the asymmetric multitasking approach more naturally accomodates a reasonably large number of regression tasks, and relates each closely to the primary task of interest. This structure lends itself well to our goal

of using several DFT data sets to support prediction at a higher level of accuracy. Unlike the Δ method, we are not required to assume an ordering of all DFT methods, nor do we need different methods to supply predictions for the same molecules. In Chapters 5 and 6, we will examine the posterior mean and variance predictions made by asymmetric multitasking. We will also provide comparisons to the Δ method and basic GP regression. In the next chapter, we will examine additional logistics for designing and training a statistical model with data from electronics structure calculations.

Chapter 4

Design: Statistical Inference in the Quantum Chemical Setting

4.1 Molecular Features

The success of regression techniques relies on a reasonable choice of feature set, $\{\mathbf{X}_i\}_{i=1}^n$, to distinguish the observations (and the targets as well). In our setting, we intend to train a model using a set of predictions for a given quantity of interest, $\left\{\{Y_{ij}\}_{i=1}^n\right\}_{j=1}^m$, where i indicates some molecular system and j indicates an electronics structures method. Consequently, we require features that can describe molecular systems, and depending on our choice of statistical model, we may also welcome features that can distinguish between electronic structures methods.

Much work has been done to design features for molecular systems, often for the purpose of fitting potential energy surfaces [25, 10]. In general, a mapping from molecular space to feature space should be injective and well-defined. Inputs describing each molecular system are given as a list of elements and corresponding Cartesian coordinates for atomic locations. The output of the featurization map should be the same even if we swap the order of entries in the list, or provide new coordinates for the same system after translation or rotation. In this work, we consider Smooth Overlap of Atomic Positions (SOAP) features which have been successfully used to fit interatomic potentials in a Gaussian process based approach [3, 2, 1, 5].

The SOAP feature is constructed from the power spectrum of a local environment model for each atom in the molecular system. Each local environment model has been projected onto the unit sphere, and the power spectrum of this projection represents the amount it fluctuates per angular scale. SOAP design involves a sequence of modeling choices, and we will discuss hyperparameter selection later in this section. First, we define a neighborhood model for each atom i by summing over Gaussian representations of all other atoms within some cutoff radius.

$$\rho_i(\boldsymbol{r}) \equiv \sum_{i}^{\text{neigh.}} \exp\left(\frac{-|\boldsymbol{r}-\boldsymbol{r}_{ij}|^2}{2\sigma_{atom}^2}\right)$$

where $\boldsymbol{r}$ is projected onto the unit sphere and $\boldsymbol{r}_{ij}$ is the vector from the position of atom i to atom j. Next, we expand each neighborhood using spherical harmonics Y_{lm} and a radial basis set g_n

$$\rho_i(\boldsymbol{r}) = \sum_{nlm} c_{nlm}^{(i)} g_n(\boldsymbol{r}) Y_{lm}(\boldsymbol{r})$$

and use the coefficients of expansion to compute a power spectrum

$$p_{nn'l}^{(i)} = \frac{1}{2l+1} c_{nlm}^{(i)} (c_{nlm}^{(i)})^*$$

To fix reasonable values for parameters required by SOAP (r_{cut}, σ_{atom}, n_{max}, l_{max}), we turn both to conventional wisdom and experimentation. The cutoff radius, r_{cut} controls the size of each local neighborhood, and a small value may lead to lost geometric insight. Unfortunately, a large cutoff radius does not necessarily provide proportionate insight: Deringer et al. report that increasing radii larger than 6-8 Å is rarely if ever useful [10]. Similarly, Musil et al. state that 5 Å is a relatively large choice of r_{cut}, and 2 Å may perform better given limited data [25].

Our choice of σ_{atom} also influences our model of each atom's neighborhood: it

determines the lengthscale of the Gaussians placed on each of the surrounding atoms. The larger our choice of σ_{atom}, the more chance that Gaussian tails will slip passed the r_{cut} border. Deringer et al. indicate that the best practice when working with the first three rows of the periodic table is to use $\sigma_{atom} = 0.3$ Å for systems with hydrogen and $\sigma_{atom} = 0.5$ Å for systems without [10]. Finally, we choose the parameters n_{max} and l_{max} to control the size of our expansions of the local neighborhoods. According to Deringer et al. choosing $n_{max} = 12$ with $l_{max} = 6$ is sufficient for high accuracy [10]. In practice, the best values for n_{max} and l_{max} will be sensitive to the choice of radial basis set.

Following the guidance of Deringer et al. and Musil et al., we select candidate values for each SOAP parameter and test the performance of the resultant features in Gaussian process regression. All features for this work were calculated using the DScribe Python package [17]. Our training and test sets consist of small organic molecules from the ANI-1x data set [31]. For more detail on molecule selection, see Duan et al.'s description in [11]. We consider both conventional GP regression and multitask regression to predict two quantities of interest: the ionization potential and electron affinity of a given molecule. To train our models, we pair the SOAP feature representing each training molecule, X_i, with a prediction for the quantity of interest, Y_{ij}, calculated at level of theory, j. The levels of theory we consider include DFT functional approximations PBE and PBE0 as well as a higher level method: Coupled Cluster Singles, Doubles, and Perturbative Triples, CCSD(T).

Figure 4-1 shows the mean absolute error (MAE) obtained with different SOAP parameters when GP regression trained with PBE level data is used to predict Ionization potential (IP). Additional results are included in Appendix A. Our tests include all combinations of $r_{cut} \in \{3, 4, 5\}$, $\sigma_{atom} \in \{0.3, 0.4, 0.5\}$, $l_{max} \in \{2, 4, 6, 8\}$, and $n_{max} \in \{6, 8, 10, 12\}$. Units for r_{cut} and σ_{stom} are Å. We find that our accuracy is comparable to probabilistic predictions of IP found in literature [9]. Of the SOAP parameters, the choice of σ_{atom} has the largest impact of predictive performance. This behavior was observed across all tests we conducted on SOAP parameters. For our experiments later in this work, we fix $\sigma_{atom} = 0.4$ Å because this choice demonstrated

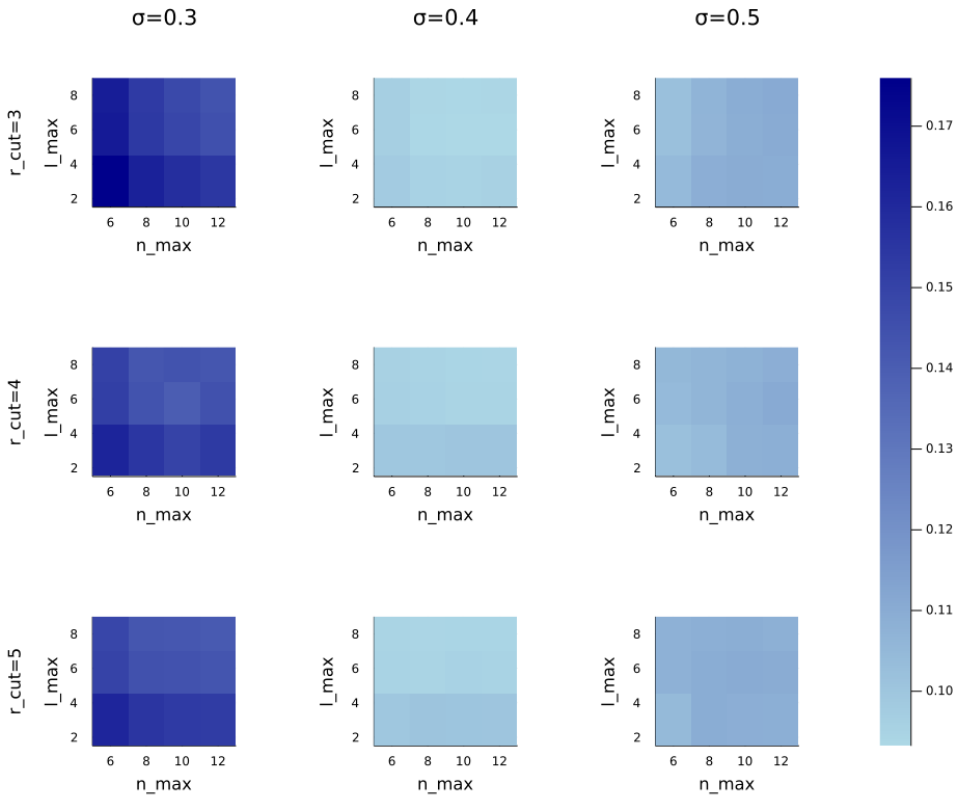


Figure 4-1: Example of inference results for a range of SOAP parameters. The colorbar gives the mean absolute error of predictions made for the ionization potential of small organic molecules.

the best overall performance across our tests. The other parameters are chosen to balance reasonable accuracy and cost. Throughout the remaining tests, we fix both l_{max} and n_{max} at 8, and in general, we use $r_{cut} = 4$ Å. The exception to the latter are cases where the representation of a particular molecular system benefits from a larger neighborhood cutoff radius. For example, in Chapter 5, we featurize water dimers using the difference between the SOAP representation of the dimer and the SOAP representation of its constituent monomers. For some systems, when the cutoff radius is set to 4, the dimer representation is identical to the concatenation of the monomer representations, so we increase the cutoff radius to 7 to better capture the dimer structure.

When using SOAP to compare two molecular systems, we must also make modeling choices to standardize our representation of entire systems. SOAP features are

constructed based on the neighborhood of each atom in a system, so challenges arise when two systems contain different numbers of constituent atoms of elements. De et al. propose a few strategies for constructing "global" features to capture entire systems [9]. The simplest approach is to average the SOAP features for each atom in the system. When using this strategy, researchers should keep in mind that it is heuristic, and the loss of information from averaging may reduce our ability to distinguish between the features of similar molecular systems. A variation to this approach averages the local features corresponding to each element contained in a molecular system. If we wish to compare systems A and B, and system B contains more elements than A, we insert SOAP features corresponding to isolated atoms of those excess elements in the representation of A. This model suggests that an atom of such an element does not interact with the rest of the system. De et al. also introduce the Regularized entropy match (REMatch) strategy for constructing global features. This approach solves a regularized optimization problem to find the match between the sets of local features for each molecular system which maximizes information entropy [9].

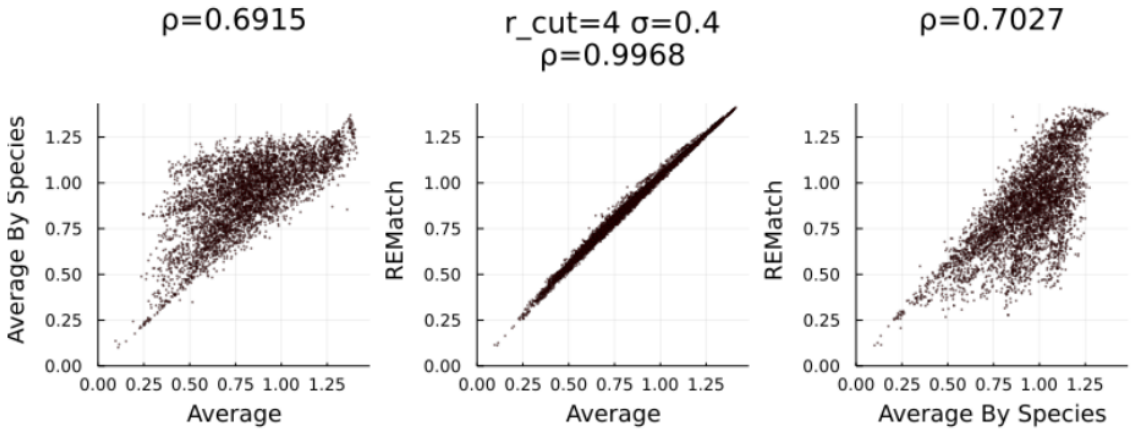


Figure 4-2: Correlation between different methods of constructing global features when calculating the distance between molecule pairs. [9]

With the same set of small organic molecules which we use to test SOAP parameters, we investigate how well the global features constructed by different approaches agree. For pairs of molecules, we construct global features using each of the three approaches outlined in the previous paragraph. We represent difference between each

pair by squaring an inner product of their features. This procedure is the same as using a polynomial kernel of degree 2; kernel functions will be discussed in more detail in the next section. By comparing the output of the kernel for different featurization strategies, we determine whether the strategies generally agree about which molecules are similar.

Figure 4-2 shows the correlation in kernel outputs corresponding to each pair of molecules for different global featurization strategies. Pearson's correlation coefficient (ρ) is reported for each comparison. Consistently, in our tests, we find strong correlation ($\rho > 0.99$) between the REMatch features and features computed by averaging all local representations. Additional results are included in Appendix A. These two strategies are also both positively correlated with the "Average by Species" approach which produces an averaged feature for each element in the system and inserts isolated atoms to represent elements not including the system, but these correlations are not as strong as that between the REMatch and totally averaged features. We may see this result because the REMatch and totally averaged features share the same dimension, n, whereas features that are constructed by species specific averaging have dimension of 2n to 4n in the cases we considered.

Because averaging features is much more computationally efficient than constructing REMatch features, we use this approach to represent small organic molecules. In later parts of this work, we also consider energy difference between water dimers and monomers. In these cases, a totally averaged features can be insufficient to distinguish between systems, so we represent these systems using features that are averaged by element type. We are not required to insert any isolated atoms into the water representations because systems contain only oxygen and hydrogen.

SOAP features are a useful tool for our investigation of different inference models applied to DFT data because they can represent molecule geometry in enough detail to produce reasonable accuracy in mean predictions, but this design choice has limitations. We have several goals: prediction of some quantity of interest for a given molecular system, evaluation of our confidence in our own prediction, and indication of how accurately DFT can predict the quantity of interest. From (3.4), we see

that when we employ a Gaussian process inference approach, our posterior mean and variance predictions rely strongly on how we construct our features, $\{X_i\}_{i=1}^n$. SOAP features capture the location of different atoms relative to each other but may neglect relevant information encapsulated in the electron density of the system, the true input to Density Functional Theory. Additionally, SOAP features will not include any information about a particular density functional approximation. Thus, they may not be able to represent subtleties which make particular molecular systems challenging for particular DFAs to capture accurately.

We also face challenges when interpreting the feature space that results from SOAP. Even for small molecules, SOAP features have dimension on the order of 1000. Our statistical models rely on kernel functions to represent the distance between features, and the high dimension of SOAP vectors can cause challenges for choosing appropriate kernel functions and evaluating how well their representations match reality. When we use inference models, prediction confidence and accuracy are generally lower when we extrapolate than when we interpolate, and the high dimensionality of SOAP feature space makes it challenging to classify these cases. Here, we investigate the performance of inference models when we choose a SOAP feature representation, but there is certainly room in future work for considerable investigation into appropriate features to inform probabilistic inference and uncertainty quantification.

4.2 Kernel Functions

We must choose some kernel function to compute the similarity between feature representations of molecular systems. GP inference models use these similarity values to weight the relevance of our observation data points in order to make new predictions and to estimate the variance of the posterior distribution. Equation (3.4) shows the influence of kernel functions in GP regression predictions, as do the formulations of the Δ, multifidelity, and multitask models.

There are many standard choices of kernel function. Most applications of Gaussian

process inference to materials modeling use low order polynomial kernels [10]. These functions have the form

$$k_{PK}\big(X, X'\big) \;=\; \big(X^T X'\big)^{\xi} \tag{4.1}$$

One appeal of this function is that it only introduces one new parameter, ξ, which is typically set to 2 or 4 for materials applications. Additionally, the inner product can offer a useful route for mapping the high dimensional SOAP features to scalars though it risks loss of fine scale characteristics of the feature.

In practice, inference models that rely on polynomial kernels generally make posterior mean predictions with low mean absolute error, and this ability is a major reason for practitioners to use these functions. Unfortunately, accuracy in posterior mean prediction does not imply accuracy in posterior variance prediction. For a molecule with feature X_*, GP inference yields a posterior variance given by

$$\big(X_*^T X_*\big)^{\xi} \;-\; \big(X_*^T \mathbb{X}_f\big)^{\xi} \Big(K_{ff} + \sigma^2 \mathbb{I}\Big)^{-1} \big(\mathbb{X}_f^T X_*\big)^{\xi}$$

where $\mathbb{X}_f$ is a matrix with columns corresponding to the features of training molecules and exponents of ξ are applied element-wise. Thus, our prior guess of the variance for prediction of the quantity of interest for molecule $*$ is an exponentiated inner product of $*$'s SOAP feature. The prior is then corrected with an estimate of how much knowledge we have gained from our observations based of our estimate of closeness of the observations to *, calculated with the polynomial kernel. The magnitude of the inner product of SOAP features increases for larger molecules and for SOAP parameter settings that allow for more detailed representations of these molecules. In general, this quantity does not correlate with error or uncertainty. Thus, by construction, we have limited expectation that the posterior variance obtained with a polynomial kernel can be a good indicator of error or uncertainty of the posterior mean prediction. Multifidelity and multitask methods also produce posterior variance estimates which are structured as some correction to this prior variance. It may be tempting

to augment the feature with additional entries that do correlate to uncertainty. To be successful with this approach, however, we would require a reliable method for indicating uncertainty, exactly what we wished for from our probabilistic model.

As an alternative, we may consider the Squared Exponential (SE) kernel for multidimensional features. This kernel is widely used in Gaussian process literature and takes the form

$$k_{SE}\big(X, X'\big) \;=\; v \exp\left(-\sum_{\alpha=1}^{M} \frac{(X_\alpha - X'_\alpha)^2}{2\ell_\alpha^2} \right) \tag{4.2}$$

This function introduces variance, v, and lengthscale, ℓ, parameters. We may choose ℓ to be a scalar or to have dimension to match the feature, X. The latter choice presents interesting possibilities for magnifying important components in the feature, but also can translate to a tricky parameter estimation problem, particularly when working with SOAP features which have thousands of entries. Estimating ℓ via a fully Bayesian approach is intractable [4], so we instead obtain point estimates of the both ℓ and σ^2 by maximizing the log likelihood. Note that this approach easily extends to multifidelity and multitask cases by treating $Y_{ij} - \rho_i Y_{i1}$ as the i^{th} observation, as demonstrated by Forrester et al. [12]. If tasks 1 and j do not have data for the same set of features, a conventional GP regression model may be used to fill in missing data for task j. Estimating unique components of ℓ corresponding to each SOAP component tends to be more computationally expensive than it is accurate, so for our investigation we restrict ℓ to be a scalar. We also choose to use the mean of our training data as our prior mean estimate, as this value performed better in practice than the estimate obtained from optimization.

In contrast to the Polynomial kernel, the SE kernel is isotropic—it depends only on the absolute difference between features, not the individual features or the direction of difference. Both kernels may be limited in their ability to pick up patterns within features, but in practice both kernels lead to posterior mean predictions with reasonable mean absolute error (MAE). Figure 4-3 compares the performance of the two kernels when used in a GP inference model trained with CCSD(T) predictions

of the ionization potential for small organic molecules. We see that MAE decreases at a similar rate for both kernels as the number of training data points increases. Furthermore, an SE kernel with unoptimized parameters (set to a default value of 1) generally produces smaller MAE than the polynomial kernel, and the SE kernel with optimized parameters in turn produces smaller MAE than the unoptimized SE kernel.

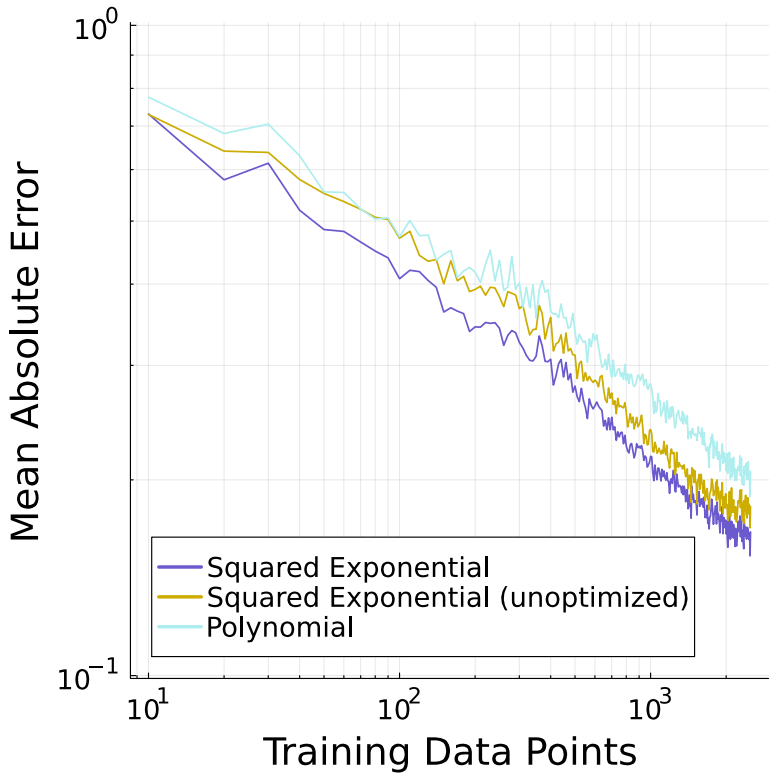


Figure 4-3: The performance of GP regression for a polynomial kernel with degree 2 as well as a squared exponential kernel with and without optimized parameters.

We can also compare the posterior variance produced by the two kernels. For GP regression, the SE variance is

$$v\mathbf{1} \; - \; K_{f*}^{T}\Big(K_{ff} + \sigma^2\mathbb{I}\Big)^{-1} K_{f*}$$

where **1** is a vector of 1s with appropriate dimension. Note that each diagonal element of each K matrix is also equal to v. Thus, the SE kernel sets the prior variance to its variance hyper parameter, which is optimized with observational data. This value is a more reasonable initial estimate of uncertainty than the inner product of the SOAP feature of the target, but it is not specific to any given target molecule. We rely on the correction term introduced by inference to differentiate our uncertainty predictions for each molecule. By introducing a multifidelity or multitask scheme, we may have different prior variances for different tasks and levels, but differentiation within tasks is still limited. Consequently, to obtain reasonable uncertainty indicators from the posterior variance, the SE kernel's computation of distance between features must correspond to a reasonable relationship between molecules. The ideal measurement of distance would characterize DFT's ability to accurately describe each molecule. We will use the SE kernel in our experiments, and we will discuss the challenges of representing uncertainty in this setting further in Chapter 6.

4.3 Dataset Construction

The training data set for our statistical models will include predictions from CCSD(T) as well as from each DFT method we consider. There are many choices that can be made in designing the training data set. For example, given a computational budget for data generation, we might choose between computing 100 training data points each from four different DFT methods or 200 training data points from two methods. We must also choose whether we should generate data for the same set of molecular configurations with each quantum chemistry method or if we should make predictions for different configurations with different methods to cover a wider region of chemical space. We may also want to leverage the ability of the multitask method to train a model using DFT data corresponding to the molecules for which we want to make predictions at CCSD(T) level accuracy. This section will describe how we construct a variety of data sets to investigate the two examples which the remainder of this thesis focuses on: the organic molecules case study and the water case study.

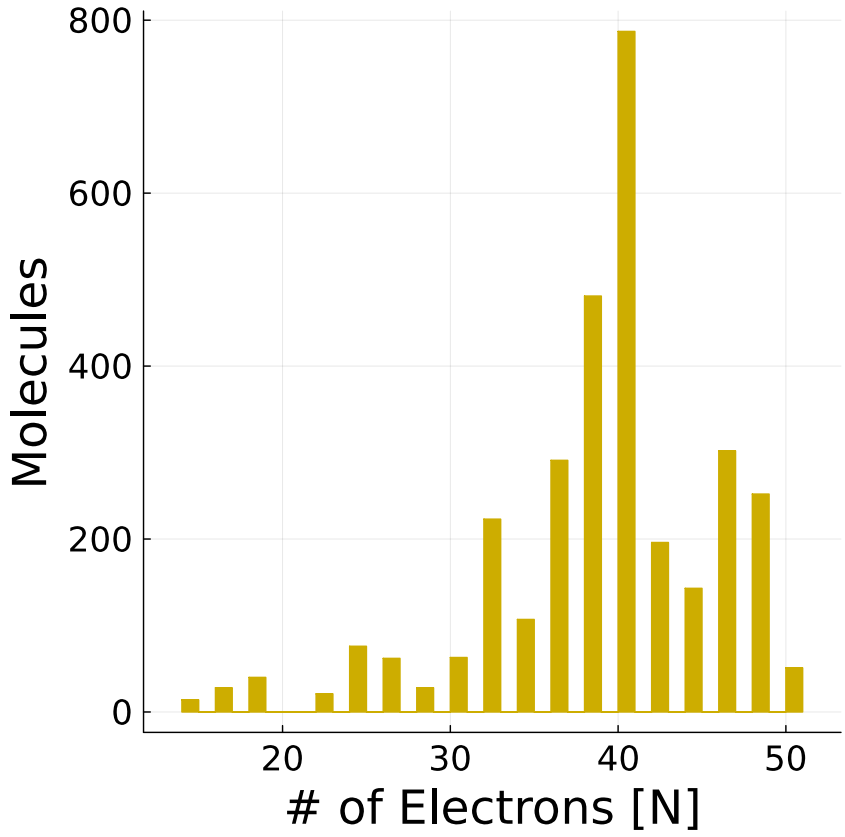


Figure 4-4: Distribution of the number of electrons in molecular configurations used in the organic molecules case study.

In both case studies, we will use CCSD(T) calculations as our primary level of theory. All other levels of theory will correspond to different density functional approximations. For the organic molecules case study, we will consider up to four DFAs: PBE, BLYP, PBE0, and PBE0_DH. For each level of theory, our data set consists of predictions of the ionization potential of small organic molecular configurations. The set contains a total of 3,165 molecular configurations, combinations of 479 different molecules and 7 configuration options. This selection of these molecular configurations from the ANI-1 data set is described by Duan et al. [11]. They contain only elements from the set $\{H, C, N, O\}$, and the distribution of the number of electrons in the configurations is shown in Figure 4-4.

We assign the molecular configurations to three sub data sets—Core (C), Supplemental (S), and Target (T)—referred to in Figure 4-5. Shading in the figure indicates

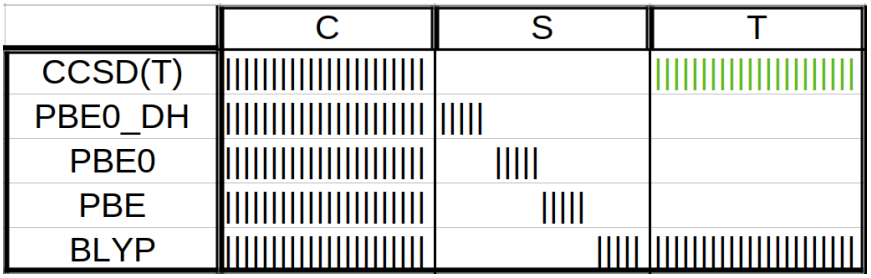


Figure 4-5: **Small Organic Molecules Case.** Example data set structure. The supplemental set, S, contains only DFT predictions, and in this case the S sets for different DFA do not overlap. That is, the predictions for different DFAs are for entirely diferent molecules. The target predictions are on the CCSD(T) level and are highlighted in green. All other predictions are used for training.

when configurations are available at a given level of theory. Black indicates availability to the training set, and green indicates membership of the testing set. All subsets are randomly drawn from the full set of molecular configurations. The C set contains all configurations that the model can train on at a CCSD(T) level. In the example given by Figure 4-5, we can train on these configurations at a DFT level as well. The S set contains all configurations that the model can train on only at DFT levels of theory. Figure 4-5 shows a case where different levels of theory do not have molecular configurations in common, but we may also consider cases where different levels in the S set overlap partially or completely. For the results reported in Chapters 5 and 6, we iteratively change the sizes of the C and S set from 16 to 512 configurations.

Finally, the T set contains the configurations we will use to test our inference model. We will make predictions for the ionization potential of these molecules at the highest level of theory and compare our accuracy to a held out set of CCSD(T) calculations. The configurations that we test on are green in Figure 4-5. When we draw on multiple levels of theory to train our model, we may also include lower level predictions of target molecules in our training set—represented by the black shading in the T column of Figure 4-5. Note that to apply the Δ method, we must have these low level target predictions to add to our predicted disparity to make a full prediction of the quantity of interest. The multifidelity and multitask methods do not require low level target predictions, but we will see in Chapter 5 that including them in the

training set generally produces improved accuracy. For the organic molecules case study, we construct a T set of 500 molecules. Our testing framework is designed so that models trained with DFT target data can readily be compared with equivalent models that do not train on this data. Since target predictions can be made one by one, for each target molecule we train a new model using the C set, S set, and the DFT prediction for only that target molecule. While in practice this training process would be inefficient, by comparing each of these models to the corresponding model trained on only the C and S sets, we can isolate the effect of the DFT level target data on prediction performance.

For our second case study, we test the performance of our statistical models when predicting the interaction energies of water dimer configurations. Dimers consist of a pair of weakly bonded molecules (called monomers), and their interaction energy is the difference between the total energy of the system and the sum of the energies of isolated copies of the constituent molecules. Different water dimers are distinguished by the different distances between the constituent monomers, different OH bond lengths, and different HOH angles. To train and test our models, we draw from a set of 70,000 calculations on the CCSD(T)/CBS level of theory [35].

We support our primary inference task using DFT predictions for the interaction energies of water dimers. For this work, we computed 1000 dimer interaction energies on the PBE0/aug-cc-pvtz and PBE/aug-cc-pvtz levels. Counterpoise correction was performed to ameliorate basis set superposition error. For additional investigation of our model's variance predictions, we also compute 6000 energies of water monomers at the CCSD(T)/aug-cc-pvtz level. To complete the water dataset, we use PBE0/aug-cc-pv5z and PBE/aug-cc-pv5z level energy calculations for the same 6000 molecules, obtained from [2]. It is worth emphasizing that multitask modeling can leverage data sets compiled from multiple sources.

Recall that when we train our inference models, we pair each data point, Y_{ij} with some feature X_i. In our case, we use SOAP features which represent the geometry of molecular systems. We are interested in a strategy of representation that can be applied to energy difference between a dimer and its constituent monomers as

well as energy difference between two monomers. To this end, we construct the full SOAP feature for each atom in each system and take the difference between corresponding atoms in the two systems. We then compute one average of the vectors corresponding to H atoms and a second average of the vectors correspond to O atoms and concatenate the two.

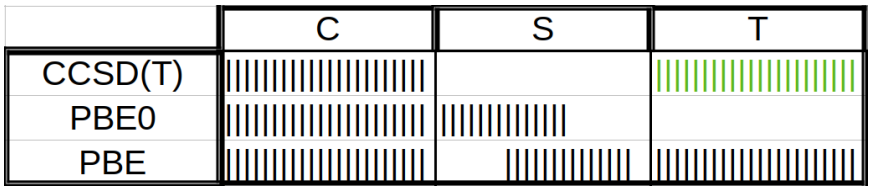


Figure 4-6: **Water Dimers Case.** Example data set structure. Here, the S sets are shown to partially overlap for different levels of theory, indicating that the data set includes predictions by different DFA for some of the same molecules. The testing data set is highlighted in green.

As in the organic molecules case, we can construct various training sets by partitioning our data into Core, Supplemental, and Target sets. The definitions of the sets are as before. Figure 4-6 provides an example schematic. The partially overlapping shaded region in the S columns of Figure 4-6 indicate a case where our predictions for different levels of theory have some molecular systems in common but not all. We will also consider cases with complete overlap or no overlap between levels of theory in the S set. When training models with multiple levels of theory, we iteratively construct C and S sets containing up to 320 dimers.

In the following chapters of the work, we will use abbreviations such as "CS" or "CST" to indicate what DFT level data is included in the training set. It is assumed that CCSD(T) data from the C set is always included the training data. Note that CCSD(T) level data from the T set is strictly reserved for testing, and never included in the training data set.

Chapter 5

Strengths of the Multitask Approach: Efficient Mean Prediction

This chapter demonstrates that multitask and Δ inference can produce more accurate predictions than conventional Gaussian process models for a given computational budget. All inference models that we consider make predictions in the form of a joint normal distribution on some quantity of interest where each marginal corresponds to a target molecule. Within this chapter, we evaluate accuracy by comparing the mean predictions of these marginals to CCSD(T) predictions of the quantity of interest for the target molecules.

We consider two examples: prediction of the ionization potential of small organic molecules and prediction of the interaction energies of water dimers. These cases are described in more detail in Chapter 4. Appendix B also provides some summary statistics for the data sets considered. In the organic molecules case, our data set contains systems with between 20 and 50 electrons, so the computational cost of quantum chemistry calculations to build our training set is variable. To estimate cost in these cases, we assume that a CCSD(T) calculation is 1000 times more expensive than a DFT calculation. This assumption is fairly conservative; DFT scales like N^3 and CCSD(T) scales like N^7 where N is the number of electrons in the molecular configuration. For systems in our data set, the ratio between the costs of CCSD(T) and DFT for a given molecule will exceed 1000. For each prediction, we report

the estimated cost of generating the statistical model's training data in units of the number of CCSD(T) calculations—that is, we convert DFT calculation cost to a fraction of CCSD(T) calculations and add this quantity to the CCSD(T) cost. We will compare multitask models to GP models trained only on CCSD(T) for a given cost estimate. By assuming a conservative cost ratio, we estimate the multitask models to be more expensive than they actually are in comparison to the CCSD(T) GP models. Thus, we expect true performance of the multitask models per cost to be even better than shown in our results.

For the water dimers case, we estimate data set generation cost based on the expense of our own calculations of the interaction energies of water dimers and energies of water monomers. The order of the costs are given in Table 5.1, and in our results, we report cost in units of seconds. Note that computational cost scales with the number of electrons in the system, so a greater gap between CCSD(T) and DFT cost is expected for water n-mers with $n > 2$.

System	QOI	Method	Cost [s]
Water Dimer	Interaction Energy	CCSD(T)	≈ 3700
Water Dimer	Interaction Energy	DFT	≈ 42
Water Monomer	Energy	CCSD(T)	≈ 25
Water Monomer	Energy	DFT	≈ 3.1

Table 5.1: **Water Dimer Case.** The order of cost for CCSD(T) and DFT computations to generate the training data set

The following two subchapters examine the performance of the multitask method for the organic molecules example and the water dimers example, respectively. The final subchapter highlights the relative strengths of the Δ and multitask methods.

5.1 Multitask Method: Organic Molecules Case

We will examine the impact of statistical model design on the accurate prediction of ionization potential for 500 target small organic molecules. See Chapter 4 for details on how different training sets can be constructed by dividing data into Core (C),

Supplementary (S), and Target (T) categories; Figure 4-5 provides a visualization of a data set used to train and test a multitask inference model for the organic molecules example. Figure 5-1 reports results for models constructed with only two rows of data from Figure 4-5; that is, the model considers one primary task (regression on CCSD(T) data) and one secondary task (regression on DFT data). Each scatter point on the left subfigure corresponds to an inference model where PBE was used as the density functional approximation (DFA) to generate the secondary training data, and each scatter point on the right subfigure corresponds to a model where PBE0 was used to generate secondary data. The colors of the points indicate what combination of C, S, and T the secondary training set includes. The black line marks the results of a conventional GP inference model trained only on CCSD(T) data. Cost is given in units of CCSD(T) calculations as described at the beginning of this Chapter, and the error is the MAE that results from predicting 500 target ionization potentials, averaged over three random constructions of C, S, and T.

Each multitask model represented in Figure 5-1 performs at least as well as the GP reference. For many constructions of the training data set, the multitask method performs substantially better than the GP method. In particular, inclusion of DFT level data for target molecules (the T set) when the overall training set is relatively small can enable the multitask method to exhibit accuracy comparable with a GP model an order of magnitude more expensive. Performance of the multitask method is similar for both DFAs considered, but PBE0 demonstrates a slight edge in accuracy. In both cases, Figure 5-1 shows clear stratification in the performance of models trained with secondary training sets constructed from the C, CS, and CST data. There is a clear advantage to adding low level data for additional molecules (the S set) compared to merely using DFT to duplicate predictions we have on the CCSD(T) level (the C set). There is an additional advantage to training on DFT level data for the target molecule (the T set).

The two subplots also show six clear groupings of points with steep slope. This effect is a consequence of the high cost of CCSD(T) relative to DFT. Each of the six groupings corresponds to a different size of the C set. Since we also consider six sizes

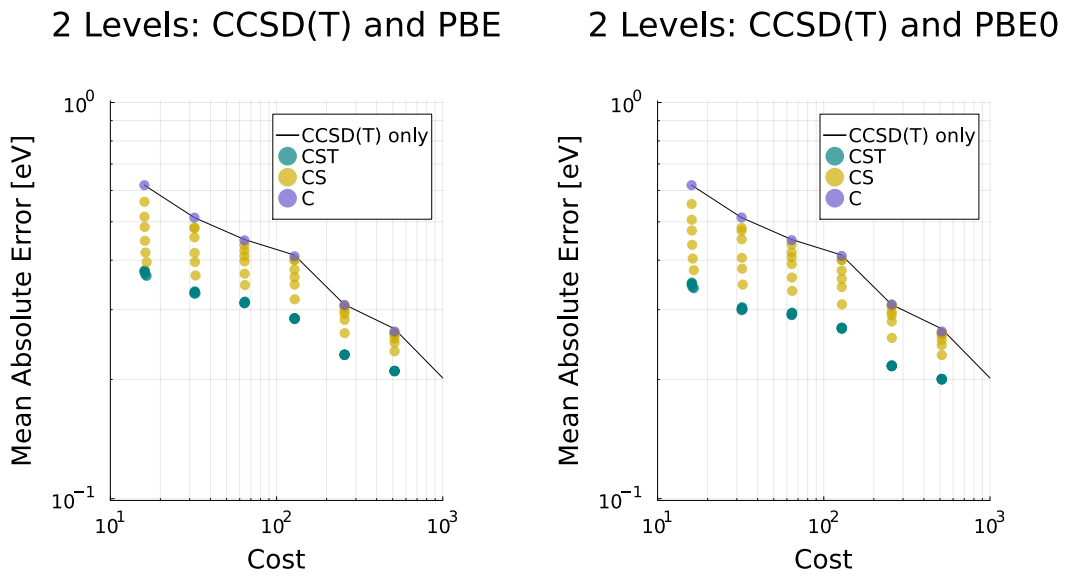
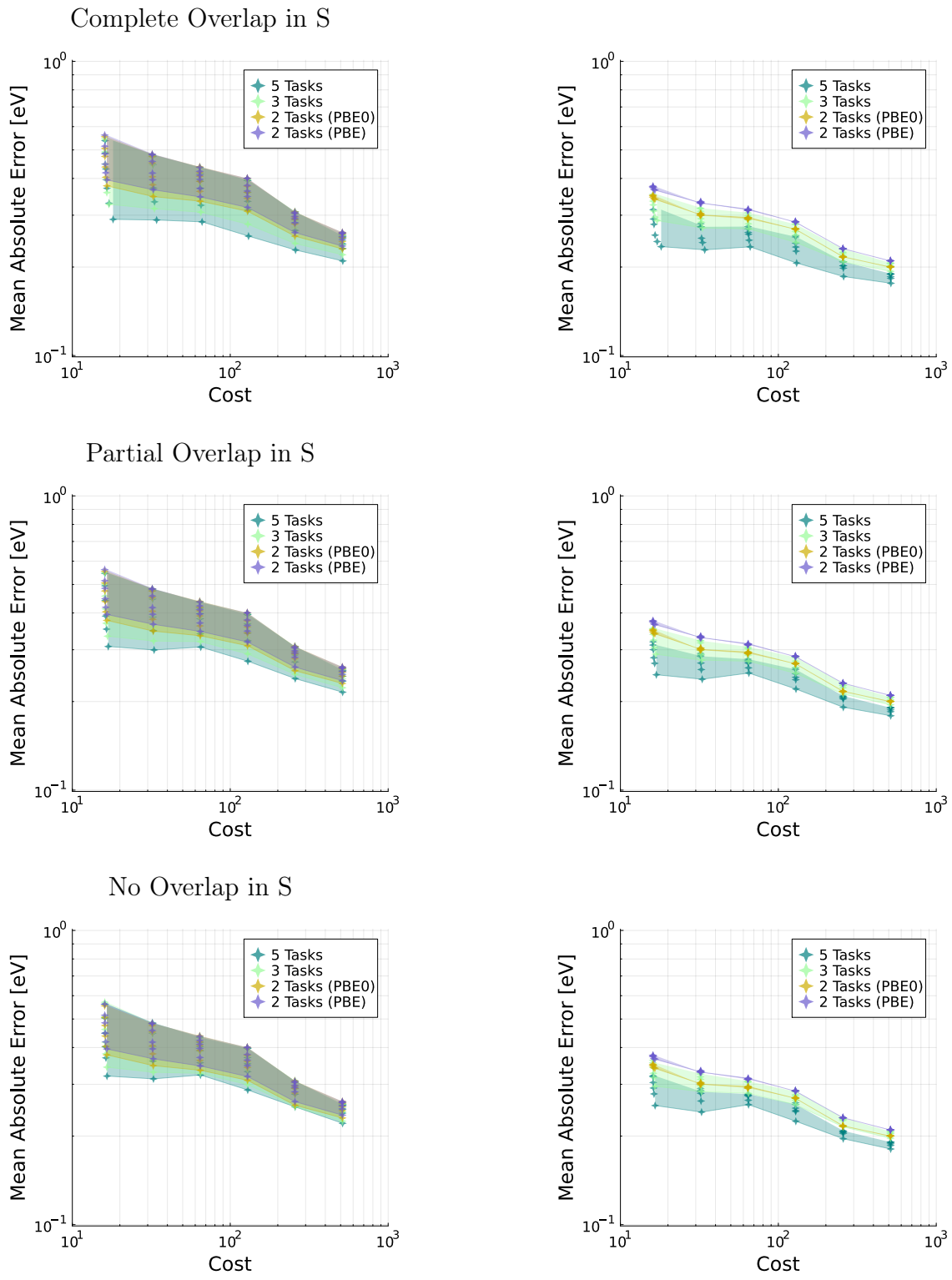


Figure 5-1: **Organic Molecules Case.** A comparison of different choices of levels of theory as well as the inclusion of different sets (C,S,T) in the training set for the multitask approach. The left plot shows the mean absolute error of an inference model trained with CCSD(T) and PBE produced data, and the right plot shows MAE of a model trained with CCSD(T) and PBE0. The indigo points correspond to inference models trained on only molecules from the core set, the gold points correspond to models trained on both C and S sets, and the teal points correspond to models trained on molecules from the C, S, and T sets. The GP inference results for a training set of CCSD(T) only are plotted as a black line.

for the S set, each grouping contains one C result, six CS results, and six CST results. The C results demonstrate no significant improvement on the reference GP model while CS results show steady improvement as the size of the S set increases. This behavior suggests that for the multitask method to be beneficial, secondary training data must cover molecular space that is not included in the CCSD(T) training data. It may also be possible that the abilities of our C tests are limited by our decision to use features dependent only on molecular systems as well as our assumptions about the relationships between regression functions for different tasks. The models trained on DFT data from the T set outperform CS models of comparable cost. Note that for the CST cases, increasing the size of S does not produce the steady improvement in error seen in the CS cases. It appears that the advantage gained by including T dominates the advantage offered by increasing S.



(a) Trained without DFT data from Target Set (b) Trained with DFT data from Target Set

Figure 5-2: **Organic molecules case.** For different numbers of levels of theory (indicated by color), plots show MAE versus cost. The top row corresponds to a fully overlapping S set, the second to a partially overlapping set, and the bottom to a non-overlapping set.

We can also consider the impact of low level data from the target set when we extend to multitask implementations with more than two tasks. Figure 5-2 compares the results for the two task implementations of multitask inference from Figure 5-1 to implementations with three and five tasks. The three task case includes predictions by both PBE0 and PBE as secondary tasks, and the five task case additionally uses predictions by the PBE0_DH and BLYP DFAs. This latter case includes exactly the five tasks represented in Figure 4-5. The first column of subplots presents models with a CS construction of the secondary training sets, and the second column corresponds to a CST construction of secondary training sets. Note that for models with multiple secondary tasks, a design choice can be made about how much of the S set is included for each secondary task. Figure 4-5 represents a case where there is no intersection in the part of the S set accessed by each secondary task. Such cases are considered in the bottom row of Figure 5-2. The middle row considers cases where there is partial overlap in the S set for each secondary task. To illustrate the set up, suppose we have secondary tasks W, V, and U; then, V would share half of the molecules in its S set with W, and the other half with U. Models represented by the top row of Figure 5-2 have complete overlap in the S set for each secondary task. Each subfigure reports accuracy for a given training set generation cost, where accuracy is the average MAE obtained from three random constructions of C, S, and T.

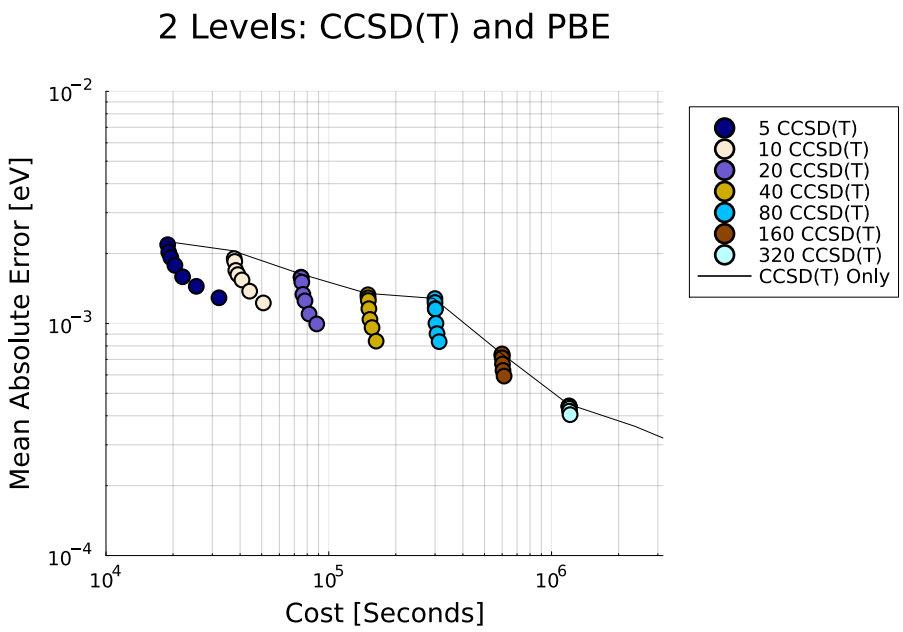
Figure 5-2 indicates that there is a benefit to training an inference model with a larger number of tasks for a given budget. On each subfigure, for each of the six groupings corresponding to difference sizes of the C set, the distribution of error of the three and five task models dips lower than the distribution of error of the two task models. This effect is larger both when there is a greater degree in overlap in S sets for different secondary tasks and when models are trained on low level data from the target set. Thus, the five task implementation has the greatest lead on other approaches in the top right plot, the plot corresponding to models with complete overlap in the S set for secondary tasks and a CST training set construction. These results are useful for determining how best to generate new data sets to train multitask inference models, given a fixed budget, but they also indicate that there are many

ways to construct a training data set and achieve reasonable accuracy. The multitask method can be a tool for bringing together multiple existing data sets for inference, effectively using a "data set of opportunity" to inform regression.

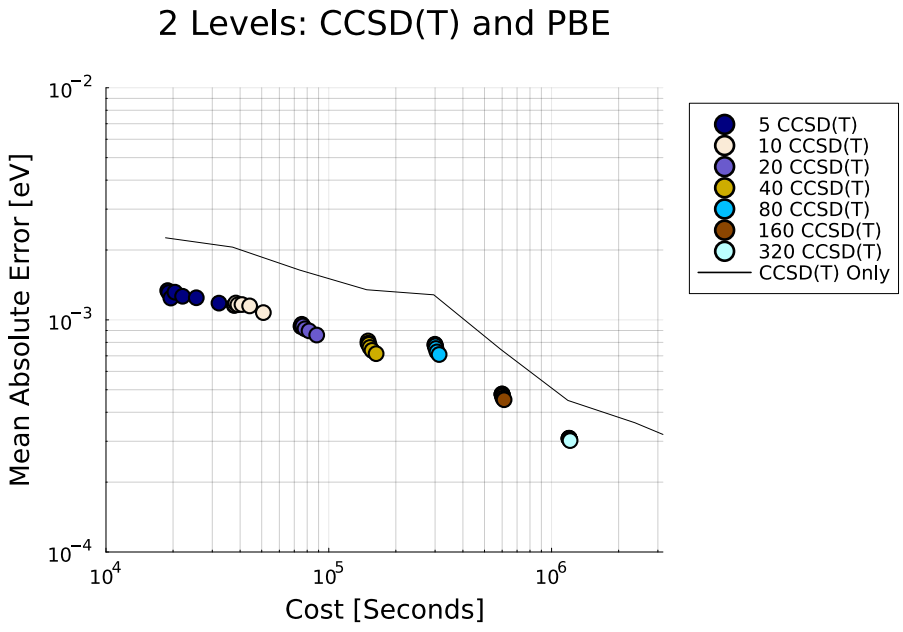
5.2 Multitask Method: Water Dimers Case

The multitask method also performs well compared to a conventional GP model when we predict the interaction energies of water dimers. There are some distinctions to make between this example and the case of predicting interaction energies for organic molecules. While the median CCSD(T) prediction for ionization potential of the organic molecules data set is 9.409 eV, the absolute median of the CCSD(T) predictions for dimer interaction energy is 0.0028 eV. Consequently, we expect smaller absolute prediction error for interaction energies. For more information on the statistics of these data sets, see Appendix B. Furthermore, in the small organic molecules examples, we built data sets both from different molecules and from different configurations, and in this example our entire data set corresponds to different configurations of water dimers. Finally, we construct inputs to our model from the difference between the SOAP feature for the dimer and the concatenation of SOAP features for its constituent monomers. We then average the vectors corresponding to O atoms and the vectors corresponding to H atoms before concatenating the two averages. This construction is motivated by its utility for future tests on data sets which include energy differences of n-mers where n varies.

Figure 5-3 shows that with only 5 CCSD(T) predictions, the multitask method can achieve MAE comparable with a GP model trained on a CCSD(T) data set which costs an order of magnitude more to generate. Subfigure (a) presents the results of multitask models which use a CS structure for the secondary training set while subfigure (b) presents results for a CST training structure. The target data set contains 200 dimer configurations, and the C and S sets are allowed to vary in size from 5 to 320 configurations. See Figure 4-6 for more detail on the C and S sets. The colors of the scatter points indicate the number of CCSD(T) calculations in the C



(a) No target DFT data



(b) Target DFT data

Figure 5-3: **Water Dimers Case.** Scatter points show the performance of different implementations of the multitask model for a given cost. Color indicates the number of CCSD(T) data points used to train models, and all multitask models are trained with supplemental DFT data. The black line marks the accuracy of a GP model trained only on CCSD(T) for a given cost. All accuracy statistics are based on the average of six tests with different random assignments of dimers to data sets.

set. The line marks the performance of a GP model trained only on CCSD(T) data. We report the average MAE value obtained by six random assignments of each data set. As in the organic molecules case, the multitask models perform at least as well as the CCSD(T) only model at each cost. When the S set is sufficiently large or DFT data for the target dimers is included in the training set, the multitask model can perform significantly better than the CCSD(T) only model.

Note that as the number of CCSD(T) calculations used to train the multitask models increases, the improvement from increasing the size of the S set seems to decrease. This effect is likely caused by the size of S relative to the size of C (the number of CCSD(T) in the multitask training set). We consider S sets ranging in size from 5 to 320 for each C set size. Therefore, when we train with 5 CCSD(T) data points, the maximum S set size is 64 times larger than the C set size. When we train with 320 CCSD(T) data points, the maximum S is exactly the C set size. More significant improvement in accuracy may be possible if the maximum size of the S set scaled with the size of the C set. While more angles of this example are left to be explored, the results here demonstrate that patterns of performance of the multitask model found in the organic molecules case can also be found in different data sets.

5.3 Comparison of the Multitask and Δ Methods

For many of the data sets we have considered, we can also perform inference with the Δ method. Figure 5-4 compares the two methods for several iterations of the CST data set, and finds that the Δ method outperforms the multitask predictions. The left subfigure plots the average MAE of the multitask model—taken over three random assignments of molecules to the CST sets—against the average MAE of the Δ method. The right subfigure reports correlation between the true CCSD(T) predictions and the final predictions of the two inference methods, measured by three different coefficients.

In this setting, the Δ method's treatment of disparities is its advantage, but the data set structure this treatment requires is its disadvantage. The Δ method directly models the difference between two levels of theory which can be an advantage when

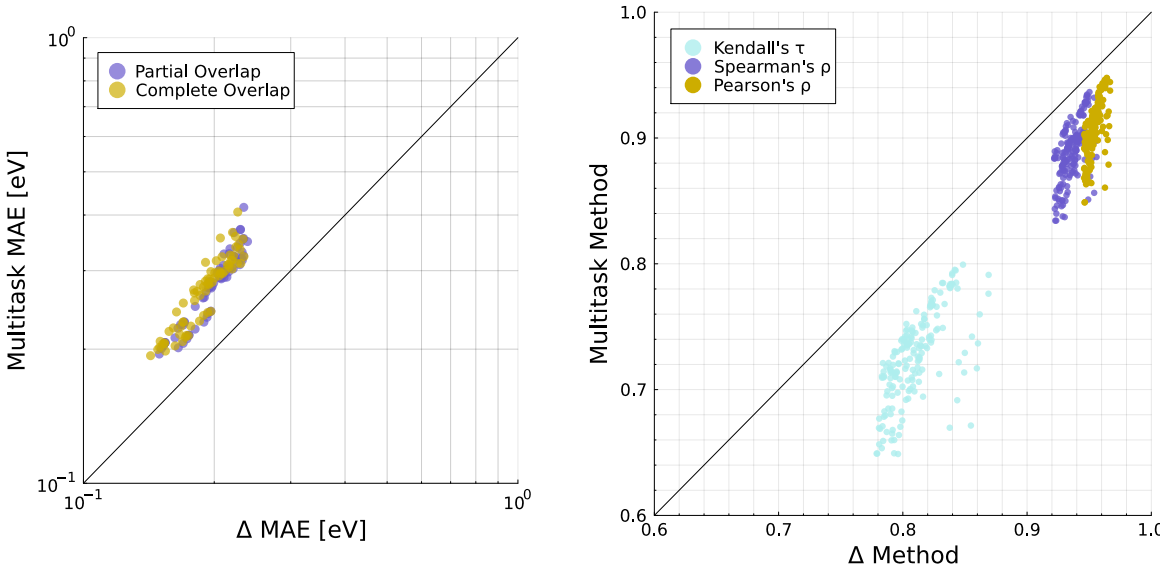
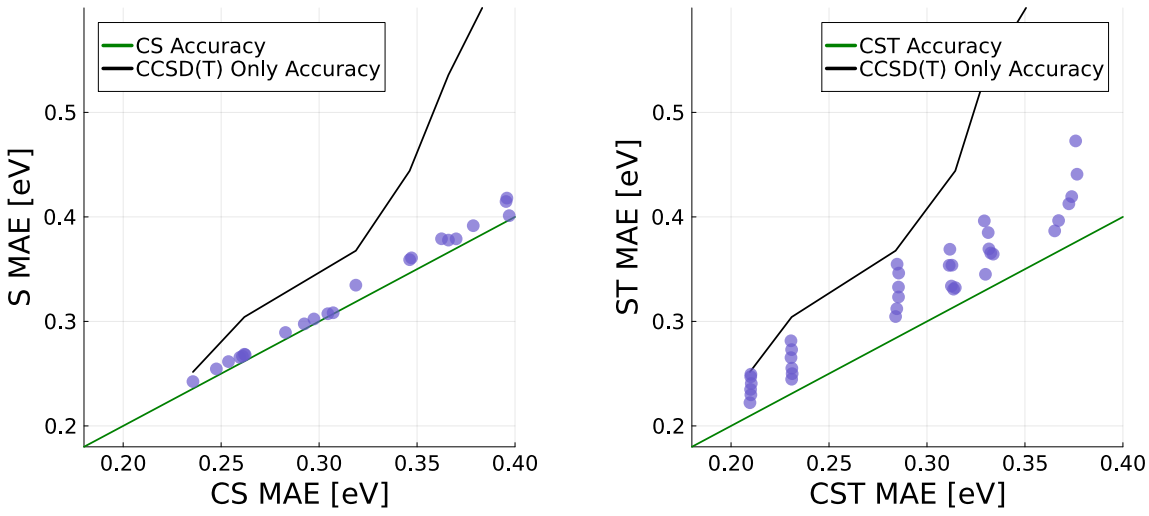


Figure 5-4: **Organic Molecules Case.** Comparison of accuracy of the multitask method and the Δ method applied to the same training data set. The left subfigure compares mean absolute error averaged over three random draws of the C, S, and T sets. Indigo points correspond to training sets with partial overlap of molecules used to train different secondary models in the S set, and gold points correspond to complete overlap in the S set. The right subfigure compares correlation coefficients between predictions and CCSD(T) calculations for the target set. Gold corresponds to Pearson's ρ, indigo to Spearman's ρ, and teal to Kendall's τ.



(a) (b)

Figure 5-5: **Organic Molecules Case.** (a) The scatter points plot the accuracy of a multitask model with a CS secondary training set against the corresponding multitask model which drops the C molecular configurations from the secondary training set. The green line is $x = y$. The black line shows the accuracy of a GP model trained on CCSD(T) at the cost of the multitask model errors reported. (b) The set up of this subfigure is analogous to (a), but a comparison is made between a CST trained model and the corresponding ST trained model. All MAE values are averaged over three random data set assignments.

these levels are highly correlated and only a small correction to the lower level is necessary to obtain higher level accuracy—as is the case for CCSD(T) and the DFAs we test. By contrast, the multitask approach models the highest level of theory as well as the difference between all secondary levels and a scaled version of the model for the highest level. This tactic involves more modeling choices per level and does not make much explicit use of the difference models. The advantage of the multitask method is that it can train on a "data set of opportunity" which brings together data from different levels of theory that do not necessarily correspond to the same molecular configurations. To apply the Δ method, we must have corresponding data for multiple levels as well as a low level prediction for our target.

Figure 5-5 demonstrates that the multitask method performs well even when the training data for different tasks do not have any molecules in common. The scatter

points in the figure compare the MAE of multitask models trained with DFT data from both the C and S molecule sets (depicted in Figure 4-5) to the MAE of corresponding multitask models trained on the same data, except that all DFT data from the C set is excluded. All of the models considered here use only two tasks, so excluding DFT data from the C set yields a training data set with no overlap in the molecules used to train each task. We will call these training sets "disjoint" to contrast with an "overlapping" training data set. The multitask models considered in subfigure (b) also train on DFT data for target molecules while the models in subfigure (a) do not. The green (lower) line plots $x = y$, and the black (upper) line shows the MAE of a GP model trained only on CCSD(T) data for the same cost which was necessary for a multitask model to achieve the MAE reported on the x axis. While we find that the multitask models trained on disjoint sets do not have the same accuracy as their counterparts with additional DFT data which overlaps with primary training data, their accuracy is reasonably close, especially in subfigure (a). All multitask models with disjoint training sets also outperform the the CCSD(T) trained reference model at comparable costs.

Note that the gap in performance between ST and CST models is larger than between S and CS models. One possible explanation is that including both primary and secondary training data for C provides the model with implicit information about disparity between levels of theory which allows it to benefit more from the inclusion of low level target data, compared to the ST model.

Another challenge of the Δ method is the imposition of hierarchy in levels of theory. When more than two levels of theory are used for training, a decision must be made on which pairs of levels to train difference models on. In many cases, it is not obvious how to order levels of theory, but as Figure 5-6 shows, even when a reasonable ordering exists, it may not produce the best outcome. In both subfigures, we compare MAE versus cost for a Δ model which assumes a hierarchy that is reasonable based on the Jacob's ladder framework as well as the structural relationship of the DFAs and a model based on a scrambled version of the hierarchy. The left subfigure uses PBE$\rightarrow$PBE0$\rightarrow$CCSD(T) as the conventional hierarchy and switches

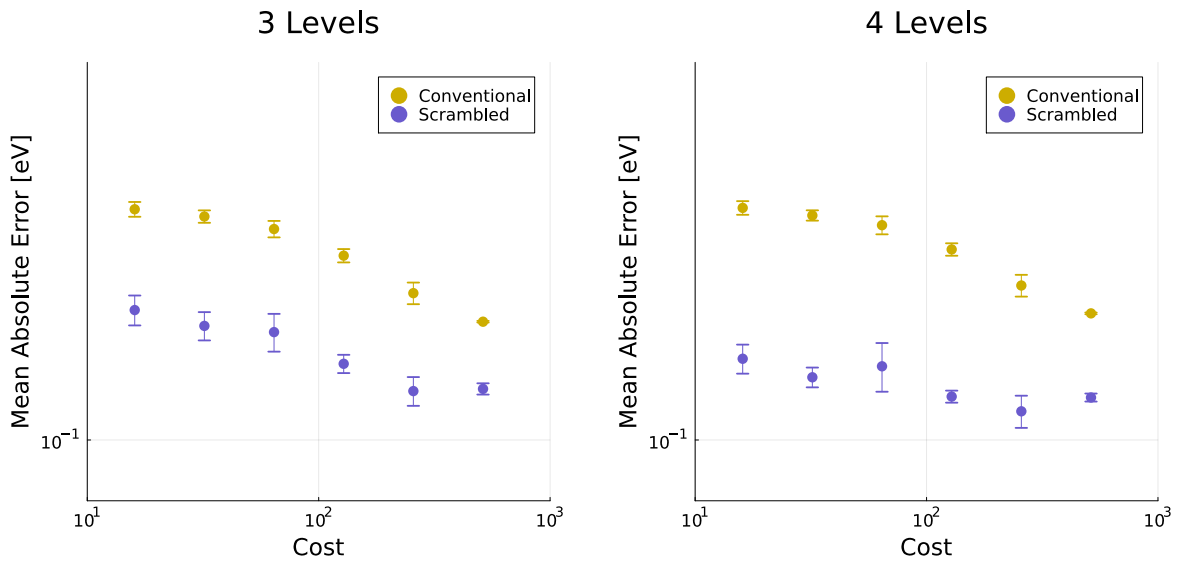


Figure 5-6: **Organic Molecules Case.** Comparison of implementations of the Δ method which use a "conventional" model ordering (gold) compared to a scrambled version of the ordering (indigo). The error bars are standard error computed based off three draws of the data set.

the order of PBE and PBE0 to create a scrambled alternative. The right subfigure takes PBE→PBE0→PBE0_DH→CCSD(T) to be a conventional hierarchy and scrambles this into PBE0_DH→PBE→PBE0→CCSD(T). Bars on each data point show standard error from three tests where molecular configurations where randomly assigned to training and target sets.

In both cases, the implementation with a scrambled hierarchy displays a clear lead over the conventional version. Because absolute error of the final prediction depends only on absolute error of the predicted difference, the lead in performance of the scrambled ordering is not a consequence of adding the difference model to a more accurate baseline. For more insight on the data sets used for these results see Appendix B. Additionally, note that the results for the Δ method in its comparison with the multitask method in Figure 5-4 are produced with the scrambled ordering to show the method at its best.

While the Δ method offers performance advantages in this setting, its rigidity may make it impossible to apply to a given training data set or may lead to less than

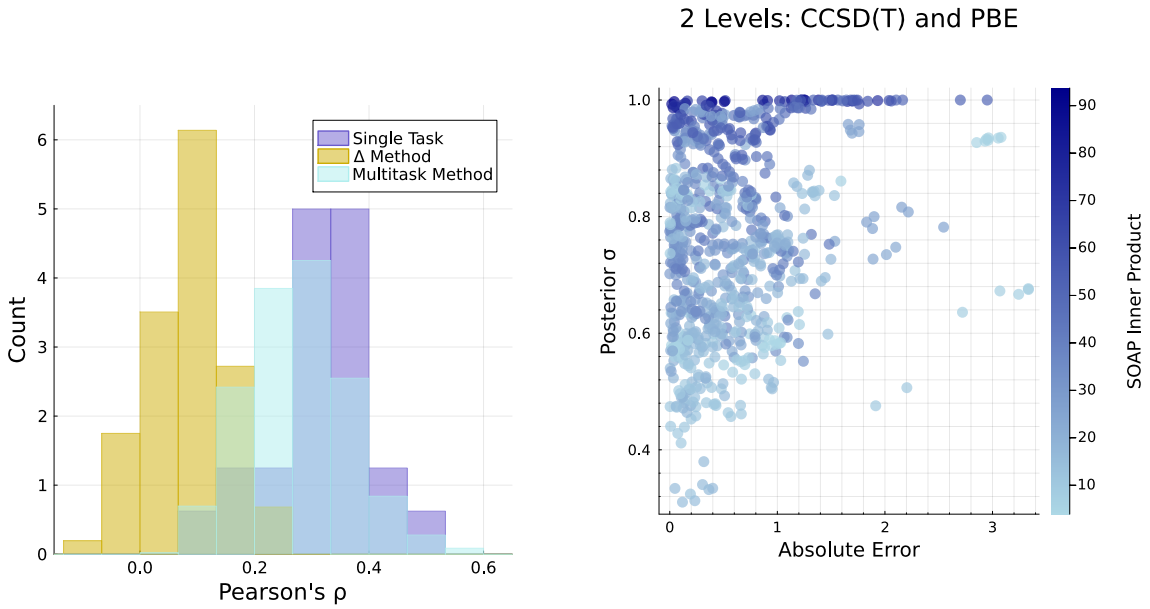
optimal results. A practitioner should seek to use both methods to their advantage, and developers may pursue models which combine these advantages.

Chapter 6

Challenges for Multitask Approach: Variance Prediction

We now turn our attention to variance predictions of Gaussian process based inference models. In a well-specified setting, the posterior σ^2 describes how far true realizations of the quantity of interest will generally deviate from the posterior mean. When we use GP inference models to predict deterministic quantities—such as a CCSD(T) calculation for a given molecule—the posterior represents how far the fixed truth may reasonably be from the predicted mean. The model determines σ^2 from an initial estimate of the uncertainty that it corrects based on how informative the training data is judged to be for a particular prediction task. Evaluation of uncertainty depends highly on kernel parameter estimation and feature construction. Ideally, uncertainty indicators accurately describe when a prediction can be trusted.

To assess the reliability of uncertainty indicators, we may consider their relationship to the error of the posterior mean against target data. Figure 6-1 (a) shows the distribution of Pearson's correlation coefficient between error in the prediction of 500 target ionization potentials and the corresponding posterior standard deviation predictions by a single task conventional GP model, the Δ method, and the multitask method. These distributions result from different constructions of C, S, and T (as defined in Figure 4-5) obtained by varying the sizes of C and S and by using three different random assignments of molecules to the C, S, and T sets. Though the



(a) (b)

Figure 6-1: **Organic Molecules Case.** Relationship of posterior σ to absolute error of posterior mean. (a) The distribution of Pearson's correlation coefficient between prediction error and posterior σ for various iterations of C, S, and, T. Indigo corresponds to a model trained on CCSD(T) only, gold represents the results of the Δ method, and light blue represents the multitask approach. (b) Posterior σ for target molecular configurations plotted against data. The color scale corresponds to the magnitude of the inner product of the SOAP feature of the molecular configuration.

three distributions show a slight positive bias, there is no evidence of a significant correlation between posterior σ and inference model prediction error. The correlation coefficients of the Δ method tend to be smaller than those of the other two approaches, and the construction of the method for more than two levels of theory may explain this lag. An inference model is trained for each pair of levels, and a final prediction is made by summing pair predictions. Since we can expect all levels of theory to be correlated, we cannot expect the variance of the sum of pair predictions to be the sum of pair variances.

Pearson's correlation coefficient is limited to the detection of linear correlations. An uncertainty indicator can still be useful if it can be mapped to a nontrivial upper bound to error. If such an indicator is below some threshold, we can trust that our posterior mean predictions have low error, otherwise we ought to retrain our

model because there is risk of high error. We qualitatively check the behavior of posterior σ predictions in Figure 6-1 (b) by plotting them against the absolute error of posterior mean predictions for individual target molecular configurations. Coloration corresponds to the magnitude of the inner product of each configuration's SOAP descriptor. By our construction, it is no surprise that darker blue corresponds to larger σ. In general, we see that as error increases, the minimum corresponding posterior σ value increases, but there are several points in the lower right hand side triangle of this plot which make it challenging to map posterior σ to a useful upper bound on error. This result is representative of various implementations of GP based inference models that we have tested. Posterior σ values are not robust indicators of uncertainty.

Sources of uncertainty and error in our system can be categorized into design inadequacy and data limitations. The former category deals with mispecified modeling assumptions: the joint Gaussianity of the training and target quantities of interest, the kernel and parameters of the covariance model, the feature representation of inputs, and the homoscedastic noise model. These assumptions are too simplistic to fully capture the behavior of electronics structure calculations. For the GP based models discussed in this work, the posterior σ has no mechanism for translating the error caused by these assumptions into uncertainty indicators.

Uncertainty due to data limitations includes remaining parameter estimation error after parameter design inadequacy is taken into account and insufficient training data coverage of our targets. We may consider representing the former separately by methods designed for parameter estimation uncertainty quantification or by a fully Bayesian approach to parameter inference. The question of whether training data coverage is "sufficient" depends on whether our feature representation can determine if our target data is far from our training data. We expect greater uncertainty if we must interpolate sparse data or if we must extrapolate. Provided that our feature representation is reasonable, the posterior variance is designed to capture this type of uncertainty.

Figure 6-1 (b) suggests that the posterior σ predictions are not without meaning

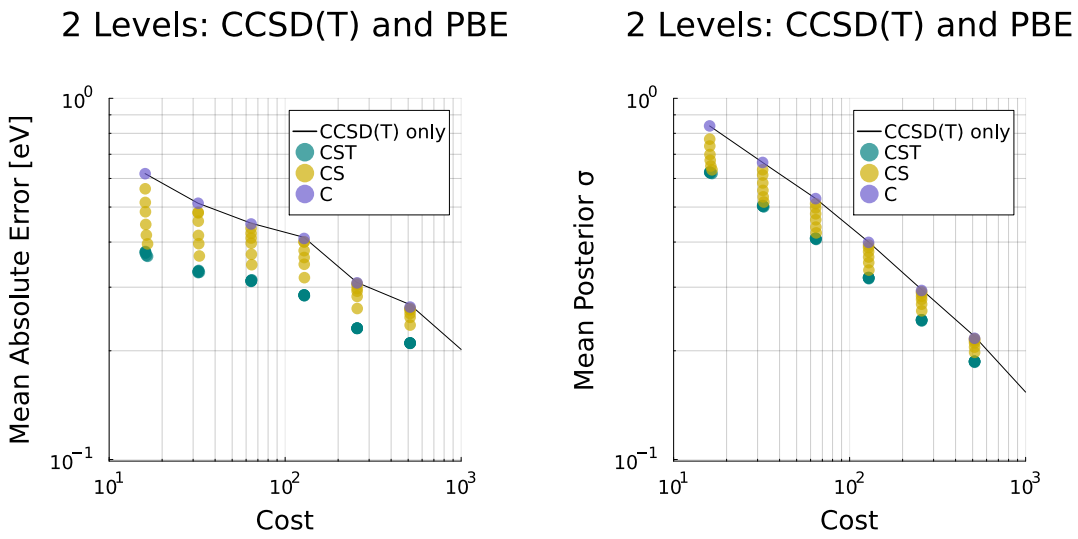


Figure 6-2: **Organic Molecules Case.** The left plot shows MAE versus cost for a multitask method, trained on CCSD(T) and PBE. The right plot compares mean posterior σ to cost for the same tests which appear in the left plot. The indigo points correspond to inference models trained on only molecules from the core set, the gold points correspond to models trained on both C and S sets, and the teal points correspond to models trained on molecules from the C, S, and T sets. The GP inference results for a training set of CCSD(T) only are plotted as a black line.

since the majority of points cluster in the upper left corner, the behavior we would expect of an upper bound in error. To find the extent of their meaning, we must look to their design. The comparison in Figure 6-2 shows that the average posterior σ values for models with different training set constructions (right subfigure) qualitatively match the trends of the mean absolute error of the models (left subfigure). Each point on the subfigures corresponds to the average result of three random assignments of the training data set for a multitask model with one secondary task informed by PBE, and the color indicates what combination of C, S, and T informed the secondary training set. The line provides the average MAE from three implementations of a single task GP model at each cost. Both plots show the same stratification of performance by training set structure: C trained models perform as well as the GP models but worse than the CS models which in turn perform worse than the CST models.

Part of the reason that the average over multiple posterior σ for a given model works as an uncertainty indicator for that model is that averaging smooths over

difficult points that prevent σ from functioning as an uncertainty indicator on a molecule by molecule basis. The difficult points in question would be the predictions in Figure 6-1 (b) where the uncertainty is smaller than the error. It is also important to note that Figure 6-2 compares models with different training set sizes. The more data that a model has to make an inference, the narrower its predicted posterior distributions will be. This rule clearly does not account for all behavior in Figure 6-2: we can note that C trained models use training sets twice as large as the single task cases with comparable cost, yet for a given cost, these models have nearly the same accuracy and average posterior σ. Furthermore, many CST trained models have error and posterior standard deviation lower than C and CS trained models with larger training sets. The explanation may be our choice of features and covariance function. Both DFT and CCSD(T) predictions for the same molecular configuration are paired with the same SOAP feature. The covariance structure for secondary task data is distinct from but correlated with the covariance structure for the primary task, as shown by (3.7). Consequently, secondary C data is judged by the model to be minimally informative, and secondary T data—with a feature matching our target configuration—is very informative. The design of multitask model contributes to the ability of the average posterior σ to distinguish the uncertainty resulting from different training data set constructions.

We can also demonstrate that posterior σ can identify outliers in chemical space. As it can be difficult to define an outlier in our small organic molecules data set, we consider an alternate data set with clear geometric descriptors—specifically, we train a GP model on energy differences between water monomers, all with an HOH angle of 108° and with a range of OH distances from 0.7 Å to 1.8 Å. We define the "monomer stretch" as

$$\big(OH_1(B) - OH_1(A)\big) \; + \; \big(OH_2(B) - OH_2(A)\big)$$

where OH_i(A) is the i^{th} OH distance of monomer A. Thus, we can put all monomers

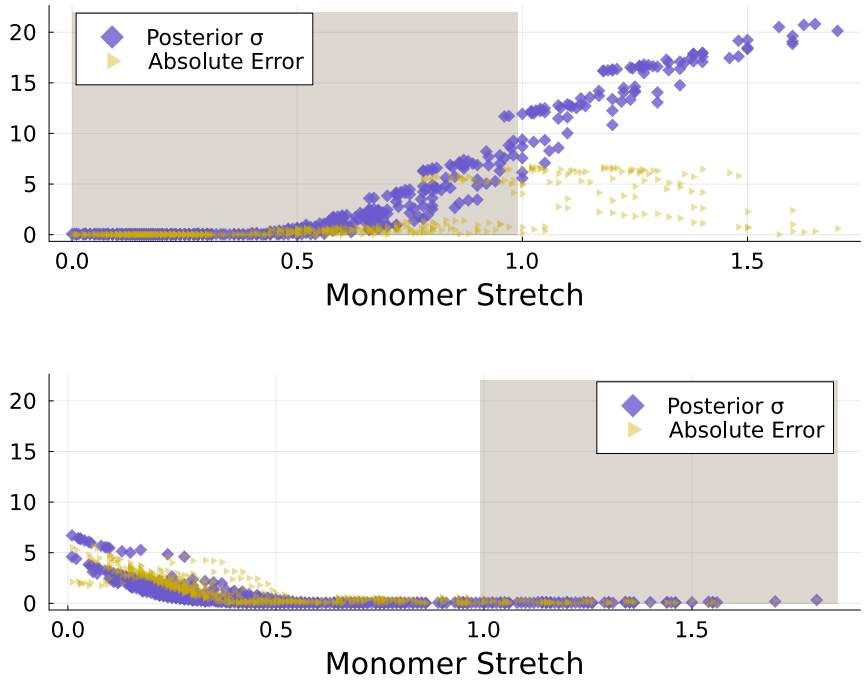


Figure 6-3: **Water Monomers Case.** Comparison of absolute error and posterior σ for an extrapolation task. Statistical models are trained to predict energy differences between two water monomers and the x axis gives the stretch between monomers for a given prediction task. Training molecules are drawn only from the shaded region of each plot. Indigo points represent the predicted σ, and gold points represent absolute error.

on a one dimensional continuum. For more details on the construction of these tests, see Appendix C.

Figure 6-3 shows the impact on posterior mean and σ when a model is trained on half of the continuum. In both subfigures, training configurations are drawn only from the shaded region, and target molecules are drawn from the entire continuum. For each target, the absolute prediction error is plotted against the monomer stretch as a gold triangle, and the posterior σ is plotted as an indigo diamond. On both subfigures, as we move from the interpolation region to the extrapolation region, prediction error increases, and posterior σ marks the monomer stretch where error begins to increase. Note that when monomer stretch is large, our quantity of interest is the energy difference between a monomer with relative long OH bond lengths and one with small OH bond lengths. By contrast, when monomer stretch is small, we are

interested in the energy difference between monomers that both have relatively small OH bond lengths. This observation might explain the empirical observation that it is a more challenging task to predict energy differences corresponding to large monomer stretches than small ones. When the training region includes only small monomer stretches, error prediction error increases as monomer stretch becomes large, even before entering the extrapolation region. Conversely, when we train on large monomer stretches, error remains small even in some portions of the extrapolation region close the interpolation region. The posterior σ marks both of these trends. In Appendix C, we find these general trends also exist when relative absolute error is considered.

It is not surprising to find that a GP inference model can identify outliers, but it is useful to confirm this ability for a particular application and model design. In cases of extreme model mispecification, posterior σ may not give any meaningful uncertainty information. From Figure 6-3, we can argue that the relationship between training and target configurations represented by SOAP features is relevant to uncertainty. Additionally, SOAP feature space can capture certain difficult interpolation cases—as in the top subfigure—and easier extrapolation cases—as in the bottom subfigure. Further research can contribute to a fuller understanding of the behavior of these posterior σ uncertainty indicators for applications which suffer from design inadequacy. Such work could ultimately determine how to use the uncertainty captured by these indicators and account for the uncertainty missed.

Chapter 7

Conclusion

Multitask inference offers an efficient, flexible approach to regression on highly accurate primary data supported by multiple secondary data sets without clear hierarchy. Such a method is well-suited to the challenge of leveraging a suite of quantum chemistry calculations to produce new predictions for target molecules. We apply this method to the prediction of the ionization potential of small organic molecules and the interaction energies of water dimers. In both cases, we find that for a given cost the mean absolute error of the multitask method is less than that of a single task GP trained only on the most accurate electronic structure calculations. Furthermore, there is an accuracy benefit to increasing the number of secondary tasks used for training. While in the cases tested, the multitask method falls short of the accuracy of a Δ method for comparable data sets, the multitask method can be applied to data sets where the Δ method is impossible. It can make use of already existing "data sets of opportunity" where there is no overlap in the molecules represented in the primary and various secondary training sets.

These test cases also provide us with insight into variance prediction by GP based methods when training data sets fall short of modeling assumptions. Ideally, the posterior variance would represent a GP model's uncertainty, but we find that predictions do not reliably map to an upper bound GP absolute prediction error, the functionality we want from an uncertainty indicator. Averages of the variance predicted for models trained with different data set constructions do capture trends in mean absolute error

of these models, and the variance can successfully indicate clear cases of predictive extrapolation. The posterior variance retains some ability to identify the sources of uncertainty it was designed to capture, and a question remains of how best to amplify and augment these abilities.

Appendix A

Feature and Kernel Design

This appendix provides additional results from tests described in Chapter 4.

A.1 SOAP Construction

Figures A-1 and A-2 show results from SOAP parameter tests similar to those performed to create Figure 4-1. The three figures are distinguished only by the level of theory used to train the GP models. Data from the PBE functional approximation was used for 4-1 while A-1 and A-2 are produced with CCSD(T) and PBE0 data, respectively. These are the three levels of theory that results in the text rely on most heavily. Together, the figures demonstrate that prediction performance is most sensitive to the SOAP σ_{atom} parameter, and that $\sigma_{atom} = 0.4$ Å provides the best results on average.

The next set of figures corroborate our claim that global SOAP features obtained by averaging local SOAP features perform comparably to features computed through the REMatch approach [9]. In Chapter 4, Figure 4-2 demonstrated that distances computed between pairs of molecules by the averaging approach are highly correlated to those computed with the REMatch method when we choose SOAP parameters $\sigma_{atom} = 0.4$ Å and $r_{cut} = 4$ Å. Because the size of the molecular systems that we consider may motivate us to change the SOAP cutoff radius, here we present comparisons between the globalization methods for when $r_{cut} = 3$ Å (Figure A-3)

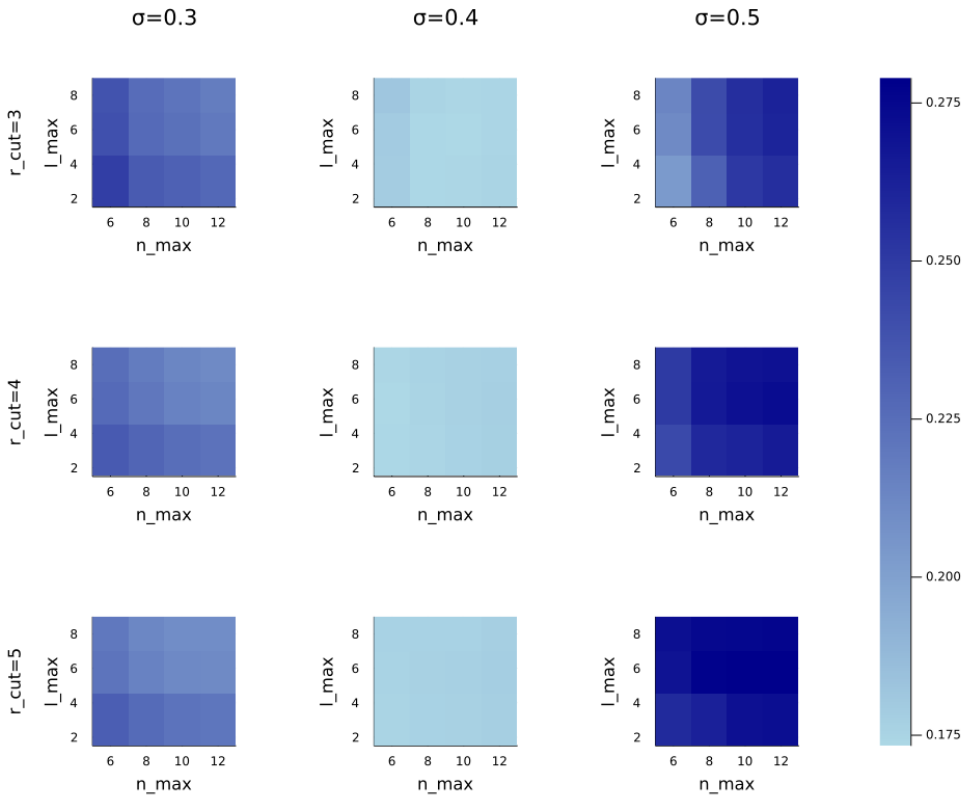


Figure A-1: **Organic Molecules Case.** The impact of SOAP parameters on performance of a GP model trained on CCSD(T) data. The color bar reports mean absolute error.

and $r_{cut} = 5$ Å (Figure A-4). We find more evidence that the average features and REMatch features produce results with strong linear correlation, while global features constructed from species-based averages are less correlated to the other approaches.

A.2 Prediction Calibration of Kernels

In Chapter 4, Figure 4-3 provides a comparison of the impact of the Polynomial and Squared Exponential kernels on the accuracy of the posterior mean predictions of GP models. The chapter also includes some discussion on the role of these kernels in posterior variance prediction based on their mathematical construction. We can also empirically compare the accuracy of the posterior variances predicted by both kernels. We refer to the concept of distribution calibration described by Kuleshov et

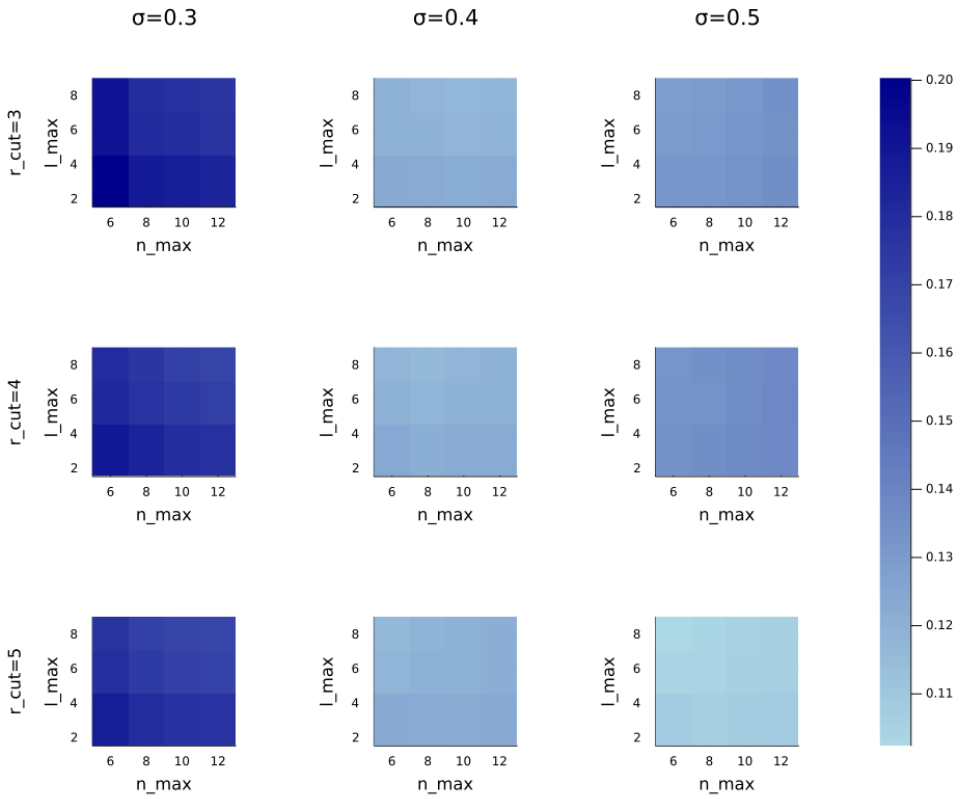


Figure A-2: **Organic Molecules Case.** The impact of SOAP parameters on performance of a GP model trained on PBE0 data. The color bar reports mean absolute error.

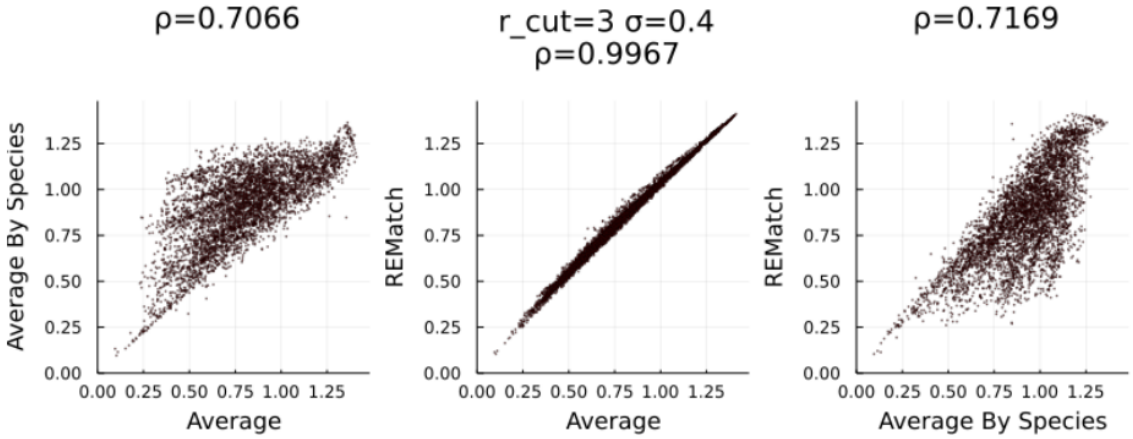


Figure A-3: Comparison of strategies for constructing global SOAP features. The cutoff radius is set to 3 Å.

al. which tests whether predicted distributions have reasonable width on average over the observation data set [20].

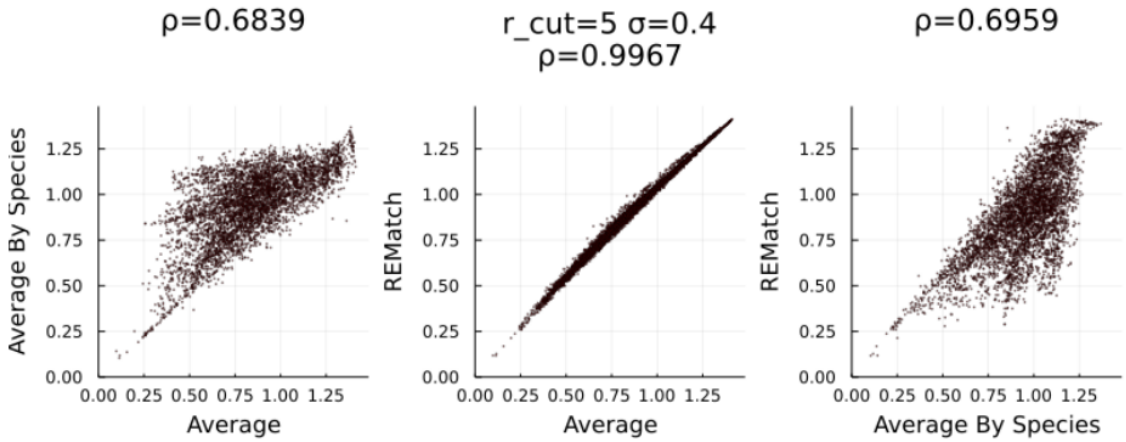


Figure A-4: Comparison of strategies for constructing global SOAP features. The cutoff radius is set to 5 Å.

Let $p \in [0, 1]$. Suppose $\{Y_i\}_{i=1}^n$ is our observation data set. Define

$$R(p) \;=\; \frac{\sum_{i=1}^{n} \mathbb{1}\Big(Y_i \leq F_i^{-1}(p)\Big)}{n}$$

If F_i is the distribution function of Y_i, then $\mathbb{P}\big(Y_i \leq F_i^{-1}(p)\big) = p$. We expect

$$R(p) \;\xrightarrow[n\to\infty]{}\; p$$

For our data set of ionization potentials for small organic molecules and $n = 1000$, we compute $R(p)$ at $0 \leq p \leq 1$, and plot the results in Figure A-5. We find that $R(p)$ values corresponding to the Squared Exponential kernel are generally closer to p than those calculated for the polynomial kernel. Additionally, the steep slope near $p = 0.5$ of the $R(p)$ curve of the polynomial kernel indicates that this kernel generally predicts distributions that are too wide. Likely, this behavior is due to the dependence of the polynomial kernel's posterior variance predictions on the magnitudes of SOAP features, as described in Chapter 4. By contrast, the $R(p)$ curve of the Squared Exponential kernel demonstrates the largest slope near $p = 0$ and $p = 1$, indicating that predicted distributions tend to be too narrow. This issue may be addressed through the kernel's hyperparameter optimization procedure.

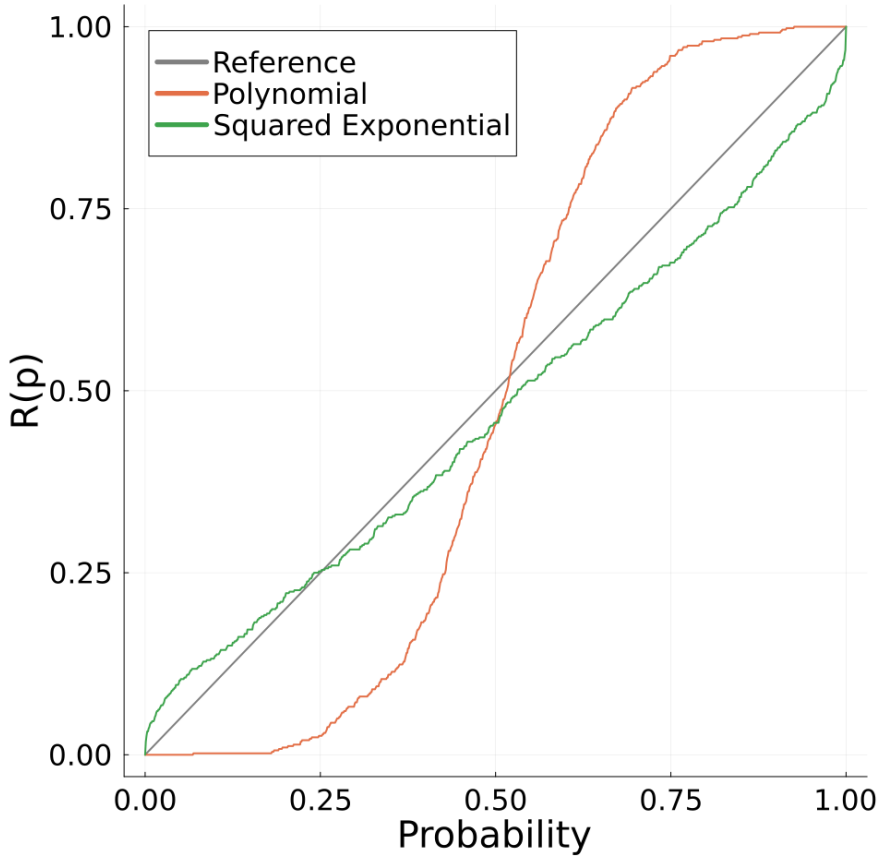


Figure A-5: **Organic Molecules Case.** Comparison between Polynomial and Squared Exponential kernels with respect to the calibration indicator R(p). A result closer to the reference line is better.

Appendix B

Posterior Mean

This appendix expands on the results of Chapter 5.

B.1 Data Set Statistics

Table B.1 presents a comparison of summary statistics describing the CCSD(T) data sets used for our inference examples. All statistics are performed on the absolute value of the CCSD(T) calculations. Ionization potentials are computed for the small organic molecules represented by Figure 4-4, interaction energies are computed for water dimer configurations, and energy differences are taken between 1000 randomly selected pairs of water monomers. The scale of each data set impacts our expectations for successful mean absolute error predictions.

	|Ionization Potentials| [eV]	|Interaction Energies| [eV]	|Energy Differences| [eV]
Minimum	5.894	1.6×10^{-7}	0.0014
Median	9.296	0.0028	1.908
Maximum	15.23	0.2871	6.650
Mean	9.409	0.0076	1.992
σ	0.9446	0.0150	1.338

Table B.1: CCSD(T) Data Set Statistics

	CCSD(T) vs. PBE0_DH	CCSD(T) vs. PBE0	CCSD(T) vs. PBE	PBE0_DH vs. PBE0	PBE0_DH vs. PBE	PBE0 vs. PBE
Pearson's ρ	0.9727	0.9616	0.9380	0.9966	0.9725	0.9845
Spearman's ρ	0.9640	0.9509	0.9135	0.9950	0.9548	0.9744
Kendall's τ	0.8694	0.8371	0.7618	0.9433	0.8232	0.8674

Table B.2: **Organic Molecules Case.** Correlation coefficients between pairs of observation data sets.

	CCSD(T) vs. PBE0	CCSD(T) vs. PBE	PBE0 vs. PBE
Pearson's ρ	0.9985	0.9978	0.9998
Spearman's ρ	0.9955	0.9950	0.9994
Kendall's τ	0.9875	0.9876	0.9914

Table B.3: **Water Dimers Case.** Correlation coefficients between pairs of observation data sets.

	|CCSD(T) − PBE0_DH|	|CCSD(T) − PBE0|	|CCSD(T) − PBE|	| PBE0_DH − PBE0|	|PBE0_DH − PBE|	|PBE0 − PBE|
Minimum	2.7×10^{-4}	3.8×10^{-4}	1.6×10^{-4}	0.0019	4.576e-5	0.0010
Median	0.1181	0.1154	0.3197	0.1381	0.3899	0.2486
Maximum	2.491	2.586	2.808	0.6482	1.319	0.9130
Mean	0.1534	0.1677	0.3817	0.1570	0.4227	0.2666
σ	0.1694	0.2048	0.2895	0.0794	0.2208	0.1495

Table B.4: **Organic Molecules Case.** Summary statistics on the pairwise absolute differences of observation data sets.

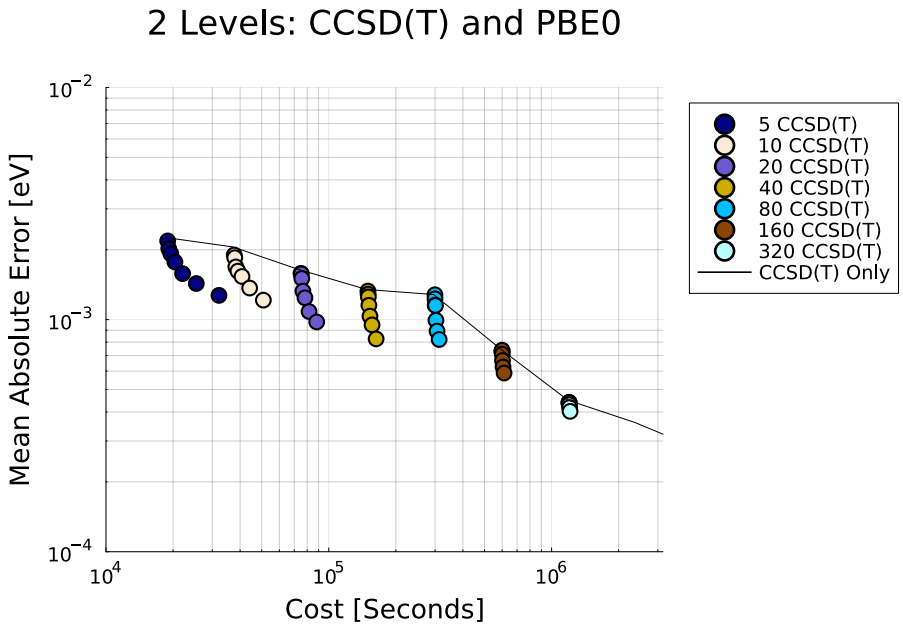
Tables B.2, B.3, and B.4 give some insight into the relationships between observation data sets. Two of these tables report correlation coefficients between the predictions of various pairs of electronic structures methods. Table B.2 lists Pearson's ρ, Spearman's ρ, and Kendall's τ for CCSD(T) and DFA calculations of the ionization potential of small organic molecules. The same correlation coefficients for the water dimers data set can be found in Table B.3. Predictions by different methods are in general highly correlated, and different DFAs are more correlated to each other than to CCSD(T). The correlation coefficients reported for the water dimer data sets are larger than those reported for the ionization potential sets—in fact, these coefficients

are very close to 1. This strong correlation may be because all calculations in these data sets correspond to different configurations of only one molecular system.

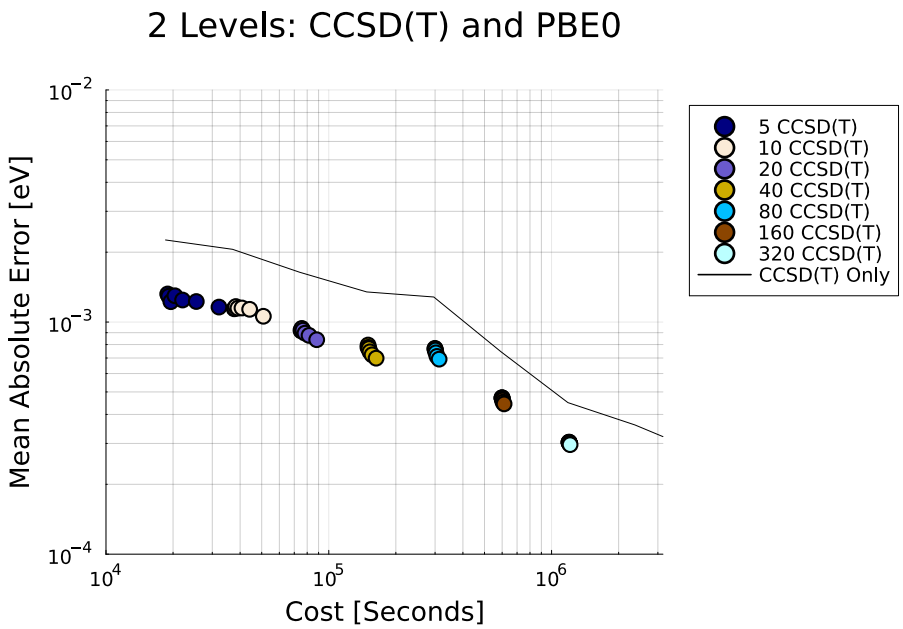
In Chapter 5, we discuss the Δ method's sensitivity to the ordering of observation data sets. For instance, when data sets computed with CCSD(T), PBE0, and PBE were used for inference, the method performed better when it learned difference models $\Delta(CCSD(T), PBE)$ and $\Delta(PBE, PBE0)$ as opposed to $\Delta(CCSD(T), PBE0)$ and $\Delta(PBE0, PBE)$. From Table B.2, we can see that CCSD(T) is less correlated with PBE than PBE0. From the correlation coefficients alone, it remains unclear why the PBE0→PBE→CCSD(T) ordering performs better. We can also consider Table B.4 which provides summary statistics on the absolute values of differences between predictions by various pairs of methods. Unfortunately, a clear indicator for why one ordering of observation data sets yields better performance than another remains elusive. Future work may undertake more detailed analyses to determine why this effect occurs and how widespread it may be.

B.2 Water Dimers Case

Figure B-1 demonstrates the accuracy of multitask models trained on CCSD(T) and PBE0 calculations to predict interaction energies of water dimer configurations. The example is summarized in Section 5.2 where Figure 5-3 shows how multitask models perform when trained on CCSD(T) and PBE data. When Figures 5-3 and B-1 are compared, it is apparent that their results are nearly identical. This finding is believable given the high correlation between PBE0 and PBE data sets given in Table B.3.



(a) No target DFT data



(b) Target DFT data

Figure B-1: **Water Dimers Case.** Scatter points correspond to implementations of the multitask method with a range of data set sizes. The black line shows the accuracy of a GP model trained only on CCSD(T) data. All results are the average MAE values obtained from six random assignments of the data sets.

Appendix C

Posterior Variance

Here, we provide additional results relevant to the discussion in Chapter 6.

C.1 Posterior Distribution and Error

Figure C-1 continues the search for possible patterns in posterior σ behavior initiated by Figure 6-1. It plots posterior mean predictions of the ionization potential of the target molecules against posterior σ predictions and colors these predictions by their absolute error. In the plot, the maximum posterior σ is scaled to 1, and when this maximum is achieved, corresponding posterior mean predictions are close to 9.5. We expect that the prior mean is close to this value because for the posterior σ to reflect maximum uncertainty, the GP model must judge that observation data is uninformative for a given target. If the data is uninformative, the posterior mean will be the same as the prior. Additionally, the prior mean is set to the average of the training data, and consultation with Table B.1 indicates that the overall average of the CCSD(T) predictions for ionization potential is 9.4.

Table B.1 also tells us that the true range of ionization potential values which appears in our data set is wider than the range of posterior mean predictions in Figure C-1. It is more challenging for the model to capture the extreme points in the target data set. While Figure C-1 visualizes the behavior of the GP model's predictions, it does not suggest a staightforward transformation of the posterior mean and σ that

would result in a prediction of absolute error. It is likely that more information would need to be considered alongside these predictions to draw a meaningful conclusion about error.

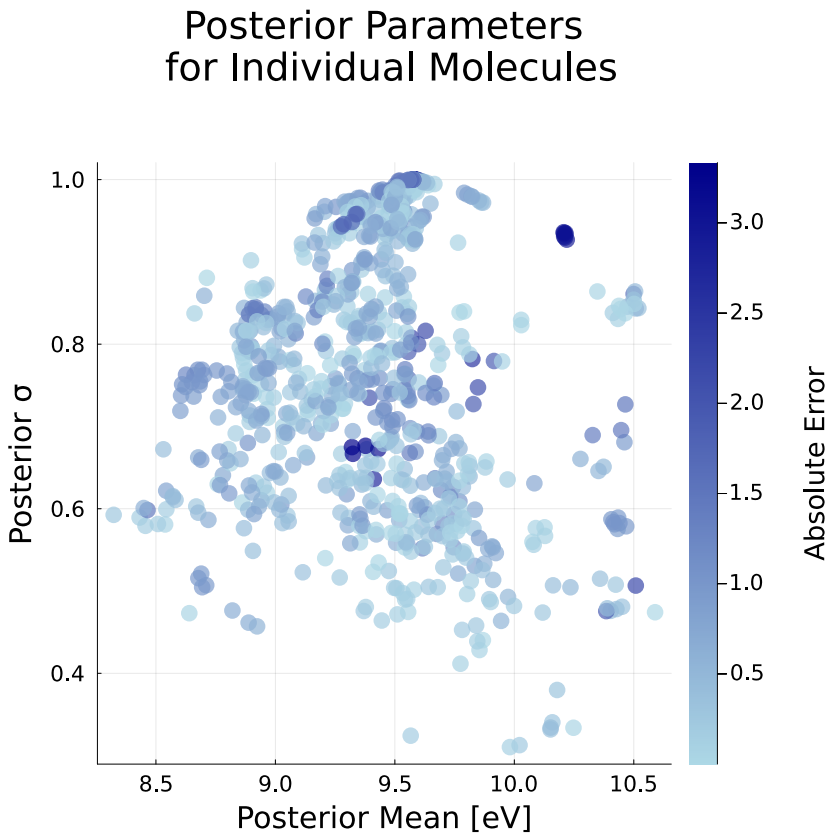


Figure C-1: **Organic Molecules Case.** Mean versus standard deviation predictions of a Gaussian process model. Darker blue points have greater absolute error.

C.2 Extrapolation

The final series of figures supply details for the example of predicting the energy differences between water monomers, introduced by Figure 6-3. The quantity "monomer stretch" is defined in Chapter 6 to simplify visualization of predictions for energy differences between two monomers, one stretched relative to another. This single axis representation leaves out some complexities of the data set. For instance, suppose

that $\{A_i, B_i\}_{i=1}^n$ are the set of monomer pairs for which we compute n energy differences. A_i and B_i each have two OH bonds, and the lengths of all four of these are allowed to vary in our data set. The only restriction is that the bond lengths of B_i must be stretched relative to A_i. Consequently, the true energy differences are not a function of monomer stretch—multiple energy differences may correspond to the same stretch value. The truth also does not vary monotonically as monomer stretch increases.

The lowest subfigures of C-2 and C-3 show the true absolute energy differences for the target monomer pairs when we train on low and high stretch cases, respectively. We can compare these values to the absolute error and posterior σ predictions displayed in Chapter 6 as well as in the top subfigures of these plots. The middle subfigures show the magnitude of relative error in each case. Note that the vertical axes of the plots have different scales. Additionally, two outlying relative error values were left out of C-2 and one was excluded from C-3 so that the vertical axis scales need not be excessively large. Both figures show that relative error tends to be elevated in the same regions where absolute error is elevated. Both plots also show several points with unexpectedly high relative error, for instance, the scatter points with monomer stretch ≈ 0.6 Å in C-3. Likely, these points correspond to monomer pairs with low true absolute energy difference though more work may be necessary to understand the details of this behavior.

Finally, we can again observe that when the GP model is trained on monomer pairs exhibiting large "stretch" it can perform well predicting many cases with small stretch. One explanation may be that this model is trained with energy differences between one monomer with relatively long OH bonds lengths and another with short bond lengths, then use to predict the energy differences between two monomers with relatively short bond lengths. Since the training set has some information about monomers with relatively short bond lengths, this task is a less extreme extrapolation than the reverse.

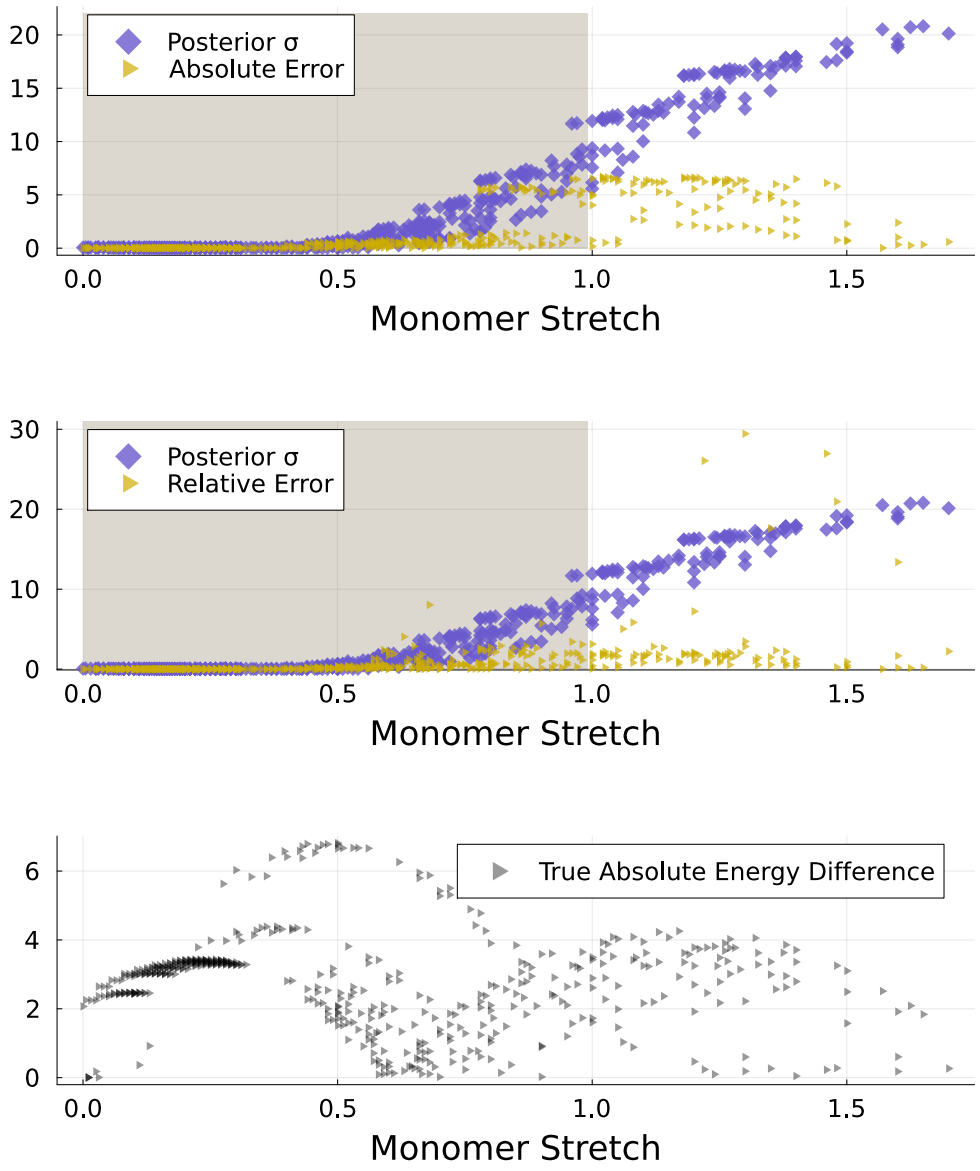


Figure C-2: **Water Monomers Case.** GP model error and posterior σ predictions for energy differences of monomer pairs. Training data is drawn from the shaded region. The lowest subfigure shows the true energy differences.

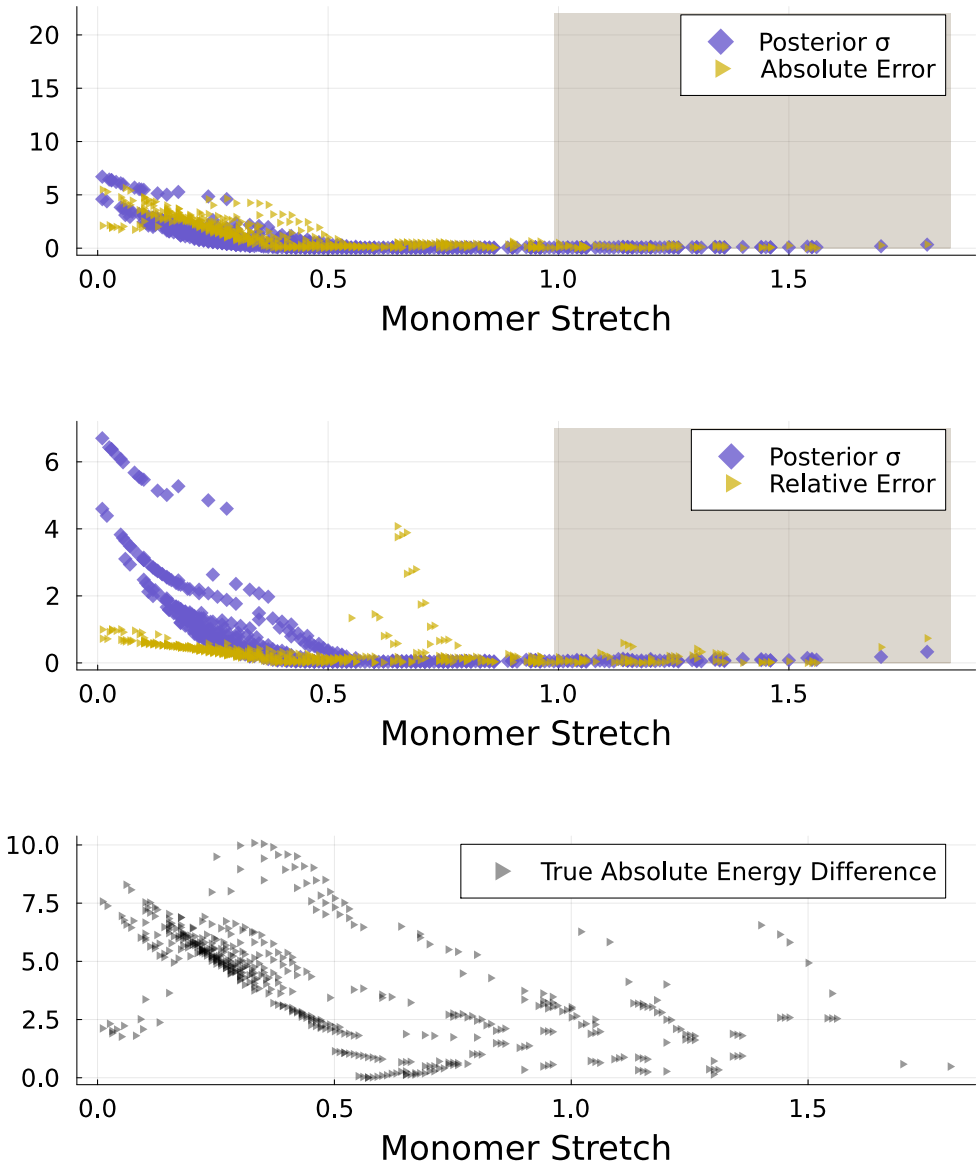


Figure C-3: **Water Monomers Case.** Prediction of energy differences of monomer pairs by a GP model trained on pairs exhibiting large monomer stretch. The true energy differences are reported in the bottom subfigure.

Bibliography

[1] Albert P. Bartók and Gábor Csányi. Gaussian approximation potentials: A brief tutorial introduction. *International Journal of Quantum Chemistry*, 115:1051–1057, 8 2015.

[2] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 87:184115, 5 2013.

[3] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical Review Letters*, 104, 4 2010.

[4] Leonardo S. Bastos and Anthony O'Hagan. Diagnostics for gaussian process emulators. *Technometrics*, 51(4):425–438, 2009.

[5] R. Batra, G. Pilania, B.P. Uberuaga, and R. Ramprasad. Multifidelity information fusion with machine learning: A case study of dopant formation energies in hafnia. *ACS Applied Materials & Inference*, 11:24906–24918, 2019.

[6] E.V. Bonilla, K.M.A Chai, and C.K.I. Williams. *Multi-task Gaussian process prediction*, pages 153–160. MIT Press, Cambridge, Massachusetts, 2008.

[7] R. Christensen, T. Bligaard, and K.W. Jacobsen. Bayesian error estimation in density functional theory. *Uncertainty Quantification in Multiscale Materials Modeling*, pages 77–91, 2020.

[8] B. Civalleri, D. Presti, R. Dovesi, and A. Savin. On choosing the best density functional approximation. In Michael Springborg, editor, *Uncertainty Quantification in Multiscale Materials Modeling*, chapter 6, pages 168–185. RSC Publishing, 2012.

[9] Sandip De, Albert P. Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics*, 18(20):13754–13769, 2016.

[10] V. Deringer, A. Bartók, N. Bernstein, D. Wilkins, M. Ceriotti, and G. Csányi. Gaussian process regression for materials and modelling. *Chemical Reviews*, 121:10073–10041, 2021.

[11] Chenru Duan, Fang Liu, Aditya Nandy, and Heather J. Kulik. Data-driven approaches can overcome the cost–accuracy trade-off in multireference diagnostics. *Journal of Chemical Theory and Computation*, 16(7):4373–4387, 2020.

[12] A. I. J. Forrester and A. J. Keane A. Sóbester. Multi-fidelity optimization via surrogate modelling. *Proceedings of Royal Society A*, 463:3251–3269, 2007.

[13] C. J. García-Cervera, J. Lu, and Y. Xuan. Linear-scaling subspace-iteration algorithm with optimally localized nonorthogonal wave functions for kohn-sham density functional theory. *Physical Review*, 79:115110.1–115110.13, 2009.

[14] L. Goerigkab and S. Grimme. A thorough benchmark of density functional methods for general main group thermochemistry, kinetics, and noncovalent interactions. *Physical Chemistry Chemical Physics*, 13:6670–6688, 2011.

[15] Michael E. Harding, Thorsten Metzroth, Jürgen Gauss, and Alexander A. Auer. Parallel calculation of ccsd and ccsd(t) analytic first and second derivatives. *Journal of Chemical Theory and Computation*, 4(1):64–74, 2008.

[16] M. Herbst. Dftk: A julian approach for simulating electrons in solids. Presented at JuliaCon 2020, 2020.

[17] Lauri Himanen, Marc O. J. Jäger, Eiaki V. Morooka, Filippo Federici Canova, Yashasvi S. Ranawat, David Z. Gao, Patrick Rinke, and Adam S. Foster. DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2020.

[18] https://www.top500.org/lists/top500/2022/11/. Top 500, November 2022, 2022.

[19] M. Kennedy and A O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[20] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. *CoRR*, abs/1807.00263, 2018.

[21] T. Lee, I. Bilionisa, and A. Buganza Tepole. Propagation of uncertainty in the mechanical and biological response of growing tissues using multi-fidelity gaussian process regression. *Computational Methods in Applied Mechanical Engineering*, 359, 2020.

[22] G. Leen, J. Peltonen, and S. Kaski. Focused multi-task learning in a gaussian process framework. *Machine Learning*, 1-2:157–182, 2012.

[23] K. Lejaeghere. The uncertainty pyramid for electronic-structure methods. In Y. Wang and D. L. McDowell, editors, *Uncertainty Quantification in Multiscale Materials Modeling*, chapter 2, pages 41–76. Elsevier Ltd., Atlanta, 2020.

[24] J. J. Mortensen, K. Kaasbjerg, S. L. Frederiksen, J. K. Nørskov, J. P. Sethna, and K. W. Jacobsen. Bayesian error estimation in density-functional theory. *Physical Review Letters*, 95(21):216401.1–216401.4, 2005.

[25] Felix Musil, Andrea Grisafi, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. Physics-inspired structural representations for molecules and materials. *Chemical Reviews*, 121(16):9759–9815, 2021.

[26] J.P. Perdew and K. Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. In *AIP Conference Proceedings*, 2001. Presented at AIP Conference Proceedings 577.

[27] P. Perdikaris, M.Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proceedings of the Royal Society A*, 473, 2017.

[28] G. Pilania, J.E. Gubernatis, and T. Lookman. Multi-fidelity machine learning models for accurate bandgap predictions of solids. *Computational Materials Science*, 129:156–162, 2017.

[29] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.

[30] G. Schofield, J. R. Chelikowsky, and Y. Saad. Using chebyshev-filtered subspace iteration and windowing methods to solve the kohn-sham problem. *Practical Aspects of Computational Chemistry I*, pages 167–181, 2011.

[31] J. Smith, O. Isayev, and A. Roitberg. A data set of 20 million calculated off-equilibrium conformations for organic molecules. *Scientific Data*, 4(170193), 2017.

[32] J. Toulouse. Review of approximations for the exchange-correlation energy in density-functional theory. In E. Cancés, G. Friesecke, and L. Lin, editors, *Density Functional Theory*, pages 1–63. 2021.

[33] J. Wellendorff, K. T. Lundgaard, K. W. Jacobsen, and Thomas Bligaard. mbeef: An accurate semi-local bayesian error estimation density functional. *Uncertainty Quantification in Multiscale Materials Modeling*, pages 77–91, 2014.

[34] L. Ying, J. Yu, and L. Ying. Numerical methods for kohn–sham density functional theory. *Acta Numerica*, 28:405–539, 2019.

[35] Qi Yu, Chen Qu, Paul L. Houston, Riccardo Conte, Apurba Nandi, and Joel M. Bowman. q-aqua: A many-body ccsd(t) water potential, including four-body interactions, demonstrates the quantum nature of water from clusters to the liquid phase. *The Journal of Physical Chemistry Letters*, 13(22):5068–5074, 2022. PMID: 35652912.

[36] Y. Zhou, Y. Saad, M. L. Tiago, and J. R. Chelikowsky. Self-consistent-field calculations using chebyshev-filtered subspace iteration. *Journal of Computational Physics*, 219(1):172–184, November 2006.