UNIVERSITY OF ICELAND

Ph.D. Dissertation
in Physical Chemistry

Efficient Exploration of Chemical Kinetics

Development and application of tractable Gaussian Process Models

Rohit Goswami

October 2025

FACULTY OF PHYSICAL SCIENCES

Efficient Exploration of Chemical Kinetics

Rohit Goswami

Dissertation submitted in partial fulfillment of a
Philosophiae Doctor degree in Physical Chemistry

Ph.D. Committee
Hannes Jónsson
Birgir Hrafnkelsson
Morris Riedel
Egill Skúlason
Thomas Bligaard

Opponents
Sigurður I. Erlingsson
Normand Mousseau

Faculty of Physical Sciences
School of Engineering and Natural Sciences
University of Iceland
Reykjavik, October 2025

Abstract

Spatio-temporal control of chemical systems to tune relative rates of competing reactions has been the goal of chemistry since early alchemy. Today, the estimation of the products and rates of chemical reactions as well as the stability of chemicals and materials are fundamental tasks for the chemical industry. Despite leaps in mathematical modeling, with insightful representations of electronic structure to describe many body quantum systems, and inspite of exascale computing resources, efficient methods for determining reaction rates in large scale simulations has remained out of reach. Direct simulation of atomic dynamics is limited by short timescale and small length scale. Recently, there has been rapid advance in the generation of machine learned potential functions, but they require large data sets as input and are not practical when the task is to quickly screen thousands of chemicals or materials to identify optimal candidates for technological applications. They have, furthermore, been limited so far to regions of stable configurations of the atoms and are not reliable for the transition state regions which are needed for estimating reaction rates. Attempts to explore reaction networks in an automated manner at sufficient accuracy suffer from the large computational cost of the electronic structure calculations. Simplifying approximations for rate calculations recognise that reactions represent slow processes on the time scale of atomic vibrations and thermal equilibration, and make use of statistical approximations for chemical rate calculations. In the simplest approximation, the harmonic approximation to transition state theory, they boil down to finding first order saddle points on the energy surface describing how the system's energy depends on the position of the atoms. Even so, the computational effort in saddle point searches is prohibitively large in many cases especially when the energy and atomic forces are obtained from electronic structure calculations. Surrogate model based acceleration of saddle point searches have been described as promising for almost a decade now, but in practical terms have remained crippled by large computational overhead and numerical instabilities that negate the advantage in wall time.

This dissertation presents a solution based on a holistic approach that co-designs the physical representation, statistical model, and systems architecture. This philosophy is embodied in the Optimal Transport Gaussian Process (OT-GP) framework, which uses a physics-aware representation based on optimal transport metrics to create a compact and chemically relevant surrogate of the potential energy surface. This defines a statistically robust approach and uses targeted sampling to reduce the computational effort. Alongside rewrites for the EON software for long timescale simulations, we present a reinforcement-learning approach for the minimum-mode following method when final state is not known and nudged elastic band method when both initial and final state are specified. Collectively, these advances establish a representation-first, service-oriented paradigm for chemical kinetics simulations. The success of this paradigm is demonstrated through large-scale benchmarks where the framework shows state of the art performance characteristics, validated with Bayesian hierarchical models. By delivering a framework for high performance open-source tooling, this work transforms a long-held theoretical promise into a practical engine for exploring chemical kinetics.

Ágrip

Stjórnun efnakerfa í rúmi og tíma til að hafa áhrif á samverkandi efnahvörf hefur verið markmið efnafræðinnar allt frá dögum gullgerðarlistarinnar. Í dag er mat á afurðum og hraða efnahvarfa, ásamt mati á stöðugleika efna og efniviða, grundvallarverkefni í efnaiðnaði. Þrátt fyrir stökk í stærðfræðilegri líkanagerð, með nákvæmum lýsingum á rafeindaskipan til að lýsa fjöleinda skammtafræðikerfum, og þrátt fyrir aðgengi að stórauknu reikniafli (exascale), vantar enn skilvirkar aðferðir til að ákvarða hvarfhraða í stórum hermunum. Bein hermun á gangverki atóma takmarkast af stuttum tímaskala og litlum lengdarkvarða. Nýlega hefur orðið hröð framþróun í gerð véllærðra mættisfalla (machine learned potential functions), en þær krefjast stórra gagnagrunna sem inntaks og eru ekki hagnýtar þegar verkefnið er að skima hratt í gegnum þúsundir efna eða efniviða til að finna bestu kandídatana fyrir tæknilega nýtingu. Þær hafa ennfremur hingað til takmarkast við svæði þar sem atómin eru í stöðugri uppröðun og eru ekki áreiðanlegar fyrir hvarfástönd (transition state regions) sem ákvarða að miklu leiti hvarfhraðann. Tilraunir til að kanna hvarfanet á sjálfvirkan hátt með nægilegri nákvæmni fela í sér of háan kostnað við reikninga á rafeindaskipan. Einfaldandi nálganir fyrir hraðaútreikninga gera ráð fyrir því að efnahvörf séu hægir ferlar miðað við titring atómanna svo að varmalegt jafnvægi náist og nýta því tölfræðilegar nálganir fyrir útreikninga á hvarfhraða. Í einföldustu nálguninni, kjörsveifilsnálgun (harmonic approximation) við virkjunarástandskenninguna (transition state theory), snúast þær um að finna fyrsta stigs söðulpunkta á orkuyfirborðinu sem lýsir því hvernig orka kerfisins er háð staðsetningu atómanna. Jafnvel þá er reikniþörfin við leit að söðulpunktum of mikil í mörgum tilfellum, sérstaklega þegar orka og atómkraftar eru fengnir úr reikningum á rafeindaskipaninni. Hröðun á söðulpunktaleit byggð á staðgengilslíkönum (surrogate models) hefur verið lýst sem vænlegri í nærri áratug, en hefur í reynd verið hömluð af mikilli yfirbyggingu og tölulegum óstöðugleika sem gera að engu ávinninginn í rauntíma.

Þessi ritgerð kynnir lausn sem byggir á heildrænni nálgun á þessu verkefni sem samþættir hönnun á eðlisfræðilegri framsetningu, tölfræðilegu líkani og kerfisarkitektúr. Þessi hugmyndafræði birtist í Optimal Transport Gaussian Process (OT-GP) umgjörðinni, sem notar eðlisfræðilega meðvitaða (physics-aware) framsetningu byggða á mælikvörðum fyrir bestun flutnings (optimal transport) til að búa til þjappaðan og efnafræðilega viðeigandi staðgengil fyrir stöðuorkuyfirborðið. Þetta skilgreinir tölfræðilega trausta nálgun og notar markvissa sýnatöku til að draga úr reikniþörfinni. Samhliða endurskrifun á EON hugbúnaðinum fyrir hermun á löngum tímaskala, er sett fram styrktarnámsnálgun (reinforcement-learning) fyrir lágháttarfylgni (minimum mode following) aðferðina þegar lokaástand er ekki tiltekið og hnikateygjubands (nudged elastic band) aðferðina þegar bæði upphafs- og lokaástand eru tilgreind. Samanlagt marka þessar framfarir nýja hugmyndafræði fyrir hermun á efnahvörfum sem byggir á framsetningunni fyrst (representation-first) og er þjónustumiðuð (service-oriented). Árangur þessarar aðferðafræði er sýndur með stórum viðmiðunarprófunum sem sýna góða frammistöðu, greinda með líkönum Bayes. Með því að þróa aðferð fyrir afkastamikil opinn-hugbúnaðar (open-source) verkfæri, umbreytir þessi vinna gömlu fræðilegu loforði í hagnýta tæki til að kanna gang og hraða efnahvarfa.

Dedication

To anyone who still wants to write a monograph, and to my family and pets, for waiting longer for my own.

Preface

This document serves as a monograph that complements the research which forms my doctorate. By design, all text and figures within this thesis are original and do not appear in the associated papers.

This work is therefore not a standalone repository of those publications but is specifically intended to be read in conjunction with them. The chapters herein provide a guiding narrative, expanded theoretical context, software design, and novel analysis that link, support, and build upon the findings presented in my published articles. The reader is encouraged to consult the primary papers for the detailed methodologies and core results, using this thesis as a companion for deeper integration and supplementary insight.

I have always known I would like to be an academician. My preferred field of study has been a bit fluid though. For academia as a whole, I've been waiting since I was six pontificating on my father's manuscript. My mom has always fanned the flames of academic curiosity, as my sister races ahead with brilliant deductions. For computers, never trusting to learn as a trade what I enjoy-I started messing with machines in high school. Computational chemistry was a bit of a late interest, mostly when I realized I thought I knew better. I have worn many hats over the years, software engineer, editor, reviewer, author. This is a short demi-monograph, but it has truly been a lifetime in the making.

Table of Contents

List of Figures

List of Tables

List of Publications

Publications directly related to the thesis

Paper I: Čertík, Ondřej, Pask, John E., Fernando, Isuru, **Rohit Goswami**, Sukumar, N., Collins, Lee. A., Manzini, Gianmarco, and Vackář, Jiří, 2023, High-Order Finite Element Method for Atomic Structure Calculations. *Computer Physics Communications*, Vol. 315, pp. 109051. Accessed at `https://dx.doi.org/10.1016/j.cpc.2023.109051`. Rohit finalized the code, reproducibly generated figures and revised the article.

Paper II: **Rohit Goswami**, Maxim Masterov, Satish Kamath, Alejandro Pena-Torres, and Hannes Jónsson, 2025, Efficient Implementation of Gaussian Process Regression Accelerated Saddle Point Searches with Application to Molecular Reactions. *Journal Chemical Theory and Computation*, Vol. 1, Issue 2, pp. 1–10. Accessed at `https://dx.doi.org/10.1021/acs.jctc.5c00866`. Rohit and Maxim led the development of the C++ code. Rohit performed calculations, created figures and wrote the article.

Paper III: **Rohit Goswami**, 2025, Bayesian Hierarchical Models for Quantitative Estimates for Performance Metrics Applied to Saddle Search Algorithms. *AIP Advances*, Vol. 15, Issue 8. Accessed at `https://dx.doi.org/10.1063/5.0283639`. Rohit developed and validated the models described, wrote the paper, and made the figures.

Paper IV: **Rohit Goswami**, Hannes Jónsson, 2025, Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches. *ChemPhysChem*. Accessed at `https://doi.org/10.1002/cphc.202500730`. Rohit developed the OTGPD algorithm, ran simulations, made figures, and wrote the article.

Other publications submitted during the doctorate

Paper A: Laurence Kedward, Balint Aradi, Ondrej Certik, Milan Curcic, Sebastian Ehlert, Philipp Engel, **Rohit Goswami**, Michael Hirsch, Asdrubal Lozada-Blanco, Vincent Magnin, Arjen Markus, Emanuele Pagone, Ivan Pribec, Brad Richardson, Harris Snyder, John Urban, and Jeremie Vandenplas, 2022, The State of Fortran. *Computing in Science & Engineering*. Accessed at `https://dx.doi.org/10.1109/MCSE.2022.3159862`.

Paper B: Moritz Sallermann, Amrita Goswami, Alejandro Peña-Torres, and **Rohit Goswami**, 2025, Flowy: High Performance Probabilistic Lava Emplacement Prediction. *Computer Physics Communications*, Vol. 315, pp. 109745. Accessed at `https://dx.doi.org/10.1016/j.cpc.2025.109745`

Paper C: **Rohit Goswami**, Ashwini Kumar Rawat, Sonaly Goswami, and Debabrata Goswami, 2025, Compositional Analysis of Fragrance Accords Using Femtosecond Thermal Lens Spectroscopy. *Chemistry – an Asian Journal*, e00521. Accessed at `https://dx.doi.org/10.1002/asia.202500521`.

Publications submitted before the doctorate

Paper 1: Prerna, **Rohit Goswami**, Atanu K. Metya, S. V. Shevkunov, and Jayant K. Singh, 2019, Study of Ice Nucleation on Silver Iodide Surface with Defects. *Molecular Physics*, pp. 1–13. Accessed at `https://dx.doi.org/10.1080/00268976.2019.1657599`.

Paper 2: **Rohit Goswami**, Amrita Goswami, and Jayant Kumar Singh, 2020, d-SEAMS: Deferred Structural Elucidation Analysis for Molecular Simulations. *Journal of Chemical Information and Modeling*. Accessed at `https://dx.doi.org/10.1021/acs.jcim.0c00031`.

Paper 3: Ligesh Theeyancheri, Subhasish Chaki, Nairhita Samanta, **Rohit Goswami**, Raghunath Chelakkot, and Rajarshi Chakrabarti, 2020, Translational and Rotational Dynamics of a Self-Propelled Janus Probe in Crowded Environments. *Soft Matter*. Accessed at `https://dx.doi.org/10.1039/D0SM00339E`.

Glossary

Armijo line search Method for step size selection in optimization to ensure sufficient decrease. xxvii

Cholesky decomposition Matrix decomposition into lower and upper triangular matrices, used for solving systems. xxvii

credible interval Bayesian equivalent of a confidence interval; range of parameter values with given probability. xxvii

data pruning Technique to reduce dataset size by removing less informative or distant data points. xxvii

deflated standard error Standard error estimate that is artificially small, typically due to ignoring data correlations or non-independence. Leads to overconfident statistical conclusions.. xxvii, 56

eigenvalue Scalar associated with an eigenvector; quantifies curvature along that direction. xxvii

eigenvector Vector indicating a principal direction in a matrix, often used in optimization. xxvii

extrema Minima, maxima, saddle points. xxvii, 13, 33

fixed effect Model term estimating the effect of predictors assumed to be constant across all groups. xxvii

Gaussian Process A collection of random variables, any finite number of which have a joint multivariate normal (Gaussian) distribution; defines a distribution over functions.. xi, xix, xxvii, 7, 15, 16, 18, 19, 27, 43, 59–61, 63–66, 75, 77, 86, 89, 93

Gaussian Process Regression Non-parametric Bayesian approach for regression and function approximation. xxvii, 18

Generalized Linear Mixed Model Statistical model incorporating both fixed and random effects. xxvii, 56

Gram-Schmidt process Procedure for orthonormalizing a set of vectors. xxvii

Hessian Second derivative matrix of a function, used to analyze curvature and stationary points. xxvii

homoscedasticity Statistical property where the variance of residuals remains constant across all observations. xxvii, 56

hyperparameters Parameters governing model behavior but not directly optimized during training. xxvii

kernel function Function defining similarity between data points in GPR; determines covariance structure. xxvii

likelihood Probability of observed data given specific model parameters. xxvii

marginal likelihood Evidence; probability of observed data under a model, integrating over parameters. xxvii

non-independence Condition where observations are correlated or related, violating the assumption of independent samples. xxvii, 56

posterior distribution Probability distribution representing updated beliefs after observing data. xxvii

Potential Energy Surface Multidimensional surface representing energy as a function of geometry. xvi, xxvii, 2, 3, 5, 6, 9, 12, 13, 15, 16, 18, 19, 21, 33, 38, 39, 45, 48, 56, 58, 74, 77, 85, 93

prior distribution Initial probability distribution before observing data. xxvii

pseudoreplication Error in statistical inference caused by treating non-independent observations as independent. xxvii, 56

Radial Basis Function Common kernel function in GPR, based on distance between points. xxvii

random intercept Model term allowing each group (e.g., a chemical system) to have its own baseline value. xxvii

robust Describes an estimator or method that maintains performance even when assumptions are violated. xxvii, 57

saddlepoint Point of zero force, with a single negative eigenvalue of the Hessian. xxvii, 14

standard error Estimate of the variability of a sample statistic, often used to quantify uncertainty in parameter estimates. xxvii

variance Measure of spread in a dataset or uncertainty in a model prediction. xxvii

Acronyms

AKMC adaptive kinetic Monte Carlo. xxvii, 2, 6, 89

CG conjugate gradient. xxvii, 10, 56, 57

CI-NEB climbing-image nudged elastic band. xxvii, 14, 40

DAG directed acyclic graph. xvii, xxvii, 31, 40

DFT density functional theory. xxvii, 22, 23, 26

EMD earth mover's distance. xiii, xxvii, 78, 80, 82

FPS Farthest point sampling. xxvii, 83

GPDimer Gaussian process regression dimer. xvi, xvii, xx, xxi, xxvii, 19, 46, 50, 51, 71, 73, 74, 77, 78, 84, 89

GPU graphics processing unit. xxvii

GUI graphical user interface. xxvii, 32

H-TST harmonic transition state theory. xxvii, 2, 4–6

HF Hartree-Fock. xxvii, 26, 48, 50, 89

HPC high performance computing. xxvii, 40

IDPP image dependent pair potential. xxvii, 12, 13, 39

IRA iterative rotations and assignments. xxvii, 34

KMC kinetic Monte Carlo. xxvii, 6

KS Kohn-Sham. xxvii, 22, 23

L-BFGS limited-memory Broyden-Fletcher-Goldfarb-Shanno. xxvii, 10, 15, 39, 47, 55–57

MD molecular dynamics. xxvii, 2–4

MEP minimum energy path. xxvii, 10, 12, 13

MLIP machine learned interatomic potential. xxvii, 19

MLL marginal log-likelihood. xxvii, 17, 77, 78, 93

MMF minimum mode following. xxvii, 39, 49

MMF-NEB minimum mode following nudged elastic band. xxvii, 7, 31, 36, 51

MVN multivariate normal. xxvii, 15, 16, 93

NEB nudged elastic band. xvii, xxvii, 10, 13, 14, 33, 34, 39–41, 51

OTGP optimal transport Gaussian process framework. xxvii

OTGPD optimal transport Gaussian process dimer. xxi, xxvii, 7, 84–86, 88, 89

PSD positive semi-definite. xxvii, 82

RKHS reproducing-kernel Hilbert space. xxvii, 77

RMSD root-mean-square displacement. xvii, xxvii, 34, 35, 38, 50, 51

RPC remote proceedure call. xvii, xxvii, 42, 43

SCF self consistent field. xxvii, 22, 26, 48

SCG scaled conjugate gradient. xxvii, 47, 62, 73

SPD symmetric positive definite. xxvii

TST transition state theory. xxvii, 2

WBO Wiberg bond order. xii, xxvii, 32, 33

Acknowledgements

My work has spanned a couple of fields, and for that, I'm grateful to have been given an opportunity to be inspired by so many.

My family comes first. The PhD has been a bit of a white whale for me—the bare minimum requirement I demanded of myself. I know this ambition must have been grating to them, as I am the youngest and have long been looking up to their accomplishments. Their unwavering support has sustained this dream since I was in pre-school.

I'd also like to specifically thank Prof. Rajarshi Chakrabarti at IIT Bombay. He was instrumental early on, introducing me to the fascinating world of soft matter systems when I was a first-year undergraduate. My sister Dr. Amrita Goswami, has been a colleague and inspiration, both when we were under Prof. Singh and after.

By day, of course, I can think of no one who embodies my degree more than my advisor, Prof. Hannes Jónsson. Since we met at IISc so many years ago, I've learned a lot from him. He has been an inspiration whose support provided the diving board I needed to go off into statistics, tensor methods, and software development.

Dr. Vilhjálmur Ásgeirsson stands out as someone who not only got Hannes' attention back to my tentative email but who has also been a staunch friend.

My degree spanned the COVID-19 pandemic, but thankfully, for me, this only opened the doors to more science and more people who had time outside the usual bubble of real life. Dr. Ondřej Čertík was my first collaborator and confidant, helping steer me through both corporate shenanigans and academic endeavors. He introduced me to the delight that is modern Fortran and compilers, and for that, I'm ever grateful.

All my co-authors are special; especially family, Dr. Moritz, Dr. Amrita, Sonaly, Ruhila, and Prof. Debabrata. Ondrej, Maxim and Satish in particular, spent long hours with me everyday for discussing the nuances of Fortran and MATLAB respectively. Amongst my near or to-be collaborators come Dr. Andreas Vishart, Prof. Laurent Béland, and Dr. Miha Gunde. My in-laws Nagaselvamani and Sankaranarayan, though non-collaborators have provided immense warmth. My non-human collaborators, Arí, Yoda, Jude, Crystee, TuiTui, Thor, Loki, and all the garden cats are ever driving towards being more curious about everything. Dr. Suryakant and Dr. Ashwini have both been great sources of support for the family and me.

I haven't been very social, but my labmates managed to find a way to connect, especially Ellie, Ivan, Alejandro, Oskar, Olafur, Andre, Liam, and everyone else. At the University, I was honored to be able to teach for Prof. Helmut Neukirchen and support Prof. Steinn Guðmundsson. My graduate committee as a whole, Prof. Birgir Hrafnkelsson, Prof. Morris Riedel, Prof. Hannes Jónsson, Prof. Egill Skúlason and Prof. Thomas Bligaard greatly enriched my time at the university. A special mention to my cactus, Ami, named recently, but who has been by my side since my first apartment in Alfheimar, and led to having many more plants.

Early on, when I was shopping for my grant, I was introduced to Prof. Birgir, who has been invaluable and a pillar of knowledge, statistical and otherwise. I met Prof. Morris Riedel at the first Icelandic HPC meeting and have been attending ever since. He has always given great advice, not to mention providing support for me when I was between funds.

Funding has been a long thread for me. I briefly received income from Quansight Labs for working on foundational open-source projects, an association which was initially interesting but eventually tiresome. I was lucky to have a chance to work under Prof. Gianluca Levi on software enhancements for GPAW within the context of plane wave calculations, leaving even fewer aspects of chemistry untouched.

A few years ago, I was lucky enough to have Dr. Miha Gunde sit behind me in the lab. I've had the pleasure of being able to count on him at all hours of the day or night to respond and discuss any and every aspect related to kinetic Monte Carlo and molecular systems.

It was around the same time that I met several others who provided support, financially and academically, within the context of adaptive kinetic Monte Carlo. In particular, Prof. Laurent Béland has supported me for months while I visited his lab in spite of great administrative hurdles, all the while working actively with me on several interesting projects. Prof. Normand Mousseau, whom I also met around the same time, has provided inputs and insight on much of my recent work, and I am very grateful. He has done a stellar job helping shape my thesis through his role as my opponent as well.

I owe a great debt to Prof. Michele Ceriotti at EPFL. Prof. Ceriotti, whom I first met at several conferences, offered unbridled support despite the baggage of my extended degree timeline. He, along with Dr. Guillaume Fraux, exhibited remarkable patience and commitment while I worked on this thesis. Dr. Fraux also took time to provide suggestions for improvement.

In my conviction that my work must be read widely, I solicited many external reviews from people who earned my respect. I have been humbled by the positive responses, and am even more grateful to those who provided constructive criticism along with approbation. In particular, Prof. Martin Gruebele furnished historical context and insightful academic points. I relished the opportunity to reconnect collaborators and wellwishers including Dr. John Pask, Prof. Baron Peters, Prof. Zubin Jacbin and Prof. Pedro Costa amongst many others.

I'd also like to take the opportunity to thank every other person I've met with or interacted with, with special note for my landlady Guðrun, who gave my family and I a lovely house. Finally, I thank my wife, Ruhi again, for making us feel home wherever we are, and for lending me her brilliant perceptually uniform, colorblind friendly colorscheme ("ruhi") used for most of the figures.

1 Introduction

> If you wish to make an apple pie from scratch, you must first invent the universe.
>
> Carl Sagan

The central pursuit of chemistry is the rational control and transformation of matter. Progress in this domain hinges on the strategic use of abstraction, specifically in how we choose to represent chemical systems. This evolution of representation is profound, moving from the empirical models of alchemy to the rigorous mathematical frameworks of today. The advent of formalisms like second quantization, for instance, provided a language to systematically treat many-body quantum effects, fundamentally changing our ability to model molecular interactions.

Words, too, function as representations. They imperfectly ferry an idea from my mind to yours and, in doing so, enact a subtle form of control. A leaky abstraction, like a rumor retold, degrades signal and invites failure. We therefore cultivate models that compress without distorting, that guide computation without surrendering physics.

By creating simplified yet powerful models that capture essential phenomena, we make intractable problems computationally solvable. Implementing these representations computationally introduces a physical cost that is often overlooked. The ultimate goal of this pursuit is not merely descriptive understanding, but predictive control [1] with direct applications in materials science, pharmacology, and industry [2]. Achieving this requires immense computational power, and the efficiency of these computations is constrained by fundamental thermodynamics. Echoing the principles behind Maxwell's Demon [3], Landauer's principle [4] dictates a minimum energy cost for erasing information: the "cost of forgetting." An "efficient Demon" would avoid accessing the entire distributed dataset simultaneously, thereby eliminating the need to continually churn space in memory. From this perspective, modern high-performance computing—specifically distributed networks—functions as a strategy to manage this information burden, even though the total energy expenditure often remains orders of magnitude higher in practice. This trade-off succeeds by reducing the time-to-solution and facilitating work with datasets that exceed the storage capacity of a single node. Thus, by partitioning a large problem across many computational units, we reduce the information load—the specific memory the core must retain and process. While not offsetting the total energy cost of distributed computing, this distribution mitigates the local difficulty of information processing. Practically, this approach enables explorations of reactive systems on experimentally relevant time and length scales, accelerating the pace of discovery.

1.1 Chemistry for computers: Space, Time and Temperature

Computational chemistry relies on a foundational spatial representation. We begin by defining a high-dimensional space where each point corresponds to a unique configuration of atomic nuclei. A scalar potential energy associates with each point to generate a landscape known as the Potential Energy Surface. The exploration of this landscape—finding stable minima, identifying transition saddle points, and defining reaction paths—manifests fundamentally as a problem in mechanics and optimization. This framework remains atemporal and zero-temperature; a distance metric defines "closeness," and geometric paths measure "progress" rather than the passage of time.

Upon this static, spatial landscape, we superimpose representations of dynamics that introduce time and temperature. The most direct of these, molecular dynamics (MD) traces a time-resolved trajectory of the system according to Newton's laws. Figure 1.2 [5] illustrates this concept. A variety of processes, from phase transitions in fixed-topology force fields to reactive potentials with bond formation and breaking may be understood in this context. Here, temperature arises naturally from the kinetic energy of the particles.

From numerical stability considerations MD requires a time step on the order of one to two femtoseconds (10^{-15} s), to capture the bond vibrations as the motion of interest. Consequently, achieving simulation times on the order of milliseconds necessitates billions (10^{12}) of steps. This computational expense often prohibits the observation of diffusion effects and point defects that occur on macro-scales.

To address these temporal constraints, we move to a representation that is thermodynamically equivalent in the long-time limit but computationally distinct. This approach abstracts the explicit integration of femtosecond vibrations by leveraging statistical mechanics to connect key spatial features of the landscape—the reactant minima and the transition state saddle points—to a macroscopic rate, which has units of inverse time. Temperature enters not through kinetic energy, but through the partition functions that describe the probability of occupying these critical states (Figure 1.3) [6].

These form the class of Kinetic Monte Carlo (KMC) methods which exploit the fact that the long-term dynamics of many chemical systems may be modeled as a series of jumps from state to state. Instead of following the trajectory through every vibrational period, KMC treats state-to-state transitions directly. By assuming that the first escape time from a state follows an exponential distribution, KMC allows access to much longer time scales—typically seconds and beyond. In standard implementations, the algorithm selects a transition path from a catalog of rates, typically calculated with harmonic transition state theory (H-TST), updates the time, and places the system in a new state.

In practice, adaptive kinetic Monte Carlo (AKMC) [7, 8, 9] discovers relevant transitions on the fly, builds local catalogs of events, and samples waiting times from exponential clocks. This effectively rewrites dynamics in the language of statistically weighted hops. From a thermodynamic perspective, this is not an approximation but a change in basis: provided the rate constants are accurate, the master equation describes the exact same probabilistic evolution of the system as the ensemble of MD trajectories, but efficiently skips the non-reactive vibrational basins. We shift therefore from covering every possible state, to the study of rare events.

Thus, the challenge manifests as two-fold: first, to efficiently map the high-dimensional spatial landscape, and second, to employ either direct (MD) or statistical (transition state theory (TST), KMC) methods to model the temporal evolution of a system across it. This dissertation focuses on developing efficient representations for both aspects. We recognize that predictive control—steering a reaction towards a desired product—requires mastery over both the spatial representation of what exists as possible and the temporal representation of what becomes probable at a given temperature.

We return to space, because intuition begins there. The Potential Energy Surface derives from the Born–Oppenheimer separation of electronic and nuclear motion [10]. For anything beyond simple models, the curse of dimensionality [11, 12] defeats direct visualization. The Lennard-Jones 38-atom cluster (LJ38) offers a classic case [13, 14]: a rugged landscape with many contending structures. As an interatomic potential, the Lennard-Jones historically has been a stand-in for the study of Argon [15] which forms the model system in Figure 1.1 as well.

To trace a "meaningful" path through such a space, we must first create a map. Tools built on principles of dimensionality reduction, unsupervised learning, and the identification of landmark configurations enable projection onto a two dimensional figure [16]. Figure 1.1 demonstrates this approach. We must first tame this complexity visually before we can interrogate it physically. Although such maps often aim to preserve metrics [1], they lack explicit information regarding thermodynamic and kinetic transformations. Mapping basins in itself, provides no explicit transition paths or rates between them and cannot characterize or categorize the actual reactive events. We dispense with such post-hoc methods; for the remainder of this work, we focus on the means to explore interesting aspects of phase space and trace paths explicitly.

Even without a low dimensional projection, as soon as we characterize the landscape through the Potential Energy Surface and choose a metric for distance, we can simulate time-dependent behavior. MD evolves the system's coordinates according to Newton's equations of motion, providing a microscopic view of atomic motion [5]. The result is a continuous path through configuration space, which mostly samples thermal vibrations within energy wells, with rare stochastic events of barrier crossing, and the characteristic recrossings that occur near the transition state, often called a trajectory. Figure 1.2 illustrates such a trajectory on a model double-well potential. Panel (a) shows the three-dimensional landscape with the trajectory (red line) snaking across the surface, while panel (b) projects this motion onto the reaction coordinate plane overlaid with the energy landscape as a colored background. The trajectory begins localized in the reactant basin (yellow circle), undergoes several thermal explorations, and eventually crosses the dividing surface at $q_0 = 0$ (magenta dashed line) into the product basin (cyan circle). Critically, the path exhibits multiple recrossings of the barrier—the trajectory does not simply traverse from reactant to product, but rather crosses back and forth, reflecting the stochastic nature of barrier passage at finite temperature.

Panels (c) and (d) reveal why MD, while dynamically rigorous, contains far more information than needed for determining basins. While the temperature and energy fluctuate as expected for a finite-temperature system with a stochastic thermostat, the quantitative accuracy is less

[1] in the sense that relative Euclidean distances are often preserved

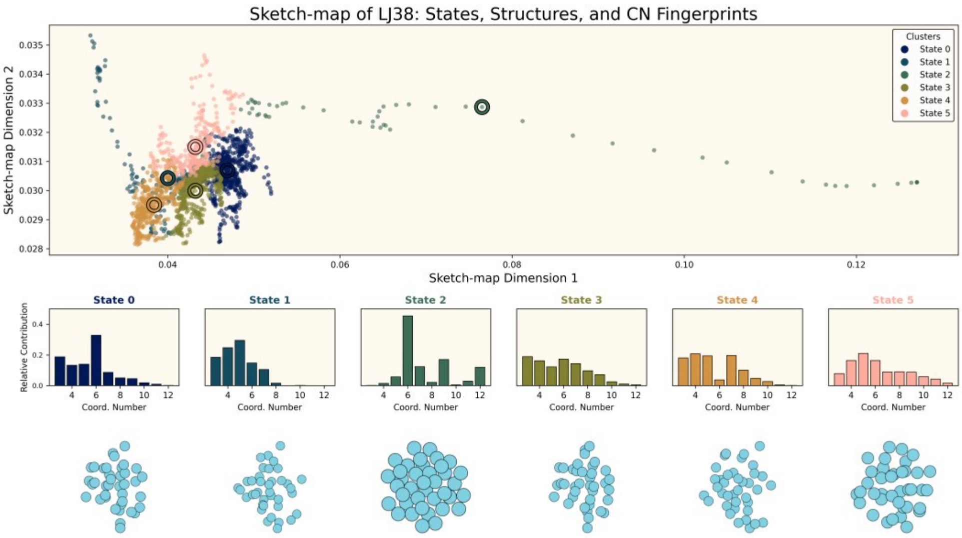


Figure 1.1. *The complex structural landscape of the Lennard-Jones 38-atom (LJ38) cluster is visualized using a sketch-map projection. The data is sourced from a molecular dynamics simulation thermostatted at 80.0 K, modeling an Argon cluster. Each point in the trajectory was characterized by a high-dimensional vector representing its coordination number (CN) histogram, which was then projected onto this 2D map. The resulting visualization clearly separates distinct structural basins. An unsupervised clustering algorithm identifies six major structural families, shown as colored regions. This approach allows for an intuitive understanding of the system's structural diversity, with the inset panels providing the characteristic CN histogram "fingerprint" for each distinct state.*

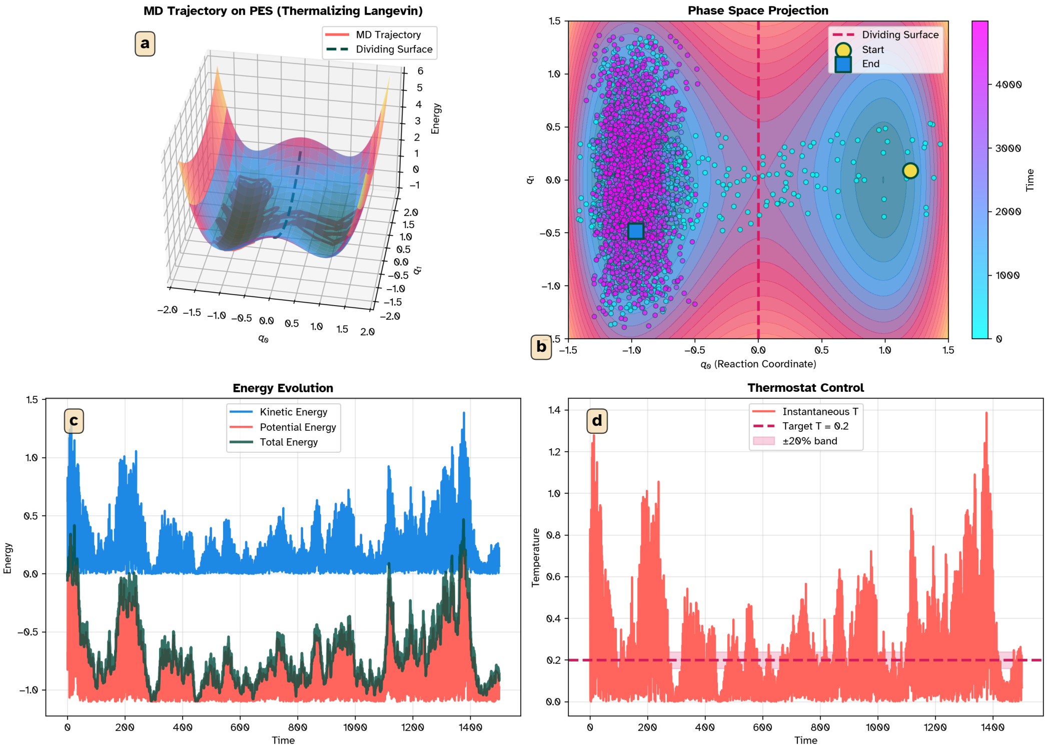


Figure 1.2. ***(a)*** *A simulated Molecular Dynamics trajectory (red line) on a double-well Potential Energy Surface, shown in 3D with the dividing surface (black dashed line) separating reactant and product regions.* ***(b)*** *Top-down projection of the same trajectory onto the reaction coordinate q_0 and bath coordinate q_1, with energy contours shown as background. The trajectory begins at the reactant minimum (yellow circle) and eventually reaches the product basin (cyan circle), with multiple recrossings of the dividing surface.* ***(c)*** *Energy evolution during the simulation, showing kinetic, potential, and total energy.* ***(d)*** *Temperature control by the Langevin thermostat, maintaining the target temperature around $T = 0.2$ K within a ±20% band.*

important here than the qualitative demonstration of basin residence, barrier crossing, and recrossing behavior. The temperature and energy fluctuate as expected for a finite-temperature system coupled to a heat bath; these fluctuations characterize the canonical ensemble. The direct simulation of time presents a practical barrier.

Most of the computational effort in MD is expended on thermal basin exploration—the rapid, high-frequency oscillation of atoms within a stable energy well. While this exchange between kinetic and potential energy is physically rigorous and essential for defining the free energy of the state, it contributes little to the analysis of rare events that drive structural evolution.

This observation motivates the further simplification of H-TST. Rather than tracking every atomic coordinate along the entire trajectory, H-TST posits that the reaction rate depends primarily on the properties of a few critical points on the PES. These are the reactant minimum (low-energy starting configuration), the product minimum (low-energy final configuration), and the transition state, often simplified to the highest-energy saddle point connecting them along the minimum free energy path. By focusing on these stationary points rather than the full dynamics, we can derive reaction rates with far fewer calculations.

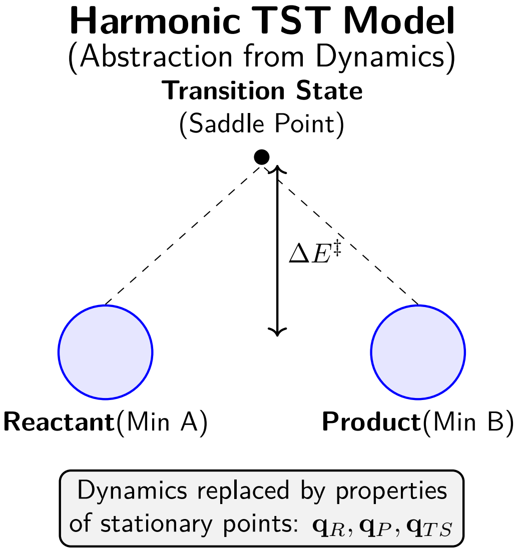

Harmonic TST Model
(Abstraction from Dynamics)
Transition State
(Saddle Point)
$\Delta E^{\ddagger}$
Reactant(Min A)
Product(Min B)
Dynamics replaced by properties of stationary points: $\mathbf{q}_R, \mathbf{q}_P, \mathbf{q}_{TS}$

Figure 1.3. *The conceptual abstraction for Harmonic Transition State Theory (H-TST) simplifies the complex dynamics. This model replaces the entire trajectory with an analysis of the energetic and vibrational properties of three critical stationary points: the reactant minimum, the product minimum, and the transition state saddle point connecting them.*

H-TST approximates the potential near minima and the transition state with quadratic forms, enabling statistical-mechanical evaluation of partition functions and the derivation of a rate constant. Figure 1.3 demonstrates how the trajectory of Figure 1.2 collapses to three stationary points and their local curvatures. This chain of reasoning from quantum mechanics to a harmonic model underpins how we compute and understand chemical reactivity.

The abstraction from the full dynamics in Figure 1.2 to the simple rate model in Figure 1.3 forms a compelling progression. However, most systems, even small ones like the LJ38 in Figure 1.1 have complex landscapes containing a vast network of states, not just a single reactant–product pair.The notion of “connectivity” becomes malleable, as is the notion of “close”, and many escape paths may exist, with widely varying barriers and time scales. Perhaps the disconnect between kinetics and theromdynamics is most clearly understood from the engineering perspective, where the Haber-Bosch [17, 18, 19] in practice requires conditions which do not provide maximum yield, but does allow the reaction to proceed in reasonable timeframes. For slowly evolving systems, evaluating H-TST rates for all viable escape paths, allows the continuous Potential Energy Surface to be recast as a discrete set of states linked by thermally activated transitions (Figure 1.4). The long-term evolution then follows a memoryless, Markovian journey on that network, leading to two distinct simulation methodologies. From the perspective of long-timescale evolution, MD and KMC are thermodynamically equivalent representations; they both aim to sample the equilibrium distribution. The distinction lies in their sampling efficiency for activated processes.

On-lattice KMC adopts a predefined lattice and a fixed catalog of allowed processes [20, 21]. Practitioners precompute or fit the energy barriers and rate constants for these processes to various local chemical environments. During a simulation, events are drawn stochastically from this lookup table with probabilities determined by H-TST rates, and the occupation numbers (or site identities) are updated accordingly. The system time advances via an exponential clock: if N events are available with rates $k_1, k_2, \ldots, k_N$, the time to the next event is $\Delta t = -\ln(\xi)/\sum_i k_i$, where ξ is a uniform random number. This approach delivers exceptional speed for crystalline diffusion and lattice-respecting reactions, yet it omits events that a prior catalog fails to include.

In contrast, off-lattice methods like AKMC [8, 9, 22] lift the fixed lattice and process table constraints and treat states in continuous space. The rate catalog and kinetic network grow on the fly: from the current minimum, the method performs single-ended saddle searches to uncover escape routes, evaluates rates via H-TST, and assembles a local event table. We then proceed as in on-lattice KMC, select an event, advance time with an exponential clock, and resume discovery from the new minimum. This adaptive loop replaces a fixed move set with active discovery and suits defects, amorphous phases, and surface reconstructions without prior assumptions [23]. The general utility of this method relies on the landscape topology. Systems characterized by high disorder, such as bulk liquids, possess a landscape dense with shallow minima and low barriers. In such regimes, the algorithm tends to trap itself within a superbasin, expending resources to catalog negligible diffusive events rather than describing meaningful phase transitions. Consequently, while clusters and crystalline defects remain tractable, bulk liquids often require different approaches to avoid this combinatorial explosion of states. The cost of discovering pathways in this framework must then be faster than the time taken to traverse the landscape using dynamical methods. The off lattice method is primraily of importance for in principle being free from the bias of having to select events.

We must acknowledge, however, that identifying the saddle point is a necessary but not sufficient condition for describing the full dynamics. Recrossing events, variational effects, and anharmonicity all contribute to the transmission coefficient. Yet, without an efficient method to locate these saddle points in the first place, the higher-order corrections cannot even be attempted.

A significant advantage of this network representation lies in sensitivity analysis; in large reaction networks, the macroscopic evolution often exhibits sensitivity to only a small subset of critical rates, rendering errors in non-rate-limiting steps inconsequential.

From this chain of reasoning then, faster identification of transition states with fewer evaluations

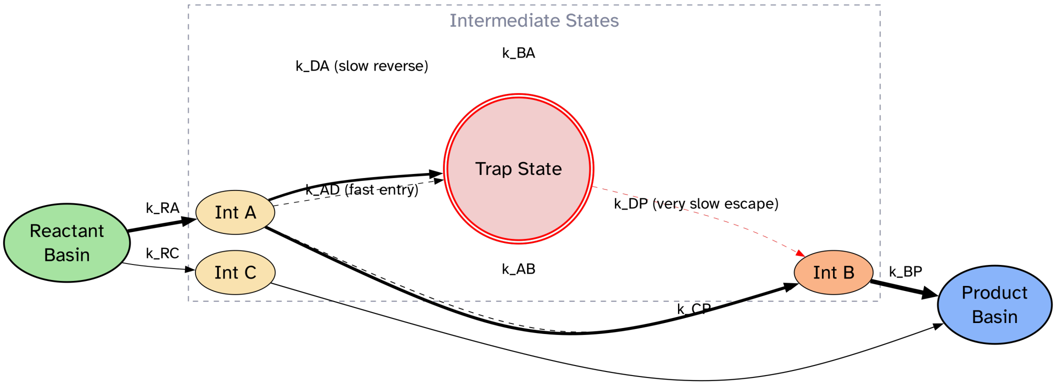


Figure 1.4. *Abstraction of a Potential Energy Surface to a Discrete Kinetic Network. The continuous, high-dimensional PES is simplified into a network of states (nodes) and transitions (arrows). Each node represents a stable energy basin. Each arrow's thickness is proportional to its TST-calculated rate constant (k), visualizing the system's kinetic preferences. The network reveals key dynamical features: a dominant pathway (Reactant → Int A → Int B → Product) with high-flux transitions (thick lines), a slower side-channel (Reactant → Int C → Product), and a kinetic trap (Trap State). The trap is characterized by a fast entry rate (k_{AD}) and very slow escape rates (k_{DA} and k_{DP}), representing a long-lived metastable state that can dominate the system's evolution. This abstraction allows methods like aKMC to simulate timescales far beyond the reach of direct molecular dynamics.*

of a high-accuracy solver on the Potential Energy Surface unlocks reaction networks and, with them, practical control over chemical change, feeding into the goal of accurate in-silico control of materials.

1.2 Motivation

This dissertation concerns itself primarily with the development of computational representations to efficiently model inhospitable regions of chemically interesting phase space.

1.3 Research objectives and hypotheses

The central thesis of this work is that the efficiency of rare-event sampling is constrained not just by the cost of electronic structure, but by the rigidity of standard software architectures and the fragility of optimization algorithms. To address this towards the in-silico control of materials to accelerate scientific applications, we investigate the following across various length and time scales:

Relativistic spurious state evasion Can a squared Hamiltonian formulation in a finite element modular Fortran framework provide state of the art stability[2] and numerical performance for single atom all-electron mean-field calculations of both the Schrodinger and Dirac equations?

Saddle diagnostics Can the interpretation of saddle structures be improved beyond the

[2] by avoiding spurious states since no variational principle holds for the Dirac case

"eyeball norm" and baseline methods of calculating bonds based on covalent radii with approximate energy surfaces? Can optimization trajectories towards these points be improved?

Software designs Does the modernization of existing software unlock novel scientific algorithms for reducing the number of samples needed to find a saddle point?

Internal coordinates Can performance gains from complex internal coordinate generating / coordinate drive methods be beaten by better engineered Cartesian coordinate based local surrogate saddle searches?

Statistical performance modeling Average-cost benchmarks and point estimates of performance mask algorithmic reliability despite havnig poor statistical backing due to repeated data. Can more robust Bayesian methods lead to novel insights?

Wall-time efficiency GP acceleration has been known to be theoretically efficient for almost a decade, but no implementation has managed to show high reliability and wall-time gains in time, can this be alleviated by careful analysis of hyperparameters and software?

1.4 Overview

The introductory Chapter 2 outlines basic preliminaries and provides pointers to the wider literature for the three fields under consideration. Across the programming languages and academic domains covered, wall-time efficiency functions as the core constraint, as illustrated in Figure 1.5.

We begin with the fundamental physics in Chapter 3, which introduces `featom`, a high-order finite element method (FEM) solver for the Dirac and Schroedinger equations in Fortran. Unlike the frozen core approximations common to valence chemistry, `featom` facilitates heavy element relativistic calculations in the all-electron mean-field approximation. This tool stands distinct from the kinetic methods discussed in the remainder of the thesis.

Chapter 4.3 shifts focus to the EON software suite, a hybrid C++ and Python framework designed for long-timescale simulations. We detail the client-server architecture and the evolution of the software stack. Notably, we demonstrate how these scalable designs enable hitherto hard-to-implement workflows, including state-of-the-art nudged elastic band methods and a novel hybrid method, the minimum mode following nudged elastic band (MMF-NEB).

These efforts evolve into an implementation of Gaussian Process acceleration for single-ended saddle searches in C++, exploring a large, though historically challenging, benchmark set. We extend the methodology to surface calculations and vet it for use within proximal reaction network explorations. Because exhaustive case studies fail to scale indefinitely, Chapter 6 explores performance modeling in high-throughput regimes. We employ Bayesian hierarchical methods in R and Stan to highlight computational bottlenecks, before directly tackling the data efficiency of Gaussian Process methods in Chapter 7.

These efforts culminate in Chapter 8 with the OTGPD, a framework leveraging optimal transport distance metrics. This approach proves chemically intuitive and transferable across systems, demonstrating state-of-the-art wall-time efficiency. A brief summary and conclusions

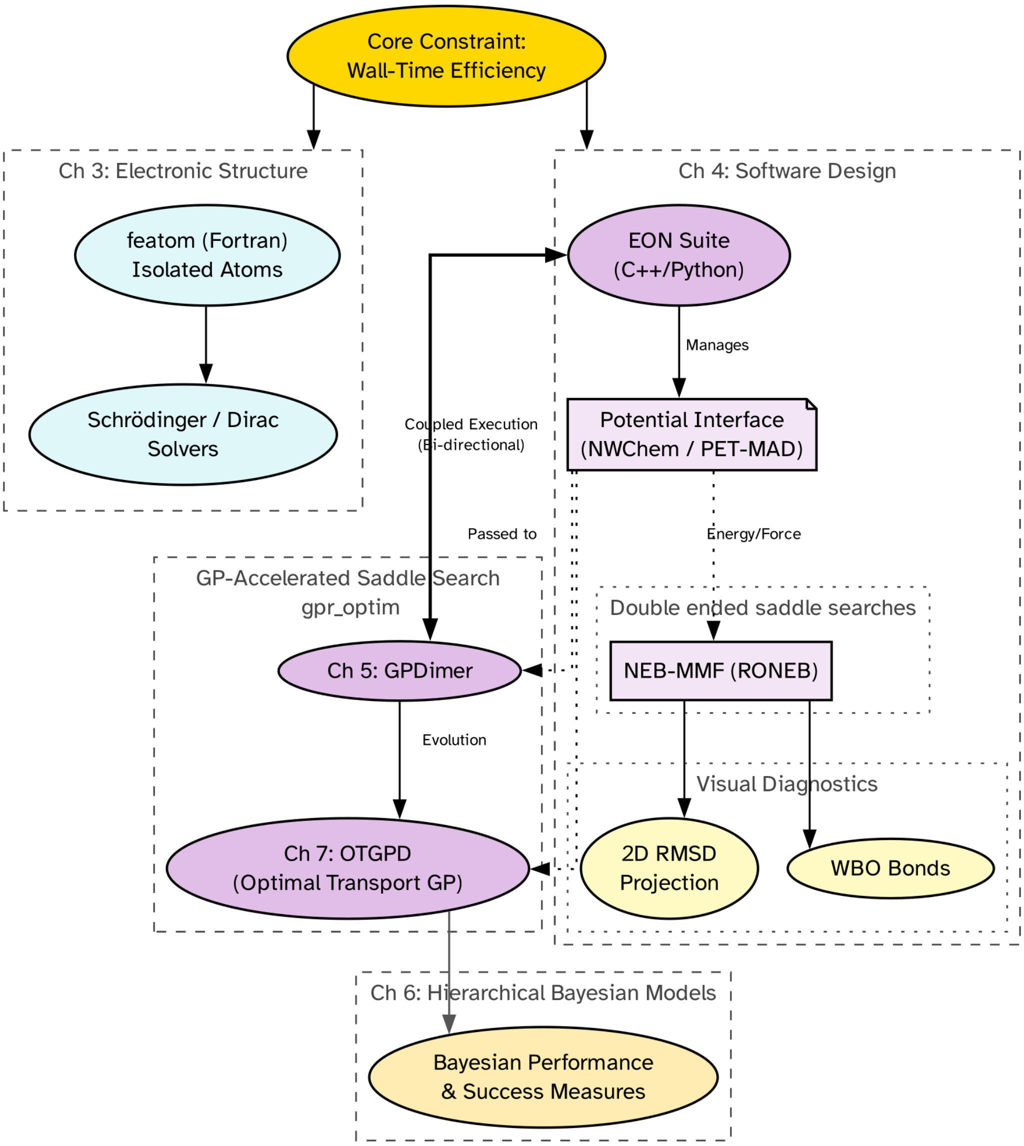


Figure 1.5. *The structural connectivity of this dissertation. Wall-time efficiency serves as the primary constraint driving innovation across three domains: relativistic electronic structure (cyan), chemical kinetics software (purple), and statistical diagnostics (amber). EON (Chapter 4) functions as the central orchestrator, managing interfaces to potential energy and force calculators and maintaining a tight, bidirectional coupling with the distinct GP-accelerated package developed for the GPDimer in Chapter 5 and the OTGPD in 7. Quantification via the Bayesian performance metrics in Chapter 6 and visual diagnostics in Chapter 4 round out the software and methods developed.*

pave the way for the reader to engage with the primary articles in the appendix [3].

[3] on ArXiv these are not appended, but are freely available

2 Theory

> The derivation can be made to look slightly less juvenile by introducing an obscure notation at this point.
>
> P. Pechukas
> Dynamics of Molecular Collisions, Part B

2.1 Minimum mode following

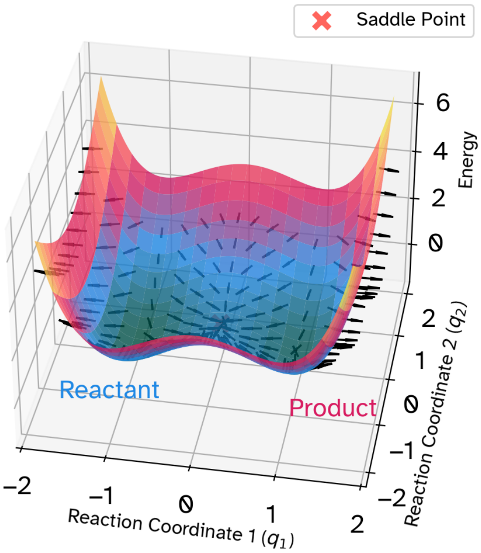


Figure 2.1. *An illustration of the forces from the dimer method for locating a first-order saddle point on a two-dimensional Potential Energy Surface. The vector field represents the effective dimer force, a transformation of the true potential gradient ($-\nabla V$). This modified force guides an optimization uphill along the minimum-energy pathway while minimizing energy in orthogonal directions, enabling an efficient climb to the saddle point (red 'X').*

To seek a mountain pass on a Potential Energy Surface, one need not survey the entire range [24]. The dimer method is a robust algorithm designed to locate first-order saddle points on a Potential Energy Surface without calculating the full Hessian matrix [25]. It belongs to the class of minimum mode following methods [26, 27, 28], where the search is guided by the eigenvector corresponding to the lowest eigenvalue of the Hessian. The core concept is to apply a pointwise transformation to the force, enabling an efficient climb uphill while simultaneously minimizing the energy in all orthogonal directions, as illustrated in Figure 2.1. This section recaps the standard formulation, similar to that implemented in software such as EON [29].

The "dimer" itself consists of two replicas (or images) of the system, $\mathbf{R}_1$ and $\mathbf{R}_2$, defined by their separation from a central point $\mathbf{R}$ along a normalized orientation vector $\hat{\mathbf{N}}$:

$$\mathbf{R}_1 = \mathbf{R} - \frac{\Delta R}{2}\hat{\mathbf{N}} \tag{1}$$

$$\mathbf{R}_2 = \mathbf{R} + \frac{\Delta R}{2}\hat{\mathbf{N}} \tag{2}$$

The algorithm proceeds by iteratively alternating between rotation and translation steps.

2.1.1 Rotational Step

The primary goal of the rotational step involves aligning the dimer orientation vector, $\hat{\mathbf{N}}$, with the minimum mode at the midpoint $\mathbf{R}$. To this end, we rotate the dimer to minimize the total energy, $E = E_1 + E_2$. With this, an effective rotational force, or torque $\mathbf{F}_{rot}$ can be derived from the forces at the endpoints, $\mathbf{F}_1$ and $\mathbf{F}_2$, to estimate the change in local curvature without computing the Hessian. The curvature C along the dimer axis can be approximated using a finite difference:

$$C(\hat{\mathbf{N}}) \approx \frac{(\mathbf{F}_2 - \mathbf{F}_1) \cdot \hat{\mathbf{N}}}{\Delta R} \tag{3}$$

This $\mathbf{F}_{rot}$ defines the search direction perpendicular to $\hat{\mathbf{N}}$, depicted in Figure 2.2. We use an optimization algorithm, such as conjugate gradient (CG), to determine the orientation at which $\hat{\mathbf{N}}$ is aligned with the minimum mode. Practically, we recognize that the curvature within a rotational plane spanned by the current orientation $\hat{\mathbf{N}}$ and the normalized torque direction approximates a sinusoidal curve

$$C(\phi) = A + B\cos(2\phi) + D\sin(2\phi) \tag{4}$$

By evaluating the gradients at the current position we determine the angle which minimizes the curvature along a given search direction, and the dimer rotates to this orientation. This process occurs without the coordinates of the midpoint of the dimer changing and continues until the predicted rotation angle falls below a fixed convergence threshold. For standard calculations a rather loose tolerance of 5 degrees typically suffices.

2.1.2 Translational Step

Once the dimer is aligned, the translational step moves the midpoint $\mathbf{R}$ towards the saddle point. This is guided by a modified force, $\mathbf{F}_{\text{trans}}$, where the component of the true force parallel to the minimum mode is inverted. This transformation effectively turns the saddle point into a local minimum from the perspective of an optimization algorithm. The modified force is given by:

$$\mathbf{F}_{\text{trans}}(\mathbf{R}) = \mathbf{F}(\mathbf{R}) - 2(\mathbf{F}(\mathbf{R}) \cdot \hat{\mathbf{N}})\hat{\mathbf{N}} \tag{5}$$

This effective force field, shown in Figure 2.1, ensures the system moves uphill towards the saddle. An optimizer, commonly the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (Alg. 1), takes a step using this modified force.

The cycle of rotation and translation is repeated until the true force at the midpoint $\mathbf{R}$ falls below a defined convergence threshold, indicating that a saddle point has been successfully located.

2.2 The Nudged Elastic Band (NEB) Method

Minimum mode following methods prove useful for reaction network generation and general kinetic applications. However, when we distinguish specific reactant and product basins, the objective shifts. In such cases, we seek to determine the most probable paths connecting

Algorithm 1 L-BFGS Two-Loop Recursion

```
Given: Current gradient $\nabla f_k$, history of $m$ updates $\{s_i, y_i\}_{i=k-m}^{k-1}$, where $s_i = x_{i+1} - x_i$ and $y_i = \nabla f_{i+1} - \nabla f_i$.
$q \leftarrow \nabla f_k$
for $i = k-1, \ldots, k-m$ do
    $\rho_i \leftarrow 1/(y_i^T s_i)$
    $\alpha_i \leftarrow \rho_i s_i^T q$
    $q \leftarrow q - \alpha_i y_i$
end for
$H_k^0 \leftarrow \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}} I$                ▷ Initial Hessian approximation
$z \leftarrow H_k^0 q$
for $i = k-m, \ldots, k-1$ do
    $\beta \leftarrow \rho_i y_i^T z$
    $z \leftarrow z + s_i(\alpha_i - \beta)$
end for
return Search direction $p_k = -z$
```

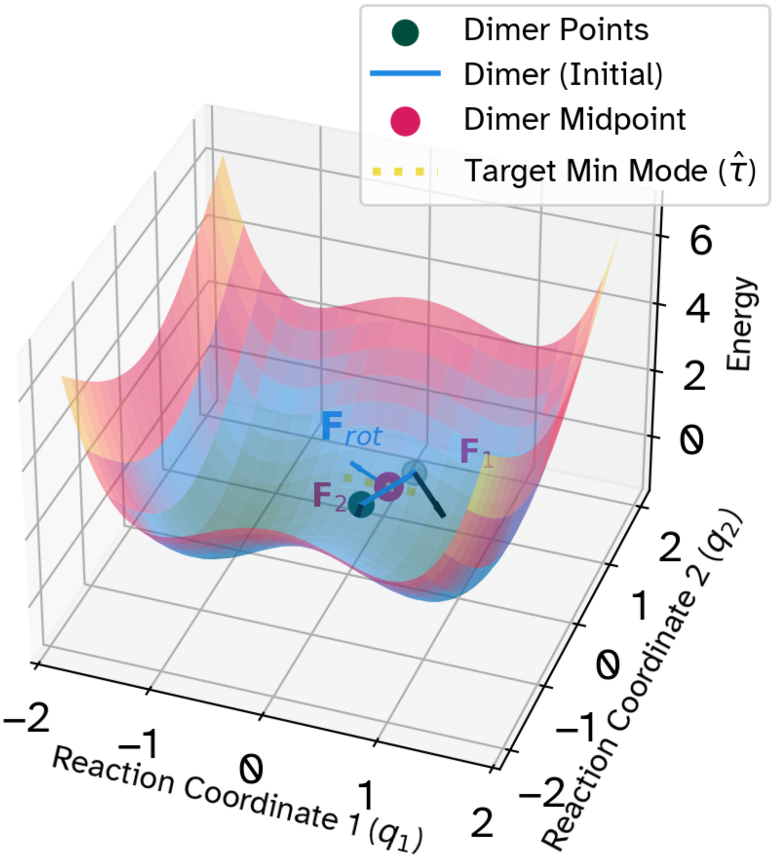


Figure 2.2. *The rotational step of the Dimer Method. The effective rotational force* ($\mathbf{F}_{rot}$) *is derived from the atomic forces* ($\mathbf{F}_1, \mathbf{F}_2$) *and applies a torque to the misaligned dimer. This torque drives the dimer's orientation to align with the minimum mode* ($\hat{\tau}$)*, which is the prerequisite for the translational step.*

these fixed points in configuration space. A discrete approximation to a trajectory between the reactant and product can be formed through a “chain of states”. Conceptually, this can be viewed as an extension of the dimer images, and the subsequent string is then connected through springs.

The NEB method [30] is a form of double ended saddle search tecnique, used to determine the minimum energy path (MEP) between a known reactant and product state [31]. The MEP on the free energy surface represents the path of highest statistical weight. The method works by creating a discrete representation of the path, known as a “band,” which is a series of system configurations (or “images”) connected by springs.

The total force on each image is a combination of the perpendicular component of the atomic force derived from the Potential Energy Surface and a fictitious spring force, which are projected to guide the system across the transition state.

2.2.1 Path Discretization and Initialization

The continuous reaction path is approximated by $P + 1$ images, denoted $\{\mathbf{R}_0, \mathbf{R}_1, \ldots, \mathbf{R}_P\}$, where $\mathbf{R}_0$ and $\mathbf{R}_P$ are the fixed reactant and product endpoints. The $P - 1$ intermediate images are movable. The initial path is typically generated by a linear interpolation between the endpoints:

$$\mathbf{R}_i = \mathbf{R}_0 + i\frac{\mathbf{R}_P - \mathbf{R}_0}{P} \quad \text{for } i = 1, \ldots, P - 1 \tag{6}$$

Alternatively, an initial path can be constructed from a series of provided configuration files, or by using a cheap surrogate potential like the image dependent pair potential (IDPP) [32] which generates a more physically realistic path by minimizing the energy of a system described by a simple, classical potential. The total potential for an image $\mathbf{R}_k$ is a sum of pair potentials V_{ij}:

$$V_{\text{IDPP}}(\mathbf{R}_k) = \sum_{i<j} V_{ij}(\mathbf{R}_k) \tag{7}$$

Each pair potential is a harmonic spring that is switched on or off by a connectivity function C_{ij}. The spring’s equilibrium length, d_{ij}^{ref}, is interpolated along the path.

$$V_{ij}(\mathbf{R}_k) = \frac{1}{2}k_{ij}\left(d_{ij}(\mathbf{R}_k) - d_{ij}^{\text{ref}}(k)\right)^2 C_{ij} \tag{8}$$

The reference distance for a pair in image k is a linear interpolation between its distance in the reactant ($k = 0$) and product ($k = P$).

$$d_{ij}^{\text{ref}}(k) = \left(1 - \frac{k}{P}\right) d_{ij}(\mathbf{R}_0) + \frac{k}{P} d_{ij}(\mathbf{R}_P) \tag{9}$$

The connectivity function C_{ij} ensures that a potential is only applied between atoms that are considered bonded in either the reactant or the product.

$$C_{ij} = \begin{cases} 1 & \text{if bond } (i, j) \text{ exists in } \mathbf{R}_0 \text{ or } \mathbf{R}_P \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The primary benefit of the IDPP method is that it often generates a more chemically reasonable initial path. Because the potential is based on the equilibrium bond connectivity of the endpoints, it avoids the unphysical atomic overlaps that often occur with linear interpolation. However, a significant limitation is that the method is not permutationally invariant, and has no explict handling for mass. The IDPP relies on a fixed, one-to-one mapping of atomic indices from reactant to product. If two identical atoms are swapped in the product coordinates, the IDPP method will generate a completely different and likely unphysical path, as it cannot recognize chemical equivalence.

2.2.2 The NEB Force

The core of the NEB method is the definition of the force acting on each intermediate image i. This force is constructed to prevent the path from sliding downhill, colloquially called "corner cutting", and to maintain equal spacing of the images. It is composed of the perpendicular component of the true force and the parallel component of the spring force:

$$\mathbf{F}_i^{\text{NEB}} = \mathbf{F}_i^{\perp} + \mathbf{F}_i^{||,\text{spring}} \tag{11}$$

Here, $\mathbf{F}_i^{\perp}$ is the component of the atomic forces on the Potential Energy Surface, $\mathbf{F}_i^{\text{true}} = -\nabla V(\mathbf{R}_i)$, perpendicular to the path tangent. $\mathbf{F}_i^{||,\text{spring}}$ is the component of the spring force parallel to the path tangent.

2.2.3 Tangent Vector Estimation

A crucial element is the estimation of the local tangent to the path, $\hat{\tau}_i$, at each image. We include, in EON, several schemes:

Old Tangent A simple central-difference vector [30] [4].

$$\hat{\tau}_i = \text{normalize}(\mathbf{R}_{i+1} - \mathbf{R}_{i-1}) \tag{12}$$

Improved Tangent A more robust method [31] that prevents kinks in the path. It selects the tangent based on the local energy landscape:

$$\hat{\tau}_i = \begin{cases} \text{normalize}(\mathbf{R}_{i+1} - \mathbf{R}_i) & \text{if } V_{i+1} > V_i > V_{i-1} \\ \text{normalize}(\mathbf{R}_i - \mathbf{R}_{i-1}) & \text{if } V_{i-1} > V_i > V_{i+1} \\ \text{weighted average} & \text{otherwise} \end{cases} \tag{13}$$

At extrema, the tangent is a weighted average of the vectors to the neighboring images, giving preference to the vector on the higher energy side.

2.2.4 Force Components

With the tangent $\hat{\tau}_i$ defined, the force components are calculated as:

Perpendicular Force This component moves the image to minimize energy perpendicular to the path, relaxing it onto the MEP.

$$\mathbf{F}_i^{\perp} = \mathbf{F}_i^{\text{true}} - (\mathbf{F}_i^{\text{true}} \cdot \hat{\tau}_i)\hat{\tau}_i \tag{14}$$

[4] Implemented as a forward difference in EON [29]

Parallel Spring Force This component adjusts the position of the image along the path to ensure equal spacing.

$$\mathbf{F}_i^{||,\mathrm{spring}} = k(|\mathbf{R}_{i+1} - \mathbf{R}_i| - |\mathbf{R}_i - \mathbf{R}_{i-1}|)\hat{\tau}_i \tag{15}$$

where k defines the spring constant.

2.2.5 Implementation Modalities and Improvements

Variations of the NEB method in EON also include:

Climbing-image nudged elastic band (CI-NEB) To accurately locate the saddlepoint, the spring force on the highest energy image (the "climbing image," $\mathbf{R}_{\mathrm{climb}}$) is removed, and the parallel component of its true force is inverted. This forces the image to move uphill along the path to converge exactly on the saddle point.

$$\mathbf{F}_{\mathrm{climb}} = \mathbf{F}_{\mathrm{climb}}^{\mathrm{true}} - 2(\mathbf{F}_{\mathrm{climb}}^{\mathrm{true}} \cdot \hat{\tau}_{\mathrm{climb}})\hat{\tau}_{\mathrm{climb}} \tag{16}$$

Energy-Weighted Springs In a standard NEB calculation, a uniform spring constant connects all images. A more adaptive approach, known as the energy-weighted spring method [33], dynamically adjusts the spring constants along the path. This method applies stronger (stiffer) springs in high-energy regions, typically near the transition state, and weaker springs in lower-energy regions. This procedure effectively concentrates images around the saddle point, improving the resolution of the reaction barrier without necessitating an increase in the total number of images.

The spring constant, k_i, for the segment connecting image $i-1$ and image i, is determined by a linear interpolation between a defined maximum spring constant, k_{max}, and a minimum, k_{min}. The interpolation depends on the energy of that segment.

We define a reference energy, E_{ref}, as the lower of the two endpoint energies (reactant or product). We also identify the maximum energy found along the current path, E_{max}. For each spring segment between images $i-1$ and i, we define an effective energy, E_i, as the higher of the two adjacent image energies:

$$E_i = \max\left(V(\mathbf{R}_{i-1}), V(\mathbf{R}_i)\right) \tag{17}$$

If this effective energy E_i exceeds the reference energy E_{ref}, we calculate a dimensionless weighting factor, α_i:

$$\alpha_i = \frac{E_{\mathrm{max}} - E_i}{E_{\mathrm{max}} - E_{\mathrm{ref}}} \tag{18}$$

This factor, α_i, ranges from 0 (when $E_i = E_{\mathrm{max}}$) to 1 (when $E_i = E_{\mathrm{ref}}$). This factor interpolates the spring constant k_i between k_{max} and k_{min}. If the segment's energy E_i does not exceed E_{ref}, the spring constant defaults to the minimum value, k_{min}. This leads to:

$$k_i = \begin{cases} (1-\alpha_i)k_{\mathrm{max}} + \alpha_i k_{\mathrm{min}}, & \text{if } E_i > E_{\mathrm{ref}} \\ k_{\mathrm{min}}, & \text{otherwise} \end{cases} \tag{19}$$

These dynamically adjusted spring constants are then used to calculate the parallel component of the spring force, $\mathbf{F}_i^{s,\|}$, acting on each image i. This force depends on the tension from the springs on its left (k_i) and right (k_{i+1}):

$$\mathbf{F}_i^{s,\|} = (k_{i+1}|\mathbf{R}_{i+1} - \mathbf{R}_i| - k_i|\mathbf{R}_i - \mathbf{R}_{i-1}|)\,\hat{\boldsymbol{\tau}}_i \tag{20}$$

Here, $\hat{\boldsymbol{\tau}}_i$ represents the normalized tangent vector at image i. This formulation ensures that the net effect of the springs pulls images toward the saddle point, refining the path's most critical region. It should be noted that the implicit assumption here is of a single maxima along the path, and the formulation as implemented and presented does not have special handling for spanning multiple basins. Numerical jitter may be used to handle degeneracy from the denominator in Eq. 18.

2.2.6 Optimization and Path Analysis

The set of movable images is relaxed using a standard optimization algorithm (e.g., L-BFGS), which iteratively updates the image positions based on the calculated NEB forces until a convergence criterion is met. In EON the most common criteria is on the largest force component on any atom.

After convergence, a Hermite polynomial interpolation between the final image energies and forces estimates the location and height of the energy barrier, providing a more refined value than the energy of the highest image alone.

2.3 Acceleration strategies

The core methodology described thus-far has formed an integral part of computational chemistry calculations for several decades. In order to obtain results on larger systems or longer time-scales, these must be accelerated. Within this thesis we consider two primary approaches to this acceleration:

Software design As an engineering endeavor, generating efficient machine code leads to faster solutions and thus unlocks larger systems. Under this broad remit falls parallelization strategies, caching, handling large data, better use of specialized hardware like graphics processing unit (GPU) computations etc.

Algorithmic improvements The total computational time depends on the algorithms employed, which in turn demand energy and atomic force evaluations from high-level theory. Consequently, algorithms that minimize the number of calculations required to achieve a specific solution accuracy constitute the primary metric for optimization. Utilizing cheaper approximate surfaces for portions of the algorithm represents a significant focus of current research.

An ideal surrogate demands ease of computation, autonomy from manual intervention, and "transparent" progression without the manual construction of training sets. We therefore mandate that the method:

Learns on the fly Efficiently learning from low amounts of often correlated data.

Encode physics Providing, at minimum, a functional form that aids analysis.

Trains quickly Enabling the efficient incorporation of new points into the model.

We explicitly define the scope: strictly avoiding the goal of matching the underlying potential energy surface globally. These surrogates serve only to guide algorithms toward specific points in the solution space; thus, absolute errors in energy and forces hold little value compared to local gradient accuracy. Similarly, we exclude general energy surface construction, the basis of machine learning interatomic potentials. We therefore focus on regression techniques [11]. Computing atomic descriptors are based on cartesian coordinates [34] requires both user intervention for parameter selection, and increases the computational cost of on-the-fly learning. Consequently, an ideal surrogate for our consideration operates directly on raw Cartesian coordinates. We briefly evaluate the candidates:

Linear regression The oldest fitting technique, which may be competitive with enough data over a small region of space, but only with descriptors to account for non-linearities, which are often incomplete by construction [35].

Neural networks Deep neural networks [36] interpolate arbitrary systems well and can encode specific qualities through filters (e.g., convolutional architectures). However, they tend to overfit in the low-data regime [37], and require careful selection of both data and long training times for the hyperparameters. The resulting models have no closed form analytical solution and rely on being able to backpropagate gradients [38, 39], which also forms a computational bottleneck.

Gaussian processes Often understood as limiting cases of spline models [40], neural networks [41, 42], quadrature rules [43], stochastic partial differential equations [44], finite realizations of these methods form multivariate normal distributions [45] which are analytically and computationally compact. The connection to standard statistics enables a rich set of uncertainity quantification, and the ability to constrain the function spaces based on distance measures between data points makes for more interpretable surrogates.

With this in mind, we move on with further contextualizing the Gaussian Process for molecular systems, though we note that descriptor based methods have also been applied for activated searches [46].

2.4 Gaussian Process Regression

We begin by positing that the unknown Potential Energy Surface, a function $f(\mathbf{x})$, represents a single realization from a Gaussian Process. A Gaussian Process defines a probability distribution over a space of functions. Any finite collection of function values drawn from this process follows a joint multivariate normal (MVN) distribution. It is worth noting that such an assumption can be justified rather rigorously for molecular potential energy surfaces with pronounced global minimum as demonstrated by [47] in terms of an asymptotic study of vibrational degrees of freedom, which leads to the potential energy surface on a random coordinate frame manifesting as a sum of many contributions, which in turn through the Central limit theorem [11] leads to an MVN perspective, equivalent to a perturbative approach for small molecules.

Such theoretical modeling of molecular potential energy surfaces [47] rests upon three postulates: the potential constants represent averages over all unitary transformations; the topology possesses a distinct global minimum; and the number of vibrational modes approaches the asymptotic limit. While these constraints, in particular the topological requirement strictly preclude the multiple-minima landscapes inherent to the reactive systems under investigation here, the derivation nevertheless offers robust physical support for approximating the energy surface locally as a multivariate Gaussian.

In practice, one never works with the infinite-dimensional function directly. Instead, we select a finite set of **M** input configurations, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$. The GP specifies that the corresponding vector of function outputs, $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_M)]^T$, constitutes a single draw from an **M**-dimensional MVN. This finite vector becomes our computational representation of the underlying function.

$$\mathbf{o}(\mathbf{x}) = \begin{pmatrix} E(\mathbf{x}) \\ \mathbf{F}(\mathbf{x}) \end{pmatrix} \in \mathbb{R}^{3N+1} \tag{21}$$

When we evaluate the Potential Energy Surface at a set of **M** distinct configurations, $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_M\}$, the Gaussian Process framework posits that the collection of all corresponding observation vectors follows a single, large MVN distribution:

$$p(\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_M) = \mathcal{N}(\mathbf{0}, \mathbf{K}) \tag{22}$$

The full covariance matrix **K** is built from a kernel function $k(x_i, x_j)$ that defines the similarity between any two configurations [45, 48]. Because the observation vector o contains both a scalar (energy) and a vector (forces), the kernel itself produces a block covariance matrix for any two configurations:

$$\mathbf{o}(\mathbf{r}) = \begin{pmatrix} E(\mathbf{r}) \\ \mathbf{F}(\mathbf{r}) \end{pmatrix} = \begin{pmatrix} E(\mathbf{r}) & F_{x_1}(\mathbf{r}) & F_{y_1}(\mathbf{r}) & F_{z_1}(\mathbf{r}) & F_{x_2}(\mathbf{r}) & \ldots & F_{z_N}(\mathbf{r}) \end{pmatrix}^T \in \mathbb{R}^{3N+1} \tag{23}$$

where:

- $E(\mathbf{r})$ is the potential energy at atomic configuration $\mathbf{r}$.
- $\mathbf{F}(\mathbf{r})$ is the $3N-$ dimensional vector of atomic forces, where F_{x_i}, F_{y_i}, and F_{z_i} are the x, y, and z components of the force on atom i, respectively. N is the number of atoms.

Each draw from the Gaussian Process forms a concrete MVN fully specified by a mean vector **m** and a covariance matrix **K**. We construct these from a mean function $m(\mathbf{x})$, assumed to be zero throughout this work, and a covariance function or kernel $k(\mathbf{x}, \mathbf{x}')$ which determines the covariance between the energy and forces at different atomic configurations $\mathbf{x}$ and $\mathbf{x}'$. The choice of kernel encodes prior assumptions about the functional form of the Potential Energy Surface. For systems with strong repulsive forces at short distances, the most commonly used kernel, the infinitely differentiable squared exponential (SE) kernel can be too restrictive [49], which is only partially resolved by choosing different kernels like the Matern. The stationarity of the SE kernel, i.e. the assumption of uniform fluctuations across the domain makes it difficult to handle the steep gradients of repulsive walls in Cartesian space. By transforming the input space through an inverse distance metric, we effectively “precondition” the energy surface by homogenizing the effective length scale of the landscape, which enables the Gaussian Process to resolve high gradient regions without sacrificing resolution in energetic wells. We therefore employ a SE kernel based on an inverse distance metric [49, 50] since it provides a strong physical prior, leveling out the sharp increase in repulsive force when pairs of atoms get too close.

For the chosen design sites $\mathbf{X}$, the components of the mean vector and covariance matrix are:

$$(\mathbf{m})_i = m(\mathbf{x}_i) \quad \text{and} \quad (\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \tag{24}$$

In other words, we model the joint distribution of the energy and forces at multiple configurations $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_M\}$ as a multivariate Gaussian distribution Eq. **??** and we now identify the elements of $\mathbf{K}$ are given by a kernel function $k(\mathbf{r}_i, \mathbf{r}_j)$ that measures the similarity between configurations $\mathbf{r}_i$ and $\mathbf{r}_j$. This kernel operates on the input space of atomic geometries ($\mathbb{R}^{N\times 3}$) and outputs the covariance between the combined energy and force vectors ($\mathbb{R}^{3N+1}$) at those geometries.

Including force derivatives improves performance for models with limited samples [51]. The covariance between the combined energy and force vectors at two different geometries $\mathbf{r}$ and $\mathbf{r}'$ is given by:

$$\mathrm{Cov}(\mathbf{o}(\mathbf{r}), \mathbf{o}(\mathbf{r}')) = \begin{pmatrix} k_{EE}(\mathbf{r}, \mathbf{r}') & \mathbf{k}_{EF}(\mathbf{r}, \mathbf{r}') \\ \mathbf{k}_{FE}(\mathbf{r}, \mathbf{r}') & \mathbf{K}_{FF}(\mathbf{r}, \mathbf{r}') \end{pmatrix} \tag{25}$$

where

- $k_{EE}(\mathbf{r}, \mathbf{r}') \in \mathbb{R}$ is the covariance between the energies.
- $\mathbf{k}_{EF}(\mathbf{r}, \mathbf{r}') \in \mathbb{R}^{1\times 3N}$ is the covariance between the energy at $\mathbf{r}$ and the forces at $\mathbf{r}'$.
- $\mathbf{k}_{FE}(\mathbf{r}, \mathbf{r}') = \mathbf{k}_{EF}(\mathbf{r}', \mathbf{r})^\top \in \mathbb{R}^{3N\times 1}$ is the covariance between the forces at $\mathbf{r}$ and the energy at $\mathbf{r}'$.
- $\mathbf{K}_{FF}(\mathbf{r}, \mathbf{r}') \in \mathbb{R}^{3N\times 3N}$ is the covariance matrix between the forces.

For energy-energy covariance, this kernel takes the form:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_c^2 + \sigma_f^2 \exp\left(-\frac{1}{2}\sum_i \sum_{j>i} \left(\frac{1/r_{ij}(\mathbf{x}) - 1/r_{ij}(\mathbf{x}')}{l_{\phi(i,j)}}\right)^2\right) \tag{26}$$

where σ_f^2 is the signal variance, σ_c^2 is a constant offset, and $l_{\phi(i,j)}$ is the characteristic length scale for a specific pair type of atoms $\phi(i, j)$ [5].

The force-related blocks of the covariance matrix (k_{EF}, k_{FE}, K_{FF}) are derived by differentiating the energy-energy kernel with respect to the atomic coordinates, leveraging the relationship $\mathbf{F} = -\nabla E$. This requires the first and second partial derivatives of the squared distance measure, $\mathcal{D}^2_{1/r} = \sum \left(\frac{\Delta(1/r)}{l}\right)^2$:

$$\frac{\partial \mathcal{D}^2_{1/r}(\mathbf{x}, \mathbf{x}')}{\partial x_{i,d}} = \sum_{j\neq i} \left[\frac{-2(x_{i,d} - x_{j,d})}{l^2_{\phi(i,j)} r^3_{ij}(\mathbf{x})} \left(\frac{1}{r_{ij}(\mathbf{x})} - \frac{1}{r_{ij}(\mathbf{x}')}\right)\right] \tag{27}$$

[5] only pairs are considered, a single atom type, O, for instance will not have an O-O term

$$\frac{\partial^2 \mathcal{D}^2_{1/r}(\mathbf{x},\mathbf{x}')}{\partial x_{i_1,d_1}\partial x'_{i_2,d_2}} = \begin{cases} \frac{2(x_{i_1,d_1}-x_{i_2,d_1})(x'_{i_1,d_2}-x'_{i_2,d_2})}{l^2_{\phi(i_1,i_2)} r^3_{i_1,i_2}(\mathbf{x}) r^3_{i_1,i_2}(\mathbf{x}')}, & \text{if } i_1 \neq i_2 \\ \sum_{j\neq i} \frac{-2(x_{i,d_1}-x_{j,d_1})(x'_{i,d_2}-x'_{j,d_2})}{l^2_{\phi(i,j)} r^3_{ij}(\mathbf{x}) r^3_{ij}(\mathbf{x}')}, & \text{if } i_1 = i_2 = i \end{cases} \tag{28}$$

The kernel hyperparameters $\boldsymbol{\theta} = \{\sigma_f^2, \sigma_c^2, l_\psi, \dots\}$ are not known *a-priori*. Most commonly, the hyperparameters derive from the marginal likelihood. Practically, optimizing takes place in log-space ($\eta = \log\theta$) since this provides better numerical scaling through two benefits.

Firstly, it implicitly maintains the strict positivity constraints required by variances and length scales. The mapping $\theta = \exp(\eta)$ ensures that the optimization algorithm, which typically operates in unconstrained Euclidean space, yields strictly positive physical parameters without requiring barrier functions or Lagrange multipliers.

Secondly, logarithmic scaling preconditions the objective function landscape. In chemical physics applications, hyperparameters frequently vary by orders of magnitude. A gradient step size appropriate for a large signal variance σ_f^2 could prove disastrously large for a short-range length scale l_ψ. Working in the log-space equilibrates these sensitivities, as a constant step size in $\boldsymbol{\eta}$ corresponds to a proportional change in $\boldsymbol{\theta}$ rather than an absolute one. This effectively balances the gradient contributions across disparate dimensions and accelerates convergence.

From a practical standpoint, the gradient computation for the transformed parameters follows the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \eta_i} = \frac{\partial \mathcal{L}}{\partial \theta_i}\frac{\partial \theta_i}{\partial \eta_i} = \frac{\partial \mathcal{L}}{\partial \theta_i}\theta_i \tag{29}$$

This scaling factor θ_i naturally dampens the gradient for small parameters and amplifies it for large ones, further stabilizing the descent. Thus the kernel hyperparameters are learned from the training data by maximizing the logarithm of the marginal log-likelihood (MLL):

$$\log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{1}{2}\log\det(\mathbf{K}) - \frac{M(3N+1)}{2}\log(2\pi) \tag{30}$$

This optimization is performed using gradient-based methods, which require the partial derivatives of the kernel with respect to each hyperparameter. For the length scales l_ψ, this involves the following derivatives of the distance measure:

$$\frac{\partial \mathcal{D}^2_{1/r}(\mathbf{x},\mathbf{x}')}{\partial l_\psi} = \sum_{\substack{i,j>i \\ \phi(i,j)=\psi}} \frac{-2\left(\frac{1}{r_{ij}(\mathbf{x})} - \frac{1}{r_{ij}(\mathbf{x}')}\right)^2}{l_\psi^3} \tag{31}$$

$$\frac{\partial^2 \mathcal{D}^2_{1/r}(\mathbf{x},\mathbf{x}')}{\partial x_{i,d}\partial l_\psi} = \sum_{\substack{j\neq i \\ \phi(i,j)=\psi}} \left[\frac{4(x_{i,d}-x_{j,d})}{l_\psi^3 r_{ij}^3(\mathbf{x})}\left(\frac{1}{r_{ij}(\mathbf{x})} - \frac{1}{r_{ij}(\mathbf{x}')}\right)\right] \tag{32}$$

$$\frac{\partial^3 \mathcal{D}^2_{1/r}(\mathbf{x},\mathbf{x}')}{\partial x_{i_1,d_1}\partial x'_{i_2,d_2}\partial l_\psi} = \begin{cases} 0, & \text{if } i_1 \neq i_2 \wedge \phi(i_1,i_2) \neq \psi \\ -\frac{4(x_{i_1,d_1}-x_{i_2,d_1})(x'_{i_1,d_2}-x'_{i_2,d_2})}{l^3_\psi r^3_{i_1,i_2}(\mathbf{x}) r^3_{i_1,i_2}(\mathbf{x}')}, & \text{if } i_1 \neq i_2 \wedge \phi(i_1,i_2) = \psi \\ \sum_{\substack{j\neq i \\ \phi(i,j)=\psi}} \frac{4(x_{i,d_1}-x_{j,d_1})(x'_{i,d_2}-x'_{j,d_2})}{l^3_\psi r^3_{ij}(\mathbf{x}) r^3_{ij}(\mathbf{x}')}, & \text{if } i_1 = i_2 = i \end{cases} \tag{33}$$

Once the model is trained, it can be used to predict the energy and forces at new configurations $\mathbf{X}_*$. The posterior predictive mean gives the best estimate for the surface at the new locations:

$$\bar{\mathbf{f}}_* = \mathbf{K}_{*y}(\mathbf{K}_{yy} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y} \tag{34}$$

where $\mathbf{y}$ is the vector of training observations, $\mathbf{K}_{yy}$ is the covariance of the training data, $\mathbf{K}_{*y}$ is the covariance between the test and training points, and the $\sigma_n^2\mathbf{I}$ term is a regularization or noise term that ensures numerical stability. The posterior predictive covariance quantifies the model uncertainty:

$$\mathrm{cov}(\mathbf{f}_*) = \mathbf{K}_{**} - \mathbf{K}_{*y}(\mathbf{K}_{yy} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_{y*} \tag{35}$$

From here, adding atomic features [34, 52] leads to the smoothed overlap of atomic positions class of models [53, 54] and other machine learned interatomic potentials [55]. Pivoting slightly towards active, or reinforcement learning [56], we utilize the posterior mean to guide the search for stationary points on a series of approximate Potential Energy Surface. The posterior covariance, while a formal measure of uncertainty, may be unreliable in an iterative refitting scheme (Section 7.4) and we therefore disregard it here. Essentially, because the global hyperparameters are re-optimized with each new data point, a local reduction in variance does not reliably indicate an improvement in the model's true accuracy.

2.4.1 Scaling in Time and Storage

The computational cost of Gaussian Process Regression [57] is dominated by the inversion of the covariance matrix $\mathbf{K}$, and the determination of the hyperparameters for conditioning on the data as each new point is acquired. More precisely, the costs of a Gaussian Process involve:

Storage The covariance matrix $\mathbf{K}$ carries dimensions $(M(3N+1)) \times (M(3N+1))$, where M equals the number of training configurations. Storage therefore scales as $O(M^2N^2)$.

Fitting Time Inverting $\mathbf{K}$ requires $O(M^3N^3)$. This cubic scaling in both M and N renders standard GPR expensive for large systems and datasets.

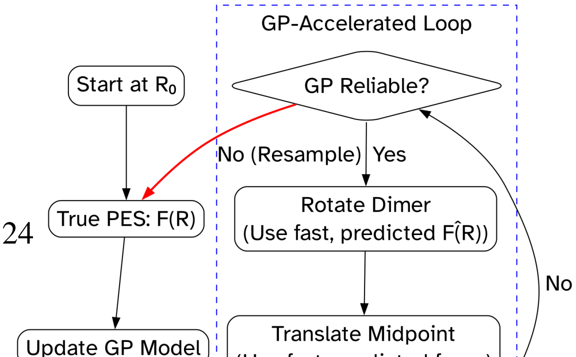
Inference time Once the kernel components are determined, prediction involves a single kernel vector between the new point and training points, so this scales linearly with the number of training points.

The key advantage of a Gaussian Process approach is that we can constrain the functional form of the posterior, determined by the inverse distance kernel models physical constraints, while the model remains relatively cheap to re-fit and predict with. Physical constraints like smoothness may also be enforced, and data agumentation can encode non-linear prior assumpsions.

2.5 Gaussian Process as an accelerator

A simplified flowchart of the logic is presented in Figure 2.3.

The GPDimer method constructs a local surrogate model for the Potential Energy Surface through a finite, targeted set of samples. The approach functions like a reinforcement learning agent [56]. At each step of a search, the



model decides whether to trust its current surrogate Potential Energy Surface or to query the "true" Potential Energy Surface via an expensive quantum chemical calculation. This "on-the-fly" model building refines a highly local and task-specific Potential Energy Surface with a minimal number of data points [6].

As breifly covered before, this strategy differs fundamentally from that of a general-purpose machine learned interatomic potential (MLIP). The principal objective of an MLIP: create a single, global, transferable Potential Energy Surface, which demands vast datasets that may contain millions of configurations. To achieve transferability across diverse chemical environments, the MLIP architecture must respect the system's physical symmetries. This requirement demands outputs that remain invariant to translation, rotation, and permutation of identical atoms, often achieved through atom-centered symmetry functions [59] or specialized equivariant neural-network layers [52, 60]. Although such models can deliver high energy accuracy, force accuracy generally remains worse and improves more slowly with training-set size, with errors often exceeding 0.1 eV/ Å [61, 62]. Because forces drive the methods in this thesis, most MLIPs offer limited value here. Foundational models like [63] with easy finetuning support may still prove useful.

By contrast, the GPDimer method pursues a different objective, that is, high fidelity within a localized region of the Potential Energy Surface tied to a specific process (e.g., a single saddle search). This focused scope obviates large-scale sampling. We develop models that operate effectively on raw Cartesian coordinates, bypassing the heavy data and architectural requirements of global MLIPs.

2.6 Conclusions

This chapter has delineated the theoretical and algorithmic framework necessary for navigating high-dimensional potential energy surfaces. We detailed the topological search methods based on the number of known minima, specifically the single ended Dimer and double ended Nudged Elastic Band algorithms which serve as the primary vehicles for locating first-order saddle points and minimum energy pathways. While these techniques provide robust convergence properties, in practice, their reliance on expensive *ab-initio* force evaluations necessitates an acceleration strategy capable of operating within the strict data constraints of on-the-fly exploration.

Our analysis identified Gaussian Process Regression as the optimal surrogate for this regime. By combining the asymptotic derivation of local energy landscapes with a physically motivated

[6] we will demonstrate results typically around 30 samples [50, 58]

inverse-distance kernel, we show how one constructs a model that balances flexibility with the regularization required by repulsive atomic cores. Crucially, this approach bypasses the need for the extensive datasets required by global machine learning potentials, favoring instead a rigorous local approximation that respects the underlying physics of the reaction channel. With these mathematical and computational foundations established, we later proceed to their practical implementation and application in the study of complex reactive systems.

However, a surrogate model acts merely as an efficient interpolator; it possesses no intrinsic knowledge of the quantum mechanics. The fidelity of any reaction path search depends ultimately on the accuracy of the underlying electronic structure calculations. Furthermore, while we discuss methods to minimize the number of evaluations, the computational cost per evaluation remains a critical bottleneck, particularly for heavy-element systems where relativistic effects dominate. Therefore, before applying these exploration algorithms to complex reactive systems, we cover some eletronic structure theory for chemical systems and foray into a rigorous and efficient foundation for computing the potential energy surface for single atom systems. The subsequent chapter addresses this challenge, introducing a high-order finite element approach to the self-consistent field problem that balances relativistic accuracy with numerical efficiency.

3 Electronic structure calculations

> the chemical difference between silver and gold may mainly be a relativistic effect.
>
> P. Pyykkö
> Chemistry – A European Journal

This chapter is based on Ondřej Čertík, John E. Pask, Isuru Fernando, Rohit Goswami, N. Sukumar, Lee. A. Collins, Gianmarco Manzini, and Jiří Vackář. "High-Order Finite Element Method for Atomic Structure Calculations." In: *Computer Physics Communications* (Dec. 2023), p. 109051. DOI: 10.1016/j.cpc.2023.109051

To explore the Potential Energy Surface with the methods described previously, we must compute energies and atomic forces for specific nuclear configurations. This task reduces to solving the many-electron Schrödinger equation in a form that scales to useful systems. We proceed by choice of representation: we begin with a mean field to obtain a tractable one-body problem; then we refine the physics or numerics as necessary.

After a brief introduction to the general electronic structure methods for chemical systems, we focus on a state of the art solver, `featom` which solves for isolated single atom systems, the Dirac and Schrödinger equations.

3.1 Mean-field quantum chemistry

The Hartree approximation represents the simplest mean field theory approximation. It assumes that each electron moves in the Coulomb field generated by nuclei and the spherically averaged density of all other electrons [65]. This local, multiplicative field removes two-body integrals and invites efficient solvers. However, it leaves self-interaction uncorrected. The Hartree-Fock (HF) method [66] improves upon this by treating electron correlation through an antisymmetrized wave function characterized by a single Slater determinant to incorporate exact exchange. As such, the non-local Fock operator introduces an integral operator which couples space points and removes self-interaction, though it considers only exchange correlations.

For closed shells, the HF Fock operator reads

$$\hat{F} = \hat{h} + \hat{J}[n] - \hat{K}[\{\phi\}], \qquad (\hat{K}\psi)(\mathbf{r}_1) = \sum_j \phi_j(\mathbf{r}_1) \int \frac{\phi_j^*(\mathbf{r}_2)\psi(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_2, \tag{36}$$

with $\hat{h} = -\frac{1}{2}\nabla^2 + v(\mathbf{r})$. In a spherical atom, orbitals separate as $P_{nl}(r) = rR_{nl}(r)$, and the

Hartree term reduces to a 1D Poisson solve for $V_H(r)$. Exchange remains nonlocal after angular reduction. A multipole expansion yields

$$(\hat{K}\,P_a)(r_1) = \sum_{b\in\mathrm{occ}} \sum_{k=|l_a-l_b|}^{l_a+l_b} g_k(l_a,l_b)\,Y_k^{(ab)}(r_1)\,P_b(r_1), \tag{37a}$$

$$Y_k^{(ab)}(r) = \int_0^\infty P_a(r')P_b(r')\,\frac{r_<^k}{r_>^{k+1}}\,dr', \tag{37b}$$

$$g_k(l_a,l_b) = \frac{4\pi}{2k+1}\,(2l_b+1)\begin{pmatrix} l_a & k & l_b \\ 0 & 0 & 0 \end{pmatrix}^2. \tag{37c}$$

From this basic formulation, subsequent expansions involving the occupancy of spin-orbitals yield unrestricted, restricted, and restricted open-shell Hartree-Fock methods [67]. All of these are solved iteratively through the self consistent field (SCF) approach. The addition of spin-orbit occupancies at this stage leaves the fundamental nature of the mean field approximation unchanged. Since instantaneous interactions typically prove more repulsive than an average interaction, the difference between the exact energy and the SCF approximation constitutes the "correlation" energy. More sophisticated treatments of the two-electron interaction can approximate this energy, leading to "post-Hartree-Fock" methods [68]. These prescriptions offer mathematical rigor but scale too poorly for frequent computation in large systems. A numerically efficient alternative arrives from a different angle: grounding the theory in the density of electrons rather than exhaustively enumerating their positions.

3.1.1 Kohn–Sham DFT: exact in principle, orbital constrained by construction

The Hohenberg–Kohn [69] theorem formulates the ground-state energy in terms of the electron density. Kohn-Sham (KS) density functional theory (DFT) builds a noninteracting system of virtual electrons that reproduces the interacting density, which brings the concept of orbitals back to the fore. The KS equations employ a single multiplicative potential

$$V_{\mathrm{eff}}[n](r) = v(r) + V_H[n](r) + v_{xc}[n](r). \tag{38}$$

If $E_{xc}[n]$ were known, we could recover the exact density and energy. In practice we choose approximations [70] (LDA/GGA/meta-GGA) and gain correlation corrections at modest cost. This local-potential form dovetails with radial finite elements and enables a fast, stable self-consistent loop.

We find the Hartree potential by solving the Poisson equation, $\nabla^2 V_H(\mathbf{x}) = -4\pi n(\mathbf{x})$, where $n(r)$ represents the radial electron density constructed from the wave functions. Because the potential depends on the wave functions and the wave functions depend on the potential, these equations require iterative solution until they reach self-consistency.

3.2 The Physical and Mathematical Problem

We describe the electronic structure of an isolated, spherically symmetric atom at two primary levels of theory, depending on the required rigor: the non-relativistic Schrödinger equation and the relativistic Dirac equation.

In computational terms, the primary entities in this domain are the `Atom`, the `ElectronState`, and the `Potential`.

- An `Atom` is the central entity, characterized by its nuclear charge (Z). It is composed of a set of `ElectronState` s.
- An `ElectronState` is defined by its quantum numbers (e.g., n, l, or κ) and is primarily described by a `WaveFunction` entity and its corresponding energy `Eigenvalue`.
- The `Potential` is an entity that governs the behavior of the `ElectronState` s.

Since the KS framework of DFT maps the complex many-electron problem onto a tractable set of single-particle equations, a cyclic dependence arises: the potential depends on the wave functions (via the electron density) and the wave functions depend on the potential. We must solve the KS equations iteratively to a self-consistent solution. The governing mathematical models for the `WaveFunction` entity thus become the radial Schrödinger and Dirac equations, solved within this self-consistent loop. The discussion thus far does not restrict the number of atoms, however, for the remainder of the chapter we consider single atom systems, though we consider all electrons without approximations typically applied to larger systems [7].

3.2.1 The Radial Schrödinger Equation

For a spherically symmetric potential $V(r)$, the wave function separates into radial and angular parts, $\psi_{nlm}(\mathbf{x}) = R_{nl}(r)\,Y_{lm}(\theta, \phi)$. By substituting $P_{nl}(r) = rR_{nl}(r)$, the problem reduces to solving the one-dimensional radial Schrödinger equation:

$$-\frac{1}{2}P''_{nl}(r) + \left(V(r) + \frac{l(l+1)}{2r^2}\right)P_{nl}(r) = EP_{nl}(r) \tag{39}$$

where l is the angular momentum quantum number and E is the energy eigenvalue. The function $P_{nl}(r)$ must be normalized such that $\int_0^\infty P^2_{nl}(r)\,\mathrm{d}r = 1$.

3.2.2 The Radial Dirac Equation

For heavy atoms, the appropriate single-particle theory requires a relativistic treatment. The central-field Dirac equation leads to two coupled first-order radial equations for the large and small components $(P_{n\kappa}, Q_{n\kappa})$ of a four-component spinor, with the relativistic quantum number κ encoding (l, j) [72].

$$P'_{n\kappa}(r) = -\frac{\kappa}{r}P_{n\kappa}(r) + \left(\frac{E - V(r)}{c} + 2c\right)Q_{n\kappa}(r), \tag{40a}$$

$$Q'_{n\kappa}(r) = -\left(E - \frac{V(r)}{c}\right)P_{n\kappa}(r) + \frac{\kappa}{r}Q_{n\kappa}(r), \tag{40b}$$

where c is the speed of light and κ is the relativistic quantum number that encodes both total and orbital angular momentum.

[7] often treated with non-relativistic methods since core electrons are represented with a pseudo-potential or projected augmented wave [71]

The Dirac Hamiltonian possesses a spectrum unbounded from below, a feature that historically plagued basis-set discretizations with “variational collapse” and spurious states [72, 73]. Typical remedies over the past three decades involve shooting methods [74] which require trial solutions for each eigenfunction and convergence parameteres for the solver grid, complicating the need for robust and computationally efficient solutions. Basis set methods [75, 76, 77, 78] solve for all states at once through diagonalization, but struggle with the spurious states of he Dirac Hamiltonian, despite attempts to mitigate these with changes in basis for large and small components [75, 79, 80, 76, 81, 82], Hamiltonian modifications [83, 77, 78, 84, 85], and boundary value constraints [86, 87, 80, 77].

3.3 Robust finite element solvers for isolated atoms

We now consider a methodology which circumvents the spurious states of the Dirac while providing robust and efficient solutions with a high-order finite element basis. The `featom` code employs solves the governing equations using a finite element basis, implemented in modern modular Fortran. The radial coordinate is discretized into a mesh of finite elements, and within each element, the solution is expanded in a basis of high-order Lagrange polynomials defined on Gauss-Lobatto nodes. This spectral element approach yields exponential convergence with respect to the polynomial order, providing high accuracy with a relatively small number of basis functions. The success of this approach, however, rests on a cascade of intelligent choices in representation at the mathematical, numerical, and software levels.

The finite element expressions and code derived for the squared Dirac Hamiltonian [88, 83] are novel, and these have the same eigenfunctions as the operator, and remains bounded from below while preserving convergence to the Schrödinger limit [83]. Numerical stability arises from the squared operator providing second derivative terms, while known asymptotic forms near the origin allow side-stepping solving for large and small components of the Dirac wavefunction components. We view these concepts and their synthesis through the lens of varying representations below.

3.3.1 Layer 1: The Mathematical Representation (Squared Hamiltonian)

Direct discretization of the Dirac Hamiltonian operator poses significant difficulties because its energy spectrum remains unbounded from both above and below, leading to spurious, unphysical solutions. To circumvent this, we utilize a different mathematical representation: we solve the eigenvalue problem for the square of the Dirac Hamiltonian, $(H + \mathbb{I}c^2)^2$. This squared operator shares eigenfunctions with the original operator, and its eigenvalues relate simply as the square of the original eigenvalues, $(E + c^2)^2$. Crucially, the squared operator remains bounded from below. This allows the application of standard variational techniques, like the FEM, without generating spurious states.

A key principle here follows from eliminating the need for kinetic balance and other such constraints, and can instead proceed with Galerkin discretization [89, 90] directly in a polynomial basis, the same for both large and small components.

3.3.2 Layer 2: The Functional Representation (Asymptotic Correction)

For Coulombic potentials, the relativistic wave functions for states with $\kappa = \pm 1$ exhibit non-polynomial behavior near the origin ($r \longrightarrow 0$), with derivatives that diverge. This slow convergence poisons standard polynomial-based approximation schemes. We address this by changing the functional representation of the solution. Instead of solving for $P(r)$ and $Q(r)$ directly, we solve for modified functions $\tilde{P}(r) = P(r)/r^\beta$ and $\tilde{Q}(r) = Q(r)/r^\beta$, where $\beta = \sqrt{\kappa^2 - (Z/c)^2}$ is the known asymptotic exponent. The new functions $\tilde{P}(r)$ and $\tilde{Q}(r)$ are smooth and well-behaved at the origin, allowing for rapid, exponential convergence in the polynomial basis for all quantum states.

3.3.3 Layer 3: The Numerical Representation (The Golub-Welsch Algorithm)

Beyond the theoretical framework, the choice of numerical representation is critical for obtaining reliable results [91, 92]. A pivotal enhancement involved resolving a critical instability in the Gauss-Jacobi quadrature routine [93], essential for accurately integrating terms involving the asymptotic correction factor. The original implementation, based on a direct recurrence relation, was susceptible to floating-point errors. To correct this, the routine was re-implemented using the stable Golub-Welsch algorithm, which recasts the problem of finding quadrature points (x_i) and weights (w_i) for integrals of the form

$$\int_{-1}^{1} (1-x)^\alpha (1+x)^\beta f(x) \mathrm{d}x \approx \sum_{i=1}^{n} w_i f(x_i) \tag{41}$$

into a well-conditioned matrix eigenvalue problem. A symmetric tridiagonal Jacobi matrix, $\mathbf{J}$, is constructed, and its eigenvalues correspond precisely to the quadrature nodes x_i:

$$\mathbf{J}\mathbf{v}_i = x_i \mathbf{v}_i \tag{42}$$

The corresponding weights w_i are then calculated from the first components of the normalized eigenvectors $\mathbf{v}_i$:

$$w_i = \mu_0 (v_{i,1})^2, \quad \text{where} \quad \mu_0 = 2^{\alpha+\beta+1} \frac{\Gamma(\alpha+1)\Gamma(\beta+1)}{\Gamma(\alpha+\beta+2)} \tag{43}$$

This stable numerical representation was essential for guaranteeing the physical integrity of the simulations.

3.3.4 Layer 4: The Software Representation (Modern, Maintainable Code)

The final and most concrete layer of representation is the software itself. The `featom` library is a modern Fortran 2008 [94] implementation with a strong emphasis on modularity, reusability, and the absence of global state [95]. This design is crucial for enabling its use as a component in larger, more complex simulation workflows. Interoperability is guaranteed through backwards-compatible C bindings, allowing the high-performance Fortran core to be called from other languages like C++ or Python.

This robust code is supported by a professional software engineering infrastructure and involved introducing the flexible Meson build system alongside the existing Fortran Package Manager (`fpm`), establishing a comprehensive automated test harness, and refining continuous integration (CI) pipelines. This focus on the software representation ensures correctness through automated validation, lowers the barrier for collaboration, and guarantees long-term maintainability and scientific reproducibility.

3.4 From KS to HF: conceptually simple, practically subtle in spherical FE

Conceptually, Kohn–Sham replaces the nonlocal exchange operator by a multiplicative $v_{xc}[n](r)$, which fits perfectly into the radial `featom` framework that already solves Schrödinger/Dirac with a local potential. Practically, three nontrivial points arise:

Local vs nonlocal. HF exchange is nonlocal; KS uses a local $v_x[n](r)$. "Exact exchange" (EXX) within KS requires solving an optimized effective potential (OEP) [96] equation even in spherical symmetry. This adds a numerically involved integral equation for $v_x(r)$ to the formulation.

Orbital-dependent quantities are not a common potential Using $U_x^{(a)}(r)$ directly as "the" primary $v_x(r)$ breaks the KS structure and is unstable at nodes. A robust local proxy is the Slater average

$$v_x^{\text{Slater}}(r) = \frac{1}{n(r)} \sum_{a \in \text{occ}} \frac{f_a}{2} n_a(r)\, U_x^{(a)}(r), \qquad n_a(r) = \frac{P_a(r)^2}{4\pi r^2}, \quad n(r) = \sum_a f_a n_a(r), \tag{44}$$

which is multiplicative and stable in SCF.

Partial-wave assembly inside SCF Whether building HF (nonlocal) or local approximations, the spherical FE code benefits from the same partial-wave machinery: accurate $Y_k^{(ab)}(r)$, correct angular algebra, and careful treatment near nodes and $r \to 0$. Implementations that instead attempt to fold nonlocal exchange into a single multiplicative potential without these steps tend to diverge or collapse the spectrum.

A simple extension towards an HF/KS implementation in the radial FE code follows a "local-in-the-loop, exact-after" strategy [8]:

- In the SCF loop we use a multiplicative exchange potential of Slater–LDA form,

$$v_x^{\text{LDA}}(r) = -\left(\frac{3}{\pi}\right)^{1/3} n(r)^{1/3}, \tag{45}$$

which is local and stable to iterate together with $V_H(r)$ from the spherical Poisson solver.

- After SCF convergence we compute the exact Hartree-Fock (HF) exchange energy a posteriori using the multipole machinery in (37) on the converged orbitals, with numerically stable global-sorting and prefix–suffix accumulations for $Y_k^{(ab)}(r)$.
- This yields total energies close to the restricted HF benchmarks while preserving the robustness of a local SCF. For example, for a Beryllium atom (Z=4):

$$E_{\text{tot}} = -14.57067378 \text{ Ha} \quad (\text{RHF ref} - 14.57541503 \text{ Ha}),$$

a $\sim$ 4.7 mHa gap consistent with using a local v_x instead of nonlocal HF in the loop. Figure 3.1 demonstrates the convergence characterstics.

[8] gh-26 to featom

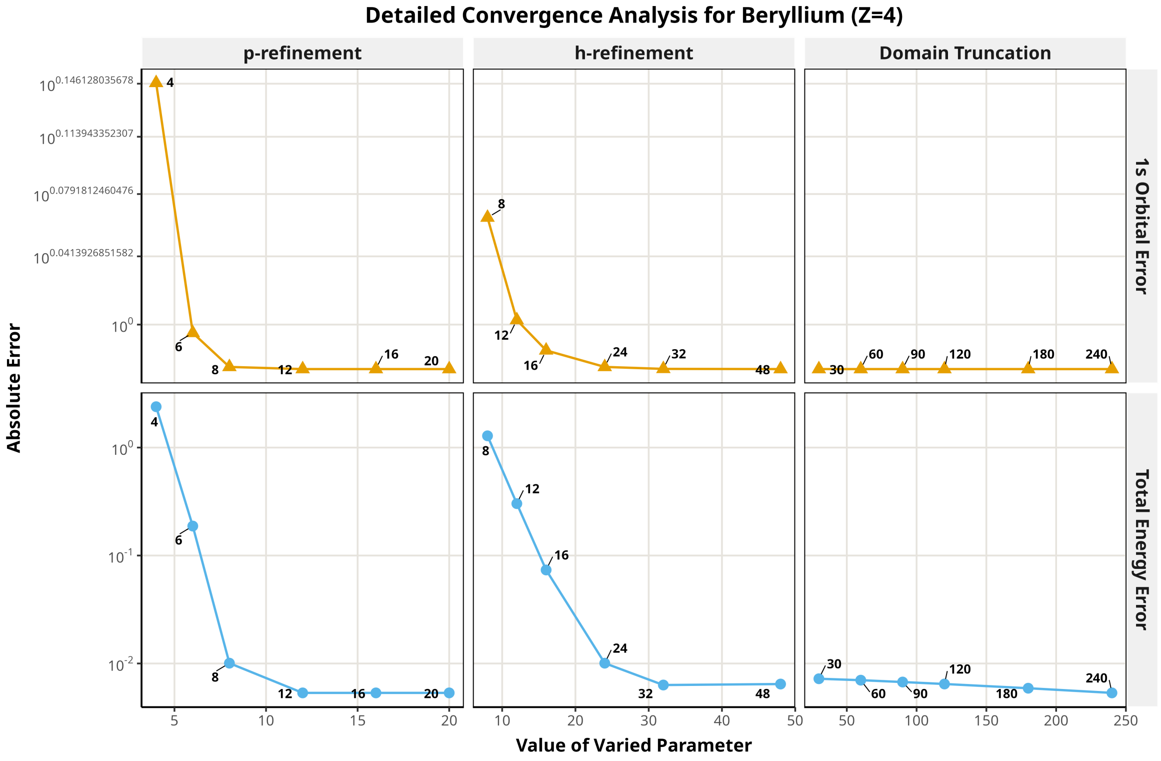


Figure 3.1. *Systematic convergence of the radial finite element solver for a Restricted Hartree-Fock calculation on a Beryllium atom (Z=4). The grid validates the two primary modes of convergence.* ***Bottom Row (Total Energy)*** *and* ***Top Row (1s Orbital)*** *show the error for three distinct refinement studies. (**Left Column**) p-refinement: For a fixed mesh, the error decreases exponentially with increasing polynomial order (p), demonstrating rapid convergence to high accuracy. (**Middle Column**) h-refinement: For a fixed polynomial order, the error decreases more slowly (algebraically) with the number of elements (N_e). (**Right Column**) Domain Truncation: The solution is stable and well-converged with respect to the domain cutoff (r_{max}).*

3.5 Performance and accuracy

Figure 3.2 demonstrates the systematic, reproducible convergence of the solver for uranium in three complementary regimes: domain truncation, \$p\$- and \$h\$-refinement, and the achieved energy precision.

As designed, the `featom` code is tailored towards being state-of-the-art as a DFT solver for relativistic calculations, which are otherwise treated only through shooting approaches [74], often in the context of expensive quantum chemical finite difference based [97]. Despite this, for systems with a high number of states, `featom` outperforms shooting method state of the art calculators as well. Here, we appeal to the transitive nature of benchmarks since the state of the art `dftatom` compares favorably to Desclaux [98], `atompp` [99], `grasp2k` [97], MCHF [100], `atompaw` [101], `PEtot` [102] and Elk [103], we assert superiority by directly comparing against `dftatom`. DFT based formulations are always faster, due to the ability to skip the two particle exchange integral. For instance, a B-spline based Hartree-Fock solver [104] reports, for a toy 3-electron model of Uranium ("Lithium like", `2s(1)`), a time of 1.53

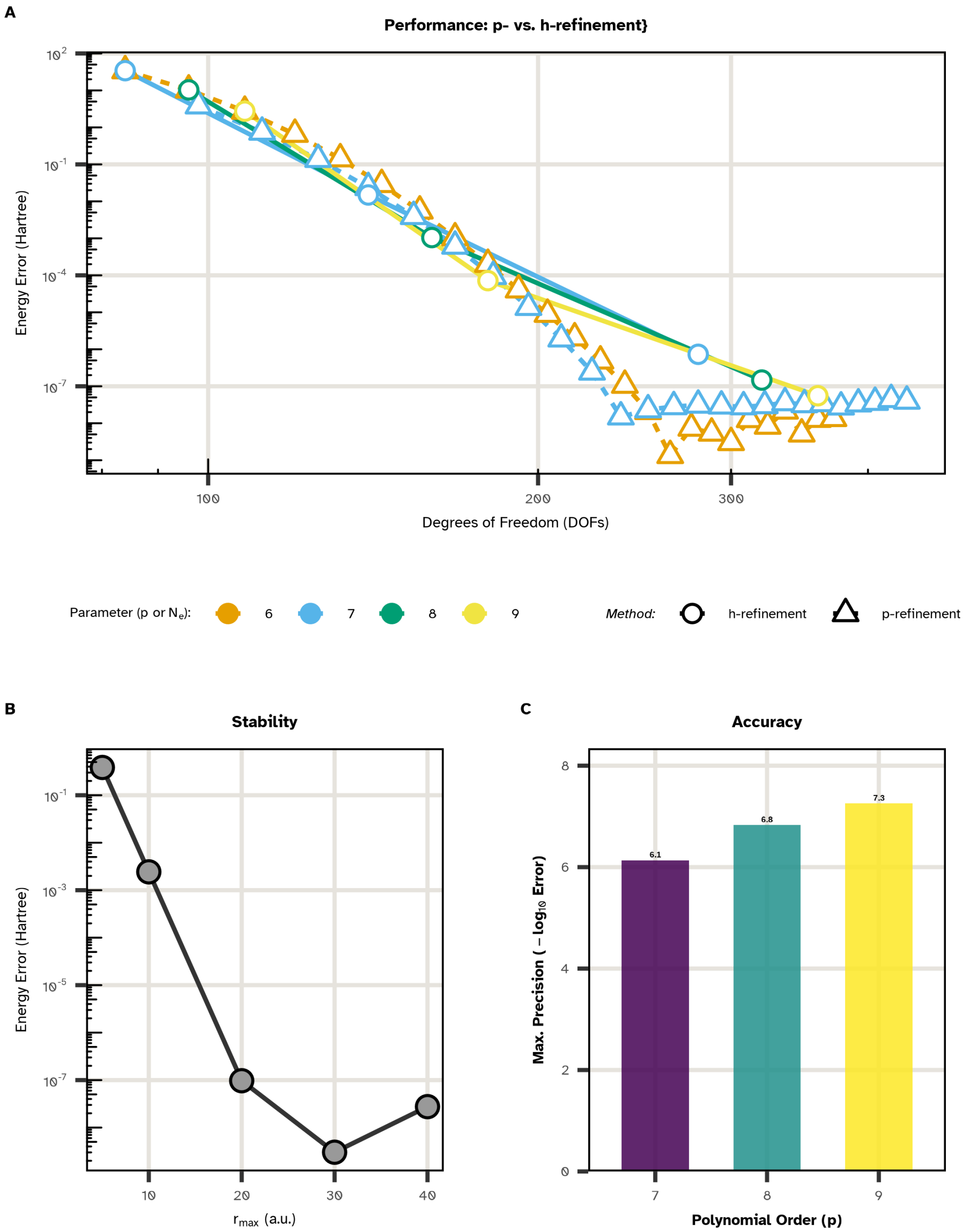


Figure 3.2. *Systematic convergence and precision of the* `featom` *finite element solver for relativistic Dirac–Kohn–Sham calculations of uranium (Z=92). (a) p- vs. h-refinement: Both p-refinement (increasing polynomial order p, colored) and h-refinement (increasing number of elements N_e) yield systematic error reductions. The plot shows energy error as a function of the total degrees of freedom (DOFs), with shape and color encoding the refinement parameter and method; exponential convergence in p and algebraic in N_e are both evident. (b) Domain cutoff stability: The total energy error decreases rapidly as the radial domain boundary r_{max} increases and quickly plateaus, demonstrating insensitivity to the outer cutoff. (c) Accuracy: Bar plot of the maximum precision (number of correct digits, -$\log_{10}$(error)) reached for each p value, highlighting the accuracy attainable with moderate p. Collectively, these results establish* `featom` *as a robust, high-precision, and reproducible tool for atomic DFT, confirming correct asymptotic error behavior for both p- and h-refinement, as well as stability against domain truncation.*

seconds without the Breit interaction while we demonstrate a full 29 state Uranium atom:

$$1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^2 4p^6 4d^{10} 4f^{14} 5s^2 5p^6 5d^{10} 5f^3 6s^2 6p^6 6d^1 7s^2.$$

solution in 0.36 seconds. Furthermore, we compare the wall time against the state-of-the-art shooting-method code `dftatom`. For this DFT calculation of uranium converged to an accuracy of 10^{-6} Hartree, `featom` shows a significant speedup for non-relativistic calculations and competitive performance for relativistic ones, validating the effectiveness of the chosen representations.

Table 3.1. *Timings for a DFT calculation of a uranium atom on an Apple M-1 Max processor.*

Solver	`featom`	`dftatom`
Schrodinger	28 ms	166 ms
Dirac	360 ms	276 ms

The benchmark results in Table 3.1 reflect the distinct scaling characteristics of the FEM used in `featom` versus the shooting method used in `dftatom`.

For the Schrödinger equation, `featom` is significantly faster (28 ms vs 166 ms) because it solves for all eigenvalues in a given angular momentum channel simultaneously via a single diagonalization of a small matrix ($N_b \times N_b$). In contrast, `dftatom` must perform iterative radial integrations for each individual electronic state, which accumulates a higher computational cost for high-Z atoms with many electrons.

For the Dirac equation, the performance shifts. In the FEM framework (`featom`), the basis set size effectively doubles to represent both the large and small spinor components, resulting in matrices of size $2N_b \times 2N_b$. Since the cost of the dense eigensolver scales cubically with matrix size ($O(M^3)$), the computational effort increases by a factor of roughly 8 (2^3), alongside increased matrix assembly costs. In contrast, the shooting method (`dftatom`) only sees a linear increase in cost associated with integrating two coupled first-order ODEs rather than one second-order ODE. Consequently, while `featom` slows down by a factor of ~ 12 (28 ms to 360 ms) due to the cubic scaling penalty, `dftatom` slows down by only ~ 1.6 (166 ms to 276 ms), making it slightly faster in the fully relativistic case for this specific benchmark. Despite the associated overhead, the optimized software ensures `featom` remains competitive in absolute time compared to `dftatom` and outstrips other solvers.

3.6 Conclusions

This chapter has established the theoretical foundations—specifically the mean-field approximations of Hartree-Fock and Kohn-Sham density functional theory—requisite for interpreting electronic structure calculations throughout this work. While the specific numerical innovations presented here address the challenge of relativistic atomic precision, the broader investigations into reactive systems presented in the remainder of this thesis rely on these fundamental mean-field principles. The calculations for molecular systems in this thesis are at Hartree-Fock level, with spin-unrestricted formalisms for radical species and restricted closed-shell formalisms for stable intermediates.

Within this theoretical framework, we presented a robust finite element formulation for the solution of the radial Schrödinger, Dirac, and Kohn-Sham equations for isolated atoms. The success of the `featom` solver exemplifies the central thesis that overcoming computational barriers in physics requires a holistic approach to representation. By moving from the Dirac Hamiltonian to its squared operator, we eliminated the spectral pollution of spurious states that has historically plagued basis-set methods. By transforming the dependent variables to account for asymptotic behavior near the nucleus, we resolved the representational conflict between singular Coulombic potentials and smooth polynomial bases, recovering exponential convergence. The practical outcome of these choices is a solver that occupies a unique niche in the current software ecosystem. As demonstrated by the benchmarks, `featom` provides sub-second wall times for full relativistic DFT calculations of heavy elements like uranium ($Z = 92$), outperforming state-of-the-art shooting methods in the relativistic regime while avoiding the complexity of kinetic balance constraints required by B-spline approaches.

This efficiency is not merely a convenience but should translate eventually towards generating better basis sets for high-fidelity reference data needed to train Gaussian Process surrogates discussed in the previous chapter.

Finally, the implementation of `featom` underscores the critical importance of software accessibility and reproducibility in computational science. For instance, while reviewing the landscape of relativistic solvers, we attempted to evaluate the `BERTHA` [105] code but were unable to locate source code, finding only Python wrappers dependent on opaque, closed-source binaries[9]. In contrast, `featom` is provided as a modern, modular, open-source library. This ensures that the distinct layers of representation—mathematical, numerical, and algorithmic—remain transparent, verifiable, and adaptable.

However, the converse is also true, not every "conceptually simple" mapping (e.g. Hartree-Fock → "a potential") respects the representation. Where the physics demands an operator (HF exchange), we either keep the operator or solve a re-representation problem, e.g. through an optimized effective potential [96, 106, 107].

We will return to this concept in later chapters; when we formulate efficient reaction-path searches and have Gaussian Process accelerators succeed because their internal representations (forces, curvatures, kernels) are chosen to make the numerics stable and the computation scalable.

This serves to clarify that the "correctness" of a scientific result is inextricably linked to the design of the software that produces it. Physical rigor cannot be separated from software architecture. In the subsequent chapter, we will expand this scope from the design of a single solver to the design of the broader computational frameworks required for complex chemical exploration.

[9] to say nothing of "closed source" code results

4 Aspects of software design

> Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?
>
> Member of the House of Commons
> asked of Charles Babbage

Computational science confronts a fundamental representational problem: physical laws, typically expressed as continuous differential equations, require evaluation on digital hardware that operates with discrete logic and finite-precision arithmetic [108, 109]. The evaluation of a function $f(x)$ on a computer therefore necessitates its approximation by a discrete counterpart $\hat{f}(\mathbf{x})$ that maps a finite vector of inputs to a finite vector of outputs. This transition from the continuous domain to a discrete, floating-point representation introduces unavoidable errors, including truncation error from the discretization scheme and rounding error from the limitations of floating-point number representation [110, 90, 92, 111]. The central challenge of scientific software engineering lies not in eliminating these errors, an impossible task, but in designing computational structures that control them and guarantee the physical fidelity of the final result.

The preceding chapter on relativistic atomic calculations provided a concrete example of a successful strategy for managing this challenge. The accuracy and stability of the `featom` solver originate from a deliberate, multi-layered cascade of representations, each chosen to mitigate a specific class of error. Briefly, this involved a mathematically stable squared Hamiltonian to handle the unbounded Dirac spectrum, a functionally smooth set of corrected wave functions to accelerate polynomial convergence, a numerically robust algorithm to guarantee accurate quadrature, and finally, a modular [95] software implementation to ensure correctness and maintainability.

This chapter dissects the principles of such software redesign and the novel scientific capabilities enabled by such an undertaking. We examine how conscious architectural choices directly impact the quality and reliability of scientific outcomes, focusing on the implementation of novel scientific algorithms, such as the hybrid MMF-NEB method, which were made possible only after a fundamental re-engineering of the software's state management and potential interfaces; the choice between geometric and electronic-structure representations for defining a chemical bond; the interpretation of double ended saddle point data; the representation of a complex scientific protocol as a formal DAG using a workflow engine; along with future directions. In each case, the software design reflects a conscious strategy to build powerful and reliable computational models upon the discrete and finite foundation of the computer.

4.1 Bonding analysis

To unambiguously distinguish covalently bonded molecular fragments from transient non-covalent contacts, a robust analysis of the system's bonding network is essential. This can be approached from two distinct perspectives, namely a heuristic geometric definition or a more rigorous definition based on the system's electronic structure. We implement in `rgpycrumbs` both methods, with a `pyvista` backend [10] allowing for a flexible and chemically aware analysis of molecular connectivity.

4.1.1 Geometric method: Covalent cutoff

The simplest and most computationally efficient method for defining a bonding network is based on geometry. In this approach, we define that a bond exists between two atoms, i and j, based on their interatomic distance, d_{ij}. Specicially, the bond exists when atoms are closer to each other than a scaled sum of their tabulated covalent radii, r_i and r_j. This relationship is governed by the inequality:

$$d_{ij} < M \cdot (r_i + r_j) \tag{46}$$

Here, M is a dimensionless scaling multiplier [11] used to adjust the strictness of the criterion. While this method is extremely fast, and widely available due to being the method used by the ASE [112] graphical user interface (GUI); a purely geometric measure for molecules is a significant drawback, as the base unit of calculations are centered on electrons (Chapter 3). Such measures lack "chemical intuition" and can fail in sterically crowded environments where non-bonded atoms are forced into close proximity, leading to the false identification of covalent bonds (as illustrated in Figure 4.1 B and C.

4.1.2 Electronic density: Wiberg bond order

A more physically meaningful approach defines connectivity based on the electronic structure of the system, specifically; from the density matrix obtained in a quantum chemical calculation. A simple form of this is Wiberg bond order (WBO), which represents the electron density shared between two atoms, A and B, by the sum of squares of the density matrix elements corresponding to the atomic orbitals on each atom [113]. We define:

$$\mathrm{WBO}_{AB} = \sum_{\mu \in A} \sum_{\nu \in B} (P_{\mu\nu})^2 \tag{47}$$

where

- WBO_{AB} is the Wiberg Bond Order between atom A and atom B.
- $\sum_{\mu \in A}$ sums over all atomic orbitals μ on atom A.
- $\sum_{\nu \in B}$ sums over all atomic orbitals ν on atom B.
- $P_{\mu\nu}$ is an element of the density matrix.

[10] inspired by `solvis`

[11] typically between 1.1 and 1.3

The density matrix element $P_{\mu\nu}$ for a closed-shell system is calculated from the molecular orbital coefficients (C) of the occupied molecular orbitals (i):

$$P_{\mu\nu} = 2 \sum_{i}^{\text{occupied}} C_{\mu i} C_{\nu i} \tag{48}$$

The WBO correlates well with the intuitive chemical concept of single, double, and triple bonds. A bond between atoms i and j is defined to exist only if their calculated bond order, WBO_{ij}, exceeds a predefined threshold, T_{bond}:

$$\text{WBO}_{ij} > T_{\text{bond}} \tag{49}$$

Such a measure can be significantly more robust than the geometric approach as it hinges on the calculation of the actual chemical interactions. It can reliably distinguish between genuine covalent bonds, which have significant shared electron density (typically WBO > 0.7), and close non-covalent contacts, which exhibit negligible bond orders. As shown in Figure 4.1, the WBO between sterically close but non-bonded atoms evaluates to a near zero value, correctly identifying them as belonging to separate molecular fragments. The primary trade-off for this increased accuracy is the higher computational cost associated with performing the underlying electronic structure calculation, which is largely alleviated by using GFN2-xTB semi-empirical calculation [114] [12].

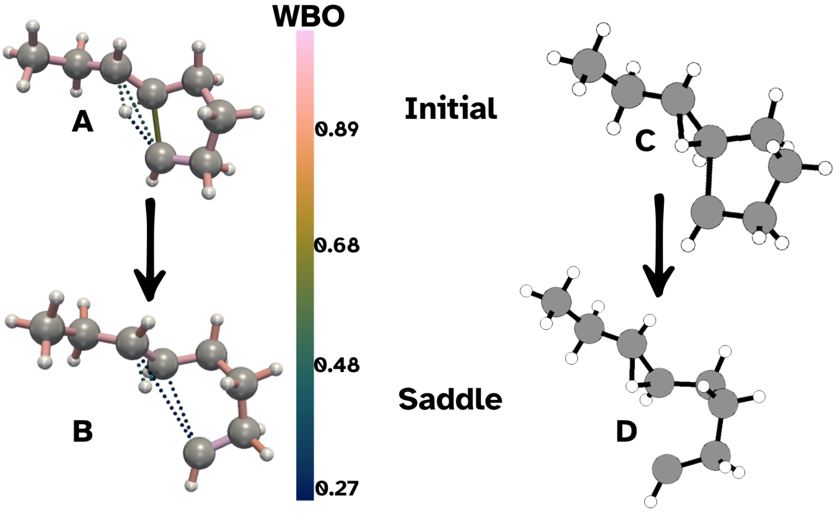


Figure 4.1. *Wiberg Bond Order (WBO) analysis of a radical hydrogen transfer reaction (doublet system D004) from an initial reactant complex to the saddle point. Panels (a) and (b) visualize the system with interatomic connections colored by their WBO, where bonds are above 0.5, revealing the subtle electronic changes during the reaction: the weak C-C bond with a WBO of ~0.5 in the initial state (a) is broken (b). In contrast, the standard geometric stick representation from ASE in panels (c) and (d) shows a nonsensical three center bond involving hydrogen, which is geometrically close but not actively bonded.*

[12] this is still not quick enough for extended systems however.

4.2 RMSD projections for path visualization

Methods like the NEB form chains-of-states pathways to connect configurations on the Potential Energy Surface, and visualzation of these is provided within `rgpycrumbs` [13]. Within EON [29] profiles of the extrema are written out in quadruplets of the terms of the image number, the energy difference relative to the "reactant", "path" reaction coordinate and parallel force. We define the "path" reaction coordinate (s), to be the cumulative Cartesian distance between successive images $\mathbf{R}_i$ along the path, or

$$s_i = \sum_{j=1}^{i} \left\| \mathbf{R}_j - \mathbf{R}_{j-1} \right\|_2 \tag{50}$$

where $\mathbf{R}_j$ are atomic positions for the j-th image and $s_0 = 0$. The energy difference against the path coordinate for the converged path is the most common visualization with insets indicating the climbing image and end-points. To create a smooth curve from the discrete images, we use a Cubic Hermite Spline [92, 115], as is used internally in EON as well. Unlike a standard spline, this constructs a piecewise cubic polynomial $H(s)$ that matches the energy E_i at each image i but also the projected derivative E_i' which we define to be the negative of the force component parallel to the path tangent $\hat{\tau}_i$:

$$\left.\frac{dE}{ds}\right|_{s_i} = -\mathbf{F}(\mathbf{R}_i) \cdot \hat{\tau}_i = -F_{\|,i} \tag{51}$$

where $\hat{\tau}_i$ is the normalized tangent for image i. Hermite interpolation ensures that both the energy and the slope are matched at each discrete point, yielding a consistent and smooth profile as used internally, that preserves barrier heights and avoids artifacts. In this thesis, we use the augmented form of this visualization with the history of the path optimization [33, 116], as shown later, in Figure 4.2.

For higher-dimensional visualization, we project the NEB path onto the plane defined by RMSD from reactant and RMSD from product:

$$(x_i, y_i) = \big(d_{\text{RMSD}}(\mathbf{R}_i, \mathbf{R}_{\text{reactant}}), d_{\text{RMSD}}(\mathbf{R}_i, \mathbf{R}_{\text{product}})\big) \tag{52}$$

To resolve atom mapping and orientation ambiguities, particularly in symmetric systems, we use the iterative rotations and assignments (IRA) Fortran routine from Python [117]. This finds the optimal atom permutation, rotation, and translation to minimize RMSD:

$$d_{\text{RMSD}}(\mathbf{A}, \mathbf{B}) = \min_{\mathbf{R},\mathbf{P}} \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left\| \mathbf{R}\mathbf{a}_{\mathbf{P}(j)} - \mathbf{b}_j \right\|^2} \tag{53}$$

where $\mathbf{P}$ is the atom permutation and $\mathbf{R}$ is the rotation matrix.

To visualize the local structure of the potential energy surface, we interpolate the scattered energies onto the RMSD plane. This is achieved using a Radial Basis Function interpolator with a Thin Plate Spline kernel [118], as implemented in SciPy [115].

[13] A pure Python library for snippets, here `https://github.com/HaoZeke/rgpycrumbs/`

$$f(\mathbf{x}) = \sum_{i=1}^{N} w_i \phi(||\mathbf{x} - \mathbf{x}_i||) + P(\mathbf{x}) \tag{54}$$

where $\phi(r) = r^2 \log(r)$ is the thin plate spline radial function and $P(\mathbf{x})$ is a low-degree polynomial term included to ensure solvability. The weights w_i are determined by solving a linear system that interpolates the energy values E_i at the sampled coordinates (x_i, y_i). A smoothing parameter λ (set to 0.009 in this work) is applied to the diagonal of the interpolation matrix to handle noise in the optimization data and prevent overfitting to high-energy artifacts. This produces the smooth, physically continuous contour maps used to visualize the topography of the energy landscape, shown in section 4.3.3, Figure 4.3.

The combination of Hermite-spline profile interpolation and two-dimensional landscape projection provides mechanistic insights into both the energetic barriers and the geometric progression of the reaction. The 1D profile quantifies how forces and energy change along the path, while the 2D landscape exposes the multidimensional structure of the transition region. Together, these methods allow us to visualize not only the minimum energy pathway but also the broader context of atomic rearrangements and surface topography that govern chemical transformations.

4.3 EON

Most of the calculations in this thesis go through EON. Rather than exessively modifying the SVN copy, a new release was drafted, v2.8.0 [14]. All the methods presented in this thesis are either the direct result of, or stemmed from the landmark modernization of the EON client code. This multi-year development effort, spawning millions of lines of code and documentation, overhauled the entire framework to be more powerful, flexible, and robust, transforming it from a legacy tool into a modern scientific platform. The effort focused on several key areas of software engineering.

First, the core C++ backend was modernized to the C++17 standard, and adopting modern STL libraries like `<filesystem>` for cross-platform compatibility. Second, the entire build process was migrated from legacy Makefiles to the Meson build system, and a comprehensive continuous integration (CI) pipeline was established to test automatically across Linux, Windows, and macOS. This professionalized infrastructure guarantees correctness, portability, and long-term maintainability.

Crucially, this architectural refactoring enabled a fundamental shift in the software's capabilities. The redesigned state management and potential interfaces made it possible, for the first time, to instantiate and control multiple, different potential energy surface evaluators within a single simulation. This unlocked a vast expansion of interoperability, with new interfaces to a dozen external quantum chemistry and machine learning codes (NWChem [119], ORCA [120], XTB [114], ASE [112], PET-MAD [116, 63]). More importantly, this architectural flexibility provided a platform for inventing arbitrarily novel and efficient hybrid algorithms. The Hybrid MMF-NEB method, detailed next, is a direct product of this new design, as it leverages the

[14] accompanying documentation : `https://eondocs.org`

ability to combine different optimizers and energy-weighted spring forces within a single, cohesive calculation—a capability that was previously impossible.

4.3.1 Eliminating I/O Bottlenecks with a Client-Server Architecture

A primary performance bottleneck in complex simulation workflows is the reliance on file-based I/O to communicate with external potential energy surface calculators. This traditional approach—repeatedly writing input files, executing an external program as a separate process, and parsing text output files—suffers from immense overhead from disk access and process creation, rendering many computationally demanding methods infeasible, especially those which use large wavefunctions.

To overcome this limitation, a significant part of the EON refactoring involved implementing a high-performance, in-memory communication layer based on the i-PI [121] client-server protocol. In this modern architecture, EON acts as a persistent server that orchestrates the simulation, while a quantum chemistry code like NWChem runs as a long-lived client. The communication of atomic coordinates and the resulting energies and forces occurs directly through low-latency TCP/IP or high-performance UNIX domain sockets. This transforms the external potential from a slow, "black box" command-line tool into a responsive, integrated library.

This effort required not only developing the server architecture within EON [15] but also contributing directly to the NWChem codebase to improve its capabilities as a client. A key pull request, which was merged into the official NWChem repository [16] , enhanced its socket client with a robust polling and retry mechanism. This modification allows the NWChem client to wait patiently for the EON server to become available, a critical feature for ensuring stable, loosely-coupled communication between the two persistent programs.

The performance impact of this architectural shift is dramatic, as shown in the benchmark timings for a 16-step minimization.

Table 4.1. *Performance comparison of communication methods for an identical 16-step minimization task. Wall times were measured on a ThinkPad X1.*

Communication Method	Wall Time	Speedup vs. File-based
File-based (ASE Wrapper)	78 s	1.0x
TCP/IP Socket	47 s	1.7x
UNIX Domain Socket	40 s	2.0x
UNIX Socket (Patched NWChem)	17 s	4.6x

The socket-based communication layer provides a 2× speedup out of the box, and a remarkable 4.6× speedup when combined with a fully optimized NWChem build. This architectural change from a file-based to a socket-based representation of the potential energy surface is another enabler for the wall time efficient methods explored in this thesis.

[15] gh-244 to EON

[16] gh-1145 to NWChem

4.3.2 Hybrid Climbing Image NEB with Minimum Mode Following (CI-NEB-MMF)

While the standard Climbing Image NEB (CI-NEB) method is effective at converging to a saddle point, the final stages of relaxation for the climbing image can be slow, particularly on flat potential energy surfaces. To accelerate this final convergence, a hybrid approach has been implemented that integrates a dedicated minimum-mode following (MMF) saddle search directly into the NEB optimization cycle. This method can be seen as a two-stage refinement strategy within each NEB iteration once the path is sufficiently relaxed [17].

The core idea is to use the robust path-finding capability of NEB to bring the climbing image close to the saddle point, and then switch to a more aggressive and efficient local saddle search algorithm for a few steps to rapidly refine the climbing image's position.

The methodology is controlled by several key parameters: a boolean switch to enable the feature (`nebciWithMMF`), a force threshold for activation (`nebciMMFAfter`), and the number of MMF steps to perform per NEB iteration (`nebciMMFnSteps`).

The force applied during the local refinement phase is the standard MMF force, which inverts the true force component along the lowest-energy mode (approximated by the NEB tangent $\hat{\tau}_{\text{climb}}$):

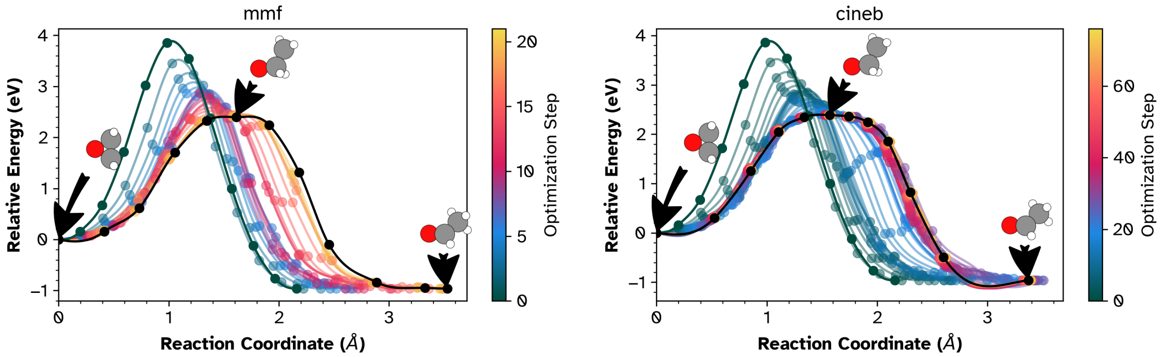
$$\mathbf{F}_{\text{MMF}}(\mathbf{R}_{\text{climb}}) = \mathbf{F}(\mathbf{R}_{\text{climb}}) - 2(\mathbf{F}(\mathbf{R}_{\text{climb}}) \cdot \hat{\tau}_{\text{climb}})\hat{\tau}_{\text{climb}} \tag{55}$$

This force is identical to the one used in the standard CI-NEB (Equation 16), but its application within a dedicated saddle search optimizer (such as one based on the Dimer method) allows for more efficient convergence on the saddle point. The overall process is outlined in Algorithm 2.

Algorithm 2 Hybrid CI-NEB with Minimum Mode Following (CI-NEB-MMF)

```
Initialize NEB path $\{\mathbf{R}_0, \ldots, \mathbf{R}_P\}$
while not converged do
    Find highest energy image, $\mathbf{R}_{\text{climb}}$
    Calculate tangents $\hat{\tau}_i$ for all images
    Calculate NEB forces $\mathbf{F}_i^{\text{NEB}}$ for all non-climbing images using Eq. 11
    Calculate force for climbing image $\mathbf{F}_{\text{climb}}$ using Eq. 16
    $F_{\max} \leftarrow \max_i(|\mathbf{F}_i^{\text{NEB}}|)$
    if $F_{\max} < F_{\text{MMF\_threshold}}$ and MMF enabled then
                                ▷ Switch to local MMF refinement for the climbing image
        Create a temporary MMF optimizer for $\mathbf{R}_{\text{climb}}$
        for $k = 1$ to $N_{\text{MMF\_steps}}$ do
            Update $\mathbf{R}_{\text{climb}}$ using a step from the MMF optimizer with $\mathbf{F}_{\text{MMF}}$ (Eq. 55)
        end for
        Update the full path's forces after MMF refinement
        Take a global optimization step on all images with their respective forces
    else
                                ▷ Perform standard NEB optimization step
        Take a global optimization step on all images
    end if
end while
```

[17] gh-230 to EON



***(a)** Hybrid CI-NEB-MMF*

***(b)** Standard CI-NEB*

Figure 4.2. *Comparison of the optimization process for the ethylene oxide to acetaldehyde isomerization using (a) the hybrid CI-NEB-MMF and (b) the standard CI-NEB methods. Each colored line represents the reaction path at a specific point in the optimization, progressing towards the final, converged path (based on the climbing image) shown in black. The reaction coordinate on the x-axis is defined as the cumulative Cartesian distance (Å) between successive images along the path. While both methods find the identical transition state estimate, the color bars highlight the significantly greater efficiency of the hybrid method, which converges in approximately 20 steps, whereas the standard method requires over 70 steps.*

4.3.3 Case Study: Isomerization of Ethylene Oxide to Acetaldehyde

To demonstrate the effectiveness and computational efficiency of the hybrid CI-NEB-MMF method, it was applied to the isomerization reaction of ethylene oxide to acetaldehyde, the results of which are shown in Figure 4.2.

The reaction involves the rearrangement of ethylene oxide, a three-membered cyclic ether (epoxide), into its more stable isomer, acetaldehyde. The primary thermodynamic driving force for this exothermic reaction is the release of significant ring strain present in the ethylene oxide molecule. The C-C-O bond angles in the epoxide ring are constrained to approximately 60 deg, a severe deviation from the ideal 109.5 deg for \$sp^3\$-hybridized atoms. This strain makes ethylene oxide a high-energy, reactive species. The rearrangement allows the ring to open, forming the more stable carbonyl and methyl groups of acetaldehyde and releasing the stored strain energy.

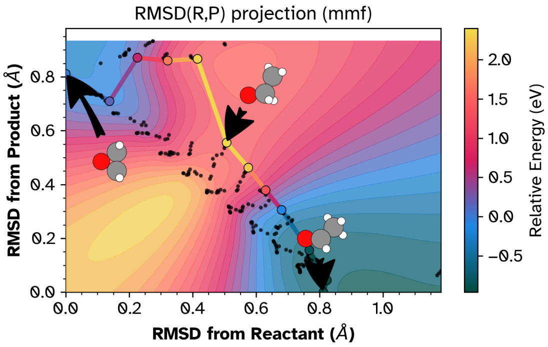


Figure 4.3. *2D landscape projection of the converged hybrid CI-NEB-MMF path for the ethoxy acetal system. The trajectory is plotted on a coordinate system of RMSD from the reactant vs. RMSD from the product. Black points represent the discrete sampling history during optimization. The interpolated energy*

While the 1D projection in Figure 4.2 is essential for comparing optimization efficiency and visualizing the energy barrier, it fundamentally compresses the multidimensional pathway into a single coordinate. We can visualize the geometric progression of the reaction and the topographic context of the Potential Energy Surface with the landscape projection method developed in section 4.2, shown in Figure 4.3. This projects the converged path onto a coordinate system defined by the RMSD of the endpoints. The trajectory traces the channel from the reactant basin (top-left, $RMSD_P \approx 0.8$ Å) to the product basin (bottom-right, $RMSD_R \approx 0.8$ Å).

The path is overlaid on an energy surface interpolated from the discrete image data (scattered black points). This "top-down" view provides clear mechanistic insights detailed in Goswami [122]. The reactant valley (blue, top-left) represents the metastable basin corresponding to the strained ethylene oxide molecule in the reactant state, at a relative energy of approximately 0.0 eV. The path follows a ridge-line to cross the transition state—the highest energy point on the path, indicated in bright yellow/orange—located at an RMSD of ≈ 0.50 Å from the reactant and ≈ 0.55 Å from the product. This saddle point corresponds to a barrier exceeding 2.0 eV. Following the transition, the path descends into the product valley (dark teal, bottom-right), which forms a broad, deep basin with a relative energy below -0.5 eV. This visual confirmation of the product's stability relative to the reactant aligns with the thermodynamic release of ring strain expected in the ring-open acetaldehyde species.

The reaction was modeled using the PET-MAD [63] machine learning potential (v1.1.0) through the novel Metatomic interface [116] implemented in EON. The entire process, from initial path generation using the IDPP [32] method to the final NEB calculations, was automated using a Snakemake workflow [123]. Both the standard and hybrid NEB calculations started from identical, pre-minimized endpoints and an IDPP-generated [32] initial path from ASE [112]. An L-BFGS optimizer was used to relax the path until the maximum force on any image fell below the convergence criterion of 0.01 eV/Å.

In the standard CI-NEB calculation, the climbing image was activated once the maximum force fell below 0.5 eV/Å. In the hybrid CI-NEB-MMF calculation, the climbing image was activated earlier at 1.5 eV/Å, while the local minimum-mode following refinement was triggered only after the force reached 0.5 eV/Å. Up to 20 minimum mode following (MMF) steps were applied to the climbing image during each subsequent NEB iteration.

Table 4.2. *Performance comparison for the standard and hybrid NEB methods.*

Method	NEB Steps	PES Calls
Standard CI-NEB	77	1564
Hybrid CI-NEB-MMF	22	582

The hybrid approach required approximately 63% fewer Potential Energy Surface calls and converged in 71% fewer NEB steps. This efficiency gain stems from the MMF method's ability to rapidly converge the climbing image to the saddle point once the NEB path is in the correct region, avoiding the slow relaxation characteristic of standard CI-NEB on flat or nearly flat potential energy surfaces.

4.4 Workflow engines

Modern computational science relies on complex, multi-step procedures that can be difficult to manage, reproduce, and scale. Workflow engines are software tools designed to address this challenge by providing a framework to define, execute, and automate these computational pipelines. For this work, the Snakemake workflow management system was used. Snakemake utilizes a Python-based, human-readable syntax to define a series of rules in a file known as a `Snakefile`. These rules, along with their specified input and output dependencies, implicitly form a DAG, which Snakemake automatically resolves to determine the correct order of execution for all required tasks.

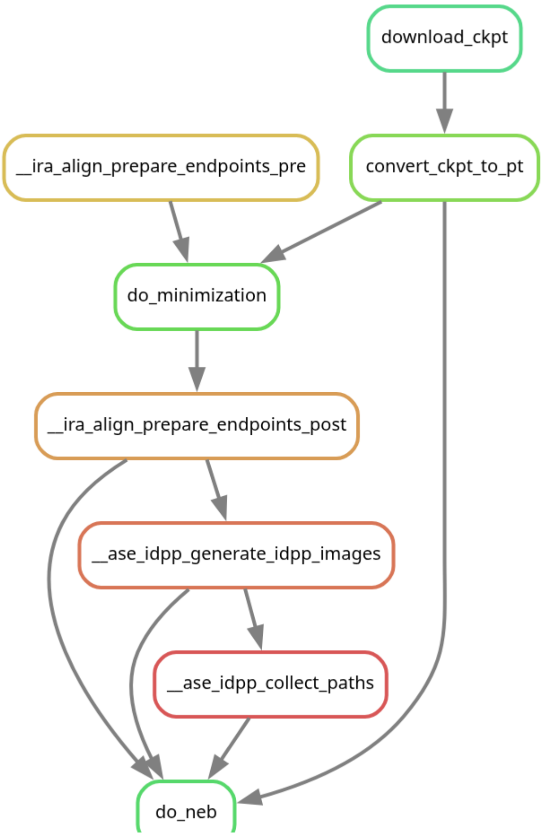

download_ckpt
__ira_align_prepare_endpoints_pre
convert_ckpt_to_pt
do_minimization
__ira_align_prepare_endpoints_post
__ase_idpp_generate_idpp_images
__ase_idpp_collect_paths
do_neb

Figure 4.4. *DAG ensures critical pre-processing steps, e.g. endpoint minimization and initial path generation, are systematically executed before the main NEB.*

A key advantage of Snakemake in a research environment is its seamless integration with high performance computing (HPC) resources. It abstracts the underlying job scheduler (e.g., Slurm, PBS, SGE), allowing the same workflow definition to be executed on a local machine for testing or scaled up to a large cluster for production runs. This portability ensures that the computational environment can be easily adapted without altering the scientific logic of the workflow itself.

Another critical function of a workflow engine in a scientific context is the ability to programmatically encode best practices and enforce reproducibility. Many computational methods, such as the NEB technique, require a specific sequence of preparatory and execution steps for reliable results. Manually performing this sequence can be tedious and prone to human error, such as forgetting a critical step or using mismatched model versions. By defining the entire protocol as a series of dependent rules, Snakemake transforms a manual checklist into a robust, automated, and self-documenting scientific component.

For example, a best-practice NEB workflow, as illustrated in Figure 4.4, involves several distinct stages: first, ensuring the reactant and product structures are fully minimized and consistently aligned; second, generating a sensible initial path between these mapped endpoints; and only then, executing the main NEB optimization. Additionally, the workflow enforces reproducibility by automatically fetching and converting the specific version of the machine learning potential required for the simulation. By encoding this logic in a `Snakefile`, one can guarantee that the initial path is never generated with unrelaxed or unaligned endpoints, thus preventing erroneous calculations and embedding expert knowledge directly into the computational tool. This approach ensures that every calculation is performed consistently and correctly, forming the foundation of truly reproducible research.

4.5 Towards maximal concurrency

The architectural modernizations detailed in the preceding sections represent crucial steps away from a traditional, monolithic software design paradigm and towards a more flexible and powerful future. The dominant model in High-Performance Computing has historically relied on large, statically linked executables communicating via the Message Passing Interface (MPI). While effective for tightly-coupled, homogeneous tasks, this approach has significant drawbacks: component libraries must be chosen at compile time, leading to massive, inflexible binaries; the static allocation of resources can be inefficient; and the model is fragile, as a fault in any single component can terminate the entire multi-node calculation.

Legacy scientific codes, often developed over decades, typically feature tightly-coupled components that communicate through global state. A canonical example of this design is the Runtime Database (RTDB) in NWChem. The RTDB functions as a centralized, string-keyed, key-value store—a clever design for its time to decouple modules from the input file, but one that comes at the cost of type safety and creates a strong dependency on a single, shared resource that complicates external interoperability. This monolithic model hinders rapid prototyping, the integration of new tools, and the creation of flexible, polyglot workflows, conceptually shown in Figure 4.5.

A more robust and flexible paradigm, inspired by modern distributed systems, recasts scientific workflows as a collection of smaller, decoupled services that communicate "on the wire.", e.g. through ZeroMQ. Realizing this vision, however, presents two orthogonal philosophies for achieving high-performance interoperability.

The first philosophy relies on a common shared library and a standardized C Application Binary Interface (ABI). The Metatensor library exemplifies this approach [116]. It defines a language-agnostic C ABI and a strict in-memory data layout for its core tensor structures. This allows a library written in Rust to operate on the exact same memory buffer created by a Python script without any serialization or data copying, achieving true zero-copy performance. The trade-off is a strong dependency at the build and link stages: all components must compile and link against the same version of the shared library, which can lead to a cascade of bindings (e.g., from C++ to a C-API, then to Python) to achieve interoperability.

Part of this work demonstrates an initial step in this direction with the client-server architecture for potential energy calculations, and parameters within the Gaussian Process and EON. The logical extension of this concept is a system of interchangeable, polyglot libraries communicating through RPC. In this model, a Python-based workflow engine could orchestrate a simulation by sending requests to a high-performance Fortran optimizer, which in turn queries a potential energy surface provided by a C++ machine learning library, with each component running concurrently on the most appropriate hardware.

This schema acts as an unambiguous contract that completely decouples the client and server. The Cap'n Proto compiler auto-generates the necessary code, enabling a C++ server to communicate seamlessly with a Python client, even across a network. This design provides strong type safety, a stark contrast to the RTDB's untyped lookups, and remarkable flexibility.

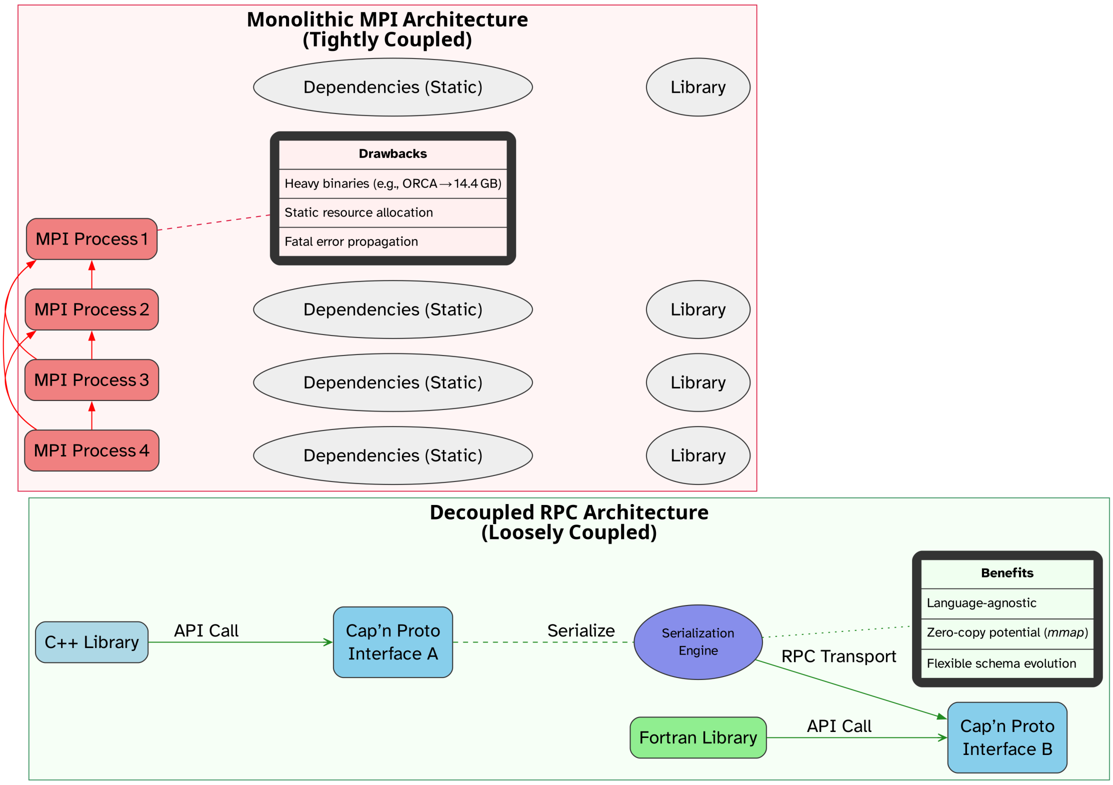


Figure 4.5. *A schematic comparison of two software architecture paradigms in scientific computing. (Left) The traditional monolithic model, based on MPI, statically links all dependencies into large, identical processes. This tight coupling results in heavy binaries, static resource allocation, and system-wide fragility where an error in one process can be fatal to the entire calculation. (Right) The modern decoupled model separates components into independent services (e.g., C++ and Fortran libraries) that communicate through a well-defined, language-agnostic RPC interface. This loose coupling enables interoperability, modularity, and flexibility, allowing components to be developed and deployed independently.*

Listing 1 A Cap'n Proto schema defines a strict, language-agnostic contract for RPC, based on `potlib`

```
@0xbd1f89fa17369103;

struct ForceInput {
  natm   @0 :Int32;
  pos    @1 :List(Float64);
  atmnrs @2 :List(Int32);
  box    @3 :List(Float64);
}

struct PotentialResult {
  energy @0: Float64;
  forces @1: List(Float64);
}

interface Potential {
  calculate @0 (fip :ForceInput) -> (result :PotentialResult);
}
```

Because the schema can evolve, a client can ignore new fields it does not understand, allowing for independent updates without downtime. Furthermore, for co-located processes, the serialized message can be memory-mapped (`mmap`) [124], providing a path to zero-copy communication without the rigid dependency of a shared library.

Both philosophies work towards the same grander vision: a "BLAS for computational science." Just as BLAS [125] standardized low-level linear algebra, a future ecosystem could be built upon standardized high-level interfaces for tasks like geometry optimization or kinetic Monte Carlo. The choice of implementation, a tightly-coupled C-ABI for maximum on-node performance, or a loosely-coupled RPC for maximum flexibility and distributability, or both, would become a design decision rather than a fundamental limitation. This architecture represents the future of scientific software: a federated system of specialized, best-in-class tools, seamlessly interoperable, enabling maximal concurrency and accelerating the pace of discovery.

With these concepts and pre-emptively developed tools, we can return to the problem of discovering sadde points.

4.6 Conclusions

The algorithmic and architectural developments presented in this chapter serve a specific, unifying purpose: to extend the timescale and complexity of accessible chemical transformations. While standard computational packages offer some solutions for routine equilibrium calculations, the exploration of rare events on high-dimensional potential energy surfaces imposes strict requirements on efficiency, stability, and reproducibility that monolithic software architectures cannot meet.

We have established that the “technical” details of implementation are inseparable from the validity of the physical model. The distinction between geometric and electronic criteria for bonding (Section 4.1.2) enables interpretations of saddle geometries. Similarly, the projection and interpolation methods developed for path visualization (Section 4.2, [122]) provide the necessary topological verification that the computed minimum energy pathways connect the intended basins of attraction, a non-trivial confirmation in complex molecular rearrangements.

Furthermore, the restructuring of the EON framework demonstrates that algorithmic efficiency determines scientific feasibility. We demonstrate this through the hybrid CI-NEB-MMF optimizer (Section 4.3.2), which reduces the computational cost of double ended saddle point searches by an order of magnitude [18]. Thus, modularization in code allows for the determination of transition states with machine learned interatomic potentials that are otherwise untreatable by standard CI-NEB methods due to pathologies of the landscape or the cost of the electronic structure theory. This concept leads to the more wide ranging concept of a distributed micro-service ecosystem for interopble code, a fom of “chemical BLAS (Basic linear algebra subprograms [125])”, leading to a new epoch of standardization.

Finally, the encoding of these procedures into a formal Directed Acyclic Graph via the Snakemake workflow engine (Section 4.4) moves the methodology beyond manual execution. By strictly defining the dependencies between minimization, alignment, and path optimization, we ensure that the complex simulation protocols are robust against operator error and computationally reproducible.

This infrastructure forms the necessary foundation for the work in the subsequent chapter [19]. The application of Gaussian Process Regression to saddle point searches with the GPDimer algorithm requires an underlying engine capable of rapid, asynchronous querying of the potential and robust error recovery. We demonstrated how this software environment minimizes overhead and maximizes modularity while providing diagnostic. We can now move towards addressing the theoretical challenge of constructing data-efficient surrogate models for on-the-fly exploration of the energy landscape, demonstrating state of the art saddle search methods using local Gaussian Process acceleration in a production setting.

[18] submission in progress to Frontiers chemistry.

[19] Chronologically developed alongside all chapters.

5 Efficient Gaussian Process Regression

> With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.
>
> John von Neumann

> This chapter is based on Rohit Goswami, Maxim Masterov, Satish Kamath, Alejandro Pena-Torres, and Hannes Jónsson. "Efficient Implementation of Gaussian Process Regression Accelerated Saddle Point Searches with Application to Molecular Reactions." In: *Journal of Chemical Theory and Computation* (July 2025). DOI: `10.1021/acs.jctc.5c00866`

In modern computational chemistry, the discovery process is often a fragmented and manual workflow. A researcher might use one high-performance engine to calculate energies (e.g., ORCA [120], Psi4 [126], NWChem [119]), export the results to a text file, import that data into a scripting environment (e.g., MATLAB, Python) for analysis, and finally use a specialized library (e.g., GPStuff [127]) for machine learning. This process, while functional for a single system, is untenable at scale. By establishing a clear, internal representation of the energy and force engines, and decoupling algorithms from specific data implementations, we can create a framework that replaces the brittle manual workflow with a robust and scalable platform as discussed in the previous chapter.

This chapter presents the `gpr_optim` as a concrete first step towards solving this workflow problem for scientific discovery. The core contribution of this work is afterall, not only to achieve state of the art [20] performance in terms of the number of calculations, but also to provide an architectural blueprint that demonstrates how to move away from monolithic, single-purpose applications towards an interoperable ecosystem.

5.1 Design

At its highest level, a simulation is a stateful process. The GPR model maintains the state of the learned Potential Energy Surface in terms of the internal state, which includes training data, hyperparameters, matrix decompositions; and the Dimer method maintains the state of the geometric search, i.e. the dimer position, orientation, optimization history. To orchestrate this at a per-instance scale, EON is used, as it provides generics for potentials, and the parallelism across systems is handled by Snakemake. Thus the overall framework captures this scientific endeavor with an object-oriented design.

The algorithm itself proceeds as described in Alg. 3.

[20] always a moving target

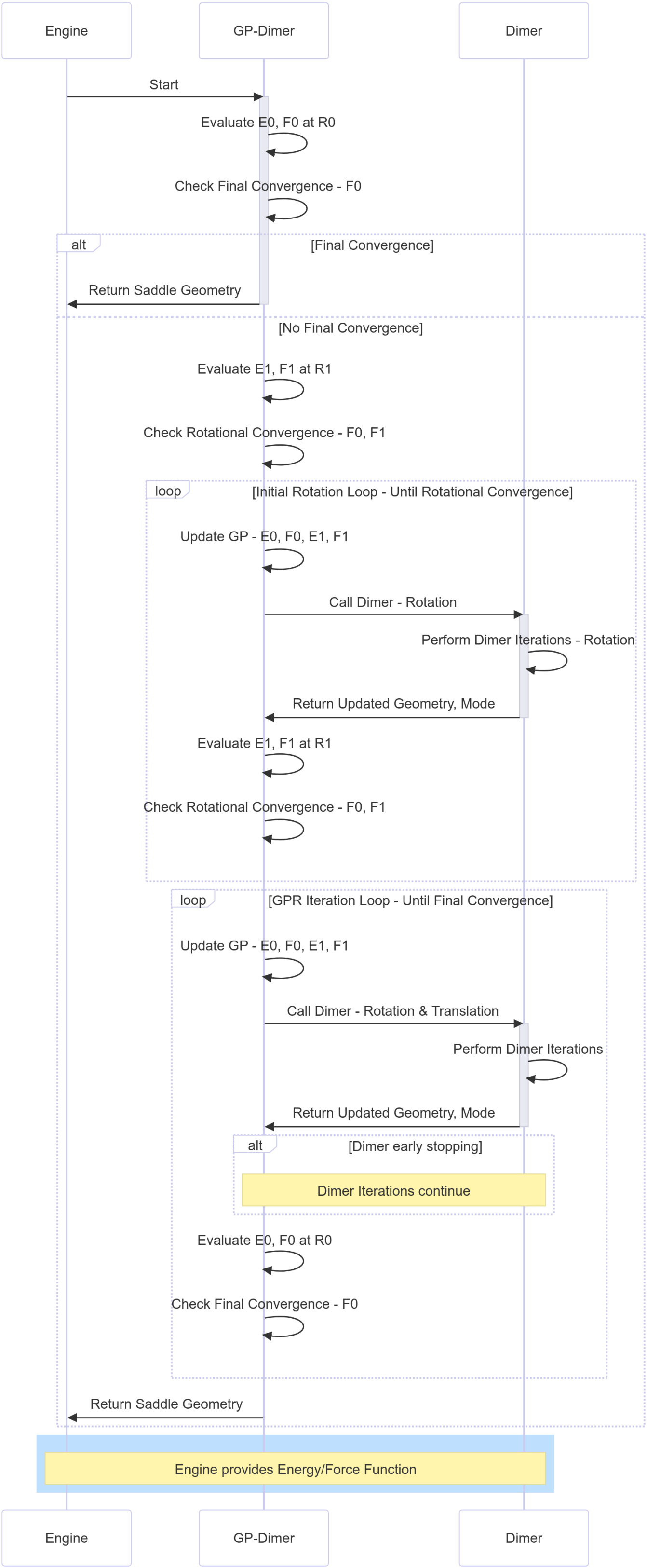

Engine
GP-Dimer
Dimer
Start
Evaluate E0, F0 at R0
Check Final Convergence - F0
alt
[Final Convergence]
Return Saddle Geometry
[No Final Convergence]
Evaluate E1, F1 at R1
Check Rotational Convergence - F0, F1
loop
[Initial Rotation Loop - Until Rotational Convergence]
Update GP - E0, F0, E1, F1
Call Dimer - Rotation
Perform Dimer Iterations - Rotation
Return Updated Geometry, Mode
Evaluate E1, F1 at R1
Check Rotational Convergence - F0, F1
loop
[GPR Iteration Loop - Until Final Convergence]
Update GP - E0, F0, E1, F1
Call Dimer - Rotation & Translation
Perform Dimer Iterations
Return Updated Geometry, Mode
alt
[Dimer early stopping]
Dimer Iterations continue
Evaluate E0, F0 at R0
Check Final Convergence - F0
Return Saddle Geometry
Engine provides Energy/Force Function
Engine
GP-Dimer
Dimer

Figure 5.1. *The GPDimer method as an entity-relation diagram showing connections to EON and the Dimer method.*

Algorithm 3 GPR Prediction (Energy and Gradient)

```
Given: Training data $\{\mathbf{X}, \mathbf{y}\}$, new configuration $\mathbf{x}^*$, covariance function $k(\mathbf{x}_i, \mathbf{x}_j; \theta)$ with hyperparameters $\theta$.
Training Phase (executed once per hyperparameter update):
Construct the training covariance matrix $K$ where $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j; \theta)$.
Add observation noise: $K_y = K + \sigma_n^2 I$.
Perform Cholesky decomposition: $L = \text{chol}(K_y)$.
Solve for $\alpha = L^T \backslash (L \backslash \mathbf{y})$.                ▷ This is the weight vector

Prediction Phase:
Compute the vector of covariances between the new point and training points: $\mathbf{k}^* = [k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_n)]^T$.
Predict mean energy: $\bar{E}^* = (\mathbf{k}^*)^T \alpha$.
Predict mean gradient: $\bar{\mathbf{g}}^* = (\nabla_{\mathbf{x}^*} \mathbf{k}^*)^T \alpha$.
return Predicted energy $\bar{E}^*$ and gradient $\bar{\mathbf{g}}^*$.
```

Hyperparameters are optimized with the scaled conjugate gradient (SCG) [128] detailed in Alg. 4. Since the energy and forces are modeled as a single output vector, the SCG is required for stability, though the code also implements a per-iteration fixed factor scaling for other optimizers like ADAM [129]. We find that the SCG is more efficient as implemented.

Algorithm 4 Scaled Conjugate Gradient (SCG)

```
Require: Initial weights $\mathbf{w}_0$, data $\mathbf{x}$, target $\mathbf{y}$, loss $f$, gradient $\nabla f$
Initialize search direction $\mathbf{p}_0 = -\nabla f(\mathbf{w}_0)$ and scaling factor $\lambda = 1$
for optimization iterations do
    Compute gradient difference $\mathbf{r}_k = \nabla f(\mathbf{w}_k) - \nabla f(\mathbf{w}_{k-1})$
    Compute curvature approximation from gradient differences
    Scale search direction and evaluate function and gradient at a trial point
    Compute step size $\alpha$ using the scaled curvature
    Update weights: $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \alpha \mathbf{p}_k$
    Check for non-finite function values and adjust step size if needed
    Check convergence criteria
    Adapt scaling factor $\lambda$
    Update search direction using Conjugate Gradients (or restart periodically)
end for
return best weights found
```

Finally the dimer itself is translated and rotated through the L-BFGS, which is indpendently implemented within the codebase. The scaling of the Gaussian Process methodology in this chapter proceeds as described in Section 2.4.1, that is, exhibiting $O(M^2N^2)$ complexity for storage and $O(M^3N^3)$ for training dominated by the inversion of the covariance matrix, whereas the inference cost scales linearly with the size of the training set M.

5.2 Surface systems

To handle extended systems, a finite cutoff is taken for determining pairs. Active pairs are updated on each new iteration. For molecular systems, the cutoff is arbitrarily large. The copper hydrogen dissociation in Figure 5.2 has been the unit test for the development, and the MATLAB results of 230.6 seconds dropped reliably to 12.9 seconds [21].

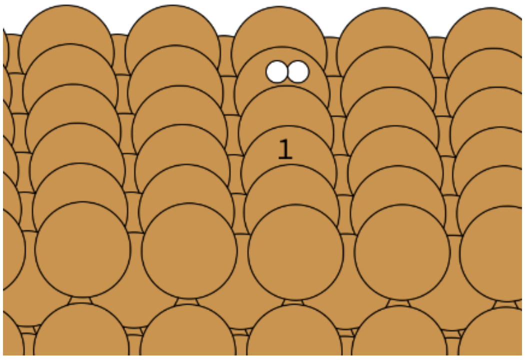


Figure 5.2. *Hydrogen molecule dissociating on a copper slab. Including the two nearest copper atoms to any moving hydrogen atom in the active set proved to further accelerate convergence, achieving a runtime of just 4.3 seconds.*

Radial cutoffs are generally used [130] to define the local environment. These can fail in complex systems [131]. Structural defects, interfaces, or thermal fluctuations cause the number of atoms within a fixed radius to vary, creating an incompatibility with machine learning models that require a fixed-size input vector.

To resolve this, the radial cutoff has been replaced with a nvel nearest-neighbor selection scheme[22]. This approach guarantees a fixed number of atoms are chosen to represent the local environment, ensuring a consistent number of degrees of freedom for the model's input. The environment is constructed by selecting a constant number of atoms that are closest to any of the primary moving atoms (e.g., the dissociating molecule or a hopping adatom).

The number of user defined parameters remains the same, swapping a radial cutoff for the number of neighbors to consider. This directly controls the scaling of the method, though sensitivity analyses show that simply increasing the number of atoms does not necessarily improve model accuracy. Future studies will involve using hyperparameter searches [132] with surrogates on more diverse systems.

The results here highlight the manner in which the nearest-neighbor cutoff effectively decouples the computational cost from the total system size, solving the scalability issue inherent to GPR without succumbing to the instability of radial cutoffs. While the remainder of this thesis focuses on molecular benchmarks, this validation ensures the architecture remains applicable to extended systems for future applications.

5.3 Gas phase molecular benchmark

Throughout this thesis, a data set of small organic molecules curated by Hermes, Sargsyan, Najm, and Zádor [133] has been used. This consists of 500 initial configurations of small gas-phase organic molecules, ranging from 7 to 25 atoms.

[21] From `gpr_optim` at commit `a4d1fdaa0ed943d0c9e8b2931db12a4148be0ba4`

[22] From HaoZeke/gpr$_{\text{optim}}$ at a837ec75f537c551d058e7a170ec880031879f52

In all cases, the Potential Energy Surface is evaluated at the HF level of theory with the 3-21G basis set, using the NWChem software package [119]. The calculations employ a spin-restricted formalism for singlet states and a spin-unrestricted formalism for doublet states. The SCF convergence threshold is set to 10^{-8} Hartree. A saddle point search is considered converged when the maximum per-atom norm of the atomic forces falls below 0.01 eV/Å.

We highlight that the criteria for inclusion in the dataset is that `sella` converges for each system. Automated tests are carried out to ensure a single negative eigenvalue is present at the saddle configuration. However, due to the inherent selection bias, success is not considered to be a factor in assessing performance in the work introducing the benchmark, since, by construction, `sella` failures are excluded [133]. This circular dependency limits in some sense the ability to compare the results across algorithms.

It is also important to note that MMF methods are often "finishing" methods, after a reasonable guess to the saddle has been generated. To this end, many single ended searches in practice are rarely over an eV or two away in energy from the nearest minima or point of interest. We show in Figure 5.3 that the distribution of energy differences (ΔE) is excessively broad, ranging over a span of nearly 20 eV.

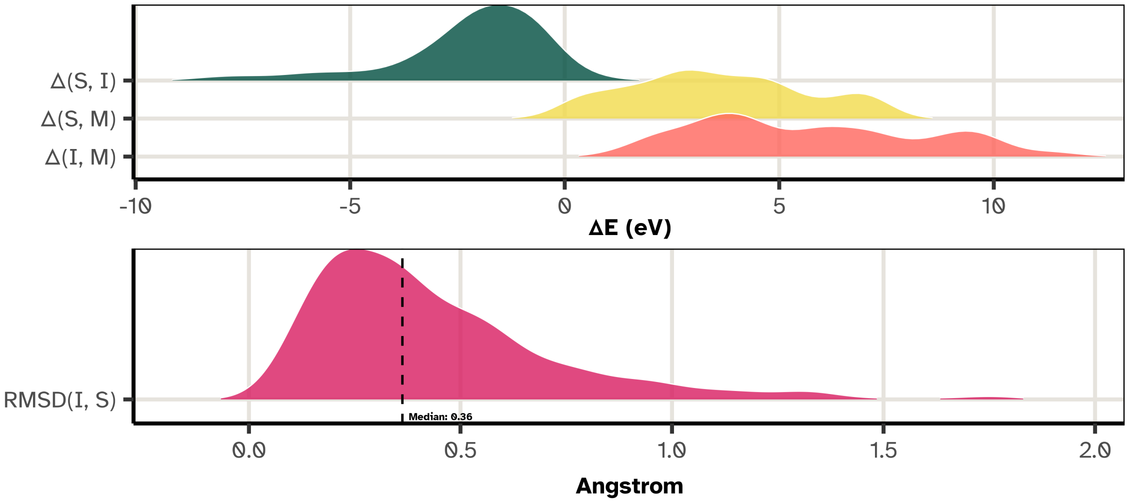


Figure 5.3. *Distribution of Energetic and Structural Properties for the Sella Transition State Dataset.* ***(Top)*** *Probability density of energy differences: Δ(S, I) between the final saddle and initial geometry, Δ(S, M) between the final saddle and the minimized initial geometry, and Δ(I, M) between the initial geometry and its minimized form. The distributions, particularly for Δ(S, I), are extremely broad, spanning nearly 20 eV. This range far exceeds expected energy barriers at reasonable temperatures and pressures for the small organic molecules in this dataset, suggesting many initial geometries are highly unstable and unrepresentative of approximate saddle points.* ***(Bottom)*** *Probability density of the root-mean-square deviation (RMSD) between the initial (I) and final saddle (S) structures. The distribution is highly skewed, with a median of 0.36 Å and a significant tail extending to large structural deviations. This, coupled with the wide energy distributions, indicates that the optimization process often involves large, chemically questionable geometric changes rather than the refinement of a reasonable guess. Plotted from data published in [50].*

The bottom panel in Figure 5.3 displays the RMSD deviation between the initial (I) and final saddle (S) structures. The RMSD distribution peaks sharply, with a median near 0.36 Å, yet a non-negligible tail stretches toward large structural deviations. Thus, while most starting guesses lie structurally close to their target saddle, a notable fraction requires substantial geometric rearrangement.

A direct inspection of the energy distributions reveals significant energetic instability in many initial guesses, a result that contrasts with the low median RMSD. The Δ(S,I) $= E_S - E_I$ distribution (top panel, green) does not center near 0 eV. Instead, its principal mode lies near -2.5 eV and a broad tail extends down to about -10 eV. This pattern indicates that initial geometries often occupy much higher electronic energy than their converged saddle points.Comparing Δ(I,M) (red) and Δ(S,M) (yellow) further quantifies this instability. The Δ(I,M) ridge shifts to more positive energy relative to Δ(S,M). This displacement simply reflects the $E_I - E_S$ energy difference, reproducing the finding from the Δ(S,I) plot and confirming that initial points frequently have higher energies than the corresponding saddles.

This disconnect between structural proximity and energetic instability introduces a substantive complication. A starting geometry that sits energetically far from the saddle region may not lie in the quadratic basin that leads to that saddle despite having proximal configurations in space (w.r.t the RMSD); instead, the high potential energy can place the start in a region of the potential energy surface that grants access to multiple competing downhill pathways. A local finishing method that follows the local gradient therefore cannot guarantee convergence toward the structurally proximal saddle. The elevated starting energy can allow the optimization to descend along an alternative reaction coordinate and converge to an unrelated minimum or a different stationary point. The tail in the RMSD(I,S) distribution corroborates this concern, as many cases require large structural changes before convergence. In sum, the dataset displays a pronounced disconnect between structural and energetic closeness. Many initial geometries function as highly unstable configurations that happen to lie near a saddle in Cartesian space but fail to occupy the correct energetic basin, which in turn makes them unreliable “reasonable initial structures”.

5.4 Performance characteristics

A comparative analysis of the GPDimer and Sella methods was conducted on a subset of 345 systems for which both optimizers converged to the same saddle point, defined as having an energy difference of less than 0.01 eV. Although this quantification does not rigorously rule out that the geometries of the saddle point configurations are comparable, the distributional analysis and general conclusions do not change. The primary metric for comparison is the number of HF calculations required to reach convergence. On average, the performance of the two methods is comparable. The GPDimer method required a median of 29 HF calculations, while Sella required a median of 31. Notably, the GPDimer achieves this efficiency using Cartesian coordinates [50], whereas Sella employs internal coordinates, which are generally considered more suitable for the varied stiffness of molecular degrees of freedom [134, 135, 136, 133]. The GPDimer proved more efficient in 57% of the cases, on average reducing the computational cost by 8 HF calculations relative to Sella.

Aggregate statistics alone obscure a key relationship. Figure 5.4 reveals how computational cost depends on the quality of the initial guess. For initial structures close to the final saddle point, possessing a RMSD below 0.6 Å, the GPDimer often requires slightly fewer

HF calculations, as the initial surfaces constructed are not optimal. Its efficiency advantage increases as the initial structural deviation increases. In the intermediate range, between 0.6 Å and 1.2 Å RMSD, the methods perform almost identically. For the most difficult cases, with initial displacements greater than 1.2 Å, the GPDimer again demonstrates superior efficiency.

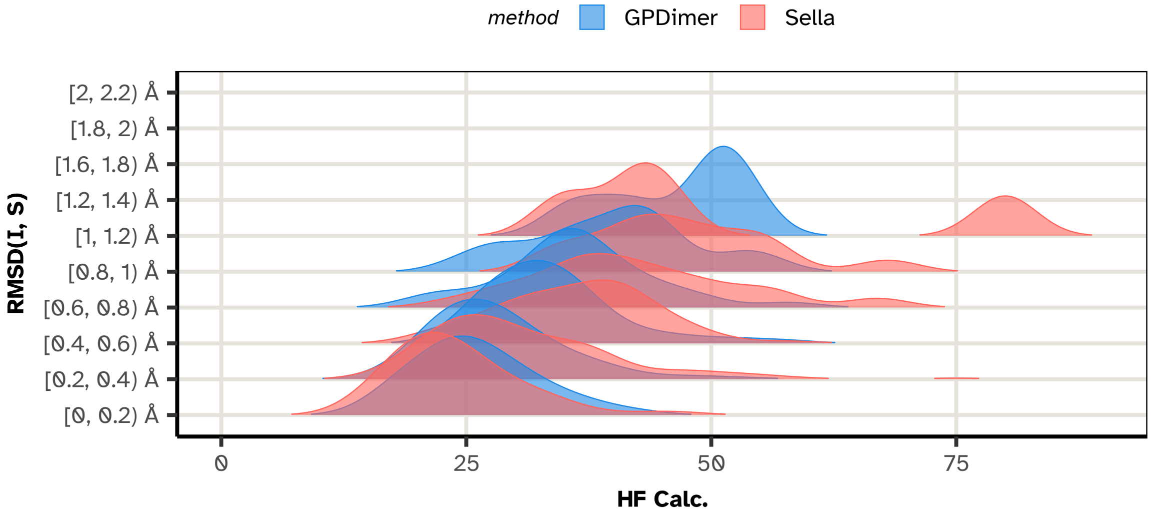


Figure 5.4. Ridgeline plot showing the distribution of Hartree-Fock (HF) calculation counts required for convergence for the GPDimer (blue) and Sella (red) methods. The data fall into bins according to the root-mean-square deviation (RMSD) between the initial and final saddle geometries. At higher RMSD values (> 0.6 Å), GPDimer shows clear efficiency. The methods perform comparably in the intermediate RMSD range. At high RMSD values, GPDimer again holds a performance advantage. The visualization confirms that algorithm efficiency depends strongly on the quality of the initial guess. Plotted from data published in [50].

The use of internal coordinates in Sella presents challenges for specific molecular geometries. In systems that approach a near-linear arrangement of three or more atoms, Sella introduces algorithmic “ghost atoms” to avoid coordinate singularities. In the 10 systems where this occurred, Sella required an average of 47.1 HF calculations, representing an 18.8% increase over its typical performance of 39.6 HF calculations (median of 31 from previous analysis). The GPR-dimer, which operates entirely in Cartesian coordinates, showed no such penalty and needed an average of 35 HF calculations on the same subset, consistent with its overall performance of 34.5 calculations. These results confirm that the Sella internal coordinate framework introduces additional computational overhead when handling challenging geometries, and is explored further in Sec. 8.

5.5 Cataloging saddles

As a diagnostic, the NEB (described in Section 2.2 and Section 4.3.2) can be used to determine the quality of the saddles found by `sella` and the GPDimer. We examine the first system in the benchmark, `singlet 000`, a 16-atom acyclic ether (C_5OH_{10}), for which both `sella` and GPDimer converge to distinct saddle points.

The GPDimer identifies a hydrogen transfer saddle point with a RMSD of just 0.2 Å after only

23 HF calculations, while Sella locates a saddle corresponding to a methyl group rotation which is lower in energy by 0.4 eV, but the configuration is significantly further away (RMSD of 0.6 Å).

Since energy alone cannot be considered sufficient for cataloging saddles [137] we explore the connectivity between the initial geometry and the Sella saddle, we employ the MMF-NEB protocol (Section 4.3.2), leveraging the fact that NEB bands can be formed between arbitrary points. The resulting optimization history, plotted against the 1D path coordinate (defined in Section 4.2, Eq. 50), is shown in Figure 5.5. The "reactant" configuration is the initial configuration (at $s = 0$, $E = 0$ eV), and the "product" is the Sella-located saddle (at $s \approx 5.2$ Å, $E \approx -1.3$ eV).

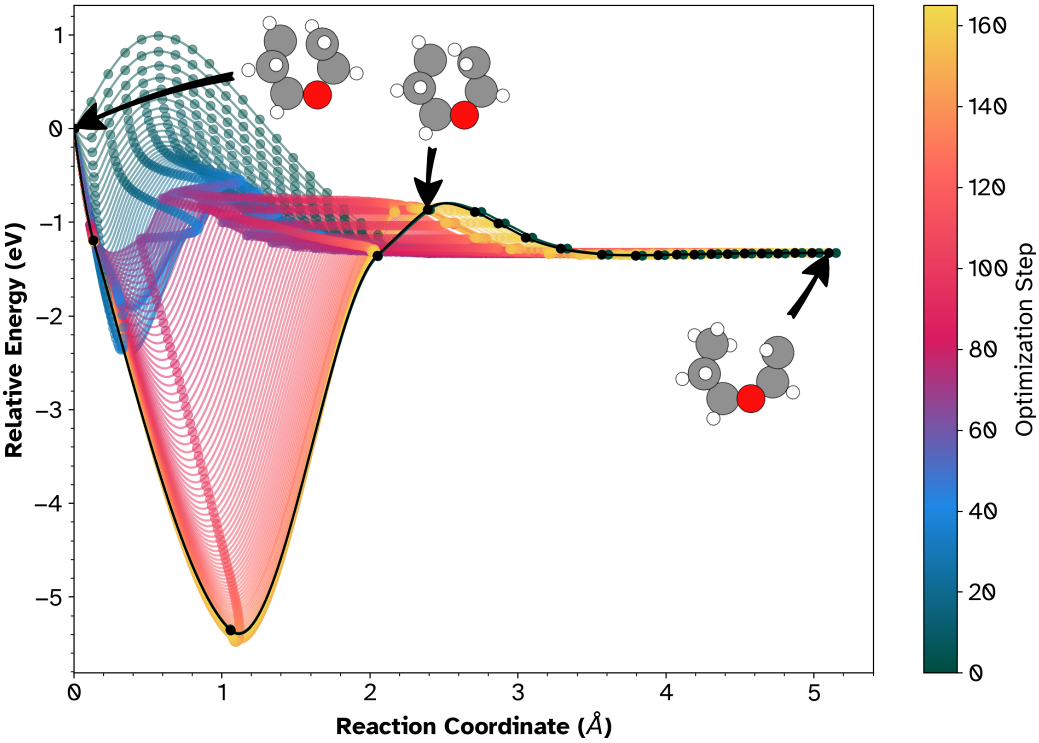


Figure 5.5. The optimization history of an NEB connecting the initial reactant (left, $s = 0$) to the Sella-located saddle point (right, $s \approx 5.2$ Å). The pathway reveals that the initial structure is not metastable; it relaxes barrierlessly into a deep intermediate minimum ($s \approx 1.1$ Å, $E \approx -5.3$ eV). To reach the Sella saddle, the system must subsequently climb out of this minimum, crossing a distinct transition state (peak at $s \approx 2.4$ Å) before proceeding to the distal Sella endpoint.

The 1D reaction coordinate profile (Figure 5.5) shows that the "initial" configuration ($s = 0$) is highly unstable and relaxes barrierlessly into a deep intermediate minimum ($E \approx -5.3$ eV). From this localized basin, the system must cross a significant barrier (the peak at $s \approx 2.4$ Å) to access the region containing the Sella saddle.The critical finding is revealed when we project these states onto the 2D landscape [122] (Figure 5.6). The saddle point found by GPDimer (marked by the white star) coincides exactly with the barrier peak at $s \approx 2.4$ Å. This indicates that the Cartesian-based GPDimer successfully identified the proximal barrier governing the immediate relaxation of the system, whereas the internal coordinate search bypassed this feature in favor of a distal stationary point. For automated kinetic network discovery (AKMC) as implemented in EON, identifying the immediate connectivity is paramount; skipping proximal barriers in favor of lower-energy distal states leads to disconnected reaction networks and missing kinetic links. Thus, GPDimer's ability to localize the geometrically nearest saddle ensures the correct step-wise mapping of the potential energy surface. The result also underscores the necessity of characterizing the local gradient manifold when sampling from high-energy initial configurations.

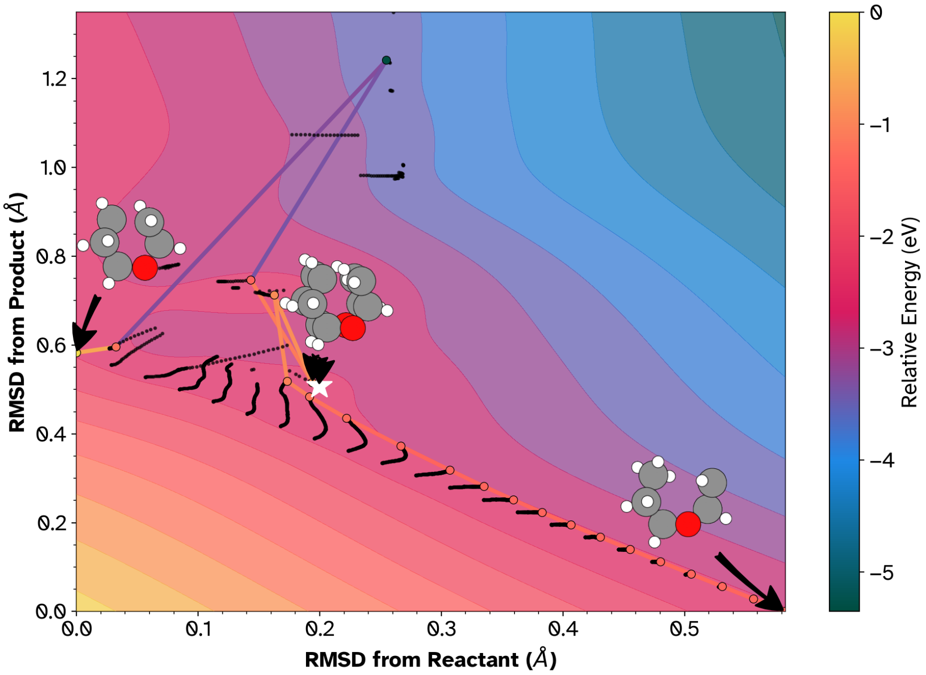


Figure 5.6. A 2D landscape projection of the NEB optimization trajectory connecting the initial reactant (top-left) to the deep intermediate minimum (bottom-right). The white star explicitly marks the GPDimer saddle, which coincides exactly with the true transition state on this path, separating the deep intermediate minimum from the subsequent product channel. Sella's trajectory (red/pink path endpoints) overshoots this state to find a lower-energy, but distal, saddle point.

5.6 Conclusions

This chapter presented gpr_optim, an implementation of Gaussian Process Regression designed to accelerate minimum mode following methods. By coupling the dimer method with a local surrogate model, we achieved an order-of-magnitude reduction in the number of electronic structure calculations required to locate transition states compared to standard dimer methods.

A critical finding of this work is that the GPR-Dimer, operating in simple Cartesian coordinates, achieves computational parity with—and often outperforms—state-of-the-art methods utilizing complex internal coordinates (Sella). This is a significant architectural advantage. For surface science and catalysis, defining non-redundant internal coordinates is fraught with difficulties due to coordinate singularities and the need for "ghost atoms." The results on the molecular benchmark demonstrate that the learning capability of the GPR can effectively compensate for the lack of coordinate sophistication. The model implicitly learns the local Hessian structure that internal coordinates attempt to explicitly encode.

Furthermore, the detailed connectivity analysis using NEB diagnostics revealed that “convergence” is an insufficient metric for success. In complex landscapes, algorithms may converge to physically valid but chemically irrelevant saddle points. The GPR-Dimer showed a robust tendency to locate the proximal saddle point governing the immediate exit from the reactant basin, whereas aggressive optimization in internal coordinates occasionally led to landscape traversal and unrelated isomerizations.

Ultimately, this framework provides a robust, coordinate-system-agnostic engine for saddle point searches. While the cubic scaling of exact GPR remains a theoretical bottleneck, the implementation of local active sets (as demonstrated with the CuH_2 unit test) ensures the method remains viable for extended systems.

Validation of results in a chemical context with efficient tools such as the MMF-NEB must be undertaken, as common point measures of success (single negative eigenvalue, convergence, “barrier” from initial) may be misleading as shown in the case study. In the twenty-first century, visual analysis is infeasible, even for the 500 systems described in this thesis (Section 5.3). Typical line plot comparisons and standard errors are dangerously oversimplified for the physical chemistry context, and we address this in the next chapter with Bayesian hierarchical models applied to modalities of dimer searches.

6 Dimer rotations and Hierarchical Bayesian models

> In God we trust; all others must bring data.
>
> W. Edwards Deming

This chapter is based on Rohit Goswami. “Bayesian Hierarchical Models for Quantitative Estimates for Performance Metrics Applied to Saddle Search Algorithms.” In: *AIP Advances* 15.8 (Aug. 2025), p. 85210. DOI: 10.1063/5.0283639 with applications in Rohit Goswami and Hannes Jónsson. “Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches.” In: *ChemPhysChem* (Nov. 2025). DOI: 10.1002/cphc.202500730

6.1 Revisiting dimer rotations

A discriminating factor among minimum mode following methods, and indeed, one of the core principles of the dimer algorithm, is to form an estimate of the minimum mode without calculating the Hessian explicitly. The dimer, as described in Section 2.1, uses an explicit rotation phase where the coordinates of the image at the midpoint remain unchanged. This procedure, designed to find the direction of lowest curvature, has evolved through several key conceptual improvements.

The original formulation involved a single finite-difference step to estimate the force gradient, followed by a single rotation to align the dimer [25]. For more complex potential energy surfaces, this strategy failed, and evolved into an iterative rotation process [140]. This transforms the rotation phase into a nested optimization problem: before each translation step, the dimer orientation is iteratively rotated until it converges upon the minimum curvature mode. The choice of numerical algorithm to perform this search profoundly impacts the method’s efficiency and reliability.

The simplest iterative approach restricts the search to a sequence of two-dimensional plane. At each rotational step k, this plane is defined by the current dimer orientation, $\hat{\mathbf{N}}_k$, and the normalized rotational force, $\hat{\mathbf{\Theta}}_k = \mathbf{F}_k^{\perp}/|\mathbf{F}_k^{\perp}|$. A new trial orientation, $\hat{\mathbf{N}}(\phi)$, is generated by a rotation within this plane:

$$\hat{\mathbf{N}}(\phi) = \hat{\mathbf{N}}_k \cos\phi + \hat{\mathbf{\Theta}}_k \sin\phi \tag{56}$$

The energy $V_D(\phi)$ is then minimized as a one-dimensional function of the angle ϕ. This method, however, can converge slowly.

Though [141] demonstrated results on the utility of the L-BFGS for translations, it has also been applied for rotations, a quirk which has cast a long shadow filtering into many different implementations including EON [29]. Recall that, as shown in Alg. 1, L-BFGS is a quasi-newton method which constructs a low-rank approximation of the inverse Hessian, $\mathbf{B}_k \approx \mathbf{H}^{-1}$, to determine a search direction. This approximation, however, provides no guarantee that the resulting search direction will converge specifically to the lowest eigenmode, $\mathbf{v}_1$. The algorithm can be deflected by or become trapped in subspaces corresponding to higher-energy eigenmodes, particularly if their eigenvalues lie close to λ_1. This can cause the optimizer to fail in its primary task of identifying the true softest mode [142].

In contrast, methods based on the CG algorithm align more naturally with the mathematical structure of the problem. The CG method and its variants, such as the Lanczos algorithm, function as powerful iterative eigensolvers specifically designed to find the extremal eigenpairs of a large symmetric matrix. The process constructs a sequence of $\mathbf{H}$ orthogonal search directions that systematically and stably isolates the lowest eigenvector. Furthermore, in the context of the dimer's outer loop of geometry steps, the converged eigenvector from the previous step provides a high-quality initial guess for the current rotation, a feature that the CG method naturally exploits to accelerate convergence. The theoretical argument therefore favors CG as the more mathematically appropriate and robust tool for the dimer rotation phase.

While theoretical arguments favor the stability of the conjugate-gradient approach, the practical performance of these optimizers can depend heavily on the specific chemical system and implementation details. To move beyond these arguments and rigorously quantify the performance trade-offs, we adopted a Bayesian statistical framework [143, 144, 145, 146].

Traditional benchmarks often neglect system-to-system variability and lack robust uncertainty quantification, making it difficult to draw reliable conclusions. Performance metrics such as the number of Potential Energy Surface calls (positive and increasing counts), total computation time (positive, skewed), and convergence (binary) frequently violate the assumptions of normality and homoscedasticity inherent in standard linear models. Furthermore, benchmark designs typically involve repeated measures, where multiple algorithmic variants are tested on the same set of chemical systems. Failing to account for the resulting non-independence non-independence of observations can lead to pseudoreplication, deflated standard error, and invalid statistical inferences.

Bluntly, any model for interpretation of success data that allows values other than true or false, any model for "efficiency" in terms of the number of calculations that permits numbers below zero, or any model for time that does not remain strictly positive throughout cannot be used to estimate metrics reliably. Implicit Gaussian distributions used for standard deviation or errors on Potential Energy Surface calls, time, or success metrics are therefore mostly meaningless.

To address these methodological shortcomings, we apply a Bayesian Generalized Linear Mixed Model framework using `brms` [147]. This approach is explicitly designed to handle such complexities, since it allows for the selection of appropriate response distributions and link functions for each type of metric, ensuring that the model's assumptions align with the data's underlying properties. We explicitly include random effects, specifically, glsp:randint for each chemical system to correctly model the hierarchical data structure, partitioning the variance between system-specific effects and the glsp:fixedeffect of the algorithmic variants. This provides a principled method for obtaining robust estimates and valid glsp:credint for the parameters of interest.

We present here the results from our full interaction models, which simultaneously estimate the main effects of the rotational optimizer (CG vs L-BFGS), the use of quaternion based external rotation removal [148] as implemented in EON [29], and their interaction term. The findings are summarized visually in Figure 6.1. This comprehensive analysis provides the most statistically powerful view of how these algorithmic choices jointly influence computational cost and convergence success.

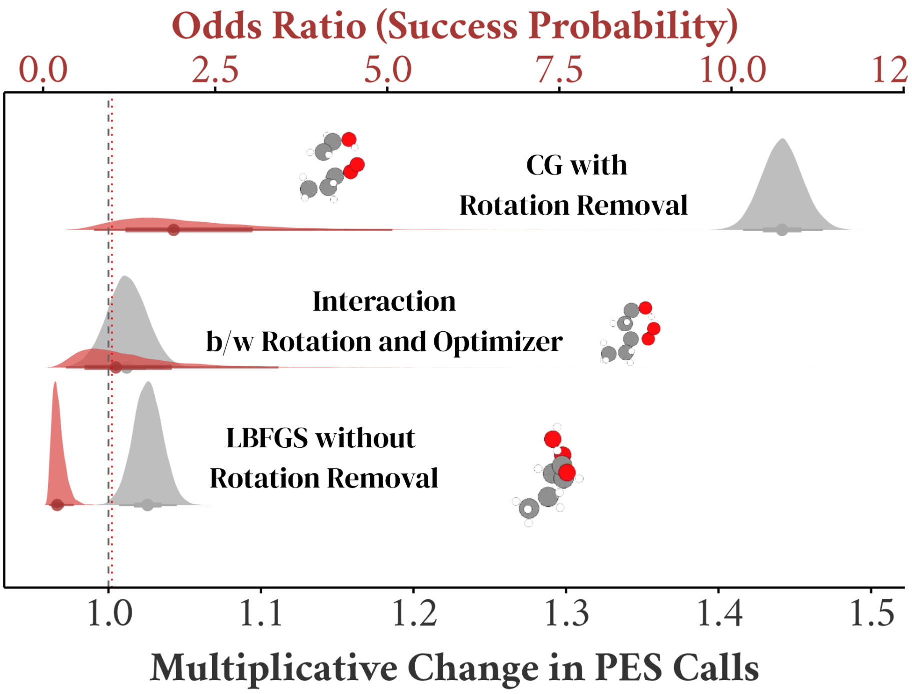


Figure 6.1. *Posterior distributions from generalized linear mixed-effects models showing the effects of algorithmic choices on computational cost (multiplicative change in PES calls, grey) and convergence probability (odds ratio, red). The analysis compares variants to a baseline of the Conjugate Gradient (CG) optimizer with rotation removal disabled. A value of 1.0 (dashed line) indicates no change relative to this baseline.* ***(Top)*** *Effect of Rotation Removal (for CG): Enabling rotation removal substantially increases the required PES calls by a factor of ~1.44 (95% CrI: [1.42, 1.47]) but has no statistically credible effect on the odds of success (the distribution overlaps 1.0).* ***(Middle)*** *Interaction Effect: The interaction between the optimizer and rotation removal settings is negligible for both cost and success probability, with distributions centered at 1.0.* ***(Bottom)*** *Effect of Optimizer (without Rotation Removal): Using the L-BFGS optimizer instead of CG results in a small but credible increase in computational cost (factor of ~1.03) and a significant reduction in the odds of a successful convergence (OR ~0.2, 95% CrI: [0.09, 0.45]). Plotted from data published in [138].*

6.2 Bayesian hierarchical model results

6.2.1 Computational Effort

It is common during benchmark calculations to use the number of samples from a energy and force calculator be used as a proxy variable for efficiency, since most studies are done with an eye towards regimes where single point calculations take a long time.

We modeled the number of calls for the Potential Energy Surface using a negative binomial distribution, as specified in Eq. 57. This choice is motivated by the physical constraints of the metric: Potential Energy Surface calls are discrete, non-negative integers. Standard linear models assume normality, which allows for non-integers and negative values. Furthermore, algorithmic performance data typically exhibits overdispersion, where the variance exceeds the mean—due to the heterogeneous nature of chemical landscapes. The Negative Binomial distribution explicitly models this overdispersion, preventing the underestimation of standard errors common in Poisson or Gaussian approximations.

$$\begin{aligned} \mathrm{PESCalls}_{ij} &\sim \mathrm{NegativeBinomial}(\mu_{ij}, \phi) \\ \log\big(\mu_{ij}\big) &= \beta_0 + \beta_1 \mathrm{DR}_{i(j))} + \beta_2 \mathrm{RR}_{i(j))} \\ &\quad + \beta_3(\mathrm{DR}_{i(j))} \times \mathrm{RR}_{i(j))}) + u_j \end{aligned} \tag{57}$$

Our analysis of this model revealed two key findings. First, enabling rotation removal (β_2) incurred a substantial computational penalty, increasing the median number of PES calls by a factor of 1.44 (95% CrI: [1.42, 1.47]). Second, using the L-BFGS optimizer (β_1) resulted in a small but statistically credible increase in cost by a factor of 1.03 (95% CrI: [1.01, 1.05]) compared to CG. The interaction term (β_3) was not credibly different from zero, indicating the effects were largely additive. The model also quantified significant system-to-system variability in baseline computational cost ($\sigma_u \approx 0.63$).

6.2.2 Wall time estimates

While the number of PES calls serves as the standard theoretical proxy for computational effort, the ultimate metric for a practitioner is the total wall-clock time elapsed. To ensure our conclusions were not merely an artifact of our chosen proxy, and to quantify the real-world time costs, we performed a parallel analysis on the total time for each calculation.

Wall time is a strictly positive, continuous variable. The distribution of runtimes is often right-skewed, featuring a long tail of difficult systems while the majority converge quickly. A standard Gaussian model would imply a symmetric distribution, potentially assigning non-zero probability to physically impossible negative times. We therefore employed a Gamma generalized linear mixed model, which is naturally defined for $x > 0$ and flexibly accommodates the skewness inherent in computational timing data. The model specification is shown in Eq. 58.

$$\begin{aligned} \mathrm{TotalTime}_{ij} &\sim \mathrm{Gamma}(\mu_{ij}, \alpha) \\ \log\big(\mu_{ij}\big) &= \beta_0 + \beta_1 \mathrm{DR}_{i(j)} + \beta_2 \mathrm{RR}_{i(j)} \\ &\quad + \beta_3(\mathrm{DR}_{i(j)} \times \mathrm{RR}_{i(j)}) + u_j \end{aligned} \tag{58}$$

The results of this analysis mirrored that of the effort proxy analysis. Enabling rotation removal (β_2) incurred a substantial time penalty, increasing the median total time by 43.0% (95% CrI: [40.4%, 45.6%]). Similarly, the choice of the L-BFGS optimizer (β_1) led to a small but credible increase in runtime of 2.6% (95% CrI: [0.7%, 4.5%]). The interaction term (β_3) remained negligible, confirming the additive nature of these effects.

This striking consistency across different metrics and model families provides strong evidence that the number of electronic structure calculations is the dominant factor driving total computation time. The performance conclusions drawn from the PES call analysis are therefore robust and directly translate to practical runtime considerations.

6.2.3 Convergence Success

We analyzed the probability of a successful search using a Bernoulli logistic regression model, detailed in Eq. 59. As convergence is a binary event ($y \in \{0, 1\}$), the search either isolates a saddle point or it does not, the Bernoulli distribution is the canonical probability model for such dichotomous outcomes.

$$\begin{aligned} \text{Success}_{ij} &\sim \text{Bernoulli}(p_{ij}) \\ \text{logit}(p_{ij}) &= \beta_0 + \beta_1 \text{DR}_{i(j))} + \beta_2 \text{RR}_{i(j))} \\ &\quad + \beta_3 (\text{DR}_{i(j))} \times \text{RR}_{i(j))}) + u_j \end{aligned} \tag{59}$$

This model demonstrated a clear and significant difference in robustness between the optimizers. Compared to the CG baseline, the L-BFGS optimizer (β_1) was substantially less likely to converge, with an estimated odds ratio (OR) of 0.20 (95% CrI: [0.09, 0.45]). In contrast, neither the main effect of rotation removal (β_2) nor the interaction term (β_3) had a statistically credible impact on the odds of success. The large standard deviation of the random intercepts ($\sigma_u \approx 3.6$) underscored that intrinsic system properties are a primary determinant of convergence success.

Our full interaction models show that the CG optimizer is both more efficient and significantly more robust than L-BFGS for this dataset and the current implementation of the rotation removal [148] process, while enabling this implementation of rotation removal increases computational cost without a corresponding benefit to the success rate. Finally, we discuss how these algorithmic choices interact with invariance removal procedures within the context of Gaussian Process models.

6.3 Revisiting rotation removal

Though quaternions may not have been best implemented EON, we return to the concept of rotations as implemented in Goswami and Jónsson [139].

A Gaussian Process model approximates the potential energy surface without inherent knowledge of the physical invariances of the system. Consequently, a proposed optimization step may contain spurious components corresponding to the external degrees of freedom: overall translation and rotation of the entire molecule. The optimizer actively removes these components from the proposed translation step vector to ensure that movements occur only along internal coordinates, which represent genuine changes in molecular geometry.

The procedure first constructs a basis set spanning the space of infinitesimal rigid-body motions. For a system of N atoms, this space has six dimensions (or five for a linear molecule). The procedure generates three basis vectors for translation, $\{\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z\}$, where each vector $\mathbf{t}_k$ represents a unit displacement of all atoms along the Cartesian axis k.

$$(\mathbf{t}_k)_{3i+k-1} = 1 \quad \forall i \in \{1, ..., N\} \tag{60}$$

Next, the procedure generates three basis vectors for rotation, $\{\mathbf{l}_x, \mathbf{l}_y, \mathbf{l}_z\}$, derived from the expression for infinitesimal rotation about the center of mass, $\mathbf{r}'_i = \mathbf{r}_i - \mathbf{r}_{\text{com}}$. An infinitesimal rotation of the entire system corresponds to a displacement $\delta\mathbf{r}_i = \delta\boldsymbol{\omega} \times \mathbf{r}'_i$. The rotational basis vectors thus take the form:

$$\mathbf{l}_x = \sum_{i=1}^{N} \hat{\mathbf{e}}_x \times \mathbf{r}'_i \tag{61}$$

$$\mathbf{l}_y = \sum_{i=1}^{N} \hat{\mathbf{e}}_y \times \mathbf{r}'_i \tag{62}$$

$$\mathbf{l}_z = \sum_{i=1}^{N} \hat{\mathbf{e}}_z \times \mathbf{r}'_i \tag{63}$$

The algorithm then applies the Gram-Schmidt process to this set of six vectors to produce an orthonormal basis, $\{\mathbf{u}_k\}$, that spans the external degrees of freedom. For any proposed translation step, $\mathbf{s} \in \mathbb{R}^{3N}$, the algorithm projects out the external components. The component of the step corresponding to translation and rotation, $\mathbf{s}_{\text{ext}}$, projects onto this basis:

$$\mathbf{s}_{\text{ext}} = \sum_k (\mathbf{s} \cdot \mathbf{u}_k)\mathbf{u}_k \tag{64}$$

The pure internal step, $\mathbf{s}_{\text{int}}$, then becomes the original step minus its external projection:

$$\mathbf{s}_{\text{int}} = \mathbf{s} - \mathbf{s}_{\text{ext}} \tag{65}$$

A feedback mechanism enhances the stability of the GP-driven search. The algorithm computes the magnitude of the removed component, $||\mathbf{s}_{\text{ext}}||$. If this magnitude exceeds a defined threshold, θ_{rot}, it signals that the GP model likely predicts a large, unphysical torque on the molecule. In such cases, the procedure discards the projection and reverts to the original, unprojected step $\mathbf{s}$. Subsequent step-size limitation guardrails then typically intercept this large, physically questionable step, triggering a resampling of the true potential energy surface to improve the GP model. When the magnitude of the removed component remains below the threshold, the algorithm accepts the purified internal step $\mathbf{s}_{\text{int}}$. This ensures a more precise update to the molecular geometry, guided only by genuine internal forces which we find to be more robust in Chapter 8.

In practice, since energy does not depend on rotations, the threshold tends to large values. Collectively, these invariance considerations and algorithmic choices move us closer to the overarching goal: a turnkey, walltime-efficient, reliable Gaussian Process optimization framework for molecular and extended systems.

In the following section, we shift our focus to data efficiency, discussing practical strategies for reducing computational cost in Gaussian Process guided optimizations.

6.4 Time-to-Solution as a Function of Effort

While independent analyses of computational effort (PES calls) and wall time provide valuable baseline metrics, they do not fully capture the dynamic relationship between the two. In a practical optimization scenario, the "cost" of a method is not just how many steps it takes, but how the time-per-step scales as the optimization trajectory lengthens.

To investigate this, we modeled the total time-to-solution (T) as a continuous function of the computational effort (N_{calls}), conditioned on the optimization method. We employed a Bayesian Hierarchical Gamma Spline model (Eq. 66) to capture the potentially non-linear scaling behaviors while accounting for the strict positivity and skewness of runtime data.

$$\begin{aligned} T_{ij} &\sim \text{Gamma}(\mu_{ij}, \alpha) \\ \log(\mu_{ij}) &= \beta_0 + \beta_{\text{method}} + f_{\text{method}}(\log(N_{\text{calls},ij})) + u_j \end{aligned} \tag{66}$$

Here, f_{method} represents a thin-plate regression spline smooth term allowed to vary by method. This approach enables us to separate the fixed overhead of a method (intercept differences) from its scaling behavior (the shape of the spline). We also incorporated random intercepts u_j for each system to control for the intrinsic expense of different chemical environments.

6.4.1 Extrapolation and Data Efficiency

A challenge in benchmarking machine-learning enhanced methods (like GPDimer and OTGPD, introduced in Chapter 8) against classical baselines (Dimer) is the disparity in convergence speed. The ML methods often converge with significantly fewer calls, leaving no observed data in the "high-effort" regime populated by the classical method.

Figure 6.2 visualizes this relationship. The solid lines represent the interpolated predictions within the observed data range for each method, while the dashed lines indicate extrapolation.

The plot reveals a distinct crossover. For extremely short runs (low PES calls), the ML-based methods incur a higher time cost due to the overhead of Gaussian Process regression. However, as the optimization difficulty increases (moving right on the x-axis), this overhead is amortized. It is also worth noting that the absolute time per call here is very low, since the Hartree-Fock calculations considered in this thesis do not take much time per configuration. Systems with more complex potential energy calculations will perform far better under the GP as the Dimer methods take an order of magnitude more calls.

6.4.2 Quantitative Comparison

To rigorously quantify this trade-off, we compared the expected time-to-solution at distinct regimes of computational effort, the overall results of which are shown in Table 6.1.

First, we evaluated the methods at a fixed budget of 30 PES calls, a typical early-phase regime for GP methods. At this point, the classical Dimer method is slightly faster (11.5 min) compared to GPDimer (12.8 min), reflecting the lower per-step cost of the classical algorithm. The OTGPD method, however, effectively mitigates this overhead (11.3 min), performing on par with the classical baseline even in this low-data regime.

However, comparing methods at a fixed **x**-value is artificial if the methods require vastly

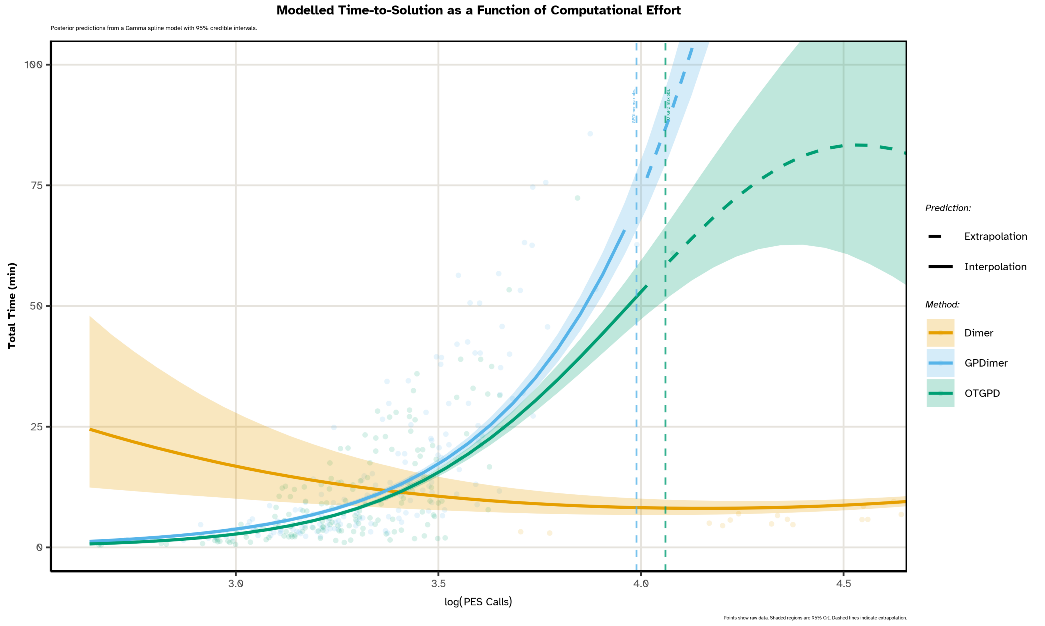


Figure 6.2. *Modelled Time-to-Solution as a Function of Computational Effort. Posterior predictions from a Gamma spline model. The solid curves indicate the interpolation range where data was observed; dashed curves indicate extrapolation. The vertical dashed lines mark the maximum observed PES calls for the GP-based methods.*

different amounts of effort to solve the same problem. A more physically meaningful comparison evaluates each method at its median convergence effort which approximates the typical “price” to solve a system.

Table 6.1. *Results of modelled time to solution as a function of computational effort.*

Method	Median PES Calls	Expected Time (min)	95% CrI
OTGPD	**28**	**9.0**	**[8.6, 9.5]**
GPDimer	30	12.8	[12.2, 13.5]
Dimer	254	20.5	[19.2, 21.8]

At their respective operating points, the efficiency gain is stark. The classical Dimer requires typically 254 calls, resulting in an expected runtime of 20.5 minutes. The OTGPD method achieves solution with only 28 calls on average, dropping the total wall time to 9.0 minutes. This represents a factor of 2.2x speedup in real-world wall time, confirming that the dramatic reduction in force calls provided by the GP surrogate translates directly to time savings even for trivial energy surfaces, despite the added computational overhead of the machine learning model.

6.5 Conclusions

This chapter established a rigorous statistical framework for evaluating computational performance, moving the discourse from anecdotal benchmarking to hierarchical Bayesian inference. By employing generalized linear mixed models with appropriate response distributions—Negative Binomial for effort, Gamma for wall time, and Bernoulli for convergence—we quantified the distinct trade-offs inherent in algorithmic choices while explicitly accounting for system-specific variability.

The analysis of the dimer method's rotation phase yielded decisive results. We demonstrated that the Conjugate Gradient optimizer offers superior robustness compared to L-BFGS, contradicting default choices in established software packages. Furthermore, we showed that the specific implementation of rotation removal within the EON suite incurred a significant computational penalty without providing a credible improvement in convergence success.

However, the inefficiency of one specific implementation does not negate the underlying physical principle. As discussed in Section 6.3, the explicit removal of external invariances functions as a critical stability mechanism for Gaussian Process surrogates [139], preventing unphysical extrapolations in the active learning loop. This distinction underscores a broader theme in computational method development: the utility of a physical constraint depends heavily on its numerical integration within the optimization framework.

Using the same framework, modeling the time-to-solution landscape (Section 6.4) resolves the "overhead vs. efficiency" debate for machine-learning enhanced optimization. While surrogate-based methods like GPDimer and OTGPD incur a higher computational cost per step for cheap potentials evident in the crossover observed in the Gamma spline analysis, this overhead is effectively amortized over the trajectory. Our analysis confirms that the On-The-Fly Gaussian Process Dimer (OTGPD) method reduces the median computational effort so drastically (from ~254 to ~28 calls) that it yields a net reduction in total wall time of approximately 56% compared to the classical baseline. This result statistically validates the core premise of this thesis: that data efficiency translates directly to wall-clock acceleration, provided the surrogate model is strictly stabilized.

With a validated statistical methodology to measure success and a refined understanding of how to handle physical invariances, we now address the primary bottleneck in machine-learning-guided optimization, that of total wall time. The following chapter applies these insights to the challenge of data efficiency, developing strategies to scale Gaussian Process surrogates to chemically relevant system sizes.

7 Data efficiency for Gaussian Processes

> I have only made this letter longer because I have not had the time to make it shorter.
>
> Blaise Pascal
> The Provincial Letters

This chapter is partially based on Rohit Goswami and Hannes Jónsson. *Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches.* Oct. 2025. DOI: `10.48550/arXiv.2510.06030`. arXiv: `2510.06030 [physics]`.

A collection of random variables, any finite number of which have a joint multivariate normal (Gaussian) distribution; defines a distribution over functions. (Section 2.4.1) surrogates promise analytic posteriors and gradients, yet their naive scaling, $O(M^3N^3)$ in time and $O(M^2N^2)$ in memory for M geometries and N atoms, blocks routine use. This chapter develops a narrative from lossless algebraic reorganizations that preserve fidelity, to lossy approximations that trade information for wall-time. Along the way, we confront pathologies that emerge in active learning loops and establish safeguards that let the later chapters stand on firm computational ground.

7.1 Quicker inversions through reshaping

In this work, the optimization routine circumvents the explicit inversion of the full kernel matrix. Instead, the log MLL and its gradient with respect to kernel hyperparameters are evaluated using efficient matrix-vector products and decompositions. This is realized by assembling a rectangular matrix $\mathbf{R}$ of size $M(3N+1) \times (3N+1)$, where rows correspond to training targets and columns to specific observables.

Reordering or reshaping a matrix is algebraically a no-op regarding fundamental operations such as inversion and decomposition [149]. Any permutation or block reorganization of a matrix $\mathbf{K}$ can be written as

$$\mathbf{K}' = \mathbf{P}\mathbf{K}\mathbf{P}^T$$

where $\mathbf{P}$ is an orthogonal permutation matrix ($\mathbf{P}^T = \mathbf{P}^{-1}$). For any vector $\mathbf{y}$,

$$\mathbf{K}\mathbf{y} = \mathbf{0} \iff \mathbf{K}'(\mathbf{P}\mathbf{y}) = \mathbf{0}$$

and

$$\mathbf{K}^{-1} = \mathbf{P}^T(\mathbf{K}')^{-1}\mathbf{P}$$

Thus, the solution to $\mathbf{Ka} = \mathbf{y}$ is unchanged under permutation:

$$\mathbf{a} = \mathbf{K}^{-1}\mathbf{y} = \mathbf{P}^T(\mathbf{K}')^{-1}\mathbf{Py}$$

and the determinant and decomposition (e.g., Cholesky) are similarly invariant up to permutation:

$$\det \mathbf{K} = \det \mathbf{K}', \quad \text{and} \quad \mathbf{K} = \mathbf{LL}^T \implies \mathbf{K}' = (\mathbf{PL})(\mathbf{PL})^T$$

Consider a "reshaped" or block-organized version of $\mathbf{K}$, e.g., storing targets grouped by configuration or observable. The Cholesky decomposition (used in SCG and Gaussian Process marginal likelihood) and matrix-vector solves are unaffected:

$$\mathbf{K} = \mathbf{LL}^T \implies \mathbf{K}' = (\mathbf{PL})(\mathbf{PL})^T$$

and

$$\mathbf{K}^{-1}\mathbf{y} = \mathbf{P}^T(\mathbf{K}')^{-1}(\mathbf{Py})$$

This means that, for SCG optimization, the coefficients obtained from the full kernel or from any reshaped, block, or permuted version are identical (after applying the corresponding permutation to the solution vector).

By exploiting this invariance, practical implementations (including this work and GPstuff [127]) use block matrices or grouped layouts for efficiency, without loss of mathematical fidelity. The organization is chosen to minimize computational overhead and maximize parallelism, but the GP predictions, marginal likelihood, and SCG updates remain unchanged.

Thus, the storage and computational cost of this approach scale as $O(MN^2)$, which is linear in the number of training geometries and quadratic in the number of atoms. This reduction is achieved without any loss of information or accuracy, as all targets are still included in the optimization; the difference is purely the result of efficient organization and evaluation of the required matrix operations. Scaling from a software design perspective is known to unlock linear scaling in computational chemistry [150].

For example, for a small system with $N = 5$ atoms and $M = 1$ configuration, both the block and full kernel have only 256 elements ($10^{2.4}$). However, at $M = 75$ configurations, the full kernel would require storage for 1.44×10^6 elements ($10^{6.2}$), while the block matrix requires only 19,200 elements ($10^{4.3}$). For a larger molecule ($N = 18$), the difference is even more dramatic: at $M = 75$, the full kernel has 1.7×10^7 elements ($10^{7.2}$), while the block matrix contains just 226,875 elements ($10^{5.4}$).

Figure 7.1 illustrates the practical scaling of the block matrix (solid lines) versus the theoretical full kernel (dashed lines). For a system with $N = 18$ atoms and $M = 75$ geometries, the full kernel requires 1.7×10^7 elements ($10^{7.2}$), whereas the block matrix requires only 2.2×10^5 elements ($10^{5.4}$). This efficiency gain enables GPR model training on significantly larger systems.

7.2 Rank one covariance updates for new data

With fixed hyperparameters, one new observation permits an $O(n^2)$ Cholesky update rather than a full $O(n^3)$ refactor:

$$\mathbf{L}_{n+1} = \begin{pmatrix} \mathbf{L}_n & \mathbf{0} \\ \mathbf{l}^T & l_d \end{pmatrix}, \quad \mathbf{L}_n\mathbf{l} = \mathbf{k}, \quad l_d = \sqrt{k_{dd} - \mathbf{l}^T\mathbf{l}}. \tag{67}$$

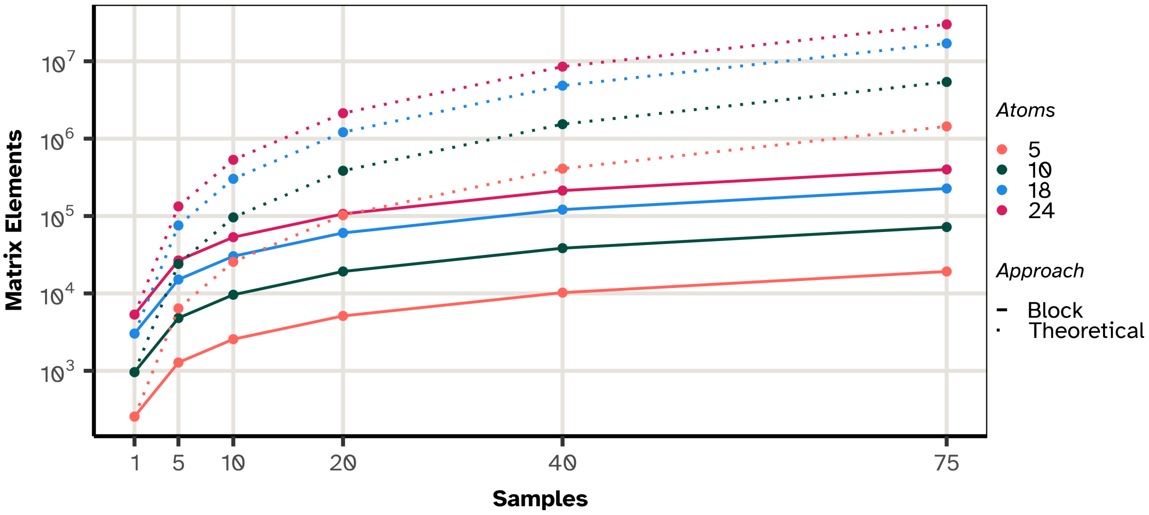


Figure 7.1. *Kernel matrix element scaling for Gaussian Process training with energy and force data. Solid lines (with points) denote the practical block matrix ($M(3N+1) \times (3N+1)$), while dotted lines denote the theoretical full kernel ($(M(3N+1))^2$). Values are log-scale versus the number of training geometries. For $N = 18$ atoms and $M = 75$ geometries, the block matrix is orders of magnitude smaller than the full kernel formulation.*

In the `gpr_optim` this approach is not pursued because the formulation of a rank one update procedure for adding the training covariance is overshadowed by the subsequent re-optimization of the hyperparameters, where the entire matrix is inverted repeatedly.

The search for optimal hyperparameters, $\boldsymbol{\theta}^*$, requires maximizing the marginal likelihood, but the covariance matrix $\mathbf{K}_{\boldsymbol{\theta}}$ is a function of the hyperparameters. Each candidate set of hyperparameters $\boldsymbol{\theta}_i$ in the search defines an entirely new matrix. The incremental update is rendered useless because the base matrix $\mathbf{K}_n$ is not fixed; it is constantly being redefined.

Therefore, for a search consisting of N_{trials} candidate hyperparameter sets, the total computational cost for the optimization step is dominated by the N_{trials} full, $O(n^3)$ decompositions required to evaluate the marginal likelihood for each candidate. The single, final $O(n^2)$ update for the chosen $\boldsymbol{\theta}^*$ is computationally insignificant in comparison.

7.3 Pruning over data

The block matrix and rank one update formulation so far forms a concrete software representation of a mathematical algebraic form, a design space solution; while providing impressive performance, the fundamental scaling in the context of a high number of points is not addressable in this space.

Approximations to the Gaussian Process [151, 48, 152] often rely on utilizing a subset of training examples instead of the global update form in Algorithm 5.

Algorithm 5 GP-Guided Optimization: No Pruning (Global Model)

```
Input: objective $U(\mathbf{x})$, gradient $\nabla U(\mathbf{x})$, initial $\mathbf{x}_0$, max iterations $T_{\max}$, initial step size $\eta_0$
Output: final position $\mathbf{x}_T$, training set $\mathcal{D}_T$
Initialize: $\mathcal{D}_1 \leftarrow \{(\mathbf{x}_0, U(\mathbf{x}_0), \nabla U(\mathbf{x}_0))\}$
$t \leftarrow 1$
while $t \leq T_{\max}$ and not converged do
                                                        ▷ Fit GP on all accumulated data
    Fit GP to $\mathcal{D}_t$ by maximizing marginal likelihood (Eq. 72)
                                                        ▷ Predict gradient at current location
    $\hat{\mathbf{g}}_t \leftarrow \nabla_{\mathbf{x}_t} \mu(\mathbf{x}_t)$ via Eq. 73
                                                        ▷ Line search for step size
    $\mathbf{x}_{t+1}, \eta_t \leftarrow \mathrm{ArmijoLineSearch}(\mathbf{x}_t, \hat{\mathbf{g}}_t, U, \eta_0)$
                                                        ▷ Observe objective and gradient at new point
    $u_{t+1} \leftarrow U(\mathbf{x}_{t+1}), \nabla u_{t+1} \leftarrow \nabla U(\mathbf{x}_{t+1})$
                                                        ▷ Accumulate into training set
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, u_{t+1}, \nabla u_{t+1})\}$
    $t \leftarrow t + 1$
end while
return $\mathbf{x}_t$, $\mathcal{D}_t$
```

However, when a data reduction heuristic is applied at each step of an active learning loop (as shown in Algorithm 6), it creates a tight coupling between the inference approximation and the data acquisition policy. The chosen approximation affects the posterior, which in turn affects the acquisition function's decision about where to sample next. This new sample then influences the subsequent approximation, creating a feedback loop that can lead to pathological behavior.

Algorithm 6 GP-Guided Optimization: Online Pruning (Local Model)

```
**Input:** objective $U(\mathbf{x})$, gradient $\nabla U(\mathbf{x})$, initial $\mathbf{x}_0$, max iterations $T_{\max}$, lengthscale $\ell$, pruning multiplier $\alpha$
**Output:** final position $\mathbf{x}_T$, final training set $\mathcal{D}_T^{\text{pruned}}$
Initialize: $\mathcal{D}_1 \leftarrow \{(\mathbf{x}_0, U(\mathbf{x}_0), \nabla U(\mathbf{x}_0))\}$
$t \leftarrow 1$
$r_p \leftarrow \alpha\ell$                                   ▷ Set pruning radius (Eq. 75)
**while** $t \le T_{\max}$ and not converged **do**
                                                   ▷ **Prune:** retain only nearby observations
    $\mathcal{D}_t^{\text{pruned}} \leftarrow \{(\mathbf{x}_i, u_i, \nabla u_i) \in \mathcal{D}_t : \|\mathbf{x}_i - \mathbf{x}_t\| \le r_p\}$ (Eq. 74)
                                                   ▷ Fit GP on pruned set only
    Fit GP to $\mathcal{D}_t^{\text{pruned}}$ by maximizing marginal likelihood
                                   ▷ Predict gradient at current location (from limited data)
    $\hat{\mathbf{g}}_t \leftarrow \nabla_{\mathbf{x}_t}\mu(\mathbf{x}_t)$ via Eq. 73
                                                   ▷ Line search for step size
    $\mathbf{x}_{t+1}, \eta_t \leftarrow \text{ArmijoLineSearch}(\mathbf{x}_t, \hat{\mathbf{g}}_t, U, \eta_0)$
                                         ▷ Observe objective and gradient at new point
    $u_{t+1} \leftarrow U(\mathbf{x}_{t+1}), \nabla u_{t+1} \leftarrow \nabla U(\mathbf{x}_{t+1})$
                      ▷ Add to global history (but pruned set still holds only nearby data)
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, u_{t+1}, \nabla u_{t+1})\}$
    $t \leftarrow t + 1$
**end while**
**return** $\mathbf{x}_t$, $\mathcal{D}_t^{\text{pruned}}$
```

We demonstrate this pathology with the Rosenbrock potential, defined as:

$$U(\mathbf{x}) = (a - x_1)^2 + b(x_2 - x_1^2)^2 \tag{68}$$

with parameters $a = 1$ and $b = 100$. The global minimum lies at $\mathbf{x}^* = (1, 1)$ with $U(\mathbf{x}^*) = 0$. The gradient is:

$$\nabla U(\mathbf{x}) = \begin{pmatrix} -2(a - x_1) - 4bx_1(x_2 - x_1^2) \\ 2b(x_2 - x_1^2) \end{pmatrix} \tag{69}$$

with a starting point at $\mathbf{x}_0 = (0.0, 1.5)$ chosen outside the valley, and convergence is when the force norm predicted by the Gaussian Process drops below $1e^{-6}$.

Here, a Gaussian Process optimizer with derivatives is compared against an identical optimizer that employs a naive online pruning rule: at each step, it discards all observations outside a fixed radius from its current position. This can be seen as a crude, state-dependent form of sparsification.

The GP model maintains a joint distribution over function values and their gradients. At each step t, one maintains a training set:

$$\mathcal{D}_t = \{(\mathbf{x}_i, u_i, \nabla u_i) : i = 1, \dots, n_t\} \tag{70}$$

where $u_i = U(\mathbf{x}_i)$ is the function value and $\nabla u_i = \nabla U(\mathbf{x}_i)$ is the gradient vector at each observed location $\mathbf{x}_i$.

The Gaussian Process prior employs a squared-exponential (RBF) kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{71}$$

with signal variance σ_f^2 and lengthscale ℓ. The posterior mean prediction at a test point $\mathbf{x}_*$ is:

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y} \tag{72}$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_{n_t})]^T$ collects covariances to observed points, and $\mathbf{K}$ is the full covariance matrix of all observations and their derivatives.

The predicted gradient at $\mathbf{x}_*$ is obtained by differentiating the posterior mean:

$$\nabla_* \mu(\mathbf{x}_*) = \frac{\partial}{\partial \mathbf{x}_*}\left(\mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}\right) = \left(\frac{\partial \mathbf{k}_*}{\partial \mathbf{x}_*}\right)^T \mathbf{K}^{-1} \mathbf{y} \tag{73}$$

This predicted gradient $\hat{\mathbf{g}}_t = \nabla_* \mu(\mathbf{x}_t)$ at the current location $\mathbf{x}_t$ drives the next step via a simple gradient descent with Armijo line search [93].

Online pruning removes observations deemed “distant” from the current location. At iteration t, given the current position $\mathbf{x}_t$ and the pruning radius r_p, one retains only observations satisfying:

$$\mathcal{D}_t^{\text{pruned}} = \left\{(\mathbf{x}_i, u_i, \nabla u_i) \in \mathcal{D}_t : \|\mathbf{x}_i - \mathbf{x}_t\| \leq r_p\right\} \tag{74}$$

The pruning radius we set as a multiple of the lengthscale:

$$r_p = \alpha \ell \tag{75}$$

where $\alpha \in (0, 1)$ is a multiplier (e.g., $\alpha = 0.8$). This filtered dataset then retrains the Gaussian Process for the next iteration.

The rationale appears sound: observations far from the current location exert minimal influence on the posterior (their kernel weight decays exponentially with distance), so discarding them saves computation without sacrificing local accuracy. However, this reasoning ignores the coupling between data support and inference.

Figure 7.2 presents two optimization trajectories guided by Gaussian Process inference with derivative observations on the Rosenbrock function (Eq. 68). Both trajectories use identical hyperparameters: a lengthscale $\ell = 1.6$ and signal variance $\sigma_f = 1.1$. The online pruning model uses a multiplier of $\alpha = 0.3$, yielding a pruning radius of $r_p = 1.3 \times 0.3 = 0.48$. Each trajectory is “warm-started” with 100 randomly selected data points. This initialization is the key mechanism driving the dramatic effect shown. The warm-start points are not sampled uniformly but are instead placed in an asymmetric cloud, predominantly on one side of the starting point $\mathbf{x}_0$. This biased initial dataset has profoundly different consequences for each optimizer. The global optimizer, following the black path, incorporates all 100 points, and its initial model of the landscape is permanently skewed by this lopsided data distribution. In contrast, the pruned optimizer, following the white path, also observes these 100 points at $t = 0$ but immediately discards the vast majority of them after its first step, as they fall outside its tight pruning radius. Its model is consequently based on only a handful of the closest points. This warm-start strategy ensures the two models begin with fundamentally different "beliefs" about the objective function, forcing their optimization paths to diverge from the very beginning.

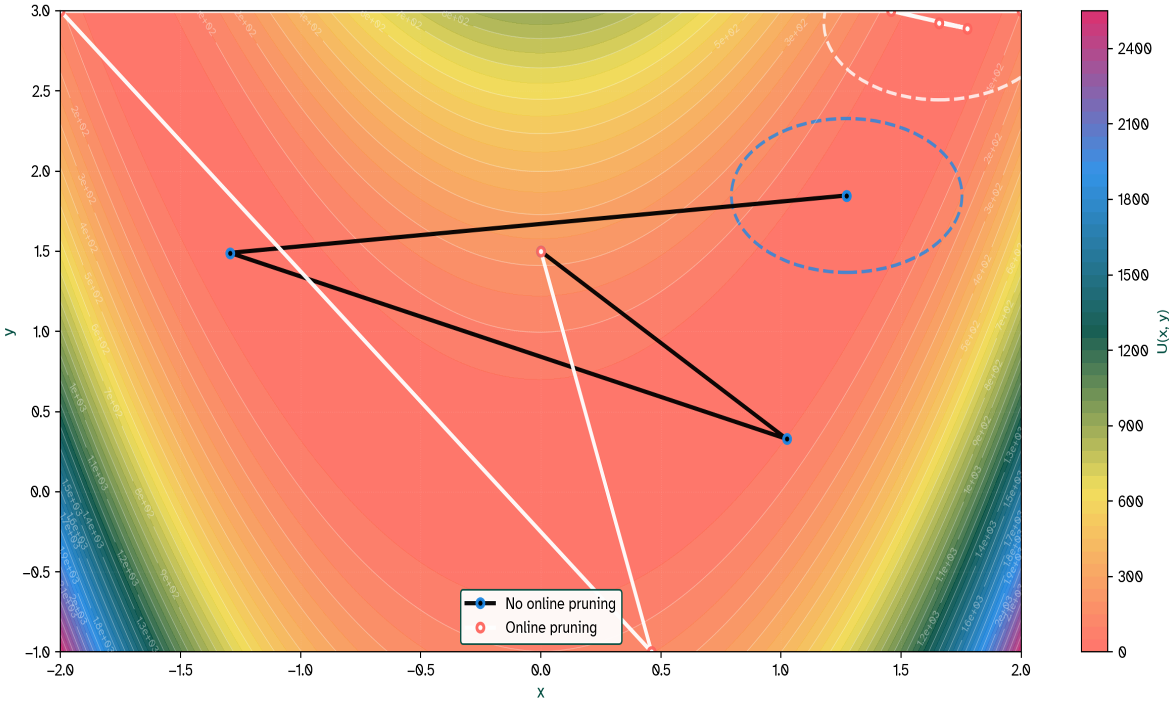


Figure 7.2. Online pruning induces trajectory divergence in GP-guided optimization of the Rosenbrock function (Eq. 68). The landscape is shown with contours. Black path (Algorithm 5): all observations retained, shown as white circles. White path (Algorithm 6): online pruning applied; white circles denote retained observations, black rings mark observations pruned away at the final step (those lying outside radius $r_p = 0.48$ from the final position). Dashed circles indicate the pruning radius r_p at each trajectory terminus. The two paths diverge markedly within the first 2–3 steps, demonstrating how the choice to discard distant data fundamentally redirects the optimization dynamics. The unpruned model converges to $\mathbf{x} \approx (1.27, 1.85)$ *in 4 steps, whereas the pruned model takes a longer, misguided path to* $\mathbf{x} \approx (1.66, 2.92)$ *in 7 steps.*

The pruned trajectory deviates substantially from the unpruned path. At the final iteration, the unpruned model has incorporated all prior observations—a global historical record encoded in the posterior covariance. The pruned model, by contrast, has "forgotten" all distant data; its posterior reflects only the local neighborhood bounded by r_p (Eq. 75). This localization shifts the gradient estimate $\hat{\mathbf{g}}_t$ (Eq. 73), which alters the next step direction via line search, which changes the location $\mathbf{x}_{t+1}$ from which future observations are sampled. The optimization trajectories differ, exploring different data histories as a result of:

Loss of global information The kernel function $k(\mathbf{x}, \mathbf{x}')$ (Eq. 71) in GP regression assigns non-negligible weight to observations at distances comparable to the lengthscale ℓ. For $\ell = 0.6$ and $\alpha = 0.3$, the pruning radius is $r_p = 0.18$—substantially smaller than the effective support range of the kernel. Discarding observations at distances $\ell < d \leq r_p$ destroys information about the broader landscape. One sacrifices the ability to maintain a coherent global model in exchange for computational savings that, in practice, amount to microseconds per iteration.

Discontinuous posterior recomputation By construction, the posterior mean (Eq. 72) and gradient prediction (Eq. 73) depend on the full training set $\mathcal{D}_t$. Removing observations does not smoothly degrade the posterior—it discontinuously recomputes it on the reduced support $\mathcal{D}_t^{\text{pruned}}$ (Eq. 74). This recomputation induces jump discontinuities in predicted gradients $\hat{\mathbf{g}}_t$, leading to erratic step sizes and divergent trajectories. The coupling between the data support set and the inference rule renders the "optimization" non-stationary: the effective objective landscape shifts with each pruning event.

Feedback amplification Each pruning event occurs at a new location $\mathbf{x}_t$. If pruning removes influential historical points that carry information about distant minima or saddle points, the model misdirects the next step to an erroneous $\mathbf{x}_{t+1}$. From this new position, a fresh set of distant points become candidates for removal (those now outside radius r_p from $\mathbf{x}_{t+1}$). The erroneous step feeds into an erroneous posterior, which guides another erroneous step. Errors compound multiplicatively across iterations.

The coupling between data support (Eq. 74) and inference (Eq. 72, 73) admits no free lunch.

7.4 Variance and accuracy

Beyond the fundamental problems with pruning data within an active learning loop, the interpetation of variance in a sequential optimization process can be subject to interpretation. To demonstrate this, consider two sampling strategies, with fixed hyperparameters (Algorithm 7) and one where the hyperparameters are optimized at each step, shown in Algorithm 8.

Algorithm 7 Sequential Sampling with Frozen Hyperparameters

```
Input: objective $U(\mathbf{x})$, gradient $\nabla U(\mathbf{x})$, frozen $\boldsymbol{\theta}_0 = (\sigma_f^0, \ell^0, \sigma_n^{\mathrm{f},0}, \sigma_n^{\mathrm{d},0})$, sample path $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$
Output: predictions $\{\mu_t, \sigma_t^2\}_{t=1}^T$, RMSE and coverage metrics
Initialize: $\mathcal{D}_1 \leftarrow \{(\mathbf{x}_1, U(\mathbf{x}_1), \nabla U(\mathbf{x}_1))\}$
for $t = 1$ to $T$ do
                                                    ▷ Fit GP with fixed hyperparameters
    Fit GP to $\mathcal{D}_t$ using $\boldsymbol{\theta}_0$ (no optimization)
                                                    ▷ Predict on dense probe grid
    $\{\mu(\mathbf{z}), \sigma^2(\mathbf{z})\}_{z \in \mathrm{GRID}} \leftarrow$ GP.predict(GRID) via Eq. 72, ??
                                                    ▷ Evaluate global and split metrics
    $\mathrm{RMSE}_t \leftarrow \sqrt{\frac{1}{|\mathrm{GRID}|}\sum_{\mathbf{z}}(\mu(\mathbf{z}) - U_{\mathrm{true}}(\mathbf{z}))^2}$
    $\{\mathrm{RMSE}_{\mathrm{in},t}, \mathrm{RMSE}_{\mathrm{out},t}\} \leftarrow \mathrm{SplitBySupport}(\mu, U_{\mathrm{true}}, \mathrm{dists}, r_p = 1.5\ell^0)$
    $\mathrm{Cov}_{1\sigma,t} \leftarrow \frac{1}{|\mathrm{GRID}|}\sum_{\mathbf{z}} \mathbb{1}[|\mu(\mathbf{z}) - U_{\mathrm{true}}(\mathbf{z})| \le \sigma(\mathbf{z})]$
                                                    ▷ Accumulate next observation
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, U(\mathbf{x}_{t+1}), \nabla U(\mathbf{x}_{t+1}))\}$
end for
return all metrics for $t = 1, \ldots, T$
```

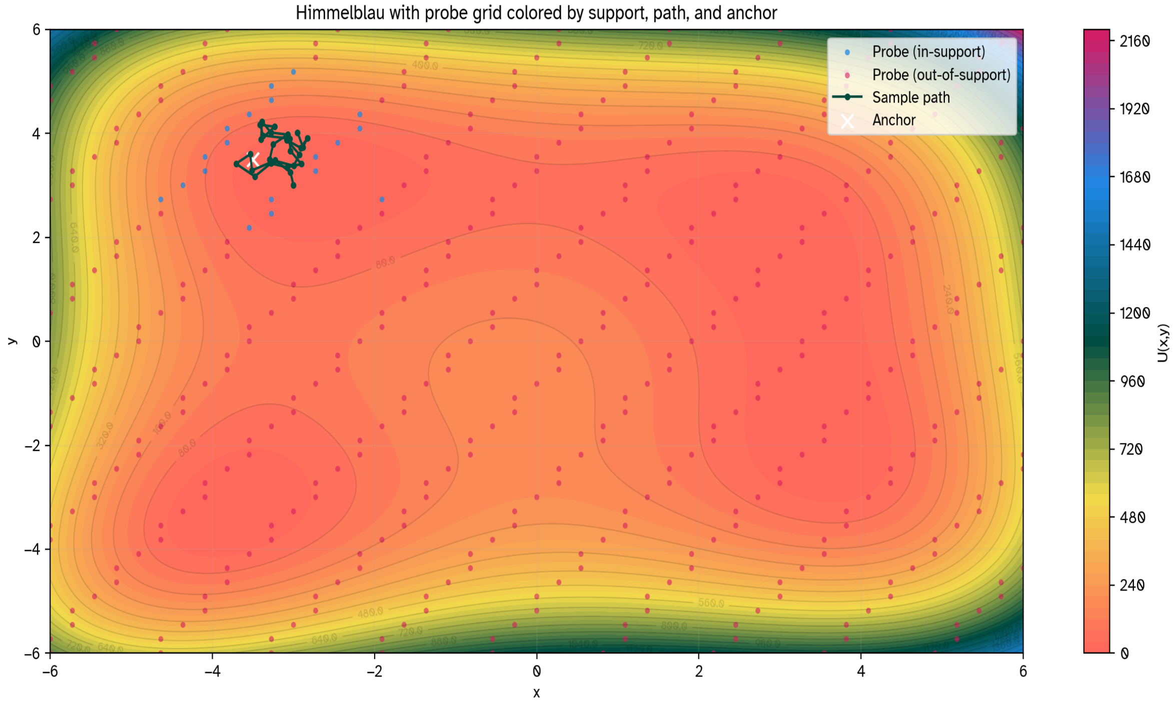


Figure 7.3. *Himmelblau surface ($T = 28$ observations). The sample path (teal line) explores the landscape in a local random walk. Probe grid points are colored: blue (in-support, within radius $r_p = 1.5\ell^*_{reopt}$) and magenta (out-of-support, beyond r_p). The anchor point (white ×) is placed within the support region, around 0.1 away from the nearest data point. The visualization reveals that most of the landscape lies out-of-support at any given iteration, a region where the re-optimized model exhibits catastrophic miscalibration.*

Algorithm 8 Sequential Sampling with Re-Optimized Hyperparameters

```
Input: objective $U(\mathbf{x})$, gradient $\nabla U(\mathbf{x})$, initial $\boldsymbol{\theta}_0$, hyperparameter ranges $\Theta$, search trials $N_{\text{trials}}$, sample path $\{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$
Output: predictions $\{\mu_t, \sigma_t^2\}_{t=1}^T$, RMSE and coverage metrics, optimized $\{\boldsymbol{\theta}_t^*\}_{t=1}^T$
Initialize: $\mathcal{D}_1 \leftarrow \{(\mathbf{x}_1, U(\mathbf{x}_1), \nabla U(\mathbf{x}_1))\}$, $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}_0$
$r_p \leftarrow 1.5\ell^0$                                  ▷ Support radius (will adapt with $\ell_t^*$)
for $t = 1$ to $T$ do
                ▷ Re-optimize hyperparameters via random search on marginal likelihood
    $\boldsymbol{\theta}^* \leftarrow \arg\max_{\boldsymbol{\theta} \in \Theta} p(\mathbf{y}|\mathcal{D}_t, \boldsymbol{\theta})$ via Eq. ?? (Eq. 78)
    Extract $\ell_t^* \leftarrow$ lengthscale$(\boldsymbol{\theta}^*)$; $r_p \leftarrow 1.5\ell_t^*$        ▷ Update support radius
                                        ▷ Fit GP with re-optimized hyperparameters
    Fit GP to $\mathcal{D}_t$ using $\boldsymbol{\theta}^*$
                                                  ▷ Predict on dense probe grid
    $\{\mu(\mathbf{z}), \sigma^2(\mathbf{z})\}_{z \in \text{GRID}} \leftarrow$ GP.predict(GRID) via Eq. 72, ??
                      ▷ Evaluate global and split metrics (using re-optimized $r_p$)
    $\text{RMSE}_t \leftarrow \sqrt{\frac{1}{|\text{GRID}|} \sum_{\mathbf{z}} (\mu(\mathbf{z}) - U_{\text{true}}(\mathbf{z}))^2}$
    $\{\text{RMSE}_{\text{in},t}, \text{RMSE}_{\text{out},t}\} \leftarrow$ SplitBySupport$(\mu, U_{\text{true}}, \text{dists}, r_p)$
    $\text{Cov}_{1\sigma,t} \leftarrow \frac{1}{|\text{GRID}|} \sum_{\mathbf{z}} \mathbb{1}[|\mu(\mathbf{z}) - U_{\text{true}}(\mathbf{z})| \leq \sigma(\mathbf{z})]$
                                                ▷ Accumulate next observation
    $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{t+1}, U(\mathbf{x}_{t+1}), \nabla U(\mathbf{x}_{t+1}))\}$
end for
return all metrics for $t = 1, \ldots, T$; $\{\boldsymbol{\theta}_t^*\}$
```

For the Himmelblau function

$$U(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{76}$$

with gradient:

$$\nabla U(\mathbf{x}) = \begin{pmatrix} 4x_1(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 4x_2(x_1 + x_2^2 - 7) \end{pmatrix} \tag{77}$$

. The derivative information is incorporated via cross-covariances between functions and gradients, scaled by a constant factor $s_d = 10$. This fixed scaling ensures that changes in the marginal likelihood stem from σ_f, ℓ, and noise parameters alone, not from coupling effects. For this section consider random search over the hyperparameter space:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} p(\mathbf{y}|\mathcal{D}_t, \boldsymbol{\theta}) \tag{78}$$

Search ranges are:

- $\ell \in [0.25, 3.0]$ (lengthscale)
- $\sigma_f \in [0.5, 5.0]$ (signal variance)
- $\sigma_n^{\text{f}} \in [10^{-5}, 10^{-1}]$ (function noise)

- $\sigma_n^{\mathrm{d}} \in [10^{-5}, 10^{-1}]$ (derivative noise)

We draw $N_{\text{trials}} = 60$ candidate hyperparameters uniformly in log-space and select the maximum. Figure 7.4 illustrates the resulting trajectories.

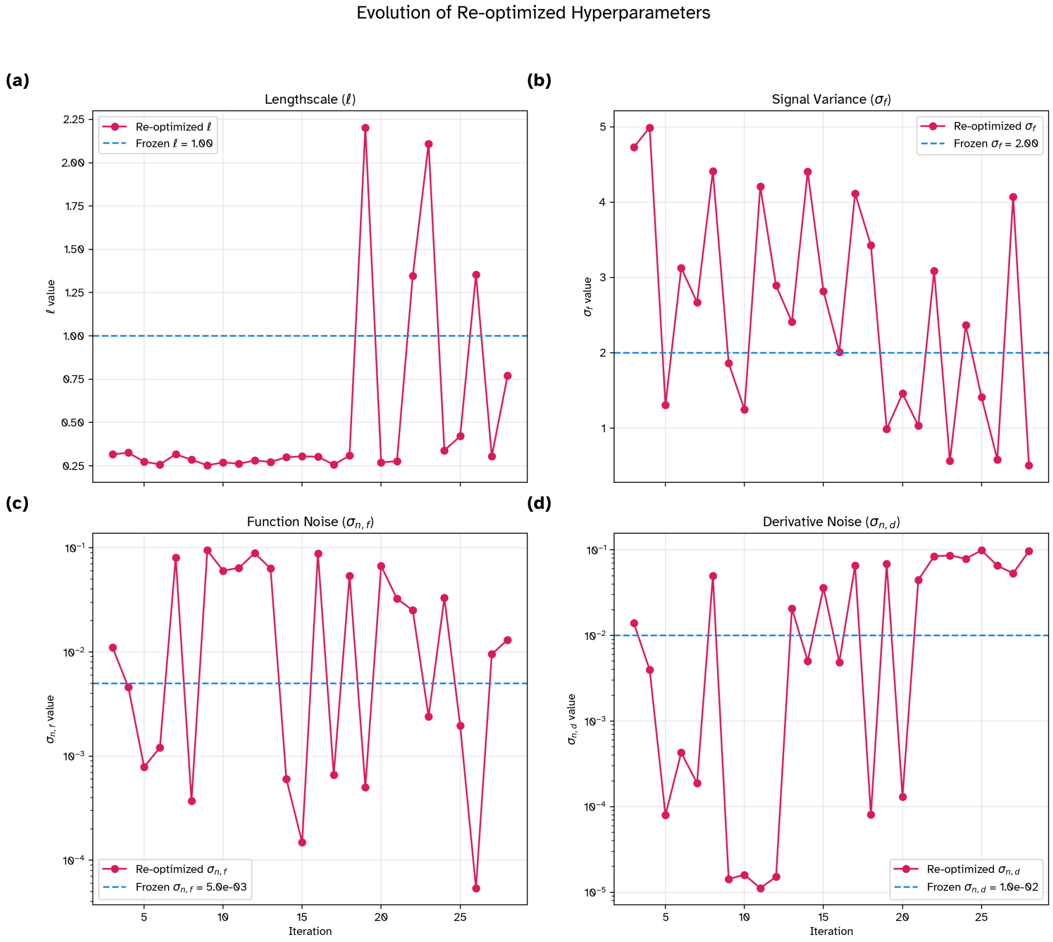


Figure 7.4. *Hyperparameter re-optimization traces. The plots show the values of the lengthscale (ℓ), signal variance (σ_f), function noise ($\sigma_{n,f}$), and derivative noise ($\sigma_{n,d}$) chosen by maximizing the marginal log-likelihood at each step. The dashed blue lines indicate the constant values used by the "Frozen θ" model. The re-optimized values, particularly for the lengthscale and signal variance, are extremely volatile. They fluctuate dramatically from one iteration to the next, indicating that the MLL optimization landscape is ill-conditioned or has multiple competing maxima, especially when trained on locally clustered data that includes derivatives.*

Figures 7.3 and 7.5 present the empirical comparison on the Himmelblau function (Eq. 76). $T = 28$ observations are taken along a random walk initialized at $\mathbf{x}_0 = (-3.0, 3.0)$, a point deep in a high-valued basin. A dense probe grid of 45×45 points covers the domain $[-6, 6]^2$.

Taken together, these results demonstrate a clear pathology in naively re-optimizing GPDimer hyperparameters within a sequential learning process, particularly when using derivative information. While re-optimization can improve global point-wise accuracy metrics, it does so by sacrificing the integrity of the model's variance estimates. The resulting model becomes

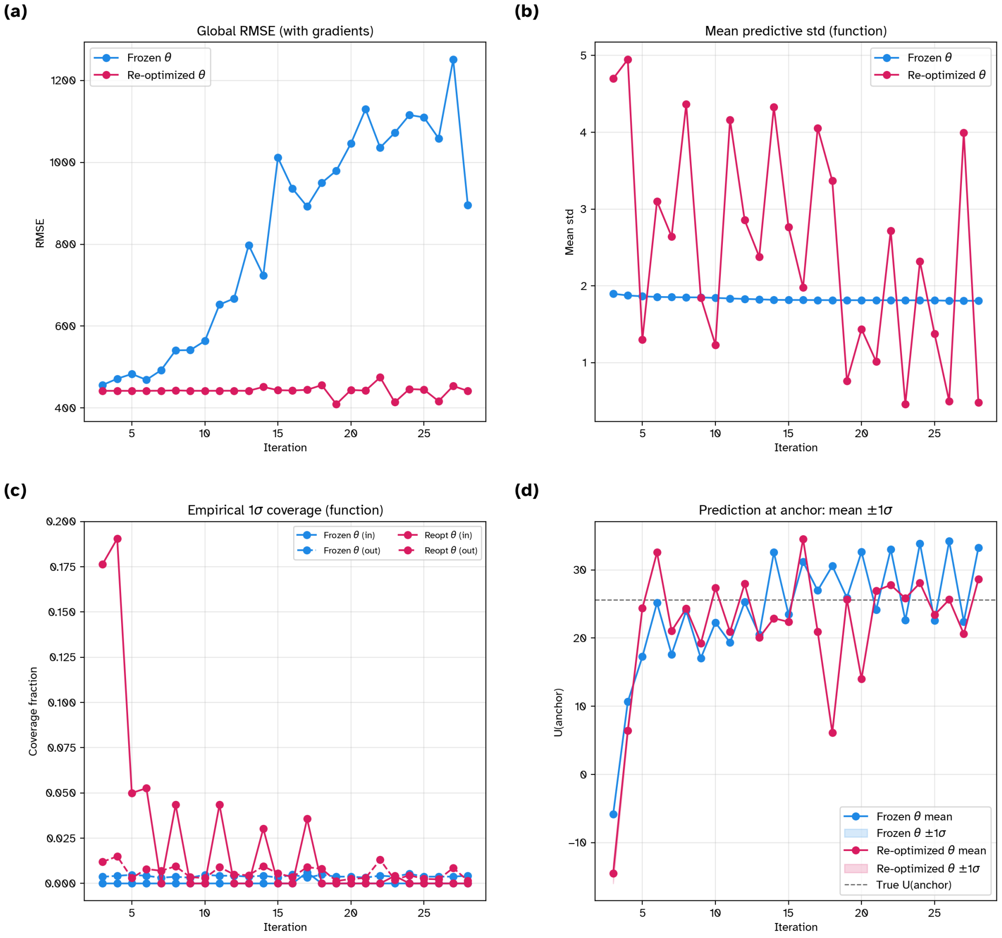


Figure 7.5. *Hyperparameter re-optimization effects on accuracy and calibration. (a) Global RMSE: The Root Mean Squared Error over the entire probe grid. The re-optimized model consistently achieves a slightly lower (better) RMSE than the frozen model, suggesting superior global accuracy.(b) Mean Predictive Standard Deviation: The average predictive uncertainty across the grid. The re-optimized model exhibits highly volatile and often significantly larger uncertainty compared to the stable uncertainty of the frozen model.(c) Empirical 1σ Coverage: The fraction of probe points where the true function value falls within the model's predicted ±1σ confidence interval. Both models show poor calibration, but the re-optimized model is particularly unreliable for out-of-support points (dashed magenta line), where its coverage fraction is frequently near zero.(d) Prediction at Anchor: The predicted mean and ±1σ confidence interval at the anchor point. The frozen model's prediction is stable and converges reasonably close to the true value (dashed black line). In stark contrast, the re-optimized model's prediction can be unstable, with both mean and uncertainty fluctuating with each new data point.*

volatile and severely overconfident, producing miscalibrated uncertainty predictions that cannot be trusted for decision-making. This reinforces the need for caution when interpreting predictive variance from models whose hyperparameters are continually adapted on growing, locally-clustered datasets.

7.5 Data driven pruning

Having established that for active learning settings, standard static data-pruning techniques are not equivalent and may hamper performance, we consider once again the ideal effect of pruning. Figure 7.6 illustrates this effect across different molecule sizes. At 150 training configurations, the pruned approach achieves a 22× speedup compared to the full block matrix for all molecule sizes, while the block matrix itself provides a 300× speedup over the full theoretical kernel. For larger molecules ($N = 30$ atoms), the full kernel computation would require on the order of 15 seconds per optimization step; the block approach reduces this to approximately 0.1 seconds, and pruning further reduces it to approximately 2 milliseconds, corresponding to a combined speedup of over 7000×.

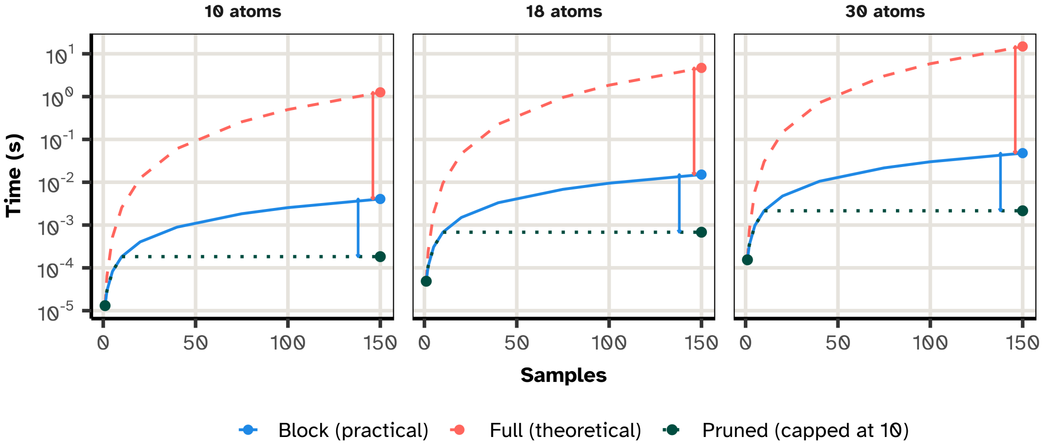


Figure 7.6. Computation time scaling with data-driven pruning. Three strategies are compared: full theoretical kernel (dashed red, $(M(3N + 1))^2$ *elements), practical block matrix (solid blue,* $M(3N + 1) \times (3N + 1)$ *elements), and pruned block matrix (dotted green, capped at 10 configurations,* $10(3N + 1) \times (3N + 1)$ *elements). Time estimates are based on benchmarks from a modern laptop (ThinkPad X1 Carbon 2021; 1538×1538 matrix inversion ~0.1 s). At 150 samples, pruning would provide consistent ~22× speedup over block scaling across all molecule sizes, with the benefit growing in absolute time for larger systems.*

7.5.1 Hyperparameter trajectories for the GPDimer

The hyperparameters in GPDimer runs are seen to stabilize after a modest amount of data [23]. To this end, we employ a local, gradient-based SCG optimizer for the hyperparameters (Alg. 4), warm-started from the converged values of the previous optimization step. This avoids the cost of a global search and leverages the fact that the Potential Energy Surface topology evolves smoothly.

[23] though these are not globally optimal like those from `optuna` [153]

Fig 7.7 demonstrates this behavior for a representative system `S000`, a 16 atom molecule C_5OH_{10} which starts from an acyclic ether with a separation of 2.2 Å between the carbon endpoints. Most of hyperparameters stabilize rapidly, which suggests that the local maximum on the likelihood surface is a function of a small subset of data. However, the signal variance (σ_f^2) exhibits fluctuations, hinting at a potential source of instability in the model. As the geometry changes, the optimizer may struggle to fit new, challenging data points, causing the variance to oscillate as an artifact of the dynamic dataset. Furthermore, the cost of optimization does not decrease monotonically; the time per step can grow even as the number of function evaluations falls, reflecting the increasing cost of matrix operations on the growing dataset.

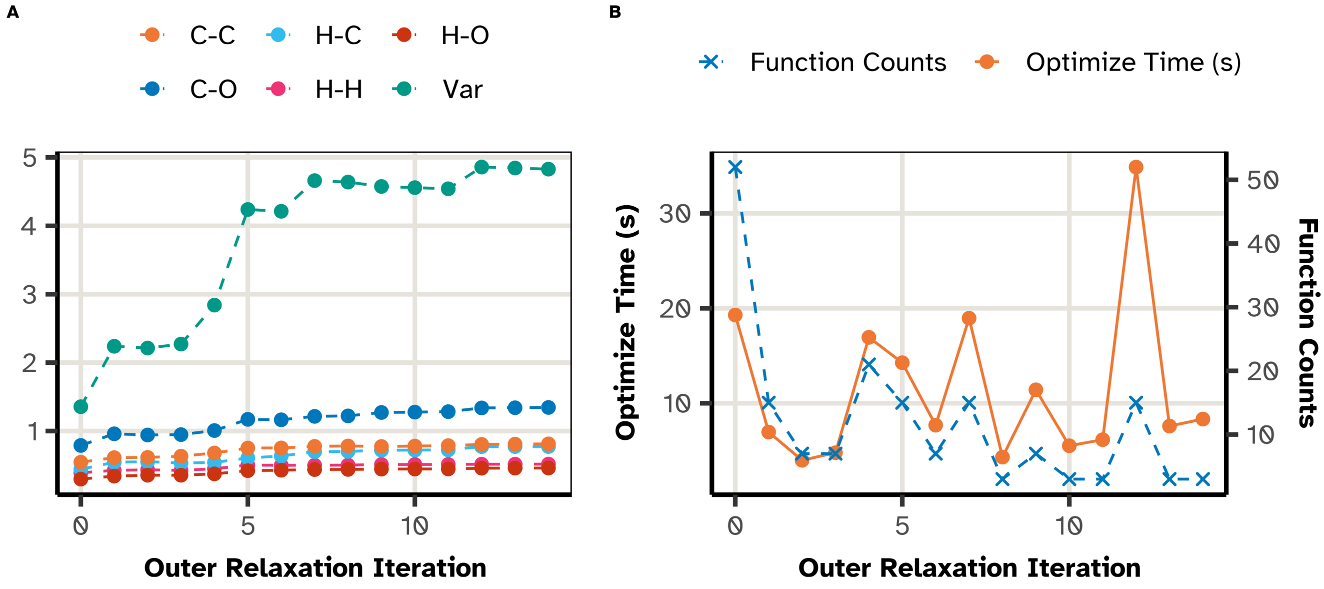


*Figure 7.7. Hyperparameter and computational cost during GPDimer. **(A)** Evolution of kernel hyperparameters for the `S000` show that lengthscales remain stable after an initial adjustment period. The signal variance fluctuates, an artifact of having to fit subsequent points. **(B)** Computational cost for the hyperparameter optimization at each relaxation loop, showing both wall time and the number of function evaluations. As the steps increase, the time taken grows even as the number of function evaluations reduce. Data from [50].*

A large signal variance causes the model to lose its physical meaning, behaving as a pure mathematical interpolator that can guide the search into unphysical regions (e.g., overlapping atoms), leading to calculation failure. To counteract this, we introduce an adaptive barrier for the signal variance, and an oscillation detection heuristic.

7.5.2 Adaptive Barrier for Signal Variance

To directly prevent the pathological growth of the signal variance, we augment the marginal log-likelihood (MLL) objective function with a logarithmic barrier:

$$\mathcal{L}_{\text{eff}}(\boldsymbol{\theta}) = \underbrace{\log p(\mathbf{y} \mid \mathcal{S}, \boldsymbol{\theta})}_{\text{MLL}} - \mu \log\left(\lambda_{\max} - \log \sigma_f^2\right) \tag{79}$$

where $\lambda_{\max}$ fixes an absolute upper bound for $\log \sigma_f^2$, and the barrier strength, μ, grows linearly with the number of collected data points, N:

$$\mu(N) = \mu_0 + \alpha N, \qquad \mu(N) \leq \mu_{\max} \tag{80}$$

This schedule allows the model to remain flexible when data is sparse (small μ) but enforces an increasingly strict bound as the dataset matures and the model should have settled on a physically reasonable amplitude. This adaptive behavior eliminates pathological variance growth while preserving the surrogate's ability to capture the true curvature of the PES, as seen in Figure **??** (B).

7.5.3 Hyperparameter Oscillation Detection

Re-optimizing hyperparameters on a dynamically changing subset of data can lead to unstable estimates that oscillate between iterations as shown in previous sections. The Hyperparameter Oscillation Detection (HOD) heuristic monitors these fluctuations over a moving window of the last W steps. We define an oscillation indicator, $O_j(t)$, for each hyperparameter θ_j at step t:

$$O_j(t) = \begin{cases} 1 & \text{if } \operatorname{sgn}\left[\Delta\theta_j(t-1)\right] \neq \operatorname{sgn}\left[\Delta\theta_j(t-2)\right], \\ 0 & \text{otherwise.} \end{cases} \tag{81}$$

where $\Delta\theta_j(t) = \theta_j(t) - \theta_j(t-1)$. If the total fraction of oscillations, f_{osc}, across all hyperparameters in the window exceeds a threshold, p_{osc}, the optimization is flagged as unstable. In response, the algorithm automatically enlarges the subset of data used for fitting, which improves the conditioning of the covariance matrix and typically results in a smoother, more stable MLL surface.

7.6 Conclusions

This chapter demonstrated that scaling Gaussian Process methodologies for high-performance chemistry requires a synthesis of algebraic restructuring and careful statistical approximation. We established that matrix reshaping into block-diagonal forms reduces memory complexity to linear scaling without sacrificing mathematical fidelity. However, the wall-time cost of inverting these matrices during iterative hyperparameter optimization remains the primary computational bottleneck.

Our investigation into naive data pruning revealed severe pathologies; coupling the data support set directly to the inference engine creates feedback loops that drive optimization trajectories toward erroneous local minima. Furthermore, aggressive hyperparameter re-optimization on dynamic local subsets sacrifices variance calibration for marginal gains in point-wise accuracy, rendering the model overconfident and unsuitable for autonomous decision-making.

To mitigate these risks while capturing the immense speedups offered by sparse kernels—exceeding 7000× for larger systems we introduced essential stability mechanisms. The adaptive signal variance barrier and Hyperparameter Oscillation Detection (HOD) prevent the unphysical divergence of the model during the volatile early stages of learning. These safeguards permit the use of reduced datasets for the hyperparameter optimization, yet the specific criterion for which data to retain remains critical. Simple Euclidean metrics fail to capture chemical similarity effectively across diverse landscapes. Consequently, the final component of this framework demands a rigorous, chemically transferable metric for point selection. Chapter 8 addresses this need through the introduction of the OTGPD, replacing geometric distance with Optimal Transport metrics to guide both pruning and convergence.

8 Optimal Transport Gaussian Process

Every method is somewhere between
random searches and gradient descent.

Hannes Jónsson
Discussion with Rohit Goswami

> This chapter is based on Rohit Goswami and Hannes Jónsson. "Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches." In: *ChemPhysChem* (Nov. 2025). DOI: `10.1002/cphc.202500730`

So far, we've demonstrated state of the art performance for the GPDimer, along with more principled measures of measuring performance, and possible prescriptions for data efficiency. Proxy-based approaches remain attractive because they avoid repeated electronic-structure calls, yet several of the reported trajectories contain chemically implausible features. While the Gaussian Process is a data driven surrogate the difference between running towards abstract high dimensional neural networks and the Cartesian representation is the belief that there is still an interpretation. We note for instance, that the GPDimer on several systems fails to constrain the generated surrogate surface and thus ends up exploring pathologically unstable regions of phase space corresponding to "cold fusion", shown in Figure 8.1. It is relevant to note that these systems are not atypical in any form, there are several similar hydrogen abstraction reactions which succeed.

To better understand these failure modes we recall the spline view of a Gaussian Process [154]. With a fixed kernel k and observation noise σ_n^2, the GP posterior mean is the function that balances fit and roughness in the reproducing-kernel Hilbert space (RKHS):

$$\hat{f} \;=\; \arg\min_{f \in \mathcal{H}_k} \; \frac{1}{\sigma_n^2} \sum_i (y_i - f(x_i))^2 \;+\; \|f\|^2_{\mathcal{H}_k}, \tag{82}$$

where kernel hyperparameters (length scales, signal variance) control what counts as "roughness" and how it is penalized. In practice, hyperparameters $\boldsymbol{\theta}$ are chosen statistically by maximizing the MLL on the training set $\mathcal{S}$,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\mathbf{y}|\mathcal{S}, \boldsymbol{\theta}), \tag{83}$$

rather than by minimizing an inaccessible physical discrepancy to the unknown Potential Energy Surface,

$$\boldsymbol{\theta}_{\text{ideal}} = \arg\min_{\boldsymbol{\theta}} \int |f(x; \boldsymbol{\theta}) - V(x)|^2 \, dx. \tag{84}$$

The Gaussian Process, defined in terms of finite realizations of multivariate normal distributions, expressed as a "function", or a series of x, y pairs reshaped to provide familiarity with 3D

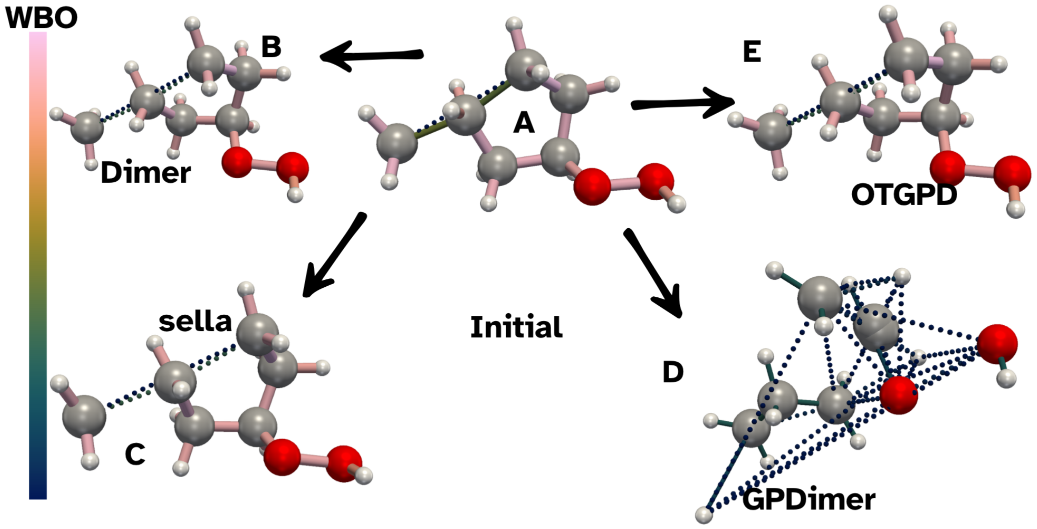


Figure 8.1. *A comparison of saddle point search trajectories for a ethoxy radical hydrogen abstraction reaction,* `doublet_150` *reaction starting from an initial configuration (A). The standard Dimer method (B), Sella (C), and the OTGPD (E) follow a chemically intuitive path. The previous GPDimer method (D) is guided towards a fractured state, leading to failures in the underlying NWChem calculator.*

matrices; thus suffers greatly from anthropomorphism of the constituent equations. The physical meaning ascribed to the hyperparameters are unjustified, and models will only be guaranteed to interpolate in the noise free regime [24]. The repeated re-optimization also doesn't preserve global accuracy as shown earlier. This distinction becomes decisive in actively learned saddle-point searches, which produce correlated trajectories of geometries which are atypical from most of the energy surface geometrically. Under such data, the MLL surface can be shallow in directions like the signal variance, encouraging variance blow-up that flattens the mean and inflates predictive uncertainty, which contributes significantly to the destabilization of the GPDimer. From a data efficiency perspective, what was covered in section 7 has even greater significance in terms of reliability. However, we still require a reasonable measure of distance to complete algorithm developed in this chapter to make good on wall time performance, which we summarize in Fig. 8.2.

8.1 Intensive EMD

Before defining the distance measure, we revisit the extant distance metrics within the methodology outlined in Section 2. We start with the guardrails on the GPDimer as formulated in Chapter 5 previous chapter and in the literature [49]. There are two, one on the interatomic distances of a given configuration, and one on the distance from a known point.

[24] also known to be unstable [45]

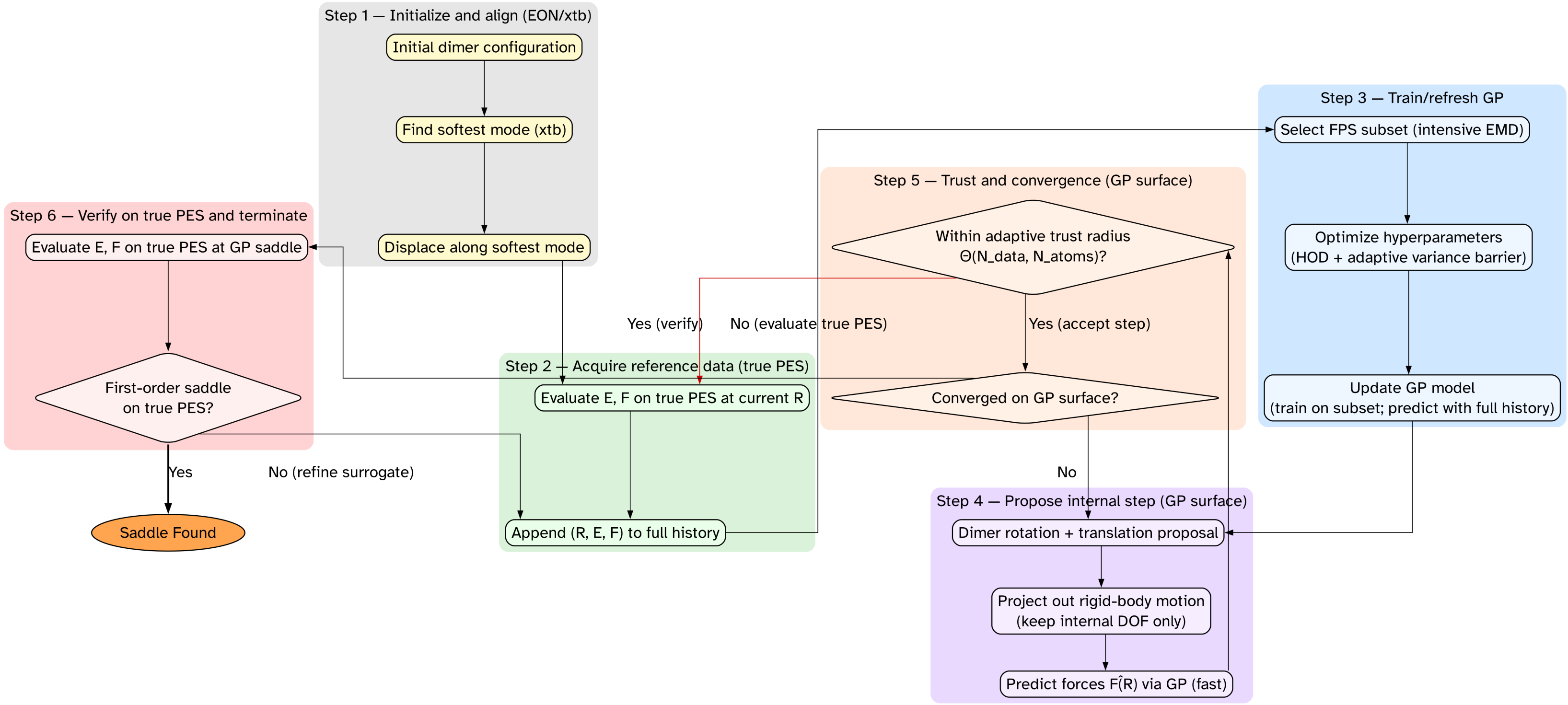

Step 1 — Initialize and align (EON/xtb)
Initial dimer configuration
Find softest mode (xtb)
Displace along softest mode
Step 2 — Acquire reference data (true PES)
Evaluate E, F on true PES at current R
Append (R, E, F) to full history
Step 3 — Train/refresh GP
Select FPS subset (intensive EMD)
Optimize hyperparameters
(HOD + adaptive variance barrier)
Update GP model
(train on subset; predict with full history)
Step 4 — Propose internal step (GP surface)
Dimer rotation + translation proposal
Project out rigid-body motion
(keep internal DOF only)
Predict forces F̂(R) via GP (fast)
Step 5 — Trust and convergence (GP surface)
Within adaptive trust radius
Θ(N_data, N_atoms)?
Yes (verify)
No (evaluate true PES)
Yes (accept step)
Converged on GP surface?
No
Step 6 — Verify on true PES and terminate
Evaluate E, F on true PES at GP saddle
First-order saddle
on true PES?
Yes
No (refine surrogate)
Saddle Found

Figure 8.2. *Optimal Transport Gaussian Process Regression framework applied to the dimer method. The algorithm begins with initialization (Step 1, grey) and acquires an initial reference point from the true PES (Step 2, green) to train the GP model (Step 3, blue). An internal, computationally 'cheap' optimization loop (Steps 4 and 5, purple/orange) then searches for a saddle point candidate on the surrogate GP surface. This internal search is governed by an adaptive trust radius (Step 5) to ensure reliability. Calls to the "expensive" calculator (Step 2) are only triggered when necessary: either when the optimizer moves outside the trusted region or after the internal optimization on the GP surface has converged. This new data is used to update and refine the GP model (Step 3). Once the entire process converges, the final candidate structure is verified on the true PES (Step 6, red) to confirm it is a valid first-order saddle point before the algorithm terminates.*

An inequality expresses the measure in the literature [49]:

$$\frac{2}{3} r_{ij}(\mathbf{x}_{\text{eval}}) < r_{ij}(\mathbf{x}_{\text{im}}) < \frac{3}{2} r_{ij}(\mathbf{x}_{\text{eval}}) \tag{85}$$

$$\Longrightarrow \left| \log \frac{r_{ij}(\mathbf{x}_{\text{im}})}{r_{ij}(\mathbf{x}_{\text{eval}})} \right| < \log(1.5) \approx 0.405 \tag{86}$$

The core of the problem is that the inverse distance we use in 26 is not invariant to the permutation of identical atoms, and since kernel's value depends on a direct, index-wise comparison of the interatomic distance vectors of two configurations. This creates a dependency on the arbitrary, fixed labels of the atoms, rather than their physical roles.

An easy way to understand this stems from observing symmetric systems. For instance, consider a proton (indexed k) transferring between two chemically equivalent sites (m and n). Physically, the initial and final states are energetically degenerate. However, a fixed-index comparison metric perceives a significant geometric change, as the distance r(k,m) transitions from short to long, while r(k,n) simultaneously transitions from long to short. The metric fails to recognize that the permutation of labels would reconcile the apparent structural difference.

While the kernel's fitted length-scale hyperparameter may partially average out this effect, a non-averaged metric for early stopping feels the full impact of the flaw. The 1D max log distance, by its definition, registers a significant, non-physical distance for this symmetric swap:

$$D_{\text{1Dmaxlog}}(\mathbf{x}_1, \mathbf{x}_2) = \max_{i,j} \left| \log \frac{r_{ij}(\mathbf{x}_2)}{r_{ij}(\mathbf{x}_1)} \right| \tag{87}$$

This sensitivity to labeling motivates using the intensive EMD. Figure 8.3 demonstrates this, by contrasting the behavior of both metrics for the asymmetric stretching of a water molecule.

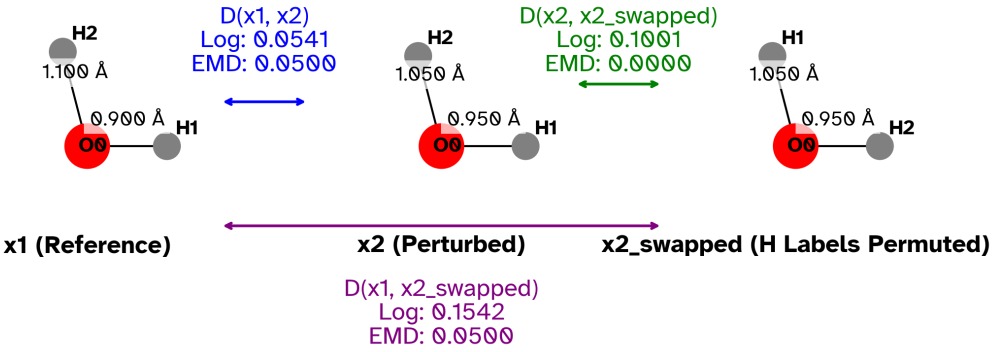


Figure 8.3. *Comparison of the 1D max log distance and the Earth Mover's Distance (EMD) for an asymmetric stretch of a water molecule. While configuration x_2 and $x_{2,s}$ are physically identical (differing only by the permutation of hydrogen atom labels), the 1D max log metric incorrectly assigns a large distance between them and the reference x_1. In contrast, the EMD correctly identifies them as being equidistant from the reference, demonstrating its permutational invariance.*

So we would preferably have a measure which is invariant to both permutations of the labels, and not be a property which grows with the system size. Optimal transport theory [155], more precisely the EMD metric which solves a linear assignment problem, often the Hungarian [117] to ensure minimal motion required to deform one discrete distribution into another. However, rather than introducing mass weighing of graphs, we opt to instead solve the EMD for each atom type, thus we solve a "colored" EMD, divided by the number of atoms to scale intuitively:

$$\bar{d}_t = \frac{1}{N_t} \min_{\pi \in \Pi_{N_t}} \sum_{k=1}^{N_t} \left\| \mathbf{r}_{k,t}^{(1)} - \mathbf{r}_{\pi(k),t}^{(2)} \right\|. \tag{88}$$

Here, N_t denotes the number of atoms of type t and Π_{N_t} the set of all permutations of the N_t indices. We then identify the largest per-type average displacement as the overall distance:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \max_t \bar{d}_t(\mathbf{x}_i, \mathbf{x}_j). \tag{89}$$

Because each $\bar{d}_t$ averages over the atoms of a particular element, it forms an intensive quantity that reflects the collective motion of a specific chemical group. Adding spectator atoms does not dilute the metric, which makes it an ideal measure for selecting a chemically diverse subset.

8.2 Adaptive trust radius

With this new distance metric, we now re-state the trust region formulation, with a few additional notes.

While our surrogate model accelerates the search for saddle points, its reliability is confined to regions of the potential energy surface where it has been trained. To prevent the algorithm from taking overly ambitious steps into uncharted territory where the surrogate's predictions may be inaccurate, we introduce a dynamic "trust radius." This mechanism acts as an intelligent guardrail, ensuring that any proposed step remains within a zone of confidence defined by the existing data.

The core of this guardrail is a simple condition. We measure the distance, using our permutationally-invariant Earth Mover's Distance (EMD), between any new candidate configuration ($\mathbf{x}_{\text{cand}}$) and its nearest neighbor ($\mathbf{x}_{\text{nn}}$) in the current training set. This step is only accepted if the distance is within an adaptive threshold, Θ:

$$d_{\text{EMD}}\left(\mathbf{x}_{\text{cand}}, \mathbf{x}_{\text{nn}}\right) \leq \Theta\left(N_{\text{data}}, N_{\text{atoms}}\right) \tag{90}$$

This threshold, Θ, is not static; it evolves as the surrogate model gathers more information. We designed its functional form to follow an exponential saturation curve, allowing the model to become more adventurous as its knowledge base grows. This "earned trust" radius is defined as:

$$\Theta_{\text{earned}}\left(N_{\text{data}}\right) = T_{\text{min}} + \Delta T_{\text{explore}} \cdot \left(1 - e^{-k\, N_{\text{data}}}\right) \tag{91}$$

Here, T_{min} provides a minimal safe radius to prevent trivially small steps, while $\Delta T_{\text{explore}}$ sets the maximum additional exploration distance the algorithm can earn. The rate of this expansion is controlled by k, which is linked to N_{half}, the number of data points needed for the threshold to reach half of its maximum value.

However, to ensure physical realism, we impose a hard ceiling on this trust radius that prevents it from becoming unphysically large, regardless of the amount of data collected. This ceiling is dependent on the size of the system:

$$\Theta_{\mathrm{phys}}(N_{\mathrm{atoms}}) = \max\Big(a_{\mathrm{floor}},\ \frac{a_A}{\sqrt{N_{\mathrm{atoms}}}} \Big) \tag{92}$$

with a_{floor} a user-defined lower bound and a_A a scaling constant.

The final, operational trust radius is simply the more restrictive of these two bounds: the one the model has earned through data collection and the one imposed by physical constraints.

$$\Theta(N_{\mathrm{data}}, N_{\mathrm{atoms}}) = \min\big(\Theta_{\mathrm{earned}}(N_{\mathrm{data}}),\ \Theta_{\mathrm{phys}}(N_{\mathrm{atoms}})\big) \tag{93}$$

If a proposed step violates the trust radius in Eq. 90, the algorithm intelligently recognizes a gap in its knowledge. It rejects the step and instead evaluates the energy at that very point of failure. This new data point is then added to the training set, and the trust radius is recomputed. This process of targeted data acquisition actively and efficiently improves the surrogate model precisely where it proves to be unreliable, ensuring our search remains both bold in its exploration and grounded in the reality of the potential energy surface. While it is nice to have principled and ultimately relatable distances, the EMD can be used for much more, in particular, to help with numerical stability.

8.3 Numerical conditioning for Gaussian Processes

We can estimate conditioning of the joint energy–force covariance using Gershgorin's Circle Theorem [156]. For a real, symmetric covariance matrix to be positive semi-definite (PSD), all its eigenvalues must be non-negative. Gershgorin's theorem provides bounds on these eigenvalues, stating that each eigenvalue must lie within at least one of the intervals (Gershgorin discs) defined by:

$$\lambda \in \bigcup_{i=1}^{n} [\, K_{ii} - R_i,\ K_{ii} + R_i \,], \quad R_i := \sum_{j \neq i} |K_{ij}|. \tag{94}$$

This yields a lower bound on the smallest eigenvalue:

$$\lambda_{\min}(\mathbf{K}) \geq \min_i \left(K_{ii} - R_i\right). \tag{95}$$

For block-structured energy–force kernels, a tighter bound groups terms by configuration [157]:

$$\lambda_{\min}(\mathbf{K}) \geq \min_i \left\{ \lambda_{\min}(\mathbf{K}_{ii}) - \sum_{j \neq i} \|\mathbf{K}_{ij}\|_2 \right\}, \tag{96}$$

In practice, the diagonal terms reflect the signal variance, constant offset, and noise:

$$K_{EE}(\mathbf{r}, \mathbf{r}) = \sigma_c^2 + \sigma_f^2 + \sigma_{n,E}^2, \tag{97}$$

while the force diagonal contains a metric-dependent second derivative and noise:

$$\left(\mathbf{K}_{FF}(\mathbf{r},\mathbf{r})\right)_{(i,d),(i,d)} = -\frac{\sigma_f^2}{2} \left. \frac{\partial^2 \mathcal{D}^2}{\partial x_{i,d}\, \partial x'_{i,d}} \right|_{\mathbf{x}'=\mathbf{x}=\mathbf{r}} + \sigma_{n,F}^2. \tag{98}$$

When sampled configurations cluster too closely in geometry, the off-diagonal terms R_i grow large, and the Gershgorin bound can become small or negative—signalling poor conditioning and risk of numerical instability. Notably, the signal variance σ_f^2 cancels out in the simple diagonal dominance test, indicating that configuration geometry and length scale govern stability, not the variance alone [158].

8.4 Farthest point sampling

The upshot of the analysis in the previous section reveals that without sufficient geometric diversity, surrogate models risk numerical instability, manifesting as failed saddle searches or unphysical predictions. To mitigate this, we employ Farthest point sampling (FPS), which systematically selects new configurations that maximize their separation from the existing set. FPS directly suppresses the magnitude of off-diagonal covariance terms, shrinking the Gershgorin radii and helps maintain the diagonal dominance needed for stable and physically meaningful surrogate surfaces.

Figure 8.4 demonstrates the practical advantage of this approach in the `singlet_016` system, where FPS and adaptive variance control enable robust and efficient saddle searches, in contrast to the instabilities observed in standard GPDimer runs, while also mitigating the effect of the size of the hyperparameter matrix optimization, which is the primary walltime bottleneck.

However, as demonstrated in the previous section, FPS is not merely an efficiency improvements; but an essential countermeasure to manage the numerical instabilities. By construction, FPS selects new data points that are maximally distant from the existing set. This strategy directly suppresses the magnitude of the off-diagonal covariance terms, systematically shrinking the Gershgorin radii (R_i). In doing so, FPS actively enforces the diagonal dominance ($D_{ii} > R_i$) required for a numerically stable and physically reliable surrogate model.

8.5 Variance control and Hyperparameter stability

Figure 8.4 also shows a sudden jump in the hyperparameters, in particular, the variance. The effect of this hyperparameter is to basically make the model pathologically sensitive to the data points, which in turn will sample configurations which crash NWChem. To counteract this instability and ensure robust performance, we implement two complementary mechanisms discussed in chapter 7: a direct control on the signal variance via an adaptive barrier, and a general heuristic for monitoring the stability of all hyperparameters. The adaptive behavior eliminates pathological variance growth while preserving the surrogate's ability to capture the true curvature of the Potential Energy Surface. Empirically, the combination of the adaptive barrier on σ_f^2 and the general HOD mechanism provides robust control. The barrier acts as a targeted preventative measure against a known failure mode, while the HOD acts as a general

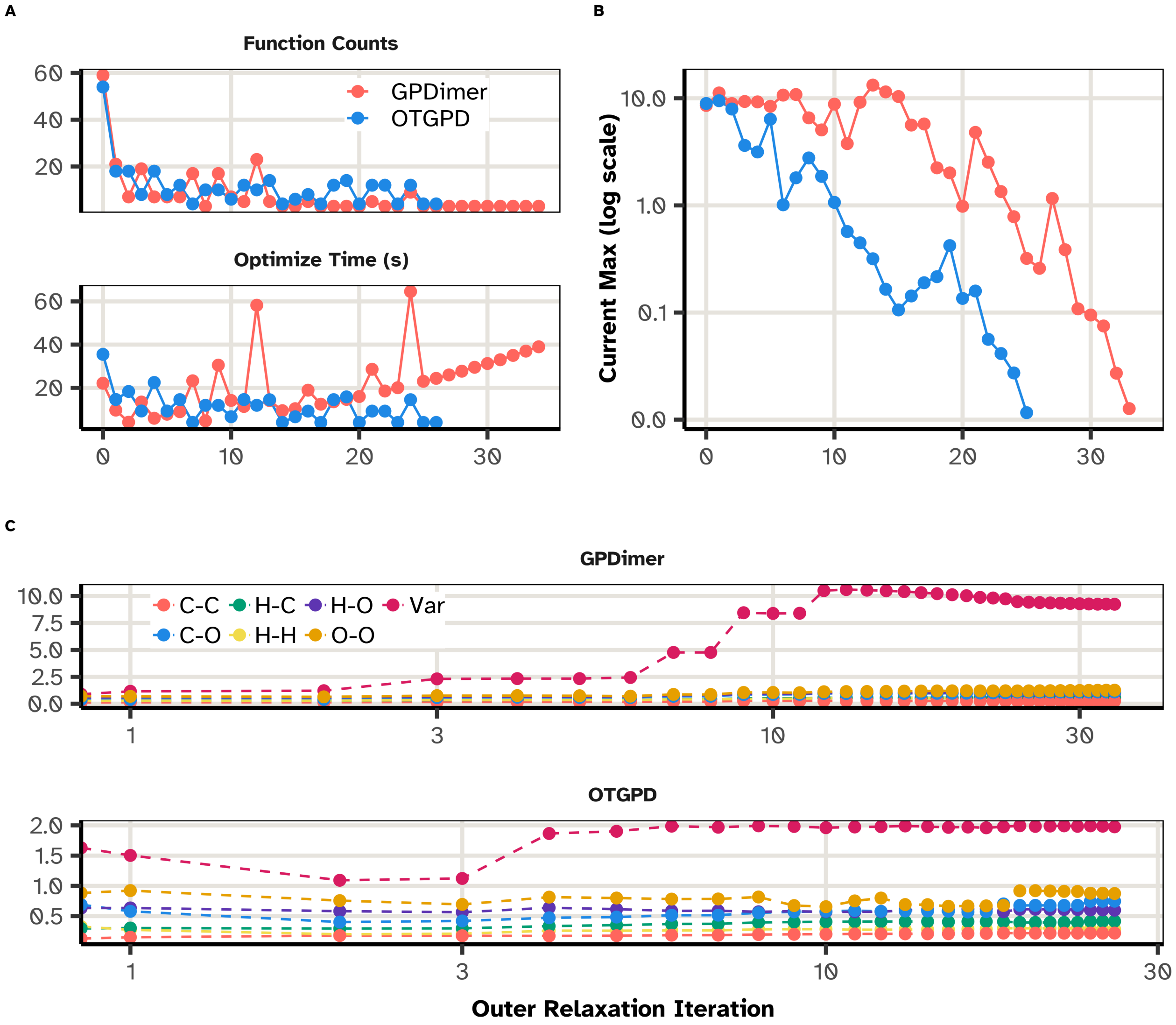


Figure 8.4. *Performance trace for the* `singlet_016` *system (Figure fig:equiv:optgd), illustrating the comparative behavior of GPDimer and OTGPD during saddle search optimization. (A) The per-iteration electronic structure function counts and wall time show that OTGPD (skyblue) consistently achieves lower and more stable computational cost per iteration compared to GPDimer (coral), which exhibits pronounced spikes and variability. (B) Convergence profiles of the maximum force component (log scale) demonstrate smoother and more rapid relaxation for OTGPD, while GPDimer progress stalls intermittently, reflecting underlying model instability. (C) Evolution of key hyperparameters over the course of the optimization, with the GP signal variance (magenta, Var) and interatomic distances (C-C, H-C, H-O, C-O, H-H, O-O) tracked for both methods. GPDimer displays episodes of pathological variance growth, coinciding with force and runtime spikes, whereas OTGPD maintains stable and physically reasonable hyperparameter values throughout.*

safety net for the entire optimization process. Together, they reduce the number of failed saddle searches from roughly twelve percent to two percent, contributing significantly to the overall efficiency and reliability of the OTGPD method.

8.6 Results

We consider systems which have less than three fragments, and any calculation exceeding 240 minutes or which lead to NWChem failures or termination conditions other than success in EON are considered to be failed.

8.6.1 Reliability

OTGPD demonstrates superior robustness compared to existing dimer-based methods, even considering the strange selection of data from Section 5.3, as illustrated in Figure 8.5. While all three approaches achieve comparable baseline success rates as shown in Table 8.1, the critical distinction emerges in systems where only one method succeeds.

Table 8.1. *Success rates on systems of up-to two fragments*

method	$\text{num}_{\text{fragments}}$	n	success
Dimer	1	26	92.3
Dimer	2	212	96.7
GPDimer	1	26	100
GPDimer	2	212	96.2
OTGPD	1	26	100
OTGPD	2	212	97.6

The comparison reveals that OTGPD's advantage lies not in marginal improvements to the baseline success rate, but in handling systems where conventional methods fail. Against GPDimer, OTGPD uniquely succeeds on 11 systems while GPDimer uniquely succeeds on only 3 (a 3.7× advantage). Against standard Dimer, OTGPD uniquely succeeds on 9 systems compared to Dimer's 4 (a 2.3× advantage). Conversely, cases where OTGPD alone fails are rare: only 2 systems fail exclusively to OTGPD versus GPDimer, and only 1 system fails exclusively to OTGPD versus Dimer.

The fact that the OTGPD captures difficult cases that other methods miss, while rarely failing alone—demonstrates that Gaussian Process acceleration can provide genuine robustness rather than simply shifting failure patterns. The method's ability to navigate challenging optimization landscapes translates to practical reliability improvements for automated saddle point discovery workflows.

8.6.2 Identifying failure modes

The OTGPD framework successfully eliminates the signal variance instability that plagued earlier iterations. However, our benchmarking revealed four specific systems—D016, D084, D100, and S242—where the OTGPD failed to converge, despite success in the GPDimer baseline. A forensic analysis of these trajectories indicates that these failures stem not from an intrinsic algorithmic instability in the OTGPD, but rather from artifacts introduced by the

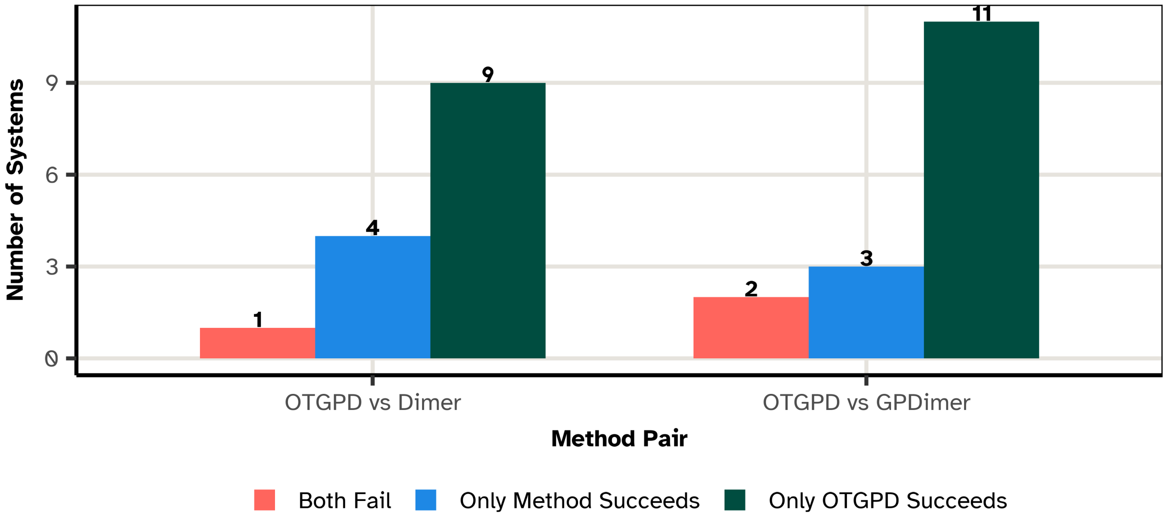


Figure 8.5. *Reliability comparison of OTGPD against GPDimer and standard Dimer methods across 238 molecular systems. A calculation exceeding 240 minutes or raising an error in the electronic structure calculation counts as a failure. The bar chart shows the distribution of outcomes for each pairwise comparison: (red) systems where both methods fail, (blue) systems where only the alternative method succeeds, and (green) systems where only OTGPD succeeds. OTGPD uniquely finds the saddle point for 11 additional systems (4.6%) compared to GPDimer and 9 additional systems (3.8%) compared to standard Dimer, demonstrating measurable advantages in challenging cases.*

dimer initialization routine.

For the OTGPD benchmark, we generated initial dimer configurations by displacing atoms along the softest mode identified by an inexpensive semi-empirical (GFN-xTB) calculation. In these four specific cases, this procedure produced pathological starting geometries characterized by excessively high forces or unphysical atomic overlaps. Figure 8.6 contrasts these starting points. For example, system D016 initialized with unphysically close carbon atoms, while S242 erroneously replaced a hydrogen on the middle carbon with a methyl end-group.

When a GP-accelerated method receives such a “poisoned” baseline—a high-energy, high-force configuration—the initial surrogate model incorporates this unphysical data. The Gaussian Process effectively learns that these extreme forces are characteristic of the landscape, leading to a warped posterior that guides the search into unstable regions. The GPDimer benchmark avoided this specific pathology by utilizing a different initialization strategy based on displacing the least coordinated atom.

To verify this diagnosis, we re-executed the OTGPD searches for these four systems using the valid, non-pathological initial configurations from the GPDimer runs. Table 8.2 presents the results. In every case, the OTGPD successfully converged. Furthermore, it retained its performance advantage: for the challenging `D084` system, OTGPD required only 3.0 minutes (181 s) compared to 6.95 minutes (417 s) for GPDimer and 4.25 minutes (255 s) for the standard Dimer. This confirms that the observed “failures” were initialization artifacts rather than deficiencies in the optimal transport framework.

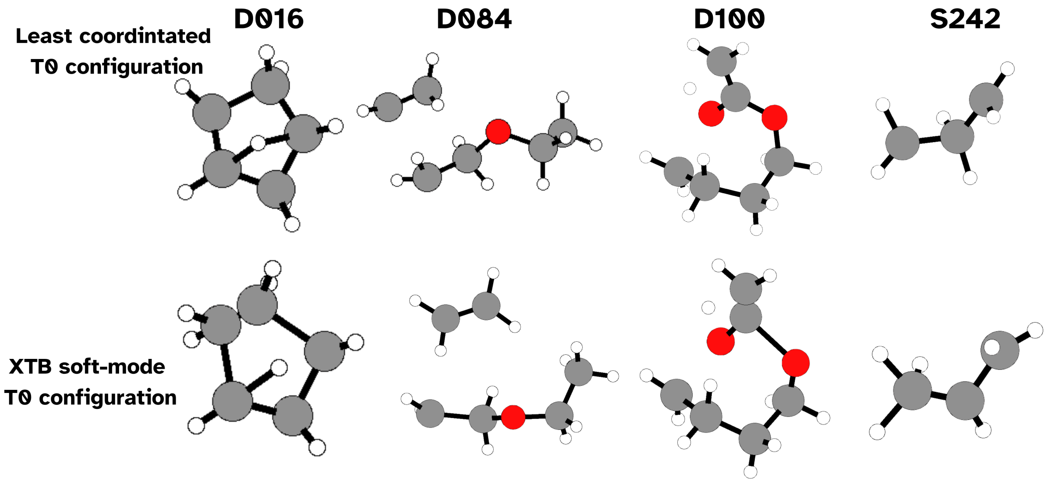


Figure 8.6. *Initializations for the GPDimer (top) and OTGPD (bottom). The xTB initialization procedure in these four cases results in unphysical initial geometries, leading to failures in the optimization routine. Specifically,* `D016` *exhibits unphysically close carbon atoms,* `D084` *possesses a shortened carbon-oxygen bond,* `D100` *shows near-overlapping carbon atoms, and* `S242` *features a misplaced methyl group.*

Table 8.2. *Performance metrics for re-run systems (D016, D084, D100, S242) using identical initial configurations. All methods converge successfully. Times reported in minutes. The OTGPD maintains superior or competitive efficiency in both electronic structure calls and total wall time.*

System	Method	HF Calls	Time (min)
D016	Dimer	106	7.4
	GPDimer	20	3.7
	OTGPD	20	3.3
D084	Dimer	2666	255.4
	GPDimer	75	417
	OTGPD	65	181.4
D100	Dimer	214	24.3
	GPDimer	28	16.2
	OTGPD	28	18.52
S242	Dimer	249	14.6
	GPDimer	25	2.3
	OTGPD	33	1.85

8.6.3 Linear bending angles and Sella

When the optimisation proceeds in Cartesian space we retain a clear mapping between the optimisation variables and the molecular geometry, which permits a post-hoc assessment of whether a reported saddle point corresponds to a chemically meaningful transition state. Systems like `singlet_016` in Figure 8.7 clearly show a wide range of saddles connected to the same initial state, one for each method, each of which are valid saddle points from a mathematical perspective, but as shown in Section 5.5 only some correspond to physically relevant reaction pathways.

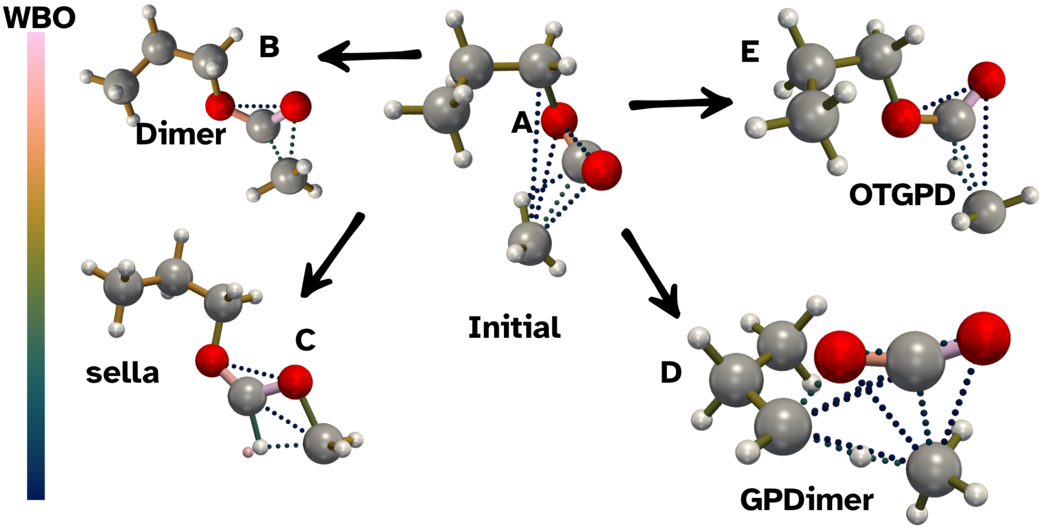


Figure 8.7. Endpoints for saddle point search trajectories of the `singlet_016` system starting from an initial configuration (A). The standard Dimer method (B) and the proposed OTGPD method (E) identifies the nearest transition state structure. The previous GPDimer method (D) and Sella (C) are guided towards a much more fractured state.

We compare the performance of the OTGPD and Sella algorithms on n-propyl acetate ($C_5H_{10}O_2$) in Figure 8.8 . Both search methods commence from the same initial geometry, a structure that subsequent optimization confirms does not represent a stable minimum on the potential energy surface. The initial structure's softest vibrational mode corresponds to a low-energy torsional motion of the hydrogens on the terminal methyl group (C5).

The OTGPD method converges upon a proximal saddle point in just 36 PES evaluations. This transition state corresponds to homolytic cleavage of the central C4–O6 ester bond, with a barrier height of 2.3 eV and geometric proximity to the initial structure (RMSD = 0.91 Å). By identifying the geometrically closest saddle, OTGPD successfully captures the most immediate reaction pathway accessible from the initial geometry.

The Sella algorithm requires 116 HF calculations for convergence—a 3.2-fold computational overhead. Rather than locating the nearby fragmentation pathway, Sella explores a more complex trajectory and identifies a chemically distinct saddle corresponding to a 1,5-hydrogen atom transfer (1,5-HAT) from C5 to the carbonyl oxygen. This saddle possesses a nearly isoenergetic barrier height of 2.38 eV, yet lies significantly further from the initial structure (RMSD = 1.28 Å). Sella has bypassed the more proximal and chemically direct C–O bond cleavage saddle entirely.

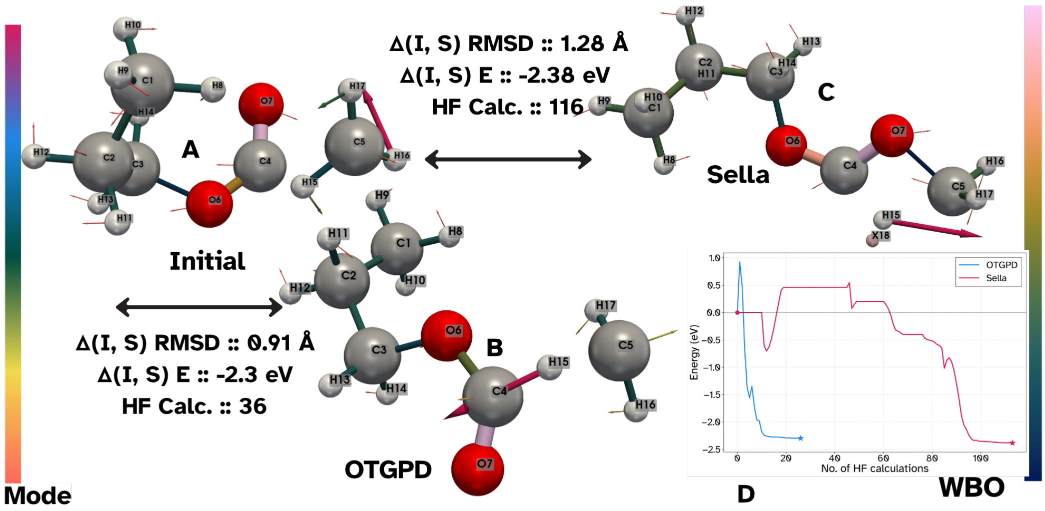


Figure 8.8. Comparison of the OTGPD and Sella algorithms for a saddle point search on n-propyl acetate (`singlet_016`), starting from the initial, non-equilibrium geometry. The OTGPD method efficiently locates the geometrically proximal saddle point corresponding to C–O bond cleavage in 36 steps. The Sella method follows a more computationally intensive path of 116 steps to find a more distant, nearly isoenergetic saddle corresponding to a 1,5-hydrogen atom transfer. The plot of the energy profiles for both searches highlights the significant difference in computational cost.

A two-dimensional landscape projection [122] (Figure 8.9) maps pathways from the initial, high-energy configuration to the deep intermediate minimum and transition states. The white star marks the converged OTGPD saddle, which is clearly proximal to the initial point, while the surface shows the trajectory of paths explored during the optimization.

This behavior reveals a critical difference between the two approaches. While both employ eigenvalue-following strategies, Gaussian process accelerated forms of the dimer in cartesian coordinates efficiently converge to nearby saddles, whereas Sella's search trajectory in internal coordinates may overshoot geometrically proximal transition states in favor of lower-energy alternatives located further from the starting configuration. For comprehensive reaction exploration, this systematic overshoot would result in undercounting of the local reaction network in automated discovery schemes. This would suggest that the OTGPD's efficiency and geometric proximity make it superior for discovering proximal transition states which is a prerequisite capability for comprehensively cataloging accessible chemical transformations on the fly for AKMC.

8.6.4 Performance

The raw solver throughput is visualized in the cactus plot (Figure 8.10 A). This plot of cumulative problems solved versus wall-clock time shows that OTGPD performance curve rises most steeply, indicating that it solves a large number of problems in significantly less time than its counterparts. GPDimer follows as the next most efficient, while the standard Dimer method exhibits a considerable lag, requiring more time to solve an equivalent number of systems. Within the first 10 seconds of wall-clock time, OTGPD successfully solves over 100 problems, while the standard Dimer has converged on fewer than 50. While GPDimer is

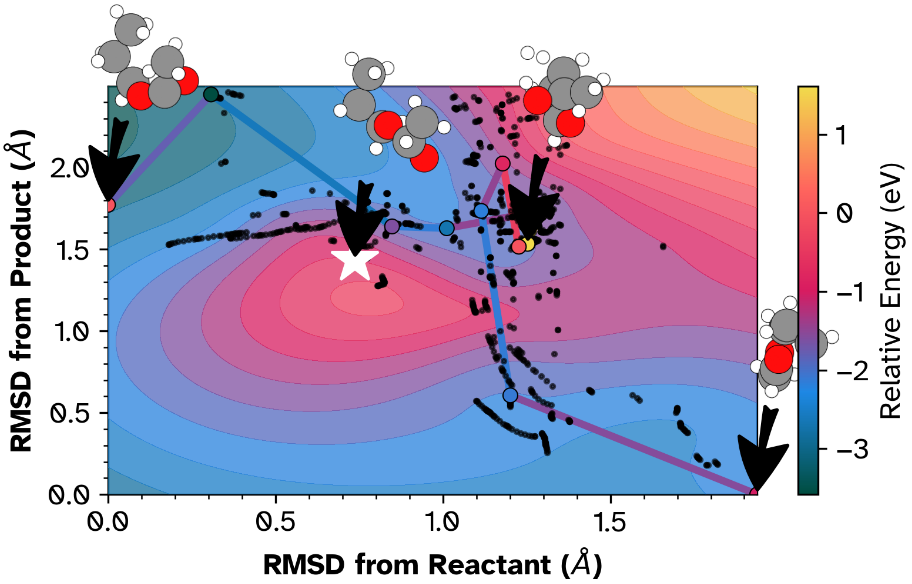


Figure 8.9. *A 2D landscape projection visualizing the potential energy surface of the n-propyl acetate system. This surface, described in Sec. 4.2, depicts the energy landscape as a function of observed paths during the optimization. The landscape clearly reveals several states. The proximal transition state (white star), corresponding to C-O cleavage, is the converged saddle point located by OTGPD and Dimer. The more distant saddle is the endpoint corresponding to the 1,5-HAT, located by Sella. This visualization strongly suggests that Sella's trajectory overshoots the first, more proximal saddle. We highlight the converged dimer saddle to illustrate that while the Sella trajectory passes near this configuration, it fails to localize the proximal transition state, instead proceeding to the distal 1,5-HAT saddle. The Sella trajectory passes near the dimer saddle configuration as reported earlier [50].*

a clear improvement over the Dimer, it consistently lags OTGPD in the number of systems solved within any given time budget.

This superior speed is partially accounted for by improved data efficiency, as shown in Figure 8.10 B. The violin plots reveal that the number of HF calculations required by the standard Dimer is an order of magnitude greater than that for the Gaussian Process -accelerated methods. We quantify this in the median number of calculations: just 28 for OTGPD and 30 for GPDimer, compared to 254 for the standard Dimer. This drastic reduction in expensive electronic structure calculations is the primary driver of the observed performance gains.

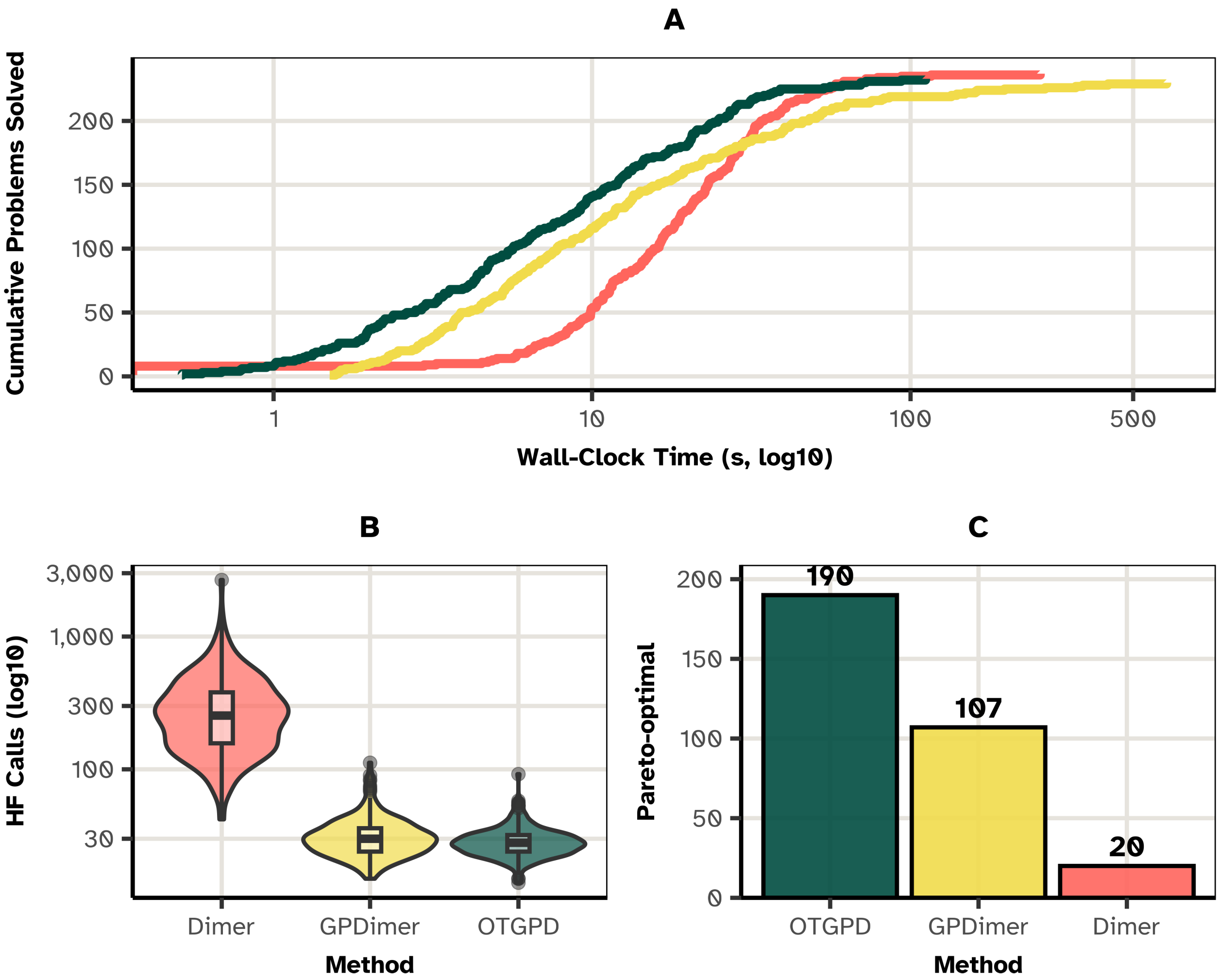


*Figure 8.10. Comparison of computational efficiency for the OTGPD, GPDimer, and standard Dimer methods. **(A)** A cactus plot shows the cumulative number of problems solved versus wall-clock time, demonstrating OTGPD's superior raw speed. **(B)** Violin plots of the number of Hartree-Fock (HF) calls show the order-of-magnitude improvement in data efficiency for the GP-accelerated methods. **(C)** A bar chart of the per-system Pareto-optimal count reveals that OTGPD most frequently provides the best trade-off between solution time and the number of HF calls, appearing on the frontier for 190 systems compared to 107 for GPDimer and 20 for the standard Dimer.*

While both Gaussian Process methods exhibit similar data efficiency, the cactus plot shows that OTGPD wall-clock performance is significantly better than GPDimer. This highlights the effect of the computational overhead reduction. By design, the OTGPD minimizes this overhead, ensuring that the gains from reduced data requirements translate more effectively into real-world speed.

The most sophisticated measure of performance comes from a per-system Pareto optimality analysis (Figure 8.10 C. This moves beyond single-metric comparisons to find the set of solutions that represent the best possible trade-offs. For a set of solutions, a given solution is Pareto-optimal if no other solution is superior in all objectives. Formally, a vector of objectives $F(x_A)$ dominates $F(x_B)$, noted $F(x_A) \prec F(x_B)$, if:

$$F(x_A) \prec F(x_B) \iff \forall i, f_i(x_A) \leq f_i(x_B) \land \exists j, f_j(x_A) < f_j(x_B) \tag{99}$$

The analysis identifies which method resides on the Pareto frontier, which is the set of non-dominated solutions for each system. The results show that OTGPD is on the Pareto-optimal frontier for 190 systems, making it the optimal choice nearly twice as often as GPDimer (107) and almost ten times more frequently than the standard Dimer (20). This confirms that the OTGPD consistently finds the best balance of computational costs.

8.7 Conclusions

This chapter transformed the Gaussian Process from a volatile statistical interpolator into a robust engine for chemical discovery. We demonstrated that naive application of machine learning to Cartesian landscapes frequently yields pathological failures, such as the “cold fusion” artifacts observed in the standard GPDimer. These failures stem not from a lack of data, but from a lack of physical constraint.

To remedy this, we replaced the arbitrary Euclidean metric with the permutation-invariant Intensive Earth Mover’s Distance (EMD). This metric equips the surrogate with a chemically meaningful sense of locality. It underpins the adaptive trust radius, a mechanism that strictly enforces the validity of the local approximation and prevents the optimizer from wandering into unphysical regions of the potential energy surface. Furthermore, by linking data selection to numerical stability via Farthest Point Sampling, we ensured that the covariance matrices remain well-conditioned, satisfying Gershgorin bounds and enabling reliable inversion even in high-throughput regimes.

Empirical validation confirms the efficacy of this approach. The OTGPD lies on the Pareto-optimal frontier for the majority of test systems, offering the most favorable trade-off between wall-clock time and electronic structure calls. Crucially, the comparison with Sella reveals a distinct topological advantage: while internal-coordinate methods often overshoot to find thermodynamically stable but geometrically distant states, OTGPD reliably converges to the proximal saddle point. This capability proves essential for kinetic Monte Carlo simulations, where discovering the immediate connectivity of the reaction network takes precedence over finding the global minimum.

Ultimately, the OTGPD represents the culmination of the efficiency strategies developed throughout this thesis. It integrates the robust saddle-finding logic of the Dimer method, the statistical power of Gaussian Processes, and the rigorous stability controls derived from optimal transport theory into a single, cohesive framework for autonomous reaction discovery.

9 Summary

> The purpose of computing is insight, not numbers.
>
> Richard Hamming

This dissertation presented a multi-faceted investigation into the role of computational representation as a primary driver of progress in chemical physics. We structured this inquiry around specific research objectives targeting relativistic electronic structure, software architecture, statistical benchmarking, and algorithmic efficiency. Here, we summarize the contributions against those initial goals.

We began by questioning whether a squared Hamiltonian formulation within a finite element framework could provide stability for all-electron relativistic calculations. Chapter 3 validated this hypothesis. The `featom` solver, built upon a high-order FEM discretization of the squared Dirac operator, successfully eliminated spectral pollution. It achieves sub-second wall times for all-electron DFT calculations of heavy elements like Uranium, outperforming state-of-the-art shooting methods in the relativistic regime while avoiding the complexity of kinetic balance constraints.

Subsequently, we investigated whether modernizing legacy software architectures could unlock novel scientific algorithms. Chapter 4 demonstrated this through the complete refactoring of the EON suite. By decoupling the potential energy surface via a client-server architecture, we enabled the creation of the hybrid MMF-NEB method. This algorithm reduces the computational cost of double-ended saddle searches by approximately 60% compared to standard approaches, a gain achievable only through the flexibility of the redesigned state management system. In Chapter 5, a C++ rewrite enabled the Cartesian coordinate based GPDimer to compete with state of the art internal coordinate methods.

To ensure these gains were robust, we sought to move beyond average-cost benchmarks and rigorously quantify algorithmic reliability. Chapter 6 applied Bayesian generalized linear mixed models to the Dimer method. This analysis overturned the conventional wisdom regarding rotational minimization, providing robust statistical evidence that the Conjugate Gradient optimizer offers superior reliability compared to L-BFGS. Furthermore, we quantified the high cost of quaternion-based rotation removal within EON, demonstrating that it incurs a significant computational penalty without a credible improvement in convergence success.

Finally, we addressed the cubic scaling and instability bottlenecks of Gaussian Process acceleration. Chapter 8 introduced the OTGPD framework. By implementing the Intensive Earth Mover's Distance for physically motivated trust regions and Farthest Point Sampling for numerical conditioning, we eliminated the "cold fusion" pathologies observed in previous methods. The resulting algorithm places on the Pareto frontier for 190 out of 238 test systems, effectively halving the time-to-solution compared to the standard GPDimer and demonstrating superior performance to internal-coordinate methods for proximal saddle searches.

10 Conclusions

> Dealing with failure is easy: Work hard to improve. Success is also easy to handle: You've solved the wrong problem. Work hard to improve.
>
> Alan Perlis

As summarized in the preceding chapter, this thesis produced a suite of novel methods and robust software tools, transforming raw computation into scientific insight. Yet, as the doctrine of Alan Perlis suggests, every solution unmasks a new layer of problems. This chapter moves beyond a summary of specific results to synthesize their broader implications, confront the limitations that remain, and outline a path forward for the future of computational science.

10.1 The Science of Scientific Software

The development of powerful, specialized tools demands a significant investment of expert effort, yet their long-term maintenance and the management of their inherent technical debt represent a systemic challenge in academic research. The future of the field depends not only on the creation of novel algorithms but on the elevation of software construction to the status of a science itself. By treating the software as a first-class research object, we ensure that the insights gained from one generation of scientific inquiry provide a robust foundation for the next.

Across the breadth of this thesis, we emphasized that efficient representations emerge only through the effort expended in "speaking binary." The era of "formula translation"—blindly transcribing mathematical notation into code—has ended. The constraints of modern hardware, particularly in the exascale era, demand a deeper engagement with memory hierarchies, vectorization, and concurrency. The success of the OTGPD serves as a testament to this philosophy; it functions essentially as an exercise in rigorous profiling. We identified that hyperparameter estimation constituted the wall-time bottleneck and addressed it not through clearer mathematics, but through algorithmic restructuring and data pruning. Thus, technical mastery over average-time complexity must accompany theoretical innovation.

10.2 The Ontological Gap: Statistics vs. Physics

A central tension persists throughout this work: the disconnect between the statistical surrogate and the physical reality it models. A Gaussian Process defines a MVN over function values. We select the specific distribution that best explains the observed data by maximizing the MLL. However, this statistical procedure implies no guarantee that the model reproduces the true Potential Energy Surface generated by electronic structure theory.

The true potential energy surface arises from the many-body expansion [159]:

$$V(x) = \sum_{i} V_1(i) + \sum_{i<j} V_2(i,j) + \sum_{i<j<k} V_3(i,j,k) + \cdots + V_N(1,\ldots,N) \tag{100}$$

This expansion possesses specific decay properties and symmetries not inherently captured by the two-point covariances of a stationary Gaussian kernel. The GP "anthropomorphizes" the data; we often ascribe unjustified physical meaning to hyperparameters like the signal variance or length scale. In reality, these parameters merely maximize the likelihood of a specific dataset. When that dataset consists of sparse, correlated points along a saddle-search path—atypical of the global surface—the MLL surface often becomes shallow or degenerate. This manifests visibly in the "cold fusion" pathologies observed in the standard GPDimer, where the signal variance explodes to accommodate high-force gradients, flattening the mean prediction and directing the search into unphysical regions.

The OTGPD mitigates this not by forcing the GP to understand physics, but by imposing physical constraints (trust regions, stability barriers) upon the statistical engine. We acknowledge that the model operates as a mathematical interpolator, not a physical simulator, and constrain it accordingly.

10.3 Future Outlook: A BLAS for Chemical Kinetics

The OT-GP framework, while successful, shares the worst-case cubic time complexity of its predecessors. Pruning over dynamic data remains an open challenge, particularly for systems exceeding the size regimes explored here. To address this, the field requires a paradigm shift in software architecture.

We propose the vision of a "BLAS for chemical kinetics." Just as the Basic Linear Algebra Subprograms (BLAS) provide a standard, highly optimized foundation for linear algebra, we require a standardized, decoupled layer for kinetic primitives—saddle searches, rate evaluations, and traversing potential landscapes. Such a framework would rely on polyglot libraries communicating through zero-copy interfaces, allowing researchers to compose high-level workflows (like AKMC or MD) from highly optimized, hardware-aware components.

This architectural evolution paves the path for scaling the methods developed herein. The OTGPD provides the foundational approach for the active learning of high-energy transition state geometries. These datasets act as the critical feedstock for training next-generation reactive machine-learned potentials. By integrating these surrogates with symmetry-adapted representations or reinforcement learning agents, we move closer to the ultimate goal: the autonomous, reliable exploration of the vastness of chemical space.

References

[1] Debabrata Goswami. “Control of Chemical Dynamics Using Arbitrary Shaped Optical Pulses and Laser-Enhanced NMR Spectroscopy.” PhD thesis. Princeton University, Jan. 1994.

[2] Amrita Goswami. “Crystal Nucleation: Challenges and Future Horizons, under Confinement and Shear.” PhD thesis. Kanpur: Indian Institute of Technology Kanpur, 2021.

[3] Kenneth George Denbigh. *The Principles of Chemical Equilibrium: With Applications in Chemistry and Chemical Engineering*. 4th ed. Cambridge [Eng.] New York: Cambridge University Press, 1997.

[4] Rolf Landauer. “Information Is Physical.” In: *Physics Today* 44.5 (May 1991), pp. 23–29. DOI: 10.1063/1.881299.

[5] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Elsevier, Oct. 2001.

[6] Baron Peters. *Reaction Rate Theory and Rare Events*. Amsterdam ; Cambrige, MA: Elsevier, 2017.

[7] Andreas Pedersen and Hannes Jónsson. “Distributed Implementation of the Adaptive Kinetic Monte Carlo Method.” In: *Mathematics and Computers in Simulation*. Multiscale Modeling of Moving Interfaces in Materials 80.7 (Mar. 2010), pp. 1487–1498. DOI: 10.1016/j.matcom.2009.02.010.

[8] Graeme Henkelman and Hannes Jónsson. “Long Time Scale Kinetic Monte Carlo Simulations without Lattice Approximation and Predefined Event Table.” In: *The Journal of Chemical Physics* 115.21 (Nov. 2001), pp. 9657–9666. DOI: 10.1063/1.1415500.

[9] Fedwa El-Mellouhi, Normand Mousseau, and Laurent J. Lewis. “Kinetic Activation-Relaxation Technique: An off-Lattice Self-Learning Kinetic Monte Carlo Algorithm.” In: *Physical Review B* 78.15 (Oct. 2008), p. 153202. DOI: 10.1103/PhysRevB.78.153202.

[10] Errol G. Lewars. *Computational Chemistry*. Cham: Springer International Publishing, 2016. DOI: 10.1007/978-3-319-30916-3.

[11] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Vol. 103. Springer Texts in Statistics. New York, NY: Springer New York, 2013. DOI: 10.1007/978-1-4614-7138-7.

[12] Trevor Hastie, Robert Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer Series in Statistics. New York, NY: Springer, 2009.

[13] Rachid Malek and Normand Mousseau. "Dynamics of Lennard-Jones Clusters: A Characterization of the Activation-Relaxation Technique." In: *Physical Review E* 62.6 (Dec. 2000), pp. 7723–7728. DOI: 10.1103/PhysRevE.62.7723.

[14] David J. Wales, Jonathan P. K. Doye, Mark A. Miller, Paul N. Mortenson, and Tiffany R. Walsh. "Energy Landscapes: From Clusters to Biomolecules." In: *Advances in Chemical Physics*. John Wiley & Sons, Ltd, 2000, pp. 1–111. DOI: 10.1002/9780470141748.ch1.

[15] Mijanur Rahman. "Modeling and numerical study of the diffusion of point defects in $\alpha-$iron." In: (), pp. 0–179.

[16] Gareth A. Tribello, Michele Ceriotti, and Michele Parrinello. "Using Sketch-Map Coordinates to Analyze and Bias Molecular Dynamics Simulations." In: *Proceedings of the National Academy of Sciences* 109.14 (Apr. 2012), pp. 5196–5201. DOI: 10.1073/pnas.1201152109.

[17] H. Scott Fogler. *Essentials of Chemical Reaction Engineering*. Upper Saddle River, NJ: Prentice Hall, 2011.

[18] Octave Levenspiel. *Chemical Reaction Engineering*. 3rd ed. New York: Wiley, 1999.

[19] David Mautner Himmelblau. *Basic Principles and Calculations in Chemical Engineering*. 5th ed. Prentice-Hall International Series in Physical and Chemical Engineering Sciences. Englewood Cliffs, N.J: Prentice Hall, 1989.

[20] Mie Andersen, Chiara Panosetti, and Karsten Reuter. "A Practical Guide to Surface Kinetic Monte Carlo Simulations." In: *Frontiers in Chemistry* 7 (2019). DOI: 10.3389/fchem.2019.00202.

[21] Corbett C. Battaile. "The Kinetic Monte Carlo Method: Foundation, Implementation, and Application." In: *Computer Methods in Applied Mechanics and Engineering*. Recent Advances in Computational Study of Nanostructures 197.41 (July 2008), pp. 3386–3398. DOI: 10.1016/j.cma.2008.03.010.

[22] Mickaël Trochet, Normand Mousseau, Laurent Karim Béland, and Graeme Henkelman. "Off-Lattice Kinetic Monte Carlo Methods." In: *Handbook of Materials Modeling: Methods: Theory and Modeling*. Ed. by Wanda Andreoni and Sidney Yip. Cham: Springer International Publishing, 2020, pp. 715–743. DOI: 10.1007/978-3-319-44677-6_29.

[23] Graeme Henkelman, Hannes Jónsson, Tony Lelièvre, Normand Mousseau, and Arthur F. Voter. "Long-Timescale Simulations: Challenges, Pitfalls, Best Practices, for Development and Applications." In: *Handbook of Materials Modeling*. Ed. by Wanda Andreoni and Sidney Yip. Cham: Springer International Publishing, 2018, pp. 1–10. DOI: 10.1007/978-3-319-42913-7_31-1.

[24] James Bisgard. "Mountain Passes and Saddle Points." In: *SIAM Review* 57.2 (Jan. 2015), pp. 275–292. DOI: 10.1137/140963510.

[25] Graeme Henkelman and Hannes Jónsson. "A Dimer Method for Finding Saddle Points on High Dimensional Potential Surfaces Using Only First Derivatives." In: *The Journal of Chemical Physics* 111.15 (Oct. 1999), pp. 7010–7022. DOI: 10.1063/1.480097.

[26] Normand Mousseau and G. T. Barkema. “Traveling through Potential Energy Landscapes of Disordered Materials: The Activation-Relaxation Technique.” In: *Physical Review E* 57.2 (Feb. 1998), pp. 2419–2424. DOI: 10.1103/PhysRevE.57.2419.

[27] Lindsey J. Munro and David J. Wales. “Defect Migration in Crystalline Silicon.” In: *Physical Review B* 59.6 (Feb. 1999), pp. 3969–3980. DOI: 10.1103/PhysRevB.59.3969.

[28] Charles J. Cerjan and William H. Miller. “On Finding Transition States.” In: *The Journal of Chemical Physics* 75.6 (Sept. 1981), pp. 2800–2806. DOI: 10.1063/1.442352.

[29] Samuel T Chill, Matthew Welborn, Rye Terrell, Liang Zhang, Jean-Claude Berthet, Andreas Pedersen, Hannes Jónsson, and Graeme Henkelman. “EON: Software for Long Time Simulations of Atomic Scale Systems.” In: *Modelling and Simulation in Materials Science and Engineering* 22.5 (July 2014), p. 055002. DOI: 10.1088/0965-0393/22/5/055002.

[30] Hannes Jonsson, Greg Mills, and Karsten W. Jacobsen. “Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions.” In: *Classical and Quantum Dynamics in Condensed Phase Simulations*. World Scientific, June 1998, pp. 385–404. DOI: 10.1142/9789812839664_0016.

[31] Graeme Henkelman and Hannes Jónsson. “Improved Tangent Estimate in the Nudged Elastic Band Method for Finding Minimum Energy Paths and Saddle Points.” In: *The Journal of Chemical Physics* 113.22 (Dec. 2000), pp. 9978–9985. DOI: 10.1063/1.1323224.

[32] Søren Smidstrup, Andreas Pedersen, Kurt Stokbro, and Hannes Jónsson. “Improved Initial Guess for Minimum Energy Path Calculations.” In: *The Journal of Chemical Physics* 140.21 (June 2014), p. 214106. DOI: 10.1063/1.4878664.

[33] Vilhjálmur Ásgeirsson, Benedikt Orri Birgisson, Ragnar Bjornsson, Ute Becker, Frank Neese, Christoph Riplinger, and Hannes Jónsson. “Nudged Elastic Band Method for Molecular Reactions Using Energy-Weighted Springs Combined with Eigenvector Following.” In: *Journal of Chemical Theory and Computation* 17.8 (Aug. 2021), pp. 4929–4945. DOI: 10.1021/acs.jctc.1c00462.

[34] Michael J. Willatt, Félix Musil, and Michele Ceriotti. “Atom-Density Representations for Machine Learning.” In: *The Journal of Chemical Physics* 150.15 (Apr. 2019), p. 154110. DOI: 10.1063/1.5090481.

[35] Sergey N. Pozdnyakov, Michael J. Willatt, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. “Incompleteness of Atomic Structure Representations.” In: *Physical Review Letters* 125.16 (Oct. 2020), p. 166001. DOI: 10.1103/PhysRevLett.125.166001.

[36] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: The MIT Press, 2016.

[37] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2023.

[38] Håvard Berland. "Automatic Differentiation." In: (), p. 22.

[39] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. "Automatic Differentiation in PyTorch." In: (), p. 4.

[40] C. A. Calder and N. Cressie. "Kriging and Variogram Models." In: *International Encyclopedia of Human Geography*. Ed. by Rob Kitchin and Nigel Thrift. Oxford: Elsevier, Jan. 2009, pp. 49–55. DOI: 10.1016/B978-008044910-4.00461-2.

[41] Radford M. Neal. "Priors for Infinite Networks." In: *Bayesian Learning for Neural Networks*. Ed. by Radford M. Neal. New York, NY: Springer, 1996, pp. 29–53. DOI: 10.1007/978-1-4612-0745-0_2.

[42] Greg Yang. "Wide Feedforward or Recurrent Neural Networks of Any Architecture Are Gaussian Processes." In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[43] Philipp Hennig, Michael A. Osborne, and Mark Girolami. "Probabilistic Numerics and Uncertainty in Computations." In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 471.2179 (July 2015), p. 20150142. DOI: 10.1098/rspa.2015.0142. arXiv: 1506.01326 [cs, math, stat].

[44] Simo Särkkä. "Linear Operators and Stochastic Partial Differential Equations in Gaussian Process Regression." In: *Artificial Neural Networks and Machine Learning – ICANN 2011*. Ed. by Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski. Vol. 6792. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 151–158. DOI: 10.1007/978-3-642-21738-8_20.

[45] Robert B. Gramacy. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. New York, NY: CRC Press ; Taylor & Francis Group, 2020.

[46] Eugène Sanscartier, Félix Saint-Denis, Karl-Étienne Bolduc, and Normand Mousseau. "Evaluating Approaches for On-the-Fly Machine Learning Interatomic Potentials for Activated Mechanisms Sampling with the Activation-Relaxation Technique Nouveau." In: *Journal of Chemical Physics* 158.24 (June 2023), p. 244110. DOI: 10.1063/5.0143211.

[47] D. Madsen, R. Pearman, and M. Gruebele. "Approximate Factorization of Molecular Potential Surfaces. I. Basic Approach." In: *Journal of Chemical Physics* 106.14 (Apr. 1997), pp. 5874–5893. DOI: 10.1063/1.473253.

[48] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press, 2006.

[49] Olli-Pekka Koistinen, Vilhjálmur Ásgeirsson, Aki Vehtari, and Hannes Jónsson. "Minimum Mode Saddle Point Searches Using Gaussian Process Regression with Inverse-Distance Covariance Function." In: *Journal of Chemical Theory and Computation* 16.1 (Jan. 2020), pp. 499–509. DOI: 10.1021/acs.jctc.9b01038.

[50] Rohit Goswami, Maxim Masterov, Satish Kamath, Alejandro Pena-Torres, and Hannes Jónsson. "Efficient Implementation of Gaussian Process Regression Accelerated Saddle Point Searches with Application to Molecular Reactions." In: *Journal of Chemical Theory and Computation* (July 2025). DOI: 10.1021/acs.jctc.5c00866.

[51] E. Solak, R. Murray-smith, W. Leithead, D. Leith, and Carl Rasmussen. "Derivative Observations in Gaussian Process Models of Dynamic Systems." In: *Advances in Neural Information Processing Systems*. Ed. by S. Becker, S. Thrun, and K. Obermayer. Vol. 15. MIT Press, 2002.

[52] Felix Musil, Andrea Grisafi, Albert P. Bartók, Christoph Ortner, Gábor Csányi, and Michele Ceriotti. "Physics-Inspired Structural Representations for Molecules and Materials." In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 9759–9815. DOI: 10.1021/acs.chemrev.1c00021.

[53] Miguel A. Caro. "Optimizing Many-Body Atomic Descriptors for Enhanced Computational Performance of Machine Learning Based Interatomic Potentials." In: *Physical Review B* 100.2 (July 2019), p. 024112. DOI: 10.1103/PhysRevB.100.024112.

[54] Albert P. Bartók. "Gaussian Approximation Potential: An Interatomic Potential Derived from First Principles Quantum Mechanics." In: *arXiv:1003.2817 [cond-mat, physics:physics]* (Mar. 2010). arXiv: 1003.2817 [cond-mat, physics:physics].

[55] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. "Machine Learning for Molecular Simulation." In: *Annual Review of Physical Chemistry* 71.1 (Apr. 2020), pp. 361–390. DOI: 10.1146/annurev-physchem-042018-052331.

[56] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second edition. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018.

[57] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for Optimization*.

[58] Rohit Goswami and Hannes Jónsson. *Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches*. Oct. 2025. DOI: 10.48550/arXiv.2510.06030. arXiv: 2510.06030 [physics].

[59] Lei Li, Ryan A. Ciufo, Jiyoung Lee, Chuan Zhou, Bo Lin, Jaeyoung Cho, Naman Katyal, and Graeme Henkelman. "Atom-Centered Machine-Learning Force Field Package." In: *Computer Physics Communications* 292 (Nov. 2023), p. 108883. DOI: 10.1016/j.cpc.2023.108883.

[60] Filippo Bigi, Sergey N. Pozdnyakov, and Michele Ceriotti. "Wigner Kernels: Body-Ordered Equivariant Machine Learning without a Basis." In: *Journal of Chemical Physics* 161.4 (2024). DOI: 10.1063/5.0208746.

[61] Yunxing Zuo, Chi Chen, Xiangguo Li, Zhi Deng, Yiming Chen, Jörg Behler, Gábor Csányi, Alexander V. Shapeev, Aidan P. Thompson, Mitchell A. Wood, and Shyue Ping Ong. "Performance and Cost Assessment of Machine Learning Interatomic Potentials." In: *The Journal of Physical Chemistry A* 124.4 (Jan. 2020), pp. 731–745. DOI: 10.1021/acs.jpca.9b08723.

[62] Danish Khan, Stefan Heinen, and O. Anatole von Lilienfeld. "Kernel Based Quantum Machine Learning at Record Rate : Many-body Distribution Functionals as Compact Representations." In: *The Journal of Chemical Physics* 159.3 (July 2023), p. 034106. DOI: 10.1063/5.0152215. arXiv: 2303.16312 [physics].

[63] Arslan Mazitov, Filippo Bigi, Matthias Kellner, Paolo Pegolo, Davide Tisi, Guillaume Fraux, Sergey Pozdnyakov, Philip Loche, and Michele Ceriotti. *PET-MAD, a Universal Interatomic Potential for Advanced Materials Modeling*. Mar. 2025. DOI: 10.48550/arXiv.2503.14118. arXiv: 2503.14118 [cond-mat].

[64] Ondřej Čertík, John E. Pask, Isuru Fernando, Rohit Goswami, N. Sukumar, Lee. A. Collins, Gianmarco Manzini, and Jiří Vackář. "High-Order Finite Element Method for Atomic Structure Calculations." In: *Computer Physics Communications* (Dec. 2023), p. 109051. DOI: 10.1016/j.cpc.2023.109051.

[65] Douglas Rayner Hartree and W. Hartree. "Self-Consistent Field, with Exchange, for Nitrogen and Sodium." In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 193.1034 (July 1948), pp. 299–304. DOI: 10.1098/rspa.1948.0047.

[66] Douglas Rayner Hartree and W. Hartree. "Self-Consistent Field, with Exchange, for Beryllium." In: *Proceedings of the Royal Society of London. Series A - Mathematical and Physical Sciences* 150.869 (May 1935), pp. 9–33. DOI: 10.1098/rspa.1935.0085.

[67] Rodney J. Bartlett and John F. Stanton. "Applications of Post-hartree—Fock Methods: A Tutorial." In: *Reviews in Computational Chemistry*. Ed. by Kenny B. Lipkowitz and Donald B. Boyd. 1st ed. Vol. 5. Wiley, Jan. 1994, pp. 65–169. DOI: 10.1002/9780470125823.ch2.

[68] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Mineola, N.Y: Dover Publications, 1996.

[69] P. Hohenberg and W. Kohn. "Inhomogeneous Electron Gas." In: *Physical Review* 136.3B (Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864.

[70] John P. Perdew. "Jacob's Ladder of Density Functional Approximations for the Exchange-Correlation Energy." In: *AIP Conference Proceedings*. Vol. 577. Antwerp (Belgium): AIP, 2001, pp. 1–20. DOI: 10.1063/1.1390175.

[71] A. O. Dohn, E. Ö. Jónsson, G. Levi, J. J. Mortensen, O. Lopez-Acevedo, K. S. Thygesen, K. W. Jacobsen, J. Ulstrup, N. E. Henriksen, K. B. Møller, and H. Jónsson. "Grid-Based Projector Augmented Wave (GPAW) Implementation of Quantum Mechanics/Molecular Mechanics (QM/MM) Electrostatic Embedding and Application to a Solvated Diplatinum Complex." In: *Journal of Chemical Theory and Computation* 13.12 (Dec. 2017), pp. 6010–6022. DOI: 10.1021/acs.jctc.7b00621.

[72] I. P. Grant. *Relativistic Quantum Theory of Atoms and Molecules: Theory and Computation*. Springer Series on Atomic, Optical, and Plasma Physics 40. New York: Springer, 2007.

[73] I. I. Tupitsyn and V. M. Shabaev. "Spurious States of the Dirac Equation in a Finite Basis Set." In: *Optics and Spectroscopy* 105.2 (Aug. 2008), pp. 183–188. DOI: 10.1134/S0030400X08080043.

[74] Ondřej Čertík, John E. Pask, and Jiří Vackář. "Dftatom: A Robust and General Schrödinger and Dirac Solver for Atomic Structure Calculations." In: *Computer Physics Communications* 184.7 (July 2013), pp. 1777–1791. DOI: 10.1016/j.cpc.2013.02.014.

[75] Kenneth G. Dyall and Knut Fægri. "Kinetic Balance and Variational Bounds Failure in the Solution of the Dirac Equation in a Finite Gaussian Basis Set." In: *Chemical Physics Letters* 174.1 (Nov. 1990), pp. 25–32. DOI: 10.1016/0009-2614(90)85321-3.

[76] Charlotte Froese Fischer and Oleg Zatsarinny. "A B-spline Galerkin Method for the Dirac Equation." In: *Computer Physics Communications* 180.6 (June 2009), p. 879. DOI: 10.1016/j.cpc.2008.12.010.

[77] I. P. Grant. "B-Spline Methods for Radial Dirac Equations." In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 42.5 (Mar. 2009). DOI: 10.1088/0953-4075/42/5/055002.

[78] Hasan Almanasreh, Sten Salomonson, and Nils Svanstedt. "Stabilized Finite Element Method for the Radial Dirac Equation." In: *Journal of Computational Physics* 236 (Mar. 2013), pp. 426–442. DOI: 10.1016/j.jcp.2012.11.020.

[79] V. Shabaev, I. Tupitsyn, V. Yerokhin, G. Plunien, and G. Soff. "Dual Kinetic Balance Approach to Basis-Set Expansions for the Dirac Equation." In: *Physical Review Letters* 93.13 (Sept. 2004), pp. 1–4. DOI: 10.1103/PhysRevLett.93.130405.

[80] K Beloy and a Derevianko. "Application of the Dual-Kinetic-Balance Sets in the Relativistic Many-Body Problem of Atomic Structure." In: *Computer Physics Communications* 179.5 (Sept. 2008), pp. 310–319. DOI: 10.1016/j.cpc.2008.03.004.

[81] Qiming Sun, Wenjian Liu, and Werner Kutzelnigg. "Comparison of Restricted, Unrestricted, Inverse, and Dual Kinetic Balances for Four-Component Relativistic Calculations." In: *Theoretical Chemistry Accounts* 129.3-5 (June 2011), pp. 423–436. DOI: 10.1007/s00214-010-0876-6.

[82] Li Guang Jiao, Yu Ying He, Aihua Liu, Yong Zhi Zhang, and Yew Kam Ho. "Development of the Kinetically and Atomically Balanced Generalized Pseudospectral Method." In: *Physical Review A* 104.2 (Aug. 2021), p. 022801. DOI: 10.1103/PhysRevA.104.022801.

[83] Werner Kutzelnigg. "Basis set expansion of the Dirac operator without variational collapse." In: *International Journal of Quantum Chemistry* 25.1 (1984), pp. 107–129. DOI: 10.1002/qua.560250112.

[84] Hasan Almanasreh. "Finite Element Method for Solving the Dirac Eigenvalue Problem with Linear Basis Functions." In: *Journal of Computational Physics* 376 (Jan. 2019), pp. 1199–1211. DOI: 10.1016/j.jcp.2018.10.022.

[85] Ji-Yu Fang, Shou-Wan Chen, and Tai-Hua Heng. "Solution to the Dirac Equation Using the Finite Difference Method." In: *Nuclear Science and Techniques* 31.2 (Jan. 2020), p. 15. DOI: 10.1007/s41365-020-0728-6.

[86] W. R. Johnson, S. A. Blundell, and J. Sapirstein. "Finite Basis Sets for the Dirac Equation Constructed from B Splines." In: *Physical Review A* 37.2 (Jan. 1988), pp. 307–315. DOI: 10.1103/PhysRevA.37.307.

[87] J. Sapirstein and W. R. Johnson. "The Use of Basis Splines in Theoretical Atomic Physics." In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 29.22 (Nov. 1996), pp. 5213–5225. DOI: 10.1088/0953-4075/29/22/005.

[88] H. Wallmeier and W. Kutzelnigg. "Use of the Squared Dirac Operator in Variational Relativistic Calculations." In: *Chemical Physics Letters* 78.2 (Mar. 1981), pp. 341–346. DOI: 10.1016/0009-2614(81)80029-2.

[89] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods*. Ed. by J. E. Marsden, L. Sirovich, and S. S. Antman. Vol. 15. Texts in Applied Mathematics. New York, NY: Springer New York, 2008. DOI: 10.1007/978-0-387-75934-0.

[90] S. S Sastry. *Introductory Methods of Numerical Analysis*. Place of publication not identified: Prentice-Hall Of India Pv, 2010.

[91] Steven C. Chapra and Raymond P. Canale. *Numerical Methods for Engineers*. 7. ed. New York, NY: McGraw-Hill Education, 2015.

[92] James F Epperson. "An Introduction to Numerical Methods and Analysis." In: (2012), p. 615.

[93] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, Sept. 2007.

[94] Laurence J. Kedward, Bálint Aradi, Ondřej Čertík, Milan Curcic, Sebastian Ehlert, Philipp Engel, Rohit Goswami, Michael Hirsch, Asdrubal Lozada-Blanco, Vincent Magnin, Arjen Markus, Emanuele Pagone, Ivan Pribec, Brad Richardson, Harris Snyder, John Urban, and Jérémie Vandenplas. "The State of Fortran." In: *Computing in Science & Engineering* 24.2 (Mar. 2022), pp. 63–72. DOI: 10.1109/MCSE.2022.3159862.

[95] John K. Ousterhout. *A Philosophy of Software Design*. First edition. Palo Alto, CA: Yaknyam Press, 2018.

[96] J. B. Krieger. "Systematic Approximations to the Optimized Effective Potential: Application to Orbital-Density-Functional Theory." In: *Physical Review A* 46.9 (1992), pp. 5453–5458. DOI: 10.1103/PhysRevA.46.5453.

[97] Per Jonsson, X He, Charlotte Froese Fischer, and I. P. Grant. "The grasp2K Relativistic Atomic Structure Package." In: *Computer Physics Communications* 177 (2007), pp. 597–622.

[98] J. P. Desclaux, D. F. Mayers, and F. O'Brien. "Relativistic Atomic Wave Functions." In: *Journal of Physics B: Atomic and Molecular Physics* 4 (1971), pp. 631–642.

[99] D. R. Hamann. "Generalized Norm-Conserving Pseudopotentials." In: *Physical Review B* 40 (1989), pp. 2980–2987.

[100] Charlotte Froese Fischer. "A General Multi-Configuration Hartree-Fock Program." In: *Computer Physics Communications* 64.3 (June 1991), p. 431. DOI: 10.1016/0010-4655(91)90137-A.

[101] A. R. Tackett, N. A. W. Holzwarth, and G. E. Matthews. "A Projector Augmented Wave (PAW) Code for Electronic Structure Calculations, Part II: Pwpaw for Periodic Solids in a Plane Wave Basis." In: *Computer Physics Communications* 135.3 (Apr. 2001), pp. 348–376. DOI: 10.1016/S0010-4655(00)00241-1.

[102] Weile Jia, Zongyan Cao, Long Wang, Jiyun Fu, Xuebin Chi, Weiguo Gao, and Lin-Wang Wang. "The Analysis of a Plane Wave Pseudopotential Density Functional Theory Code on a GPU Machine." In: *Computer Physics Communications* 184.1 (2013), pp. 9–18. doi: `10.1016/j.cpc.2012.08.002`.

[103] Kay Dewhurst. "Elk." 2012.

[104] Oleg Zatsarinny and Charlotte Froese Fischer. "DBSR_HF: A B-spline Dirac–Hartree–Fock Program." In: *Computer Physics Communications* 202 (May 2016), pp. 287–303. doi: `10.1016/j.cpc.2015.12.023`.

[105] Leonardo Belpassi, Matteo De Santis, Harry M. Quiney, Francesco Tarantelli, and Loriano Storchi. "BERTHA: Implementation of a Four-Component Dirac–Kohn–Sham Relativistic Framework." In: *The Journal of Chemical Physics* 152.16 (Apr. 2020), p. 164118. doi: `10.1063/5.0002831`.

[106] Á. Nagy. "Alternative Derivation of the Krieger-Li-iafrate Approximation to the Optimized-Effective-Potential Method." In: *Physical Review A* 55.5 (May 1997), pp. 3465–3468. doi: `10.1103/PhysRevA.55.3465`.

[107] Viktor N. Staroverov, Gustavo E. Scuseria, and Ernest R. Davidson. "Effective Local Potentials for Orbital-Dependent Density Functionals." In: *Journal of Chemical Physics* 125.8 (Aug. 2006), p. 81104. doi: `10.1063/1.2345650`.

[108] David Goldberg. "What Every Computer Scientist Should Know about Floating-Point Arithmetic." In: *ACM Computing Surveys* 23.1 (Mar. 1991), pp. 5–48. doi: `10.1145/103162.103163`.

[109] Michael L. Overton. *Numerical Computing with IEEE Floating Point Arithmetic: Including One Theorem, One Rule of Thumb, and One Hundred and One Exercises*. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics, 2001.

[110] Timothy D. Sauer. *Numerical Analysis*. Third edition. Hoboken: Pearson, 2018.

[111] Peter Gottschling. *Discovering Modern C++: An Intensive Course for Scientists, Engineers, and Programmers*. Second edition. C++ In-Depth Series. Boston: Addison-Wesley, 2021.

[112] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Du\lak, Jesper Friis, Michael N. Groves, Bjørk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, John R. Kitchin, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W. Jacobsen. "The Atomic Simulation Environment—a Python Library for Working with Atoms." In: *Journal of Physics: Condensed Matter* 29.27 (June 2017), p. 273002. doi: `10.1088/1361-648X/aa680e`.

[113] K.B. Wiberg. "Application of the Pople-Santry-Segal CNDO Method to the Cyclopropylcarbinyl and Cyclobutyl Cation and to Bicyclobutane." In: *Tetrahedron* 24.3 (Jan. 1968), pp. 1083–1096. doi: `10.1016/0040-4020(68)88057-3`.

[114] Christoph Bannwarth, Sebastian Ehlert, and Stefan Grimme. "GFN2-xTB—an Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions." In: *Journal of Chemical Theory and Computation* 15.3 (Mar. 2019), pp. 1652–1671. DOI: 10.1021/acs.jctc.8b01176.

[115] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." In: *Nature Methods* 17.3 (Mar. 2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[116] Filippo Bigi, Joseph W. Abbott, Philip Loche, Arslan Mazitov, Davide Tisi, Marcel F. Langer, Alexander Goscinski, Paolo Pegolo, Sanggyu Chong, Rohit Goswami, Sofiia Chorna, Matthias Kellner, Michele Ceriotti, and Guillaume Fraux. *Metatensor and Metatomic: Foundational Libraries for Interoperable Atomistic Machine Learning*. Aug. 2025. DOI: 10.48550/arXiv.2508.15704. arXiv: 2508.15704 [physics].

[117] Miha Gunde. "Development of IRA : A Shape Matching Algorithm, Its Implementation, and Utility in a General off-Lattice kMC Kernel." PhD thesis. Université Paul Sabatier - Toulouse III, Nov. 2021.

[118] Grace Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 1990. DOI: 10.1137/1.9781611970128.

[119] E. Aprà, E. J. Bylaska, W. A. De Jong, N. Govind, K. Kowalski, T. P. Straatsma, M. Valiev, H. J. J. Van Dam, et al. "NWChem: Past, Present, and Future." In: *Journal of Chemical Physics* 152.18 (May 2020), p. 184102. DOI: 10.1063/5.0004997.

[120] Frank Neese, Frank Wennmohs, Ute Becker, and Christoph Riplinger. "The ORCA Quantum Chemistry Program Package." In: *The Journal of Chemical Physics* 152.22 (June 2020), p. 224108. DOI: 10.1063/5.0004608.

[121] Venkat Kapil, Mariana Rossi, Ondrej Marsalek, Riccardo Petraglia, Yair Litman, Thomas Spura, Bingqing Cheng, Alice Cuzzocrea, Robert H. Meißner, David M. Wilkins, Benjamin A. Helfrecht, Przemysław Juda, Sébastien P. Bienvenue, Wei Fang, Jan Kessler, Igor Poltavsky, Steven Vandenbrande, Jelle Wieme, Clemence Corminboeuf, Thomas D. Kühne, David E. Manolopoulos, Thomas E. Markland, Jeremy O. Richardson, Alexandre Tkatchenko, Gareth A. Tribello, Veronique Van Speybroeck, and Michele Ceriotti. "I-PI 2.0: A Universal Force Engine for Advanced Molecular Simulations." In: *Computer Physics Communications* 236 (Mar. 2019), pp. 214–223. DOI: 10.1016/j.cpc.2018.09.020.

[122] Rohit Goswami. *Two-Dimensional RMSD Projections for Reaction Path Visualization and Validation*. Dec. 2025. DOI: 10.48550/arXiv.2512.07329. arXiv: 2512.07329 [physics].

[123] Felix Mölder, Kim Philipp Jablonski, Brice Letcher, Michael B. Hall, Christopher H. Tomkins-Tinch, Vanessa Sochat, Jan Forster, Soohyun Lee, Sven O. Twardziok, Alexander Kanitz, Andreas Wilm, Manuel Holtgrewe, Sven Rahmann, Sven Nahnsen, and Johannes Köster. *Sustainable Data Analysis with Snakemake*. Apr. 2021. DOI: 10.12688/f1000research.29032.2.

[124] Uresh Vahalia. *UNIX Internals: The New Frontiers*. An Alan R. Apt Book. Upper Saddle River, N.J.: Prentice Hall, 1996.

[125] Robert van de Geijn and Kazushige Goto. "BLAS (Basic Linear Algebra Subprograms)." In: *Encyclopedia of Parallel Computing*. Ed. by David Padua. Boston, MA: Springer US, 2011, pp. 157–164. DOI: 10.1007/978-0-387-09766-4_84.

[126] Justin M. Turney, Andrew C. Simmonett, Robert M. Parrish, Edward G. Hohenstein, Francesco A. Evangelista, Justin T. Fermann, Benjamin J. Mintz, Lori A. Burns, Jeremiah J. Wilke, Micah L. Abrams, Nicholas J. Russ, Matthew L. Leininger, Curtis L. Janssen, Edward T. Seidl, Wesley D. Allen, Henry F. Schaefer, Rollin A. King, Edward F. Valeev, C. David Sherrill, and T. Daniel Crawford. "Psi4: An Open-Source Ab Initio Electronic Structure Program." In: *WIREs Computational Molecular Science* 2.4 (2012), pp. 556–565. DOI: 10.1002/wcms.93.

[127] Jarno Vanhatalo, Jaakko Riihimaki, Jouni Hartikainen, Pasi Jylanki, Ville Tolvanen, and Aki Vehtari. "GPstuff: Bayesian Modeling with Gaussian Processes." In: (), p. 5.

[128] Martin Fodslette Møller. "A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning." In: *Neural Networks* 6.4 (Jan. 1993), pp. 525–533. DOI: 10.1016/S0893-6080(05)80056-5.

[129] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. arXiv: 1412.6980 [cs].

[130] Olli-Pekka Koistinen, Vilhjálmur Ásgeirsson, Aki Vehtari, and Hannes Jónsson. "Nudged Elastic Band Calculations Accelerated with Gaussian Process Regression Based on Inverse Interatomic Distances." In: *Journal of Chemical Theory and Computation* 15.12 (Dec. 2019), pp. 6738–6751. DOI: 10.1021/acs.jctc.9b00692.

[131] Rohit Goswami, Amrita Goswami, and Jayant K. Singh. "D-SEAMS: Deferred Structural Elucidation Analysis for Molecular Simulations." In: *Journal of Chemical Information and Modeling* 60.4 (Apr. 2020), pp. 2169–2177. DOI: 10.1021/acs.jcim.0c00031.

[132] Shuhei Watanabe. *Tree-Structured Parzen Estimator: Understanding Its Algorithm Components and Their Roles for Better Empirical Performance*. May 2023. arXiv: 2304.11127 [cs].

[133] Eric D. Hermes, Khachik Sargsyan, Habib N. Najm, and Judit Zádor. "Sella, an Open-Source Automation-Friendly Molecular Saddle Point Optimizer." In: *Journal of Chemical Theory and Computation* 18.11 (Nov. 2022), pp. 6974–6988. DOI: 10.1021/acs.jctc.2c00395.

[134] Jon Baker and Fora Chan. "The Location of Transition States: A Comparison of Cartesian, Z-matrix, and Natural Internal Coordinates." In: *Journal of Computational Chemistry* 17.7 (1996), pp. 888–904. DOI: 10.1002/(SICI)1096-987X(199605)17:7<888::AID-JCC12>3.0.CO;2-7.

[135] Chunyang Peng, Philippe Y. Ayala, H. Bernhard Schlegel, and Michael J. Frisch. "Using Redundant Internal Coordinates to Optimize Equilibrium Geometries and Transition States." In: *Journal of Computational Chemistry* 17.1 (1996), pp. 49–56. DOI: 10.1002/(SICI)1096-987X(19960115)17:1<49::AID-JCC5>3.0.CO;2-0.

[136] Alexander Denzel and Johannes Kästner. "Gaussian Process Regression for Transition State Search." In: *Journal of Chemical Theory and Computation* 14.11 (Nov. 2018), pp. 5777–5786. DOI: 10.1021/acs.jctc.8b00708.

[137] M. Poberznik, M. Gunde, N. Salles, A. Jay, A. Hemeryck, N. Richard, N. Mousseau, and L. Martin-Samos. "pARTn: A Plugin Implementation of the Activation Relaxation Technique Nouveau That Takes over the FIRE Minimisation Algorithm." In: *Computer Physics Communications* 295 (Feb. 2024), p. 108961. DOI: 10.1016/j.cpc.2023.108961.

[138] Rohit Goswami. "Bayesian Hierarchical Models for Quantitative Estimates for Performance Metrics Applied to Saddle Search Algorithms." In: *AIP Advances* 15.8 (Aug. 2025), p. 85210. DOI: 10.1063/5.0283639.

[139] Rohit Goswami and Hannes Jónsson. "Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches." In: *ChemPhysChem* (Nov. 2025). DOI: 10.1002/cphc.202500730.

[140] R. A. Olsen, G. J. Kroes, G. Henkelman, A. Arnaldsson, and H. Jónsson. "Comparison of Methods for Finding Saddle Points without Knowledge of the Final States." In: *The Journal of Chemical Physics* 121.20 (Nov. 2004), pp. 9776–9792. DOI: 10.1063/1.1809574.

[141] Johannes Kästner and Paul Sherwood. "Superlinearly Converging Dimer Method for Transition State Search." In: *The Journal of Chemical Physics* 128.1 (Jan. 2008), p. 014106. DOI: 10.1063/1.2815812.

[142] Jing Leng, Weiguo Gao, Cheng Shang, and Zhi-Pan Liu. "Efficient Softest Mode Finding in Transition States Calculations." In: *Journal of Chemical Physics* 138.9 (Mar. 2013), p. 94110. DOI: 10.1063/1.4792644.

[143] Andrew Gelman. *Bayesian Data Analysis*. Third edition. Chapman & Hall/CRC Texts in Statistical Science. Boca Raton: CRC Press, 2014.

[144] Andrew Gelman, Aki Vehtari, Daniel Simpson, Charles C. Margossian, Bob Carpenter, Yuling Yao, Lauren Kennedy, Jonah Gabry, Paul-Christian Bürkner, and Martin Modrák. "Bayesian Workflow." In: *arXiv:2011.01808 [stat]* (Nov. 2020). arXiv: 2011.01808 [stat].

[145] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. "Visualization in Bayesian Workflow." In: *arXiv:1709.01449 [stat]* (June 2018). DOI: 10.1111/rssa.12378. arXiv: 1709.01449 [stat].

[146] Richard McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. 2nd ed. CRC Texts in Statistical Science. Boca Raton: Taylor and Francis, CRC Press, 2020.

[147] Paul-Christian Bürkner. "Brms: An R Package for Bayesian Multilevel Models Using Stan." In: *Journal of Statistical Software* 80.1 (Aug. 2017), pp. 1–28. DOI: 10.18637/jss.v080.i01.

[148] Marko Melander and Hannes Jónsson. “Effect of H Adsorption on the Magnetic Properties of an Fe Island on a W(110) Surface.” In: *Physical Review B* 100.17 (Nov. 2019), p. 174431. DOI: 10.1103/PhysRevB.100.174431.

[149] James E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer Texts in Statistics. New York, N.Y. ; [London]: Springer, 2007.

[150] Ayako Nakata, Jack Baker, Shereif Mujahed, Jack T. L. Poulton, Sergiu Arapan, Jianbo Lin, Zamaan Raza, Sushma Yadav, Lionel Truflandier, Tsuyoshi Miyazaki, and David R. Bowler. “Large Scale and Linear Scaling DFT with the CONQUEST Code.” In: *The Journal of Chemical Physics* 152.16 (Apr. 2020), p. 164112. DOI: 10.1063/5.0005074. arXiv: 2002.07704 [cond-mat, physics:physics].

[151] Hildo Bijl, Jan-Willem van Wingerden, Thomas B. Schön, and Michel Verhaegen. “Online Sparse Gaussian Process Regression Using FITC and PITC Approximations*.” In: *IFAC-PapersOnLine*. 17th IFAC Symposium on System Identification SYSID 2015 48.28 (Jan. 2015), pp. 703–708. DOI: 10.1016/j.ifacol.2015.12.212.

[152] Andrew Gordon Wilson and Hannes Nickisch. “Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP).” In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, July 2015, pp. 1775–1784.

[153] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. “Optuna: A next-Generation Hyperparameter Optimization Framework.” In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’19. New York, NY, USA: Association for Computing Machinery, July 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[154] Carl Edward Rasmussen. “Gaussian Processes in Machine Learning.” In: *Advanced Lectures on Machine Learning: ML Summer Schools 2003, Canberra, Australia, February 2 - 14, 2003, Tübingen, Germany, August 4 - 16, 2003, Revised Lectures*. Ed. by Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 63–71. DOI: 10.1007/978-3-540-28650-9_4.

[155] Matthew Thorpe. “Introduction to Optimal Transport.” In: ().

[156] Richard S. Varga. *Geršgorin and His Circles*. Vol. 36. Springer Series in Computational Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-642-17798-9.

[157] Carlos Echeverría, Jörg Liesen, and Reinhard Nabben. “Block Diagonal Dominance of Matrices Revisited: Bounds for the Norms of Inverses and Eigenvalue Inclusion Sets.” In: *Linear Algebra and Its Applications* 553 (Sept. 2018), pp. 365–383. DOI: 10.1016/j.laa.2018.04.025.

[158] Rachid Ababou, Amvrossios C. Bagtzoglou, and Eric F. Wood. “On the Condition Number of Covariance Matrices in Kriging, Estimation, and Simulation of Random Fields.” In: *Mathematical Geology* 26.1 (Jan. 1994), pp. 99–133. DOI: 10.1007/BF02065878.

[159] A. J. Stone. *The Theory of Intermolecular Forces*. Second edition. Oxford: Oxford University Press, 2013.