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`.

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, 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 (1.2×10^{-15} s), to capture the bond vibrations as the shortest 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 diffusive 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.

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

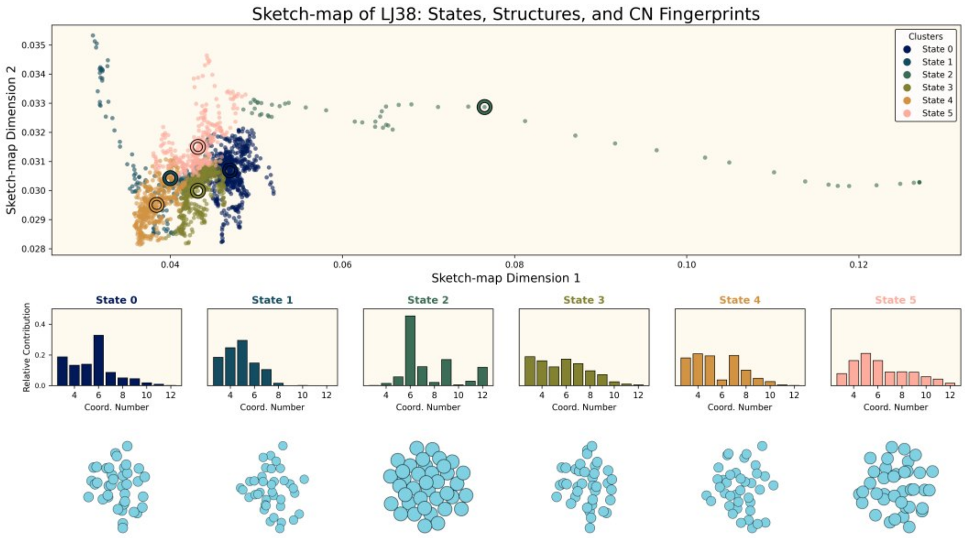


__Figure 1.1.__ The complex structural landscape of the Lennard-Jones 38-atom (LJ38) cluster is visualized using a sketch-map projection. The data is sourced from a molecular dynamics simulation thermostatted at 80.0 K, modeling an Argon cluster. Each point in the trajectory was characterized by a high-dimensional 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.

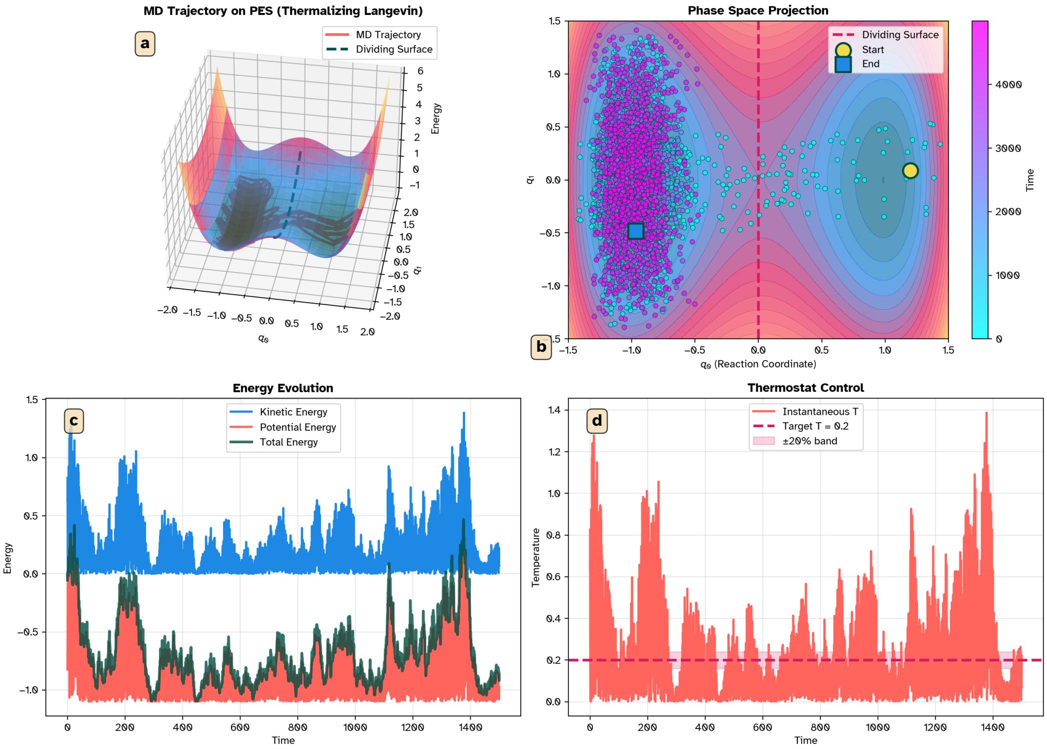


Figure 1.2. ***(a)*** *A simulated Molecular Dynamics trajectory (red line) on a double-well Potential Energy Surface, shown in 3D with the dividing surface (black dashed line) separating reactant and product regions.* ***(b)*** *Top-down projection of the same trajectory onto the reaction coordinate 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.*

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 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, however, 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.

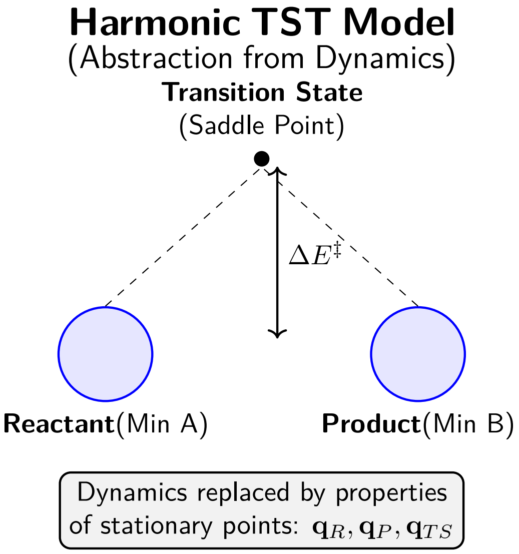

Harmonic TST Model
(Abstraction from Dynamics)
Transition State
(Saddle Point)
$\Delta E^\ddagger$
Reactant(Min A)
Product(Min B)
Dynamics replaced by properties
of stationary points: $\mathbf{q}_R, \mathbf{q}_P, \mathbf{q}_{TS}$

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.

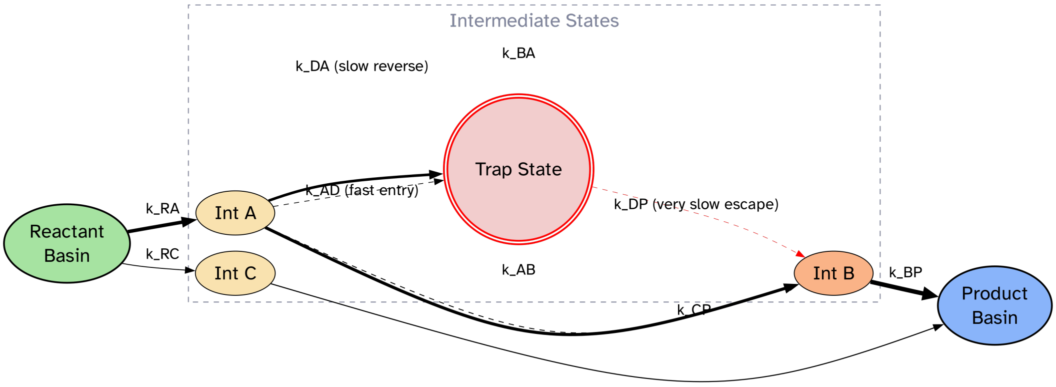


Figure 1.4. Abstraction of a Potential Energy Surface to a Discrete Kinetic Network. The continuous, high-dimensional PES is simplified into a network of states (nodes) and transitions (arrows). Each node represents a stable energy basin. Each arrow's thickness is proportional to its TST-calculated rate 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.

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 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 "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

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

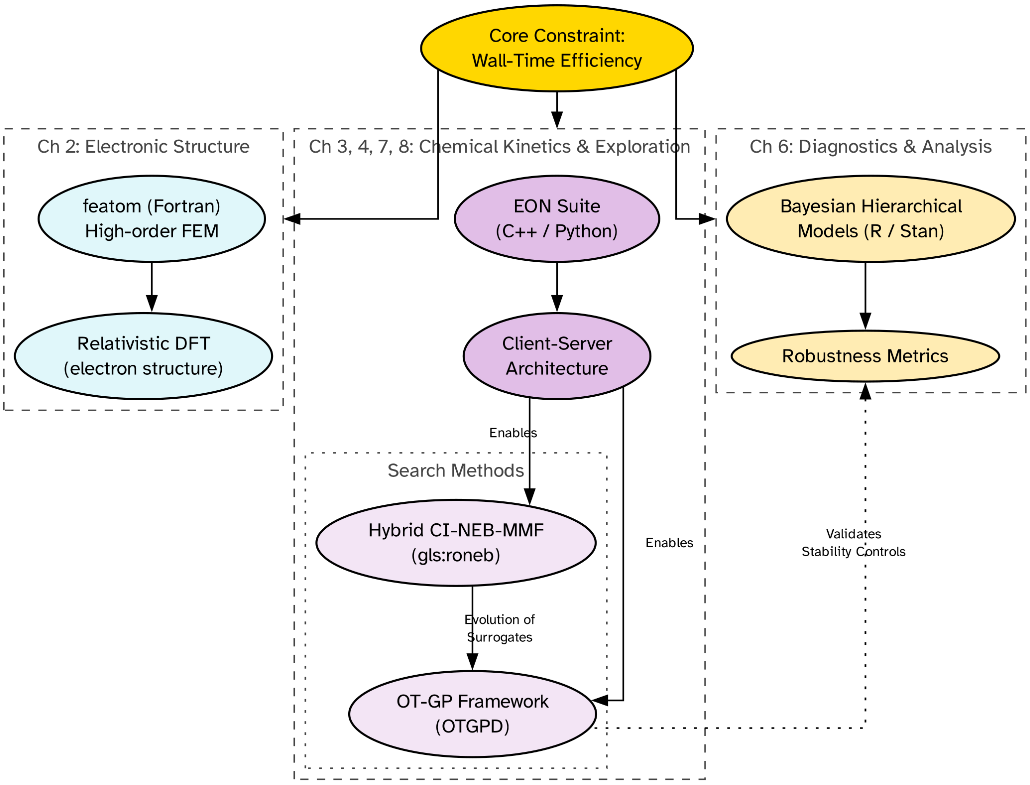


Figure 1.5. *The structural organization of this dissertation. Wall-time efficiency serves as the central constraint driving development across three distinct domains: relativistic electronic structure (cyan), chemical kinetics and exploration algorithms (purple), and statistical performance analysis (amber). Arrows indicate the flow of methodology and software evolution, highlighting how statistical insights and architectural choices drive the development of advanced search methods like OTGPD.*

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 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

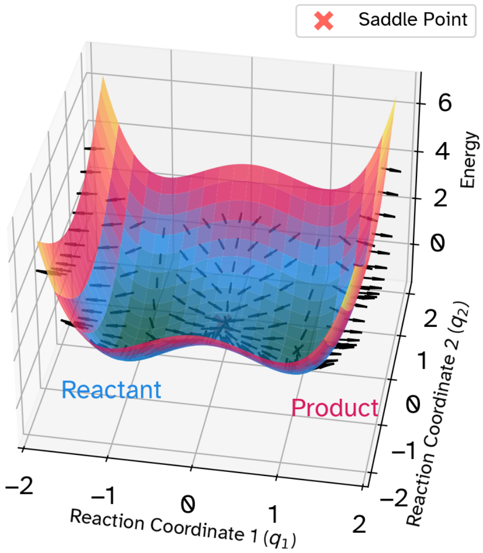


Figure 2.1. *An illustration of the forces from the dimer method for locating a first-order saddle point on a two-dimensional Potential Energy Surface. The vector field represents the effective dimer force, a transformation of the true potential gradient ($-\nabla V$). This modified force guides an optimization 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.

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 $i = k-1, \dots, k-m$ 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 $i = k-m, \dots, k-1$ do
    $\beta \leftarrow \rho_i y_i^T z$
    $z \leftarrow z + s_i(\alpha_i - \beta)$
end for
return Search direction $p_k = -z$
```

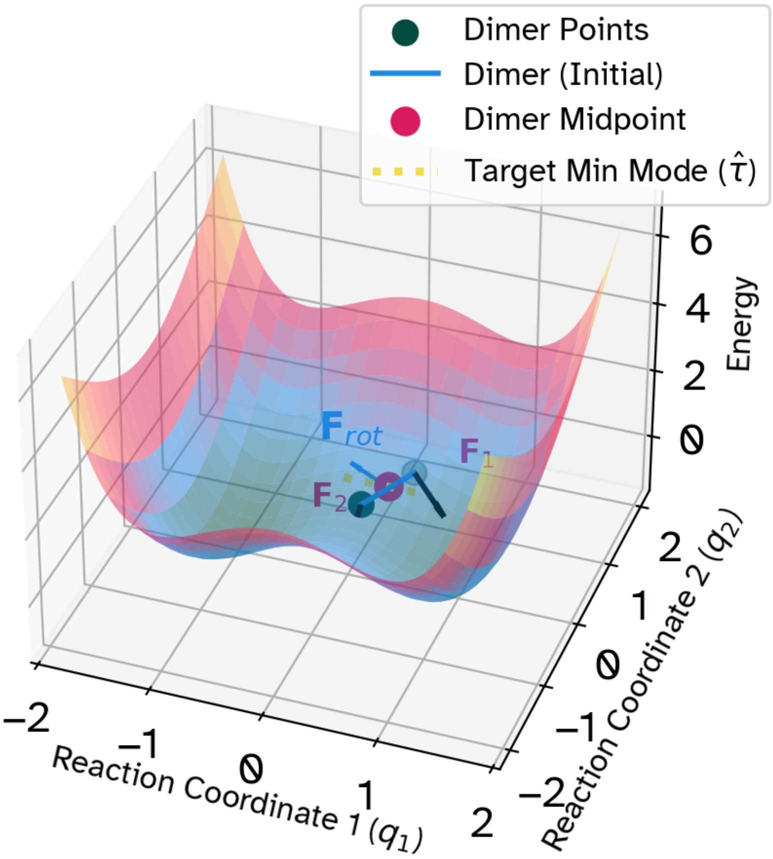


Figure 2.2. *The rotational step of the Dimer Method. The effective rotational force ($\mathbf{F}_{rot}$) is derived from the atomic forces ($\mathbf{F}_1, \mathbf{F}_2$) and applies a torque to the misaligned dimer. This torque drives the dimer's orientation to align with the minimum mode ($\hat{\tau}$),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

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 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.

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

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.

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

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

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

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," $\mathbf{R}_{\text{climb}}$) 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.

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

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:

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

This factor, α_i, ranges from 0 (when $E_i = E_{\text{max}}$) to 1 (when $E_i = E_{\text{ref}}$). 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:

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

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}):

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

Here, $\hat{\tau}_i$ 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 back-propagate 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 $\mathbf{M}$ 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 $\mathbf{M}$-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 $\mathbf{M}$ 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:

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

The full covariance matrix $\mathbf{K}$ 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:

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

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 $\mathbf{m}$ and a covariance matrix $\mathbf{K}$. 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 only partially resolves with 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 homogenizes 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 $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_M\}$ 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, $\mathcal{D}_{1/r}^2 = \sum \left(\frac{\Delta(1/r)}{l}\right)^2$:

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

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

The kernel hyperparameters $\theta = \{\sigma_f^2, \sigma_c^2, l_\psi, \dots\}$ 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 η corresponds to a proportional change in θ 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

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

are learned from the training data by maximizing the logarithm of the marginal log-likelihood (MLL):

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

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:

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

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

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

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:

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

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 $O(M^2N^2)$.

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.

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.

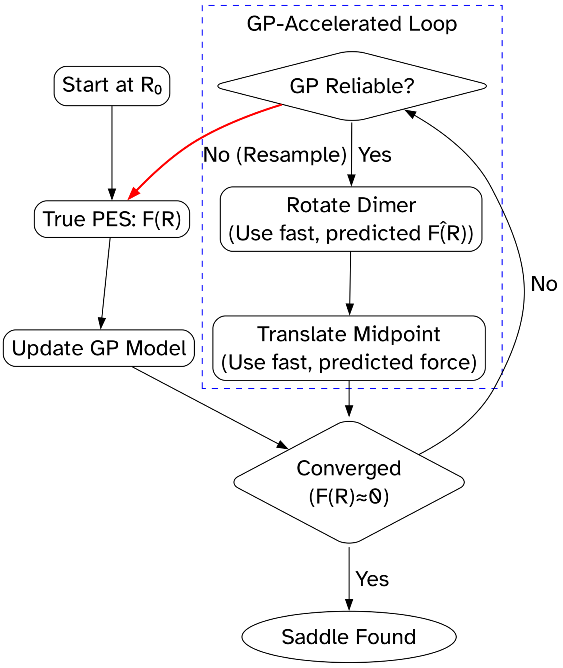


Figure 2.3. The GPDimer method.

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

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

[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 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

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

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

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

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

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

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 $\int_0^\infty P_{nl}^2(r)\,\mathrm{d}r = 1$.

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].

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

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

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

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

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, $(H + ▯c^2)^2$. 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 ($r \to 0$), 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, **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. This work 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 $Y_k^{(ab)}(r)$.

[8] gh-26 to featom

- 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 ~ 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.

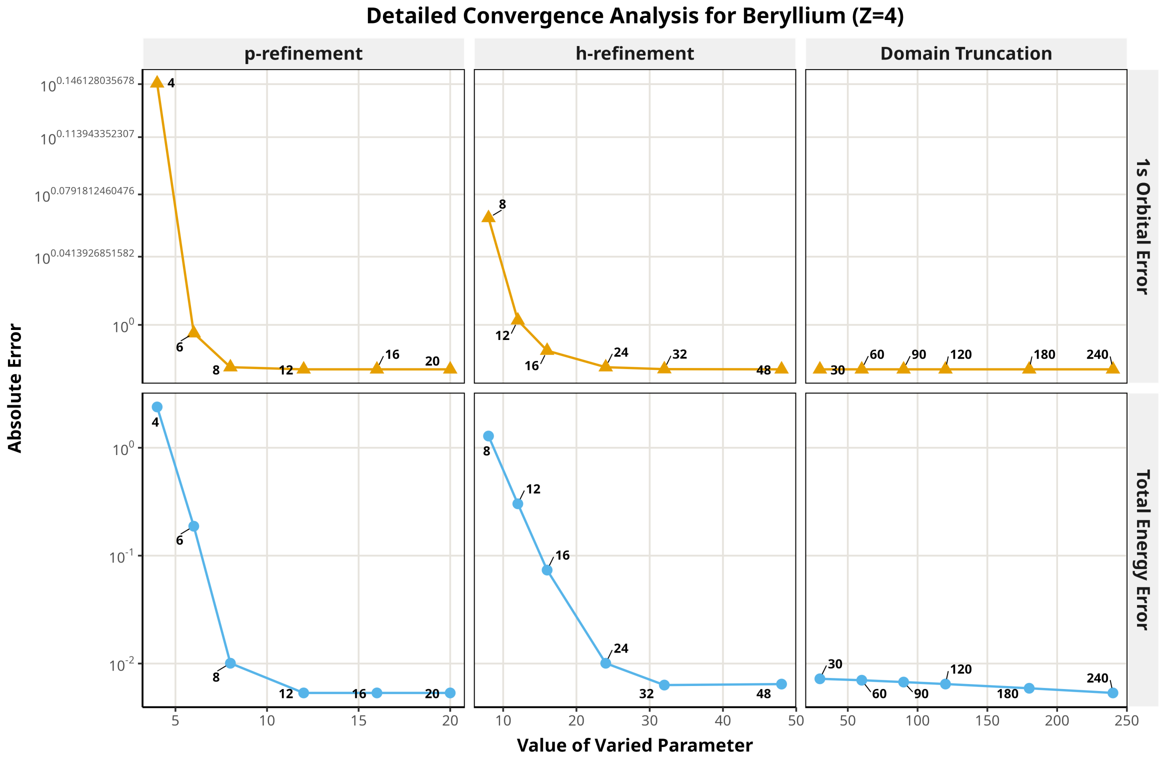


Figure 3.1. *Systematic convergence of the radial finite element solver for a Restricted Hartree-Fock calculation on a Beryllium atom (Z=4). The grid validates the two primary modes of convergence.* ***Bottom Row (Total Energy)*** *and* ***Top Row (1s Orbital)*** *show the error for three distinct refinement 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

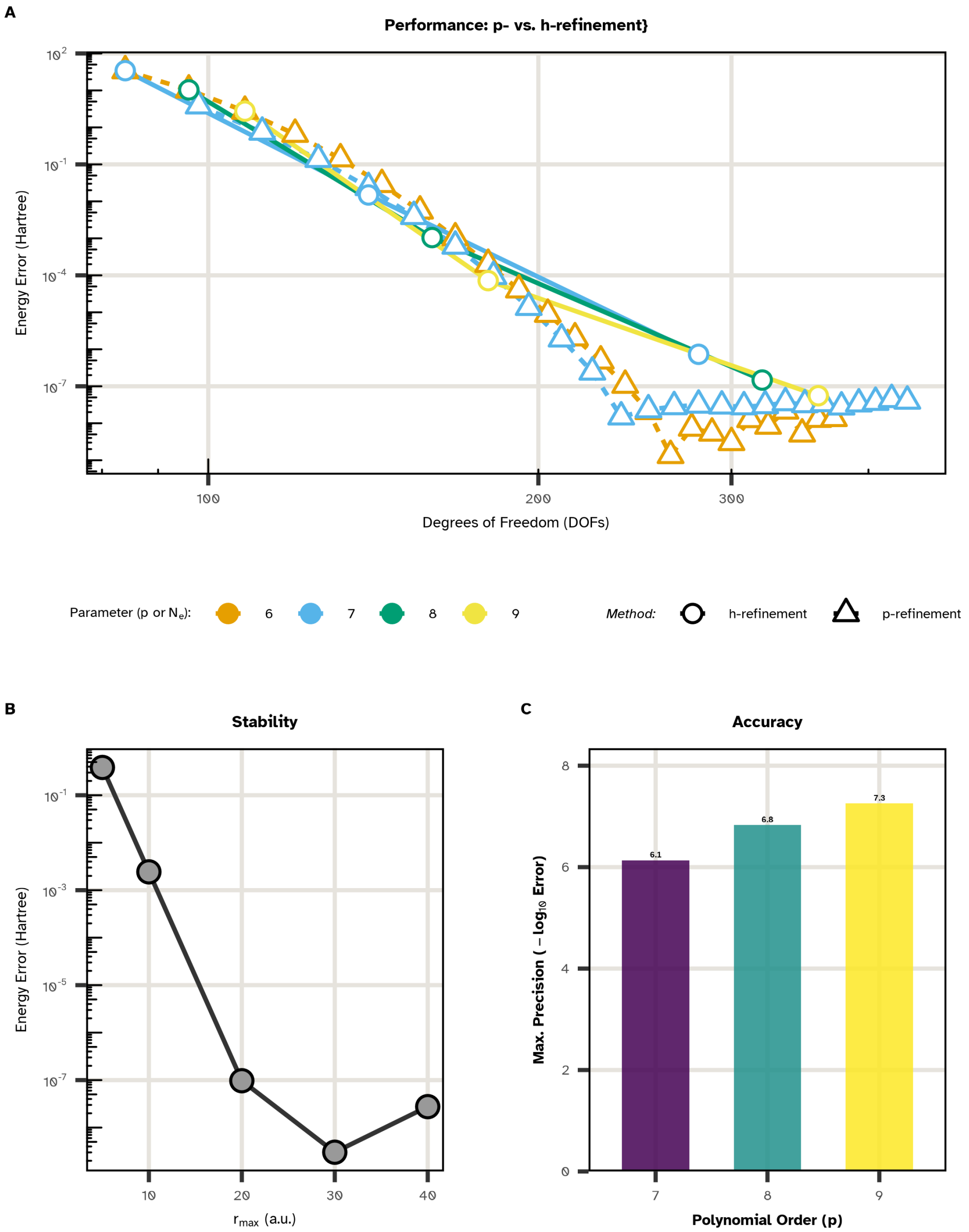


Figure 3.2. *Systematic convergence and precision of the* `featom` *finite element solver for relativistic Dirac–Kohn–Sham calculations of uranium (Z=92). (a) p- vs. h-refinement: Both p-refinement (increasing polynomial order p, colored) and h-refinement (increasing number of elements N_e) yield systematic 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.*

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 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 ($\mathcal{O}(M^3)$), 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.

We hope this clarifies that the "correctness" of a scientific result is inextricably linked

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

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.

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.

[10] inspired by `solvis`

[11] typically between 1.1 and 1.3

- $\sum_{\nu \in B}$ sums over all atomic orbitals ν on atom B.
- $P_{\mu\nu}$ is an element of the density matrix.

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].

4.2 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

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

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

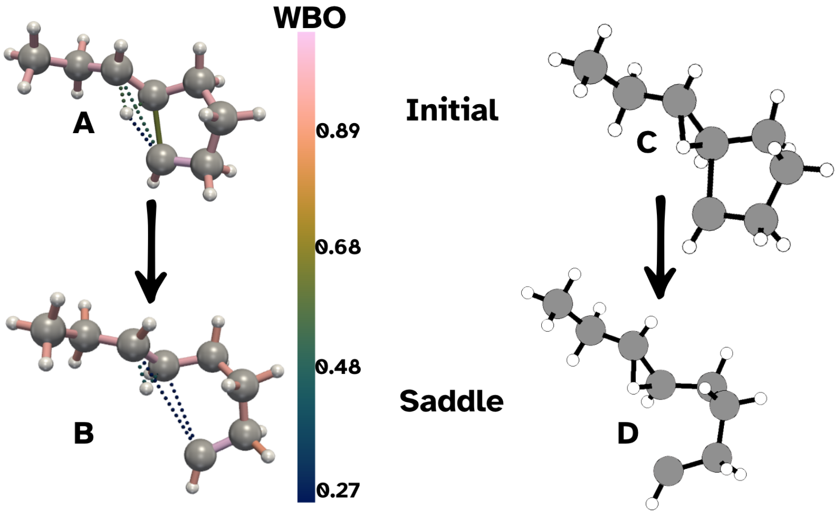


Figure 4.1. Wiberg Bond Order (WBO) analysis of a radical hydrogen transfer reaction (doublet system D004) from an initial reactant complex to the saddle point. Panels (a) and (b) visualize the system with interatomic connections colored by their WBO, where bonds are above 0.5, revealing the subtle electronic 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.

$H(s)$ 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 $\hat{\tau}_i$:

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

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:

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

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 **P** is the atom permutation and **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].

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

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

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

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 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

[15] gh-244 to EON

[16] gh-1145 to NWChem

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.

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.

4.3.3 Case Study: Isomerization of Ethylene Oxide to Acetaldehyde

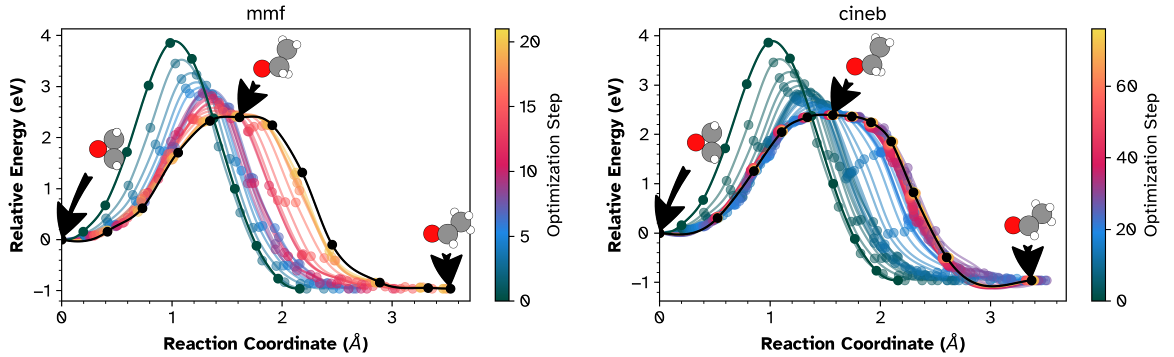
To demonstrate the effectiveness and computational efficiency of the hybrid CI-NEB-MMF method, it was applied to the isomerization reaction of ethylene oxide to acetaldehyde, the results of which are shown in Figure 4.2.

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°, a severe deviation from the ideal 109.5° 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.

[17] gh-230 to EON

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

```
Initialize NEB path $\{\mathbf{R}_0, \dots, \mathbf{R}_P\}$
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
    $F_{\text{max}} \leftarrow \max_i(|\mathbf{F}_i^{\text{NEB}}|)$
    if $F_{\text{max}} < F_{\text{MMF\_threshold}}$ 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
```



***(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.*

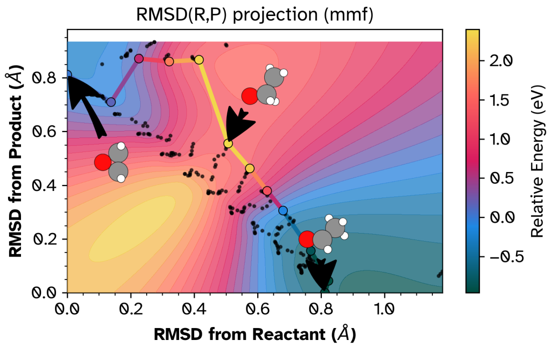


Figure 4.3. 2D landscape projection of the converged hybrid CI-NEB-MMF path for the ethoxy acetal system. The trajectory is plotted on a coordinate system of RMSD from the reactant vs. RMSD from the product. Black points represent the discrete sampling history during optimization. The interpolated energy contours (color scale in eV) reveal the topography of the potential energy surface, highlighting the minimum energy path connecting the metastable reactant to the stable product via a high-energy transition state.

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

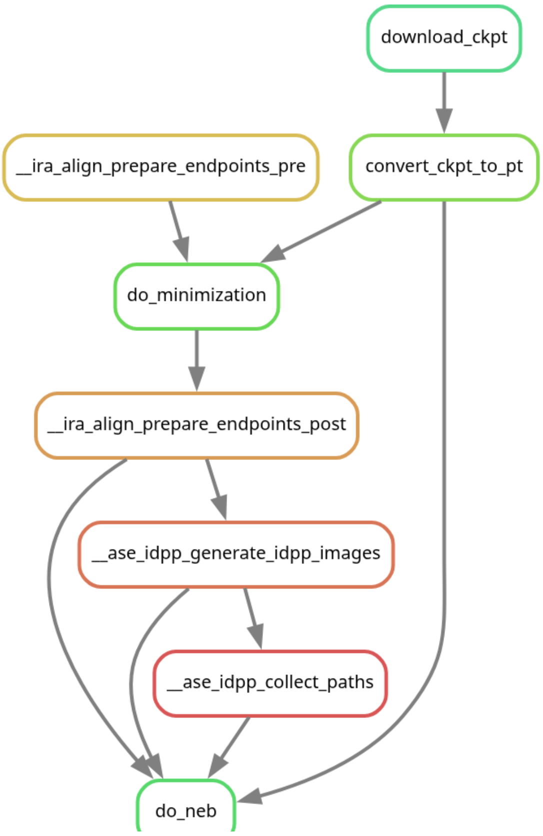

download_ckpt
__ira_align_prepare_endpoints_pre
convert_ckpt_to_pt
do_minimization
__ira_align_prepare_endpoints_post
__ase_idpp_generate_idpp_images
__ase_idpp_collect_paths
do_neb

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

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.

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

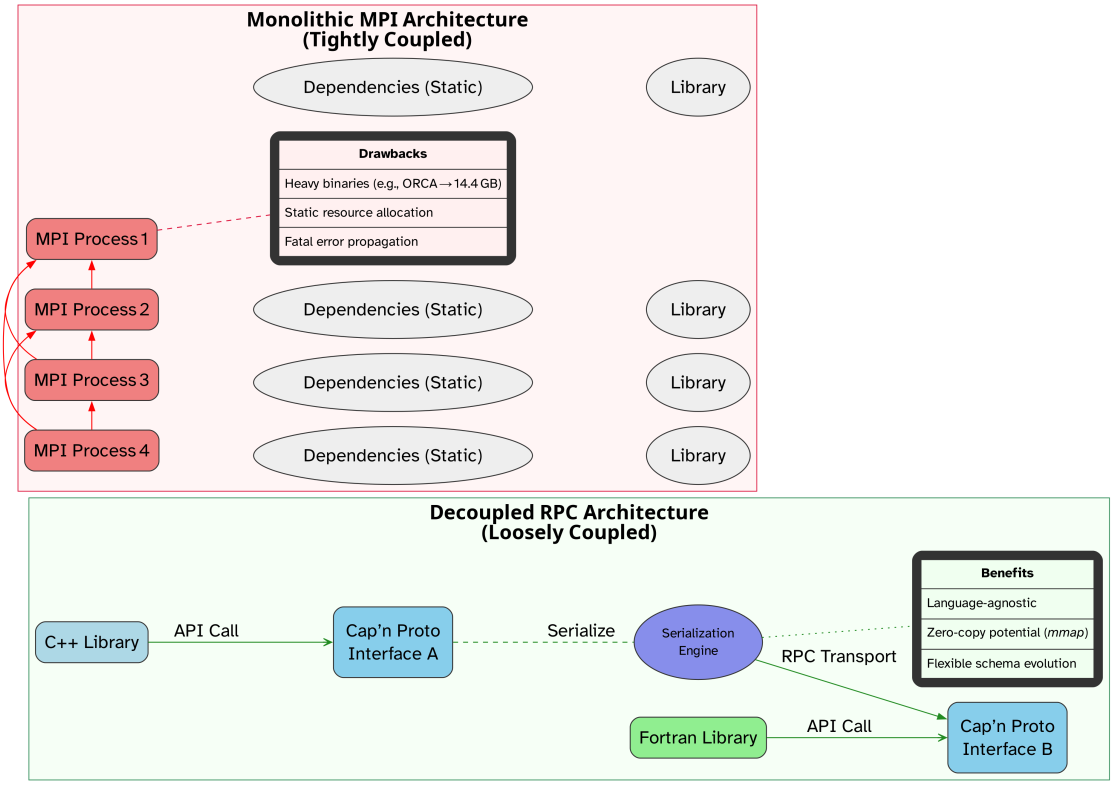


Figure 4.5. *A schematic comparison of two software architecture paradigms in scientific computing. (Left) The traditional monolithic model, based on MPI, statically links all dependencies into large, identical processes. This tight coupling results in heavy binaries, static resource allocation, and 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.*

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.

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);
}
```

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. 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 chap-

[18] submission in progress to Frontiers chemistry.

ter [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.

[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.

[20] always a moving target

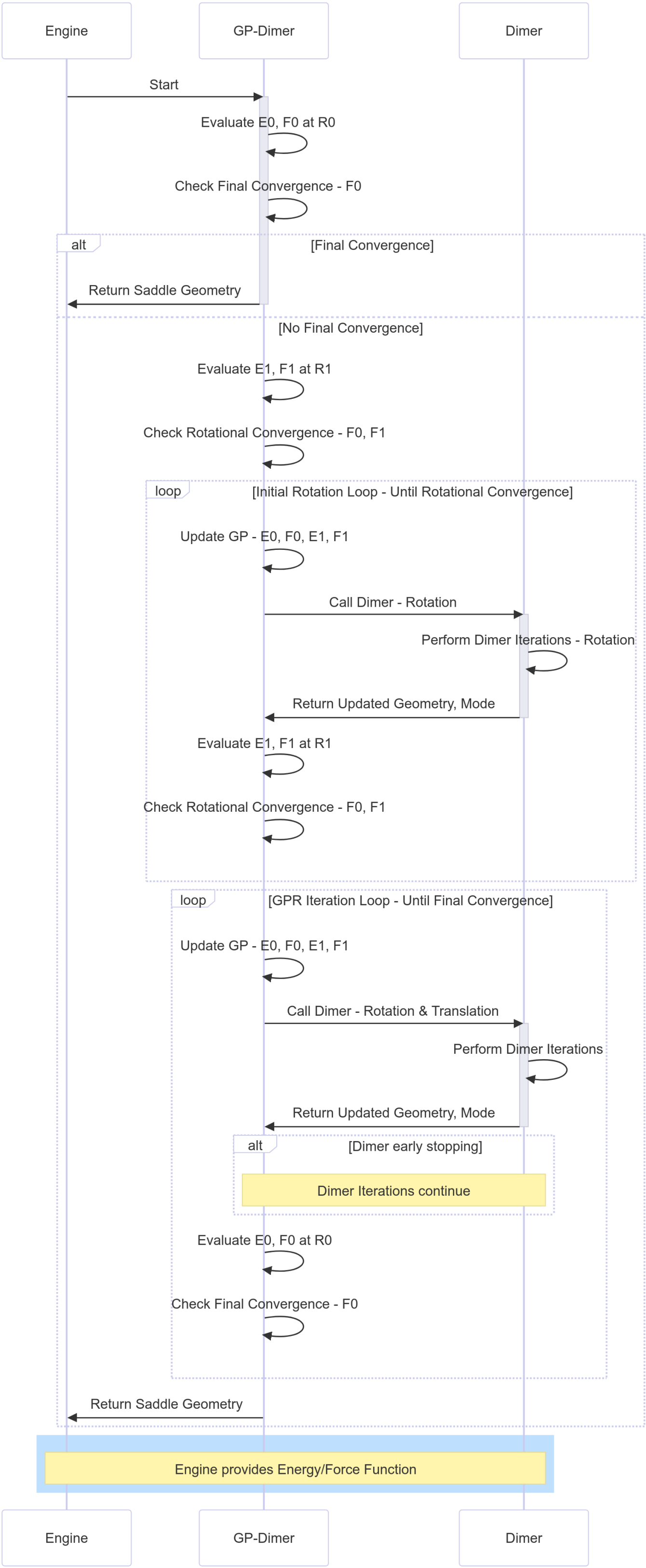

Engine
GP-Dimer
Dimer
Start
Evaluate E0, F0 at R0
Check Final Convergence - F0
alt
[Final Convergence]
Return Saddle Geometry
[No Final Convergence]
Evaluate E1, F1 at R1
Check Rotational Convergence - F0, F1
loop
[Initial Rotation Loop - Until Rotational Convergence]
Update GP - E0, F0, E1, F1
Call 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.*

The algorithm itself proceeds as described in Alg. 3.

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: $\mathbf{k}^* = [k(\mathbf{x}^*, \mathbf{x}_1), \dots, k(\mathbf{x}^*, \mathbf{x}_n)]^T$.
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.

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].

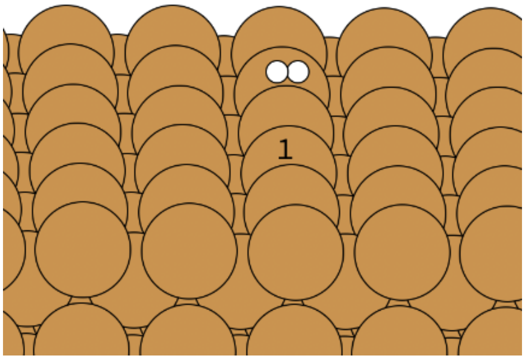


Figure 5.2. Hydrogen molecule dissociating on a copper slab. Including the two nearest copper atoms to any moving hydrogen atom in the active set proved to further accelerate convergence, achieving a runtime of just 4.3 seconds.

While effective, the common approach of using a simple radial cutoff to define the local environment can fail in realistic systems [130]. 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 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 choice of the number of neighbors is a critical hyperparameter that directly controls the computational scaling of the method. However, sensitivity analyses show that simply increasing the number of atoms does not necessarily improve model accuracy, making a careful and parsimonious selection essential for both efficiency and predictive power.

5.3 Data dredging

Throughout this thesis, a data set of small organic molecules curated by Hermes, Sargsyan, Najm, and Zádor [131] has been used. This consists of 500 initial configurations of small gas-phase organic molecules, ranging from 7 to 25 atoms.

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

[21] From $\text{gpr}_{\text{optim}}$ at commit a4d1fdaa0ed943d0c9e8b2931db12a4148be0ba4

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

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 [131]. 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.

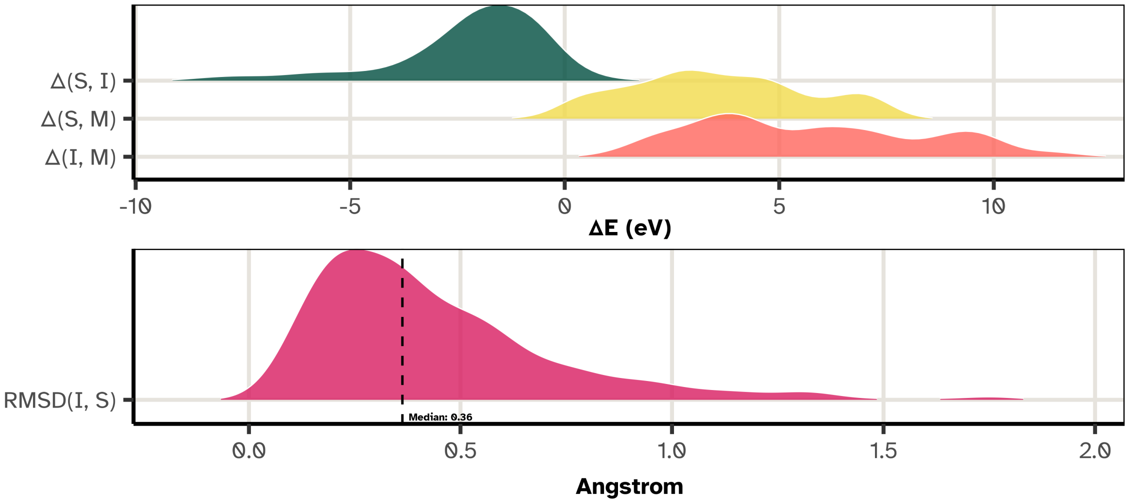


__Figure 5.3.__ Distribution of Energetic and Structural Properties for the Sella Transition State Dataset. __(Top)__ Probability density of energy differences: Δ(S, I) between the final saddle and initial geometry, Δ(S, M) between the final saddle and the minimized initial geometry, and Δ(I, M) between 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. 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, whereas Sella employs internal coordinates, which are generally considered more suitable for the varied stiffness of molecular degrees of freedom. 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 consistently requires fewer HF calculations. Its efficiency advantage diminishes as the initial structural deviation increases. In the intermediate range, between 0.6 Å and 1.2 Å RMSD, the methods perform almost identically. Here, Sella's internal coordinates likely aid the more complex optimization. For the most difficult cases, with initial displacements greater than 1.2 Å, the GPDimer again demonstrates superior efficiency. This suggests a greater robustness when converging from a poor initial structure.

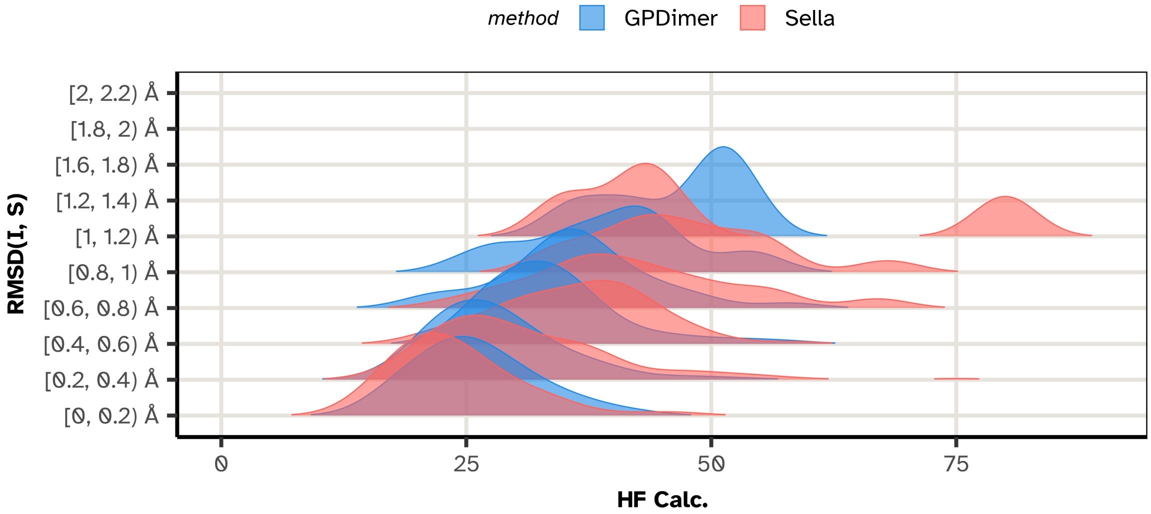


Figure 5.4. *Ridgeline plot showing the distribution of Hartree-Fock (HF) calculation counts required for convergence for the GPDimer (blue) and Sella (red) methods. The data fall into bins according to the root-mean-square deviation (RMSD) between the initial and final saddle geometries. At low RMSD 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. For the first system in the benchmark, `singlet 000`, a 16-atom acyclic ether (C_5OH_{10}), both `sella` and GPDimer calculations are launched until convergence. 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 [132] we must seek alternative ways to assess the quality. To 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).

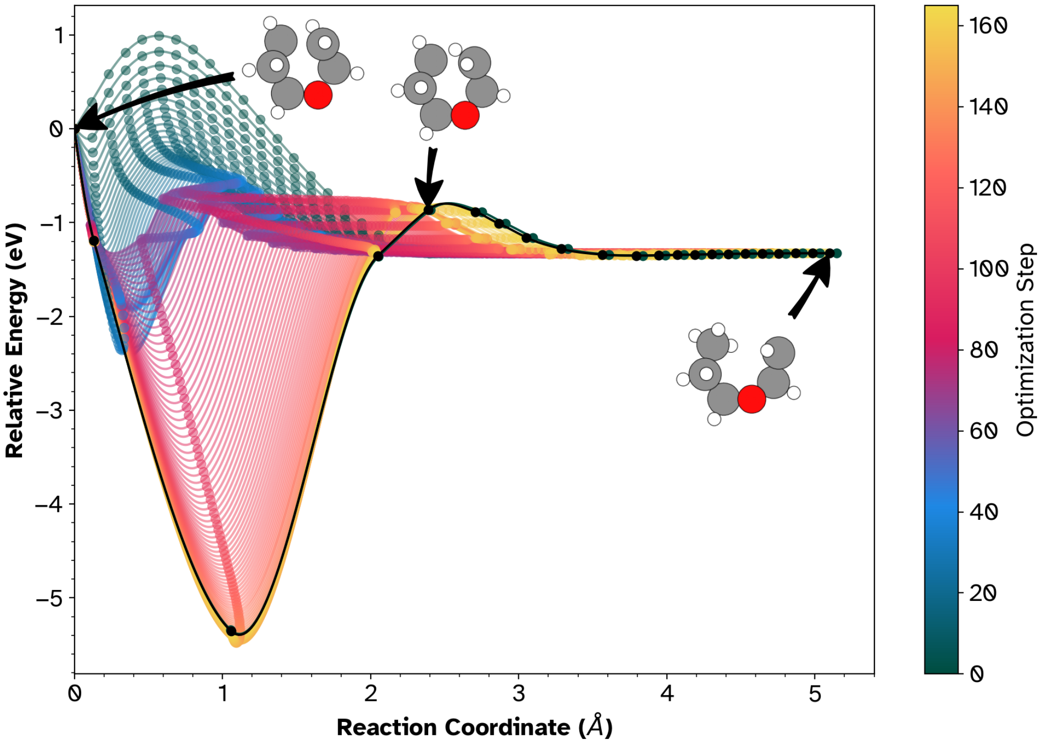


Figure 5.5. *The optimization history of an NEB connecting the initial reactant (left, $s = 0$) to the Sella-located saddle point (right, $s \approx 5.2$ Å). The path is fairly baroque, first relaxing barrierlessly into a deep intermediate minimum ($E \approx -5.3$ eV). It then climbs over a newly-identified transition state (an inflection at $s \approx 2.4$ Å) to reach the Sella saddle.*

The trajectory reveals a landscape far more complex than a simple ascent. The path does not lead directly to the Sella saddle. Instead, it first follows a steep, barrier-free relaxation from the high-energy reactant ($E = 0$ eV) into a previously uncharacterized, deep intermediate minimum at $s \approx 1.1$ Å ($E \approx -5.3$ eV). From this newly discovered basin, the path must then climb out over a newly identified transition state (the inflection point at $s \approx 2.4$ Å, $E \approx -0.9$ eV) to finally reach the Sella saddle.

The 2D landscape projection (Figure 5.6), generated using the methods [122] from Sec. 4.2, confirms this topography. The path (orange trajectory) traverses a ridge separating the reactant plateau (yellow contours) from the deep product valley (dark teal contours). Crucially, the saddle point located by the GPDimer (marked by the white star) lies precisely at the coincides with the transition state for this slice of the energy surface. 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. The result underscores the necessity of characterizing the local gradient manifold when sampling from high-energy initial configurations.

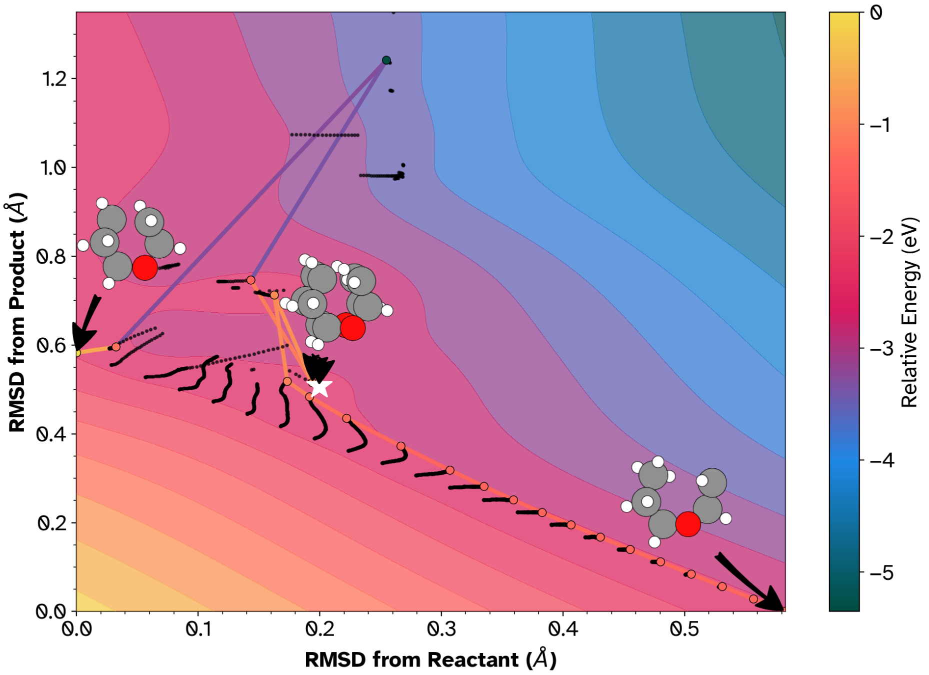


Figure 5.6. A 2D landscape projection of the MEP connecting the initial reactant (top-left) to the deep intermediate minimum (bottom-right). The white star explicitly marks the GPDimer saddle, which coincides exactly with the true transition state on this path, separating the initial configuration from the deep minimum.

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

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 [134]. 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, $\hat{\boxtimes}_k = \mathbf{F}_k^{\perp}/|\mathbf{F}_k^{\perp}|$. A new trial orientation, $\hat{\mathbf{N}}(\phi)$, is generated by a rotation within this plane:

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

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

Though [135] 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 [136].

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 [137, 138, 139, 140].

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` [141]. 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 [142] 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.

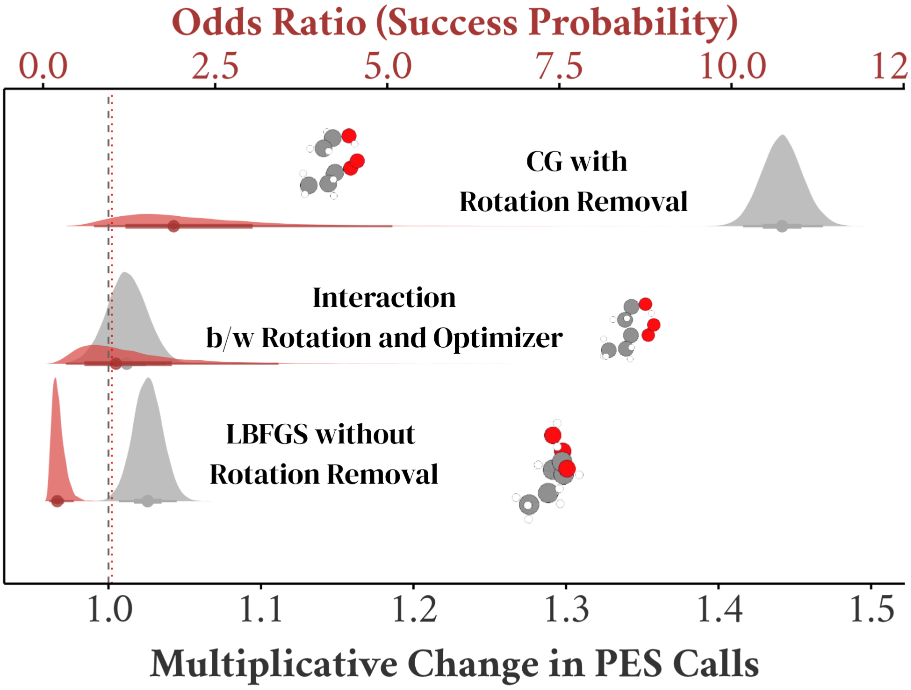


Figure 6.1. *Posterior distributions from generalized linear mixed-effects models showing the effects of algorithmic choices on computational cost (multiplicative change in PES calls, grey) and convergence probability (odds ratio, red). The analysis compares variants to a baseline of the Conjugate Gradient (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 [133].*

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.

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

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.

As time is a continuous and strictly positive quantity, we employed a Gamma generalized linear mixed model, the appropriate distributional choice for such data. The model specification is shown in Eq. 58.

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

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.

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

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 [142] 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 Rotation removal

Though quaternions may not have been best implemented EON, we return to the concept of rotations as implemented in Goswami and Jónsson [58].

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 $\delta\mathbf{r}_i = \delta\omega \times \mathbf{r}'_i$. 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 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, 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.

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 [143]. 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 $\mathcal{O}(MN^2)$, 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 softare design perspective is known to unlock linear scaling in computational chemistry [144].

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.

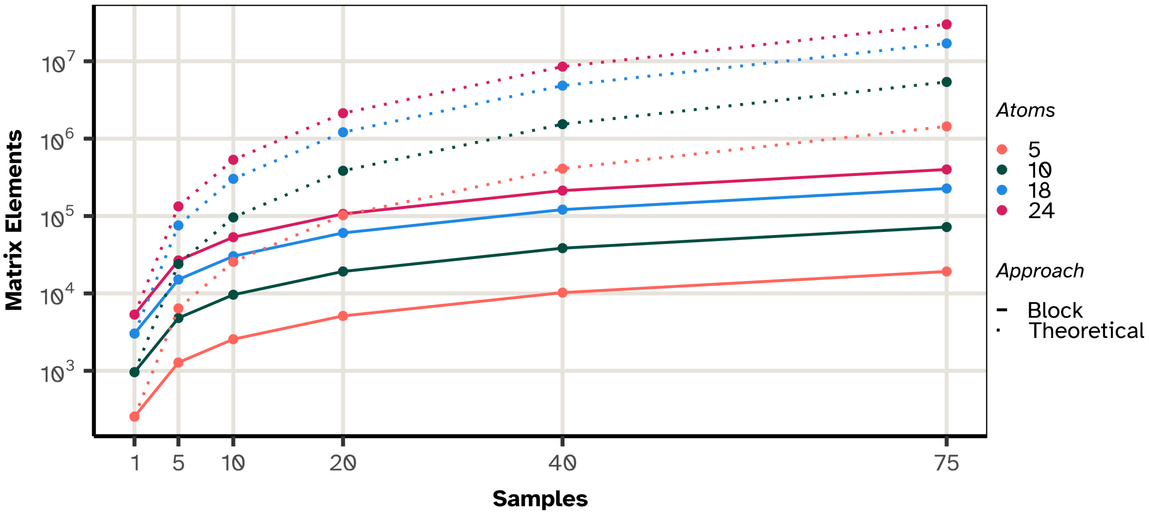


Figure 7.1. Kernel matrix element scaling for Gaussian Process training with energy and force data. Solid lines (with points) denote the practical block matrix ($M(3N+1) \times (3N+1)$), while dotted lines denote the theoretical full kernel ($(M(3N+1))^2$). Values are log-scale versus the number of training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

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:

$$\mathbf{L}_{n+1} = \begin{pmatrix} \mathbf{L}_n & \mathbf{0} \\ \mathbf{l}^T & l_d \end{pmatrix}, \quad \mathbf{L}_n \mathbf{l} = \mathbf{k}, \quad l_d = \sqrt{k_{dd} - \mathbf{l}^T \mathbf{l}}. \tag{66}$$

In the `gpr_optim` this approach is not pursued because the formulation of a rank one update proceedure 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, θ^*, requires maximizing the marginal likelihood, but the covariance matrix $\mathbf{K}_\theta$ is a function of the hyperparameters. Each candidate set of hyperparameters θ_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 θ^* 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 [145, 48, 146] 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. 71)
    ▷ Predict gradient at current location
    $\hat{\mathbf{g}}_t \leftarrow \nabla_{\mathbf{x}_t} \mu(\mathbf{x}_t)$ via Eq. 72
    ▷ 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. 74)
while $t \leq T_{\max}$ and not converged do
                                  ▷ Prune: retain only nearby observations
    $\mathcal{D}_t^{\text{pruned}} \leftarrow \{(\mathbf{x}_i, u_i, \nabla u_i) \in \mathcal{D}_t : \|\mathbf{x}_i - \mathbf{x}_t\| \leq r_p\}$ (Eq. 73)
                                  ▷ 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. 72
                                  ▷ Line search for step size
    $\mathbf{x}_{t+1}, \eta_t \leftarrow$ 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:

$$U(\mathbf{x}) = (a - x_1)^2 + b(x_2 - x_1^2)^2 \tag{67}$$

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

The gradient is:

$$\nabla U(\mathbf{x}) = \begin{pmatrix} -2(a - x_1) - 4bx_1(x_2 - x_1^2) \\ 2b(x_2 - x_1^2) \end{pmatrix} \tag{68}$$

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:

$$\mathcal{D}_t = \{(\mathbf{x}_i, u_i, \nabla u_i) : i = 1, \dots, n_t\} \tag{69}$$

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:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right) \tag{70}$$

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

$$\mu(\mathbf{x}_*) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y} \tag{71}$$

where $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \dots, k(\mathbf{x}_*, \mathbf{x}_{n_t})]^T$ 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:

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

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:

$$\mathcal{D}_t^{\text{pruned}} = \{(\mathbf{x}_i, u_i, \nabla u_i) \in \mathcal{D}_t : \|\mathbf{x}_i - \mathbf{x}_t\| \leq r_p\} \tag{73}$$

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

$$r_p = \alpha \ell \tag{74}$$

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. 67). 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.

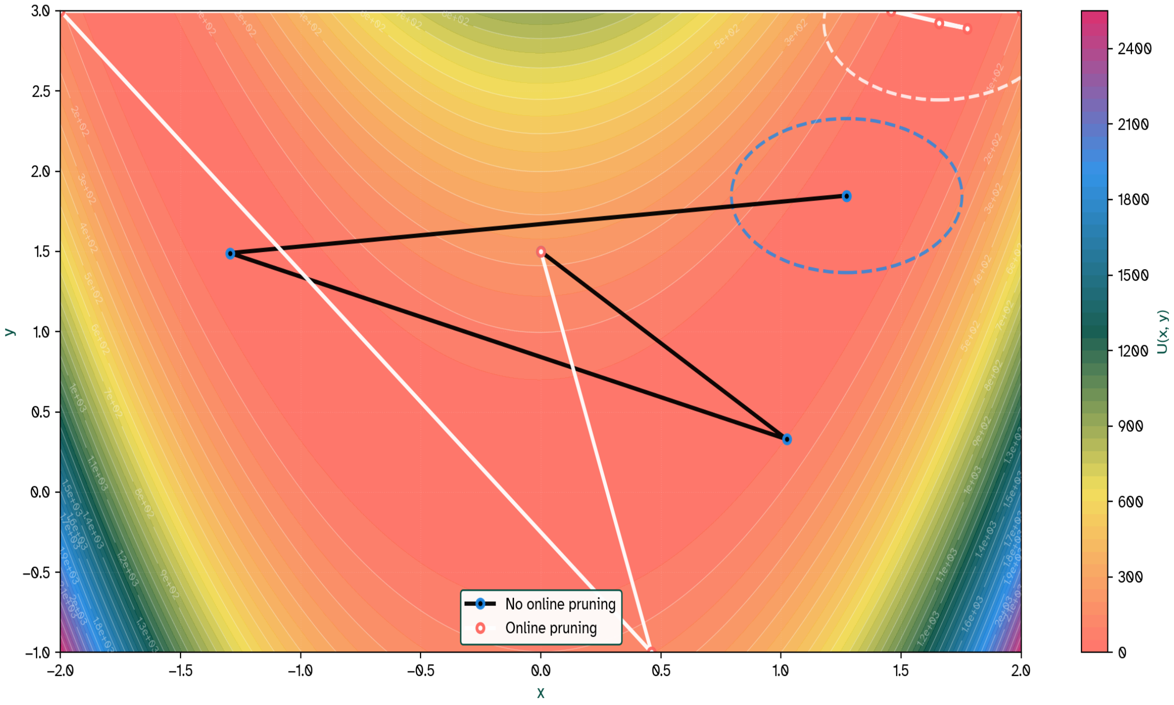


Figure 7.2. Online pruning induces trajectory divergence in GP-guided optimization of the Rosenbrock function (Eq. 67). The landscape is shown with contours. Black path (Algorithm 5): all observations retained, shown as white circles. White path (Algorithm 6): online pruning applied; white circles denote 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. 74). This localization shifts the gradient estimate $\hat{\mathbf{g}}_t$ (Eq. 72), 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. 70) in GP regression assigns non-negligible weight to observations at distances comparable to the length-scale 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. 71) and gradient prediction (Eq. 72) 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. 73). 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. 73) and inference (Eq. 71, 72) 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 $\theta_0 = (\sigma_f^0, \ell^0, \sigma_n^{\mathrm{f},0}, \sigma_n^{\mathrm{d},0})$, 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 $\theta_0$ (no optimization)
                                                  ▷ Predict on dense probe grid
    $\{\mu(\mathbf{z}), \sigma^2(\mathbf{z})\}_{z \in \mathrm{GRID}} \leftarrow$ GP.predict(GRID) via Eq. 71, ??
                                                  ▷ Evaluate global and split metrics
    $\mathrm{RMSE}_t \leftarrow \sqrt{\frac{1}{|\mathrm{GRID}|}\sum_{\mathbf{z}}(\mu(\mathbf{z}) - U_{\mathrm{true}}(\mathbf{z}))^2}$
    $\{\mathrm{RMSE}_{\mathrm{in},t}, \mathrm{RMSE}_{\mathrm{out},t}\} \leftarrow \mathrm{SplitBySupport}(\mu, U_{\mathrm{true}}, \mathrm{dists}, r_p = 1.5\ell^0)$
    $\mathrm{Cov}_{1\sigma,t} \leftarrow \frac{1}{|\mathrm{GRID}|}\sum_{\mathbf{z}} \mathbb{1}[|\mu(\mathbf{z}) - U_{\mathrm{true}}(\mathbf{z})| \leq \sigma(\mathbf{z})]$
                                                  ▷ 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 $t = 1, \ldots, T$
```

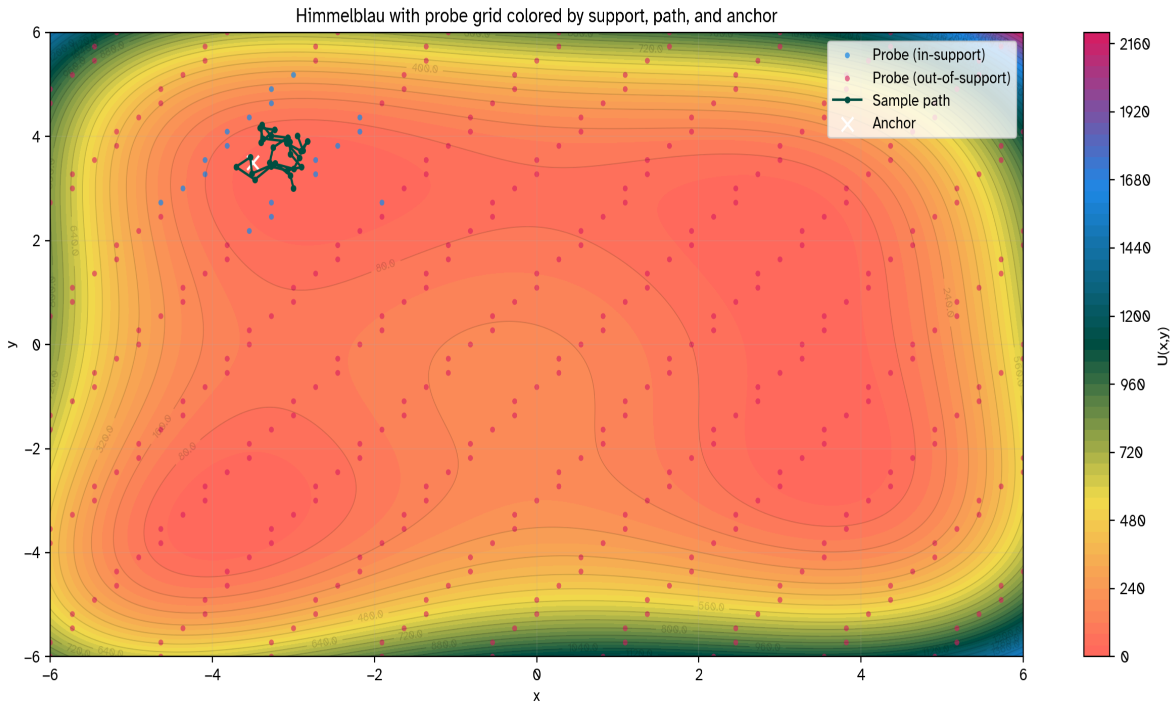


Figure 7.3. Himmelblau surface ($T = 28$ observations). The sample path (teal line) explores the landscape in a local random walk. Probe grid points are colored: blue (in-support, within radius $r_p = 1.5\ell^*_{reopt}$) and magenta (out-of-support, beyond r_p). The anchor point (white ×) is placed within 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 $\theta_0$, hyperparameter ranges $\Theta$, search trials $N_{\text{trials}}$, sample path $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$
**Output**: predictions $\{\mu_t, \sigma_t^2\}_{t=1}^T$, RMSE and coverage metrics, optimized $\{\theta_t^*\}_{t=1}^T$
Initialize: $\mathcal{D}_1 \leftarrow \{(\mathbf{x}_1, U(\mathbf{x}_1), \nabla U(\mathbf{x}_1))\}$, $\theta^* \leftarrow \theta_0$
$r_p \leftarrow 1.5\ell^0$                                  $\triangleright$ Support radius (will adapt with $\ell_t^*$)
**for** $t = 1$ to $T$ **do**
    $\triangleright$ **Re-optimize** hyperparameters via random search on marginal likelihood
    $\theta^* \leftarrow \arg\max_{\theta \in \Theta} p(\mathbf{y}|\mathcal{D}_t, \theta)$ via Eq. ?? (Eq. 77)
    Extract $\ell_t^* \leftarrow$ lengthscale$(\theta^*)$; $r_p \leftarrow 1.5\ell_t^*$        $\triangleright$ Update support radius
                                  $\triangleright$ Fit GP with re-optimized hyperparameters
    Fit GP to $\mathcal{D}_t$ using $\theta^*$
                                            $\triangleright$ Predict on dense probe grid
    $\{\mu(\mathbf{z}), \sigma^2(\mathbf{z})\}_{z \in \text{GRID}} \leftarrow$ GP.predict(GRID) via Eq. 71, ??
                  $\triangleright$ Evaluate global and split metrics (using re-optimized $r_p$)
    $\text{RMSE}_t \leftarrow \sqrt{\frac{1}{|\text{GRID}|} \sum_{\mathbf{z}} (\mu(\mathbf{z}) - U_{\text{true}}(\mathbf{z}))^2}$
    $\{\text{RMSE}_{\text{in},t}, \text{RMSE}_{\text{out},t}\} \leftarrow \text{SplitBySupport}(\mu, U_{\text{true}}, \text{dists}, r_p)$
    $\text{Cov}_{1\sigma,t} \leftarrow \frac{1}{|\text{GRID}|} \sum_{\mathbf{z}} \mathbb{1}[|\mu(\mathbf{z}) - U_{\text{true}}(\mathbf{z})| \leq \sigma(\mathbf{z})]$
                                          $\triangleright$ 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 $t = 1, \dots, T$; $\{\theta_t^*\}$
```

For the Himmelblau function

$$U(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \tag{75}$$

with gradient:

$$\nabla U(\mathbf{x}) = \begin{pmatrix} 4x_1(x_1^2 + x_2 - 11) + 2(x_1 + x_2^2 - 7) \\ 2(x_1^2 + x_2 - 11) + 4x_2(x_1 + x_2^2 - 7) \end{pmatrix} \tag{76}$$

. 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:

$$\theta^* = \arg\max_{\theta \in \Theta} p(\mathbf{y}|\mathcal{D}_t, \theta) \tag{77}$$

Search ranges are:

- $\ell \in [0.25, 3.0]$ (lengthscale)
- $\sigma_f \in [0.5, 5.0]$ (signal variance)

- $\sigma_n^{\mathrm{f}} \in [10^{-5}, 10^{-1}]$ (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.

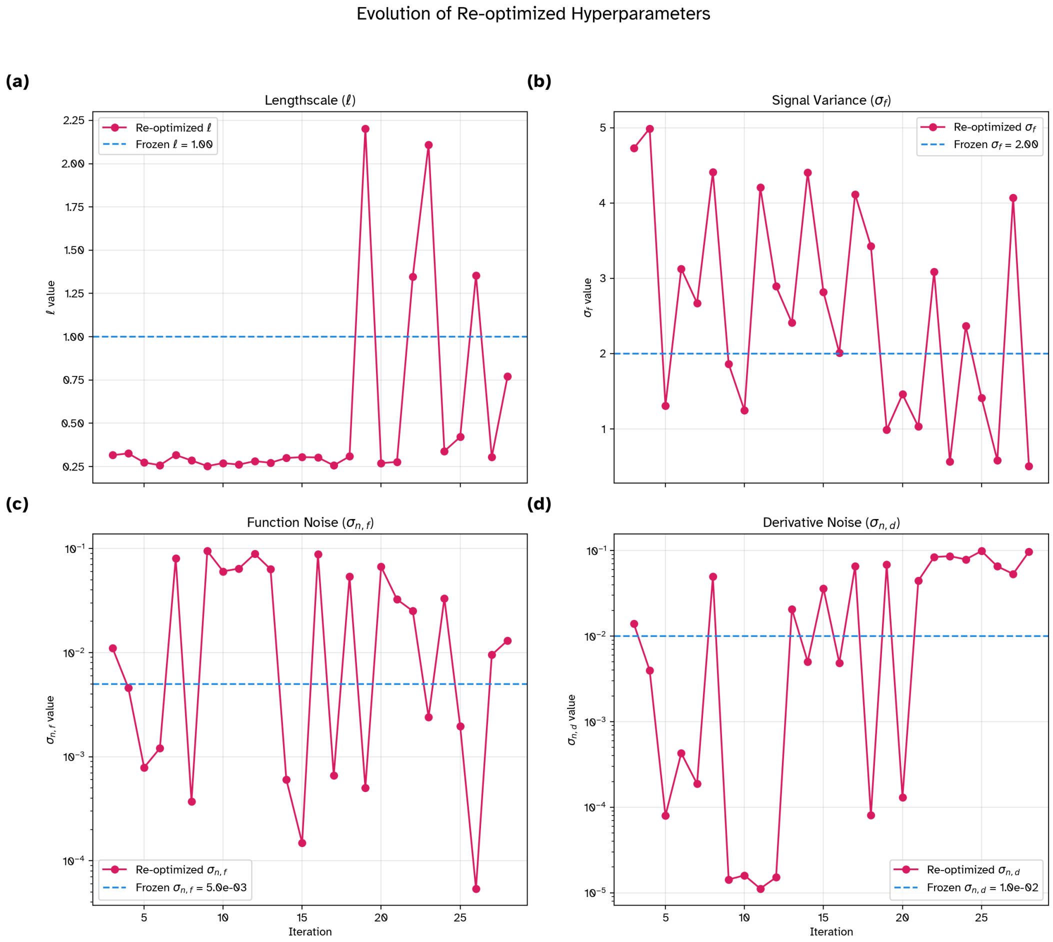


Figure 7.4. *Hyperparameter re-optimization traces. The plots show the values of the lengthscale (ℓ), signal variance (σ_f), function noise ($\sigma_{n,f}$), and derivative noise ($\sigma_{n,d}$) chosen by maximizing the marginal log-likelihood at each step. The dashed blue lines indicate the constant 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. 75). $T = 28$ 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 us-

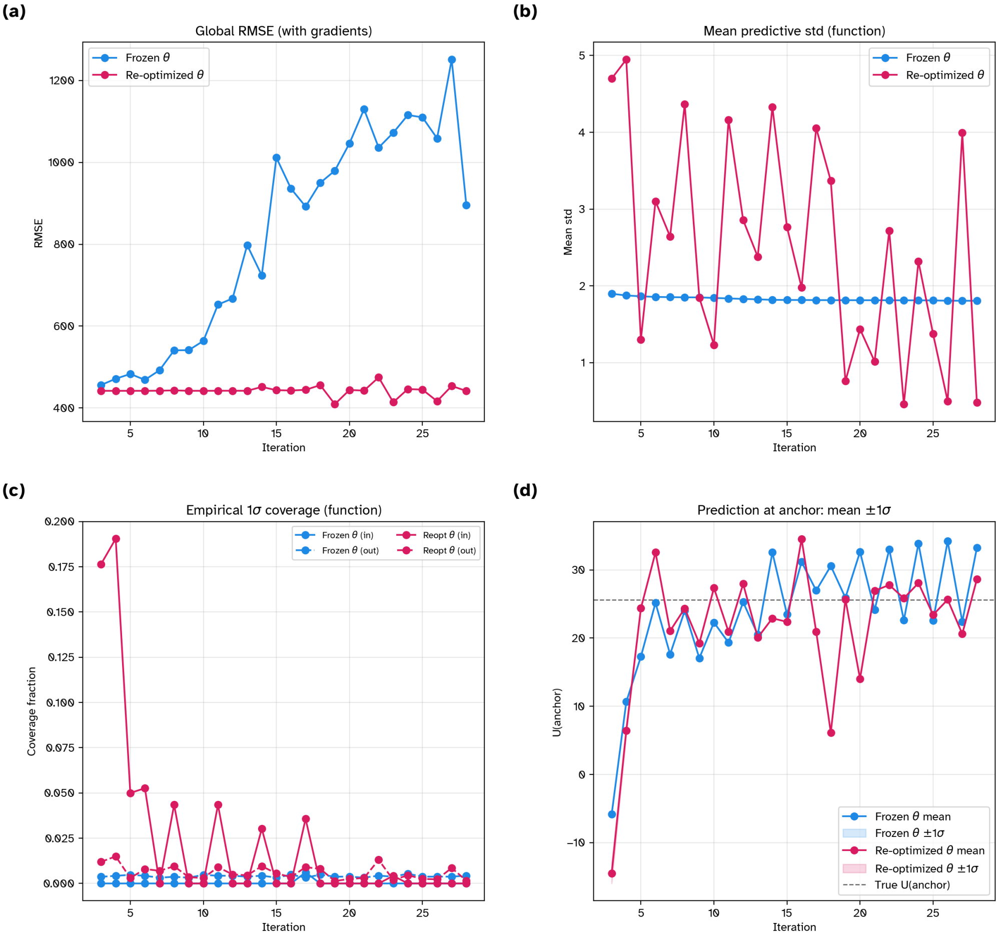


Figure 7.5. *Hyperparameter re-optimization effects on accuracy and calibration. (a) Global RMSE: The Root Mean Squared Error over the entire probe grid. The re-optimized model consistently achieves a slightly lower (better) RMSE than the frozen model, suggesting superior global accuracy.(b) Mean Predictive 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 $\pm 1\sigma$ 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 $\pm 1\sigma$ 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.*

ing derivative information. While re-optimization can improve global poinit-wise accuracy metrics, it does so by sacrificing the integrity of the model's variance estimates. The resulting model becomes 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×.

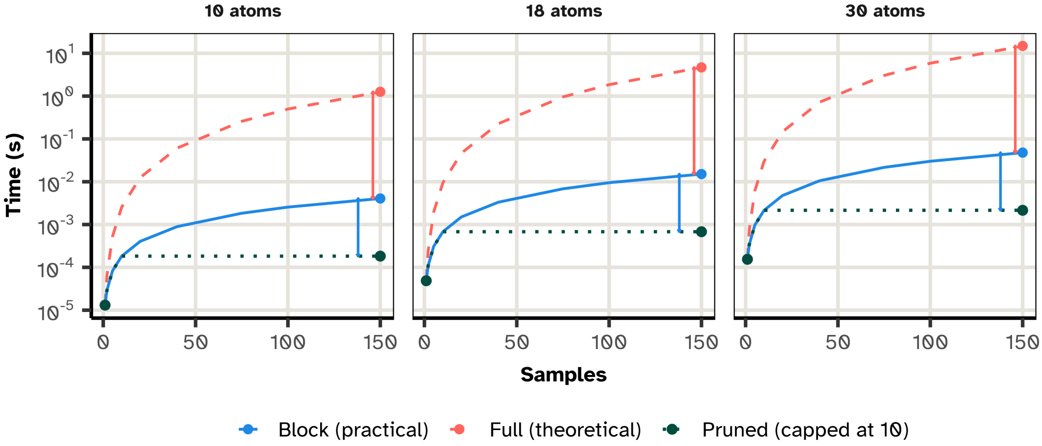


Figure 7.6. Computation time scaling with data-driven pruning. Three strategies are compared: full theoretical kernel (dashed red, $(M(3N+1))^2$ elements), practical block matrix (solid blue, $M(3N+1) \times (3N+1)$ elements), and pruned block matrix (dotted green, capped at 10 configurations, $10(3N+1) \times (3N+1)$ 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

[23] though these are not globally optimal like thos from optuna [147]

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.

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.

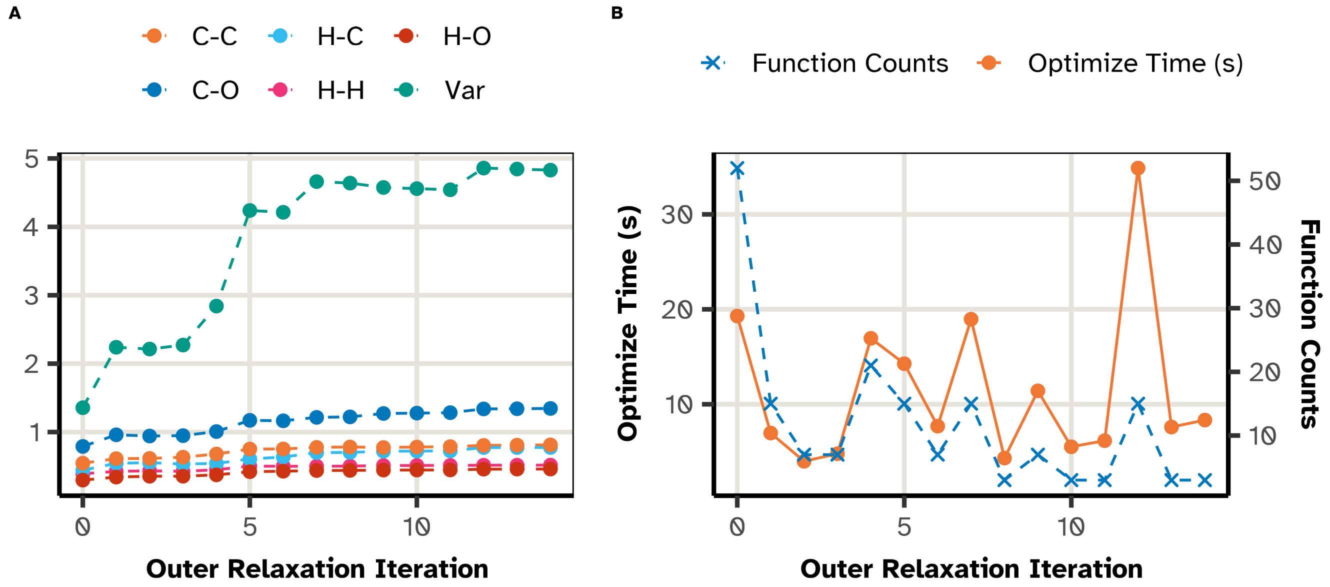


*Figure 7.7. Hyperparameter and computational cost during GPDimer. **(A)** Evolution of kernel hyperparameters for the `S000` show that lengthscales remain stable after an initial adjustment period. The signal variance fluctuates, an artifact of having to fit subsequent points. **(B)** Computational 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:

$$\mathcal{L}_{\text{eff}}(\theta) = \underbrace{\log p(\mathbf{y} \mid \mathcal{S}, \theta)}_{\text{MLL}} - \mu \log(\lambda_{\max} - \log \sigma_f^2) \tag{78}$$

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:

$$\mu(N) = \mu_0 + \alpha N, \qquad \mu(N) \leq \mu_{\max} \tag{79}$$

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 2 (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:

$$O_j(t) = \begin{cases} 1 & \text{if } \operatorname{sgn}[\Delta\theta_j(t-1)] \neq \operatorname{sgn}[\Delta\theta_j(t-2)], \\ 0 & \text{otherwise.} \end{cases} \tag{80}$$

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.* Oct. 2025. DOI: `10.48550/arXiv.2510.06030`. arXiv: `2510.06030 [physics]`

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 [148]. 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):

$$\hat{f} \;=\; \arg\min_{f\in\mathcal{H}_k} \frac{1}{\sigma_n^2}\sum_i (y_i - f(x_i))^2 \;+\; \|f\|^2_{\mathcal{H}_k}, \tag{81}$$

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

$$\theta^* = \arg\max_\theta \log p(\mathbf{y}|\mathcal{S},\theta), \tag{82}$$

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

$$\theta_{\text{ideal}} = \arg\min_\theta \int |f(x;\theta) - V(x)|^2\, dx. \tag{83}$$

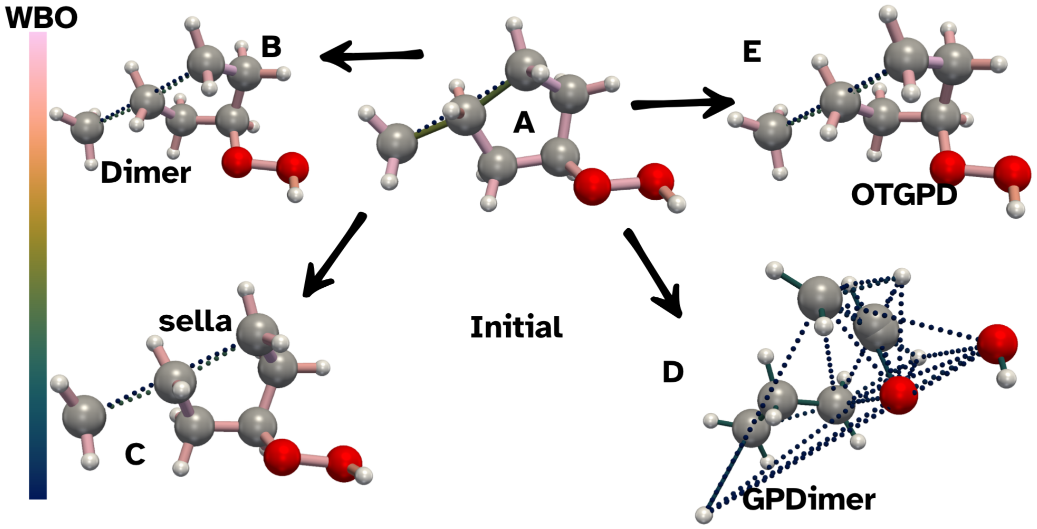


Figure 8.1. A comparison of saddle point search trajectories for a ethoxy radical hydrogen abstraction reaction, `doublet_150` *reaction starting from an initial configuration (A). The standard Dimer method (B), Sella (C), and the OTGPD (E) follow a chemically intuitive path. The previous GPDimer method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

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 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 greately 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]

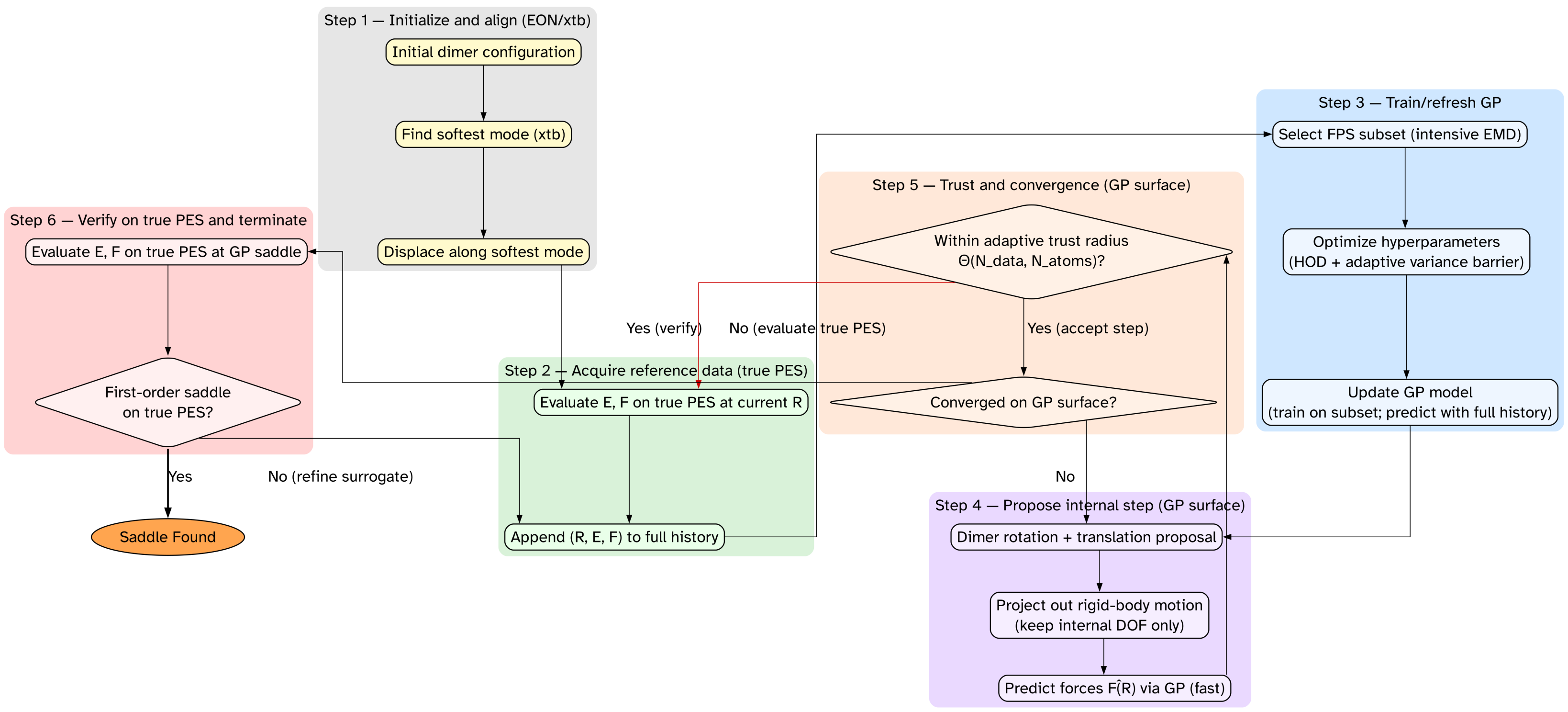

Step 1 — Initialize and align (EON/xtb)
Initial dimer configuration
Find softest mode (xtb)
Displace along softest mode
Step 2 — Acquire reference data (true PES)
Evaluate E, F on true PES at current R
Append (R, E, F) to full history
Step 3 — Train/refresh GP
Select FPS subset (intensive EMD)
Optimize 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 (accept step)
No (evaluate true PES)
Converged on GP surface?
Yes (verify)
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]:

$$\frac{2}{3} r_{ij}(\mathbf{x}_{\text{eval}}) < r_{ij}(\mathbf{x}_{\text{im}}) < \frac{3}{2} r_{ij}(\mathbf{x}_{\text{eval}}) \tag{84}$$

$$\implies \left| \log \frac{r_{ij}(\mathbf{x}_{\text{im}})}{r_{ij}(\mathbf{x}_{\text{eval}})} \right| < \log(1.5) \approx 0.405 \tag{85}$$

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:

$$D_{\text{1Dmaxlog}}(\mathbf{x}_1, \mathbf{x}_2) = \max_{i,j} \left| \log \frac{r_{ij}(\mathbf{x}_2)}{r_{ij}(\mathbf{x}_1)} \right| \tag{86}$$

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.

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 [149], 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:

$$\bar{d}_t = \frac{1}{N_t} \min_{\pi \in \Pi_{N_t}} \sum_{k=1}^{N_t} \| \mathbf{r}_{k,t}^{(1)} - \mathbf{r}_{\pi(k),t}^{(2)} \|. \tag{87}$$

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:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \max_t \bar{d}_t(\mathbf{x}_i, \mathbf{x}_j). \tag{88}$$

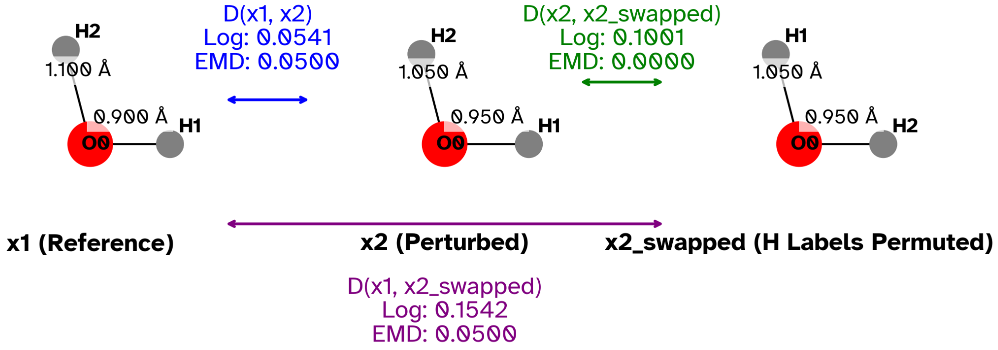


Figure 8.3. Comparison of the 1D max log distance and the Earth Mover's Distance (EMD) for an asymmetric stretch of a water molecule. While configuration x_2 and $x_{2,s}$ are physically identical (differing only by the permutation of hydrogen atom labels), the 1D max log metric incorrectly assigns a 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.

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:

$$d_{\text{EMD}}(\mathbf{x}_{\text{cand}}, \mathbf{x}_{\text{nn}}) \leq \Theta(N_{\text{data}}, N_{\text{atoms}}) \tag{89}$$

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:

$$\Theta_{\text{earned}}(N_{\text{data}}) = T_{\min} + \Delta T_{\text{explore}} \cdot \left(1 - e^{-k\,N_{\text{data}}}\right) \tag{90}$$

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:

$$\Theta_{\text{phys}}(N_{\text{atoms}}) = \max\left(a_{\text{floor}},\ \frac{a_A}{\sqrt{N_{\text{atoms}}}}\right) \tag{91}$$

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.

$$\Theta(N_{\text{data}}, N_{\text{atoms}}) = \min(\Theta_{\text{earned}}(N_{\text{data}}),\ \Theta_{\text{phys}}(N_{\text{atoms}})) \tag{92}$$

If a proposed step violates the trust radius in Eq. 19, 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 [150]. 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:

$$\lambda \in \bigcup_{i=1}^{n} \left[\, K_{ii} - R_i,\ K_{ii} + R_i \,\right], \quad R_i := \sum_{j \neq i} |K_{ij}|. \tag{93}$$

This yields a lower bound on the smallest eigenvalue:

$$\lambda_{\min}(\mathbf{K}) \geq \min_i \left(K_{ii} - R_i\right). \tag{94}$$

For block-structured energy–force kernels, a tighter bound groups terms by configuration [151]:

$$\lambda_{\min}(\mathbf{K}) \geq \min_i \left\{ \lambda_{\min}(\mathbf{K}_{ii}) - \sum_{j \neq i} \|\mathbf{K}_{ij}\|_2 \right\}, \tag{95}$$

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

$$K_{EE}(\mathbf{r}, \mathbf{r}) = \sigma_c^2 + \sigma_f^2 + \sigma_{n,E}^2, \tag{96}$$

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

$$(\mathbf{K}_{FF}(\mathbf{r}, \mathbf{r}))_{(i,d),(i,d)} = -\frac{\sigma_f^2}{2} \frac{\partial^2 \mathcal{D}^2}{\partial x_{i,d} \, \partial x'_{i,d}} \bigg|_{\mathbf{x}'=\mathbf{x}=\mathbf{r}} + \sigma_{n,F}^2. \tag{97}$$

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 [152].

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 mitigiating 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.

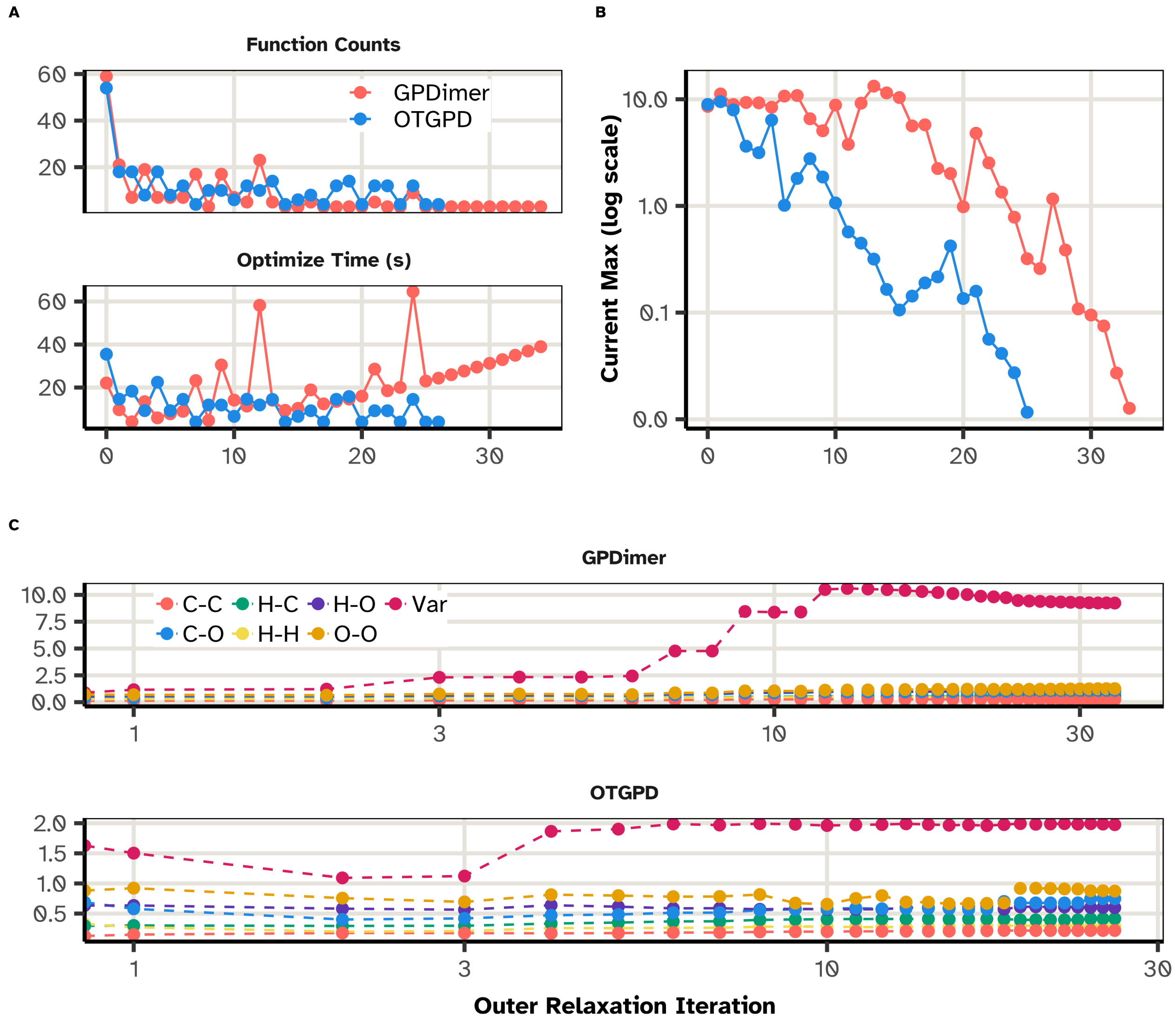


Figure 8.4. *Performance trace for the* `singlet_016` *system (Figure fig:equiv:optgd), illustrating the comparative behavior of GPDimer and OTGPD during saddle search optimization. (A) The per-iteration electronic structure function counts and wall time show that OTGPD (skyblue) consistently achieves lower and more stable computational cost per iteration compacoral 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.*

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 pathlogoically 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 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 1. 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	$\mathrm{num}_{\mathrm{fragments}}$	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.

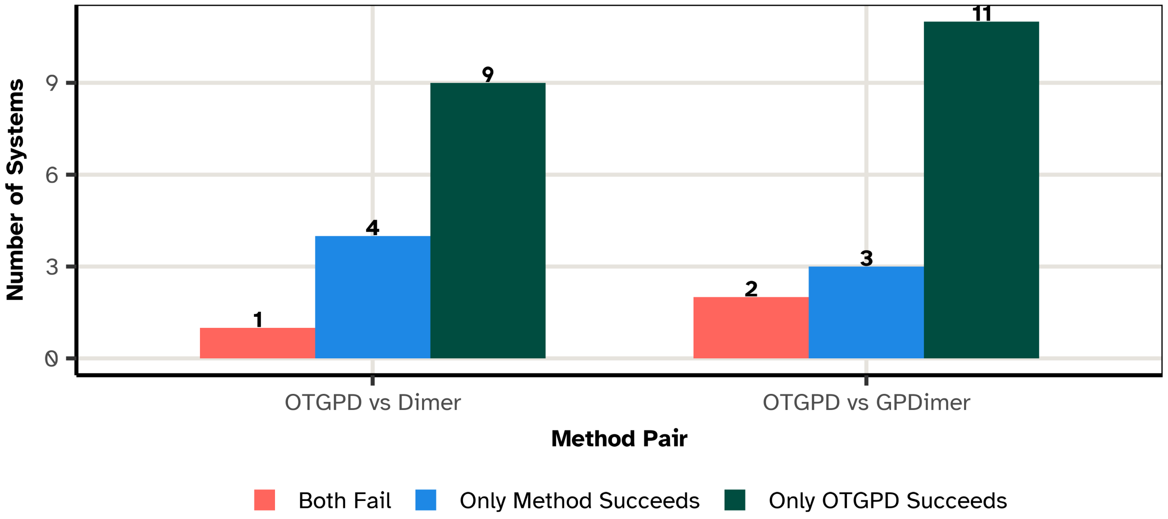


Figure 8.5. *Reliability comparison of OTGPD against GPDimer and standard Dimer methods across 238 molecular systems. A calculation exceeding 240 minutes or raising an error in the electronic structure calculation counts as a failure. The bar chart shows the distribution of outcomes for each pairwise 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.*

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 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.

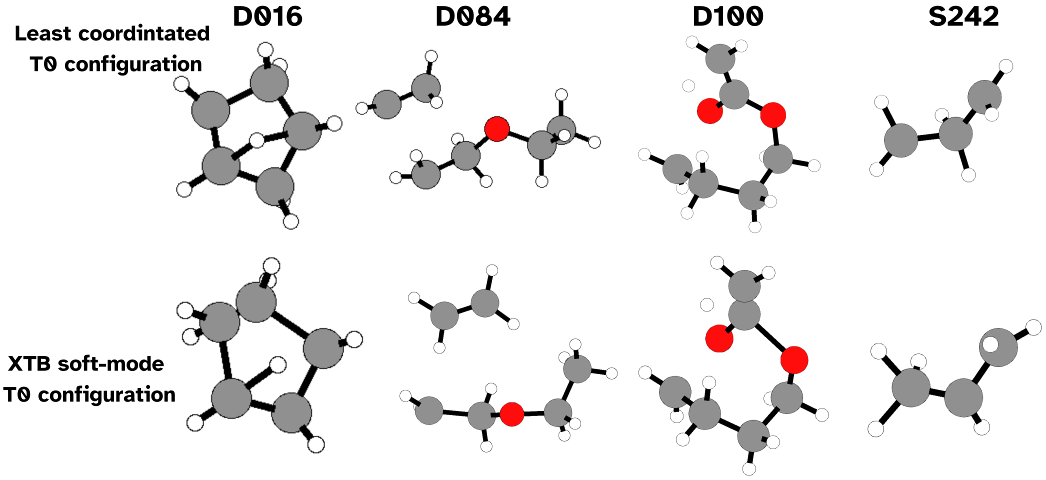


Figure 8.6. Initializations for the GPDimer (top) and OTGPD (bottom). The xTB initialization procedure in these four cases results in unphysical initial geometries, leading to failures in the optimization routine. Specifically, D016 exhibits unphysically close carbon atoms, D084 possesses a shortened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

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.

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.

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

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

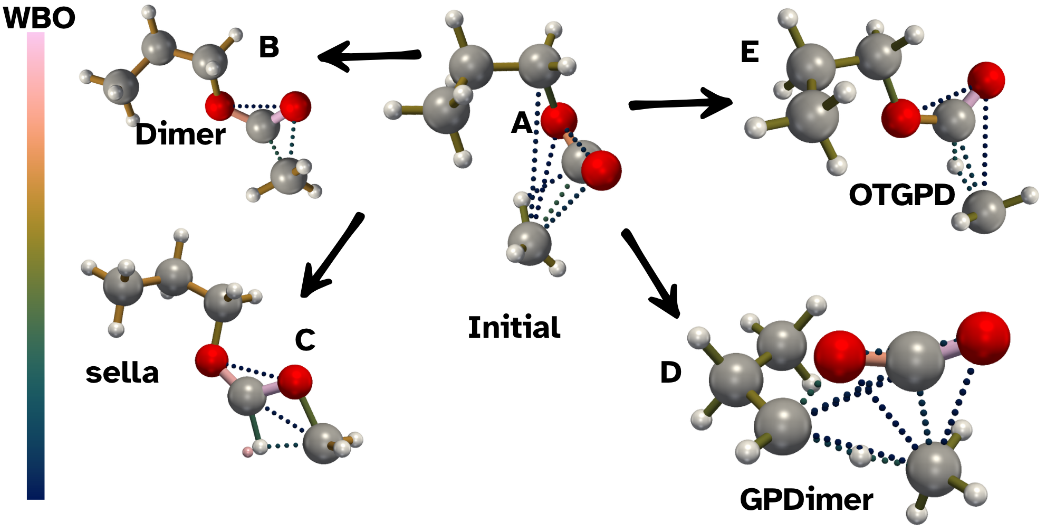


Figure 8.7. *Endpoints for saddle point search trajectories of the* `singlet_016` *system starting from an initial configuration (A). The standard Dimer method (B) and the proposed OTGPD method (E) identifies the nearest transition state structure. The previous GPDimer method (D) and Sella (C) is guided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

mode corresponds to a low-energy torsional motion of the hydrogens on the terminal methyl group (C5).

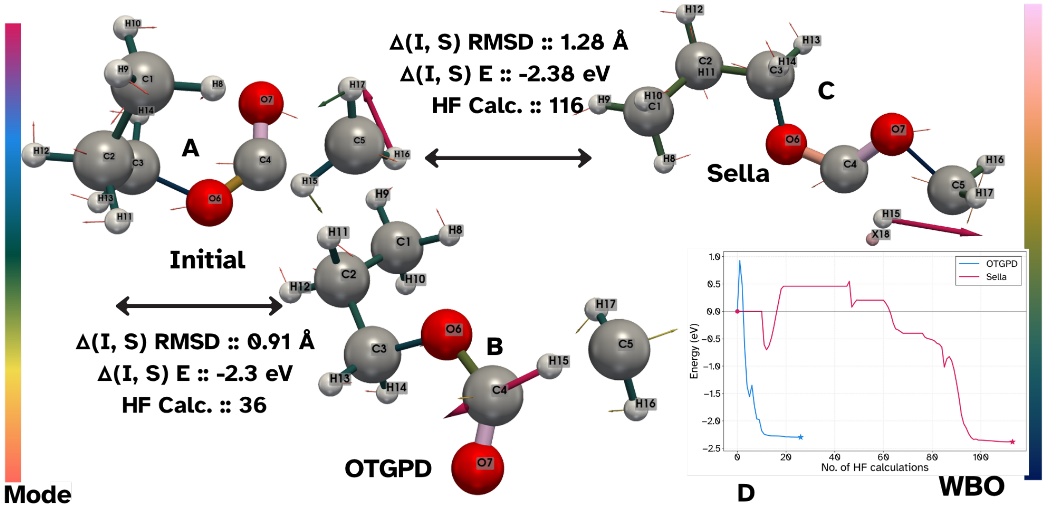


Figure 8.8. Comparison of the OTGPD and Sella algorithms for a saddle point search on n-propyl acetate (`singlet_016`*), starting from the initial, non-equilibrium geometry. The OTGPD method efficiently locates the geometrically proximal saddle point corresponding to C–O bond cleavage in 36 steps. The 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.*

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.

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

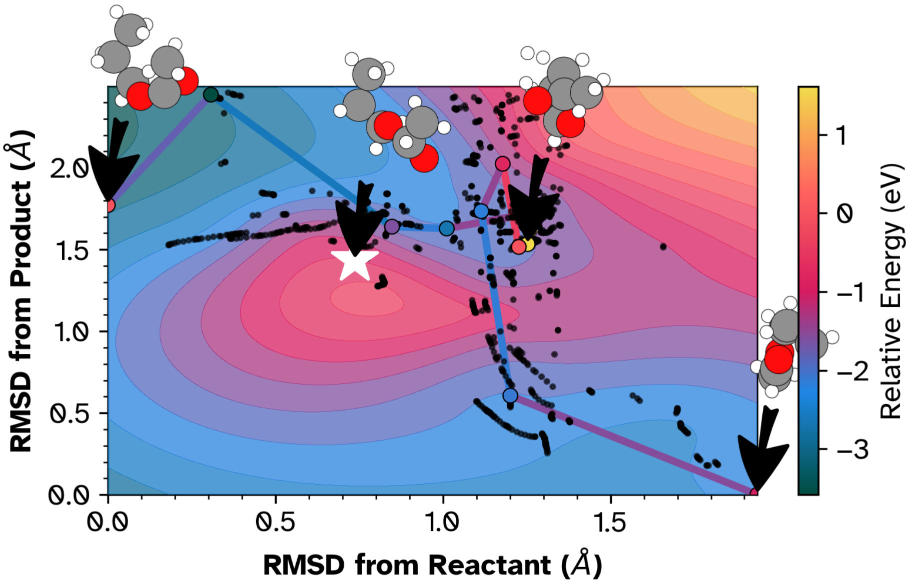


Figure 8.9. *A 2D landscape projection visualizing the potential energy surface of the n-propyl acetate system. This surface, described in Sec. 4.2, depicts the energy landscape as a function of observed paths during the optimization. The landscape clearly reveals several states. The proximal transition 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].*

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 4 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 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 4 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.

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 4 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:

$$F(x_A) \prec F(x_B) \iff \forall i, f_i(x_A) \leq f_i(x_B) \land \exists j, f_j(x_A) < f_j(x_B) \tag{98}$$

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.

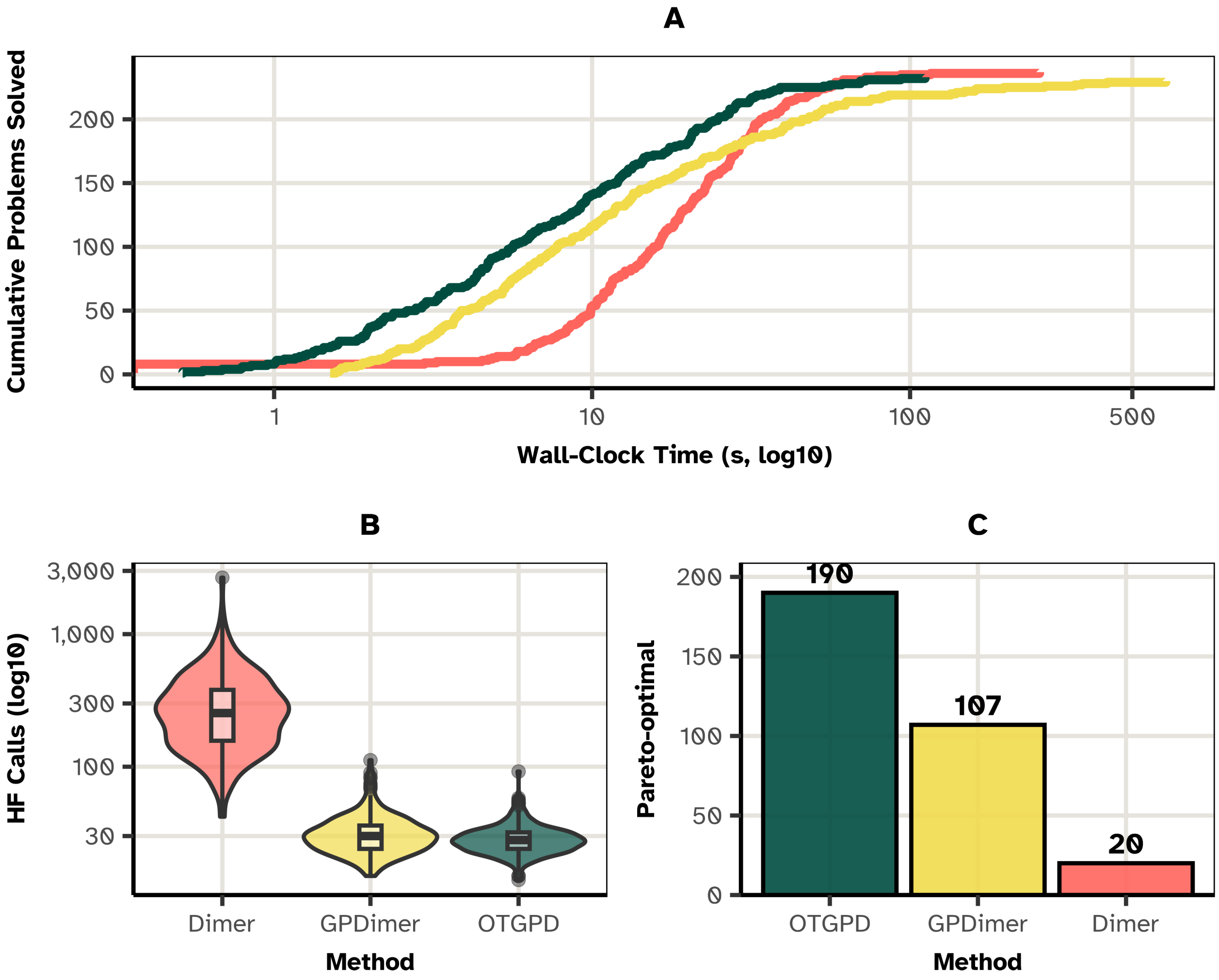


Figure 8.10. Comparison of computational efficiency for the OTGPD, GPDimer, and standard Dimer methods. ***(A)*** *A cactus plot shows the cumulative number of problems solved versus wall-clock time, demonstrating OTGPD's superior raw speed.* ***(B)*** *Violin plots of the number of Hartree-Fock (HF) 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.*

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 [153]:

$$V(x) = \sum_{i} V_1(i) + \sum_{i<j} V_2(i,j) + \sum_{i<j<k} V_3(i,j,k) + \cdots + V_N(1,\ldots,N) \tag{99}$$

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 α–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] 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.

[131] 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.

[132] 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.

[133] 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.

[134] 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.

[135] 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.

[136] 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.

[137] Andrew Gelman. *Bayesian Data Analysis*. Third edition. Chapman & Hall/CRC Texts in Statistical Science. Boca Raton: CRC Press, 2014.

[138] 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].

[139] 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].

[140] 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.

[141] 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.

[142] 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.

[143] James E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics*. Springer Texts in Statistics. New York, N.Y. ; [London]: Springer, 2007.

[144] 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].

[145] 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.

[146] 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.

[147] 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.

[148] 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.

[149] Matthew Thorpe. "Introduction to Optimal Transport." In: ().

[150] 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.

[151] 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.

[152] 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.

[153] A. J. Stone. *The Theory of Intermolecular Forces*. Second edition. Oxford: Oxford University Press, 2013.

Appendix A: Publications

Paper I

High-Order Finite Element Method for Atomic Structure Calculations

Čertík, Ondřej, Pask, John E., Fernando, Isuru, **Rohit Goswami**, Sukumar, N., Collins, Lee. A., Manzini, Gianmarco, and Vackář Jiří

Computer Physics Communications, Vol. 315, pp. 109051

Rohit finalized the code, reproducibly generated figures and revised the article.

Abstract

We introduce `featom`, an open source code that implements a high-order finite element solver for the radial Schrödinger, Dirac, and Kohn-Sham equations. The formulation accommodates various mesh types, such as uniform or exponential, and the convergence can be systematically controlled by increasing the number and/or polynomial order of the finite element basis functions. The Dirac equation is solved using a squared Hamiltonian approach to eliminate spurious states. To address the slow convergence of the $\kappa = \pm 1$ states due to divergent derivatives at the origin, we incorporate known asymptotic forms into the solutions. We achieve a high level of accuracy (10^{-8} Hartree) for total energies and eigenvalues of heavy atoms such as uranium in both Schrödinger and Dirac Kohn-Sham solutions. We provide detailed convergence studies and computational parameters required to attain commonly required accuracies. Finally, we compare our results with known analytic results as well as the results of other methods. In particular, we calculate benchmark results for atomic numbers (Z) from 1 to 92, verifying current benchmarks. We demonstrate significant speedup compared to the state-of-the-art shooting solver `dftatom`. An efficient, modular Fortran 2008 implementation, is provided under an open source, permissive license, including examples and tests, wherein particular emphasis is placed on the independence (no global variables), reusability, and generality of the individual routines.

Keywords atomic structure, electronic structure, Schrödinger equation, Dirac equation, Kohn-Sham equations, density functional theory, finite element method, Fortran 2008

1 Introduction

Over the past three decades, Density Functional Theory (DFT) [1] has established itself as a cornerstone of modern materials research, enabling the understanding, prediction, and control of a wide variety of materials properties from the first principles of quantum mechanics, with no empirical parameters. However, the solution of the required Kohn-Sham equations [2] is a formidable task, which has given rise to a number of different solution methods [3]. At the heart of the majority of methods in use today, whether for isolated systems such as molecules or extended systems such as solids and liquids, lies the solution of the Schrödinger and/or Dirac equations for the isolated atoms composing the larger molecular or condensed matter systems of interest. Particular challenges arise in the context of relativistic calculations, which require solving the Dirac equation, since spurious states can arise due to the unbounded nature of the Dirac Hamiltonian operator and inconsistencies in discretizations derived therefrom [4].

A number of approaches have been developed to avoid spurious states in the solution of the Dirac equation. Shooting methods, e.g., [5] and references therein, avoid such states by leveraging known asymptotic forms to target desired eigenfunctions based on selected energies and numbers of nodes. However, due to the need for many trial solutions to find each eigenfunction, efficient implementation while maintaining robustness is nontrivial. In addition, convergence parameters such as distance of grid points from the origin must be carefully tuned. Basis set methods, e.g., [6, 7, 8, 9] and references therein, offer an elegant alternative to shooting methods, solving for all states at once by diagonalization of the Schrödinger or Dirac Hamiltonian in the chosen basis. However, due to the unbounded spectrum of the Dirac Hamiltonian, spurious states have been a longstanding issue [4, 10]. Many approaches have been developed to avoid spurious states over the past few decades, with varying degrees of success. These include using different bases for the large and small components of the Dirac wavefunction [6, 11, 12, 7, 13, 14], modifying the Hamiltonian [15, 8, 9, 16, 17], and imposing various boundary constraints [18, 19, 12, 8]. In the finite-difference context, defining large and small components of the Dirac wavefunction on alternate grid points [20], replacing conventional central differences with asymmetric differences [20, 21], and adding a Wilson term to the Hamiltonian [17] have proven effective in eliminating spurious states. While in the finite element (FE) context, the use of different trial and test spaces in a stabilized Petrov-Galerkin formulation has proven effective in mitigating the large off-diagonal convection (first derivative) terms and absence of diffusion (second derivative) terms causing the instability [9, 16].

In the this work, we present the open-source code,`featom`[25], for the solution of the Schrödinger, Dirac, and Kohn-Sham equations in a high-order finite element basis. The FE basis enables exponential convergence with respect to polynomial order while allowing full flexibility as to choice of radial mesh. To eliminate spurious states in the solution of the Dirac equation, we square the Dirac Hamiltonian operator [22, 15]. Since the square of the operator has the same eigenfunctions as the operator itself, and the square of its eigenvalues, determination of the desired eigenfunctions and eigenvalues is immediate. Most importantly, since the square of the operator is bounded from below, unlike the operator itself, it is amenable to direct solution by standard variational methods, such as FE, without modification. This affords simplicity, robustness, and well understood convergence. Moreover, squaring the operator rather than modifying it and/or boundary conditions upon it, ensures key properties are preserved exactly, such as convergence to the correct non-relativistic (Schrödinger) limit with increasing speed of light [15]. Squaring the operator also stabilizes the numerics naturally, without approximation, by creating second-derivative terms. To accelerate convergence with respect to polynomial order, we incorporate known asymptotic forms as $r \to 0$ into the solutions: rather than solving for large and small Dirac wavefunction components $P(r)$ and $Q(r)$, we solve for $\frac{P(r)}{r^\alpha}$ and $\frac{Q(r)}{r^\alpha}$, with α based on the known asymptotic forms for P and Q as $r \to 0$. This eliminates derivative divergences and non-polynomial behavior in the vicinity of the origin and so enables rapidly convergent solutions in a polynomial basis for all quantum numbers κ, including $\kappa = \pm 1$. By combining the above ideas, `featom` is able to provide robust, efficient, and accurate solutions for both Schrödinger and Dirac equations.

The package is MIT licensed and is written in Fortran, leveraging language features from the 2008 standard, with an emphasis on facilitating user extensions. Additionally, it is designed to work within the modern Fortran ecosystem and leverages the `fpm` build system. There are several benchmark calculations that serve as tests. The package supports different mesh-generating methods, including support for a uniform mesh, an exponential mesh, and other meshes defined by nodal distributions and derivatives. Multiple quadrature methods have been implemented, and their usage in the code is physically motivated. Gauss–Jacobi quadrature is used to accurately integrate problem-

[25] Source also publicly on Github: `https://github.com/atomic-solvers/featom`

atic integrals for the Dirac equation and the Poisson equation as well as for the total energy in the Dirac-Kohn-Sham solution, whereas Gauss–Legendre quadrature is used for the Schrödinger equation. To ensure numerical accuracy, we employ several techniques, using Gauss–Lobatto quadrature for the overlap matrix to recover a standard eigenproblem, precalculation of most quantities, and parsimonious assembly of a lower-triangular matrix with a symmetric eigensolver. With these considerations we show that the resulting code outperforms the state-of-the-art code `dftatom`, using fewer parameters for convergence while retaining high accuracy of 10^{-8} Hartree in total energy and eigenvalues for uranium (both Dirac and Schrödinger) and all lighter atoms.

The remainder of the paper is organized as follows. Section 2 describes the electronic structure equations solved. This is followed by Sections 3.1 and 3.2, which detail the unified finite element solution for the radial Schrödinger and Dirac equations. Section 4.4 details the numerical techniques employed to efficiently construct and solve the resulting matrix eigenvalue problem, including mesh and quadrature methods. In Section 4, we present results from analytic tests and benchmark comparisons against the shooting solver `dftatom`, followed by a brief discussion of findings. Finally, in Section 5, we summarize our main conclusions.

2 Electronic structure equations

Under the assumption of a central potential, we establish the conventions used for the electronic structure problems under the purview of `featom` in this section. Starting from the nonrelativistic radial Schrödinger equation and its relativistic counterpart, the Dirac equation, we couple these to the Kohn–Sham equations with a Poisson equation for the Hartree potential. The interested reader may find more details of this formulation in [5]. By convention, we use Hartree atomic units throughout the manuscript.

2.1 Radial Schrödinger equation

Recall that the 3D one-electron Schrödinger equation is given by

$$\left(-\tfrac{1}{2}\nabla^2 + V(\mathbf{x})\right)\psi(\mathbf{x}) = E\psi(\mathbf{x}). \tag{1}$$

When the potential considered is spherically symmetric, i.e.,

$$V(\mathbf{x}) := V(r), \tag{2}$$

the eigenstates of energy and angular momentum can be written in the form

$$\psi_{nlm}(\mathbf{x}) = R_{nl}(r)\, Y_{lm}\left(\frac{\mathbf{x}}{r}\right), \tag{3}$$

where n is the principal quantum number, l is the orbital angular momentum quantum number, and m is the magnetic quantum number. It follows that $R_{nl}(r)$ satisfies the radial Schrödinger equation

$$-\tfrac{1}{2}\left(r^2 R'_{nl}(r)\right)' + \left(r^2 V + \tfrac{1}{2}l(l+1)\right) R_{nl}(r) = E r^2 R_{nl}(r). \tag{4}$$

The functions $\psi_{nlm}(\mathbf{x})$ and $R_{nl}(r)$ are normalized as

$$\int |\psi_{nlm}(\mathbf{x})|^2 \, \mathrm{d}^3x = 1,$$
$$\int_0^\infty R_{nl}^2(r) r^2 \mathrm{d}r = 1. \tag{5}$$

2.2 Radial Dirac equation

The one-electron radial Dirac equation can be written as

$$P'_{n\kappa}(r) = -\frac{\kappa}{r} P_{n\kappa}(r) + \left(\frac{E - V(r)}{c} + 2c \right) Q_{n\kappa}(r), \tag{6a}$$
$$Q'_{n\kappa}(r) = -\left(E - \frac{V(r)}{c} \right) P_{n\kappa}(r) + \frac{\kappa}{r} Q_{n\kappa}(r), \tag{6b}$$

where $P_{n\kappa}(r)$ and $Q_{n\kappa}(r)$ are related to the usual large $g_{n\kappa}(r)$ and small $f_{n\kappa}(r)$ components of the Dirac equation by

$$P_{n\kappa}(r) = r g_{n\kappa}(r), \tag{7a}$$
$$Q_{n\kappa}(r) = r f_{n\kappa}(r). \tag{7b}$$

A pedagogical derivation of these results can be found in the literature, for example in [5, 23, 24]. We follow the solution labeling in [5], in which the relativistic quantum number κ is determined by the orbital angular momentum quantum number l and spin quantum number $s = \pm 1$ on the basis of the total angular momentum quantum number $j = l \pm \frac{1}{2}$ using

$$\kappa = \begin{cases} -l - 1 & \text{for } j = l + \frac{1}{2}, \text{ i.e. } s = +1, \\ l & \text{for } j = l - \frac{1}{2}, \text{ i.e. } s = -1. \end{cases} \tag{8}$$

By not including the rest mass energy of an electron, the energies obtained from the radial Dirac equation can be compared to the non-relativistic energies obtained from the Schrödinger equation.

The normalization of $P_{n\kappa}(r)$ and $Q_{n\kappa}(r)$ is

$$\int_0^\infty (P_{n\kappa}^2(r) + Q_{n\kappa}^2(r)) \mathrm{d}r = 1. \tag{9}$$

We note that both $P_{n\kappa}$ and $Q_{n\kappa}$ are solutions of homogeneous equations and are thus only unique up to an arbitrary multiplicative constant.

2.3 Poisson equation

The 3D Poisson equation for the Hartree potential V_H due to electronic density n is given by

$$\nabla^2 V_H(\mathbf{x}) = -4\pi n(\mathbf{x}). \tag{10}$$

For a spherical density $n(\mathbf{x}) = n(r)$, this becomes

$$\frac{1}{r^2}(r^2 V_H')' = V_H''(r) + \frac{2}{r}V_H'(r) = -4\pi n(r), \tag{11}$$

where $n(r)$ is the radial particle (number) density, normalized such that

$$N = \int n(\mathbf{x})\, \mathrm{d}^3x = \int_0^\infty 4\pi n(r) r^2 \mathrm{d}r, \tag{12}$$

where N is the number of electrons.

2.3.1 Initial conditions

Substituting (11) into (12) and integrating, we obtain

$$\lim_{r\to\infty} r^2 V_H'(r) = -N, \tag{13}$$

from which it follows that the asymptotic behavior of $V_H'(r)$ is

$$V_H'(r) \sim -\frac{N}{r^2}, \qquad r \to \infty. \tag{14}$$

Integrating (14) and requiring $V_H \to 0$ as $r \to \infty$ then gives the corresponding asymptotic behavior for $V_H(r)$,

$$V_H(r) \sim \frac{N}{r}, \qquad r \to \infty. \tag{15}$$

For small r, the asymptotic behavior can be obtained by expanding $n(r)$ about $r = 0$: $n(r) = \sum_{j=0}^\infty c_j r^j$. Substituting into Poisson equation (11) gives

$$(r^2 V_H')' = -4\pi \sum_{j=0}^\infty c_j r^{j+2}. \tag{16}$$

Integrating and requiring $V_H(0)$ to be finite then gives

$$V_H'(r) = -4\pi \sum_{j=0}^\infty c_j \frac{r^{j+1}}{j+3}, \tag{17}$$

with linear leading term, so that we have

$$V_H'(r) \sim r, \qquad r \to 0. \tag{18}$$

Integrating (17) then gives

$$V_H(r) = -4\pi \sum_{j=0}^\infty c_j \frac{r^{j+2}}{(j+2)(j+3)} + C, \tag{19}$$

with leading constant term $C = V_H(0)$ determined by Coulomb's law:

$$V_H(0) = 4\pi \int_0^\infty r n(r) \mathrm{d}r. \tag{20}$$

Finally, from (18) we have that

$$V_H'(0) = 0. \tag{21}$$

So $V_H \to 0$ as $r \to \infty$ and $V_H'(0) = 0$. This asymptotic behavior provides the initial values and derivatives for numerical integration in both inward and outward directions.

2.4 Kohn-Sham equations

The Kohn–Sham equations consist of the radial Schrödinger or Dirac equations with an effective potential $V(r) = V_{\text{in}}(r)$ given by (see, e.g., [3])

$$V_{\text{in}} = V_H + V_{xc} + v, \tag{22}$$

where V_H is the Hartree potential given by the solution of the radial Poisson equation (11), V_{xc} is the exchange-correlation potential, and $v = -\frac{Z}{r}$ is the nuclear potential.

The total energy is given by

$$E[n] = T_s[n] + E_H[n] + E_{xc}[n] + V[n], \tag{23}$$

the sum of kinetic energy

$$T_s[n] = \sum_{nl} f_{nl}\varepsilon_{nl} - 4\pi \int V_{\text{in}}(r)n(r)r^2 \mathrm{d}r, \tag{24}$$

where ε_{nl} are the Kohn-Sham eigenvalues, Hartree energy

$$E_H[n] = 2\pi \int V_H(r)n(r)r^2 \mathrm{d}r, \tag{25}$$

exchange-correlation energy

$$E_{xc}[n] = 4\pi \int \varepsilon_{xc}(r;n)n(r)r^2 \mathrm{d}r, \tag{26}$$

where $\varepsilon_{xc}(r;n)$ is the exchange and correlation energy density, and Coulomb energy

$$V[n] = 4\pi \int v(r)n(r)r^2 \mathrm{d}r = -4\pi Z \int n(r)r \mathrm{d}r \tag{27}$$

with electronic density in the nonrelativistic case given by

$$n(r) = \frac{1}{4\pi} \sum_{nl} f_{nl} \frac{P_{nl}^2(r)}{r^2}, \tag{28}$$

where P_{nl} is the radial wavefunction in (40) and f_{nl} the associated electronic occupation. In the relativistic case, the electronic density is given by

$$n(r) = \frac{1}{4\pi} \sum_{nls} f_{nls} \frac{P_{nls}^2(r) + Q_{nls}^2(r)}{r^2}, \tag{29}$$

where P_{nls} and Q_{nls} are the two components of the Dirac solution ((6a), (6b)) and f_{nls} is the occupation. In both the above cases, $n(r)$ is the electronic particle density [electrons/volume], everywhere positive, as distinct from the electronic charge density $\rho(r)$ [charge/volume]: $\rho(r) = -n(r)$ in atomic units.

We adopt a self-consistent approach [3] to solve for the electronic structure. Starting from an initial density n_{in} and corresponding potential V_{in}, we solve the Schrödinger or Dirac equation to determine the wavefunctions R_{nl} or spinor components P and

Q, respectively. From these, we construct a new density n_{out} and potential V_{out}. Subsequently, we update the input density and potential, using for example a weighting parameter $\alpha \in [0, 1]$:

$$n_{\text{in}} \to \alpha n_{\text{in}} + (1-\alpha) n_{\text{out}}, \tag{30}$$

$$V_{\text{in}} \to \alpha V_{\text{in}} + (1-\alpha) V_{\text{out}}. \tag{31}$$

This process is repeated until the difference of V_{in} and V_{out} and/or n_{in} and n_{out} is within a specified tolerance, at which point *self-consistency* is achieved. This fixed-point iteration is known as the *self-consistent field* (SCF) iteration. We employ an adaptive linear mixing scheme, with optimized weights for each component of the potential to construct new input potentials for successive SCF iterations. To accelerate the convergence of the SCF iterations, under relaxed fixed point iteration methods are used. In particular, the code supports both `linear` [25] and (default) Periodic `Pulay` mixing schemes [26], which suitably accelerate convergence. In order to reduce the number of SCF iterations, we use a Thomas–Fermi (TF) approximation [27] for the initial density and potential:

$$V(r) = -\frac{Z_{\text{eff}}(r)}{r}, \tag{32a}$$

$$Z_{\text{eff}}(r) = Z \left(1 + \alpha\sqrt{x} + \beta x e^{-\gamma\sqrt{x}}\right)^2 e^{-2\alpha\sqrt{x}}, \tag{32b}$$

$$x = r\left(\frac{128 Z}{9\pi^2}\right)^{1/3}, \tag{32c}$$

$$\alpha = 0.7280642371, \quad \beta = -0.5430794693, \quad \gamma = 0.3612163121. \tag{32d}$$

The corresponding charge density is then

$$\rho(r) = -\frac{1}{3\pi^2}\left(-2V(r)\right)^{\frac{3}{2}}. \tag{33}$$

We demonstrate the methodology with the local density approximation (LDA) and relativistic local-density approximation (RLDA) exchange and correlation functionals. We note that the modular nature of the code and interface mechanism make it straightforward to incorporate functionals from other packages such as the library of exchange correlation functions, `libxc` [28, 29]. The parameters used in `featom` are taken from the same NIST benchmark data [30] as the current state-of-the-art `dftatom` program for an accurate comparison. For the local density approximation,

$$V_{xc}(r; n) = \frac{\mathrm{d}}{\mathrm{d}n}\left(n\varepsilon_{xc}^{LD}(n)\right), \tag{34}$$

where the exchange and correlation energy density ε_{xc}^{LD} can be written as [3]

$$\varepsilon_{xc}^{LD}(n) = \varepsilon_{x}^{LD}(n) + \varepsilon_{c}^{LD}(n), \tag{35}$$

with electron gas exchange term [3]

$$\varepsilon_{x}^{LD}(n) = -\frac{3}{4\pi}(3\pi^2 n)^{\frac{1}{3}} \tag{36}$$

and Vosko-Wilk-Nusair (VWN) [31] correlation term

$$\varepsilon_c^{LD}(n) \sim \frac{A}{2}\left\{\log\left(\frac{y^2}{Y(y)}\right) + \frac{2b}{Q}\arctan\left(\frac{Q}{2y+b}\right) \right. \\ \left. -\frac{by_0}{Y(y_0)}\left[\log\left(\frac{(y-y_0)^2}{Y(y)}\right) + \frac{2(b+2y_0)}{Q}\arctan\left(\frac{Q}{2y+b}\right)\right]\right\}, \quad (37)$$

in which $y = \sqrt{r_s}$, $Y(y) = y^2 + by + c$, $Q = \sqrt{4c - b^2}$, $y_0 = -0.10498$, $b = 3.72744$, $c = 12.9352$, $A = 0.0621814$, and

$$r_s = \left(\frac{3}{4\pi n}\right)^{\frac{1}{3}} \quad (38)$$

is the Wigner-Seitz radius, which gives the mean distance between electrons. In the relativistic (RLDA) case, a correction to the LDA exchange energy density and potential is given by MacDonald and Vosko [32]:

$$\varepsilon_x^{RLD}(n) = \varepsilon_x^{LD}(n)R, \quad (39a)$$

$$R = 1 - \frac{3}{2}\left(\frac{\beta\mu - \log(\beta+\mu)}{\beta^2}\right)^2,$$

$$V_x^{RLD}(n) = V_x^{LD}(n)S, \quad (39b)$$

$$S = \frac{3\log(\beta+\mu)}{2\beta\mu} - \tfrac{1}{2},$$

where $\mu = \sqrt{1+\beta^2}$ and $\beta = \frac{(3\pi^2 n)^{\frac{1}{3}}}{c} = -\frac{4\pi\varepsilon_x^{LD}(n)}{3c}$.

3 Solution methodology

Having described the Schrödinger, Dirac, and Kohn-Sham electronic structure equations to be solved, and key solution properties, we now detail our approach to solutions in a high-order finite-element basis.

3.1 Radial Schrödinger equation

To recast (4) in a manner that will facilitate our finite element solution methodology, we make the substitution $P_{nl}(r) = rR_{nl}(r)$ to obtain the canonical radial Schrödinger equation in terms of $P_{nl}(r)$ is

$$-\frac{1}{2}P_{nl}''(r) + \left(V(r) + \frac{l(l+1)}{2r^2}\right)P_{nl}(r) = EP_{nl}(r)\,. \quad (40)$$

The corresponding normalization of $P(r)$ is

$$\int_0^\infty P_{nl}^2(r)\mathrm{d}r = 1. \quad (41)$$

3.1.1 Asymptotics

The known asymptotic behavior of P_{nl} as $r \to 0$ is [33]

$$P_{nl}(r) \sim r^{l+1}, \tag{42}$$

where P_{nl}, being a solution of a homogeneous system of equations, is only unique up to an arbitrary multiplicative constant. We use (40) as our starting point but from now on drop the nl index from P_{nl} for simplicity:

$$-\frac{1}{2}P''(r) + \left(V(r) + \frac{l(l+1)}{2r^2}\right) P(r) = EP(r). \tag{43}$$

In order to facilitate rapid convergence and application of desired boundary conditions in a finite element basis, we generalize (40) to solve for $\tilde{P} = \frac{P(r)}{r^\alpha}$ for any chosen real exponent $\alpha \geq 0$ by substituting $P = r^\alpha \tilde{P}$ to obtain

$$-\frac{1}{2}\frac{1}{r^{2\alpha}}\left(r^{2\alpha}\tilde{P}'(r)\right)' + \left(V(r) + \frac{l(l+1) - \alpha(\alpha-1)}{2r^2}\right) \tilde{P}(r) = E\tilde{P}(r). \tag{44}$$

We note that:

- For $\alpha = 0$ we get $\tilde{P}(r) = \frac{P(r)}{r^0} = P(r)$ and (44) reduces to (40).
- For $\alpha = 1$ we get $\tilde{P}(r) = \frac{P(r)}{r^1} = R(r)$ and (44) reduces to (4).
- Finally, for $\alpha = l + 1$, (which corresponds to the known asymptotic (42)) we obtain $\tilde{P}(r) = \frac{P(r)}{r^{l+1}}$, which tends to a nonzero value at the origin for all l and (44) becomes

$$-\frac{1}{2}\frac{1}{r^{2(l+1)}}\left(r^{2(l+1)}\tilde{P}'(r)\right)' + V(r)\tilde{P}(r) = E\tilde{P}(r). \tag{45}$$

3.1.2 Weak form

To obtain the weak form, we multiply both sides of (44) by a test function $v(r)$ and integrate from 0 to ∞. In addition, to facilitate the construction of a symmetric bilinear form, we multipy by a factor $r^{2\alpha}$ to get

$$\int_0^\infty \left[-\frac{1}{2}\left(r^{2\alpha}\tilde{P}'(r)\right)' v(r) + \left(V(r) + \frac{l(l+1) - \alpha(\alpha-1)}{2r^2}\right) \tilde{P}(r)v(r)r^{2\alpha}\right] \mathrm{d}r \tag{46}$$
$$= E\int_0^\infty \tilde{P}(r)v(r)r^{2\alpha}\mathrm{d}r.$$

We can now integrate by parts to obtain

$$\int_0^\infty \left[\frac{1}{2}r^{2\alpha}\tilde{P}'(r)v'(r) + \left(V(r) + \frac{l(l+1)-\alpha(\alpha-1)}{2r^2}\right)\tilde{P}(r)v(r)r^{2\alpha}\right]\mathrm{d}r \qquad (47)$$

$$-\tfrac{1}{2}\left[r^{2\alpha}\tilde{P}'(r)v(r)\right]_0^\infty = E\int_0^\infty \tilde{P}(r)v(r)r^{2\alpha}\mathrm{d}r.$$

Setting the boundary term to zero,

$$\left[r^{2\alpha}\tilde{P}'(r)v(r)\right]_0^\infty = 0, \qquad (48)$$

we then obtain the desired symmetric weak formulation

$$\int_0^\infty \left[\tfrac{1}{2}\tilde{P}'(r)v'(r) + \left(V(r) + \frac{l(l+1)-\alpha(\alpha-1)}{2r^2}\right)\tilde{P}(r)v(r)\right]r^{2\alpha}\mathrm{d}r \qquad (49)$$

$$= E\int_0^\infty \tilde{P}(r)v(r)r^{2\alpha}\mathrm{d}r.$$

As discussed below, by virtue of our choice of α and boundary conditions on $v(r)$, the vanishing of the boundary term (48) imposes no natural boundary conditions on $\tilde{P}$.

3.1.3 Discretization

To discretize (49), we introduce finite element basis functions $\phi_i(r)$ to form trial and test functions

$$\tilde{P}(r) = \sum_{j=1}^{N} c_j\phi_j(r) \qquad \text{and} \qquad v(r) = \phi_i(r), \qquad (50)$$

and substitute into (49). In so doing, we obtain a generalized eigenvalue problem

$$\sum_{j=1}^{N} H_{ij}c_j = E\sum_{j=1}^{N} S_{ij}c_j, \qquad (51)$$

where H_{ij} is the Hamiltonian matrix,

$$H_{ij} = \int_0^\infty \left[\tfrac{1}{2}\phi_i'(r)\phi_j'(r) + \phi_i(r)\left(V + \frac{l(l+1)-\alpha(\alpha-1)}{2r^2}\right)\phi_j(r)\right]r^{2\alpha}\mathrm{d}r, \qquad (52)$$

and S_{ij} is the overlap matrix,

$$S_{ij} = \int_0^\infty \phi_i(r)\phi_j(r)r^{2\alpha}\mathrm{d}r. \qquad (53)$$

3.1.4 Basis

While, by virtue of the weak formulation, the above discretization can employ any basis in Sobolev space H^1 satisfying the required boundary conditions (including standard C^0 FE bases, C^1 Hermite, and C^k B-splines), we employ a high-order C^0 spectral element (SE) basis [34,35] in the present work. FE bases consist of local piecewise polynomials. SE bases are a form of FE basis constructed to enable well-conditioned, high polynomial-order discretization via definition over Lobatto nodes rather than uniformly spaced nodes within each finite element (mesh interval). In the present work, we employ polynomial orders up to $p = 31$.

In the electronic structure context, SE bases have a number of desirable properties, including:

1. Polynomial completeness and associated systematic convergence with increasing number and/or polynomial order of basis functions.
2. Well conditioned to high polynomial order by virtue of definition over Lobatto nodes.
3. Exponential convergence with respect to polynomial order, enabling high accuracy with a small basis.
4. Applicable to general nonsingular as well as singular Coulombic potentials, whether fixed or self-consistent.
5. Applicable to bound as well as excited states.
6. Can use uniform as well as nonuniform meshes.
7. Basis function evaluation, differentiation, and integration are inexpensive, enabling fast and accurate matrix element computations.
8. Dirichlet boundary conditions are readily imposed by virtue of cardinality, i.e.,

   $$\phi_i(x_j) = \delta_{ij}$$

   where x_j is a nodal point of the basis.
9. Derivative boundary conditions are readily imposed via the weak formulation.
10. Diagonal overlap matrix using Lobatto quadrature, thus enabling solution of a standard rather than generalized eigenvalue problem, reducing both storage requirements and time to solution.

Properties (1)-(4), (7), and (10) are advantageous relative to Slater-type and Gaussian-type bases, which can be nontrivial to converge; can become ill-conditioned as the number of basis functions is increased, limiting accuracies attainable in practice; and produce a non-diagonal overlap matrix, requiring solution of a generalized rather than standard matrix eigenvalue problem.

Properties (2) and (10) are advantageous relative to B-spline bases, which are generally employed at lower polynomial orders in practice and give rise to a non-diagonal overlap matrix, thus requiring solution of a generalized rather than standard matrix eigenvalue problem.

3.1.5 Boundary conditions

Since we seek bound states, which vanish as $r \to \infty$, we impose a homogeneous Dirichlet boundary condition on $v(r)$ and $\tilde{P}(r)$ at $r = \infty$ by employing a finite element basis $\{\phi_i\}$ satisfying this condition.

For $\alpha = 0$, $\tilde{P}(r) = P(r) = 0$ at $r = 0$ for all quantum numbers l according to (42). Thus we impose a homogeneous Dirichlet boundary condition on $v(r)$ and $\tilde{P}(r)$ at $r = 0$ by employing a finite element basis $\{\phi_i\}$ satisfying this condition, whereupon the boundary term (48) as a whole vanishes, consistent with the weak formulation (49).

For $\alpha = 1$, $\tilde{P}(r) = \frac{P(r)}{r} = R(r) = 0$ at $r = 0$ for all quantum numbers $\ell > 0$ according to (42). However, $\tilde{P}(r) \neq 0$ at $r = 0$ for $\ell = 0$. Thus a homogeneous Dirichlet boundary condition cannot be imposed for $\ell = 0$. However, for $\alpha > 0$, the $r^{2\alpha}$ factor in (48) vanishes at $r = 0$, whereupon the boundary term as a whole vanishes, consistent with the weak formulation (49), regardless of boundary condition on $v(r)$ and $\tilde{P}(r)$ at $r = 0$. Hence for $\alpha > 0$, we impose a homogeneous Dirichlet boundary condition on $v(r)$ and $\tilde{P}(r)$ at $r = \infty$ only, by employing a finite element basis $\{\phi_i\}$ satisfying this condition. This is sufficient due to the singularity of the associated Sturm-Liouville problem.

Similarly, for $\alpha = \ell + 1$, $\tilde{P}(r) = \frac{P(r)}{r^{\ell+1}} \neq 0$ at $r = 0$ for all quantum numbers ℓ according to (42) and, since $\alpha > 0$, we again impose a homogeneous Dirichlet boundary condition on $v(r)$ and $\tilde{P}(r)$ at $r = \infty$ only.

In practice, the `featom` code defaults to $\alpha = 0$ with homogeneous Dirichlet boundary conditions at $r = 0$ and $r = \infty$.

Note that, while $\alpha = 0$ yields a rapidly convergent formulation in the context of the Schrödinger equation, it does not suffice in the context of the Dirac equation, as discussed in Section 3.2.4.

3.2 Radial Dirac equation

For small r, the central potential has the form $V(r) = -Z/r + Z_1 + O(r)$, which gives rise to the following asymptotic behavior at the origin for $Z \neq 0$ (Coulombic) [34, 24]:

$$P_{n\kappa}(r) \sim r^{\beta}, \tag{54a}$$

$$Q_{n\kappa}(r) \sim r^{\beta} \frac{c(\beta + \kappa)}{Z}, \tag{54b}$$

$$\beta = \sqrt{\kappa^2 - \left(\frac{Z}{c}\right)^2}. \tag{54c}$$

For $Z = 0$ (nonsingular) the asymptotic is, for $\kappa < 0$ [34]

$$P_{n\kappa}(r) \sim r^{l+1}, \tag{55a}$$

$$Q_{n\kappa}(r) \sim r^{l+2} \frac{E + Z_1}{c(2l + 3)}, \tag{55b}$$

and for $\kappa > 0$

$$P_{n\kappa}(r) \sim -r^{l+2}\frac{E+Z_1}{c(2l+1)}, \tag{56a}$$

$$Q_{n\kappa}(r) \sim r^{l+1}. \tag{56b}$$

For large r, assuming $V(r) \to 0$ as $r \to \infty$, the asymptotic is [33]

$$P_{n\kappa}(r) \sim e^{-\lambda r}, \tag{57a}$$

$$Q_{n\kappa}(r) \sim -\sqrt{-\frac{E}{E+2c^2}}P_{n\kappa}(r), \tag{57b}$$

$$\lambda = \sqrt{c^2 - \frac{(E+c^2)^2}{c^2}} = \sqrt{-2E - \frac{E^2}{c^2}}. \tag{57c}$$

Consistent with the coupled equations (6a) and (6b), the Dirac Hamiltonian can be written as (see, e.g., [7, (7)])

$$H = \begin{pmatrix} V(r) & c\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r}\right) \\ c\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r}\right) & V(r) - 2c^2 \end{pmatrix}. \tag{58}$$

The corresponding eigenvalue problem is then

$$H\begin{pmatrix} P(r) \\ Q(r) \end{pmatrix} = E\begin{pmatrix} P(r) \\ Q(r) \end{pmatrix}. \tag{59}$$

To discretize, we expand the solution vector in a basis:

$$\begin{pmatrix} P(r) \\ Q(r) \end{pmatrix} = \sum_{j=1}^{2N} c_j \begin{pmatrix} \phi_j^a(r) \\ \phi_j^b(r) \end{pmatrix}, \tag{60}$$

where a and b denote the upper and lower components of the vector. We have $2N$ basis functions (degrees of freedom), N for each wave function component. As such, the function $P(r)$ is expanded in terms of basis functions $\phi_i^a(r)$ and the function $Q(r)$ in terms of $\phi_i^b(r)$. However, these two expansions are not independent but are rather coupled via coefficients c_i.

3.2.1 Squared Hamiltonian formulation

The above eigenvalue formulation of the radial Dirac equation can be solved using the finite element method. However, due to the fact that the operator is not bounded from below, one obtains spurious eigenvalues in the spectrum [10]. To eliminate spurious states, we work with the square of the Hamiltonian [22, 15], which is bounded from below, rather than the Hamiltonian itself. Since the eigenfunctions of the square of an operator are the same as those of the operator itself, and the eigenvalues are the squares, the approach is straightforward and enables direct and efficient solution by the finite element method.

Let us derive the equations for the squared radial Dirac Hamiltonian. First we shift the energy by c^2 to obtain the relativistic energy, making the Hamiltonian more symmetric, using (58) to get

$$H + \mathbb{1}c^2 = \begin{pmatrix} V(r) + c^2 & c\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r}\right) \\ c\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{\kappa}{r}\right) & V(r) - c^2 \end{pmatrix}. \tag{61}$$

Then we square the Hamiltonian using (59) to obtain the following equations to solve for $P(r)$ and $Q(r)$:

$$(H+\mathbb{1}c^2)^2\begin{pmatrix}P(r)\\Q(r)\end{pmatrix}=(E+c^2)^2\begin{pmatrix}P(r)\\Q(r)\end{pmatrix}, \tag{62}$$

where $\mathbb{1}$ is the 2×2 identity matrix.

As can be seen, the eigenvectors $P(r)$ and $Q(r)$ are the same as before but the eigenvalues are now equal to $(E+c^2)^2$ and the original non-relativistic energies E can be obtained by taking the square root of these new eigenvalues and subtracting c^2.

3.2.2 Weak form

We follow a similar approach as for the Schrödinger equation: instead of solving for $P(r)$ and $Q(r)$, we solve for $\tilde{P}(r)=\frac{P(r)}{r^\alpha}$ and $\tilde{Q}(r)=\frac{Q(r)}{r^\alpha}$, which introduces a parameter α that can be chosen to facilitate rapid convergence and the application of the desired boundary conditions in a finite element basis. We then multiply the eigenvectors by r^α to obtain $P(r)$ and $Q(r)$.

Now we can write the finite element formulation as

$$A=\int_0^\infty \left(\phi_i^a(r)\quad \phi_i^b(r)\right) r^\alpha (H+\mathbb{1}c^2)^2 r^\alpha \begin{pmatrix}\phi_j^a(r)\\\phi_j^b(r)\end{pmatrix}\mathrm{d}r, \tag{63a}$$

$$S=\int_0^\infty \left(\phi_i^a(r)\quad \phi_i^b(r)\right) r^\alpha r^\alpha \begin{pmatrix}\phi_j^a(r)\\\phi_j^b(r)\end{pmatrix}\mathrm{d}r, \tag{63b}$$

$$Ax=(E+c^2)^2 Sx. \tag{63c}$$

This is a generalized eigenvalue problem, with eigenvectors x (coefficients of $(\tilde{P}(r),\tilde{Q}(r))$), eigenvalues $(E+c^2)^2$, and 2×2 block matrices A and S. Note that the basis functions from both sides were multiplied by r^α due to the substitutions $P(r)=r^\alpha\tilde{P}(r)$ and $Q(r)=r^\alpha\tilde{Q}(r)$. We can denote the middle factor in the integral for A as $H'=r^\alpha\left(H+\mathbb{1}c^2\right)^2 r^\alpha$ and compute it using (61) with rearrangement of r^α factors as follows:

$$H'=r^\alpha\left(H+\mathbb{1}c^2\right)^2 r^\alpha \tag{64a}$$

$$=r^{2\alpha}\begin{pmatrix}V(r)+c^2 & c\left(-\frac{\mathrm{d}}{\mathrm{d}r}+\frac{\kappa-\alpha}{r}\right)\\ c\left(\frac{\mathrm{d}}{\mathrm{d}r}+\frac{\kappa+\alpha}{r}\right) & V(r)-c^2\end{pmatrix}^2. \tag{64b}$$

Let

$$H'=\begin{pmatrix}H^{11} & H^{12}\\H^{21} & H^{22}\end{pmatrix}. \tag{65}$$

Then

$$H^{11} = r^{2\alpha}\left(V(r)+c^2\right)^2 + r^{2\alpha}c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \Phi\right), \tag{66a}$$

$$H^{12} = r^{2\alpha}c\left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} - V'(r)\right), \tag{66b}$$

$$H^{21} = r^{2\alpha}c\left(2\frac{(\kappa+\alpha)}{r}V(r) + 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} + V'(r)\right), \tag{66c}$$

$$H^{22} = r^{2\alpha}\left(V(r)-c^2\right)^2 + r^{2\alpha}c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \Phi\right), \tag{66d}$$

where $\Phi = \dfrac{(\kappa(\kappa+1) - \alpha(\alpha-1))}{r^2}$.

The full derivation of these expressions is given in 7.1.

The usual approach when applying the finite element method to a system of equations is to choose the basis functions in the following form:

$$\phi_i^a(r) = \begin{cases} \pi_i(r) & \text{for } i = 1, \dots, N, \\ 0 & \text{for } i = N+1, \dots, 2N. \end{cases} \tag{67a}$$

$$\phi_i^b(r) = \begin{cases} 0 & \text{for } i = 1, \dots, N, \\ \pi_{i-N}(r) & \text{for } i = N+1, \dots, 2N. \end{cases} \tag{67b}$$

Substituting (67) into (63), we obtain the following expressions after simplification (7.2):

$$A = \begin{pmatrix} A^{11} & A^{12} \\ A^{21} & A^{22} \end{pmatrix}, \tag{68a}$$

$$S = \begin{pmatrix} S^{11} & 0 \\ 0 & S^{22} \end{pmatrix}, \tag{68b}$$

with components given by

$$A_{ij}^{11} = \int_0^\infty \left(c^2\pi_i'(r)\pi_j'(r) + \left(\left(V(r)+c^2\right)^2 + c^2\Phi\right)\pi_i(r)\pi_j(r)\right) r^{2\alpha}\mathrm{d}r, \tag{69a}$$

$$A_{ij}^{22} = \int_0^\infty \left(c^2\pi_i'(r)\pi_j'(r) + \left(\left(V(r)-c^2\right)^2 + c^2\Phi\right)\pi_i(r)\pi_j(r)\right) r^{2\alpha}\mathrm{d}r, \tag{69b}$$

$$A_{ij}^{12} = \int_0^\infty cV(r)\left(+\pi_i'(r)\pi_j(r) - \pi_i(r)\pi_j'(r) + 2\frac{\kappa}{r}\pi_i(r)\pi_j(r)\right) r^{2\alpha}\mathrm{d}r, \tag{69c}$$

$$A_{ij}^{21} = \int_0^\infty cV(r)\left(-\pi_i'(r)\pi_j(r) + \pi_i(r)\pi_j'(r) + 2\frac{\kappa}{r}\pi_i(r)\pi_j(r)\right) r^{2\alpha}\mathrm{d}r, \tag{69d}$$

$$S_{ij}^{11} = S_{ij}^{22} = \int_0^\infty \pi_i(r)\pi_j(r)r^{2\alpha}\mathrm{d}r, \tag{69e}$$

where $\Phi = \dfrac{\kappa(\kappa+1) - \alpha(\alpha-1)}{r^2}$.

These are the expressions implemented in the `featom` code.

3.2.3 Basis

Because we solve the squared Dirac Hamiltonian problem (62), whose spectrum has a lower bound, rather than the Dirac Hamiltonian problem (59), whose spectrum is unbounded above and below, we can employ a high-order C^0 SE basis [35, 36], as in the Schrödinger case, as discussed in Section 3.1.5, with exponential convergence to high accuracy and no spurious states.

As discussed in Section 1, a number of approaches have been developed over the past few decades to solve (59) directly without spurious states, with varying degrees of success. Perhaps most common is to use different bases for large and small components, $P(r)$ and $Q(r)$, of the Dirac wavefunction, e.g., [6, 11, 12, 7, 13, 14]. And among these approaches, perhaps most common is to impose some form of "kinetic balance," e.g., [13, 4] and references therein, on the bases for P and Q. However, imposing such conditions significantly complicates the basis, increasing the cost of evaluations and matrix element integrals, and while highly effective in practice is not guaranteed to eliminate spurious states all cases [13]. In any case, for a C^0 basis as employed in the present work, such balance cannot be imposed since it produces basis functions which are discontinuous and thus not in Sobolev space H^1 as required (Section 3.1.5). Another approach which has proven successful in the context of B-spline bases is to use different orders of B-splines as bases for P and Q [7, 37, 38].

However, this too creates additional complexity [38], did not eliminate all spurious states in our implementation using p and $p+1$ degree SE bases for P and Q, respectively, nor eliminate all spurious states in previous work [39]. Thus we opt instead for the squared Hamiltonian approach in the present work in order to allow the robust and efficient use of a high-order C^0 SE basis, with all desired properties enumerated in Section 3.1.5. In addition, the squared Hamiltonian approach allows direct solution for just the positive (electron) states, rather than having to solve for both positive and negative, e.g., [37], thus saving computation.

Finally, as discussed in Section 3.2.4, we note that by solving for $\tilde{P} = P/r^\alpha$ and $\tilde{Q} = Q/r^\alpha$, with α set according to the known asymptotic as $r \to 0$, rather than for P and Q themselves, we obtain rapid convergence for singular Coulomb as well as nonsingular potentials using our C^0 SE basis. On the other hand, singular Coulomb potentials have posed significant difficulty for B-spline bases [37]. However, since B-splines have polynomial completeness to specified degree, like SE bases, solving for $\tilde{P}$ and $\tilde{Q}$ rather than P and Q directly should yield rapid convergence for such singular potentials using B-spline bases as well.

3.2.4 Boundary conditions

To construct a symmetric weak form, the following two boundary terms are set to zero in the derivation (7.2):

$$c^2 r^{2\alpha} \pi_i(r) \pi_j'(r) \Bigg|_0^\infty = 0\,, \tag{70a}$$

$$c V(r) r^{2\alpha} \pi_i(r) \pi_j(r) \Bigg|_0^\infty = 0\,. \tag{70b}$$

As in the Schrödinger case, since we seek bound states which vanish as $r \to \infty$, we impose a homogeneous Dirichlet boundary condition on the test function $v(r)$ and solutions $\tilde{P}(r)$ and $\tilde{Q}(r)$ at $r = \infty$ by employing a finite element basis $\{\pi_i\}$ satisfying this condition.

Unlike the Schrödinger case, however, the appropriate exponent α and boundary condition at $r = 0$ depends on the regularity of the potential $V(r)$.

For nonsingular potentials $V(r)$, $P(r) \sim r^{\ell+1}$ and $Q(r) \sim r^{\ell+2}$ as $r \to 0$ for relativistic quantum numbers $\kappa < 0$ while $P(r) \sim r^{\ell+2}$ and $Q(r) \sim r^{\ell+1}$ for $\kappa > 0$ according to (55) and (56). Hence, as in the Schrödinger case, for $\alpha = 0$, $\tilde{P}(r) = P(r) = 0$ and $\tilde{Q}(r) = Q(r) = 0$ at $r = 0$ for all quantum numbers κ and ℓ. Thus we impose a homogeneous Dirichlet bounday condition on $v(r)$, $\tilde{P}(r)$, and $\tilde{Q}(r)$ at $r = 0$ by employing a finite element basis $\{\pi_i\}$ satisfying this condition, whereupon the boundary terms (70) vanish, consistent with the weak formulation (63). This is the default in the `featom` code for such potentials.

For singular potentials $V(r)$ with leading term $-\frac{Z}{r}$ (Coulombic), however, $P(r)$ and $Q(r)$ have leading terms varying as r^β as $r \to 0$ for all relativistic quantum numbers κ according to (54). However, while $\beta > 1$ for $|\kappa| > 1$, we have $0.74 < \beta < 0.99998$ for $\kappa = \pm 1$ (for $1 \le Z \le 92$), so that $P(r)$ and $Q(r)$ have non-polynomial behavior at small r and divergent derivatives at $r = 0$, leading to numerical difficulties for methods attempting to compute them directly. To address this issue, we leverage the generality of the formulation (63) to solve for $\tilde{P}(r) = P(r)/r^\alpha$ and $\tilde{Q}(r) = Q(r)/r^\alpha$ with $\alpha = \beta$, rather than solving for $P(r)$ and $Q(r)$ directly. With $\alpha = \beta$, $\tilde{P}(r)$ and $\tilde{Q}(r)$ have polynomial behavior at small r and bounded nonzero values at $r = 0$ for all κ, including $\kappa = \pm 1$, thus eliminating the aforementioned numerical difficulties completely and facilitating rapid convergence in a polynomial basis. Finally, for $\alpha = \beta$, the $r^{2\alpha}$ factors in the boundary terms (70) vanish at $r = 0$, whereupon the boundary terms as a whole vanish, consistent with the weak formulation (63), regardless of boundary condition on $v(r)$, $\tilde{P}(r)$, and $\tilde{Q}(r)$ at $r = 0$. Hence, for $\alpha = \beta$, we impose a homogeneous Dirichlet boundary condition at $r = \infty$ only, by employing a finite element basis $\{\pi_i\}$ satisfying this condition. This is the default in the `featom` code for such potentials.

For integrals involving non-integer α, we employ Gauss-Jacobi quadrature for accuracy and efficiency, while for integrals involving only integer exponents, we use Gauss-Legendre quadrature with the exception of overlap integrals where we employ Gauss-Lobatto quadrature to obtain a diagonal overlap matrix.

4 Results and discussion

To demonstrate the accuracy and performance of the `featom` implementation of the above described finite element formulation, we present results for fixed potentials as well as self-consistent DFT calculations, with comparisons to analytic results where available and to the state-of-the-art `dftatom` solver otherwise. Code outputs are collected in 8.

4.1 Coulombic systems

The accuracy of the Schrödinger and Dirac solvers was verified using the Coulomb potential $V = -\frac{Z}{r}$ for $Z = 92$ (uranium). Eigenvalues are given by the corresponding analytic formula [5]. All eigenvalues with $n < 7$ are used for the study. The reference eigenvalues here are from the analytic solutions: $E_{nl} = -\frac{Z^2}{2n^2}$ for the Schrödinger equation, and

$$E_{n\kappa} = \frac{c^2}{\sqrt{1 + \frac{(Z/c)^2}{(n-|\kappa|+\beta)^2}}} - c^2, \tag{71a}$$

$$\beta = \sqrt{\kappa^2 - (Z/c)^2} \tag{71b}$$

for the Dirac equation.

Figure 1a shows the convergence of the Schrödinger total energy error with respect to the polynomial order p for different numbers of elements N_e. The graph, when observed for a given number of elements, forms a straight line on the log-linear scale, showing that the error decreases exponentially with the polynomial order p until it reaches the numerical precision limit of approximately 10^{-9}.

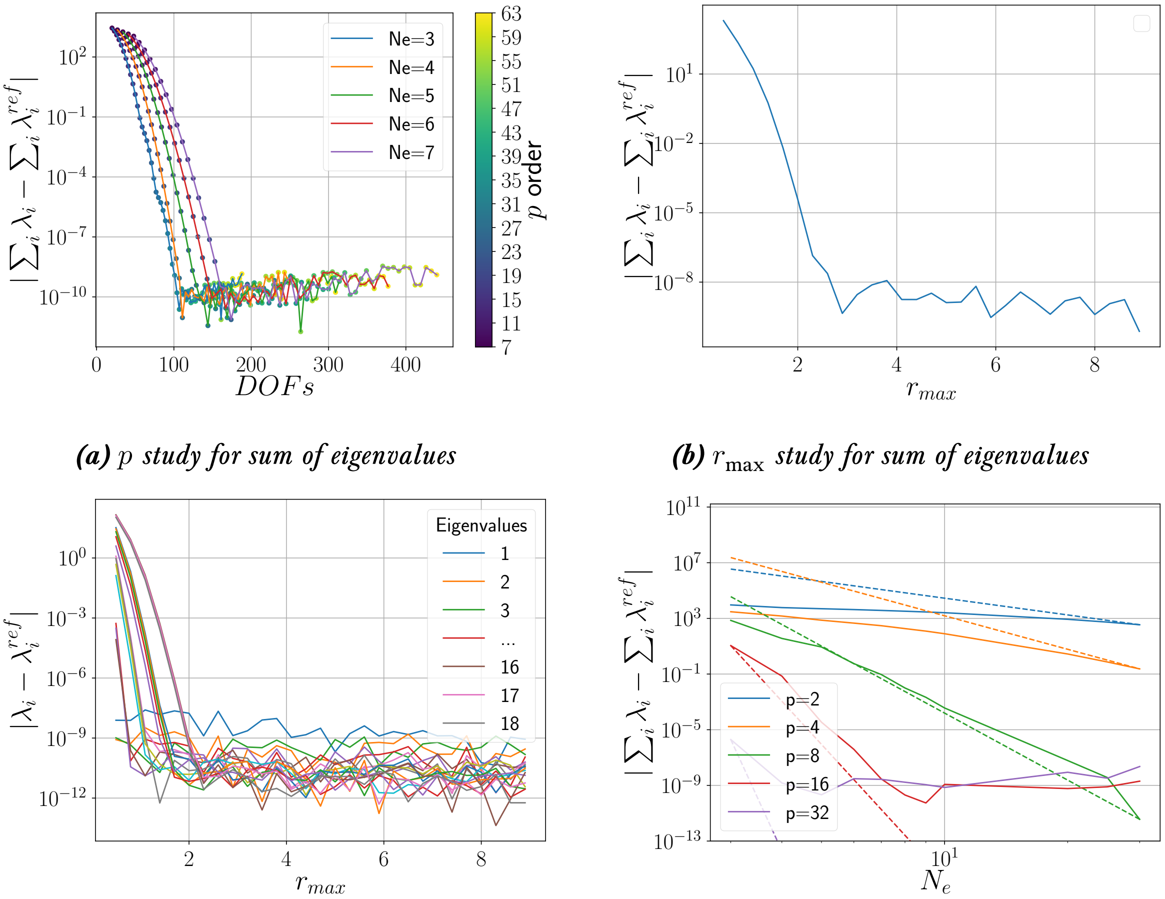
For five elements, we considered the behavior of the error with respect to r_{max}. Figure 1b shows the total energy error. It is observed that $r_{\text{max}} \geq 10$ results in an error of 10^{-8} or less. The eigenvalues converge to 10^{-9} for $r_{\text{max}} \geq 10$, as depicted in Figure 1c.

The theoretical convergence rate for the finite element method is given by N_e^{-2p}. Figure 1d shows the error with respect to N_e, juxtaposed with the theoretical convergence represented as a dotted line. The slope of the solid lines is used to determine the rate of convergence. In all instances, we observe that the theoretical convergence rate is achieved. For polynomial orders $p \leq 8$, it is attained asymptotically, while for $p > 8$, it approaches the limiting numerical precision of around 10^{-9}.

Figure 2a shows the error in the Dirac total energy with respect to p for various N_e. We find the same exponential relationship between the error and polynomial order p.

As before, for five elements, we consider the error with respect to r_{max}. Figure 2b shows the total Dirac energy error, and it is observed that $r_{\text{max}} \geq 10$ results in an error of 10^{-8} or less. The eigenvalues converge to 10^{-8} for $r_{\text{max}} \geq 10$, as depicted in Figure 2c.

Figure 2d shows the error as N_e is increased alongside the theoretical convergence represented as a dotted line. In all instances, we observe that the theoretical convergence



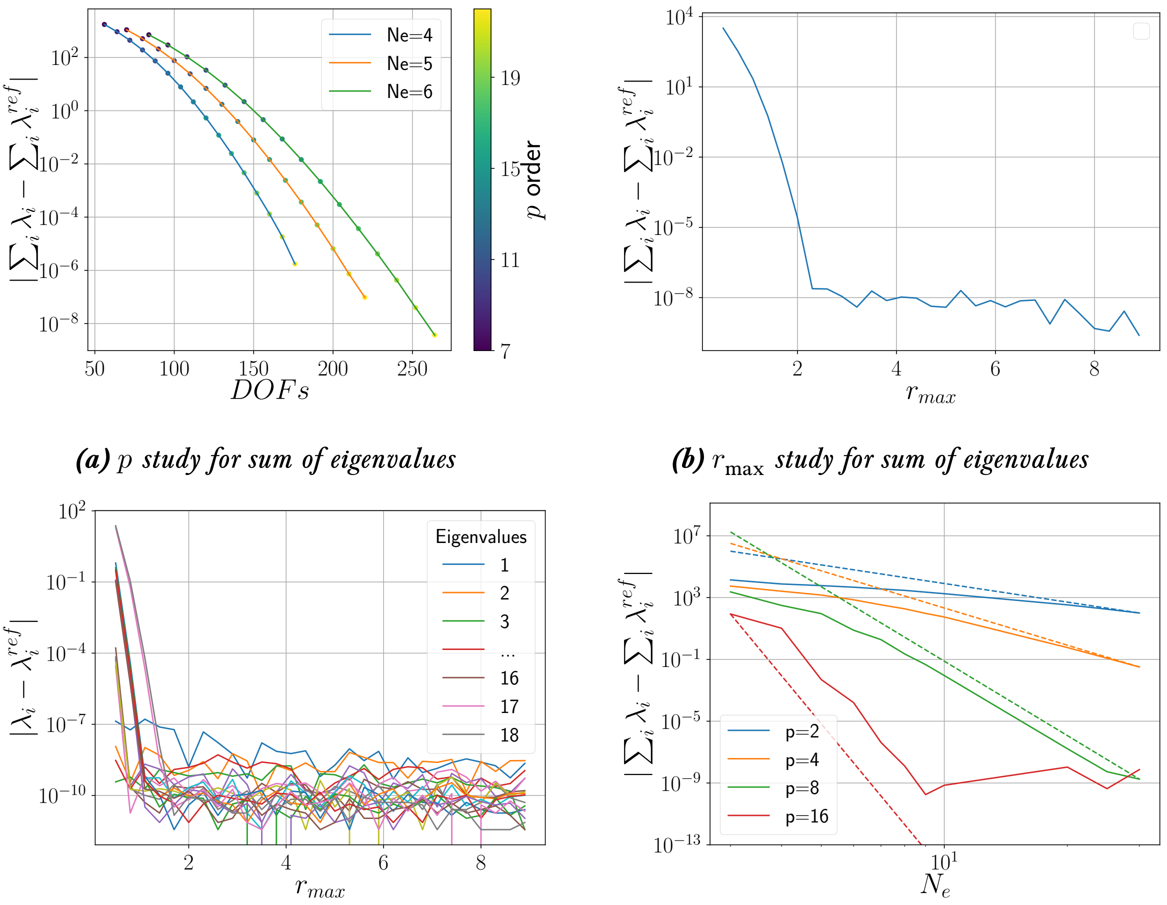
(a) *p study for sum of eigenvalues*

(b) *$r_{\max}$ study for sum of eigenvalues*

(c) *$r_{\max}$ study for eigenvalues*

(d) *N_e study for sum of eigenvalues*

Figure 1. *Convergence studies for the Schrödinger equation with a Coulomb potential with* $Z = 92$.



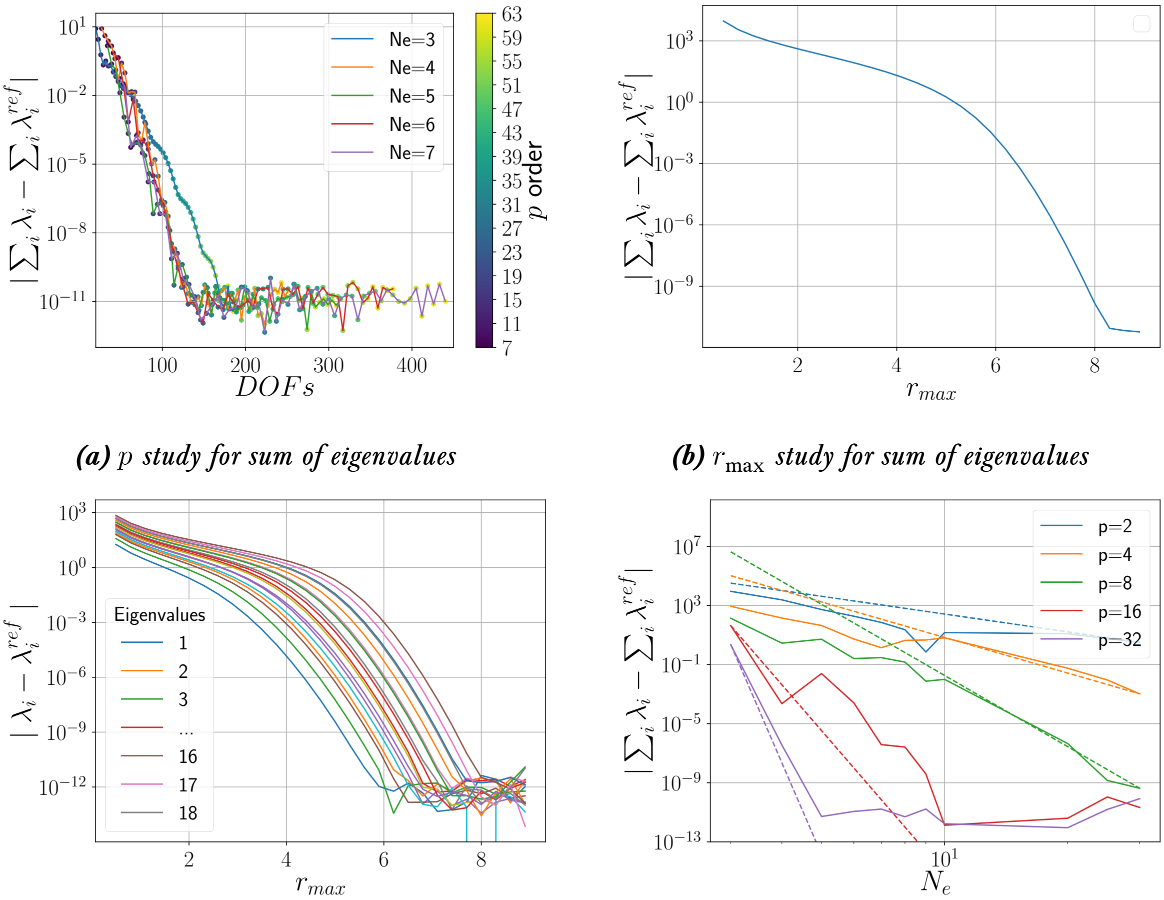
(a) *p study for sum of eigenvalues*

(b) *$r_{\max}$ study for sum of eigenvalues*

(c) *$r_{\max}$ study for eigenvalues*

(d) *N_e study for sum of eigenvalues*

Figure 2. *Convergence studies for the Dirac equation with a Coulomb potential with $Z = 92$.*



***(a)** p study for sum of eigenvalues*

***(b)** $r_{\max}$ study for sum of eigenvalues*

***(c)** $r_{\max}$ study for eigenvalues*

***(d)** N_e study for sum of eigenvalues*

***Figure 3.** Convergence studies for the Schrödinger equation with a harmonic oscillator potential.*

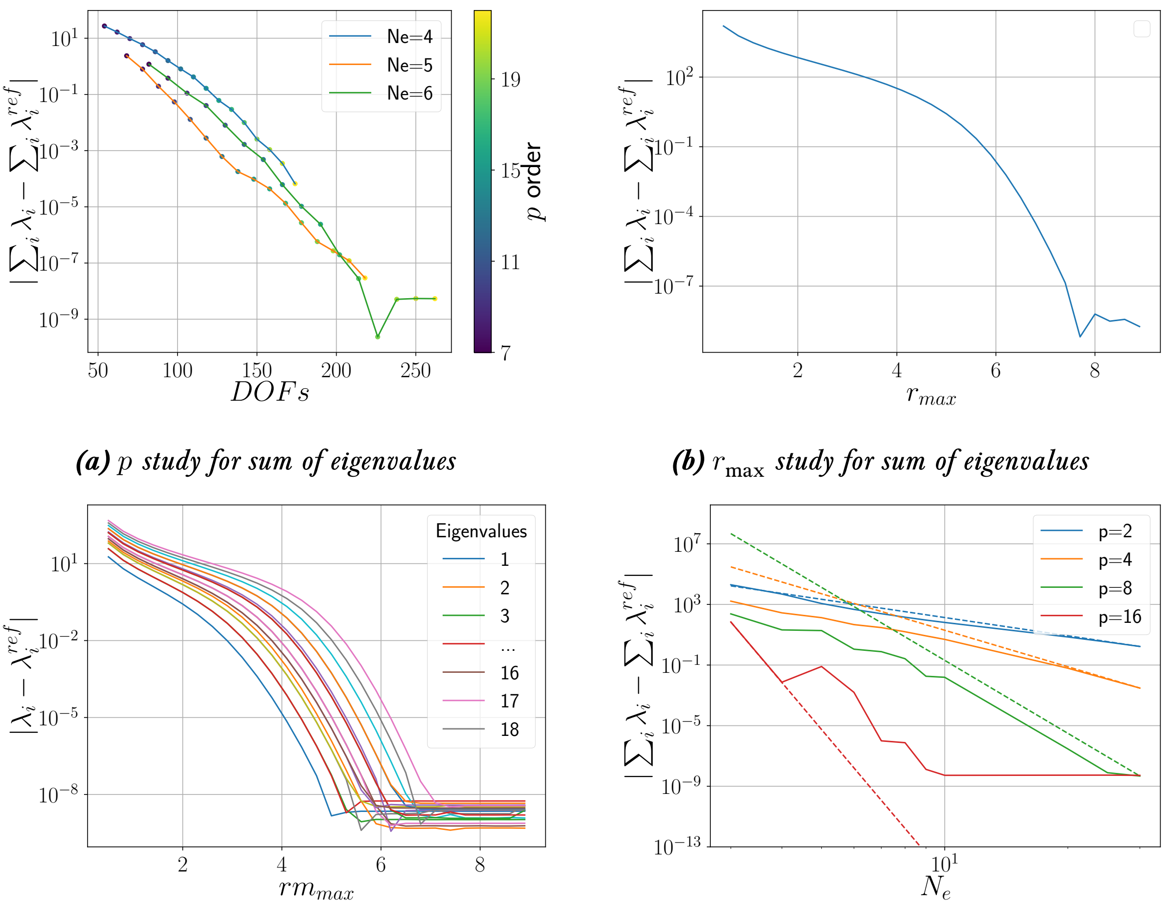
rate is achieved. For polynomial orders $p \leq 8$, it is attained asymptotically, while for $p > 8$, it approaches the limiting numerical precision of around 10^{-9}.

4.2 Quantum harmonic oscillator

Next, we consider a nonsingular potential: the harmonic oscillator potential given by $V(r) = \frac{1}{2}\omega^2 r^2$. For the Schrödinger equation, we take $\omega = 1$ and compare against the exact values given by

$$E_{nl} = \omega(2n - l - \frac{1}{2}). \tag{72}$$

Figure 3a shows the total Schrödinger energy error with respect to the polynomial order p for various N_e values. For fixed N_e, the straight line on the log-linear graph shows the exponential dependency of the error on p until hitting the numerical precision limit (10^{-10}).



***(a)** p study for sum of eigenvalues*

***(b)** $r_{\max}$ study for sum of eigenvalues*

***(c)** $r_{\max}$ study for eigenvalues*

***(d)** N_e study for sum of eigenvalues*

***Figure 4.** Convergence studies for the Dirac equation with a harmonic oscillator potential.*

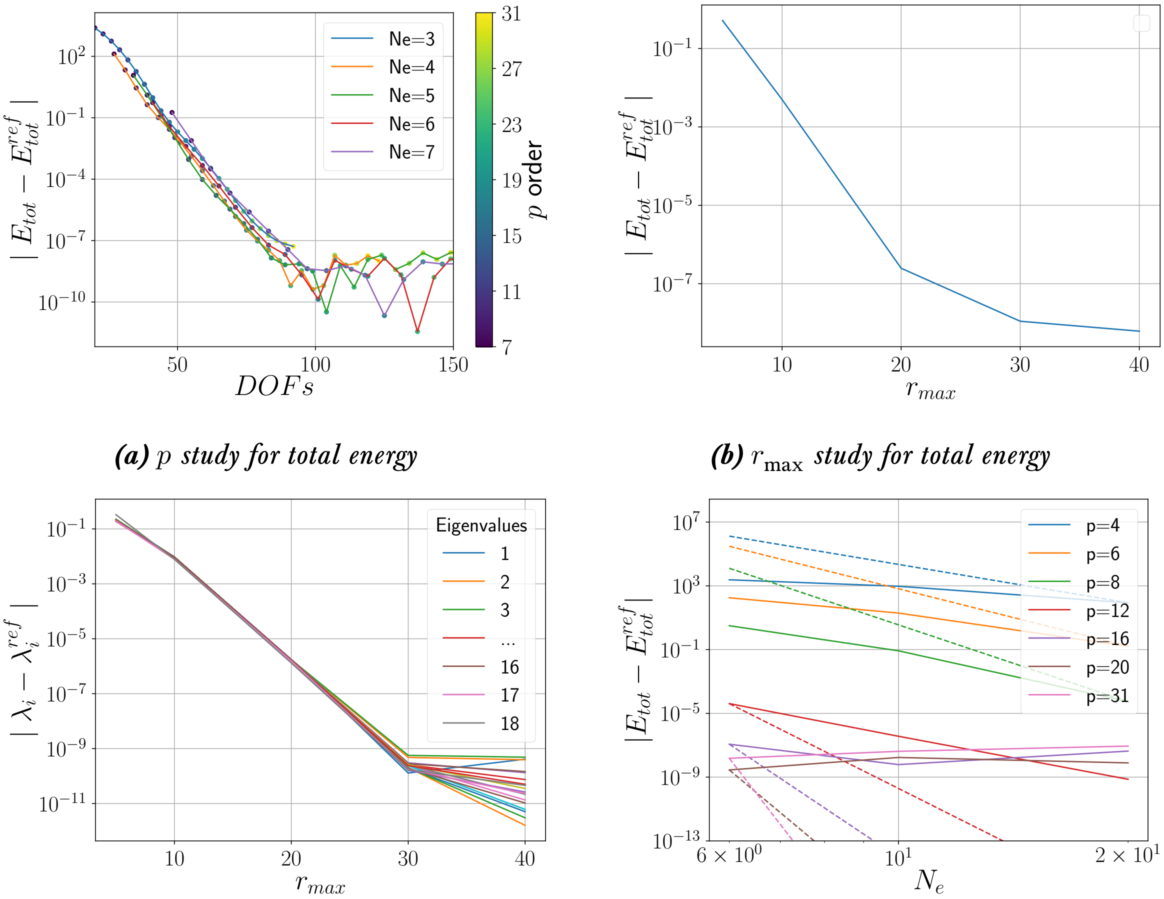
The effect of $r_{\max}$ on the error was tested with five elements. Figure 3b indicates that $r_{\max} \geq 10$ gives a total energy error of 10^{-10} or lower. Figure 3c shows that the eigenvalues converge to 10^{-11} under the same condition.

The dependence of the error on N_e, together with the theoretical convergence rate (N_e^{-2p}), is shown in Figure 3d. The slope of the solid lines confirms that the theoretical convergence rate is achieved. It is asymptotically achieved for $p \leq 8$, while for $p > 8$, it approaches the limiting numerical precision of approximately 10^{-9}.

For the Dirac equation, comparisons are with respect to `dftatom` results.

Figure 4a shows an exponential error decrease with respect to p for various N_e. A similar pattern is observed with $r_{\max} \geq 10$, reducing the total Dirac energy error to 10^{-8} or less (Figure 4b). Eigenvalue convergence to 10^{-8} is also seen (Figure 4c).

The theoretical convergence rate (N_e^{-2p}) is confirmed in Figure 4d, attained asymptotically for $p \leq 8$ and reaching a numerical precision limit of 10^{-9} for $p > 8$.



***(a)** p study for total energy*

***(b)** r_{max} study for total energy*

***(c)** r_{max} study for eigenvalues*

***(d)** N_e study for total energy*

***Figure 5.** Convergence studies for DFT with the Schrödinger equation for uranium.*

4.3 DFT calculations

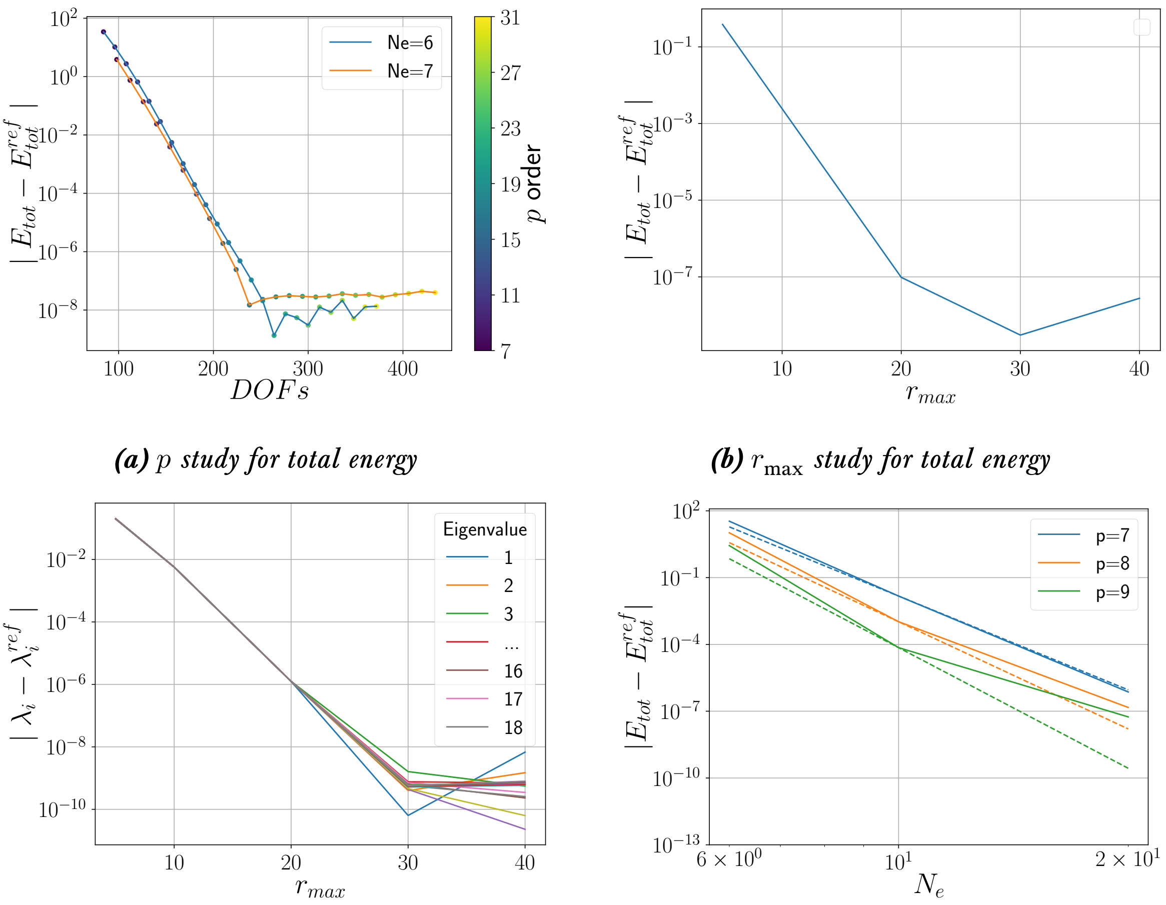
The accuracy of the DFT solvers using both Schrödinger and Dirac equations is compared against `dftatom` results for the challenging case of uranium. The non-relativistic Schrödinger calculation yields the following electronic configuration:

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

With the Dirac solver, the l-shell occupation splits according to the degeneracy of the $j = l + \frac{1}{2}$ and $j = l - \frac{1}{2}$ subshells.

Figures 5a and 6a demonstrate an exponential decrease in total energy error with increasing p for various N_e. As r_{max} increases to ≥ 25 for five elements, the error decreases to 10^{-8} or less (Figures 5b and 6b). Convergence of eigenvalues to 10^{-9} is seen at $r_{\text{max}} \geq 30$ (Figures 5c and 6c).

In Figures 5d and 6d, the theoretical convergence rate (N_e^{-2p}) manifests when $N_e \geq 12$, below which numerical instabilities prevent convergence.



***(a)** p study for total energy*

***(b)** r_{max} study for total energy*

***(c)** r_{max} study for eigenvalues*

***(d)** N_e study for total energy*

***Figure 6.** Convergence studies for DFT with the Dirac equation for uranium.*

4.4 Numerical considerations

The mesh parameters used to achieve 10^{-8} a.u. accuracy for the DFT Schrödinger and Dirac calulations are shown in Table 1. The mesh parameters used for 10^{-6} a.u. accuracy are shown in Table 2.

***Table 1.** Mesh parameters for achieving* 10^{-8} *a.u. accuracy in DFT Schrödinger and Dirac calculations of uranium.*

Parameter	DFT Schrödinger	Dirac
Z	92	92
$r_{\min}$	0	0
$r_{\max}$	50	30
a	200	600
N_e	4	6
N_q	53	64
p	26	25
r_0	—	0.005

***Table 2.** Mesh parameters for achieving* 10^{-6} *a.u. accuracy in DFT Schrödinger and Dirac calculations of uranium.*

Parameter	DFT Schrödinger	Dirac
Z	92	92
$r_{\min}$	0	0
$r_{\max}$	30	30
a	200	100
N_e	4	5
N_q	35	40
p	17	24
r_0	—	0.005

4.5 Benchmarks

The presented `featom` implementation is written in Fortran and runs on every platform with a modern Fortran compiler. To get an idea of the speed, we benchmarked against `dftatom` on a laptop with an Apple M1 Max processor using GFortran 11.3.0. We carry out the uranium DFT calculation to 10^{-6} a.u. accuracy in total energy and all eigenvalues. The timings are as follows:

```
(Apple M1)  featom   dftatom
Schrödinger 28 ms     166 ms
Dirac       360 ms    276 ms
```

We also benchmarked the Coulombic system from section 4.1:

```
(Apple M1)  featom   dftatom
Dirac         49 ms     64 ms
```

5 Summary and conclusions

We have presented a robust and general finite element formulation for the solution of the radial Schrödinger, Dirac, and Kohn–Sham equations of density functional theory; and provided a modular, portable, and efficient Fortran implementation,`featom`[26], along with interfaces to other languages and full suite of examples and tests. To eliminate spurious states in the solution of the Dirac equation, we work with the square of the Hamiltonian rather than the Hamiltonian itself. Additionally, to eliminate convergence difficulties associated with divergent derivatives and non-polynomial variation in the vicinity of the origin, we solve for $\tilde{P} = \frac{P}{r^\alpha}$ and $\tilde{Q} = \frac{Q}{r^\alpha}$ rather than for P and Q directly. We then employ a high-order finite element method to solve the resulting Schrödinger, Dirac, and Poisson equations which can accommodate any potential, whether singular Coulombic or finite, and any mesh, whether linear, exponential, or otherwise. We have demonstrated the flexibility and accuracy of the associated code with solutions of Schrödinger and Dirac equations for Coulombic and harmonic oscillator potentials; and solutions of Kohn–Sham and Dirac–Kohn–Sham equations for the challenging case of uranium, obtaining energies accurate to 10^{-8} a.u., thus verifying current benchmarks [5, 40]. We have shown detailed convergence studies in each case, providing mesh parameters to facilitate straightforward convergence to any desired accuracy by simply increasing the polynomial order.

At all points in the design of the associated code, we have tried to emphasize simplicity and modularity so that the routines provided can be straightforwardly employed for a range of applications purposes, while retaining high efficiency. We have made the code available as open source to facilitate distribution, modification, and use as needed. We expect the present solvers will be of benefit to a broad range of large-scale electronic structure methods that rely on atomic structure calculations and/or radial integration more generally as key components.

6 Acknowledgements

We would like to thank Radek Kolman, Andreas Klöckner and Jed Brown for helpful discussions. This work performed, in part, under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory under Contract DE-AC52-06NA2539 and U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. RG was partially supported by the Icelandic Research Fund, grant number 217436052.

[26] Available on Github: `https://github.com/atomic-solvers/featom`

7 Derivations for the squared radial Dirac finite element formulation

7.1 Components of the Squared Radial Dirac Hamiltonian

Starting from (64), we have H',

$$H' = r^{2\alpha} \begin{pmatrix} V(r) + c^2 & c\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right) \\ c\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right) & V(r) - c^2 \end{pmatrix}^2 .$$

Let

$$r^{-2\alpha} H' = \begin{pmatrix} G^{11} & G^{12} \\ G^{21} & G^{22} \end{pmatrix}, \tag{73}$$

where

$$G^{11} = \left(V(r) + c^2\right)^2 + c^2 \left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right)\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right), \tag{74a}$$

$$G^{12} = \left(V(r) + c^2\right) c \left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right) + c\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right)\left(V(r) - c^2\right), \tag{74b}$$

$$G^{21} = c\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right)\left(V(r) + c^2\right) + \left(V(r) - c^2\right) c \left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right), \tag{74c}$$

$$G^{22} = \left(V(r) - c^2\right)^2 + c^2 \left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right)\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right) \tag{74d}$$

We now simplify each term to obtain

$$G^{11} f = \left(V(r) + c^2\right)^2 f + c^2 \left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right)\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right) f, \tag{75a}$$

$$G^{11} f = \left(V(r) + c^2\right)^2 f + c^2 \left(-\frac{\mathrm{d}^2 f}{\mathrm{d}r^2} + \frac{(\kappa-\alpha)}{r}\frac{\mathrm{d}f}{\mathrm{d}r} + \frac{\left(\kappa^2 - \alpha^2\right) f}{r^2} - \frac{\mathrm{d}}{\mathrm{d}r}\frac{(\kappa+\alpha) f}{r}\right). \tag{75b}$$

Recall that

$$-\frac{\mathrm{d}}{\mathrm{d}r}\frac{(\kappa+\alpha) f}{r} = -(\kappa+\alpha)\left(\frac{1}{r}\frac{\mathrm{d}f}{\mathrm{d}r} - \frac{f}{r^2}\right), \tag{76a}$$

$$-\frac{\mathrm{d}}{\mathrm{d}r}\frac{(\kappa+\alpha) f}{r} = -\frac{(\kappa+\alpha)}{r}\frac{\mathrm{d}f}{\mathrm{d}r} + \frac{(\kappa+\alpha) f}{r^2}. \tag{76b}$$

Substituting (76a) in (75a), with $\Phi = \dfrac{(\kappa(\kappa+1)-\alpha(\alpha-1))}{r^2}$ we have

$$\begin{aligned} G^{11}f &= \left(V(r)+c^2\right)^2 f + c^2\left(-\frac{\mathrm{d}^2 f}{\mathrm{d}r^2} + \frac{(\kappa-\alpha)}{r}\frac{\mathrm{d}f}{\mathrm{d}r} + \frac{(\kappa^2-\alpha^2)f}{r^2}\right. \\ &\qquad \left. -\frac{(\kappa+\alpha)}{r}\frac{\mathrm{d}f}{\mathrm{d}r} + \frac{(\kappa+\alpha)f}{r^2}\right) \qquad (77) \\ &= \left(V(r)+c^2\right)^2 f + c^2\left(-\frac{\mathrm{d}^2 f}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}f}{\mathrm{d}r} + \Phi f\right), \qquad (78) \end{aligned}$$

So we obtain

$$G^{11} = \left(V(r)+c^2\right)^2 + c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \Phi\right). \qquad (79)$$

For G^{22}, we have

$$\begin{aligned} G^{22} &= \left(V(r)-c^2\right)^2 + c^2\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right)\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right) \\ &= \left(V(r)-c^2\right)^2 + c^2\left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(-\kappa-\alpha)}{r}\right)\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(-\kappa+\alpha)}{r}\right). \qquad (80) \end{aligned}$$

Since

$$\begin{aligned} \left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa-\alpha)}{r}\right)\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa+\alpha)}{r}\right) &= \left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r}\right. \\ &\qquad \left. + \frac{(\kappa(\kappa+1)-\alpha(\alpha-1))}{r^2}\right), \qquad (81a) \\ \left(-\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(-\kappa-\alpha)}{r}\right)\left(\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(-\kappa+\alpha)}{r}\right) &= \left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r}\right. \\ &\qquad \left. + \frac{(-\kappa\left(-\kappa+1\right)-\alpha\left(\alpha-1\right))}{r^2}\right), \qquad (81b) \\ &= \left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r}\right. \\ &\qquad \left. + \frac{(\kappa\left(\kappa-1\right)-\alpha\left(\alpha-1\right))}{r^2}\right). \qquad (81c) \end{aligned}$$

Therefore, after substituting (81a) in (80)

$$G^{22} = \left(V(r)-c^2\right)^2 + c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \frac{(\kappa\left(\kappa-1\right)-\alpha\left(\alpha-1\right))}{r^2}\right). \qquad (82)$$

For G^{12}, we have

$$
\begin{aligned}
G^{12}f &= \left(V(r)+c^2\right)c\left(-\frac{\mathrm{d}}{\mathrm{d}r}+\frac{(\kappa-\alpha)}{r}\right)f + c\left(-\frac{\mathrm{d}}{\mathrm{d}r}+\frac{(\kappa-\alpha)}{r}\right)\left(V(r)-c^2\right)f, \\
&= c\frac{(\kappa-\alpha)}{r}\left(V(r)+c^2+V(r)-c^2\right)f - \left(V(r)+c^2\right)c\frac{\mathrm{d}f}{\mathrm{d}r} - c\frac{\mathrm{d}\left(V(r)-c^2\right)f}{\mathrm{d}r}, \\
&= c\left(2\frac{(\kappa-\alpha)}{r}V(r)f - \left(V(r)+c^2\right)\frac{\mathrm{d}f}{\mathrm{d}r} - \left(V(r)-c^2\right)\frac{\mathrm{d}f}{\mathrm{d}r} - V'(r)f\right), \\
&= c\left(2\frac{(\kappa-\alpha)}{r}V(r)f - 2V(r)\frac{\mathrm{d}f}{\mathrm{d}r} - V'(r)f\right),
\end{aligned} \tag{83}
$$

so

$$
G^{12} = c\left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} - V'(r)\right). \tag{84}
$$

Similarly

$$
\begin{aligned}
G^{21}f &= c\left(\frac{\mathrm{d}}{\mathrm{d}r}+\frac{(\kappa+\alpha)}{r}\right)\left(V(r)+c^2\right)f + \left(V(r)-c^2\right)c\left(\frac{\mathrm{d}}{\mathrm{d}r}+\frac{(\kappa+\alpha)}{r}\right)f \\
&= c\frac{(\kappa+\alpha)}{r}\left(V(r)+c^2+V(r)-c^2\right)f + \left(V(r)-c^2\right)c\frac{\mathrm{d}f}{\mathrm{d}r} + c\frac{\mathrm{d}\left(V(r)+c^2\right)f}{\mathrm{d}r} \\
&= c\left(2\frac{(\kappa+\alpha)}{r}V(r)f + \left(V(r)-c^2\right)\frac{\mathrm{d}f}{\mathrm{d}r} + \left(V(r)+c^2\right)\frac{\mathrm{d}f}{\mathrm{d}r} + V'(r)f\right) \\
&= c\left(2\frac{(\kappa+\alpha)}{r}V(r)f + 2V(r)\frac{\mathrm{d}f}{\mathrm{d}r} + V'(r)f\right)
\end{aligned} \tag{85}
$$

which leads to

$$
G^{21} = c\left(2\frac{(\kappa+\alpha)}{r}V(r) + 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} + V'(r)\right). \tag{86}
$$

7.2 Weak formulation of the squared Hamiltonian for the radial Dirac

Starting from (63),

$$
A = \int_0^\infty \begin{pmatrix}\phi_i^a(r) & \phi_i^b(r)\end{pmatrix} H' \begin{pmatrix}\phi_j^a(r) \\ \phi_j^b(r)\end{pmatrix} \mathrm{d}r, \qquad A = \begin{pmatrix}A^{11} & A^{12} \\ A^{21} & A^{22}\end{pmatrix}
$$

and also

$$\phi_i^a(r) = \begin{cases} \pi_i(r) & \text{for } i = 1, \dots, N, \\ 0 & \text{for } i = N+1, \dots, 2N. \end{cases}$$

$$\phi_i^b(r) = \begin{cases} 0 & \text{for } i = 1, \dots, N, \\ \pi_{i-N}(r) & \text{for } i = N+1, \dots, 2N. \end{cases}$$

With $\Phi = \dfrac{(\kappa(\kappa+1) - \alpha(\alpha-1))}{r^2}$ we have

$$A_{ij}^{11} = \int_0^\infty \pi_i(r)\left(r^{2\alpha}(V(r)+c^2)^2 + r^{2\alpha}c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \Phi\right)\right)\pi_j(r)\mathrm{d}r, \tag{87a}$$

$$A_{ij}^{12} = \int_0^\infty \pi_i(r)r^{2\alpha}c\left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} - V'(r)\right)\pi_j(r)\mathrm{d}r, \tag{87b}$$

$$A_{ij}^{21} = \int_0^\infty \pi_i(r)r^{2\alpha}c\left(2\frac{(\kappa+\alpha)}{r}V(r) + 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} + V'(r)\right)\pi_j(r)\mathrm{d}r, \tag{87c}$$

$$A_{ij}^{22} = \int_0^\infty \pi_i(r)\left(r^{2\alpha}(V(r)-c^2)^2 + r^{2\alpha}c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r} + \Phi\right)\right)\mathrm{d}r. \tag{87d}$$

The second term in (87a) and (87d) can be simplified as

$$\int_0^\infty \pi_i(r)r^{2\alpha}c^2\left(-\frac{\mathrm{d}^2}{\mathrm{d}r^2} - \frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r}\right)\pi_j(r)\mathrm{d}r \tag{88a}$$

$$= \int_0^\infty \frac{\mathrm{d}\pi_j(r)}{\mathrm{d}r}c^2\left(r^{2\alpha}\frac{\mathrm{d}\pi_i(r)}{\mathrm{d}r} + 2\alpha r^{2\alpha-1}\pi_i(r)\right)\mathrm{d}r$$

$$- \pi_i(r)\left(r^{2\alpha}c^2\frac{2\alpha}{r}\frac{\mathrm{d}}{\mathrm{d}r}\right)\pi_j(r) - \cancelto{0}{\pi_i(r)r^{2\alpha}c^2\frac{\mathrm{d}\pi_j(r)}{\mathrm{d}r}\bigg|_0^\infty} \tag{88b}$$

$$= \int_0^\infty \frac{\mathrm{d}\pi_j(r)}{\mathrm{d}r}c^2 r^{2\alpha}\frac{\mathrm{d}\pi_i(r)}{\mathrm{d}r}\mathrm{d}r. \tag{88c}$$

Therefore,

$$A_{ij}^{11} = \int_0^\infty r^{2\alpha}\left(\pi_i(r)\left(\left(V(r)+c^2\right)^2 + c^2\Phi\right)\pi_j(r) + \pi_j'(r)c^2\pi_i'(r)\right)\mathrm{d}r \tag{89a}$$

$$A_{ij}^{22} = \int_0^\infty r^{2\alpha}\left(\pi_i(r)\left(\left(V(r)-c^2\right)^2 + c^2\Phi\right)\pi_j(r) + \pi_j'(r)c^2\pi_i'(r)\right)\mathrm{d}r \tag{89b}$$

which implies that A^{11} and A^{22} are symmetric.

The off diagonal terms can be simplified by rewriting the derivative of the potential V' using integration by parts

$$\begin{aligned}
A^{12}_{ij} &= \int_0^\infty \pi_i(r) r^{2\alpha} c \left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} - V'(r)\right)\pi_j(r)\mathrm{d}r && \text{(90a)}\\
&= \int_0^\infty \pi_i(r) r^{2\alpha} c \left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r}\right)\pi_j(r)\mathrm{d}r \\
&\quad + \int_0^\infty \pi_i(r) r^{2\alpha} c \left(-V'(r)\right)\pi_j(r)\mathrm{d}r && \text{(90b)}\\
&= \int_0^\infty \pi_i(r) r^{2\alpha} c \left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r}\right)\pi_j(r)\mathrm{d}r \\
&\quad + \int_0^\infty V(r)\left(\pi_i(r) r^{2\alpha} c \pi_j(r)\right)'\mathrm{d}r + \left.\cancelto{0}{V(r)\pi_i(r)r^{2\alpha}c\pi_j(r)}\right|_0^\infty && \text{(90c)}\\
&= \int_0^\infty \pi_i(r) r^{2\alpha} c \left(2\frac{(\kappa-\alpha)}{r}V(r) - 2V(r)\frac{\mathrm{d}}{\mathrm{d}r}\right)\pi_j(r)\mathrm{d}r \\
&\quad + \int_0^\infty cV(r)\left(\pi_i'(r)\pi_j(r) + \pi_i(r)\pi_j'(r) + 2\pi_i\pi_j\frac{\alpha}{r}\right) r^{2\alpha}\mathrm{d}r && \text{(90d)}\\
&= \int_0^\infty r^{2\alpha} cV(r)\left(-\pi_i(r)\pi_j'(r) + 2\frac{\kappa}{r}\pi_i(r)\pi_j(r) + \pi_i'(r)\pi_j(r)\right)\mathrm{d}r. && \text{(90e)}
\end{aligned}$$

Similarly,

$$\begin{aligned}
A^{21}_{ij} &= \int_0^\infty \pi_i(r) r^{2\alpha} c \left(2\frac{(\kappa+\alpha)}{r}V(r) + 2V(r)\frac{\mathrm{d}}{\mathrm{d}r} + V'(r)\right)\pi_j(r)\mathrm{d}r \\
&= \int_0^\infty r^{2\alpha} cV(r)\left(+\pi_i(r)\pi_j'(r) + 2\frac{\kappa}{r}\pi_i(r)\pi_j(r) - \pi_i'(r)\pi_j(r)\right)\mathrm{d}r && \text{(91)}
\end{aligned}$$

By exchanging the ij indices, we see that $A^{12}_{ij} = A^{21}_{ji}$. Since A^{11} and A^{22} are symmetric and $A^{12}_{ij} = A^{21}_{ji}$, the finite element matrix A is symmetric. The last four equations for A^{11}, A^{22}, A^{12} and A^{21} are the equations that are implemented in the code.

8 Converged runs for systems

The Coulomb potential results and the harmonic Schrödinger results are compared to the analytic results. The remaining systems are compared against `dftatom` [5].

8.1 Schrödinger equation with a Coulomb potential with $Z = 92$

```
  Z  rmax   Ne       a  p Nq DOFs
 92  50.0    7   100.0 31 53  216   -10972.971428571371


 Comparison of calculated and reference energies
```

```
Total energy:
                  E               E_ref     error
-10972.971428571371 -10972.971428571391  2.00E-11
Eigenvalues:
  n                   E               E_ref     error
  1  -4231.999999999996  -4232.000000000000  3.64E-12
  2  -1057.999999999942  -1058.000000000000  5.80E-11
  3  -1057.999999999930  -1058.000000000000  7.03E-11
  4   -470.222222222330   -470.222222222222  1.08E-10
  5   -470.222222222260   -470.222222222222  3.74E-11
  6   -470.222222222187   -470.222222222222  3.56E-11
  7   -264.500000000015   -264.500000000000  1.49E-11
  8   -264.499999999977   -264.500000000000  2.28E-11
  9   -264.499999999988   -264.500000000000  1.24E-11
 10   -264.499999999965   -264.500000000000  3.52E-11
 11   -169.280000000003   -169.280000000000  3.27E-12
 12   -169.279999999990   -169.280000000000  9.66E-12
 13   -169.279999999997   -169.280000000000  2.96E-12
 14   -169.280000000012   -169.280000000000  1.20E-11
 15   -169.279999999994   -169.280000000000  5.68E-12
 16   -117.555555555556   -117.555555555556  7.53E-13
 17   -117.555555555550   -117.555555555556  5.37E-12
 18   -117.555555555554   -117.555555555556  1.71E-12
 19   -117.555555555564   -117.555555555556  8.38E-12
 20   -117.555555555555   -117.555555555556  4.26E-13
 21   -117.555555555565   -117.555555555556  9.02E-12
 22    -86.367346938776    -86.367346938770  6.45E-12
 23    -86.367346938773    -86.367346938770  3.27E-12
 24    -86.367346938773    -86.367346938770  3.40E-12
 25    -86.367346938780    -86.367346938770  1.02E-11
 26    -86.367346938780    -86.367346938770  9.54E-12
 27    -86.367346938787    -86.367346938770  1.73E-11
 28    -86.367346938776    -86.367346938770  6.38E-12
```

8.2 Dirac equation with a Coulomb potential with $Z = 92$

```
Comparison of calculated and reference energies

Total energy:
                  E               E_ref     error
-16991.208873101066 -16991.208873101074  7.28E-12
Eigenvalues:
  n                   E               E_ref     error
  1  -4861.198023119523  -4861.198023119372  1.51E-10
```

```
 2  -1257.395890257783  -1257.395890257889  1.06E-10
 3  -1089.611420919755  -1089.611420919875  1.20E-10
 4  -1257.395890257871  -1257.395890257889  1.82E-11
 5   -539.093341793807   -539.093341793890  8.37E-11
 6   -489.037087678178   -489.037087678200  2.18E-11
 7   -539.093341793909   -539.093341793890  1.82E-11
 8   -476.261595161184   -476.261595161155  2.91E-11
 9   -489.037087678164   -489.037087678200  3.64E-11
10   -295.257844100346   -295.257844100397  5.09E-11
11   -274.407758840011   -274.407758840065  5.46E-11
12   -295.257844100397   -295.257844100397  0.00E+00
13   -268.965877827151   -268.965877827130  2.18E-11
14   -274.407758840043   -274.407758840065  2.18E-11
15   -266.389447187838   -266.389447187816  2.18E-11
16   -268.965877827159   -268.965877827130  2.91E-11
17   -185.485191678526   -185.485191678552  2.55E-11
18   -174.944613583455   -174.944613583462  7.28E-12
19   -185.485191678534   -185.485191678552  1.82E-11
20   -172.155252323719   -172.155252323737  1.82E-11
21   -174.944613583462   -174.944613583462  0.00E+00
22   -170.828937049882   -170.828937049879  3.64E-12
23   -172.155252323751   -172.155252323737  1.46E-11
24   -170.049934288658   -170.049934288552  1.06E-10
25   -170.828937049879   -170.828937049879  0.00E+00
26   -127.093638842631   -127.093638842631  0.00E+00
27   -121.057538029541   -121.057538029549  7.28E-12
28   -127.093638842609   -127.093638842631  2.18E-11
29   -119.445271987144   -119.445271987141  3.64E-12
30   -121.057538029545   -121.057538029549  3.64E-12
31   -118.676410324362   -118.676410324351  1.09E-11
32   -119.445271987144   -119.445271987141  3.64E-12
33   -118.224144624910   -118.224144624903  7.28E-12
34   -118.676410324355   -118.676410324351  3.64E-12
35   -117.925825597409   -117.925825597293  1.16E-10
36   -118.224144624906   -118.224144624903  3.64E-12
37    -92.440787600957    -92.440787600943  1.46E-11
38    -88.671749052020    -88.671749052017  3.64E-12
39    -92.440787600932    -92.440787600943  1.09E-11
40    -87.658287631897    -87.658287631893  3.64E-12
41    -88.671749052013    -88.671749052017  3.64E-12
42    -87.173966671959    -87.173966671948  1.09E-11
43    -87.658287631897    -87.658287631893  3.64E-12
44    -86.888766390952    -86.888766390941  1.09E-11
45    -87.173966671948    -87.173966671948  0.00E+00
46    -86.700519572882    -86.700519572809  7.28E-11
47    -86.888766390948    -86.888766390941  7.28E-12
48    -86.566875102300    -86.566875102359  5.82E-11
```

```
 49    -86.700519572820    -86.700519572809  1.09E-11
```

8.3 Schrödinger equation with a harmonic oscillator potential

```
  Z  rmax   Ne       a  p  Nq  DOFs
 92  50.0    7   100.0 31  64   216      209.999999999991

 Comparison of calculated and reference energies

 Total energy:
                  E               E_ref     error
    209.999999999991    210.000000000000  9.35E-12
 Eigenvalues:
   n                  E                E_ref     error
   1      1.500000000000      1.500000000000  3.41E-13
   2      3.499999999999      3.500000000000  1.43E-12
   3      2.500000000000      2.500000000000  3.60E-14
   4      5.499999999999      5.500000000000  1.27E-12
   5      4.500000000000      4.500000000000  1.22E-13
   6      3.500000000000      3.500000000000  3.87E-13
   7      7.499999999998      7.500000000000  1.59E-12
   8      6.500000000000      6.500000000000  6.84E-14
   9      5.500000000000      5.500000000000  9.41E-14
  10      4.500000000000      4.500000000000  3.95E-13
  11      9.499999999998      9.500000000000  1.71E-12
  12      8.500000000000      8.500000000000  4.33E-13
  13      7.500000000000      7.500000000000  1.64E-13
  14      6.500000000000      6.500000000000  2.39E-13
  15      5.500000000000      5.500000000000  1.23E-13
  16     11.499999999998     11.500000000000  1.68E-12
  17     10.499999999999     10.500000000000  6.34E-13
  18      9.500000000000      9.500000000000  2.36E-13
  19      8.500000000000      8.500000000000  8.70E-14
  20      7.500000000000      7.500000000000  1.68E-13
  21      6.500000000000      6.500000000000  5.24E-14
  22     13.499999999998     13.500000000000  1.64E-12
  23     12.500000000000     12.500000000000  3.57E-13
  24     11.500000000000     11.500000000000  1.30E-13
  25     10.500000000000     10.500000000000  2.38E-13
  26      9.500000000000      9.500000000000  1.07E-14
  27      8.500000000000      8.500000000000  1.90E-13
  28      7.500000000000      7.500000000000  2.22E-14
```

8.4 Dirac equation with a harmonic oscillator potential

```
 Z  rmax   Ne       a  p Nq DOFs
92  50.0    7   100.0 23 53  320      367.470826694655

Comparison of calculated and reference energies

Total energy:
                  E                E_ref     error
   367.470826694655    367.470826700800  6.15E-09
Eigenvalues:
  n              E          E_ref     error
  1     1.49999501     1.49999501  3.46E-11
  2     3.49989517     3.49989517  8.93E-12
  3     2.49997504     2.49997504  5.07E-11
  4     2.49993510     2.49993511  6.83E-10
  5     5.49971548     5.49971548  3.35E-11
  6     4.49983527     4.49983527  1.15E-12
  7     4.49979534     4.49979534  3.15E-09
  8     3.49994176     3.49994176  2.21E-11
  9     3.49987520     3.49987520  2.43E-11
 10     7.49945594     7.49945594  8.47E-11
 11     6.49961565     6.49961565  8.60E-12
 12     6.49957572     6.49957572  3.51E-10
 13     5.49976206     5.49976206  1.70E-11
 14     5.49969551     5.49969551  3.34E-12
 15     4.49989517     4.49989517  1.82E-11
 16     4.49980199     4.49980199  1.92E-11
 17     9.49911657     9.49911657  1.39E-10
 18     8.49931620     8.49931620  9.65E-11
 19     8.49927627     8.49927627  3.27E-10
 20     7.49950252     7.49950252  8.64E-11
 21     7.49943598     7.49943598  6.13E-11
 22     6.49967554     6.49967554  1.81E-11
 23     6.49958238     6.49958238  1.16E-10
 24     5.49983526     5.49983526  3.35E-11
 25     5.49971547     5.49971547  1.40E-11
 26    11.49869739    11.49869739  1.91E-10
 27    10.49893692    10.49893692  1.04E-10
 28    10.49889700    10.49889700  2.10E-10
 29     9.49916315     9.49916315  1.59E-10
 30     9.49909661     9.49909661  1.22E-10
 31     8.49937608     8.49937608  7.65E-11
 32     8.49928292     8.49928292  1.31E-10
 33     7.49957572     7.49957572  7.69E-11
 34     7.49945594     7.49945594  6.51E-11
 35     6.49976205     6.49976205  1.62E-12
```

```
  36     6.49961565     6.49961565  4.14E-11
  37    13.49819839    13.49819839  2.13E-10
  38    12.49847782    12.49847782  1.97E-10
  39    12.49843791    12.49843790  6.08E-10
  40    11.49874396    11.49874396  1.25E-10
  41    11.49867742    11.49867742  1.74E-10
  42    10.49899680    10.49899680  1.30E-10
  43    10.49890365    10.49890365  1.45E-10
  44     9.49923634     9.49923634  1.23E-10
  45     9.49911657     9.49911657  1.47E-10
  46     8.49946258     8.49946258  5.86E-11
  47     8.49931619     8.49931619  1.05E-10
  48     7.49967553     7.49967553  5.61E-11
  49     7.49950251     7.49950251  5.40E-11
```

8.5 DFT with the Schrödinger equation for uranium

```
Total energy:
              E           E_ref     error
-25658.41788885 -25658.41788885  3.17E-09

Eigenvalues:
  n              E           E_ref     error
  1  -3689.35513984  -3689.35513984  7.72E-10
  2   -639.77872809   -639.77872809  3.69E-10
  3   -619.10855018   -619.10855018  3.77E-10
  4   -161.11807321   -161.11807321  1.40E-11
  5   -150.97898016   -150.97898016  3.23E-11
  6   -131.97735828   -131.97735828  1.87E-10
  7    -40.52808425    -40.52808425  4.94E-11
  8    -35.85332083    -35.85332083  6.29E-11
  9    -27.12321230    -27.12321230  8.18E-11
 10    -15.02746007    -15.02746007  1.07E-10
 11     -8.82408940     -8.82408940  5.36E-11
 12     -7.01809220     -7.01809220  5.26E-11
 13     -3.86617513     -3.86617513  5.65E-11
 14     -0.36654335     -0.36654335  1.05E-11
 15     -1.32597632     -1.32597632  4.04E-11
 16     -0.82253797     -0.82253797  2.32E-12
 17     -0.14319018     -0.14319018  1.42E-11
 18     -0.13094786     -0.13094786  5.33E-11
```

8.6 DFT with the Dirac equation for uranium

```
Total energy:
              E           E_ref     error
-28001.13232549 -28001.13232549  2.47E-09
```

```
Eigenvalues:
  n               E           E_ref    error
  1  -4223.41902046  -4223.41902046 5.20E-09
  2   -789.48978233   -789.48978233 3.31E-10
  3   -761.37447597   -761.37447597 4.48E-10
  4   -622.84809456   -622.84809456 4.34E-10
  5   -199.42980564   -199.42980564 2.78E-10
  6   -186.66371312   -186.66371312 4.58E-10
  7   -154.70102667   -154.70102667 5.59E-10
  8   -134.54118029   -134.54118029 5.01E-10
  9   -128.01665738   -128.01665738 4.67E-10
 10    -50.78894806    -50.78894806 4.61E-10
 11    -45.03717129    -45.03717129 4.90E-10
 12    -36.68861049    -36.68861049 5.42E-10
 13    -27.52930624    -27.52930624 5.68E-10
 14    -25.98542891    -25.98542891 4.89E-10
 15    -13.88951423    -13.88951423 5.00E-10
 16    -13.48546969    -13.48546969 5.56E-10
 17    -11.29558710    -11.29558710 5.65E-10
 18     -9.05796425     -9.05796425 5.41E-10
 19     -7.06929563     -7.06929563 5.24E-10
 20     -3.79741623     -3.79741623 5.96E-10
 21     -3.50121718     -3.50121718 5.47E-10
 22     -0.14678838     -0.14678838 5.73E-10
 23     -0.11604716     -0.11604717 6.20E-10
 24     -1.74803995     -1.74803995 7.11E-10
 25     -1.10111900     -1.10111900 6.52E-10
 26     -0.77578418     -0.77578418 6.41E-10
 27     -0.10304081     -0.10304082 5.51E-10
 28     -0.08480202     -0.08480202 5.48E-10
 29     -0.16094728     -0.16094728 3.25E-10
```

References for Paper I

[1] P. Hohenberg and W. Kohn. “Inhomogeneous Electron Gas.” In: *Physical Review* 136.3B (Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864.

[2] W. Kohn and L. J. Sham. “Self-Consistent Equations Including Exchange and Correlation Effects.” In: *Physical Review* 140.4A (Nov. 1965), A1133–A1138. DOI: 10.1103/PhysRev.140.A1133.

[3] Richard M. Martin. *Electronic Structure: Basic Theory and Practical Methods.* Cambridge, UK ; New York: Cambridge University Press, 2004.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Sten Salomonson and Per Öster. "Relativistic All-Order Pair Functions from a Discretized Single-Particle Dirac Hamiltonian." In: *Physical Review A* 40.10 (Nov. 1989), pp. 5548–5558. DOI: 10.1103/PhysRevA.40.5548.

[21] Ying Zhang, Yuxuan Bao, Hong Shen, and Jinniu Hu. "Resolving the Spurious-State Problem in the Dirac Equation with the Finite-Difference Method." In: *Physical Review C* 106.5 (Nov. 2022), p. L051303. DOI: 10.1103/PhysRevC.106.L051303.

[22] 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.

[23] Paul Strange. *Relativistic Quantum Mechanics: With Applications in Condensed Matter and Atomic Physics.* Cambridge University Press, Sept. 1998.

[24] Jan Zabloudil, Robert Hammerling, Lászlo Szunyogh, and Peter Weinberger. *Electron Scattering in Solid Matter: A Theoretical and Computational Treatise.* Springer Science & Business Media, Mar. 2006.

[25] Matyáš Novák, Jiří Vackář, Robert Cimrman, and Ondřej Šipr. "Adaptive Anderson Mixing for Electronic Structure Calculations." In: *Computer Physics Communications* 292 (Nov. 2023), p. 108865. DOI: 10.1016/j.cpc.2023.108865.

[26] Amartya S. Banerjee, Phanish Suryanarayana, and John E. Pask. "Periodic Pulay Method for Robust and Efficient Convergence Acceleration of Self-Consistent Field Iterations." In: *Chemical Physics Letters* 647 (Mar. 2016), pp. 31–35. DOI: 10.1016/j.cplett.2016.01.033.

[27] M. Oulne. "Variation and Series Approach to the Thomas–Fermi Equation." In: *Applied Mathematics and Computation* 218.2 (Sept. 2011), pp. 303–307. DOI: 10.1016/j.amc.2011.05.064.

[28] Miguel A. L. Marques, Micael J. T. Oliveira, and Tobias Burnus. "Libxc: A Library of Exchange and Correlation Functionals for Density Functional Theory." In: *Computer Physics Communications* 183.10 (Oct. 2012), pp. 2272–2281. DOI: 10.1016/j.cpc.2012.05.007.

[29] Susi Lehtola, Conrad Steigemann, Micael J. T. Oliveira, and Miguel A. L. Marques. "Recent Developments in Libxc — A Comprehensive Library of Functionals for Density Functional Theory." In: *SoftwareX* 7 (Jan. 2018), pp. 1–5. DOI: 10.1016/j.softx.2017.11.002.

[30] E. Richard Cohen and Barry N. Taylor. “The 1986 Adjustment of the Fundamental Physical Constants.” In: *Reviews of Modern Physics* 59.4 (1987), pp. 1121–1148.

[31] S. H. Vosko, L. Wilk, and M. Nusair. “Accurate Spin-Dependent Electron Liquid Correlation Energies for Local Spin Density Calculations: A Critical Analysis.” In: *Canadian Journal of Physics* 58.8 (Aug. 1980), pp. 1200–1211. DOI: 10.1139/p80-159.

[32] A. H. MacDonald and S. H. Vosko. “A Relativistic Density Functional Formalism.” In: *J. Phys. C: Solid St. Phys.* 12.2977-2990 (1979).

[33] Walter R. Johnson. *Atomic Structure Theory: Lectures on Atomic Physics.* Berlin ; London: Springer, 2007.

[34] I. P. Grant. “The Dirac Operator on a Finite Domain and the R-matrix Method.” In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 41.5 (Mar. 2008), pp. 1–9. DOI: 10.1088/0953-4075/41/5/055002.

[35] Anthony T Patera. “A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion.” In: *Journal of Computational Physics* 54.3 (June 1984), pp. 468–488. DOI: 10.1016/0021-9991(84)90128-1.

[36] Muhammad Bilal Hafeez and Marek Krawczuk. “A Review: Applications of the Spectral Finite Element Method.” In: *Archives of Computational Methods in Engineering* 30.5 (June 2023), pp. 3453–3465. DOI: 10.1007/s11831-023-09911-2.

[37] 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.

[38] Charlotte Froese Fischer. “Towards B-Spline Atomic Structure Calculations.” In: *Atoms* 9.3 (July 2021), p. 50. DOI: 10.3390/atoms9030050.

[39] Akinori Igarashi. “B-Spline Expansions in Radial Dirac Equation.” In: *Journal of the Physical Society of Japan* 75.11 (Nov. 2006), p. 114301. DOI: 10.1143/JPSJ.75.114301.

[40] Charles Clark. *Atomic Reference Data for Electronic Structural Calculations, NIST Standard Reference Database 141.* 1997. DOI: 10.18434/T4ZP4F.

Paper II

Efficient Implementation of Gaussian Process Regression Accelerated Saddle Point Searches with Application to Molecular Reactions

Rohit Goswami, Maxim Masterov, Satish Kamath, Alejandro Pena-Torres, and Hannes Jónsson

Journal Chemical Theory and Computation, Vol. 21, Issue 16, pp. 1–10

Rohit and Maxim led the development of the C++ code. Rohit performed calculations, created figures and wrote the article.

Abstract

The task of locating first order saddle points on high-dimensional surfaces describing the variation of energy as a function of atomic coordinates is an essential step for identifying the mechanism and estimating the rate of thermally activated events within the harmonic approximation of transition state theory. When combined directly with electronic structure calculations, the number of energy and atomic force evaluations needed for convergence is a primary issue. Here, we describe an efficient implementation of Gaussian process regression (GPR) acceleration of the minimum mode following method where a dimer is used to estimate the lowest eigenmode of the Hessian. A surrogate energy surface is constructed and updated after each electronic structure calculation. The method is applied to a test set of 500 molecular reactions previously generated by Hermes and coworkers [J. Chem. Theory Comput. 18, 6974 (2022)]. An order of magnitude reduction in the number of electronic structure calculations needed to reach the saddle point configurations is obtained by using the GPR compared to the dimer method. Despite the wide range in stiffness of the molecular degrees of freedom, the calculations are carried out using Cartesian coordinates and are found to require similar number of electronic structure calculations as an elaborate internal coordinate method implemented in the Sella software package. The present implementation of the GPR surrogate model in C++ is efficient enough for the wall time of the saddle point searches to be reduced in 3 out of 4 cases even though the calculations are carried out at a low Hartree-Fock level.

Keywords Machine Learning, Transition State, Saddle Search

1 Introduction

An important task in the modelling of thermally induced transitions, such as chemical reactions, diffusion events and conformational changes of molecules, is finding the mechanism and estimating the corresponding rate. For transitions that are slow on the time scale of atomic vibrations, the rate can be estimated using transition state theory and when the atoms are in a confined environment vibrating about a well defined average position the harmonic approximation can suffice. Then, the key challenge is finding first-order saddle points on the energy surface describing the variation of the system with respect to the atomic coordinates. At a first-order saddle point, the atomic forces vanish and the Hessian matrix has a single negative eigenvalue. Several methods to systematically identify saddle points have been developed over the past decades, with varying implementations in software packages (see, for example, Refs. [1, 2, 3, 4]). When combined directly with electronic structure calculations, the key issue is to reduce as much as possible the number of times the energy and atomic forces need to be calculated as each calculation may require hours of computations.

There are two types of saddle point searches. Either both the initial state and the final state minima are specified beforehand, i.e. a two endpoint boundary condition, or only the initial state is specified beforehand, i.e. a single endpoint boundary condition. The former is often approached by discretising the path and converging on a minimum energy path connecting the two endpoints, using for example, the nudged elastic band (NEB) method [5, 6]. The latter is more challenging and can provide unexpected results for the transition mechanism as well as the resulting final state. Such calculations can, therefore be used as the basis for simulations of the long timescale evolution of a system with, e.g. the adaptive kinetic Monte Carlo method [7]. Furthermore, single endpoint methods can provide an efficient way to complete two endpoint calculations when a rough convergence to the saddle point has been obtained [6].

Machine learning can be used to accelerate searches for saddle points in many ways. A potential energy surface can be generated to mimic the atomic interactions in a system using various machine learning approaches with input from electronic structure calculations and then the evaluation of energy and atomic forces in saddle point searches is relatively fast. The accuracy of such energy surfaces is, however, often not good enough in regions near saddle points because training sets tend to lack data from these regions. It is then important to keep retraining the energy surface as information on saddle point regions is acquired. The training of a potential energy surface in this manner requires a large set of electronic structure calculations. When many systems are being screened for a particular property, using for example, workflow engines (e.g. AiiDA [8], Snakemake [9], etc.), the training of a transferable potential energy surface for each candidate is not viable because it requires a large investment in the training set. It is then better to use machine learning to accelerate each saddle point search using just the minimum number of electronic structure calculations required to reach convergence. This is the approach demonstrated here, namely the acceleration of a single saddle point search so as to reduce the number of electronic structure calculations as much as possible.

A single endpoint calculation is often started near the initial state minimum starting with little or no bias for the possible transition mechanism. A climb up the energy surface is then carried out until a saddle point is reached. Another option is to start from a configuration of the atoms that is likely to be close to a saddle point and thereby attempt to reduce the number of steps needed for the search, at the risk of biasing the calculation towards a preconceived transition mechanism. A database of known saddle point configurations possibly augmented by generative machine learning can be used to then suggest saddle point configurations for systems under study [10, 11]. The starting structure may then have significantly higher energy than the saddle point, as will be demonstrated below. In either case, an automated procedure can be used to build a local approximation to the energy surface using input from the electronic structure calculations that need to be carried out as the atomic coordinates are sequentially improved to reach the saddle point geometry. This concept has mostly been used in two endpoint calculations of minimum energy paths, such as the NEB using Gaussian process regression (GPR) [12, 13] or neural networks [14]. A similar approach can, of course, also be used in local minimization [15, 16, 17, 18].

Cartesian coordinates have most often been used in saddle point search algorithms because of the ease of implementation [19, 20, 21, 22]. However, the various degrees of freedom of a molecule can have a wide range in stiffness, for example, vibration of a strong covalent bond vs. rotation of a methyl group. It can, therefore, be advantageous to use “internal coordinates” instead of Cartesian coordinates [23]. An automated construction of internal coordinates can be challenging when considering a wide range of atomic structures. This is in particular the case when near linear configurations of three or more atoms are present. In calculations of heterogeneous catalysis involving the reaction of molecules and surfaces of crystals, such linear arrangements of atoms occur frequently. Recently, an elaborate saddle point search method based on nonredundant internal coordinates has been presented and implemented in software called Sella [24]. The algorithm involves a geodesic update of constrained internal reaction coordinates [25]. When near-linearity in the arrangement of atoms is encountered, ghost atoms are introduced in an automated manner to help define meaningful internal coordinates. The efficacy of the method has been demonstrated in calculations of 500 systems generated using a database approach for predicting saddle point configurations of the atoms. We note that this data set contains only systems where the saddle point searches using Sella converge [24].

Here, as noted earlier, we present a different approach, namely the use of Cartesian coordinates in combination with a surrogate energy surface generated with GPR. A local approximation to the energy surface is generated using the electronic structure calculations carried out during each saddle point search. The eigenvector corresponding to the lowest eigenvalue of the Hessian, the so-called minimum mode, is used to define the search direction and it is found using the dimer method [26]. We refer to this method as the GPR-dimer. An earlier version of this algorithm has been presented [27], but the work presented here uses an improved implementation in C++ within the EON software package [28]. We compare the performance of our method to regular dimer calculations without a surrogate surface, as well as the internal coordinate approach of Sella for the 500 system dataset. Our results show that by using the GPR, the number of electronic structure calculations needed to converge to saddle points can be reduced by an order of magnitude and the overall computational time can be reduced in most cases despite the additional overhead even though the electronic structure calculations are carried out at a low Hartree-Fock (HF) level. A performance similar to that of Sella is found even though Cartesian coordinates are used for this dataset of molecular reactions. This is a promising result because our goal is to use GPR-dimer in calculations of molecular reactions on surfaces of crystals, where near linear configurations of atoms are common and the use of internal coordinates would require the introduction of a number of ghost atoms. In some cases, the GPR dimer method converges on saddle points that are closer to the initial guess, both in terms of atomic coordinates and total energy.

The article is organized as follows: In Section 2, the saddle point search method and GPR is reviewed for completeness. The performance of saddle point searches for the 500 systems is presented in Section 3, and in Section 4, the results are discussed. Conclusions are given in Section 5.

2 Methods

2.1 Minimum mode following with a dimer

The minimum mode following (MMF) method [26, 29] is a technique for iteratively moving from some starting configuration of the atoms to a configuration corresponding to a first order saddle point on the energy surface. The search is guided by the eigenvector corresponding to the lowest eigenvalue of the Hessian, i.e. the minimum mode [26]. If at least one of the eigenvalues of the Hessian is negative, the component of the atomic force in the direction of the minimum mode is inverted to form the search direction. Otherwise the search direction is uphill along the minimum mode. The minimum mode can be found in several ways without explicitly constructing the Hessian.

One option is to use a "dimer" which consists of two replicas of the system (i.e. configurations of all atoms) separated by a small, fixed distance. Letting the two images be denoted as $\mathbf{R}_1$ and $\mathbf{R}_2$, they are positioned on either side of the midpoint $\mathbf{R}$, such that $\mathbf{R}_1 = \mathbf{R} + \Delta R\hat{\mathbf{N}}$ and $\mathbf{R}_2 = \mathbf{R} - \Delta R\hat{\mathbf{N}}$, where $\hat{\mathbf{N}}$ is a unit vector defining the dimer's orientation and ΔR is the distance parameter. Starting from an initial guess for the midpoint $\mathbf{R}$, for example using a preconceived approximate saddle point configuration, and a randomly oriented unit vector $\hat{\mathbf{N}}$, the energy of each image, (E_1 and E_2), with the corresponding atomic forces ($\mathbf{F}_1$, $\mathbf{F}_2$) are evaluated at each image. The dimer is rotated so as to minimize the total dimer energy, E, which is the sum of the energy of the two images, $E = E_1 + E_2$, while keeping the separation between the images fixed. This aligns the orientation of the dimer, $\hat{\mathbf{N}}$, with the minimum mode. An iterative process is used to converge to the direction of the minimum mode to a given tolerance in the force [19]. During the rotation phase, the atomic forces are projected to make them perpendicular to the current dimer orientation, $\mathbf{F}^{\perp}$. Each rotational step involves finding a plane defined by the rotational force and the previous rotation plane, and the rotational force is then zeroed within this plane. Here, a conjugate gradient (CG) method is used where at each step a linear combination of the force vector and the previous search direction is calculated. The coefficient of the previous direction, or the CG parameter, is also a sensitive indicator of convergence. The Polak-Ribiere method [30]:

$$\mathbf{G}_i^{\perp} = \mathbf{F}_i^{\perp} + \beta_i(\mathbf{G}_{i-1}^{\perp} + \hat{\mathbf{N}}_{i-1}^{**}) \tag{1}$$

$$\beta_i = \frac{(\mathbf{F}_i^{\perp} - \mathbf{F}_{i-1}^{\perp}) \cdot \mathbf{F}_i^{\perp}}{\mathbf{F}_i^{\perp} \cdot \mathbf{F}_i^{\perp}} \tag{2}$$

where $\hat{\mathbf{N}}_{i-1}^{**}$ refers to a unit vector on the previous plane of rotation, and perpendicular to the current dimer orientation. The number of energy and force calculations is reduced further at the risk of increasing the error in the estimation of the rotational force by using a small angle approximation [20].

Once the dimer has been oriented sufficiently close to the minimum mode, a translational step moves the midpoint of the dimer forward. If at least one of the eigenvalues of the Hessian is negative, a modified force, $\mathbf{F}^{\dagger}$, is used, where the component of the force along the dimer direction is inverted: $\mathbf{F}_R - 2(\mathbf{F}_R \cdot \hat{\mathbf{N}})\hat{\mathbf{N}}$. This inversion ensures that the dimer moves uphill on the energy surface along the minimum mode and downhill along all other directions. The dimer is iteratively rotated and translated in this way until the atomic forces become zero to a given tolerance, indicating that a saddle point has been located. The limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L-BFGS) on the midpoint of the dimer at each translation step, as it has previously been found to be efficient [21, 31]. The maximum displacement in each iteration is chosen here to be relatively small, 0.05 {Å}, in order to reduce the probability of the dimer search escaping from the saddle point basin of attraction the starting configuration belongs to.

2.2 Gaussian Process Regression

A Gaussian Process Regression (GPR) is a non-parametric method for learning from data. It can be understood as a generalization of multivariate Gaussian distributions to function spaces, effectively providing a distribution over possible functions that fit the observed data. Here, we use a GPR to learn a relationship between Cartesian coordinates of the atoms and their corresponding energy and atomic forces. The GPR model infers this relationship by finding similarities, determined by means of a covariance function (often referred to as the kernel). The expressivity of these models stems from the fact that they may be understood as the infinite limit of a neural network [32] and that any neural network of a finite width is essentially a GP [33].

The inverse-distance modified squared exponential kernel [27, 34] is used

$$k_{1/r}(\mathbf{x},\mathbf{x}') = \sigma_c^2 + \sigma_m^2 \exp\left(-\frac{1}{2}\sum_i \sum_{j>i} \left(\frac{\frac{1}{r_{ij}(\mathbf{x})} - \frac{1}{r_{ij}(\mathbf{x}')}}{l_{\phi(i,j)}}\right)^2\right) \tag{3}$$

where the summations are over all atoms in the system, $(\mathbf{x},\mathbf{x}')$ are the input vectors of atomic coordinates, and $r_{i,j}(\mathbf{x}) = \sqrt{\sum_{d=1}^{3}(x_{i,d} - x_{j,d})^2}$ is the Euclidean distance between atoms i and j within configuration $\mathbf{x}$. $\phi(i,j)$ is a function specifying the type of an atom pair and $l_{\phi(i,j)}$ is the corresponding length scale parameter controlling how quickly the covariance changes as a function of the inverse distance between atoms. The parameter σ_c^2 is a constant variance term while σ_m^2 is a variance scaling factor for the distance-dependent part of the kernel. This form of the kernel is chosen because the inverse distance term, $1/r_{ij}$, penalizes configurations where atoms are close to each other. This takes into account that atoms cannot occupy the same point in space, effectively modeling the strong repulsive forces at short distances.

This form of the kernel is chosen because the inverse distance term, $(\frac{1}{r_{ij}(\mathbf{x})})$, penalizes configurations where atoms are very close to each other. This incorporates the constraint that atoms cannot occupy the same point in space, effectively modeling the strong repulsive forces at short distances.

Given this kernel, the GPR prediction of energy and forces for a new configuration takes the form of a Gaussian distribution. The mean of this distribution provides the predicted energy and forces, while the covariance represents the uncertainty in these predictions. The mean function returns a vector containing predicted energy and the (x, y, z) components of the atomic force on each atom.

The GPR-dimer calculation starts out with up to 6 HF calculations as the dimer is rotated to identify the minimum mode direction, but the usual dimer rotation convergence criteria also apply, so for several systems there are fewer HF calculations in this initial phase. These HF calculations constitute the data set for the initial training of the GPR.

The fitting process involves optimization of the kernel hyperparameters. The computations involve an inversion of a matrix of dimensionality $O(M^3N^3)$ and requires memory on the order of $O(M^2N^2)$, where M is the number of electronic structure calculations that are being used as the training set (i.e. "observations") and N is the number of atomic coordinates, three times the number of atoms. The hyperparameters are optimized using the scaled conjugate gradient method [35]. A Cholesky decomposition is used to speed up the prediction of multiple data points [36].

2.3 Computational specifications

The systems analyzed consist of 500 initial configurations of small organic molecules with between 7 and 25 atoms. The electronic structure calculations are carried out at the HF level with a 3-21G basis set using the NWChem [37] software. The spin unrestricted formalism is used for doublets and spin restricted closed shell formalism is used for the singlet configurations. The self-consistent field (SCF) threshold is set at 10^{-8} Hartree, and the criterion for convergence to a saddle point is that the norm of the atomic force vector drops below 0.01 eV/Å. The electronic structure calculations carried out here are at the same level and use the same software as those of [24].

For the GPR-dimer calculations, which are carried out using EON [28], the geometry verification pre-processing of NWChem is turned off. For both EON and Sella calculations, the ASE [38] file I/O interface to NWChem is used, which means no symmetrization or centering takes place within NWChem, unlike the default settings for NWChem. As mentioned above, the conjugate gradient method [20] is used for the dimer rotations and the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm [39, 21] is used for the translation. Since the validity of the surrogate energy surfaces is limited to the region where data is available, early stopping criteria are used to prevent the calculation from venturing too far outside the region [27, 34].

In some exceptional cases, 10 out of the 500, saddle point searches are aborted. This occurs when: (1) NWChem reports a segmentation fault, or (2) when 1000 iterations have been taken, or (3) when the energy has increased by 20 eV. This is discussed further in the results section. Most of the aborted searches are due to condition (1) when the surrogate model generates atomic structures for which NWChem reports errors.

The calculations are carried out on the IRHPC `elja` computer cluster with `snakemake` as the workflow engine [9]. The cluster nodes have Intel Xeon Platinum 8358 (128 @ 2.600 GHz) CPUs which run Rocky Linux release `8.10`. The Hartree Fock calculations are run with NWChem built with `spack`, and each run requests 16 cores exclusively for NWChem with `3 GB` of memory per node. To reduce I/O overheads, the scratch directory was used. `dvc` was used to vendor intermediate data, and the final `csv` data required for the plots along with the `orgmode` literate programming document [40] is part of the associated monorepo.

The NWChem calculations carried out from EON "cold start" the HF calculations from generated input files on disk at each step. Sella, however, makes use of the i-Pi [41] engine to use previously obtained wave functions as an initial guess and additionally amortize the startup cost associated with running NWChem from scratch on a new set of inputs. Despite this notable disparity in efficiency for the HF calculations, the median increase in time required for the computations using the GPR-dimer is around 9 minutes longer than the Sella calculations.

Similarity measures were computed using the Hausdorff distance as implemented in the iterative rotations and assignment (IRA) shape-matching algorithm [42]. This was also extended to calculate the root mean square deviation, or RMSD for comparing structures.

3 Results

3.1 Savings by using GPR

Figure 1 shows a comparison of the number of HF calculations needed to reach convergence with and without the use of GPR. The same MMF method with the dimer for finding the minimum mode is used in both cases, but for the GPR-dimer, these calculations are carried out on the surrogate energy surface. An order of magnitude reduction in the number of HF calculations is obtained by using the GPR. The median number of calculations needed for convergence drops from 308 to 31. While HF calculations are relatively fast, this difference in the number of electronic structure calculations translates to large savings in computational effort when higher-level methods are used. The GPR surrogate energy surface construction and overhead are independent of the electronic structure method used for generating the data.

In general, the initial guesses of saddle point configurations of the atoms in the Sella benchmark have higher energy than the saddle points. This is unusual as saddle point searches are more often started near an energy minimum corresponding to an initial state and the saddle point then approached from below. The results in Figure 1 are arranged in a histogram according to the energy difference between the initial configuration and the obtained saddle point. This energy difference can even be larger than 10 eV for a couple of systems, but these are excluded from the graphic for visual clarity. The number of systems in the data set binned in 2 eV intervals is shown in Figure 1 (b). Most systems start within 2 eV above the saddle point. Only a few systems start from a configuration that has lower energy than the saddle point. The two sets of calculations, with and without the use of a GPR, most often converge on the same saddle point, but not always, as can be discerned from the color coding of Figure 1 (b). For example, there are more systems that end up on a saddle point that is within 2 eV of the starting point when the GPR is not used, while there are more systems that end up between 4 and 6 eV below the initial point in the GPR-dimer calculations.

More detailed information on this is shown in Figure 2. There the focus is on systems where the energy difference between saddle points obtained with and without the use of the GPR is less than 0.01 eV, a total of 373 cases. The systems are arranged according to the minimized RMSD of the Cartesian coordinates in the initial and converged configurations of the atoms. The range is from 0 to 30 Å. The savings in the number of HF calculations by using the GPR is clearly greater for larger RMSD values. This is to

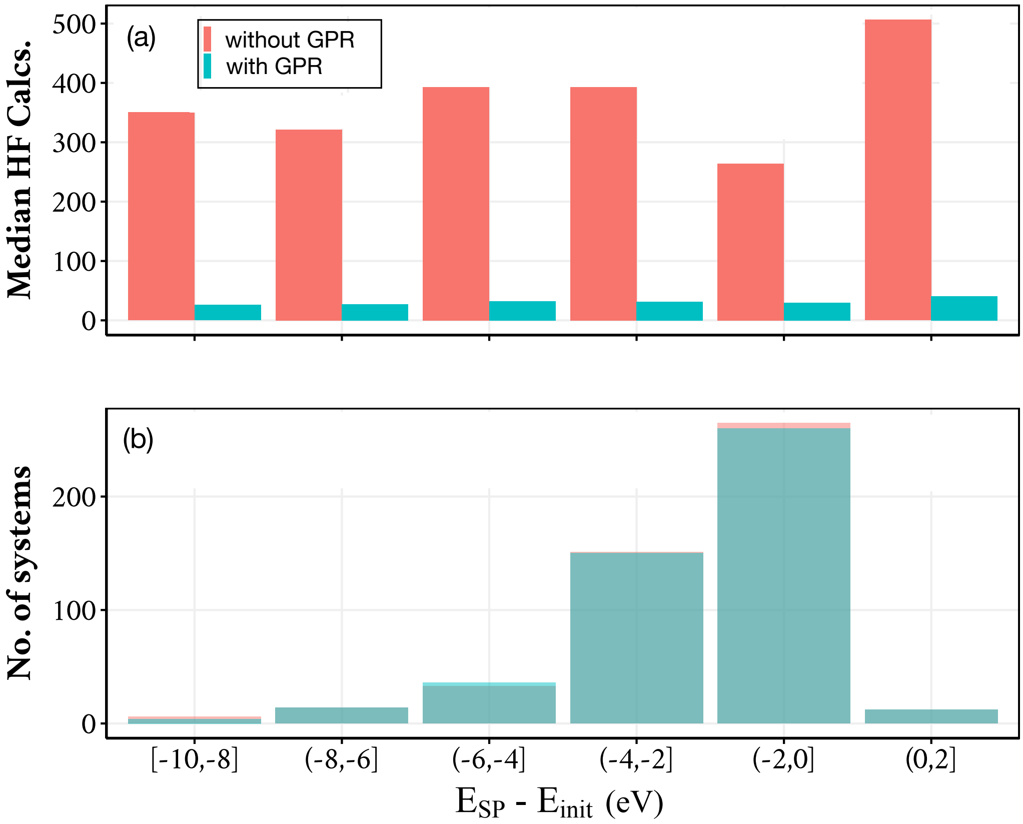


Figure 1. *(a) Comparison of the median number of HF electronic structure calculations needed to converge on a saddle point, with and without the GPR acceleration, in minimum mode following calculations based on the dimer method applied to the 500 system test set of [24]. The median number of HF calculations drops from 308 to 31 by using the GPR. (b) Number of systems where the difference between the energy of the initial atomic structure and that of the converged saddle point is within each of the 2 eV intervals shown on the horizontal axis. In almost all cases, the saddle point has lower energy than the initial configuration, most often within 2 eV, but there are some examples where the difference is even larger than 10 eV.*

be expected, since the reduction in the number of HF calculations comes from being able to take several steps on the GPR surface without having to resample from the true energy surface. The GPR-dimer algorithm effectively dynamically adjusts the step size as seen from the perspective of the sampling of the true energy surface. The median number of HF calculations for this set is 29 when using the GPR-dimer while it is 301 without the use of the GPR. The average reduction is 271 HF calculations. However, the distribution is broad, for three cases, where the RMSD is between 10 and 30 Å, the reduction in the number of calculations is over 1500. The savings by using a GPR is particularly large when the initial guess of the saddle point is poor.

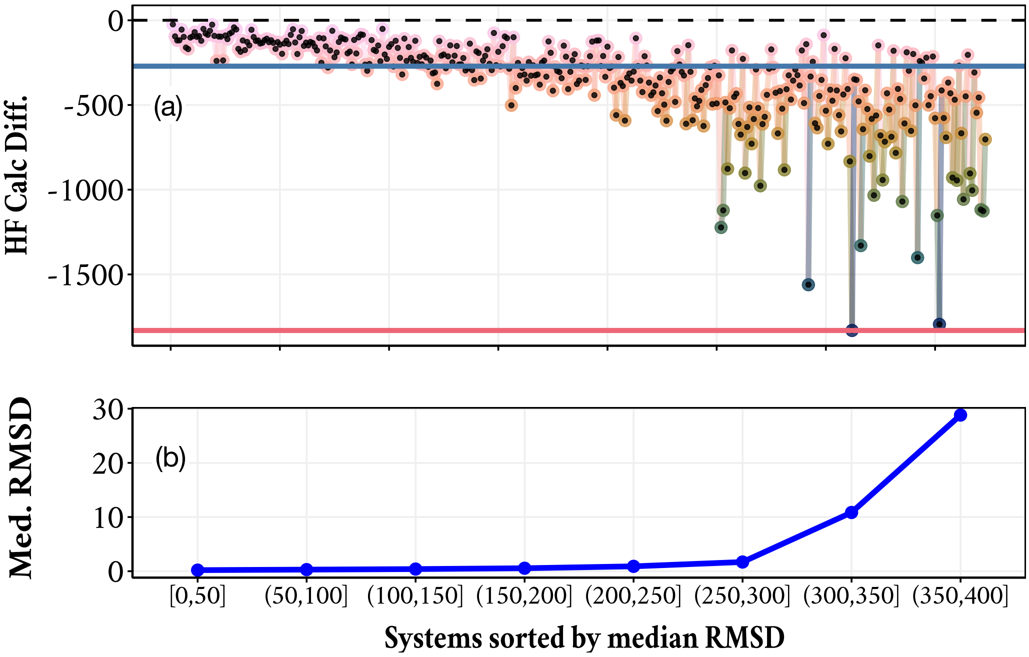


Figure 2. *(a) Difference between the number of HF electronic structure calculations needed to converge on a saddle point with and without the GPR acceleration for each of the systems in the dataset, ordered along the horizontal axis according to the root mean square distance between the initial configuration of the atoms and that of the saddle point found. With the GPR acceleration the median is 29 compared to 301 without it. The mean reduction is indicated by the blue-gray line, 271 HF calculations. Only calculations where the two methods give the same saddle point energy to within 0.01 eV are included, a total of 373 systems. (b) The RMSD distance (in Å) between the initial configuration of the atoms and that of the converged saddle point. Comparison with (a) shows how the efficiency of the GPR acceleration increases with the RMSD distance.*

3.2 Comparing GPRD and Sella

Figure 3 shows a comparison of the number of HF calculations needed to reach convergence using the GPR-dimer and using Sella. The comparison includes 345 systems where the two methods lead to the same saddle point energy to within 0.01 eV. The number of HF calculations is on average similar for the two methods, GPR-dimer requiring a median of 29 HF calculations while Sella requires 31, despite the fact that the GPR-dimer calculations are carried out using Cartesian coordinates while Sella makes use of internal coordinates which are more natural for the large range in stiffness of the molecular degrees of freedom. The GPR-dimer outperforms Sella in 57% cases and the mean improvement is 8 HF calculations fewer than Sella.

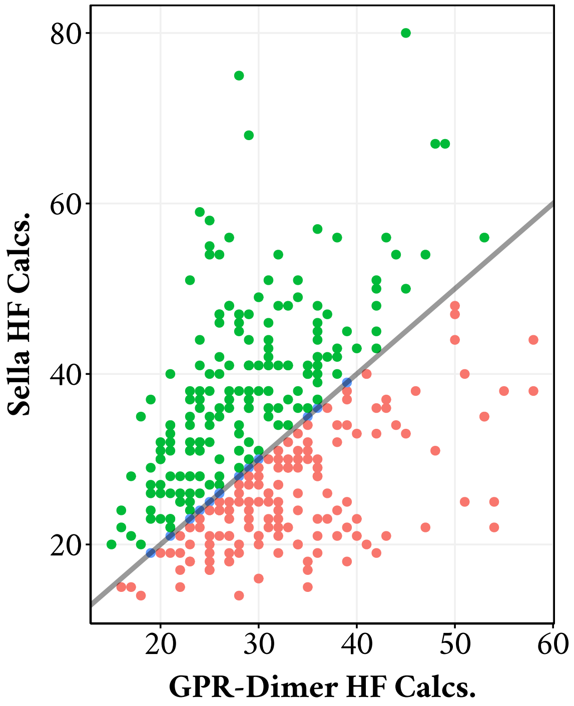


Figure 3. *Comparison of the number of HF electronic structure calculations needed to reach convergence using the GPR-dimer method which uses Cartesian coordinates in combination with GPR acceleration, and the Sella method which makes use of internal coordinates. The number of HF calculations is on average similar for both methods, with the GPR-dimer requiring a median of 29 HF calculations while Sella requires 31. The comparison includes 345 systems where the two methods lead to the same saddle point energy to within 0.01 eV. In several cases, the two methods do not converge to the same saddle point, as illustrated in Figure 4.*

There are several cases where the MMF calculation, with or without GPR, does not converge on the same saddle point as the calculation with Sella. This happens, for example, for the first system in the dataset, `singlet 000`, as illustrated in Figure 4. The system in question is a 16-atom molecule with a stoichiometry of C_5OH_{10}. The initial configuration of the atoms corresponds to an acyclic ether with a fairly large separation between the carbon endpoints of around 2.2 Å. As can be seen in Figure 4, the saddle point found by the GPR-dimer corresponds to the hydrogen from one of the endpoints transitioning to the other. At the saddle point, the C-H bond distances are ≈ 1.25 Å and the energy has dropped by 0.9 eV. The RMSD between the initial configuration of the atoms and the saddle point is only 0.2 Å. The Sella calculation comes close to this configuration during the course of the saddle point search, but instead of converging, it proceeds to the final state of the transition and eventually converges on a saddle point for an irrelevant rotation of a methyl group, 0.4 eV lower in energy than the saddle point found by the GPR-dimer method. The RMSD between this saddle point and the initial configuration is 0.6 Å. The GPR-dimer calculation requires 23 HF calculations while the Sella calculation requires 197. This example illustrates how important it is to check the configuration of atoms obtained in a saddle point search. An NEB calculation of the minimum energy path connecting the initial configuration and the saddle point obtained with Sella shows that the saddle point obtained with the GPR-dimer is indeed located along the path and that the final state of the corresponding transition is an intermediate minimum before the methyl group starts rotating in the Sella calculation (see inset in Figure 4).

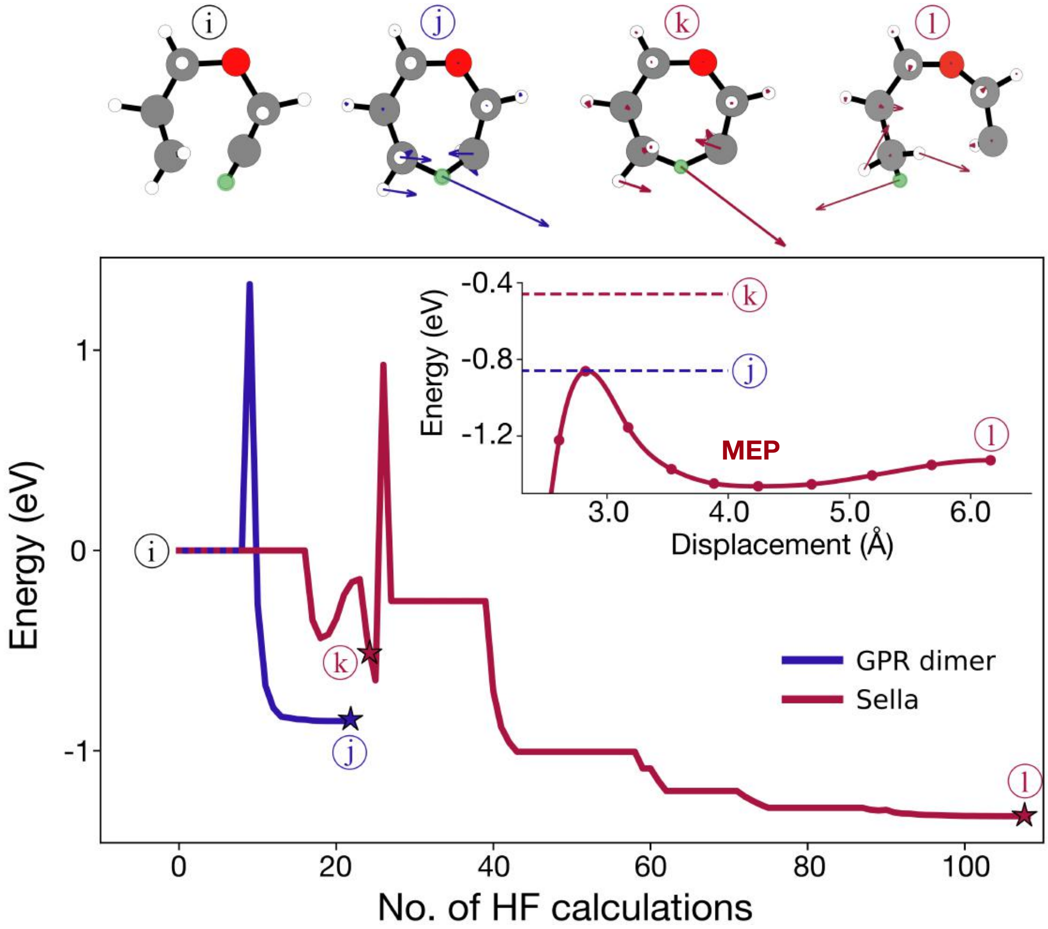


Figure 4. *Subfigure (a) depicts the path taken during the saddle search for the GP-Dimer and Sella. Subfigure (b) shows the results of an NEB between the initial geometry and the saddle found by each system.*

The use of internal coordinates is particularly challenging when a geometry near a linear arrangement of three or more atoms is encountered. There, ghost atoms are introduced algorithmically in Sella to avoid singularities in the internal coordinates. For the 8 systems in the dataset in Figure 2 where this occurs, the GPR-dimer requires 32 HF calculations, close to the average, while Sella requires 44, which is 42% larger than its average.

The first system within the dataset that involves the introduction of ghost atoms is system `singlet 016`. It involves two fragments, a methyl group (CH_3) and a molecule with two ether linkages ($C_4H_7O_2$), i.e. 1-methoxy-2-(2-methoxyethoxy)ethane. The calculation with Sella first increases the distance between the two fragments, but then eventually attaches the methyl group to form a single chain. The saddle point search requires 114 HF calculations and the saddle point reached is 2.4 eV below the initial configuration with an RMSD of 4.2 Å. The GPR-dimer, however, converges after 42 HF calculations to a different saddle point where an H atom jumps onto the methyl group to form %C_4 CH_4 and a CO_2 dissociates from the molecule. This saddle point is 0.4 eV higher than the one found by Sella and the RMSD with respect to the initial configuration is 1.7 Å. As in the `singlet 000` case, the path taken by the Sella calculation comes close to the saddle point found by the GPR-dimer but proceeds to a saddle that is lower in energy and farther away from the initial guess for the saddle point configuration.

4 Discussion

The 500 system data set is constructed in such a way that it only includes cases where Sella converges to some saddle point [24]. For 10 of these systems, the GPR-dimer calculation ends up being aborted using one of the 3 criteria listed in Section 2. In most cases, this occurs because the electronic structure calculation fails and the NWChem software reports an error when atoms have come too close together during the saddle point search on the GPR surrogate energy surface. Also, several of the systems in the dataset involve two separate molecular fragments and in a few cases, the saddle point calculation moves them away from each other until the calculations are aborted. The reason for this is that the direction of the minimum mode then corresponds to the uniform displacement of the two fragments with little change in their internal structure. The energy decreases as the fragments move closer together and so the saddle point search moves the fragments in the opposite direction further apart so as to climb uphill on the energy surface. Similar behaviour is observed in the Sella calculations of these systems. However, in most cases where this occurs, the sideways minimization of the energy eventually leads to the search path turning around and the two fragments then eventually come together in a chemical reaction. However, in a few cases, this does not happen and the GPR-dimer calculation is eventually aborted after a maximum number of iterations has been reached. When the initial configuration of the atoms consists of separated molecular fragments, it could make more sense to start with a local minimization of the energy to bring the fragments to an optimal (relative) position, and then start the saddle point search from a configuration of the atoms that is only slightly perturbed from the minimum energy configuration. Here, we have chosen to work with the input as published by [24].

The application of a saddle point search method that is designed to move uphill in energy is actually questionable when the initial state consists of two molecular fragments since the uphill climb in energy will likely move the two fragments apart. An NEB calculation of the minimum energy path for such a case does not run into this kind of a problem, but then the final state of the reaction needs to be specified. Furthermore, the energy surface of the two fragments may be poorly described with a harmonic approximation. Then, harmonic transition state theory is not expected to provide a good estimate of the rate. In principle, a free energy calculation should be applied in such cases, and, an optimal dividing surface should be determined by maximizing the free energy [43].

An important aspect of the use of a surrogate energy surface is not to venture too far from the region of the data points in the training set since extrapolation is not reliable. This is implemented by early stopping criteria in the GPR-dimer saddle point searches [27, 34]. . The number of times early stopping criteria are applied correlates with longer searches, as the structures added to the training data due to the early stopping are not typically near a saddle point. Here, Euclidean distances have been used to decide on early stopping, but we expect that a Hausdorff distance measure could work better in tandem, and this will be tested in future refinements of the method. Additionally, the Euclidean distance measure is not sensitive to changes across chemical species, which can lead to chemically unfeasible structures being marked as within the cutoff.

The GPR-dimer saddle point search on the surrogate surface can lead to configurations of the atoms that have high energy. The NWChem software used for the HF calculations performs checks on the input structures before calculating the energy and forces, and these are triggered even when the GPR is not used. While the GPR-dimer explores unphysical structures, high energy and atomic forces are in principle not problematic as the GPR can successfully handle such regions. However, NWChem often terminates when atoms come too close and this prevents the GPR-dimer from navigating these regions. We hypothesise that by providing some energy/force value, even a large one representing extreme repulsion, as a pseudo-output from the electronic structure calculation through a wrapper in ASE or EON would allow the GPR-dimer to successfully converge to a saddle point in such cases, while here they represent a failure.

Another improvement that is being pursued for future refinements of the method is the pruning of input data, namely, the elimination of some of the configurations that have been calculated with the electronic structure method early in the search and are far from the saddle point. This will help reduce the computational cost of the matrix inversion and the memory requirement.

In future work we intend to apply these and other methods to larger and more diverse benchmark sets that are not biased towards any particular method. The results of such studies will be reported in an ongoing collaboration to extend the OptBench suite of benchmarks for finding saddle points and minimum energy paths of thermally activated transitions [44].

5 Conclusions

The results presented here demonstrate that saddle point searches for molecular reactions can be performed efficiently using Cartesian coordinates despite the large disparity in the stiffness of the molecular degrees of freedom if the calculation makes use of GPR to generate a surrogate energy surface. The MMF method is used here in combination with a dimer estimate of the minimum mode. The use of a GPR reduces the number of electronic structure calculations needed to reach convergence by an order of magnitude.

The calculations are carried out for a dataset of 500 systems presented previously by [24]. It consists of guesses for saddle points and by construction only includes cases where calculations using Sella are successful in that some saddle point is obtained. A comparison is made between the performance of the GPR-dimer method using Cartesian coordinates and the algorithm in Sella based on internal coordinates measured in terms of the number of HF calculations required for convergence.

The performance of the two methods is found to be similar. However, in several cases the Sella calculation bypasses the saddle point that is closest to the initial guess and converges instead on some rearrangement of the atoms in the final state, such as a rotation of a methyl group. This demonstrates the importance of checking the saddle point configuration of atoms obtained in a single endpoint calculation. This can be done efficiently by using the NEB method to find the minimum energy path as it can reveal whether the saddle point found is adjacent to the initial state configuration or whether another saddle point is present in between, as is illustrated in Figure 4.

While the use of a GPR presents a significant overhead for the saddle point search, the efficient implementation presented here in C++ leads to a reduction in the overall computational time even though the electronic structure calculations are carried out here at a low HF level. When higher level electronic structure calculations are used, the overhead will be an even smaller fraction of the total computational effort. The computational effort in the generation of the GPR surrogate energy surface scales rapidly with the dimensionality of the system and the number of electronic structure results included in the training set. Further optimisation is required in order to be able to apply this approach to systems with a large number of movable atoms. Pruning of the input data to drop points far from the saddle point region is one option and the use of only a subset of the atom coordinates of a large system focusing thereby on the region of interest is another, possibly followed by later refinement of the saddle point found with respect to all coordinates.

6 Acknowledgements

This work was supported by the Icelandic Research Fund (grant no. 217436-053) and by “ReaxPro”, funded by the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 814416.

RG thanks Dr. Miha Gunde, Dr. Amrita Goswami, Dr. Moritz Sallermann, Prof. Debabrata Goswami, Mrs. Sonaly Goswami and Mrs. Ruhila Goswami for discussions.

Reproduction details

The authors confirm that the data supporting the findings of this study are available within the article and/or its supplementary materials. The supplementary material includes: a single repository, which is tagged. These include pinned versions of ASE, EON (from `https://theochemui.github.io/eOn/`), the GPR dimer code, and a copy of the `spack` recipes [45] repository for building NWChem used for the HartreeFock calculations along with spinout utilities from Wailord [46]. A `conda` ecosystem based environment managed by `pixi` is provided to further facilitate reproducibility, with the modules either pinned to versions or bundled within the same repository via `gitsubrepo` following best practices for literate programming [47, 48].

Data related to the results presented in this article and instructions on the generation thereof are available in [**goswamiEfficientImplementationGaussian2025MCDat**] hosted on the Materials Cloud Archive [49].

Conflict of interest

The authors declare no conflict of interest.

References for Paper II

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

[2] Graeme Henkelman, Gísli Jóhannesson, and Hannes Jónsson. "Methods for Finding Saddle Points and Minimum Energy Paths." In: *Theoretical Methods in Condensed Phase Chemistry.* Ed. by Steven D. Schwartz. Vol. 5. Dordrecht: Kluwer Academic Publishers, 2002, pp. 269–302. DOI: 10.1007/0-306-46949-9_10.

[3] 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.

[4] 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.

[5] 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.

[6] Vilhjálmur Ásgeirsson and Hannes Jónsson. "Exploring Potential Energy Surfaces with Saddle Point Searches." In: *Handbook of Materials Modeling.* Ed. by Wanda Andreoni and Sidney Yip. Cham: Springer International Publishing, 2018, pp. 1–26. DOI: 10.1007/978-3-319-42913-7_28-1.

[7] 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.

[8] Sebastiaan P. Huber. "Automated Reproducible Workflows and Data Provenance with AiiDA." In: *Nature Reviews Physics* 4.7 (July 2022), pp. 431–431. DOI: 10.1038/s42254-022-00463-1.

[9] 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.

[10] Sunghwan Choi. "Prediction of Transition State Structures of Gas-Phase Chemical Reactions via Machine Learning." In: *Nature Communications* 14.1 (Mar. 2023), p. 1168. DOI: 10.1038/s41467-023-36823-3.

[11] Ruben Van de Vijver and Judit Zádor. "KinBot: Automated Stationary Point Search on Potential Energy Surfaces." In: *Computer Physics Communications* 248 (Mar. 2020), p. 106947. DOI: 10.1016/j.cpc.2019.106947.

[12] O-P. Koistinen, E. Maras, A. Vehtari, and H. Jónsson. "Minimum Energy Path Calculations with Gaussian Process Regression." In: *Nanosystems: Physics, Chemistry, Mathematics* 7 (Dec. 2016), pp. 925–935. DOI: 10.17586/2220-8054-2016-7-6-925-935.

[13] Olli-Pekka Koistinen, Freyja B. Dagbjartsdóttir, Vilhjálmur Ásgeirsson, Aki Vehtari, and Hannes Jónsson. "Nudged Elastic Band Calculations Accelerated with Gaussian Process Regression." In: *The Journal of Chemical Physics* 147.15 (Sept. 2017), p. 152720. DOI: 10.1063/1.4986787.

[14] Andrew A. Peterson. "Acceleration of Saddle-Point Searches with Machine Learning." In: *The Journal of Chemical Physics* 145.7 (Aug. 2016), p. 074106. DOI: 10.1063/1.4960708.

[15] Malthe K. Bisbo and Bjørk Hammer. "Global Optimization of Atomic Structure Enhanced by Machine Learning." In: *Physical Review B* 105.24 (June 2022), p. 245404. DOI: 10.1103/PhysRevB.105.245404.

[16] 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.

[17] Alexander Denzel, Bernard Haasdonk, and Johannes Kästner. "Gaussian Process Regression for Minimum Energy Path Optimization and Transition State Search." In: *The Journal of Physical Chemistry A* 123.44 (Nov. 2019), pp. 9600–9611. DOI: 10.1021/acs.jpca.9b08239.

[18] Alexander Denzel and Johannes Kästner. "Hessian Matrix Update Scheme for Transition State Search Based on Gaussian Process Regression." In: *Journal of Chemical Theory and Computation* 16.8 (Aug. 2020), pp. 5083–5089. DOI: 10.1021/acs.jctc.0c00348.

[19] 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.

[20] Andreas Heyden, Alexis T. Bell, and Frerich J. Keil. “Efficient Methods for Finding Transition States in Chemical Reactions: Comparison of Improved Dimer Method and Partitioned Rational Function Optimization Method.” In: *The Journal of Chemical Physics* 123.22 (Dec. 2005), p. 224101. DOI: 10.1063/1.2104507.

[21] 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.

[22] 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.

[23] H. Bernhard Schlegel. “Optimization of Equilibrium Geometries and Transition Structures.” In: *Journal of Computational Chemistry* 3.2 (June 1982), pp. 214–218. DOI: 10.1002/jcc.540030212.

[24] 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.

[25] Eric D. Hermes, Khachik Sargsyan, Habib N. Najm, and Judit Zádor. “Geometry Optimization Speedup through a Geodesic Approach to Internal Coordinates.” In: *The Journal of Chemical Physics* 155.9 (Sept. 2021), p. 094105. DOI: 10.1063/5.0060146.

[26] 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.

[27] 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.

[28] 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.

[29] Manuel Plasencia Gutiérrez, Carlos Argáez, and Hannes Jónsson. “Improved Minimum Mode Following Method for Finding First Order Saddle Points.” In: *Journal of Chemical Theory and Computation* 13.1 (Jan. 2017), pp. 125–134. DOI: 10.1021/acs.jctc.5b01216.

[30] E. Polak and G. Ribiere. “Note Sur La Convergence de Méthodes de Directions Conjuguées.” In: *Revue française d’informatique et de recherche opérationnelle. Série rouge* 3.16 (1969), pp. 35–43. DOI: 10.1051/m2an/196903R100351.

[31] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. "Optimization Methods for Finding Minimum Energy Paths." In: *The Journal of Chemical Physics* 128.13 (Apr. 2008), p. 134106. DOI: 10.1063/1.2841941.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* 2nd ed. Springer Series in Operations Research. New York: Springer, 2006.

[40] Rohit Goswami. "Continuous Integration and TeX with Org-Mode." In: *TUGboat* 42.2 (2021), pp. 135–138. DOI: 10.47397/tb/42-2/tb131goswami-ci-org.

[41] 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.

[42] Miha Gunde, Nicolas Salles, Anne Hémeryck, and Layla Martin-Samos. "IRA: A Shape Matching Approach for Recognition and Comparison of Generic Atomic Patterns." In: *Journal of Chemical Information and Modeling* 61.11 (Nov. 2021), pp. 5446–5457. DOI: 10.1021/acs.jcim.1c00567.

[43] Gísli H. Jóhannesson and Hannes Jónsson. "Optimization of Hyperplanar Transition States." In: *The Journal of Chemical Physics* 115.21 (Nov. 2001), pp. 9644–9656. DOI: 10.1063/1.1415499.

[44] Samuel T. Chill, Jacob Stevenson, Victor Ruehle, Cheng Shang, Penghao Xiao, James D. Farrell, David J. Wales, and Graeme Henkelman. "Benchmarks for Characterization of Minima, Transition States, and Pathways in Atomic, Molecular, and Condensed Matter Systems." In: *Journal of Chemical Theory and Computation* 10.12 (Dec. 2014), pp. 5476–5482. DOI: 10.1021/ct5008718.

[45] Todd Gamblin, Matthew LeGendre, Michael R. Collette, Gregory L. Lee, Adam Moody, Bronis R. de Supinski, and Scott Futral. "The Spack Package Manager: Bringing Order to HPC Software Chaos." In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis.* SC '15. New York, NY, USA: ACM, 2015, 40:1–40:12. DOI: 10/gf7tmd.

[46] Rohit Goswami. "Wailord: Parsers and Reproducibility for Quantum Chemistry." In: *Proceedings of the 21st Python in Science Conference* (2022), pp. 193–197. DOI: 10.25080/majora-212e5952-021.

[47] Rohit Goswami and Ruhila S. "High Throughput Reproducible Literate Phylogenetic Analysis." In: *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC).* Nov. 2022, pp. 337–340. DOI: 10.1109/PDGC56933.2022.10053210.

[48] Rohit Goswami, Ruhila S., Amrita Goswami, Sonaly Goswami, and Debabrata Goswami. "Reproducible High Performance Computing without Redundancy with Nix." In: *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC).* Nov. 2022, pp. 238–242. DOI: 10.1109/PDGC56933.2022.10053342.

[49] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper Welzel Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. "Materials Cloud, a Platform for Open Computational Science." In: *Scientific Data* 7.1 (Sept. 2020), p. 299. DOI: 10.1038/s41597-020-00637-5.

Paper III

Bayesian Hierarchical Models for Quantitative Estimates for Performance Metrics Applied to Saddle Search Algorithms

Rohit Goswami

AIP Advances, Vol. 15, Issue 8

Rohit developed and validated the models described, wrote the paper, and made the figures.

Abstract

Rigorous performance evaluation is essential for developing robust algorithms for high-throughput computational chemistry. Traditional benchmarking, however, often struggles to account for system-specific variability, making it difficult to form actionable conclusions. We present a Bayesian hierarchical modeling framework that rigorously quantifies performance metrics and their uncertainty, enabling a nuanced comparison of algorithmic strategies. We apply this framework to analyze the Dimer method, comparing Conjugate Gradient (CG) and L-BFGS rotation optimizers, with and without the removal of external rotations, across a benchmark of 500 molecular systems. Our analysis confirms that CG offers higher overall robustness than L-BFGS in this context. While the theoretically-motivated removal of external rotations led to higher computational cost (>40

Keywords Bayesian statistics, performance measures

1 Introduction

Locating transition states (TS), first-order saddle points on potential energy surfaces (PES), is fundamental to understanding reaction mechanisms and kinetics in chemistry [1, 2]. Numerous algorithms target this challenge, with minimum mode following (MMF) methods forming a prominent class [3]. These algorithms, initiated from a single point on the PES, iteratively ascend the minimum curvature mode while relaxing orthogonal directions. Notable MMF families include Lanczos-based Activation Relaxation Technique Nouveau (ARTn) methods [4, 5, 6, 7, 8] and the two-point Dimer method [9], which uses a pair of images to estimate the minimum mode direction. This is in turn an independent implementation of the hybrid eigenvector-following algorithm [10]. Other variations employ internal coordinates [11] or iteratively build PES approximations [12, 13].

Evaluating and comparing such algorithms typically relies on benchmarks, from established sets like Baker's [14] to more recent collections [15]. However, studies introducing methodological advancements often demonstrate performance on limited sets of small systems, frequently using computationally inexpensive potentials [12, 16, 17, 18, 19, 20]. This traditional benchmarking paradigm appears increasingly misaligned with current computational practice, where peta- and exa-scale resources combined with workflow engines [21, 22] enable large-scale, systematic explorations across diverse chemical spaces. Simple performance comparisons based on small benchmarks or average metrics often neglect significant system-to-system variability and potential implementation biases [23, 24], providing an inadequate basis for selecting optimal methods for demanding high-throughput applications.

To bridge this gap, we advocate for and demonstrate a rigorous statistical framework using Bayesian generalized linear mixed-effects models (GLMMs) for analyzing algorithm performance data from large benchmark sets. This approach, implemented via the `brms` R package [25] interfacing with Stan [26], robustly handles the statistical modeling of various performance metrics (e.g., computation time, PES call counts, binary success) and explicitly models the hierarchical structure of benchmark data (e.g., multiple runs within chemical systems) using random effects. Crucially, the Bayesian paradigm provides comprehensive uncertainty quantification through full posterior distributions and credible intervals, facilitating nuanced and reliable comparisons beyond simple point estimates or visual inspection.

In this work, we apply this Bayesian GLMM framework to analyze the performance trade-offs associated with two common variations within the Dimer method: the choice between Conjugate Gradient (CG) and Limited-memory BFGS (L-BFGS) optimizers for the rotational step [18], and the effect of enabling or disabling the removal of external rotation and translation degrees of freedom [16]. We utilize the EON software package [27] interfaced with NWChem [28] at the HF/3-21G level of theory, analyzing performance across a diverse benchmark set of 500 initial configurations near saddle points introduced by Hermes et al. [11].

The remainder of this paper is structured as follows: Section 2 details the Dimer method variants and the statistical models employed. Section 3 presents the results of applying these models to the benchmark dataset. Section 4 discusses the implications of the findings and the advantages of the statistical approach, and Section 5 provides concluding remarks.

2 Methods

2.1 Dimer method

The Dimer method [9] is a single-ended saddle search method that algorithmically finds first-order saddle point geometries by iteratively refining the position R (a 3N-dimensional vector representing the system coordinates) and orientation $\hat{N}$ (a unit vector in $3N$ dimensions) of a “dimer”. An alternative formulation of the same was demonstrated earlier via the hybrid-eigenvector following using the variational Rayleigh-Ritz to find the eigenvalue and eigenvector pair [10]. The dimer consists of two system images, R_1 and R_2, separated by a small fixed distance ΔR along the dimer axis $\hat{N}$:

$$R_1 = R - \frac{\Delta R}{2}\hat{N} \tag{1}$$

$$R_2 = R + \frac{\Delta R}{2}\hat{N} \tag{2}$$

The core idea is to rotate the dimer orientation $\hat{N}$ to align with the lowest curvature mode at the midpoint R, and then translate R uphill along this mode (if curvature is negative) while minimizing energy in all other directions. This process typically involves two main steps per iteration: rotation and translation.

The goal of the rotation step is to find the orientation $\hat{N}$ that minimizes the dimer’s energy, keeping the midpoint R fixed. This corresponds to aligning $\hat{N}$ with the direction of lowest local curvature. The rotational dynamics are driven by the component of the sum of forces at the endpoints perpendicular to the dimer axis, $F_{rot} = (F_1 + F_2) - ((F_1 + F_2) \cdot \hat{N})\hat{N}$, where $F_1 = F(R_1)$ and $F_2 = F(R_2)$. The rotation aims to drive F_{rot} to zero.

Calculating the exact energy curvature requires the Hessian matrix. To avoid this computational expense, the Dimer method typically employs approximations based only on energies and forces at the endpoints R_1 and R_2. Following the approach refined by Olsen et al. [19], the curvature along the dimer axis $C(\hat{N})$ and an effective torque or rotational force $\tau(\hat{N})$ can be estimated efficiently and iteratively until a threshold for the rotational force is reached. The finite difference approximation for the curvature is:

$$C(\hat{N}) \approx \frac{(F_2 - F_1) \cdot \hat{N}}{\Delta R} \tag{3}$$

The rotational force $\tau(\hat{N})$, which is perpendicular to $\hat{N}$, can also be estimated from F_1 and F_2 [29]. An optimization algorithm, such as Conjugate Gradient (CG) or Limited-memory BFGS (L-BFGS) as investigated in this work, is then used to iteratively update the orientation $\hat{N}$ (typically by small rotations) to minimize the energy with respect to rotation, effectively driving $\tau(\hat{N})$ towards zero.

Once the dimer is aligned with the estimated lowest mode direction $\hat{N}$, the midpoint R is moved towards the saddle point. The translation is guided by an effective force F_{trans}. If the estimated curvature $C(\hat{N})$ is negative (indicating an uphill direction towards the saddle), the force component parallel to $\hat{N}$ is inverted. The effective force for the optimizer is:

$$F_{trans}(R) = F(R) - 2(F(R) \cdot \hat{N})\hat{N} \quad \text{if } C(\hat{N}) < 0 \tag{4}$$

If $C(\hat{N}) \geq 0$, the unmodified force $F(R)$ might be used, or other strategies employed to ensure movement towards a saddle point. A standard optimization algorithm (often L-BFGS in many implementations) then takes a step using this effective force to update the midpoint:

$$R_{k+1} = \text{OptimizerStep}(R_k, F_{trans}(R_k)) \tag{5}$$

The rotation and translation steps are repeated iteratively until convergence criteria, typically based on the magnitude of the total force $F(R)$, are met.

2.2 Statistical models

Quantitative assessment of saddle point search algorithm performance requires rigorous statistical methods, particularly when comparing multiple algorithms across diverse chemical systems. While foundational studies have provided crucial insights [14, 11, 12, 18], analyses often rely on qualitative comparisons or the eyeball norm and summary statistics based on relatively small datasets. Such approaches may not fully capture performance variability or allow for robust statistical inference regarding the significance and magnitude of performance differences. When larger system sets are used they are often hampered by point-estimates of error like the root mean square deviation [30, 31]. It is important to note that much of the saddle search literature for molecular systems makes several approximations valid for small angles or well behaved portions of the energy landscape, which may not hold in practice.

The number of potential energy surface (PES) calls (computational cost) represents count data, total computation time is typically positive and right-skewed, and search success is a binary outcome. These data types often violate the assumptions inherent in standard linear models (e.g., normality, homoscedasticity) [32]. Additionally, comparing algorithm performance typically involves applying multiple methods to the same set of chemical systems, resulting in repeated measures or clustered data. Observations from the same system are not independent, and failing to account for this structure can lead to inaccurate standard errors and potentially flawed conclusions. Variability arising from different software implementations or environments can also present challenges [23], underscoring the need for reproducible analyses.

To analyze the saddle search results, we focus on interpreting three algorithmic concerns, the computational cost, the total time taken, and the success ratio. These are considered in the context of varying the optimizer used in the rotation phase (CG versus LBFS) and the effect of using rotation removal (yes versus no) thus forming 4 sets of measurements on the 500 systems. We employed Bayesian generalized linear mixed-effects models [33]. This framework is well-suited to the data structure and research questions. GLMMs explicitly model the appropriate response distribution for each outcome (counts, continuous time, binary success) using link functions, while incorporating random effects to account for the repeated measures structure (i.e., baseline differences between systems). The Bayesian approach provides full posterior distributions for parameter estimates, facilitating comprehensive uncertainty quantification via credible intervals and enabling principled model comparison [25]. We use `brms` [25] to construct a Stan model [26] which is compiled into C++ code for analysis with the no U turn sampler (NUTS) [34]. These models are so efficient that they are run on a local machine with CPU configuration of 13th Gen Intel i7-1365U (12) @ 5.200GHz with 30 GB of available memory. Details of model fitting, prior specification, and convergence diagnostics are provided in the Supplementary Information.

2.2.1 Effect structures considered

All statistical models employed shared a common approach to constructing the linear predictor, denoted by η_{ij}, which corresponds to observation i on system j. The expected value of the outcome (μ_{ij}) is related to this linear predictor via an appropriate link function $g(\cdot)$, i.e., $g(\mu_{ij}) = \eta_{ij}$, with the specific link function and response distribution detailed below for each outcome type.

Each linear predictor η_{ij} included a random intercepts u_j, to account for the repeated measures within system j, i.e. $u_j \sim \text{Normal}(0, \sigma_u^2)$.

The fixed-effects component was varied systematically to address different analytical questions. The specific fixed-effects structures examined were:

$$\text{Fixed Effects}_{ij} = \begin{cases} \beta_0 + \beta_1 \text{DR}_{i(j))} & \text{(RotOptimizer)} \\ \beta_0 + \beta_2 \text{RR}_{i(j))} & \text{(RotRemoval)} \\ \beta_0 + \beta_1 \text{DR}_{i(j))} + \beta_2 \text{RR}_{i(j))} & \\ \quad +\beta_3 (\text{DR}_{i(j))} \times \text{RR}_{i(j))}) & \text{(Full)} \end{cases} \tag{6}$$

Here, β_k represents fixed-effects coefficients (intercept and slopes), $\mathrm{DR}_{i(j))}$ is an indicator variable for the dimer rotation optimizer used (CG or L-BFGS) for observation i within system j, and $\mathrm{RR}_{i(j))}$ is an indicator variable for rotation removal ('no' or 'yes') within system j. The $\mathrm{DR}_{i(j))} \times \mathrm{RR}_{i(j))}$ term represents their interaction.

The first two effect settings focus on the main effect of a single factor (RotOptimizer or RotRemoval, respectively). When fitted to the complete dataset, these estimate the effect of the included factor while marginalizing over the levels of the excluded factor. Alternatively, as detailed in the Results for specific targeted comparisons, these structures might be applied to subsets of the data (e.g., comparing optimizers only when rotatons are removed).

The third effect setting considers the main effects of the dimer rotation optimizer and the rotation removal. The fourth effect setting represents the full interaction model. It simultaneously estimates the main effects of both factors and their interaction term. This model generally utilizes data from all four conditions per system (where available) to provide the most comprehensive assessment of how the factors jointly influence the outcome.

This approach allows for both focused investigation of individual factor effects, potentially mirroring simpler experimental designs, and a detailed analysis of their interplay using the full statistical power of the dataset within the multilevel framework. The specific response distribution and link function for each outcome metric are detailed in the following sections.

2.2.2 Modeling algorithmic efficiency

The primary figure of merit for saddle search algorithms, particularly when introducing new methodologies, is the number of calls made to the underlying potential energy surface (PES). This metric, referred throughout this manuscript as "PES calls," directly correlates with computational cost, especially when high-level electronic structure methods are employed.

PES calls represent count data [32], which typically violate the assumptions of standard linear models, which assume the response variable is normally distributed. Beyond this, the difficulty of applying frequentist measures of statistical significance like the p-value for applied situations is fraught with controversy [35, 36]. As demonstrated in the Supplementary Information, even linear mixed-effects models, which account for per-molecule variability, can produce misleading results when applied directly to count data without appropriate transformations or distributional assumptions.

For comparing PES calls, we employed Bayesian generalized linear mixed-effects models (GLMMs) with a negative binomial response distribution with a logarithmic link function, which has support on the non negative integers. This distribution is appropriate for overdispersed count data like PES calls [32]. We thus use:

$$\begin{aligned} \mathrm{PESCalls}_{ij} &\sim \mathrm{NegativeBinomial}(\mu_{ij}, \phi) \\ \log(\mu_{ij}) &= \mathrm{Fixed\ Effects}_{ij} + u_j \end{aligned} \tag{7}$$

where $\mathrm{PESCalls}_{ij}$ represents the observed PES calls for the i^{th} observation on system j;

μ_{ij} is the expected value; ϕ is the shape parameter of the negative binomial distribution; Fixed effects Eq. 6 comprise of the overall intercept and effect of the method; and u_j is a random intercept for chemical system j, accounting for the paired nature of the data and modeled as $u_j \sim \text{Normal}(0, \sigma_u^2)$.

2.2.3 Modeling total time

Total time analysis for saddle search algorithms is often perilous, since the bulk of compute intesive calculation are offloaded to external codes like VASP, ORCA etc. which may have strong scaling and depend heavily on the specific functional form of the energy function along with the runtime environment. Additionally, compared to the cost of making PES calls, the time spent within the saddle search algorithm is almost always negligible. Nevertheless, since total wall time elapsed is of practical importance, it provides a valid algorithmic constraint for decision making.

For estimating the effect on total computation time, a similar GLMM structure was employed. Given that time is a continuous, positive, and often right-skewed variable, we used a Gamma response distribution with a logarithmic link function:

$$\begin{aligned} \text{TotalTime}_{ij} &\sim \text{Gamma}(\mu_{ij}, \alpha) \\ \log(\mu_{ij}) &= \text{Fixed Effects}_{ij} + u_j \end{aligned} \tag{8}$$

where TotalTime_{ij} is the computation time for observation i on system j, μ_{ij} is the expected time, α is the shape parameter of the Gamma distribution (inversely related to the variance when the variance is presented in terms of the squared mean). The linear predictor $\log(\mu_{ij})$ incorporates the fixed effects specified in Eq. 6 and the system-specific random intercept u_j, which is modeled as $u_j \sim \text{Normal}(0, \sigma_u^2)$.

2.2.4 Modeling success probabilities

To analyze the factors influencing the likelihood of a successful saddle point search convergence, we fitted a Bayesian generalized linear mixed model appropriate for binary outcomes. This involved using a Bernoulli response distribution and a logit link function:

$$\begin{aligned} &\text{Success}_{ij} \sim \text{Bernoulli}(p_{ij}) \\ &\text{logit}(p_{ij}) = \log\left(\frac{p_{ij}}{1 - p_{ij}}\right) = \text{Fixed Effects}_{ij} + u_j \end{aligned} \tag{9}$$

Here, Success_{ij} is a binary indicator for observation i on system j, p_{ij} is the probability of success, $\text{logit}(p_{ij})$ is the log-odds of success, and the linear predictor incorporates the fixed effects specified in Eq. (6) and the system-specific random intercept u_j, which is modeled as $u_j \sim \text{Normal}(0, \sigma_u^2)$.

This model allows us to estimate how the different factors influence the odds of a successful search, while accounting for baseline differences in success rates across systems.

3 Applications

To demonstrate the utility of leveraging a Bayesian approach to performance statistices, we analyse the performance of the dimer method variants on the dataset of 500 initial configurations for small gas-phase organic molecules (7-25 atoms) introduced by Hermes, Sargsyan, Najm, and Zádor [11]. Saddle point searches were conducted using the EON software package [27] [27], interfaced with NWChem [28] as the quantum mechanical engine. Energies and forces were calculated at the Hartree-Fock (HF) level of theory using RHF (restricted HF) for singlets and UHF (unrestricted HF) for doublets [37] with the 3-21G basis set. Saddle searches algorithms were considered converged when the maximum component of the atomic forces fell below 0.01 eV/Å. This rather loose tolerance was chosen to highlight the effect of rotation removal, as the step sizes decrease on approaching the saddle and are less likely to lead to spurious rotation. Additionally, tighter tolerances tend to put more focus on the parameters of the underlying computational engine which is not the primary software of interest here. Comprehensive details regarding computational parameters, including self-consistent field thresholds, specific EON settings, and interface configurations, are provided in the Supplementary Information.

All models involved pairwise comparisons of dimer control modalities, using data filtered to include only systems where both methods in the pair successfully converged, and with the same saddle, with an energy difference of less than 0.01eV. Full interaction models are also fit. Since both PES calls and total time are proxies for computational effort, only the PES calls are presented here, with the total time model in the Supplementary.

3.1 Data overview

We applied the four dimer method variants (combinations of CG/L-BFGS rotation optimizers and enabling/disabling rotation removal) to the benchmark set of 500 initial configurations. An initial exploratory analysis of the performance metrics provides context for the subsequent Bayesian modeling, focusing here on the number of potential energy surface (PES) calls for successful runs and overall convergence success.

Figures 1 and 2 illustrate the difference in PES calls required for convergence, comparing pairs of calculations that reached the same saddle point (within 0.01 eV). When rotation removal was disabled (Fig. 1, 440 comparable systems), the CG-rotations dimer required fewer PES calls than L-BFGS-rotations in a majority of cases (55%, median reduction 21 calls), although L-BFGS is rather comparable in performance (in 43% cases, median reduction 16 calls). With CG-rotations (Fig. 2, 465 comparable systems), disabling rotation removal was markedly more efficient, requiring fewer PES calls than enabling it in over 99% of systems (median reduction 126 calls). Both figures indicate that performance differences tend to be larger for systems starting with higher initial RMSD values relative to the saddle point.

Beyond the computational cost for successful runs, the overall convergence success is critical. Figure 3 presents a comparison of binary success outcomes across all 500 sys-

[27] Using the Github variant: `https://github.com/TheochemUI/eOn`

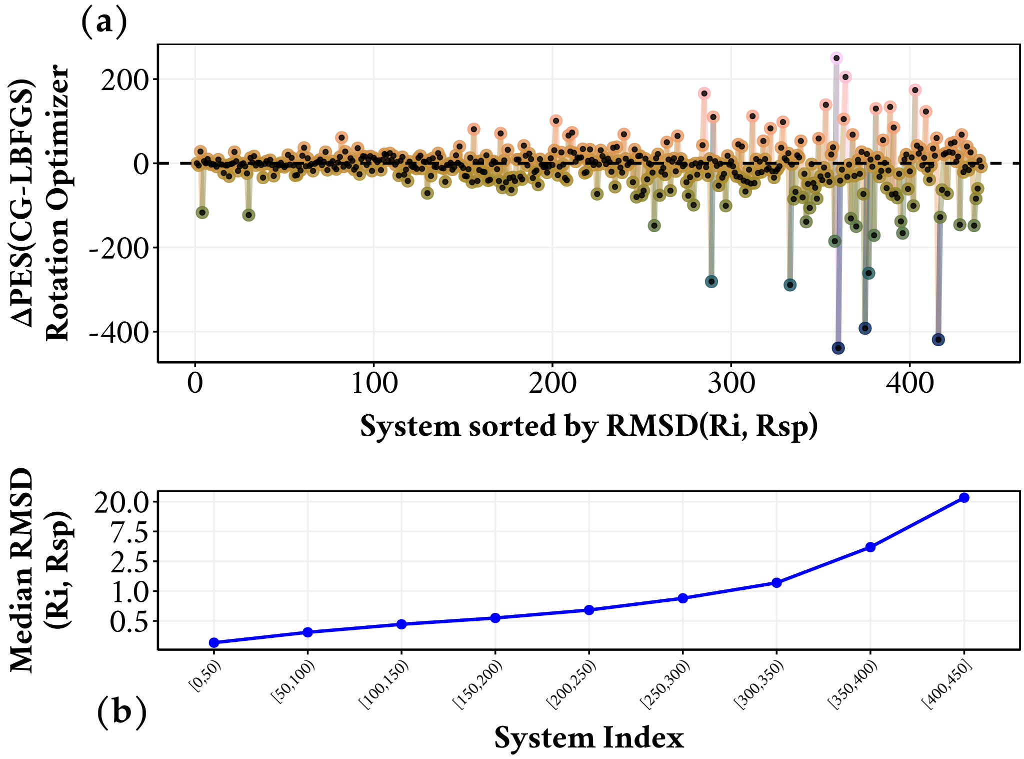


Figure 1. *(a) Difference in the number of potential energy surface (PES) calls between L-BFGS and conjugate gradient (CG) rotations for dimer calculations that converged to the same saddle, defined as having a final energy difference of less than 0.01 eV. The x-axis represents the system index, sorted by the median RMSD. The y-axis displays the difference in PES calls, for changing the optimizer for the rotation of the dimer when rotation removal is not used. Consequently, negative values indicate that the CG algorithm required fewer PES calls (i.e., CG performed better), while positive values indicate that L-BFGS required fewer PES calls (i.e., LBFGS performed better). The median improvement is 21 calls occuring in 55.23% systems for CG-rotations compared to a median improvement of 16 calls in 43.18% systems for LBFGS-rotations across 440 systems. Systems which are farther away from the saddle point show a larger improvement with CG-rotations. (b) For the systems shown in (a), their log(median RMSD) plotted against System Index, with Median RMSD displayed on the y-axis.*

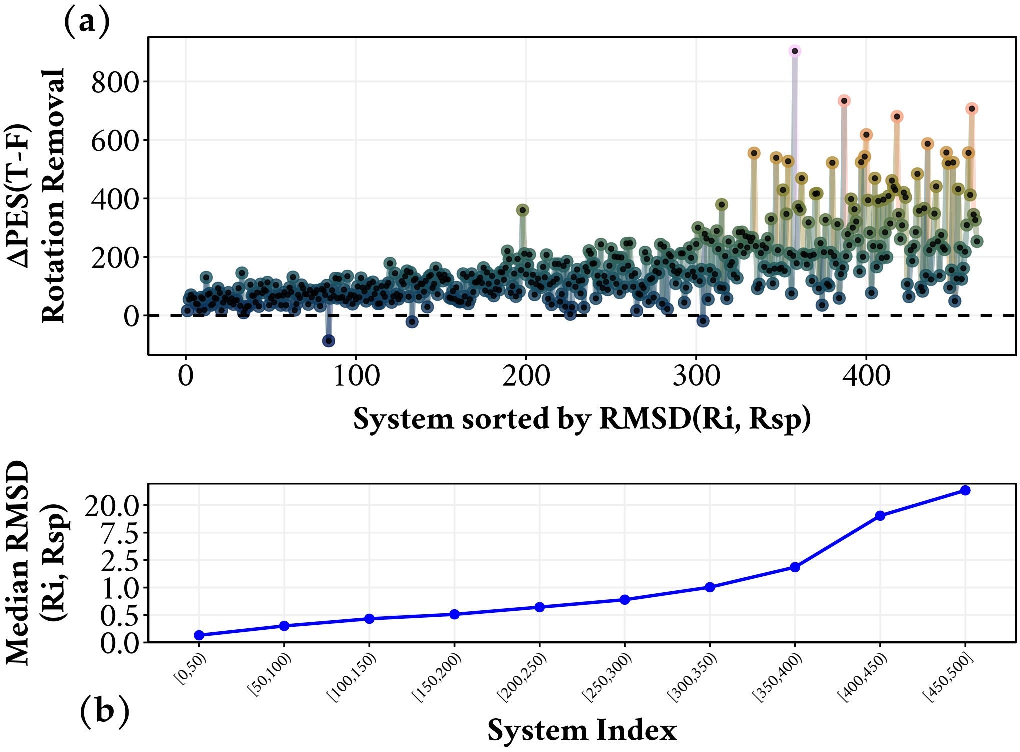


Figure 2. *(a) Difference in the number of potential energy surface (PES) calls between using rotation removal (T) and not using rotation removal (F) for dimer calculations that converged to the same saddle, defined as having a final energy difference of less than 0.01 eV. The x-axis represents the system index, sorted by the median RMSD. The y-axis displays the difference in PES calls, using CG-rotations with and without rotation removal. Consequently, negative values indicate that the rotation removal required fewer PES calls (i.e., it performed better), while positive values indicate turning off the rotation removal took fewer PES calls (i.e., no rotation removal performed better). The median improvement is 126 calls occuring in 99.35% systems for rotation removal not being used compared to a median improvement of 22 calls in 0.65% systems when rotation removal is used across 465 systems. Systems which are farther away from the saddle point show a larger improvement without rotation removal. (b) For the systems shown in (a), their log(median RMSD) plotted against System Index, with Median RMSD displayed on the y-axis.*

tems for which both methods in each comparison pair were run. High rates of joint success (where both compared methods converged) were generally observed across conditions. However, differences favouring the CG optimizer's robustness are apparent. When comparing optimizers directly, CG achieved unique success (succeeding when LBFGS failed) considerably more often than LBFGS did, both with rotation removal enabled (3.6% vs 0.6%, panel a) and disabled (5.8% vs 1.6%, panel b). Disabling rotation removal slightly reduced joint success rates and increased joint failures from 1.8% to 4.2% when comparing the optimizers. Examining the impact of rotation removal itself highlights CG's robustness further: using the CG optimizer (panel c), success rates remained very high (~96% joint success) with minimal sensitivity, as disabling rotation removal resulted in unique success in only 0.8% of cases. Conversely, using the LBFGS optimizer (panel d), disabling rotation removal significantly impacted performance; while joint success was still high (90.6%), unique successes for the enabled rotation removal setting were more frequent (4.0% vs 1.8% for disabled), and the joint failure rate rose notably to 3.6%. This suggests CG is less perturbed by the absence of rotation removal. As observed previously, convergence failures tend to concentrate at higher initial RMSD values, particularly in scenarios involving LBFGS without rotation removal.

This initial overview highlights apparent performance trends and differences between the method variants. However, these direct comparisons and visual inspections do not fully quantify the uncertainty associated with these effects or rigorously account for system-specific variability and potential interactions between the optimizer choice and rotation removal. Therefore, we employ the Bayesian hierarchical models detailed in Section 2.2 for a more robust statistical analysis of these metrics.

It is important to note that there are alternative methods for removing global rotations and translations, such as those implemented in OPTIM [38]. In OPTIM, the Rayleigh-Ritz procedure [10] and subsequent partial tangent space minimization involve orthogonalization to overall translation or rotation. However, the application demonstrated here follows the quaternion-based method implemented in EON [16].

3.2 Effort analysis model

To quantify the effects of the dimer rotation optimizer and rotation removal on computational effort, we analyzed the number of PES calls using the Bayesian negative binomial generalized linear mixed-effects model described in Section 2.2 (Eq. 7). We fitted models corresponding to the different fixed-effects structures defined in Eq. 6, examining main effects within relevant data subsets as well as the full interaction model using all comparable data.

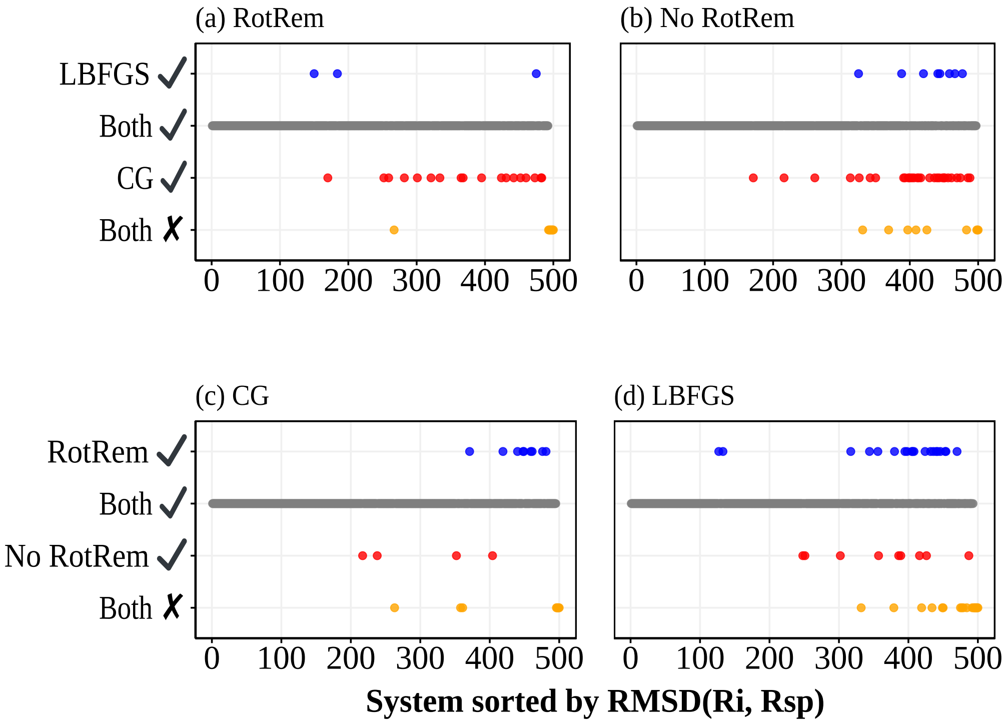


Figure 3. *Binary convergence outcomes for dimer method calculations across 500 systems, ordered by median root-mean-squared-displacement (RMSD) of the intial geometry relative to the saddle point geometry. Each panel directly compares two settings, with the vertical position and color indicating the outcome. Gray points indicate both compared settings succeed. Yellow points indicate where both compared settings fail. (a) Comparing LBFGS (blue) and CG (red) with rotation removal enabled, both methods (gray) succeeded in 94.0% of systems, while CG additionally succeeded (red) in 3.6% (LBFGS in 0.6%, blue) and both failed (yellow) in 1.8%. (b) Without rotation removal, comparing LBFGS (blue) to CG (red) yielded 90.8% joint success (gray), with 5.8% (red) unique CG successes (LBFGS in 1.6%, blue) while the joint failure is 4.2% (yellow). (c) Contrasting rotation removal (blue) with no rotation removal (red) using the CG-rotations shows high joint success (95.8%, gray) with 0.8% additional (red) without removal (1.8% with rotation removal, blue) and 1.6% cases where both fail (yellow). (d) For the LBFGS optimizer, comparing across rotation removal (blue) resulted in 90.6% joint success (gray), without rotation removal (red) 1.8% additional successes while with rotation removal (blue) 4% additional cases succeed, and there are 3.6% joint failures (yellow). Failures, particularly the unique failures with LBFGS without rotation removal, are concentrated at higher RMSD values.*

$$
\begin{aligned}
\text{PESCalls}_{ij} &\sim \text{NegativeBinomial}(\mu_{ij}, \phi) \\
\log(\mu_{ij}) &= \eta_{ij} \\
\eta_{ij} &= \begin{cases} \beta_0 + \beta_1 \text{DR}_{i(j))} + u_j & \text{(a)} \\ \beta_0 + \beta_2 \text{RR}_{i(j))} + u_j & \text{(b)} \\ \beta_0 + \beta_1 \text{DR}_{i(j))} & \\ +\beta_2 \text{RR}_{i(j))} + \beta_3 (\text{DR}_{i(j))} \times \text{RR}_{i(j))}) + u_j & \text{(c)} \end{cases} \\
u_j &\sim \text{Normal}(0, \sigma_u^2)
\end{aligned}
\tag{10}
$$

First, applying the RotOptimizer model structure (Eq. (10)a), specifically to calculations where rotation removal was disabled, we examin the effect of the rotation optimizer (β_1). Switching from the CG reference to the L-BFGS optimizer resulted in a small but statistically credible increase in the median number of PES calls by 2.6% (95% CrI: [1.1%, 4.1%]). This suggests a slight efficiency advantage, in terms of PES calls, for the CG optimizer under these conditions.

Next, using the RotRemoval model structure (Eq. (10)b) applied to calculations using the CG optimizer, we examine the effect of enabling rotation removal (β_2). Enabling rotation removal led to a substantial and credible increase in the median PES calls by 41.5% (95% CrI: [38.6%, 44.4%]) compared to disabling it. This indicates a significant computational cost penalty associated with enabling rotation removal when using the CG optimizer in this setup.

Finally, the full interaction model (Eq. (10)c), incorporating both factors and their interaction simultaneously across 1805 valid observations, corroborated the main effects observed in the subset models (posterior distributions shown in Figure 4). The estimated median increase in PES calls for L-BFGS relative to CG (when rotation removal is off, β_1) was 2.6% (95% CrI: [0.7%, 4.5%]), corresponding to a multiplicative factor credibly just above 1 (median 1.03). Similarly, the median increase for enabling rotation removal relative to disabling it (when using CG, β_2) was substantial at 44.2% (95% CrI: [41.6%, 46.8%]), clearly visualized by the distribution centered around a multiplicative factor of 1.44 in Figure 4. The 95% credible interval for the interaction term (β_3), visually overlapping 1 in Figure 4 (multiplicative factor median 1.01, 95% CrI [0.99, 1.04]), provides no strong evidence that the effect of the optimizer choice significantly depends on the rotation removal setting, or vice versa. Furthermore, this full model estimated the standard deviation of the random intercepts for each system to be substantial (median σ_u = 0.63, 95% CrI: [0.59, 0.67] on the log scale), indicating significant system-specific variation in the baseline number of PES calls required for convergence, independent of the method variations.

These findings from the hierarchical models, which account for system variability via random intercepts ($\sigma_u \approx$ 0.63), reinforce the trends observed in the exploratory data analysis (Fig. 1 and Fig. 3). They quantify the consistent, albeit small, PES call advantage of CG over L-BFGS (β_1) and the considerable computational overhead incurred by enabling this particular rotation removal implementation (β_2) for this dataset. The model diagnostics, including posterior predictive checks and residual analyses (shown for individual models in the preceding code sections, further details in SI), indicate reasonable model fit for the purpose of comparing these relative effects, despite some remaining heteroscedasticity common in complex chemical datasets.

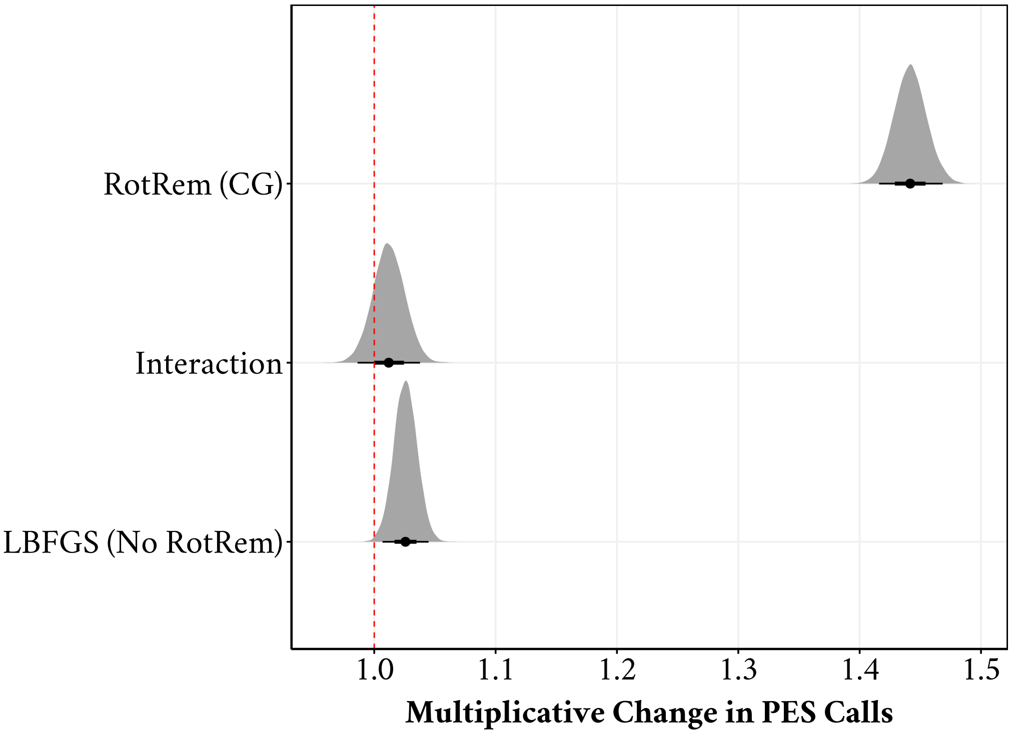


Figure 4. *Posterior distributions of fixed effects on the multiplicative scale for PES calls from the full interaction model. The x-axis represents the factor by which expected PES calls change relative to the baseline (CG optimizer, rotation removal disabled), with the dashed line at x=1 indicating no effect. Distributions show the main effect of using L-BFGS (vs CG, when rotation removal is disabled), the main effect of enabling rotation removal (vs disabling, when using the CG optimizer), and their interaction. Thick lines represent 95% credible intervals with median dots. Enabling rotation removal shows a large multiplicative increase (~1.44), L-BFGS shows a small increase (~1.03), and the interaction credibly includes 1.*

3.3 Success rate model

We next analyzed the factors influencing the probability of successful convergence using the Bayesian Bernoulli generalized linear mixed-effects model specified in Section 2.2 (Eq. 11). Effects are reported as Odds Ratios (OR), representing the multiplicative change in the odds of success compared to a reference condition. An OR > 1 indicates increased odds, OR < 1 indicates decreased odds, and a 95% Credible Interval (CrI) that excludes 1 suggests a statistically credible effect.

$$
\begin{aligned}
&\text{Success}_{ij} \sim \text{Bernoulli}(p_{ij}) \\
&\text{logit}(p_{ij}) = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \eta_{ij} \\
&\eta_{ij} = \begin{cases} \beta_0 + \beta_1 \text{DR}_i + u_j & \text{(a)} \\ \beta_0 + \beta_2 \text{RR}_i + u_j & \text{(b)} \\ \beta_0 + \beta_1 \text{DR}_i + \beta_2 \text{RR}_i + \beta_3 (\text{DR}_i \times \text{RR}_i) + u_j & \text{(c)} \end{cases} \\
&u_j \sim \text{Normal}(0, \sigma_u^2)
\end{aligned}
\tag{11}
$$

First, applying the RotOptimizer model structure (Eq. (11)a) specifically to the subset of calculations where rotation removal was disabled, we examined the effect of the optimizer (β_1). L-BFGS showed significantly lower odds of success compared to the CG reference. The estimated median OR for L-BFGS relative to CG was 0.3 (95% CrI: [0.13, 0.60]), indicating that the odds of convergence with L-BFGS were roughly only 30% of the odds with CG under these conditions. This model also revealed substantial variability between systems in their baseline odds of success (posterior median s_u = 2.6 on the log-odds scale, 95% CrI: [1.67, 3.81]).

Next, using the RotRemoval model structure (Eq. (11)b) specifically to the subset of calculations using CG-rotations, we examined the effect of rotation removal (β_2). The median OR for enabling rotation removal versus disabling it was 2.0 (95% CrI: [0.73, 6.35]). Although the point estimate suggests doubled odds of success when enabling rotation removal, the wide credible interval comfortably includes 1. Consequently, this model provides no strong statistical evidence that enabling rotation removal significantly alters the odds of success compared to disabling it when using the CG optimizer. Substantial between-system variability in baseline success odds was again noted (median sd(Intercept) = 4.2, 95% CrI: [2.63, 6.91] from this model's summary; the full model estimate is used below for consistency).

Finally, the full interaction model (Eq. (11)c), analyzing all 2000 observations across the four conditions, allowed for a detailed assessment of success odds. While the predicted probability of success for an average system is very high (>0.98) under all conditions, as visualized in the main panel of Figure 5, the model revealed statistically credible differences when analyzing the underlying odds ratios as seen in the inset of Figure 5. It reinforced the substantially lower odds of success for L-BFGS compared to CG (median OR = 0.2, 95% CrI: [0.09, 0.45]); this relative difference, indicating higher robustness for CG, is subtly reflected in Figure 5 by the slight shift of the L-BFGS probability distributions (right column) towards lower values compared to CG (left column). The model also showed no credible main effect for rotation removal (median OR = 1.9, 95% CrI: [0.74, 5.07]) nor a credible interaction (median Interaction OR = 1.1, 95% CrI: [0.33, 3.42]), consistent with the high degree of similarity between the top ('no') and bottom ('yes') panels within each optimizer in Figure 5. Thus, while both optimizers perform well on average, the primary factor credibly influencing the relative odds of success in this analysis is the choice of optimizer, with CG demonstrating significantly higher odds of convergence than L-BFGS across both rotation removal settings in this dataset. The large estimated standard deviation for the system-specific random intercepts from the full model (posterior median σ_u = 3.6, 95% CrI: [2.79, 4.80]) underscores that system-specific properties remain a major determinant of convergence success or failure for individual cases, explaining deviations from the high average success probability. Model diagnostics indicated good convergence and fit (details in SI).

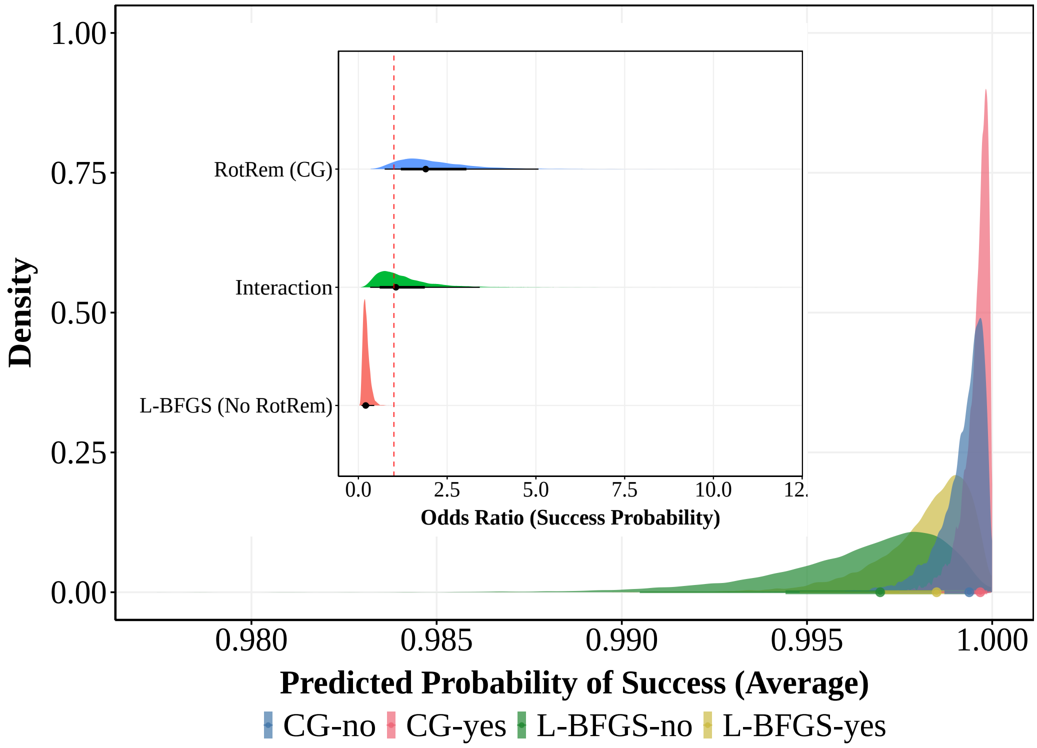


Figure 5. *Posterior distributions for convergence success derived from the full interaction model (brms$_{pfidall}$). The main panel displays the predicted probability of success for an average system under the four method conditions (Optimizer-Rotation Removal, see legend), illustrating high probabilities (>0.98) for all settings. The inset shows the posterior distributions for the fixed effects on the Odds Ratio (OR) scale relative to the baseline (CG, rotation removal disabled), with the dashed line at OR=1 indicating no change in odds. While absolute probabilities are high, the inset quantifies the relative effects, showing the OR for L-BFGS vs CG (bottom distribution) is credibly below 1, while the ORs for rotation removal (top) and the interaction (middle) credibly include 1. The distributions for CG are slightly shifted towards higher probabilities compared to L-BFGS. Distributions for enabling ('yes') versus disabling ('no') rotation removal are nearly identical within each optimizer column, but do tend to cause a rightward shift. Thick lines in the inset represent 95% credible intervals with median dots.*

The preceding analyses provide a quantitative, statistically robust assessment of how the choice of dimer rotation optimizer and the application of rotation removal influence both the computational effort (PES calls) and the likelihood of success for saddle point searches within this benchmark set. Having established these effects and accounted for system-specific variability using Bayesian hierarchical models, we now turn to discussing the implications of these findings in the context of theoretical expectations, practical algorithm choice, and the broader methodology of benchmarking computational chemistry methods.

4 Discussion

Our analysis, employing Bayesian hierarchical models to account for system-specific variability, provides quantitative insights into the performance of CG and L-BFGS optimizers for dimer rotation, and the effect of enabling rotation removal, across a large benchmark set. The key findings indicate a small but credible PES call advantage and significantly higher convergence odds for the CG optimizer compared to L-BFGS. Conversely, enabling rotation removal incurred a substantial PES call penalty without providing a statistically credible improvement in success probability in the full model applied to this dataset.

The superior performance of CG over L-BFGS, particularly its enhanced robustness demonstrated by the success probability analysis (OR for L-BFGS vs CG $\approx$ 0.2 [0.09, 0.45]), aligns with potential theoretical advantages of CG in navigating complex potential energy surfaces. While modern L-BFGS methods incorporate strategies to ensure positive-definite Hessian approximations [39], they may still be susceptible to challenges in regions with complex curvature compared to methods like CG that directly seek movement along specific eigenmodes [20]. The observed difference in computational effort per successful run was modest (L-BFGS requiring ~2.6% [0.7%, 4.5%] more PES calls than CG when rotation removal was off), suggesting that the primary practical advantage of CG in this context lies in its higher probability of reaching convergence.

Counter-intuitively, enabling rotation removal, theoretically designed to simplify the optimization by removing global rotations of the coordinate vector [16], resulted in a significant increase in computational cost (median +44% PES calls vs disabled) without a corresponding statistically credible improvement in success probability in our full model (OR for RotRem Yes vs No $\approx$ 1.9 [0.74, 5.07]). This discrepancy likely highlights the sensitivity of optimization performance to implementation details and the specific nature of the PES. The projection involved in rotation removal interacts with the optimizer's steps and Hessian updates; for this dataset containing potentially fragmenting or aggregating systems near saddle points, where the separation of internal and external coordinates can be ill-defined [40], this projection may hinder convergence pathways that are accessible without it. Thus we observe less efficient trajectories when rotation removal is active for the majority of these systems.

We underscore the value of applying rigorous statistical modeling to computational benchmarks, moving beyond simple averages or qualitative comparisons. Traditional approaches can be easily confounded by the large system-to-system variability inherent in chemical datasets, as evidenced by the substantial standard deviation estimated for the random intercepts in our models i.e. $\sigma_u \approx 0.63$ for PES calls (on the logarithmic scale), \$$\sigma_{\mathrm{u}}$≈\$3.6 for success odds(on the logit scale). Bayesian hierarchical models explicitly account for this variability through random effects (u_j). Furthermore, they leverage partial pooling (shrinkage), effectively borrowing strength across systems to produce more stable and reliable estimates for the effects of different methods, mitigating the influence of outlier systems - an effect sometimes colloquially related to "regression towards the mean". This framework also provides direct quantification of uncertainty via full posterior distributions and credible intervals (e.g., Figs. 4, 5), facilitating more nuanced conclusions than point estimates or p-values alone [33, 35]. The ability to model interactions formally is another advantage for dissecting performance factors.

Further theoretical considerations help to contextualize these empirical results. The rotational phase of the Dimer method represents a specialized optimization problem: isolating a single minimum-curvature mode. The Conjugate Gradient algorithm is inherently well-suited for such eigenvector-following tasks. It can also be less susceptible to being trapped by higher-energy eigenpairs during the search [20]. Conversely, while L-BFGS is a highly effective general-purpose optimizer, it does not, in most cases, share the same quadratic convergence guarantees as the full BFGS method [41]. This theoretical background aligns well with our findings, where the most significant performance gap between the two methods appeared in their relative robustness and success rates, rather than in the computational cost of already successful runs.

Given the results, particularly the higher robustness of CG and the general inefficiency introduced by enabling rotation removal in this context, a practical strategy for high-throughput workflows might involve defaulting to the CG optimizer without rotation removal. A fallback mechanism could then activate rotation removal only for specific systems that initially fail to converge, potentially capturing benefits for problematic cases where external modes interfere, although identifying such cases a priori remains challenging. This “chain of methods” approach warrants further study but illustrates how statistically grounded benchmarks can inform practical workflow design beyond simply selecting a single “best” method.

While this study has focused on variants of the Dimer method, the Bayesian hierarchical framework presented here is broadly applicable. An important future direction is its application to a comparative benchmark of fundamentally different classes of algorithms, such as single-ended methods (Dimer, Sella) versus chain-of-states methods (NEB, GSM). Such an analysis could quantitatively address long-standing questions, for example, by modeling the probability of finding the correct path as a function of the number of images in an NEB calculation or comparing the raw computational cost versus the likelihood of success for finding a transition state starting from only the reactant and product states. This would provide the community with a robust, data-driven basis for selecting the optimal algorithm class for a given high-throughput workflow.

5 Conclusions

We introduced and applied a Bayesian hierarchical modeling framework to rigorously analyze performance metrics from computational chemistry benchmarks, demonstrating its utility for comparing dimer method variants across a large, diverse dataset. This statistically robust approach moves beyond simple averages or visualizations, accounting for system-specific variability and providing nuanced uncertainty quantification, offering a more actionable view for benchmarking algorithms.

Our analysis of the Dimer method variants revealed that for this large benchmark set, the Conjugate Gradient (CG) optimizer for dimer rotation provided a clear advantage in convergence robustness over L-BFGS (Section 2.2). The decision to enable rotation removal is more complex. While theoretically motivated to simplify the potential energy surface, our findings indicate it led to higher computational costs for the majority of systems studied. However, our models also suggest that rotation removal may offer a reliability benefit when paired with the L-BFGS optimizer, highlighting a subtle interplay between algorithmic components that warrants further study.

We assert that these results, along with the automated capabilities of most workflow engines suggest a "chain" of methods approach rather than an absolute best. Although given the slight increase in performance and the significant increase in robustness it is hard to argue for the LBFGS rotations, the situation for the chemically intuitive external degree of freedom removal is more nuanced.

From the generalized linear mixed models (Section 2.2), using the LBFGS for rotations showed a small but statistically discernible increase of 3.4% (95% CrI: [1.1%, 5.8%]) in PES calls. Furthermore, the Bernoulli model suggests a substantial improvement in success probability for Dimer CG, with an estimated 23% higher success rate (95% CI: 5%, 40%), supporting the conclusion that CG-rotations are more robust.

There are however, situations where the rotation removal is required for convergence. Since the data demonstrates that for these random saddle point initializations that the rotation removal for such fragmenting systems is not optimal in most cases, it is more natural to first run CG-rotation without external rotation removal, and have a fallback rule to run CG-rotation with external rotation only for systems which otherwise fail. This strategy leverages the higher success rate of CG in the majority of cases while providing a fallback option for more challenging systems.

Our benchmark results and analysis are packaged in a user-friendly repository, showing integration with Snakemake for high throughput calculations and EON for the energy and forces calculation through NWChem along with R code for the figures and running the Bayesian analysis. It is also, to the best of our knowledge, the largest benchmark study done on the modalities of the dimer method. Our analysis provides a more nuanced and statistically robust assessment of performance than traditional approaches, which often rely on simple averages or visualizations that can obscure important details. We expect this reproducible resource will be indispensible to others in the community performing high throughput chemistry or algorithm development.

6 Supplementary Material

Model validations for the Bayesian models are fairly standard and verbose, so they are present in the accompanying supplementary material.

Acknowledgments

This work was supported by the Icelandic Research Fund (grant no. 217436-053).

RG thanks Dr. Miha Gunde, Dr. Amrita Goswami, Dr. Moritz Sallermann, Prof. Debabrata Goswami, Prof. Laurent Karim Béland, Prof. Morris Riedel, Mrs. Sonaly Goswami and Mrs. Ruhila Goswami for discussions. RG also thanks Prof. Normand Mousseau and Prof. Birgir Hrafnkelsson for manuscript suggestions and edits.

Data Availability Statement

The full computational workflow, analysis code, and data supporting this study are openly available. A version-tagged GitHub repository (`https://github.com/HaoZeke/brms_idrot_repro`) provides all resources needed to reproduce the findings. To ensure full reproducibility, the repository contains a complete computational environment managed by pixi, with version-pinned software dependencies including EON, NWChem, and ASE, alongside utilities from Wailord [42]. In line with best practices for literate and reproducible programming [43, 44, 45, 46], some dependencies are bundled directly via `gitsubrepo`. For permanent access and citation, the final curated dataset is hosted on the Materials Cloud Archive [47] under the entry Goswami [48].

Conflict of interest

The authors declare no conflict of interest.

References for Paper III

[1] Henry Eyring. "The Activated Complex in Chemical Reactions." In: *Journal of Chemical Physics* 3.2 (Feb. 1935), pp. 107–115. DOI: `10.1063/1.1749604`.

[2] Philip Pechukas. "Recent Developments in Transition State Theory." In: *Berichte der Bunsen-Gesellschaft für Physikalische Chemie* 86.5 (1982), pp. 372–378. DOI: `10.1002/bbpc.19820860509`.

[3] 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`.

[4] 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`.

[5] E. Cancès, F. Legoll, M.-C. Marinica, K. Minoukadeh, and F. Willaime. "Some Improvements of the Activation-Relaxation Technique Method for Finding Transition Pathways on Potential Energy Surfaces." In: *Journal of Chemical Physics* 130.11 (Mar. 2009), p. 114711. DOI: `10.1063/1.3088532`.

[6] Normand Mousseau, Laurent Karim Béland, Peter Brommer, Jean-François Joly, Fedwa El-Mellouhi, Eduardo Machado-Charry, Mihai-Cosmin Marinica, and Pascal Pochet. "The Activation-Relaxation Technique: ART Nouveau and Kinetic ART." In: *Journal of Atomic and Molecular Physics* 2012.1 (2012), p. 925278. DOI: `10.1155/2012/925278`.

[7] 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`.

[8] M. Gunde, A. Jay, M. Poberžnik, N. Salles, N. Richard, G. Landa, N. Mousseau, L. Martin-Samos, and A. Hemeryck. "Exploring Potential Energy Surfaces to Reach Saddle Points above Convex Regions." In: *The Journal of Chemical Physics* 160.23 (June 2024), p. 232501. DOI: 10.1063/5.0210097.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Samuel T. Chill, Jacob Stevenson, Victor Ruehle, Cheng Shang, Penghao Xiao, James D. Farrell, David J. Wales, and Graeme Henkelman. "Benchmarks for Characterization of Minima, Transition States, and Pathways in Atomic, Molecular, and Condensed Matter Systems." In: *Journal of Chemical Theory and Computation* 10.12 (Dec. 2014), pp. 5476–5482. DOI: 10.1021/ct5008718.

[16] Marko Melander, Kari Laasonen, and Hannes Jónsson. "Removing External Degrees of Freedom from Transition-State Search Methods Using Quaternions." In: *Journal of Chemical Theory and Computation* 11.3 (Mar. 2015), pp. 1055–1062. DOI: 10.1021/ct501155k.

[17] Yi Zeng, Penghao Xiao, and Graeme Henkelman. "Unification of Algorithms for Minimum Mode Optimization." In: *The Journal of Chemical Physics* 140.4 (Jan. 2014), p. 044115. DOI: 10.1063/1.4862410.

[18] 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.

[19] 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.

[20] 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.

[21] Sebastiaan P. Huber. “Automated Reproducible Workflows and Data Provenance with AiiDA.” In: *Nature Reviews Physics* 4.7 (July 2022), pp. 431–431. DOI: 10.1038/s42254-022-00463-1.

[22] 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.

[23] Hans-Peter Kriegel, Erich Schubert, and Arthur Zimek. “The (Black) Art of Runtime Evaluation: Are We Comparing Algorithms or Implementations?” In: *Knowledge and Information Systems* 52.2 (Aug. 2017), pp. 341–378. DOI: 10.1007/s10115-016-1004-2.

[24] Philip J. Fleming and John J. Wallace. “How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results.” In: *Communications of the ACM* 29.3 (Mar. 1986), pp. 218–221. DOI: 10.1145/5666.5673.

[25] 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.

[26] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. “Stan: A Probabilistic Programming Language.” In: *Journal of Statistical Software* 76.1 (2017). DOI: 10.18637/jss.v076.i01.

[27] 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.

[28] 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.

[29] Andreas Heyden, Alexis T. Bell, and Frerich J. Keil. “Efficient Methods for Finding Transition States in Chemical Reactions: Comparison of Improved Dimer Method and Partitioned Rational Function Optimization Method.” In: *The Journal of Chemical Physics* 123.22 (Dec. 2005), p. 224101. DOI: 10.1063/1.2104507.

[30] Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. *Forces Are Not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations.* Oct. 2022. arXiv: `2210.07237 [physics]`.

[31] Wojciech G. Stark, Cas van der Oord, Ilyes Batatia, Yaolong Zhang, Bin Jiang, Gábor Csányi, and Reinhard J. Maurer. *Benchmarking of Machine Learning Interatomic Potentials for Reactive Hydrogen Dynamics at Metal Surfaces.* Mar. 2024. arXiv: `2403.15334 [physics]`.

[32] Robert B. O'Hara and D. Johan Kotze. "Do Not Log-Transform Count Data." In: *Methods in Ecology and Evolution* 1.2 (2010), pp. 118–122. doi: `10.1111/j.2041-210X.2010.00021.x`.

[33] 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.

[34] Matthew D. Homan and Andrew Gelman. "The No-U-turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo." In: *The Journal of Machine Learning Research* 15.1 (Jan. 2014), pp. 1593–1623.

[35] Blakeley B. McShane, David Gal, Andrew Gelman, Christian Robert, and Jennifer L. Tackett. "Abandon Statistical Significance." In: *The American Statistician* 73.sup1 (Mar. 2019), pp. 235–245. doi: `10.1080/00031305.2018.1527253`.

[36] Todd A. Kuffner and Stephen G. Walker. "Why Are P-Values Controversial?" In: *The American Statistician* 73.1 (Jan. 2019), pp. 1–3. doi: `10.1080/00031305.2016.1277161`.

[37] Frank Jensen. *Introduction to Computational Chemistry.* Third edition. Chichester, UK ; Hoboken, NJ: Wiley, 2017.

[38] 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`.

[39] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* 2nd ed. Springer Series in Operations Research. New York: Springer, 2006.

[40] Jeff Nichols, Hugh Taylor, Peter Schmidt, and Jack Simons. "Walking on Potential Energy Surfaces." In: *Journal of Chemical Physics* 92.1 (Jan. 1990), pp. 340–346. doi: `10.1063/1.458435`.

[41] Albert S. Berahas, Frank E. Curtis, and Baoyu Zhou. "Limited-Memory BFGS with Displacement Aggregation." In: *Mathematical Programming* 194.1 (July 2022), pp. 121–157. doi: `10.1007/s10107-021-01621-6`.

[42] Rohit Goswami. "Wailord: Parsers and Reproducibility for Quantum Chemistry." In: *Proceedings of the 21st Python in Science Conference* (2022), pp. 193–197. doi: `10.25080/majora-212e5952-021`.

[43] Rohit Goswami and Ruhila S. "High Throughput Reproducible Literate Phylogenetic Analysis." In: *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC).* Nov. 2022, pp. 337–340. DOI: 10.1109/PDGC56933.2022.10053210.

[44] Rohit Goswami, Ruhila S., Amrita Goswami, Sonaly Goswami, and Debabrata Goswami. "Reproducible High Performance Computing without Redundancy with Nix." In: *2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC).* Nov. 2022, pp. 238–242. DOI: 10.1109/PDGC56933.2022.10053342.

[45] Moritz Sallermann, Amrita Goswami, Alejandro Peña-Torres, and Rohit Goswami. *Flowy: High Performance Probabilistic Lava Emplacement Prediction.* June 2024. DOI: 10.48550/arXiv.2405.20144. arXiv: 2405.20144 [physics].

[46] Rohit Goswami, Maxim Masterov, Satish Kamath, Alejandro Peña-Torres, and Hannes Jónsson. *Efficient Implementation of Gaussian Process Regression Accelerated Saddle Point Searches with Application to Molecular Reactions.* May 2025. DOI: 10.48550/arXiv.2505.12519. arXiv: 2505.12519 [physics].

[47] Leopold Talirz, Snehal Kumbhar, Elsa Passaro, Aliaksandr V. Yakutovich, Valeria Granata, Fernando Gargiulo, Marco Borelli, Martin Uhrin, Sebastiaan P. Huber, Spyros Zoupanos, Carl S. Adorf, Casper Welzel Andersen, Ole Schütt, Carlo A. Pignedoli, Daniele Passerone, Joost VandeVondele, Thomas C. Schulthess, Berend Smit, Giovanni Pizzi, and Nicola Marzari. "Materials Cloud, a Platform for Open Computational Science." In: *Scientific Data* 7.1 (Sept. 2020), p. 299. DOI: 10.1038/s41597-020-00637-5.

[48] Rohit Goswami. *Dataset for Bayesian Hierarchical Models for Quantitative Estimates for Performance Metrics Applied to Saddle Search Algorithms.* May 2025. DOI: 10.24435/materialscloud:xc-5e.

Paper IV

Adaptive Pruning for Increased Robustness and Reduced Computational Overhead in Gaussian Process Accelerated Saddle Point Searches

Rohit Goswami, Hannes Jónsson

ChemPhysChem, accepted

Rohit developed the OTGPD algorithm, ran simulations, made figures, and wrote the article.

Abstract

Gaussian process (GP) regression provides a strategy for accelerating saddle point searches on high-dimensional energy surfaces by reducing the number of times the energy and its derivatives with respect to atomic coordinates need to be evaluated. The computational overhead in the hyperparameter optimization can, however, be large and make the approach inefficient. Failures can also occur if the search ventures too far into regions that are not represented well enough by the GP model. Here, these challenges are resolved by using geometry-aware optimal transport measures and an active pruning strategy using a summation over Wasserstein-1 distances for each atom-type in farthest-point sampling, selecting a fixed-size subset of geometrically diverse configurations to avoid rapidly increasing cost of GP updates as more observations are made. Stability is enhanced by permutation-invariant metric that provides a reliable trust radius for early-stopping and a logarithmic barrier penalty for the growth of the signal variance. These physically motivated algorithmic changes prove their efficacy by reducing to less than a half the mean computational time on a set of 238 challenging configurations from a previously published data set of chemical reactions. With these improvements, the GP approach is established as a robust and scalable algorithm for accelerating saddle point searches when the evaluation of the energy and atomic forces requires significant computational effort.

Keywords Saddle Point Search, Gaussian Process Regression, Dimer Method, Optimal Transport, Data Pruning, Computational Chemistry

1 Introduction

An understanding of the transition mechanisms and estimation of the rate of atomic rearrangements, such as chemical reactions and diffusion events, requires analyzing the high-dimensional potential energy surface (PES) and identifying transition states. The PES specifies how the energy of the system varies as a function of the atomic coordinates and the negative gradient gives the force acting on the atoms. Reactant and product states correspond to local minima on the PES. Within the harmonic approximation of transition state theory, first order saddle points on the PES, i.e. stationary points where the Hessian matrix has one negative eigenvalue, characterize the transition states.[1] Similar considerations apply to magnetic transitions where magnetic vectors rotate.[2]

Several methods have been developed for finding first order saddle points on energy surfaces given only a starting configuration of the atoms [3, 4, 5, 6, 7, 8, 9, 10] without a predefined final state of the transition. These are distinct from the “double ended” methods like the Nudged elastic band [11, 1, 12, 13] and variants thereof [14] which require known initial and final states. We focus on general-purpose methods which do not require user-defined constraints on the transition state optimization [15] and compare to the extant state of the art within such methods [16]. A given initial state, without reference to a known product state, can in general lead to many different saddle points. By making a catalog of the lower energy saddle points, which correspond to the faster transitions, long time scale simulations of the system can be carried out for a given temperature. [17, 18] But, the computational effort involved when the PES is obtained from an electronic structure calculation makes this intractable. General purpose machine learned interatomic potentials, such as PET-MAD, [19, 20] can alleviate some of this cost for systems where a large set of electronic structure calculations has been performed, but these often suffer from under-sampling of the regions near the transition states as they are of low probability and are harder to sample.[21].

To improve efficiency, a local surrogate model of the PES can be created on-the-fly, most successfully using Gaussian Processes (GPs) [22, 16, 23], or neural networks [24]. This, however, introduces a second, often overlooked cost: the scaling of the computational effort in generating and updating the surrogate model. For GPs, the hyperparameter optimization involves repeatedly inverting a covariance matrix whose size grows with both the number of collected data points, M_{data}, and the number of atoms, N_{atoms}. [25, 26, 27] This steep computational scaling, especially when the gradient of the energy, i.e. atomic forces, are included in the training, can cause the wall-time cost of the surrogate model to eventually exceed that of the electronic structure calculations, creating a new performance bottleneck. It is, therefore, important to develop ways to generate accurate enough surrogate models with lower computational overhead.

Here, a framework for generating a surrogate GP energy surface is introduced where the scaling of the computational overhead remains nearly constant as additional data is acquired. Several features are, furthermore, introduced to make the construction and the use of the surrogate model more robust. We refer to the method as Optimal Transport Gaussian Process (OT-GP). Only a representative subset of the data is used to fit the hyperparamenters in the GP model, selected by a Farthest Point Sampling (FPS) strategy with the Earth Mover's Distance (EMD) measure. Several stability enhancements are also introduced, including a data-driven adaptive early stopping threshold to balance safety in exploration, a hyperparameter oscillation detection heuristic to ensure stable learning, an adaptive variance barrier to prevent unphysical jumps in the predicted PES, and a rigorous removal of overal rotation of the system. This geometry-aware subsampling and stabilization unlocks true wall-time efficiency for the GP accelerated saddle searches.

In Section 2, the methodology is described, including a review of the previous state-of-the-art in 2.1, a presentation of the OT-GP algorithm in 2.2 with computational specifics for the systems studied given in 2.3. Section 3 presents results of saddle point searches for molecular systems in terms of reliability, quality of saddles found, and performance metrics, using illustrative examples to validate the algorithmic choices and a large-scale benchmark test. Section 4 provides discussions on the physical and data-driven insights, analyzing the algorithm and the interpretation of the hyperparameters. Conclusions are given in Section 5 as well as indications of future directions, highlighting the method's robustness and long-term software maintenance.

2 Computational Methods

2.1 Dimer Method and GP Acceleration

The dimer method and its variants have been described in several articles [6, 28, 29, 30, 31, 32, 33, 16]. Here, we recap the standard formulation as implemented in the EON software [34] [28]. The dimer method is a minimum mode following approach where the eigenvector corresponding to the minimum eigenvalue of the Hessian, the matrix of second derivatives of $V(\mathbb{R})$ with respect to atom coordinates, is found by rotating two copies of the system, the dimer, so as to find the minimum in total energy for a given location of its center. The component of the gradient of the energy $V(\mathbb{R})$ in the direction of the minimum mode is then inverted and the dimer displaced opposite to the direction of the modified gradient. This leads in an iterative way to convergence on a first order saddle point. Let $x \in \mathbb{R}^{3n}$ denote the position vector in the extended configuration space of all the movable atoms $\mathbb{R}^{n\times 3} \to \mathbb{R}^{3n}$. The iterative path towards the saddle point involves moving uphill in energy along the minimum mode while going downhill in all orthogonal directions

$$\dot{x} = \mathbf{F}^{\perp} - \mathbf{F}^{\parallel} \tag{1}$$

$$\mathbf{F}^{\parallel} = -\hat{\tau}^{T} \nabla V(x) \hat{\tau} \tag{2}$$

$$\mathbf{F}^{\perp} = -\nabla V(x) - \mathbf{F}^{\parallel} \tag{3}$$

where $\hat{\tau}$ represents the unit eigenvector corresponding to the lowest eigenvalue of the Hessian matrix. The vector $\hat{\tau}$ is referred to as the minimum mode, or the softest mode of the Hessian.

The direction $\hat{\tau}$ is found iteratively using a finite difference between two replicas of the system, $\mathbf{R}_{1,2} = \mathbf{R}_0 \pm \delta R\hat{\tau}$, around a known midpoint configuration $\mathbf{R}_0$ starting with some orientation vector $N \approx \hat{\tau}$. After the optimal orientation has been found, the dimer is moved in the direction opposite to the transformed gradient. The saddle point search algorithm alternates between estimating the minimum mode by rotating the dimer and translating the center of the dimer until the atomic forces at the midpoint tend to zero within user defined tolerance [6, 33].

Both the rotation and the translation require the use of some optimization algorithm. For rotation, often the conjugate gradient with Polak-Ribiere [33] is used, and the limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm, L-BFGS for the translation [30, 35]. The convergence characteristics are determined most strongly by the choice of the maximum translation step and the initial direction of the dimer.

The Gaussian Process accelerated dimer, referred to here as GPDimer, involves constructing successive surrogate models of the PES on the fly.[22, 16] The GP approximation learns a function based on the previously computed points, typically electronic structure calculations for given positions of the atoms, and a set of optimized hyperparameters which maximize the likelihood of seeing the previous data:

$$V_{\text{GP}}(\mathbf{x}_{\text{new}} | M_{data}, \theta_{\text{opt}}) \approx V(\mathbf{x}_{\text{new}}) \tag{4}$$

where M_{data} form the observed pairs of $(V(x), -\nabla V(x))$ for x and $\boxtimes_{\text{opt}}$ are the hyperparameters of the kernel covariance function. Fig. 1 provides an overview of the algorithm.

The similarity between two configurations, $\mathbf{x}$ and $\mathbf{x}'$, is specified by the inverse-distance

[28] Documentation available at `https://eondocs.org`

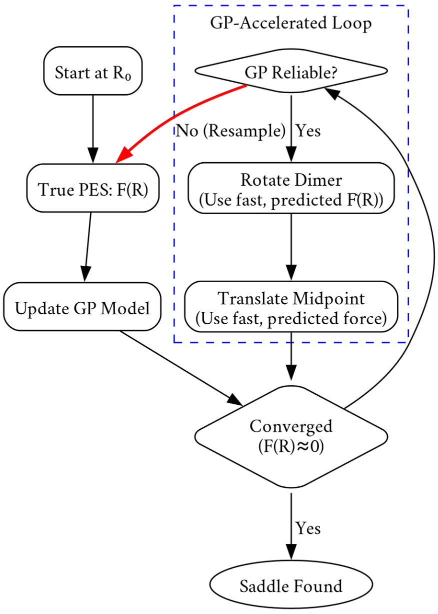


Figure 1. *The previously developed GPDimer method for accelerating saddle point searches using Gaussian process regression. The GP generated surrogate surface is considered reliable if the guardrails of Eq. 8 and 7 are satisfied.*

based squared exponential kernel [22, 16]:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_c^2 + \sigma_f^2 \exp\left(-\frac{1}{2} \sum_i \sum_{j>i} \left(\frac{1/r_{ij}(\mathbf{x}) - 1/r_{ij}(\mathbf{x}')}{l_{\phi(i,j)}} \right)^2 \right) \quad (5)$$

where r_{ij} defines the Euclidean distance between atoms i and j, σ_f^2 indicates the signal variance, along with the constant offset σ_c^2, and the length scales $l_{\phi(i,j)}$ for each atom pair type $\phi(i, j)$. The $1/r_{ij}$ term provides a strong physical prior, leveling out the sharp increase in repulsive force between pairs of atoms if the distance between them becomes too short.

The electronic structure calculations at the most commonly used levels of methodology, such as density functional theory, provide the atomic forces with insignificant additional computational effort and thereby also the gradient of the energy. In order to make use of this information and more accurately reproduce the PES with the surrogate model surface, the gradients are included in the covariance matrix. This requires extending the GP to a vector-valued framework. The joint distribution over predicted energy and atomic forces involves a block covariance matrix:

$$\mathbf{K}_{\text{full}} = \begin{bmatrix} \mathbf{K}_{ff}(\mathbf{X}, \mathbf{X}) & \mathbf{K}_{f\nabla}(\mathbf{X}, \mathbf{X}) \\ \mathbf{K}_{\nabla f}(\mathbf{X}, \mathbf{X}) & \mathbf{K}_{\nabla\nabla}(\mathbf{X}, \mathbf{X}) \end{bmatrix} \quad (6)$$

The additional blocks, such as $\mathbf{K}_{f\nabla}$ and $\mathbf{K}_{\nabla\nabla}$, represent the covariances between energy and forces, and between different force components, respectively, and derive from analytical derivatives of the base kernel k_{ff}. [22, 16]

To ensure the GPDimer algorithm is stable and avoids unphysical regions of the PES, [22, 16] two primary guardrails are employed. First, a trust radius prevents the GP from extrapolating too far into unknown regions. New configurations must fall within the defined radius relative to known configurations in the "1D max log" distance metric, defined by:

$$D_{\text{1Dmaxlog}}(\mathbf{x}_1, \mathbf{x}_2) = \max_{i,j} \left| \log \frac{r_{ij}(\mathbf{x}_2)}{r_{ij}(\mathbf{x}_1)} \right| \tag{7}$$

In other words, the maximum absolute log-ratio of interatomic distances between two configurations determines the trust radius. This metric possesses the useful property of invariance to molecular rotations. However, its fundamental reliance on a fixed-index comparison between atoms creates a critical inconsistency with the physical symmetries of the system. This discrepancy motivates our later use of a permutation-invariant Optimal Transport measure.

Second, a constraint on the step-size prevents atoms from moving too close to each other as this would produce extreme atomic forces. The maximum allowed step length for any atom gets directly limited by its distance to its nearest neighbor, modulated by a `ratio_at_limit` parameter:

$$L_{\text{max}} = 0.5 \cdot (1.0 - \text{ratio}_{\text{limit}}) \cdot d_{\text{min}} \tag{8}$$

where d_{min} represents the minimum interatomic distance in the current configuration. A `ratio_at_limit` value approaching 1.0 enforces a highly cautious, stability-focused search with small steps, while a value approaching 0.0 allows for more aggressive, larger steps to accelerate convergence in well-behaved systems.

2.2 Optimal Transport GP

The robustness and computational scaling of GP-accelerated searches link directly to the metric used to define dissimilarity between molecular configurations, $D(\mathbf{x}_i, \mathbf{x}_j)$. Metrics that depend on fixed atom indices, such as the "1D max log" metric (Eq. 7), lack invariance to the permutation of identical atoms. Even low-energy events, such as the rotations of a methyl group, may swap indices of two hydrogen atoms, resulting in a large, unphysical distance in this metric. To solve this, we ground our dissimilarity measure in Optimal Transport (OT) theory [36, 37]. OT provides a deterministic prescription to "lift" a geometric distance between points into a distance between entire distributions of points. We treat each molecular configuration $\mathbf{x}$ as a discrete distribution of element-typed (i.e., "colored") points in three-dimensional space. Conceptually, OT frames the distance calculation as a "transport problem." If we consider a configuration $\mathbf{x}_i$ to be piles of "mass" (the atoms) and configuration $\mathbf{x}_j$ as a set of "holes" into which the mass must be moved, then the cost of moving one unit of mass from a point $\mathbf{r}_a$ to a point $\mathbf{r}_b$ is the Euclidean distance $\|\mathbf{r}_a - \mathbf{r}_b\|$. The Earth Mover's Distance (EMD), the computational analogue of the Wasserstein-1 distance, finds the "transport plan" that minimizes the total work, i.e., the (mass × distance) summed over all movements. By enforcing coloring, we have that "carbon" masses only move to "carbon" holes, respecting chemical identity. More importantly, the EMD inherently solves the permutation problem as the optimal transport plan corresponds to the optimal matching between the two sets of atoms, automatically assigning atom 1 in $\mathbf{x}_i$ to its "correct" counterpart in $\mathbf{x}_j$, regardless of its index. This yields a physically meaningful distance

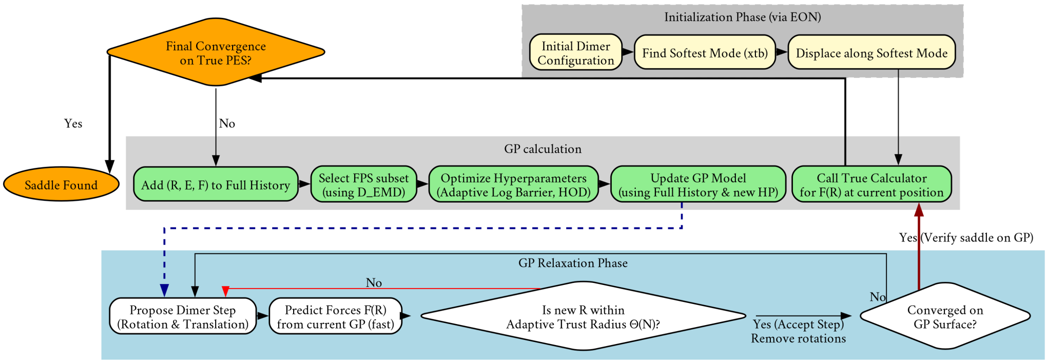

Initialization Phase (via EON)
Initial Dimer Configuration
Find Softest Mode (xtb)
Displace along Softest Mode
Final Convergence on True PES?
Yes
No
Saddle Found
GP calculation
Add (R, E, F) to Full History
Select FPS subset (using D_EMD)
Optimize Hyperparameters (Adaptive Log Barrier, HOD)
Update GP Model (using Full History & new HP)
Call True Calculator for F(R) at current position
Yes (Verify saddle on GP)
GP Relaxation Phase
No
Propose Dimer Step (Rotation & Translation)
Predict Forces F(R) from current GP (fast)
Is new R within Adaptive Trust Radius Θ(N)?
Yes (Accept Step) Remove rotations
No
Converged on GP Surface?

Figure 2. Overview of the OT-GP method. Red arrows indicate rejected proposals (outside the trust radius); blue dashed arrows denote the transition from a freshly-updated GP model back to the relaxation loop; dark-red bold arrows highlight the final verification step on the true PES. The flowchart therefore encapsulates the hierarchical control strategy of OT-GP: (i) cheap GP-driven exploration, (ii) data-driven trust-radius and variance regularisation, and (iii) intermittent high-fidelity validation that guarantees convergence to the first-order saddle point on the PES.

that correctly identifies symmetrically equivalent configurations as identical. We leverage this robust, permutation-invariant metric in our Farthest Point Sampling (FPS) to select a truly diverse set of configurations for hyperparameter optimization. Figure 2 summarizes the key modifications to the algorithm workflow in this improved GP acceleration of saddle point searches.

2.2.1 Farthest Point Subsampling

The hyperparameter optimization required by a conventional GP requires a repeated inversion of a covariance matrix whose dimensions equal the total number of scalar degrees of freedom $O((M_{data} \cdot N_{atoms})^3)$. To avoid this rapid scaling of computational effort, we instead construct a compact, maximally diverse training subset $\mathcal{S} \subset \mathcal{X}$ (where $\mathcal{X}$ denotes the full set of configurations for which the PES has been sampled) by means of Farthest Point Sampling (FPS) [38]. At each iteration, the distance $D(\mathbf{x}_i, \mathbf{x}_j)$ between every candidate configuration $\mathbf{x}_i \in \mathcal{X} \setminus \mathcal{S}$ and all members of the current subset $\mathcal{S}$ is evaluated. The next point is the one whose minimum distance to the subset attains the maximum.

$$\mathbf{x}_{\text{next}} = \arg\max_{\mathbf{x}_i \in \mathcal{X} \setminus \mathcal{S}} \Big[\min_{\mathbf{x}_j \in \mathcal{S}} D(\mathbf{x}_i, \mathbf{x}_j) \Big]. \tag{9}$$

The application of Eq. 9 is repeated until a user-specified cardinality M_{sub} is reached. For guaranteed continuity of the dimer path, the two most recent configurations are always retained. For maximum efficiency, we avoid representations of atomic environments, and operate instead with Cartesian coordinates. However, the Euclidean distance measure for a molecule is a poor metric for distinguishability, since it lacks invariance to rotations or translations and possesses no concept of atomic types. We therefore introduce a new measure of distance for the FPS.

2.2.2 Intensive Earth Mover's Distance

The standard Earth mover's distance quantifies the minimum work to transport one distribution of mass to another, resulting in an extensive property that scales with system size. We introduce an intensive formulation $D(\mathbf{x}_i, \mathbf{x}_j)$ that respects chemical identity, permutation invariance, and size independence. This formulation allows the measure to be interpreted with chemical intuition and transferred across different molecules in a size-independent manner.

For a given atom type, t a linear assignment problem is solved to match the atoms of that type between two configurations, yielding the per-type average displacement

$$\bar{d}_t = \frac{1}{N_t} \min_{\pi \in \Pi_{N_t}} \sum_{k=1}^{N_t} \left\| \mathbf{r}_{k,t}^{(1)} - \mathbf{r}_{\pi(k),t}^{(2)} \right\|. \tag{10}$$

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:

$$D(\mathbf{x}_i, \mathbf{x}_j) = \max_t \bar{d}_t(\mathbf{x}_i, \mathbf{x}_j). \tag{11}$$

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.

The hyperparameter vector, θ, which includes length scales and signal variance, undergoes optimization by maximizing the marginal log-likelihood,

$$\log p(\mathbf{y}|\mathcal{S}, \theta), \tag{12}$$

on the FPS subset $\mathcal{S}$. To ensure robust and efficient convergence, we implemented an iterative optimization scheme with key enhancements. As with the earlier GPDimer, a Student's t-distribution prior is used, but for OTGPD we adapt the prior to apply stronger constraints as the chemical system's complexity increases, thereby preventing the model from overfitting. We set a floor of 28 for the prior's degrees of freedom to ensure a stable, near-Gaussian shape, and increase the value proportionally with the number of interaction types to enforce stronger regularization where needed.

2.2.3 Computational Cost

The FPS construction costs $\mathcal{O}(M_{\text{sub}} M_{\text{data}})$ distance evaluations, each of which scales linearly with the number of atoms. For any fixed subset, hyper-parameter optimisation involves the inversion of a covariance matrix of size $(M_{\text{sub}} N_{\text{atoms}}) \times (M_{\text{sub}} N_{\text{atoms}})$, i.e.

$$\mathcal{O}((M_{\text{sub}} N_{\text{atoms}})^3), \tag{13}$$

where $M_{\text{sub}} \ll M_{\text{data}}$. An initial subset of 10 points is found to work well when used with adaptive growth of the history based on the hyperparameter oscillation and a hard user-defined upper limit of $M_{\text{sub}} = 30$. To ensure that none of the computationally intensive PES calculations goes unused, the full history $\mathcal{X}$ is used for the GP prediction of the energy on the surrogate surface.

2.2.4 Hyperparameter Oscillation Detection

Re-optimization of the hyperparameters on a dynamically changing subset can lead to unstable estimates that oscillate between iterations. A hyperparameter oscillation detection (HOD) heuristic monitors these oscillations over a moving window of the last W optimization steps. We define an oscillation indicator, $O(\theta_j, t)$, for each hyperparameter component θ_j at optimization step t:

$$O_j(t) = \begin{cases} 1 & \text{if } \operatorname{sgn}[\Delta\theta_j(t-1)] \neq \operatorname{sgn}[\Delta\theta_j(t-2)], \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

where $\Delta\theta_j(t) = \theta_j(t) - \theta_j(t-1)$. We then compute the total oscillation fraction, f_{osc}, over the window.

$$f_{\text{osc}} = \frac{1}{D(W-2)} \sum_{t=2}^{W-1} \sum_{j=1}^{D} O(\theta_j, t). \tag{15}$$

If f_{osc} exceeds a threshold, p_{osc}, the optimization is flagged as being unstable. In response, the size of the FPS subset, M_{sub}, is incrementally increased and the optimization re-run, up to a maximum of three retries.

2.2.5 Adaptive Barrier for Signal Variance

The signal variance hyperparameter, σ_f^2, governs the overall amplitude of the GP prior. When σ_f^2 grows without bounds, the GP mean surface flattens, the predictive variance explodes, which in turn may drive the dimer into unphysical regions. To prevent this, we augment the marginal log-likelihood (MLL) objective function, with a logarithmic barrier:

$$\mathcal{L}_{\text{eff}}(\theta) = \underbrace{\log p(\mathbf{y} \mid \mathcal{S}, \theta)}_{\text{MLL}} - \mu \log(\lambda_{\text{max}} - \log \sigma_f^2) \tag{16}$$

where $\lambda_{\text{max}} > 0$ fixes the absolute upper bound for $\log \sigma_f^2$. Here, we use $\lambda_{\text{max}} = \log 2$ which corresponds to $\sigma_f^2 = 2$ in the kernel units and $\mu \geq 0$ controls the barrier strength.

The barrier strength grows linearly with the total number of collected data points,

$$\mu(N) = \mu_0 + \alpha N, \qquad \mu(N) \leq \mu_{\text{max}}, \tag{17}$$

with μ_0 a small seed (e.g. 10^{-4}), α a modest growth factor (e.g. 10^{-3}), and $\mu_{\text{max}} = 0.5$ a user-defined ceiling. This schedule yields a weak barrier when the data is sparse to allow the GP to remain flexible and enforces an increasingly strict bound as the data set expands, thereby preventing runaway variance.

During optimization, the gradient of the barrier with respect to the optimization variable $w_0 = \log \sigma_f^2$,

$$\frac{\partial}{\partial w_0}[-\mu \log(C - w_0)] = \frac{\mu}{C - w_0}, \tag{18}$$

is added to the usual derivative of the MLL. The barrier therefore incurs negligible computational overhead while guaranteeing that the effective objective (Eq. 16) remains finite and well-behaved throughout the hyper-parameter optimization.

The dimer calculations on the GP surface involve two additional enhancements, described below.

2.2.6 Adaptive Trust Radius for Early Stopping

A new midpoint position of the dimer is accepted after a translation only if the atomic coordinates lie within a data-driven trust region. The distance metric is the intensive EMD defined in Eq. 11. Let

$$d_{\mathrm{EMD}}(\mathbf{x}_{\mathrm{cand}}, \mathbf{x}_{\mathrm{nn}})$$

be the EMD between the candidate configuration $\mathbf{x}_{\mathrm{cand}}$ and its nearest neighbour $\mathbf{x}_{\mathrm{nn}}$ in the current training set. We require

$$d_{\mathrm{EMD}}(\mathbf{x}_{\mathrm{cand}}, \mathbf{x}_{\mathrm{nn}}) \leq \Theta(N_{\mathrm{data}}, N_{\mathrm{atoms}}), \tag{19}$$

where Θ is an adaptive threshold that expands as the surrogate gathers information. The functional form follows an exponential saturation curve:

$$\Theta_{\mathrm{earned}}(N_{\mathrm{data}}) = T_{\mathrm{min}} + \Delta T_{\mathrm{explore}} \cdot \left(1 - e^{-k\, N_{\mathrm{data}}}\right), \tag{20}$$

$$k = \frac{\ln 2}{N_{\mathrm{half}}}. \tag{21}$$

where T_{min} denotes a minimal safe radius that prevents the dimer from taking trivially small steps, $\Delta T_{\mathrm{explore}}$ sets the maximal additional exploration distance that the algorithm may allow, and N_{half} corresponds to the number of reference points required for the threshold to reach 50% of its maximum value, thereby controlling the rate of expansion.

A physically motivated ceiling prevents the trust radius from becoming unrealistically large for any system size, defined as

$$\Theta_{\mathrm{phys}}(N_{\mathrm{atoms}}) = \max\left(a_{\mathrm{floor}}, \frac{a_A}{\sqrt{N_{\mathrm{atoms}}}}\right), \tag{22}$$

with a_{floor} a user-defined lower bound and a_A a scaling constant that reflects typical inter-atomic separations.

The more restrictive of the two bounds forms the final criterion

$$\Theta(N_{\mathrm{d,a}}) = \min(\Theta_{\mathrm{earned}}(N_{\mathrm{data}}), \Theta_{\mathrm{phys}}(N_{\mathrm{atoms}})). \tag{23}$$

When a candidate violates the inequality in Eq. 19, the algorithm rejects the step, performs an evaluation of the PES at the current midpoint, adds the new data point to the training set, and recomputes Θ. This targeted data acquisition actively improves the surrogate model where it proves to be unreliable, ensuring the step size adapts continuously to the knowledge encoded in the surrogate model.

2.2.7 Rotation Removal

Since the dimer method involves a projection of the atomic forces, Newton's third law is not necessarily obeyed even though the forces obtained from the PES do. The system as a whole could, therefore, translate and rotate. The GP model does not inherently enforce rotational or translational symmetry so net rotations and translations could influence its predictions. To prevent this this, our algorithm purifies each proposed translation step, $\mathbf{s}$, by projecting out these displacements. We calculate the pure internal step, $\mathbf{s}_{\text{int}}$, by subtracting the component of the step that lies in the subspace of rigid-body motion:

$$\mathbf{s}_{\text{int}} = \mathbf{s} - \sum_{k=1}^{6} (\mathbf{s} \cdot \mathbf{u}_k)\mathbf{u}_k \tag{24}$$

where $\{\mathbf{u}_k\}$ represents an orthonormal basis that spans the external degrees of freedom. We generate this basis through a Gram-Schmidt orthonormalization of the infinitesimal rotation and translation basis vectors, as detailed in the SI (section S1.1).

This projection method provides an alternative to other approaches. For example, point-set registration techniques, such as those found in IRA [39], or graph matching in d-SEAMS [40], can remove global rotations and translations but often incur significant computational costs. The GP kernel, Eq. 5, lacks permutation invariance, so enforcing it at the optimization step offers no real advantage. Quaternion-based methods [41] have been found to be slower and less stable for the optimization process [33].

The linear orthogonal projection requires only six inner products and a rank-6 update, incurring negligible overhead relative to the cost of evaluating the GP model, while ensuring that the dimer translation that proceeds to the next GP evaluation always uses $\mathbf{s}_{\text{int}}$.

2.3 Computational Details

The OT-GP algorithm is tested by accelerating the dimer minimum mode following search (OTGPD) on a benchmark of 238 organic molecules that contain one or two fragments, and range from 7 to 25 atoms derived from a set reported earlier [10, 16]. The `eon` software [34] is used to generate the initial orientation of the dimer and provides connection to the electronic structure calculation, as in the previous GP-Dimer implementation [16]. All codes including the OTGP implementation and analysis scripts reside on Github. Materials Archive hosts generated data (see the SI).

The PES is obtained from Hartree-Fock (HF) calculations with a 3-21G basis set using the NWChem [42] software package interfaced through the `i-pi` [43] server-client optimizer interface. The calculations account for the spin state, unrestricted HF for doublets and restricted HF for singlets. For approximate and inexpensive calculations, such as the initial orientation of the dimer and Mayer-Wiberg bond-order analysis for fragments [44], we use `ase` [45] with `xtb` and the `tblite` package for the `GFN2-xTB` [46, 47] semi-empirical method.

Snakemake [48] orchestrates runs on computer cluster with Intel Xeon Platinum 8358 CPUs. The saddle point calculations are deemed converged when the root-mean-square atomic force falls below 0.01 eV/Å. Runs terminate upon exceeding a 4-hour time limit, or if NWChem terminates with an error. The SI contains additional details.

3 Results

3.1 Benchmark Calculations

3.1.1 Reliability

The performance of the OT-GP method is compared against the previous GPDimer method and the original dimer method without GP acceleration on a benchmark set of 238 molecular systems, constructed by Hermes and coworkers.[10] Figure 1 summarizes the results of this comparison.

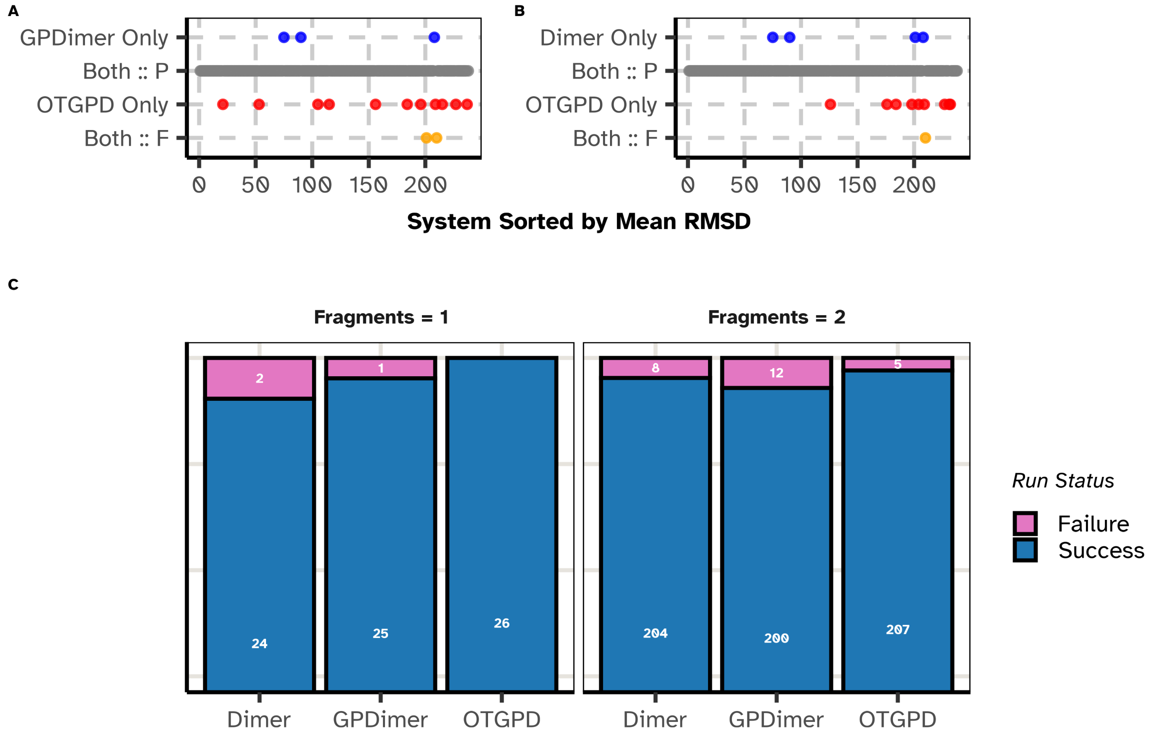


Figure 1. *Comparison of the performance of the OTGPD saddle point search method against GPDimer and regular dimer without GP acceleration for 238 molecular systems. A calculation exceeding 4 hours or raising an error in the electronic structure calculation counts as a failure. (A, B) Success outcomes for each system, ordered along the horizontal axis by mean root-mean-square deviation (RMSD) between the saddle point and the initial atomic structure. Red dots denote systems where only OTGPD succeeded, blue dots where only the alternate method succeeded, and orange dots where both failed. (C) Bar chart summarizing success rate of each method for initial structures that represent a single fragment and those representing two fragments. The single-fragment cases represent typical saddle point searches, while the two-fragment cases test performance on more complex, often arbitrary dissociation pathways. The OTGPD method demonstrates a clear advantage. In addition to a mutual success rate exceeding 93% in both comparisons, OTGPD uniquely finds the saddle point for an additional 11 systems (4.6%) compared to GPDimer (1.3%) and additional 9 systems (3.8%) compared to Dimer (1.7%).*

The comparison in Figure 1 (A, B) reveals significant performance advantage of the

OTGPD. While all three methods find a high degree of success (over 93%), demonstrating the general utility of dimer-based approach, the critical differentiator appears in the cases where only one method succeeds. OTGPD uniquely finds the correct saddle point on 4.6% of the systems compared to GPDimer, which in turn only succeeds alone on 1.3% of the cases. This trend continues against the regular dimer method, where unique success rate of the OTGPD, 3.8%, more than doubles that of the dimer, at 1.7%. These statistics show that OTGPD succeeds on a significant number of systems where the others fail.

An analysis of the failure modes in Figure 1 (C) provides further insight. For the 26 single-fragment systems, which represent typical starting configurations of the atoms, OTGPD achieves a perfect success record with zero failures. The systems where the bonding analysis of the initial atomic structure identifies two separate, non-bonded fragments are more challenging and quite unusual as starting points for saddle point searches. In most of these systems, the two fragments are far enough apart that the atomic forces are attractive and a minimization of the energy would bring them closer to each other. This displacement typically represents the minimum mode. The saddle point search method, however, is designed to drive the system against the force acting along the minimum mode and therefore pushes the fragments further apart, at least in the first few iterations. The path traced out by in the saddle point search eventually can turn around due to the minimization of the energy in orthogonal degrees of freedom, but this is indeed a highly unusual feature of benchmark data for saddle point searches.[10] We, nevertheless, choose to include these two-fragment systems in our tests, but exclude systems where the bonding analysis detects three or more fragments. OTGPD maintains the lowest failure rate of all three methods also for these two-fragment systems (5 failures, versus 12 for GPDimer and 8 for Dimer).

We further investigate the four systems (Fig. 1 (A), blue dots) where the GPDimer runs succeed but the OTGPD runs fail. This discrepancy does not stem from a flaw in the OTGPD algorithm but from an artifact of the dimer initialization procedure (visualized in Fig. 2). For OTGPD runs, we form the initial dimer configuration by displacing atoms along the softest mode found by an inexpensive semi-empirical (`xtb`, `GFN2-xTB`) calculation. In these specific cases, this displacement produced an unphysical starting geometry with high atomic forces or atoms in close contact. When any GP-accelerated method begins from such a high-energy, high-force baseline, the initial surrogate model learns high forces and energies corresponding to unphysical geometries are not anomalous, thus “poisoning” the run, leading to optimization failures. The GPDimer benchmark used a different initialization method which, for these systems, happened to produce a viable starting configuration. Robust dimer initialization routines are not a solved problem, nor the focus of this work. As detailed in the SI, when the OTGPD method starts from the same initial geometries used by the GPDimer, it converges successfully and more efficiently. For these four systems, the re-run OTGPD required a near-identical total number of HF evaluations (146 for OTGPD vs. 148 for GPDimer) while completing in less than half the average wall time (OTGPD, 51.3 minutes vs. GPDimer, 109.8 minutes), despite running with half the number of parallel processes (OTGPD 8 vs GPDimer 16).

One of the calculations, the one for system D110, is illustrated in Figure 2. Here, the GPDimer search fails after its signal variance hyperparameter explodes, leading to a physically unrealistic configuration which crashes the electronic structure calculation by NWChem. In contrast, the OTGPD has stable values for the hyperparameters and locates the saddle point efficiently.

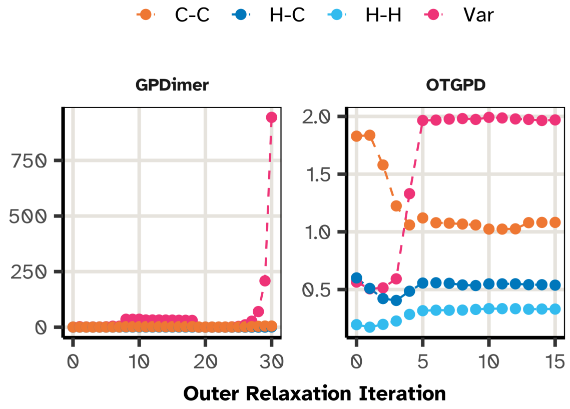


Figure 2. *Evolution of the values for the hyperparameters in a representative failure case, system D110. (Left) The GPDimer search fails after 30 iterations as the signal variance (Var) explodes, leading to an error after 38 minutes. (Right) The OTGPD framework maintains a stable signal variance, which saturates at the adaptive barrier limit. This stability allows the search to converge successfully.*

While the aggregate statistics demonstrate the OTGPD's superior reliability, individual cases provide a clearer physical intuition for the importance of its stability controls. System D150, a hydrogen abstraction reaction between a methyl and a 2-hydroperoxyethyl radical, provides an example, illustrated in Figure 3. Starting from the initial geometry shown in Figure 3 (A), the GPDimer search fails catastrophically, resulting in a fragmentation of the molecule into nine components as shown in Figure 3 (B) with atoms colliding into each other. This failure mode exemplifies a surrogate model collapse, likely driven by an unconstrained variance explosion. In contrast, both the standard dimer calculation, illustrated in Figure 3 (C) and the OTGPD correctly navigate the potential energy surface and converge on the same saddle point. This case study underscores that the innovations in OTGPD provide more than a speed increase; they introduce essential guardrails that ensure the reliability of the search, preventing the severe failures that characterized the previous generation of the GPR-accelerated saddle point search method.

3.1.2 Saddle Consistency

We next assess the quality of the saddle points located by each method. A direct comparison of barrier heights proves to be difficult particularly for systems that begin as two fragments since the search path is indirect. For these cases, the long and curved search path can result in convergence to different saddle points depending on minor variations in the trajectory, making a one-to-one comparison of single runs inconclusive.

Our analysis, therefore, focuses on algorithmic consistency across the full set of mutually successful runs. When compared against the baseline dimer method, the OTGPD method demonstrates high fidelity, locating transition states of effectively equal energy

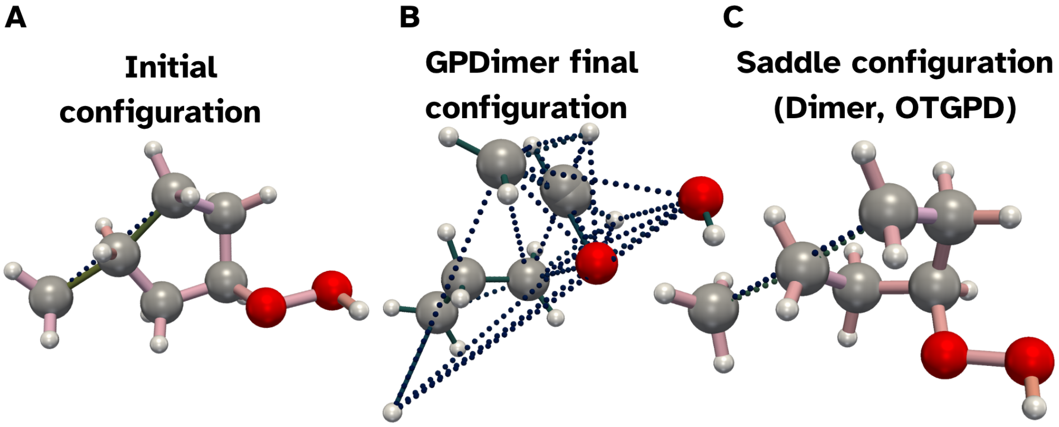

A
Initial configuration
B
GPDimer final configuration
C
Saddle configuration (Dimer, OTGPD)

Figure 3. Comparison of saddle search outcomes for system D150, highlighting the stability of the OTGPD method. Bond coloring visualizes the Wiberg bond order, with dots used when the order is below 0.5, representing non-bonded interaction. (A) The initial geometry for the saddle point searches. (B) The GPDimer search fails and ends up fragmenting the molecule. (C) Both the standard dimer and the OTGPD method successfully converge to the same saddle point.

in 89.7% of cases. In the few instances with differing outcomes, the energy differences remain small (< 0.05 eV) and show no systematic bias toward either method. The comparison with GPDimer reveals a similarly high degree of agreement, with both methods finding saddle points with the same energy in over 75% of the systems.

These results show that significant gains in speed and reliability do not come at the expense of accuracy, as it consistently reproduces the results of the baseline dimer method.

3.1.3 Performance and Computational Cost

Having established the superior reliability of the OTGPD method, we now examine its computational efficiency. An ideal algorithm must not only work well but also work fast, minimizing both the total time-to-solution and the number of expensive electronic structure calculations.

Figure 4 (A) presents a performance profile [49], a visualization for benchmarking solver efficiency which measures the proportion of best in class performance across systems. The profile for OTGPD dominates the other two methods at every performance ratio, r. OTGPD solves over 70% of the problems as the fastest method ($r = 1$) and reaches its maximum success rate within a factor of two of the best time ($r \approx 1.5$). In stark contrast, GPDimer requires four times the best time ($r = 4$) to solve 90% of its successful cases, and the standard Dimer lags even further behind.

While wall time reflects overall efficiency, the number of electronic structure calculations often represents the core computational bottleneck. In the calculations presented here a particularly fast electronic structure method is used, HF. In many cases the computational effort of each PES evaluation is much more involved and the overhead

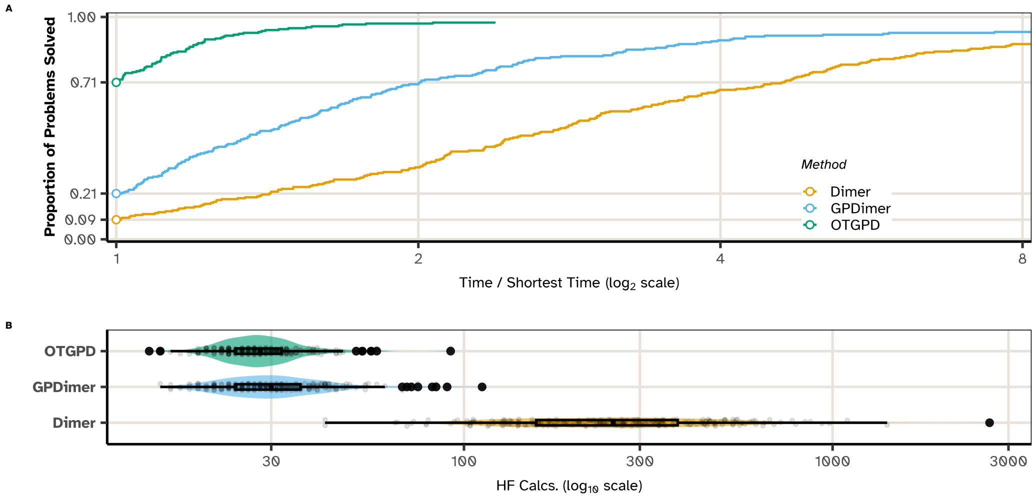


Figure 4. *Comparison of the computational efficiency of the OTGPD, GPDimer, and standard dimer methods. (A) A performance profile graph comparing the wall-clock time needed to reach the saddle point. The vertical axis shows the proportion of problems solved within a performance ratio r of the best time recorded for that system. A curve that is higher and further to the left indicates superior performance. The OTGPD method (teal) solves 70.6% of problems with the best time (r=1), reflected in its median performance ratio of 1.00. This significantly outperforms both GPDimer (win rate 20.6%, median r=1.39) and the standard dimer method (win rate 8.8%, median r=2.65). (B) Violin and box plots showing the distribution of the number of HF calculations for the successful runs of each method (populations detailed in Figure 1). Over these valid systems, both GP-accelerated methods reduce the median number of HF calculations by an order of magnitude from 254 for the standard dimer method to 30 for GPDimer and 28 for OTGPD. These gains in data efficiency directly lead to the superior wall-time performance seen in (A), with OTGPD's mean time to solution (12.6 min) reducing the wall time by nearly a half compared to the standard Dimer (23.7 min) and to less than a half compared to GPDimer (28.3 min).*

of the GP thereby smaller in comparison. Figure 4 (B) shows the the number of HF calculations for all successful runs. As previously demonstrated [16], GP-acceleration provides an order-of-magnitude reduction in this cost compared to the standard dimer method. The standard dimer method exhibits a wide distribution centered around 300 HF calculations. Both GPDimer and OTGPD shift this distribution down to a median of approximately 30 calculations. Notably, while the median number of HF calculations for OTGPD and GPDimer appear similar, the distributions and mean wall times reported in the caption reveal a crucial difference. A subset of long-running calculations skews the GPDimer performance, an issue that the algorithmic enhancements of the OTGPD resolves, resulting in a tighter performance distribution and a mean time-to-solution less than half that of GPDimer.

To understand the source of this performance difference, we examine the detailed optimization trace of a representative system, D136, illustrated in Figure 5. The GPDimer trace reveals large spikes in the per-iteration hyperparameter optimization time, which coincide with a transient, explosive increase in the signal variance and a temporary stall in force convergence. In stark contrast, the OTGPD trace shows stable optimization times and smooth, monotonic force convergence. This difference dramatically impacts the final performance: the OTGPD locates the saddle point in 19.9 minutes using 28 HF calculations, while the GPDimer needs 45.8 minutes and 39 HF calculations.

Crucially, this comparison remains valid despite the different initial orientation of the dimer. Both algorithms first perform a rotation on the PES, a step that equalizes the starting conditions by aligning the search with the physically correct softest mode. This procedure yields nearly identical starting geometries and therefore, similar GP models for the main search phase. The minor difference (≈ 2 calls) during this initial rotation vanishes in the context of the overall improvement of 11 calls for the OTGPD. The OTGPD takes 20 minutes, compared to the 45 minutes for the GPDimer. Therefore, the primary performance gains stem directly from the superior stability and efficiency of OTGPD's subsequent optimization steps. See also a detailed comparison in the SI.

Finally, a hierarchical Bayesian analysis [33] provides a rigorous confirmation of these findings (details in the SI). The model shows that OTGPD reduces the median time-to-solution to less than a half compared to the dimer method (9.0 min vs 20.5 min) and runs $\approx$30% faster than the GPDimer (9.0 min vs 12.8 min), with a greater than 99% probability that OTGPD outperforms the other two methods.

4 Discussion

The efficiency of the OTGPD method arises from a hierarchical system of data-driven controls that stabilize both the surrogate model and the geometry search it guides. By analyzing the interplay between our methods for hyperparameter optimization and the mechanisms governing the dimer translation, we can develop a physical intuition for the algorithm's robustness and speed.

We first consider the ideal scenario for a GPR-accelerated search. An ideal surrogate would learn the true potential-energy surface from sparse data; the dimer based saddle point search would then converge rapidly with stable hyperparameters, and the GP

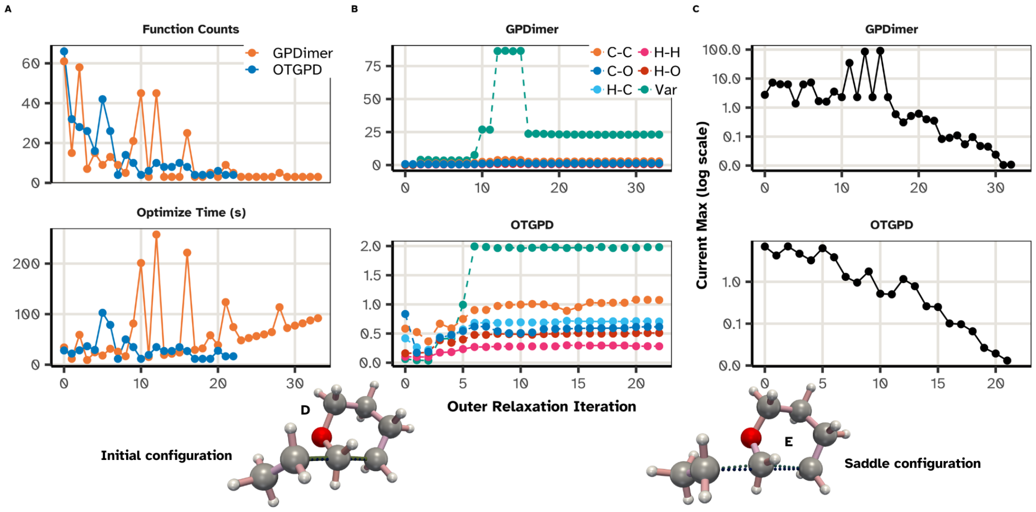


Figure 5. *A detailed performance trace for a representative system, D136, for which OTGPD converges in 19.9 min versus 45.8 min for GPDimer. (A) The per-iteration cost of hyperparameter optimization, showing the number of electronic structure calculations (top) and wall time (bottom). The GPDimer trace (orange) exhibits large, erratic spikes that are absent for OTGPD (blue). (B) The evolution of the key hyperparameters. The GPDimer method shows a transient instability where the signal variance (teal) to a large value, coinciding with the cost spikes seen in (A). The adaptive barrier in OTGPD, however, maintains hyperparameter stability. (C) The convergence of the maximum force component on the true potential energy surface, showed on a log scale. The hyperparameter instability in the GPDimer calculation disrupts the smooth convergence of the geometry search. (D, E) The initial and saddle point configurations for the hydrogen abstraction reaction. For visual clarity, a Wiberg Bond Order cutoff of 0.5 highlights the key bond-breaking and bond-forming events. Ultimately, the combination of model stability and smooth convergence for the OTGPD allows it to find the saddle point using only 28 HF calculations, compared to the 39 required by the less stable GPDimer run.*

would act as a perfect accelerator with no wall-time overhead. In practice, this ideal fails for two fundamental reasons.

First, the surrogate operates under data scarcity. A typical saddle search provides the model with only a few dozen training points, many of which represent unphysical arrangements of the atoms. This provides insufficient information to learn the true, high-dimensional energy landscape, and is in sharp contrast to machine-learned potential functions that rely on thousands of data points [50] using specialized features [51]. Second, the GP's marginal-likelihood objective possesses an intrinsically shallow ridge in the signal-variance direction. This allows the optimizer to increase σ_f^2 arbitrarily to accommodate strained outlier configurations. This uncontrolled variance growth leads to a flattening of the mean surface and an explosion of the predictive variance, which in turn guides the saddle point search algorithm to propose unphysical steps that can lead to failures in the electronic structure calculation.

4.1 Improved Wall-time Efficiency

For all GP regression models, hyperparameter optimization dominates the wall time because the covariance matrix grows with every new configuration [52, 53]. The inversion cost scales as

$$\mathcal{O}((M_{\text{data}}3N_{\text{atoms}}+1)^3).$$

The optimization step rapidly eclipses the cost of the underlying calculation, even when the optimization converges in a few steps since the hyperparameters do not change much once enough data accumulates to allow the dimer to converge on the GP surface. Figure 5 (A) shows that after 20 steps, even when the hyperparameters do not change, single evaluations of the hyperparameter objective for the GPDimer (orange) grows steadily in wall time. By introducing the FPS/EMD pipeline described in sections 2.2.1 and 2.2.2 we perform the hyperparameter optimization on a fixed-size, chemically diverse subset of data which scales as

$$\mathcal{O}((M_{\text{sub}}3N_{\text{atoms}}+1)^3).$$

Practically, this means a more than two-fold reduction in mean wall time (12.6 minutes versus 28.3 min for the previous GP-Dimer) while preserving the order-of-magnitude savings in electronic structure calculations. The accuracy and stability are improved because no sub-sampling of the data through inducing points [54] or nesting approximations to the GP [55, 27] takes place.

The most common approach to handling GPR scaling involves sparse (inducing point) methods [26, 27]. These methods work well for large, static datasets, however, our "on-the-fly" design lies within a fundamentally different, dynamic regime (Fig. 2). The dynamic re-optimization reduces the advantages of standard sparse GPR which involves an additional optimization at each training step to construct the inducing point locations. We estimated this would prove prohibitive and likely negate any savings, given our median sample size of approximately 30 PES evaluations.

The FPS method confines the expensive hyperparameter optimization to a small subset (M_{sub}) while still using the full data history ($\mathcal{X}$) for the predictive step. This "data selection" approach maintains a link to the physical system as the FPS subset consists of observed configurations that show maximal diversity in EMD metric. These points "bracket" the relevant region of the PES, an interpretation which is harder in a sparse-GP context, where the inducing points represent optimized mathematical constructs. We anticipate sparse methods may prove beneficial for much larger problems requiring hundreds of evaluations, but not for the class of problems considered here.

4.2 Interpretation of the Hyperparameters

A key insight into the GPDimer method concerns the physical meaning, or lack thereof, of its hyperparameters. One might believe that the length scales, $l_{\phi(i,j)}$ in Eq. 5, of a Gaussian Process [54] correlate to physical quantities such as covalent bond radii [56, 57]. This view, however, lacks a first-principles justification. We define the kernel's length scales per atom-pair type, for example, one for C-C, C-H, etc. not per individual covalent bond. Because we optimize these parameters over all instances of a pair type in the training subset $\mathcal{S}$, they typically do not converge to specific equilibrium bond lengths. Instead, they represent a global, averaged "stiffness" for each interaction type, which represents the local PES region explored at any given moment. The hyperparameter vector θ thus contains these length scales and the signal variance, σ_f^2 shown in Eq. 5. The GP kernel uses these few parameters to learn complex, higher-order, many-body effects implicitly.

Recall that our fundamental goal in a saddle point search involves finding a stationary point, x^*, on the true, but slow to evaluate, PES. This corresponds to locating a configuration where the atomic forces, used to drive the iterations in Eq. 3, vanish. In terms of the many-body expansion (MBE) [58, 51]:

$$V(x) = \sum_{i} V_1(i) + \sum_{i<j} V_2(i,j) + \sum_{i<j<k} V_3(i,j,k) + \cdots + V_N(1,\ldots,N) \qquad (25)$$

where V_1 represents the energy of isolated atoms, V_2 captures all pairwise interactions, V_3 the three-body effects, and so on, and higher order terms are non-negligible [59]. The computational cost of sampling $V(x)$ motivates the substitution of a GP surrogate model, $f(x;\theta)$, for $V(x)$, as shown in Eq. 4. We then transfer our physical objective to this surrogate, seeking a solution to:

$$\nabla f(x;\theta) \approx 0 \qquad (26)$$

Note that the hyperparameter objective function of a GP model, however, describes something fundamentally different than Eq. (25). A GP defines a probability distribution over functions. After observing a set of N data points, $\mathcal{S}$, our surrogate surface, $f(x;\theta)$, corresponds to the mean of a posterior multivariate normal (MVN) distribution, conditioned on that data.

A critical disconnect emerges in how we select the hyperparameters, θ. θ cannot minimize the true error between the surrogate and PES, which would require extensive sampling the PES and defeat the purpose of the surrogate approach:

$$\theta_{\text{ideal}} = \arg\min_\theta \int |f(x;\theta) - V(x)|^2 dx \quad \text{(Inaccessible)} \qquad (27)$$

Instead, we select θ by maximizing the marginal log-likelihood (MLL), a purely statistical quantity derived from our sparse training subset $\mathcal{S}$, selected here via Farthest Point Sampling. This makes our actual hyperparameter objective:

$$\theta^* = \arg\max_\theta \log p(\mathbf{y}|\mathcal{S}, \theta) \tag{28}$$

which does not correspond directly to the formal physical series in Eq. (25) with a single function realized from an MVN distribution. The process of finding optimal hyperparameters by maximizing the marginal log-likelihood, $p(\mathbf{y}|\mathcal{S}, \theta)$, which measures the model's consistency, under the assumption the underlying data follows a multivariate normal distribution, conditioned on the sparse data it has seen. No other direct information about the Eq. (25) enters the algorithm.

Thus, the standard approach implicitly relies on a chain of unjustified assumptions:

$$\underbrace{\arg\max_\theta \log p(\mathbf{y}|\mathcal{S}, \theta)}_{\text{We optimize this...}} \implies \theta^* \tag{29}$$

$$\implies \underbrace{f(x; \theta^*) \approx V(x)}_{\text{...assuming this...}} \tag{30}$$

$$\implies \underbrace{\nabla f(x^*) \approx 0 \text{ gives } \nabla V(x^*) \approx 0}_{\text{...to achieve this.}} \tag{31}$$

We optimize a statistical property and hope it reproduces a physical reality it was never designed to model. Our work's adaptive controls act as a corrective. They impose functional constraints on the function drawn from the GP, serving as a proxy for the unknown physical constraints that the statistical objective function lacks.

This perspective also clarifies the role of the signal variance, σ_f^2, which governs the overall flexibility of the surrogate model. Our analysis reveals that unstable or failed searches often correlate with an unphysically large signal variance. Such a large variance allows the model too much freedom to oscillate between energetically similar but geometrically distinct configurations. This physical insight directly motivates the adaptive barrier method described in Section 2.2.5, which constrains the variance to a physically reasonable range, thereby greatly enhancing the stability and robustness of the search.

4.3 Hyperparameter Guardrails

Having established that the hyperparameters lack a direct physical interpretation, we now consider the data-driven guardrails essential for the OTGPD method. The marginal-likelihood landscape for a GP surrogate often presents a shallow ridge in the signal-variance direction. Without regularization, the optimizer can increase σ_f^2 arbitrarily as new, strained configurations of the atoms enter the training set.

In GPDimer calculations, this is manifested as a rapid, uncontrolled growth of σ_f^2, illustrated in Figure 2 (A). As the signal variance increases, the model becomes pathologically flexible. It ceases to act as a physical model and instead behaves as a pure mathematical interpolator between sparse and increasingly high energy reference data. This compliance means the surrogate sees no penalty for proposing physically impossible configurations in efforts to honor the fit. This guides the saddle point search to unphysical arrangements of the atoms, such as overlapping atoms, causing the electronic structure calculation to eventually fail.

This specific failure mode requires a solution distinct from conventional L1 or L2 regularization [60]. Such methods apply a soft penalty to the objective to combat general overfitting, but a strong MLL gradient can overcome this penalty, failing to prevent the variance explosion seen in Fig. 2. We therefore draw inspiration from interior point methods [61] used in constrained optimization.

Embedding the logarithmic barrier of Eq. (16) directly into the objective function serves as a "hard wall" to enforce this constraint. Unlike a soft penalty, the barrier creates a diverging gradient (Eq. 18) as the optimizer approaches the bound $\lambda_{\max}$, which guarantees that the optimizer cannot enter the unstable region. Such constraints on σ_f^2 are interpretable, with a physically meaningful upper bound rather than a more abstract L1/L2 penalty strength. We find this gradient-based barrier necessary as a simple hard bound proved insufficient, with the optimizer "sticking" to the boundary.

Because the barrier strength μ scales with the data count, Eq. (17}), the surrogate remains flexible in the early steps. This allows the optimisation to explore a wide variance range and progressively tightens as the data set matures, exactly when the model should have settled on a physically reasonable amplitude. This adaptive behaviour eliminates the pathological growth of the variance while preserving the surrogate model's ability to capture the curvature of the true PES, as seen in Figure 2 (B).

The HOD mechanism, Eqs. (14) and (15, monitors the stability of all hyper-parameters, not only σ_f^2. Whenever frequent sign changes appear in the update history, the FPS subset enlarges, and if there are several high energy points, not all are taken, which improves the conditioning of the covariance matrix and yields a smoother marginal-likelihood surface. Empirically, the combination of barrier and HOD reduces the number of failed saddle searches from roughly twelve percent (baseline GPDimer) to two percent for the OTGPD method, while the median wall-time drops from twenty-eight to twelve minutes across the 238-molecule benchmark.

4.4 Step-size Control

While the translational step size is an important parameter in any saddle point search, its strategic importance on a surrogate surface gives way to the model's own adaptive, data-driven guardrails. In a conventional search, where each step requires an expensive calculation, the step size provides the user's primary, manually tuned lever for balancing progress and stability. Choosing an excessively large step size can destabilize the search, causing the system to overshoot the nearest saddle point and explore unphysical, high-energy regions of the energy surface, often leading to convergence failure. On surrogate surfaces, new samples have negligible computational cost. Thus the step size can take a small, conservative value without penalty. The primary driver for controlling the search therefore shifts from the step size to the adaptive parameters of the surrogate model itself.

The stability of the search is critically dependent on the distance metric used to define the trust radius. The previous GPDimer approach used the "1D max log" metric given by Eq. (7), which is not invariant to the permutation of identical atoms. This flaw means that a low-energy symmetric event, such as the rotation of a methyl group or proton transfer, can be misinterpreted as a large geometric step. This may incorrectly halt the search or add redundant symmetric configurations to the training set, which leads to ill-conditioned kernel matrices and numerical instability. By defining the trust radius with our intensive Earth Mover's Distance (D_{EMD} from Eq. (11)), we impose a physically meaningful and transferable criterion for a physically motivated, automated step-size controller. The metric's intuition, which corresponds to the average displacement of atom groups ensures that accepted steps correspond to sensible chemical transformations, not unphysical leaps. A proposed configuration that falls outside the trust region represents the direct analogue of taking an excessively large step in a traditional saddle point search.

The adaptive threshold $\Theta(N_{\text{data}})$ embodied in Eq. (19) embodies an "earned-exploration" principle. At the start of a search with only a few reference points, Θ remains close to the minimal safe radius T_{min} so the algorithm thus takes conservative steps. As the surrogate accrues more data, Θ grows smoothly toward $T_{\text{min}} + \Delta T_{\text{explore}}$, permitting larger moves when the model has become sufficiently accurate. The exponential form ensures rapid initial growth that tapers off, preventing runaway step sizes in later stages.

The physical ceiling Θ_{phys}, Eq. (22) safeguards against pathological expansions of the trust radius on systems with many atoms or on early iterations where the earned term could otherwise dominate. By scaling as $a_A/\sqrt{N_{\text{atoms}}}$ the bound reflects the typical size of atomic displacements and ensures that the trust region remains chemically reasonable irrespective of system size. Taking the minimum of the earned and physical limits, Eq. (23) therefore guarantees that every accepted step respects both data-driven confidence and universal geometric constraints.

By rejecting overly ambitious proposals early, the algorithm avoids failed electronic structure evaluations, which can be costly, and limits the number of additional reference points required to bring the surrogate back into a trustworthy regime. Consequently the median number of electronic structure calculations remains low while halving the average time-to-solution between the GPDimer and OTGPD.

4.5 Removal of Rotation

Any surrogate model that operates in Cartesian coordinates must distinguish genuine internal deformations from translation and rotation of the system as a whole. If the GP model receives a step containing a net translation or rotation of the system, the kernel evaluates distances between atoms that have undergone a uniform shift. Because the inverse-distance kernel treats every Cartesian component independently, such a shift introduces artificial changes in the pairwise distances. These artifacts in turn may lead to spurious force predictions and an artificial increase of the predictive variance.

By projecting each step onto the internal subspace eliminates this source of error. By construction, this procedure guarantees that the updated atom coordinates share the same center of mass and orientation as the previous configuration. The pairwise distances that enter the kernel therefore remain unchanged apart from the true internal deformation. The GP consequently observes a smooth evolution of the input space, which stabilizes both the mean prediction and the variance. While a general best practice for any GP-accelerated search, implementing this purification step contributes to the high success rate of our OTGPD method. This enhanced stability manifests in the marked reduction of failed runs (from twelve percent to less than two percent on our benchmark) and in the tighter distribution of wall-time values.

5 Conclusion

The Optimal Transport Gaussian Process (OT-GP) framework is presented, a suite of algorithms that solve the critical wall-time bottleneck and inherent instabilities of GP-accelerated saddle searches. Previous methods, while promising sample-efficiency, introduced rapidly increasing computational effort in the hyperparameter optimization that negated practical wall-time gains and suffered from frequent, failures. The work presented here provides a diagnosis of these problems and provides a robust, comprehensive solution.

We identified the core of the problem in two areas: a flawed conceptual understanding of the GP's statistical nature and a severe, inherent mathematical fragility in the standard derivative kernel. Instead of interpreting the hyperparameters as constants with physical meaning, their function is recognized merely as mathematical tools for model adaptation. This perspective justifies our central innovation: a farthest-point sampling strategy, guided by a chemically aware optimal transport metric, that selects a compact, representative subset of data solely for the hyperparameter optimization step. This helps deal with the cubic scaling of the computational effort and results in a halving of the mean wall time compared to its predecessor, while retaining all the data for predictions of new values on the surrogate surface. There are many more hyperparameter optimization steps than predictions.

Stability follows from three complementary controls. (i) An adaptive logarithmic barrier on σ_f^2 prevents the variance from diverging; (ii) a hyper-parameter-oscillation-detection (HOD) monitor enlarges the FPS subset whenever the optimiser exhibits sign-alternating updates, thereby improving the conditioning of the covariance matrix; and (iii) a data-driven trust-radius, based on the same intensive EMD, automatically limits translation steps to regions supported by the training data. Finally, a projection removes any overall translation or rotation from each trial step, ensuring that the GP kernel sees only genuine internal deformations.

Further state-of-the-art reductions will likely involve symmetry adaption, for example via SOFI [62], or by reinforcement learning of MLIPs driven by foundational frameworks like metatomic and metatensor [20].

The results from the benchmark calculations confirm the success of this principled approach. The OTGPD method requires on average less than half the computational effort of the GPDimer method in wall time while preserving the order-of-magnitude reduction in the number of electronic structure calculations. Previous local surrogate methods promised sample efficiency but often failed to deliver practical wall-time performance and robust performance. By confronting and correcting the flawed assumptions and inherent instabilities of these prior approaches, the OTGPD method finally delivers on that promise. The OTGPD requires a median of fewer than 30 electronic structure evaluations and an average time-to-solution under 13 minutes, coupled with a success rate of ninety-six percent, and a performance profile showing that it solves more than 70% of the benchmark at the best recorded time. The framework developed here provides a powerful approach for active learning, capable of efficiently generating the high-energy, strained transition state geometries essential for training robust, next-generation reactive machine-learned interatomic potentials. The framework's generality also invites application to entirely new scientific domains, such as complex excited-state potential energy surfaces relevant for photochemistry and enabling high-throughput adaptive kinetic Monte Carlo simulations coupled directly to electronic structure calculations. This work elevates GPR acceleration from a promising but computationally heavy tool into a truly robust and wall-time-efficient approach, a foundational technology that enables the routine and automated exploration of complex chemical landscapes.

Acknowledgments

R.G. received funding from the Icelandic Research Fund (grant number 217436-053) and financial support from Dr. Guillaume Fraux and Prof. Michele Ceriotti of Lab-COSMO. The authors thank Prof. Birgir Hrafnkelsson, Prof. Thomas Bligaard, Dr. Andreas Vishart and Dr. Miha Gunde for helpful discussions. R.G. also acknowledges valuable discussions with Dr. Amrita Goswami, Dr. Moritz Sallermann, Prof. Debabrata Goswami, Mrs. Sonaly Goswami, and Mrs. Ruhila Goswami. The calculations were carried out using resources supplied by the Icelandic Research e-Infrastructure project (IREI), supported by the Icelandic Infrastructure Fund. R.G. presented a preliminary account of the work presented here at UNOOS 2025, a conference held in honor of Prof. Debabrata Goswami's on the occasion of his sixtieth birthday. R.G. warmly dedicates this article to him on this occasion.

Conflict of Interest

We declare no conflicts of interest.

References for Paper IV

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

[2] Hendrik Schrautzer. "Identification of Mechanisms of Magnetic Transitions Using an Efficient Method for Converging on First-Order Saddle Points." In: *Physical Review B* 112.10 (2025). DOI: 10.1103/z673-hhnp.

[3] 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.

[4] David J. Wales. "Finding Saddle Points for Clusters." In: *Journal of Chemical Physics* 91.11 (Dec. 1989), pp. 7002–7010. DOI: 10.1063/1.457316.

[5] 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.

[6] 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.

[7] Antoine Jay, Miha Gunde, Nicolas Salles, Matic Poberžnik, Layla Martin-Samos, Nicolas Richard, Stefano de Gironcoli, Normand Mousseau, and Anne Hémeryck. "Activation–Relaxation Technique: An Efficient Way to Find Minima and Saddle Points of Potential Energy Surfaces." In: *Computational Materials Science* 209 (June 2022), p. 111363. DOI: 10.1016/j.commatsci.2022.111363.

[8] 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.

[9] H. Bernhard Schlegel. "Geometry Optimization." In: *WIREs Computational Molecular Science* 1.5 (2011), pp. 790–809. DOI: 10.1002/wcms.34.

[10] 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.

[11] 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.

[12] Graeme Henkelman, Blas P. Uberuaga, and Hannes Jónsson. "A Climbing Image Nudged Elastic Band Method for Finding Saddle Points and Minimum Energy Paths." In: *The Journal of Chemical Physics* 113.22 (Nov. 2000), pp. 9901–9904. DOI: 10.1063/1.1329672.

[13] Rohit Goswami. *Efficient Exploration of Chemical Kinetics.* Oct. 2025. DOI: 10.48550/arXiv.2510.21368. arXiv: 2510.21368 [physics].

[14] 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.

[15] Ignacio Fdez. Galván, Gerardo Raggi, and Roland Lindh. "Restricted-Variance Constrained, Reaction Path, and Transition State Molecular Optimizations Using Gradient-Enhanced Kriging." In: *Journal of Chemical Theory and Computation* 17.1 (Jan. 2021), pp. 571–582. DOI: 10.1021/acs.jctc.0c01163.

[16] 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.

[17] 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.

[18] Laurent Karim Béland, Peter Brommer, Fedwa El-Mellouhi, Jean-François Joly, and Normand Mousseau. "Kinetic Activation-Relaxation Technique." In: *Physical Review E* 84.4 (Oct. 2011), p. 46704. DOI: 10.1103/PhysRevE.84.046704.

[19] 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].

[20] 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].

[21] Mathias Schreiner, Arghya Bhowmik, Tejs Vegge, Jonas Busk, and Ole Winther. "Transition1x - a Dataset for Building Generalizable Reactive Machine Learning Potentials." In: *Scientific Data* 9.1 (Dec. 2022), p. 779. DOI: 10.1038/s41597-022-01870-w.

[22] 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.

[23] 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.

[24] Andrew A. Peterson. "Acceleration of Saddle-Point Searches with Machine Learning." In: *The Journal of Chemical Physics* 145.7 (Aug. 2016), p. 074106. DOI: 10.1063/1.4960708.

[25] Mykel J Kochenderfer and Tim A Wheeler. *Algorithms for Optimization.*

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

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

[28] 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.

[29] Andreas Heyden, Alexis T. Bell, and Frerich J. Keil. “Efficient Methods for Finding Transition States in Chemical Reactions: Comparison of Improved Dimer Method and Partitioned Rational Function Optimization Method.” In: *The Journal of Chemical Physics* 123.22 (Dec. 2005), p. 224101. DOI: 10.1063/1.2104507.

[30] 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.

[31] Andreas Pedersen, Sigurdur F. Hafstein, and Hannes Jónsson. “Efficient Sampling of Saddle Points with the Minimum-Mode Following Method.” In: *SIAM Journal on Scientific Computing* 33.2 (Jan. 2011), pp. 633–652. DOI: 10.1137/100792743.

[32] 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.

[33] 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.

[34] 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.

[35] Daniel Sheppard, Rye Terrell, and Graeme Henkelman. “Optimization Methods for Finding Minimum Energy Paths.” In: *The Journal of Chemical Physics* 128.13 (Apr. 2008), p. 134106. DOI: 10.1063/1.2841941.

[36] Justin Solomon. “Optimal Transport on Discrete Domains.” In: *Proceedings of Symposia in Applied Mathematics.* Ed. by Keenan Crane. Vol. 76. Providence, Rhode Island: American Mathematical Society, Sept. 2020, pp. 103–140. DOI: 10.1090/psapm/076/05.

[37] Gabriel Khan and Jun Zhang. “When Optimal Transport Meets Information Geometry.” In: *Information Geometry* 5.1 (July 2022), pp. 47–78. DOI: 10.1007/s41884-022-00066-w. arXiv: 2206.14791 [math].

[38] Albert P. Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R. Kermode, Gábor Csányi, and Michele Ceriotti. “Machine Learning Unifies the Modeling of Materials and Molecules.” In: *Science Advances* 3.12 (Dec. 2017), e1701816. DOI: 10.1126/sciadv.1701816.

[39] Miha Gunde, Nicolas Salles, Anne Hémeryck, and Layla Martin-Samos. "IRA: A Shape Matching Approach for Recognition and Comparison of Generic Atomic Patterns." In: *Journal of Chemical Information and Modeling* 61.11 (Nov. 2021), pp. 5446–5457. DOI: 10.1021/acs.jcim.1c00567.

[40] 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.

[41] Marko Melander, Kari Laasonen, and Hannes Jónsson. "Removing External Degrees of Freedom from Transition-State Search Methods Using Quaternions." In: *Journal of Chemical Theory and Computation* 11.3 (Mar. 2015), pp. 1055–1062. DOI: 10.1021/ct501155k.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] Christoph Bannwarth, Eike Caldeweyher, Sebastian Ehlert, Andreas Hansen, Philipp Pracht, Jakob Seibert, Sebastian Spicher, and Stefan Grimme. "Extended Tight-Binding Quantum Chemistry Methods." In: *WIREs Computational Molecular Science* 11.2 (2021), e1493. DOI: 10.1002/wcms.1493.

[47] 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.

[48] 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.

[49] Elizabeth D. Dolan and Jorge J. Moré. "Benchmarking Optimization Software with Performance Profiles." In: *Mathematical Programming* 91.2 (Jan. 2002), pp. 201–213. DOI: 10.1007/s101070100263.

[50] 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].

[51] 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.

[52] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms.* 22nd printing. Cambridge: Cambridge University Press, 2019.

[53] 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.

[54] Volker L. Deringer, Albert P. Bartók, Noam Bernstein, David M. Wilkins, Michele Ceriotti, and Gábor Csányi. "Gaussian Process Regression for Materials and Molecules." In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 10073–10141. DOI: 10.1021/acs.chemrev.1c00022.

[55] Alexander Denzel, Bernard Haasdonk, and Johannes Kästner. "Gaussian Process Regression for Minimum Energy Path Optimization and Transition State Search." In: *The Journal of Physical Chemistry A* 123.44 (Nov. 2019), pp. 9600–9611. DOI: 10.1021/acs.jpca.9b08239.

[56] Sami Kaappa, Estefanía Garijo del Río, and Karsten Wedel Jacobsen. "Global Optimization of Atomic Structures with Gradient-Enhanced Gaussian Process Regression." In: *Physical Review B* 103.17 (May 2021), p. 174114. DOI: 10.1103/PhysRevB.103.174114.

[57] Estefanía Garijo del Río, Sami Kaappa, José A. Garrido Torres, Thomas Bligaard, and Karsten Wedel Jacobsen. “Machine Learning with Bond Information for Local Structure Optimizations in Surface Science.” In: *The Journal of Chemical Physics* 153.23 (Dec. 2020), p. 234116. DOI: `10.1063/5.0033778`.

[58] A. J. Stone. *The Theory of Intermolecular Forces.* Second edition. Oxford: Oxford University Press, 2013.

[59] 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`.

[60] 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`.

[61] Florian A. Potra and Stephen J. Wright. “Interior-Point Methods.” In: *Journal of Computational and Applied Mathematics.* Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations 124.1 (Dec. 2000), pp. 281–302. DOI: `10.1016/S0377-0427(00)00433-7`.

[62] M. Gunde, N. Salles, L. Grisanti, L. Martin-Samos, and A. Hemeryck. “SOFI: Finding Point Group Symmetries in Atomic Clusters as Finding the Set of Degenerate Solutions in a Shape-Matching Problem.” In: *The Journal of Chemical Physics* 161.6 (Aug. 2024), p. 062503. DOI: `10.1063/5.0215689`.