Adaptive xTB: Toward Accelerated DFT Geometry Optimizations with Machine-Learned Parameters

by

Jonathan Woo

Supervisor: Anatole von Lilienfeld
July 2025

B.A.Sc. Thesis

Division of Engineering Science
UNIVERSITY OF TORONTO

Adaptive xTB: Toward Accelerated DFT Geometry Optimizations with Machine-Learned Parameters

Jonathan Woo
Bachelor of Applied Science in Engineering Science
Division of Engineering Science
University of Toronto
2025

Abstract

Accurate and scalable quantum mechanical simulations are essential for accelerating the discovery of new molecules and materials in domains such as energy storage, drug discovery, and climate technology. While density functional theory (DFT) offers good accuracy, its cubic scaling with system size makes it impractical for large-scale screening. Semi-empirical methods like GFN2-xTB offer orders-of-magnitude speedups by introducing fitted parameters, but suffer from reduced accuracy.

This thesis presents an adaptive approach for improving the accuracy of xTB without compromising its computational efficiency. By using Bayesian optimization to tune a small subset of key xTB parameters on a per-molecule basis, this work enables xTB to better approximate DFT-level results. Machine learning models, utilizing diverse molecular representations, were trained on a large dataset of DFT-optimized geometries and energies to predict these optimal parameter values.

Results demonstrate that Bayesian optimization effectively identifies parameter sets that significantly reduce the maximum atomic force on DFT geometries. However, learning these parameters with machine learning models proved challenging, likely due to the complexity and nonlinearity of the underlying parameter-observable relationship. Moreover, the study highlights that the choice of optimization objective is critical: minimizing maximum atomic force does not necessarily translate to improved performance in downstream tasks, such as reducing the number of DFT geometry optimization steps.

Acknowledgements

Completing this thesis has been a challenging yet profoundly rewarding experience. While this period coincided with a particularly difficult year in my life, I am incredibly fortunate to have received invaluable support from many individuals, without whom this accomplishment would not have been possible.

First and foremost, my deepest gratitude goes to my supervisor, Prof. Anatole von Lilienfeld, for his exceptional mentorship, intellectual guidance, and for providing me with the opportunity to explore, learn, and contribute to meaningful science. His willingness to entrust me with ambitious projects, even when I initially felt unprepared, was instrumental in my development. I am especially grateful for his repeated support and understanding during challenging times.

My sincere thanks extend to my fellow lab mate, Danish Khan, for his immense technical support throughout this project and for making our lab a vibrant and enjoyable place to work. I would also like to acknowledge Prof. Ethan Ritz, my collaborator, whose expertise was an invaluable resource as I navigated the fundamental principles of quantum chemistry.

I am also thankful to the Division of Engineering Science at the University of Toronto for its challenging and broad curriculum. This diverse academic foundation has truly enabled me to explore and delve into various research areas, including the one presented in this thesis.

I am deeply grateful to Prof. Alvin Chan and Prof. Giovanni Traverso for their profound mentorship and for generously providing me the opportunity to conduct research in the year preceding this thesis. It was in their lab that I truly discovered the immense fulfillment of working on ambitious scientific projects, a revelation that steered me away from the conventional path I was going down. Their guidance showed me the true essence of scientific inquiry, encouraging me to think critically and operate in highly uncertain, yet profoundly rewarding, areas.

Finally, and most importantly, I want to express my profound gratitude to my family: my mother Michelle, father Willie, sisters Victoria and Annabelle, and brother Sebastian. Thank you for your unwavering support, your constant belief in me, and for providing a stable and loving foundation.

To my friends, although the demands of this thesis meant we couldn't graduate together, your enduring friendship and understanding were essential to my ability to reach this milestone. Thank you for being by my side every step of the way.

Contents

List of Figures

Chapter 1

Introduction

To accelerate the discovery of new materials and molecules—a critical step toward addressing global challenges such as climate change, clean energy, and sustainable manufacturing—it is essential to accurately predict molecular properties from first principles [1, 2, 3]. Advances in computational chemistry have enabled such predictions through atomistic simulations grounded in quantum mechanics [4].

At the core of these simulations is the Schrödinger equation, which describes the behavior of electrons in a molecular system. While exact solutions are intractable for all but the simplest systems [5], approximations such as Kohn-Sham density functional theory (DFT) provide a practical balance between accuracy and computational cost [6]. Nevertheless, DFT still scales as $\mathcal{O}(n^3)$ with the number of electrons, limiting its applicability in high-throughput settings [7].

To address this, semi-empirical methods such as GFN2-xTB offer a compelling alternative [8]. By introducing empirically fitted parameters, xTB reduces the computational scaling to $\mathcal{O}(n^2)$, enabling simulations that are orders of magnitude faster than DFT [9]. However, this speed comes at the expense of accuracy—particularly for properties sensitive to electron correlation and exchange, such as atomization energies and forces [9].

This thesis explores an adaptive approach to improve the accuracy of xTB without sacrificing its computational efficiency. By leveraging machine learning to tune a small subset of global xTB parameters on a per-molecule basis, I demonstrate that it is possible to significantly reduce the error in quantum mechanical observables—bringing xTB closer to DFT-level accuracy while maintaining its scalability.

Chapter 2

Background

2.1 Levels of Theory in Quantum Mechanical Modelling

In the study of electronic structure, we aim to solve the non-relativistic time-independent Schrödinger equation [5]:

$$\mathcal{H}|\Phi\rangle = \mathcal{E}|\Phi\rangle \tag{2.1}$$

where $\mathcal{H}$ is the Hamiltonian operator, Φ is the wavefunction, and $\mathcal{E}$ is the corresponding energy eigenvalue.

By solving for the wavefunction, molecular properties of interest can be obtained as observables through the application of operators. Each physical observable has an associated operator $\hat{\mathcal{O}}$, and its expectation value is given by [5]:

$$\langle\hat{\mathcal{O}}\rangle = \langle\Psi|\hat{\mathcal{O}}|\Psi\rangle \tag{2.2}$$

However, the exact solution to the Schrödinger equation is intractable for systems beyond the simplest cases due to its many-body nature. For a system with N electrons, the wavefunction depends on $3N$ spatial variables [5]. Additionally, long-range Coulomb interactions couple each electron to all others, preventing their independent treatment [5].

To address this complexity, various levels of theory have been developed that trade off between computational cost and accuracy [5, 6]. Figure 2.1 illustrates several quantum chemical methods with differing computational scaling and predictive accuracy.

1. **Semi-empirical methods**: These occupy the lowest end of the accuracy-cost spectrum. They simplify quantum mechanical calculations by introducing empirically fitted parameters, retaining the formal structure of quantum mechanics while reducing computational overhead [8].

2. **Wavefunction-based methods**: These include techniques such as Hartree-Fock (HF), Møller-Plesset perturbation theory (MP2), and coupled cluster (e.g., CCSD(T)), which explicitly approximate or compute the electronic wavefunction [5].

3. **Density Functional Theory (DFT)**: Instead of solving for the many-body wavefunction, DFT reformulates the problem in terms of the electron density, offering a more computationally efficient approach while retaining reasonable accuracy for many systems [6].

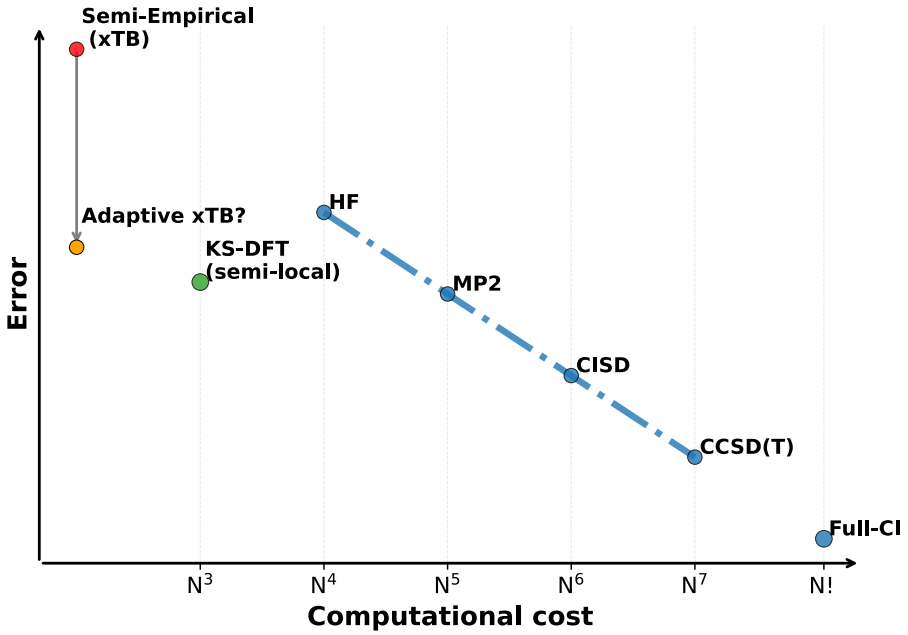

Figure 2.1: Computational cost of various quantum chemistry methods (e.g., HF, DFT, MP2, CCSD(T)) compared to adaptive xTB. These methods trade off computational scaling with accuracy [5]. While xTB with default parameters offers a significantly lower computational cost, it suffers from high error [9]. Adaptive xTB, by contrast, has the potential to reduce this error substantially—approaching the accuracy of KS-DFT on the reference dataset—while preserving its low computational cost.

2.2 The xTB Method

GFNn-xTB is a family of semi-empirical methods based on density functional tight binding (DFTB) approximations. Each model in this series is parameterized to reproduce geometries, frequencies, and non-covalent interactions—hence the name GFN.

This series comprises three models introduced chronologically: GFN-xTB, GFN2-xTB, and GFN0-xTB [10, 11, 9]. Among them, GFN2-xTB is the most accurate and complex, approaching the accuracy of density functional theory (DFT) [9]. As this study aims to investigate how closely xTB can approximate DFT-level accuracy, GFN2-xTB was selected. For convenience, all references to "xTB" hereafter refer specifically to GFN2-xTB.

This section outlines the total energy expression used in xTB and details two energy terms—the classical Coulomb repulsion energy and the extended Hückel-type energy—that contain the parameters optimized in this study.

Total Energy

In xTB, the total energy is expressed as [9]:

$$E_{\text{GFN2-xTB}} = E_{\text{rep}} + E_{\text{disp}} + E_{\text{EHT}} + E_{\text{IES+IXC}} + E_{\text{AES}} + E_{\text{AXC}} + G_{\text{Fermi}} \tag{2.3}$$

where: - E_{rep}: classical Coulomb repulsion energy - E_{disp}: dispersion energy - E_{EHT}: extended Hückel-type energy - $E_{\text{IES+IXC}}$: isotropic electrostatics and exchange-correlation energy - E_{AES}:

anisotropic electrostatics - E_{AXC}: anisotropic exchange-correlation energy - G_{Fermi}: Fermi smearing term

Equation 2.4 shows how spatial molecular orbitals ψ_i are formed as a linear combination of Gaussian atomic orbitals $\phi_\kappa(\zeta_\kappa, \mathrm{STO}\text{-}m\mathrm{G})$, which approximate Slater-type orbitals [5]. The molecular orbital coefficients $c_{\kappa i}$ are optimized by solving the following eigenvalue problem:

$$\psi_i = \sum_{\kappa}^{N_{\mathrm{AO}}} c_{\kappa i}\phi_\kappa(\zeta_\kappa, \mathrm{STO}\text{-}m\mathrm{G}) \tag{2.4}$$

This yields the generalized eigenvalue problem [5]:

$$\mathbf{FC} = \mathbf{SC}\boldsymbol{\epsilon} \tag{2.5}$$

where: - $\mathbf{F}$: tight-binding Fock-like matrix - $\mathbf{S}$: overlap matrix between atomic orbitals - $\mathbf{C}$: molecular orbital coefficient matrix - $\boldsymbol{\epsilon}$: diagonal matrix of molecular orbital energies

Self-consistent field (SCF) cycles are performed through this formulation [9].

Extended Hückel-Type Energy

The extended Hückel-type energy (E_{EHT}) captures the primary contribution from covalent bonding in the tight-binding model [11, 9]. As shown in Equation 2.6, it is calculated as the expectation value of the effective one-electron Hamiltonian $\hat{H}_0$ over all occupied molecular orbitals. The occupation number of orbital i is denoted by n_i. Applying the linear combination of atomic orbitals from Equation 2.4, the energy is expressed in terms of the density matrix P and the tight-binding Hamiltonian matrix H:

$$E_{\mathrm{EHT}} = \sum_i n_i\langle\psi_i|\hat{H}_0|\psi_i\rangle = \sum_\kappa\sum_\lambda\sum_i n_i c_{\kappa i}c_{\lambda i}\langle\phi_\lambda|\hat{H}_0|\phi_\kappa\rangle = \sum_\kappa\sum_\lambda P_{\kappa\lambda}H_{\kappa\lambda} \tag{2.6}$$

The off-diagonal Hamiltonian elements are defined in Equation 2.7 [9]. Here, $S_{\kappa\lambda}$ is the orbital overlap integral, ζ are Slater exponents, k_{EN} is a global scaling factor, $\Delta\mathrm{EN}_{AB}^2$ is the squared electronegativity difference between atoms A and B, and $\Pi(R_{AB}, ll')$ is a prefactor dependent on interatomic distance and angular momentum.

The term $k_{ll'}$ is a global empirical scaling factor between orbitals with angular momentum quantum numbers l and l'. Of the five parameters optimized in this study (see section 3.2), four were specific $k_{ll'}$ values for different angular momentum combinations [9].

$$H_{\kappa\lambda} = k_{ll'}\cdot\frac{1}{2}(H_{\kappa\kappa}+H_{\lambda\lambda})\cdot S_{\kappa\lambda}\left(\frac{2\sqrt{\zeta_\kappa\zeta_\lambda}}{\zeta_\kappa+\zeta_\lambda}\right)^{1/2}\left(1+k_{\mathrm{EN}}\Delta\mathrm{EN}_{AB}^2\right)\Pi(R_{AB}, ll')$$
$$(\kappa\in l\in A,\ \lambda\in l'\in B) \tag{2.7}$$

Coulomb Repulsion Energy

The classical Coulomb repulsion energy models the electrostatic interaction between positively charged atomic nuclei. While the classical expression is given in Equation 2.8, xTB introduces short-range damping via empirically fitted parameters, resulting in Equation 2.9 [9]. Here, Y^{eff} and

α are element-specific parameters, and k_{rep} is a global parameter. This study includes k_{rep} as one of the optimized parameters.

$$E_{\text{rep}}^{\text{Coulomb}} = \frac{Z_A Z_B}{R_{AB}} \tag{2.8}$$

$$E_{\text{rep}} = \sum_{AB} \frac{Y_A^{\text{eff}} Y_B^{\text{eff}}}{R_{AB}} \exp\left[-\left(\alpha_A \alpha_B\right)^{1/2} R_{AB}^{k_{\text{rep}}}\right] \tag{2.9}$$

Parameters

The xTB model includes two types of empirically fitted parameters: global and element-specific [11, 9].

Global parameters apply across all atoms and molecules [9]:

1. $k_{ll'}$: EHT scaling factor for orbitals with angular momenta l and l'
2. k_{rep}: Coulomb repulsion scaling factor
3. K_l: shell-specific parameter for E_{IES}
4. k_{EN}: electronegativity scaling factor for E_{EHT}
5. Δ_{val}, R_{max}: multipole scaling factors for E_{AES}
6. a_3, a_5: damping exponents for third- and fifth-order electrostatics in E_{AES}
7. a_1, a_2: dispersion damping parameters
8. s_6, s_8, s_9: dispersion scaling factors

Element-specific parameters are fitted per element up to radon [9]:

1. η_A: atomic Hubbard parameter
2. Γ_A: charge derivative
3. α_A: repulsion exponential scaling parameter
4. Y_A^{eff}: repulsion strength
5. $f_{\text{XC}}^{\mu_A}$, $f_{\text{XC}}^{\Theta_A}$: anisotropic XC scaling parameters
6. R_0^A: AES energy offset radius

Original Fitting Procedure

The original xTB parameters were fitted using the Levenberg-Marquardt algorithm to minimize the root mean square deviation (RMSD) between xTB predictions and reference data [9].

Five types of reference data were used [9]:

1. Equilibrium geometries

2. Distorted geometries (a few kcal/mol above equilibrium)
3. Harmonic vibrational frequencies
4. CM5 atomic charges
5. Noncovalent interaction energies and structures

Atomic force components were included in the fitting for both equilibrium and distorted geometries [9].

The fitting was performed in two stages:

1. Global parameters and element-specific parameters for H, C, N, and O were optimized.
2. Remaining element-specific parameters were fitted while keeping the earlier ones fixed. For lanthanides, only Ce and Lu were explicitly fitted; intermediate elements were linearly interpolated.

Geometries were optimized at the PBEh-3c level of theory, and reference energies were computed using CCSD(T) [9].

The following datasets were used [9]:

1. **HCNO**: 260 molecules with H, C, N, and O atoms (ranging from diatomics to 100 atoms)
2. **S22**: 22 dimers for noncovalent interactions
3. **S66x8**: 66 dimers sampled at 8 intermolecular distances
4. **L7**: 7 large complexes for dispersion interactions
5. **XB18**: 18 small halogen-bonded dimers

2.3 Adaptive Quantum Chemistry

Earlier, I discussed various levels of theory in quantum chemistry, where different methods trade off accuracy for computational cost. These differences typically arise from how each method approximates the electronic wavefunction and treats electron correlation. To manage this trade-off, most methods incorporate parameters fitted to a broad set of molecules.

Rather than designing an entirely new method, the adaptive approach seeks to improve performance by refining the pre-fitted parameters of existing models. The goal is to make these parameters more specific to each target system. To achieve this, machine learning is used to map a system to its optimal parameter set. Once trained, the inference cost of the machine learning model is independent of the system size, so the overall computational scaling of the original quantum chemistry method remains unchanged.

As a result, adaptive methods have the potential to move a method vertically downward on the error vs. computational cost plot (see Figure 2.1), achieving lower error at the same cost [12, 13]. However, this comes at a high upfront cost, as generating improved parameters requires reference calculations using a higher level of theory.

This section explores a related study [13] that developed an adaptive method for tuning the Hartree-Fock (HF) exchange admixture ratio in a hybrid exchange-correlation functional. This prior work served as both the motivation and framework for this thesis.

2.3.1 Adaptive Hybrid Density Functionals

Density Functional Theory (DFT) is a class of quantum mechanical methods that approximates the electronic structure of many-body systems using the electron density rather than the many-electron wavefunction [6]. However, the exact form of the exchange-correlation functional is unknown. As a result, Density Functional Approximations (DFAs) have been developed that balance computational efficiency with accuracy [5].

A notable subclass of DFAs is hybrid density functionals, which mix HF and DFA exchange energies using a parameter $a \in [0, 1]$ [14]:

$$E_{\mathrm{X}}^{\mathrm{hybrid}} = aE_{\mathrm{X}}^{\mathrm{HF}} + (1-a)E_{\mathrm{X}}^{\mathrm{DFA}} \tag{2.10}$$

Traditionally, the value of a is fixed by fitting to a broad benchmark set of molecules and properties. However, multiple studies have shown that system-specific values of a can drastically reduce prediction errors.

In response, the authors of the adaptive hybrid functional study proposed a method to predict the optimal admixture ratio a_{opt} for each molecule. Using the PBE0 functional as a base, they trained a machine learning model to map molecular structure to the optimal a_{opt} [15, 13].

Optimization

The first step involved constructing a training dataset by optimizing the atomization energy with respect to a reference value from a higher-level theory. The loss function minimized the squared error:

$$a_{\mathrm{opt}} = \arg\min_{a} \left(E_{\mathrm{atm.}}^{\mathrm{aPBE0}}(a) - E_{\mathrm{atm.}}^{\mathrm{Ref.}}\right)^2 \tag{2.11}$$

To perform the optimization, a grid search was conducted over evenly spaced values of $a \in [0, 1]$. A quartic polynomial was then fit to the loss values to interpolate the minimum. This yielded an a_{opt} for each molecule [13].

In the case of the QMSpin dataset, the same optimization was applied only to singlet states, while the triplet admixture ratio was fixed at the default PBE0 value of 25%.

Three datasets and corresponding reference methods were used:

1. QM5 + CCSD(T): 1169 molecules
2. W4-17 + FCI/CBS: 200 molecules
3. QMSpin + MRCISD+Q-F12/cc-pVDZ-F12: 1014 randomly selected singlet carbenes

The resulting a_{opt} distributions showed distinct means across datasets but were normally distributed within each dataset. This highlights the system specificity of the optimal admixture ratio [13].

Machine Learning

With the dataset constructed, a machine learning model was trained to predict a_{opt} from molecular structure [13]. Given the small dataset size, kernel ridge regression (KRR) was chosen for its simplicity and robustness [16].

Among various molecular representations, cMBDF was found to yield the best performance and was used in the final model [17, 13]. The authors selected a local Gaussian kernel to capture atomic contributions separately. The kernel computes the similarity between two systems I and J as [13]:

$$C(\mathbf{X}_I, \mathbf{X}_J) = \sum_{\mu \in I} \sum_{\nu \in J} \delta_{Z_\mu, Z_\nu} \exp\left(-\frac{||\mathbf{X}_{I\mu} - \mathbf{X}_{J\nu}||_2^2}{2\sigma^2} \right) \tag{2.12}$$

Here, $\mathbf{X}$ is a 2D molecular representation where each row encodes an atomic environment. The predicted a_{opt} for a test system I is given by [13]:

$$a_{\text{opt}} = \sum_{J}^{N_{\text{train}}} \alpha_J C(\mathbf{X}_I, \mathbf{X}_J) \tag{2.13}$$

The regression weights α_J are obtained by solving the regularized KRR system [16, 13]:

$$\boldsymbol{\alpha} = (\mathbf{K}^{\text{train}} + \lambda \mathbf{I})^{-1} \mathbf{y}^{\text{train}} \tag{2.14}$$

Results

Across several chemically relevant tasks, aPBE0 significantly reduced errors compared to standard PBE0 [13].

In a follow-up experiment, the same machine learning pipeline was used to directly predict atomization energies (i.e., observables), rather than predicting a_{opt}. This direct approach performed significantly worse, emphasizing the value of adaptive parameter tuning. In particular, learning observables directly requires the model to approximate all the underlying quantum mechanical behavior, while the adaptive method leverages the structure already embedded in DFT [13].

Finally, the authors compared different molecular representations using the same local kernel ridge regression model. Although no representation plateaued in performance, cMBDF consistently showed strong results. The lack of convergence in the learning curves suggested that additional data could further improve performance across all representations [13].

Chapter 3

Methods

3.1 Reference Data

While the original xTB parameterization used a total of 373 unique systems, this dataset was too small to train a machine learning model capable of effectively learning the optimal parameter surface across chemical space. Moreover, the chemical diversity within that dataset was relatively limited. The largest subset, the HCNO internal dataset, contained only 260 molecules ranging from diatomics to systems with up to 100 atoms [9].

As observed in the adaptive hybrid functional study, test loss continued to decrease even with 800 training examples [13]. Importantly, that study optimized a single parameter $a \in [0, 1]$, whereas adaptive xTB requires optimization over five parameters, each spanning a range greater than one. This makes the adaptive xTB parameter space at least five times larger and correspondingly more difficult to explore. Consequently, I turned to a more suitable dataset: VQM24.

VQM24 is a quantum mechanical dataset consisting of neutral, closed-shell organic molecules with up to five heavy atoms selected from the elements C, N, O, F, Si, P, S, Cl, and Br. Of particular relevance to this study, it provides geometries and energies computed at the ωB97X-D3/cc-pVDZ hybrid DFT level of theory [18]. To ensure uniqueness and avoid conformational redundancy, I used the subset of VQM24 containing only the lowest-energy conformer for each molecule.

Compared to the original xTB training set, the VQM24 subset used in this study contains 10,739 molecules and is much denser in chemical compound space. This increased density makes it far more suitable for training machine learning models, improving the model's ability to generalize across chemical diversity.

In addition, small molecules offer several practical benefits [19]:

- They are computationally efficient to process and simulate.
- Many energy-related properties are extensive, so training on small molecules can support extrapolation to larger systems.

To manage computational cost during the early stages of experimentation, a random subset of 100 molecules from VQM24 was used for all preliminary investigations. This includes parameter selection, comparisons between optimization objectives, selection of optimization settings, and

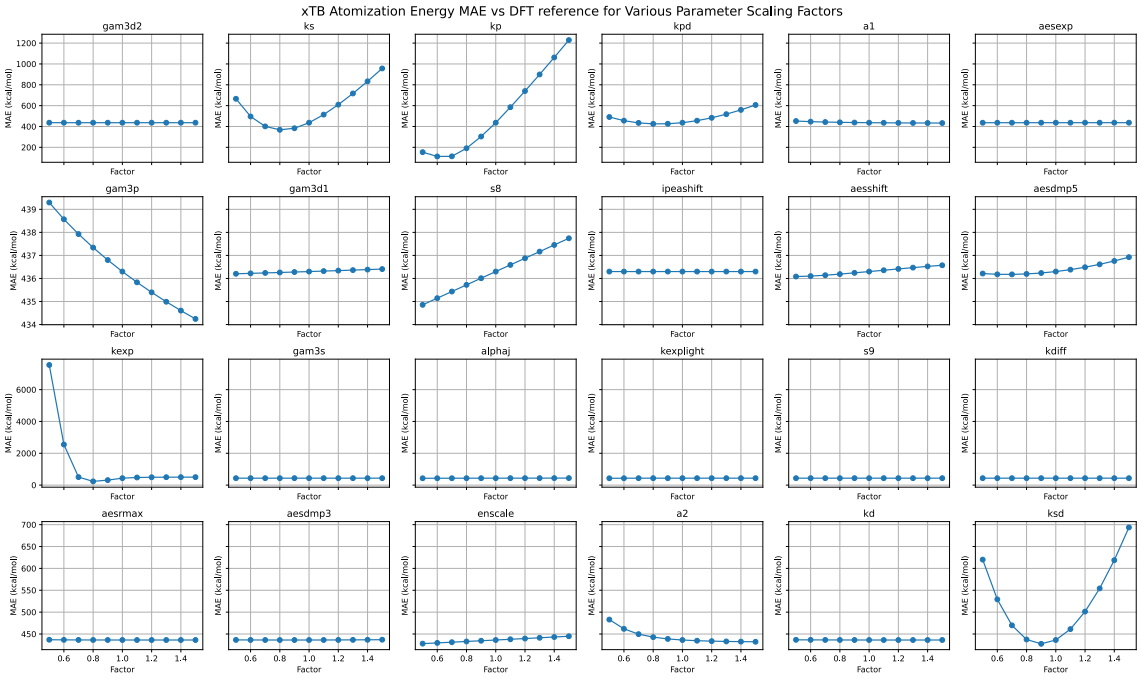

Figure 3.1: One-dimensional parameter scans of maximum atomic force MAE versus parameter scaling factor. For each parameter, all other parameters were held fixed at their default values. Results are averaged over 100 randomly sampled molecules (lowest-energy conformers) from VQM24.

hyperparameter tuning. This smaller subset served as a computationally efficient proxy prior to full-scale dataset generation.

3.2 Parameter Selection

The xTB method consists of numerous global and element-specific parameters. Simultaneously optimizing all of them would require searching a prohibitively high-dimensional space (over 100 dimensions). To keep the project tractable, I scoped the study to focus on global parameters, as these are shared across all systems in the dataset. This choice made the optimization process more data-efficient and computationally feasible.

To further improve efficiency, it was necessary to identify a promising subset of global parameters to tune. The goal was to find those parameters that most strongly influence the chosen optimization objective (see section 3.3).

To that end, I performed one-dimensional scans for each global parameter. In each scan, a single parameter was varied while all others were held at their default values. Each parameter was scaled over a uniform range from 70% to 130% of its default value. Figure 3.1 shows the results for one representative optimization objective. From these scans, it was evident that only five parameters—k_{ss}, k_{pp}, k_{sd}, k_{pd}, and k_{rep}—had a substantial impact on the objective.

These findings were consistent across all optimization objectives tested. As a result, this same set of five parameters was selected for tuning in all subsequent experiments.

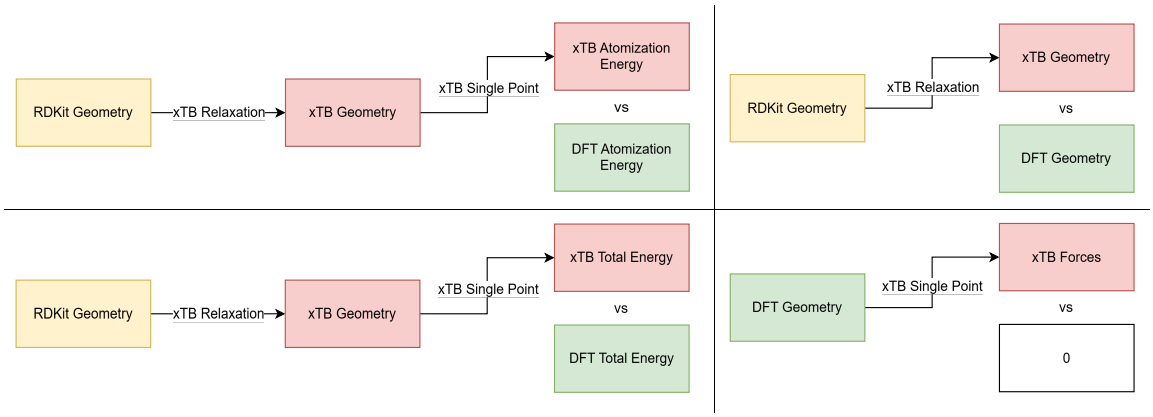

Figure 3.2: Workflows for the four optimization objectives evaluated. Subplots (a)-(d), arranged left to right, top to bottom: (a) atomization energy, (b) displacement, (c) total energy, and (d) forces. In (a) and (c), RDKit geometries were relaxed with xTB, followed by single-point calculations and comparison against DFT reference energies. In (b), RDKit geometries were relaxed with xTB and compared directly to DFT geometries. In (d), xTB forces were computed on DFT geometries and compared against zero.

3.3 Optimization Objective

Our search for a suitable optimization objective focused on two major categories: energy-based objectives and force-based (geometry-related) objectives. Details of the optimization procedure are provided in section 3.4.

3.3.1 Energy-Based Objectives

Because xTB was originally parameterized to reproduce geometries, vibrational frequencies, and non-covalent interactions, it is known to perform poorly on absolute energy-related quantities [11, 9]. Nonetheless, I initially explored energy-based objectives with the aim of adapting xTB parameters to improve performance on such tasks.

Figure 3.2 outlines the workflow used. RDKit generated geometries were created from InChI strings, relaxed with xTB, and then used in single-point calculations to compute target properties.

I first examined the atomization energy, using mean absolute error (MAE) as the loss metric:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i}^{N}\left|E_i^{\mathrm{xTB}} - E_i^{\mathrm{VQM24}}\right| \tag{3.1}$$

Following the same procedure used in parameter selection, one-dimensional parameter scans were conducted to measure how each parameter affected atomization energy MAE. Figure 3.3 shows the scan for k_{pp}, the parameter yielding the lowest minimum. For each scaling factor, both the best- and worst-performing molecules were recorded. Despite achieving a minimum MAE, the worst molecule still exhibited extremely large errors—up to approximately 300 kcal/mol.

To eliminate the possibility that these errors stemmed from free atom energy calculations, I next considered the total molecular energy as an objective. However, this approach also resulted in extremely large errors. This was expected, as noted by the original authors of xTB in a repository post: strong energy accuracy would require extensive re-parameterization. This limitation is inherent

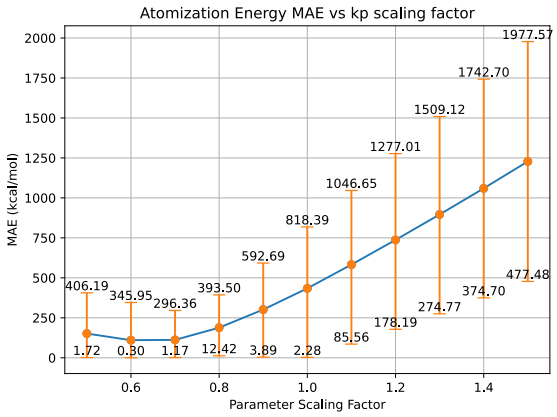

Figure 3.3: One-dimensional parameter scan of atomization energy MAE versus the scaling factor for k_{pp}. The blue line shows the mean MAE across 100 randomly selected VQM24 molecules, while the orange bars denote the range (minimum to maximum) within the sample.

to many semi-empirical methods—the simplifications made to the underlying quantum mechanical equations cause the fitted parameters to become specialized for the observables used during training. As a result, these parameters may not generalize well to other properties, such as absolute energies.

3.3.2 Geometry-Based Objectives

Given the limitations observed with energy-based objectives, I pivoted to geometry-based objectives—quantities that xTB was originally designed to predict.

Two approaches were considered: displacement-based and force-based objectives. The displacement-based metric compared relative atom-atom distances between xTB-relaxed geometries and DFT geometries. As shown in Figure 3.2, RDKit geometries were first relaxed with xTB and then aligned to their corresponding DFT geometries. To ensure rotational and translational invariance, pairwise distance matrices were computed for both geometries and compared using the Frobenius norm:

$$\Delta_{\text{dist}} = \left\| D^{\text{xTB}} - D^{\text{DFT}} \right\|_F = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} \left(D_{ij}^{\text{xTB}} - D_{ij}^{\text{DFT}} \right)^2} \tag{3.2}$$

The force-based objective sought to align the energy minima of xTB with those of DFT. As illustrated in Figure 3.2, xTB forces were computed on DFT-optimized geometries. The loss was defined as the maximum atomic force:

$$F_{\max} = \max_{i \in \{1,\ldots,N\}} \|\mathbf{F}_i\|_2 = \max_i \sqrt{F_{ix}^2 + F_{iy}^2 + F_{iz}^2} \tag{3.3}$$

This quantity directly measures how far the xTB energy surface deviates from the DFT equilibrium. If a large force is present on any atom, it implies that xTB would drive the system away from the DFT minimum during relaxation. Minimizing this quantity encourages the xTB energy landscape to more closely resemble that of DFT in the vicinity of equilibrium geometries.

3.4 Optimization

To optimize xTB parameters, a black-box optimization approach was employed. Due to the complexity of the total energy expression, analytical gradients were intractable. Moreover, the optimization landscape exhibited discontinuities, as xTB occasionally failed to converge for certain parameter combinations. These failures made finite-difference gradient estimation unreliable, ruling out gradient-based methods.

For such settings, Bayesian optimization is a widely used and effective global optimization method [16]. Given parameter bounds, Bayesian optimization constructs a probabilistic surrogate model of the objective function and iteratively selects new evaluation points by maximizing an acquisition function that balances exploration and exploitation [16].

3.4.1 Tree-Structured Parzen Estimator

The Tree-Structured Parzen Estimator (TPE) is a variant of Bayesian optimization that is particularly robust to discontinuities in the objective function [20]. Unlike traditional Bayesian optimization approaches that use Gaussian processes to model the objective function directly, TPE models the distribution of parameters conditioned on the objective function value [20].

Specifically, given observed values y, the dataset is partitioned into two subsets based on a quantile threshold y^γ [20]:

$$p(\mathbf{x} \mid y, \mathcal{D}) := \begin{cases} p(\mathbf{x} \mid \mathcal{D}^{(l)}) & \text{if } y \leq y^\gamma \\ p(\mathbf{x} \mid \mathcal{D}^{(g)}) & \text{if } y > y^\gamma \end{cases} \tag{3.4}$$

Here, $\mathcal{D}^{(l)}$ and $\mathcal{D}^{(g)}$ denote the subsets of the data corresponding to "good" (low loss) and "bad" (high loss) objective values, respectively. Kernel density estimates are then constructed for each subset [20]:

$$p(x \mid \mathcal{D}^{(l)}) = w_0^{(l)}\, p_0(x) + \sum_{n=1}^{N^{(l)}} w_n\, k(x, x_n \mid b^{(l)}), \tag{3.5}$$

$$p(x \mid \mathcal{D}^{(g)}) = w_0^{(g)}\, p_0(x) + \sum_{n=N^{(l)}+1}^{N} w_n\, k(x, x_n \mid b^{(g)}), \tag{3.6}$$

where w are weights and k are kernel functions. In this work, I used Gaussian kernels with uniform weights, resulting in a sum of IID Gaussians centered at each observed point. Because this approach models only the distribution of x and not the magnitude of y, it remains robust even in the presence of discontinuities.

To select the next parameter set to evaluate, candidates are sampled from $p(x \mid \mathcal{D}^{(l)})$, and the point that maximizes the following ratio is chosen [20]:

$$\mathbb{P}(y \leq y^\gamma \mid \mathbf{x}, \mathcal{D}) \overset{\text{rank}}{\simeq} r(\mathbf{x} \mid \mathcal{D}) := \frac{p(\mathbf{x} \mid \mathcal{D}^{(l)})}{p(\mathbf{x} \mid \mathcal{D}^{(g)})}. \tag{3.7}$$

Intuitively, this corresponds to selecting a parameter set that is most likely to belong to the region of low objective values, while being unlikely to belong to the region of high objective values.

3.5 Machine Learning

In this project, I build supervised learning models of the form:

$$f_\theta(\mathbf{X}_I) = \mathbf{y}_I \tag{3.8}$$

to predict optimal parameter sets $\mathbf{y}_I$ for a molecule represented by $\mathbf{X}_I$. These predictions are made by fitting a parametric statistical model f_θ to the training data. The feature vector $\mathbf{X}_I$ is generated by a representation function r that maps a molecule's nuclear charges and Cartesian coordinates $\{Z_i, \mathbf{R}_i\}_{i\in I}$ to a vector in $\mathbb{R}^D$:

$$\mathbf{X}_I = r(\{Z_i, \mathbf{R}_i\}_{i\in I}) \in \mathbb{R}^D \tag{3.9}$$

The choice of regressor f_θ and representation r is critical to model performance. In this section, I describe the regressors and representations explored in this study.

3.5.1 Regressors

Kernel Ridge Regression

Similar to the KRR model used in adaptive hybrid density functionals, I employed a Gaussian kernel [13]. However, instead of predicting a single scalar, I simultaneously trained on five parameters:

$$\begin{bmatrix} k_{ss}^{\text{opt}} \\ k_{pp}^{\text{opt}} \\ k_{sd}^{\text{opt}} \\ k_{pd}^{\text{opt}} \\ k_{\text{rep}}^{\text{opt}} \end{bmatrix} = \sum_{J=1}^{N_{\text{train}}} \boldsymbol{\alpha}_J \, C(\mathbf{X}_I, \mathbf{X}_J) \tag{3.10}$$

where $\boldsymbol{\alpha}_J \in \mathbb{R}^5$.

Although Equation 3.10 appears to model all targets jointly, each component is learned independently; i.e., this is mathematically equivalent to training five separate one-output KRR models. Each component of $\boldsymbol{\alpha}_J$ is only updated with respect to its corresponding target.

XGBoost

Extreme Gradient Boosting (XGBoost) is a tree-based ensemble method that builds an additive model of T regression trees to predict scalar targets [21]. Given a dataset of N examples $\{\mathbf{X}_i, y_i\}_{i=1}^N$, predictions are computed as [21]:

$$\hat{y}_q = \sum_{t=1}^{T} f_t(\mathbf{X}_q), \quad f_t \in \mathcal{F} \tag{3.11}$$

where each f_t is a regression tree drawn from the function space $\mathcal{F}$, which defines trees with fixed maximum depth.

Training proceeds by minimizing the regularized objective [21]:

$$\mathcal{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{t=1}^{T} \Omega(f_t), \tag{3.12}$$

where ℓ is a loss function (e.g., squared error), and $\Omega(f_t)$ penalizes tree complexity [21]:

$$\Omega(f) = \gamma T_{\text{leaves}} + \frac{1}{2}\lambda \sum_{j=1}^{T_{\text{leaves}}} w_j^2, \tag{3.13}$$

with w_j being the prediction for leaf j, and T_{leaves} the number of leaves.

At each step, a new tree is added to minimize a second-order Taylor approximation of the objective [21]:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{N} \left[g_i f_t(\mathbf{X}_i) + \frac{1}{2} h_i f_t(\mathbf{X}_i)^2 \right] + \Omega(f_t), \tag{3.14}$$

where g_i and h_i are the first and second derivatives of the loss with respect to the prediction [21]:

$$g_i = \frac{\partial \ell(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 \ell(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2}$$

Each leaf aggregates gradients and Hessians to compute the optimal prediction:

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{3.15}$$

where I_j is the set of samples in leaf j.

In our setup, one XGBoost regressor was trained per parameter, similar to KRR. For a molecule represented by $\mathbf{X}_I$, the predicted output vector is:

$$f_\theta(\mathbf{X}_I) = \begin{bmatrix} k_{ss}^{\text{opt}} \\ k_{pp}^{\text{opt}} \\ k_{sd}^{\text{opt}} \\ k_{pd}^{\text{opt}} \\ k_{\text{rep}}^{\text{opt}} \end{bmatrix} \tag{3.16}$$

3.5.2 Molecular Representations

For kernel and tree-based models (KRR, XGBoost), input molecules must be represented as fixed-length vectors $\mathbf{X}_I \in \mathbb{R}^D$, derived from atomic positions and charges [21, 22]. A mapping r defines this transformation:

$$\mathbf{X}_I = r(\{Z_i, \mathbf{R}_i\}_{i \in I}) \in \mathbb{R}^D \tag{3.17}$$

An effective representation should meet several criteria:

- **Uniqueness:** Distinct molecular configurations must yield distinct representations.
- **Symmetry Invariance:** Representations should be invariant to translation, rotation, and atom index permutation.

- **Differentiability:** The mapping should be differentiable with respect to atomic positions, allowing force prediction.
- **Compactness:** Dimensionality should be minimized to reduce memory and computational cost.

Several handcrafted descriptors satisfy these properties, including the Coulomb Matrix (CM) [23], Bag of Bonds (BOB) [24], Smooth Overlap of Atomic Positions (SOAP) [25], Atom-Centered Symmetry Functions (ACSF) [26], Faber-Christensen-Huang-Lilienfeld (FCHL) [27], and Axilrod-Teller-Muto potentials (SLATM) [28].

In this work, I explore both global and local representations. In local representations, each atom i is encoded as a vector $\mathbf{x}_{iI} \in \mathbb{R}^D$, and the molecule is:

$$\mathbf{X}_I = [\mathbf{x}_{iI}]_{i \in I} \in \mathbb{R}^{n \times D} \tag{3.18}$$

with n being the number of atoms. These embeddings may be pooled to produce global representations (e.g., via summation).

Training Setup. For hyperparameter tuning, a random subset of 100 molecules from VQM24 was used with an 80%/20% train-validation split. Model selection used 5-fold cross-validation. For testing, 5-fold cross-validation was again used with five disjoint sets of 200 test molecules and the remainder used for training. See Appendix C for hyperparameter details.

3.6 Density Functional Theory (DFT)

A key motivation for tuning xTB parameters is to accelerate or potentially replace geometry optimizations performed at higher levels of theory, such as Density Functional Theory (DFT). DFT is one of the most widely used quantum mechanical methods due to its favorable trade-off between accuracy and computational cost. It reformulates the many-body electronic structure problem in terms of the electron density $\rho(\mathbf{r})$ rather than the many-electron wavefunction, thereby significantly reducing the problem's dimensionality [6, 19].

While DFT is, in principle, an exact theory, in practice it relies on approximations to the unknown exchange-correlation functional [6]. In this work, I adopted the same level of theory as that used in the VQM24 dataset: the ωB97X-D3 functional with the cc-pVDZ basis set [13]. This choice combines a range-separated hybrid functional with D3 dispersion corrections and a correlation-consistent double-zeta basis, offering a reliable balance of accuracy and efficiency for a broad range of molecules [29].

The goal of tuning xTB was twofold:

1. **Reduce the number of DFT SCF cycles:** By aligning xTB's energy minima more closely with those of DFT, geometries optimized with xTB should begin closer to the DFT minimum, requiring fewer SCF iterations during refinement.
2. **Improve first-pass convergence in geometry optimization:** While most VQM24 molecules converged within 100 geometry optimization steps, some required a second optimization pass. Improved xTB initial structures could reduce or eliminate the need for such second passes.

To evaluate these goals, all DFT calculations were performed using `Psi4` [30], employing the same level of theory as in VQM24 to ensure consistency:

$$\texttt{method} = \omega\texttt{B97X-D3}, \quad \texttt{basis} = \texttt{cc-pVDZ}$$

By improving the accuracy of semi-empirical starting points, adaptive xTB offers a promising route to reducing the computational cost of large-scale quantum mechanical simulations.

3.6.1 GFN-FF

In addition to RDKit-generated geometries—which are initialized using heuristics and static bond-length look-up tables—I employed GFN-FF to improve the quality of initial molecular geometries prior to xTB and DFT calculations [31].

GFN-FF (Geometry, Frequency, Non-covalent, Force Field) is a general-purpose, quantum mechanically derived force field. Unlike classical force fields, which are often limited by empirical parameterizations tailored to specific chemistries, GFN-FF is built from first-principles-inspired approximations and is designed for broad chemical coverage. It aims to provide high-quality geometries and vibrational frequencies across organic, inorganic, and organometallic compounds, without the need for system-specific reparameterization [32].

The key motivations for using GFN-FF in this work were:

1. **Improved initialization over RDKit:** RDKit geometries, while fast to generate, are often far from equilibrium and can land the system in a different basin of the potential energy surface (PES). This mismatch can mislead both the optimization and the downstream DFT convergence behavior.

2. **Cost-effective pre-relaxation:** GFN-FF is orders of magnitude faster than semi-empirical methods like xTB or ab initio methods like DFT, yet it provides a significantly better approximation to equilibrium geometry than RDKit [32].

3. **Stabilizing xTB optimization:** When RDKit geometries are poorly initialized, xTB relaxation may diverge or settle in poor local minima. Using GFN-FF to pre-optimize the structure often results in more physically reasonable input geometries for xTB, improving reliability and reproducibility.

Chapter 4

Results & Discussion

4.1 Maximum Atomic Forces with Geometry-Based Objectives

To assess the performance of xTB with geometry-based optimized parameters, single-point energy calculations were performed on DFT-optimized geometries using both sets of optimized parameters.

Figure 4.1 shows the distribution of maximum atomic forces computed with the default xTB parameters versus those optimized using the force-based objective. As expected—since this objective directly minimizes maximum atomic force—the optimized parameters significantly reduce the average force, from 19.12 kcal/mol/bohr to 5.5 kcal/mol/bohr.

In contrast, Figure 4.2 compares maximum atomic forces computed with default parameters against those from the displacement-optimized parameters. The results are considerably worse: many molecules exhibit a substantial increase in maximum atomic force, with several exceeding 100 kcal/mol/bohr. These extreme outliers are visualized individually. Their poor performance may stem from being dispersion-bound or containing silicon atoms, a known limitation of xTB accuracy [33].

Beyond analyzing only the maximum force per molecule, Figure 4.3 shows the distribution of atomic force components (x, y, and z) across all atoms. These plots confirm that the improvements from force-based optimization are consistent across directions, and not biased along any particular coordinate axis. In contrast, displacement-optimized parameters not only fail to reduce forces but also produce significantly more outliers. Force-optimized parameters reduce both the mean and the tail of the distribution.

4.2 Analyzing Optimal Parameters

The distribution of optimal parameters is shown in Figure 4.4. Similarly to the optimal admixture ratio distributions in aPBE0 [13], distributions for k_{ss}, k_{pp}, and k_{pd} are roughly normally distributed with peaks/means different than the default value. This indicates that the optimization process successfully found optimal parameters different than the default. The consistent shift away from the default suggests that the default values may be suboptimal for the systems considered, and that tailoring these parameters on a per-molecule or per-dataset basis can lead to improved accuracy.

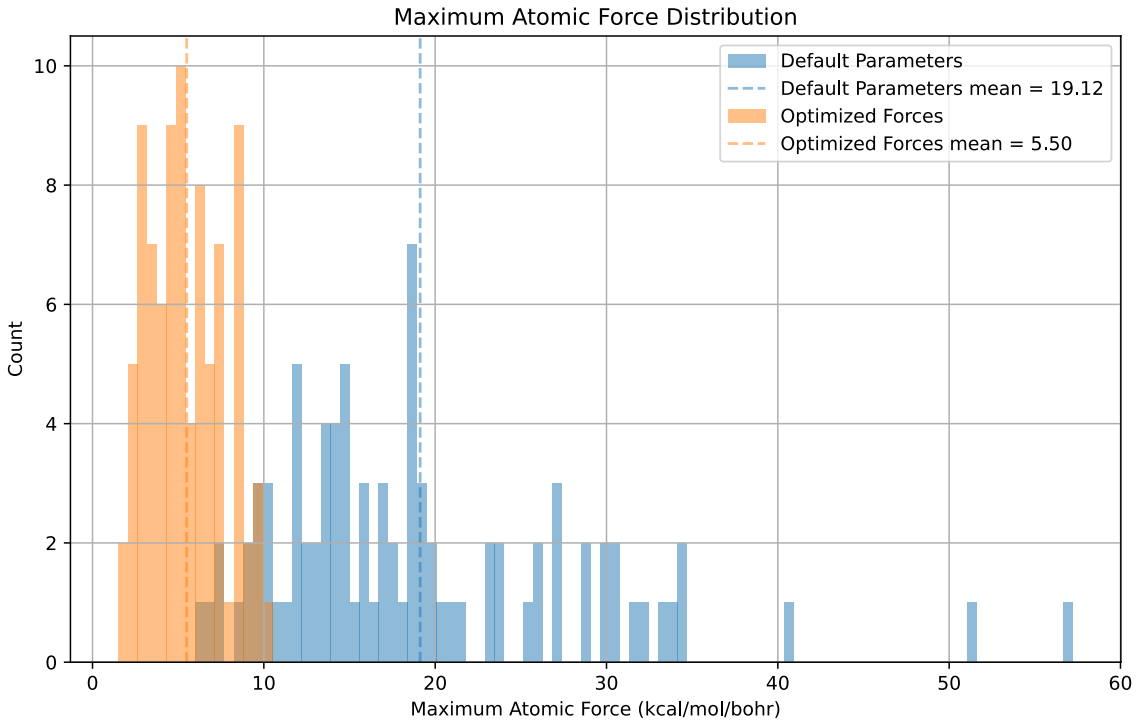

Figure 4.1: Distribution of maximum atomic forces from single-point calculations on DFT geometries using default xTB parameters vs. force-optimized xTB parameters (Equation 3.3).

With Figure 4.5, we can assess how the parameters behave jointly. While the EHT scaling factors $k_{ll'}$ exhibit low mutual correlation, there is significant correlation between k_{ss} and k_{rep}, as well as between k_{pp} and k_{rep}. This suggests that changes in the Coulomb repulsion scaling factor k_{rep} may influence or compensate for changes in the EHT parameters, particularly those involving s- and p-orbital interactions. Such parameter interactions highlight the importance of jointly optimizing parameters rather than treating them as fully independent.

Additionally, this further motivates the use of machine learning methods that model parameter dependencies—such as message passing neural networks—over methods that predict each parameter independently, such as KRR or XGBoost.

4.3 Effect on DFT Compute Cost

Figure 4.6 compares the number of DFT self-consistent field (SCF) cycles required to converge geometries starting from xTB-relaxed structures using either the default or optimized force parameters.

In the first experiment, I relaxed RDKit-generated geometries using xTB (default or optimized), followed by DFT relaxation. Contrary to expectations, the optimized parameters often resulted in an increase in the number of DFT cycles.

To probe this behavior, a second experiment replaced RDKit geometries with those pre-relaxed using GFN-FF before applying xTB and DFT. The rationale was that poor RDKit geometries may land the system in a different potential energy basin than the one targeted by the optimized parameters. If true, the optimizer might steer geometries toward local minima that diverge from

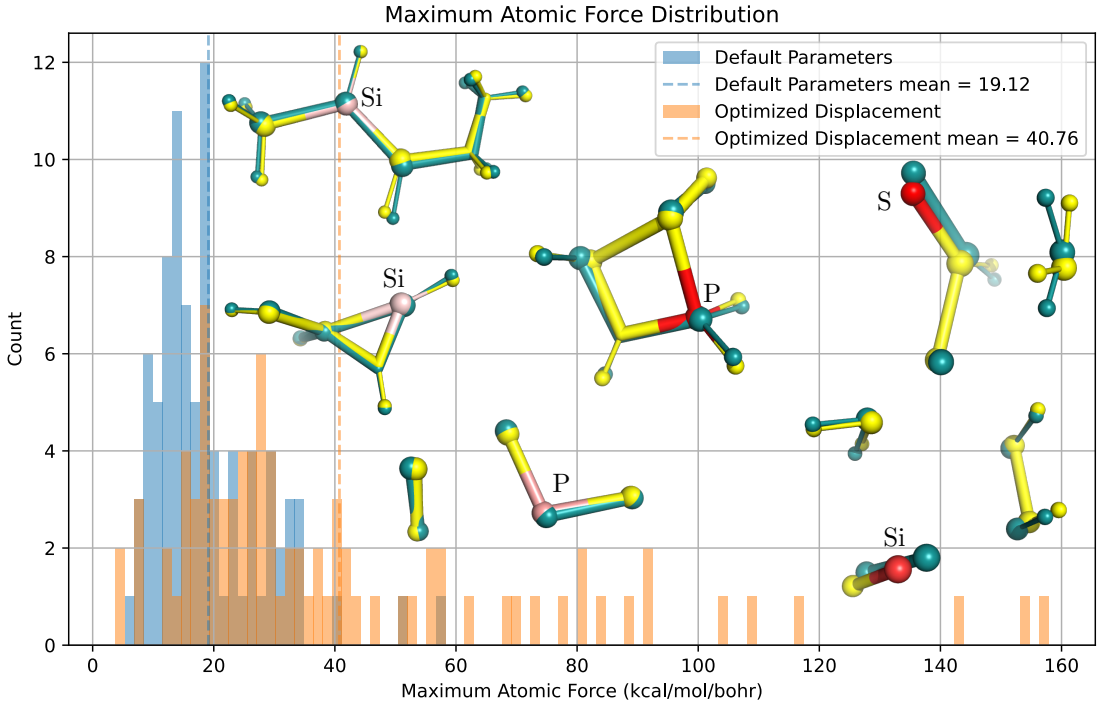

Figure 4.2: Distribution of maximum atomic forces from single-point calculations on DFT geometries using default xTB parameters vs. displacement-optimized xTB parameters. Molecules with maximum atomic forces exceeding 100 kcal/mol/bohr under the optimized parameters are visualized individually. Yellow indicates the reference DFT geometry; teal shows the relaxed xTB geometry (initialized from the RDKit structure [31]). The atom with the maximum force is highlighted in red, with intensity corresponding to force magnitude.

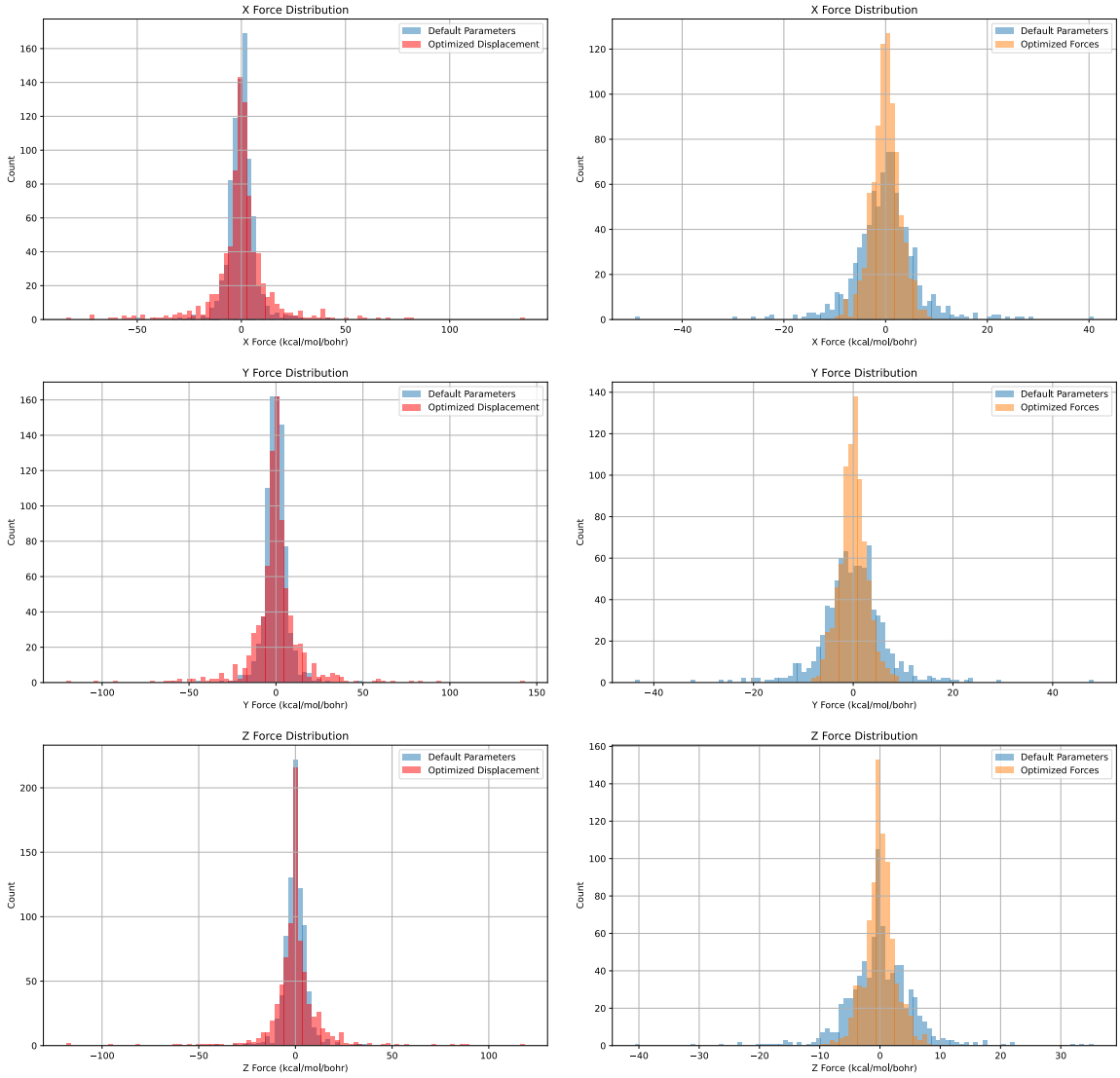

Figure 4.3: Distribution of atomic force components (x, y, and z) computed using various xTB parameter sets on DFT-optimized geometries. The left panel compares default parameters with displacement-optimized ones; the right panel compares default with force-optimized parameters.

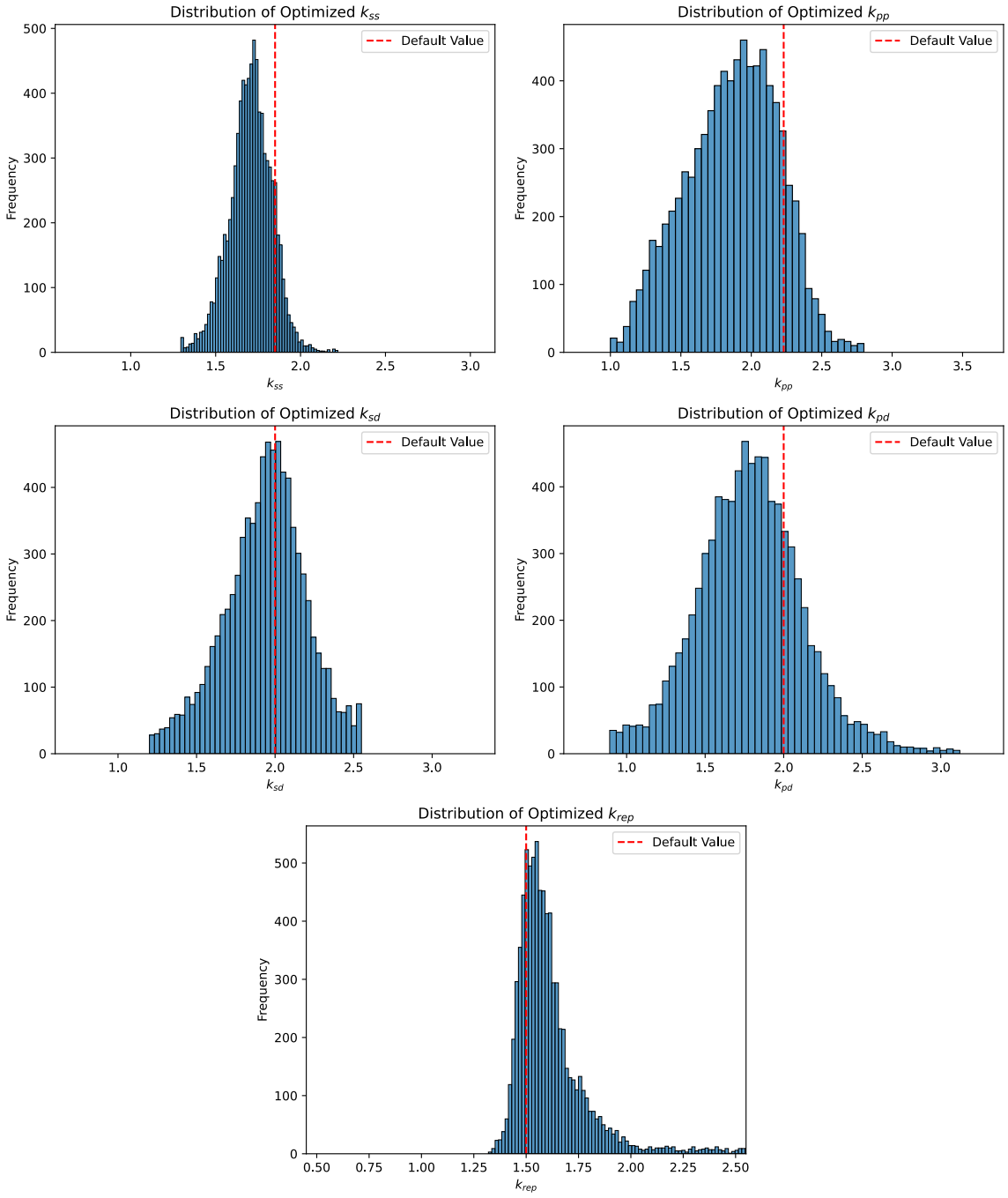

Figure 4.4: Distribution of maximum atomic force optimized parameters. (a)-(e) Left to right, top to bottom: (a) k_{ss} (b) k_{pp} (c) k_{sd} (d) k_{pd} (e) k_{rep}

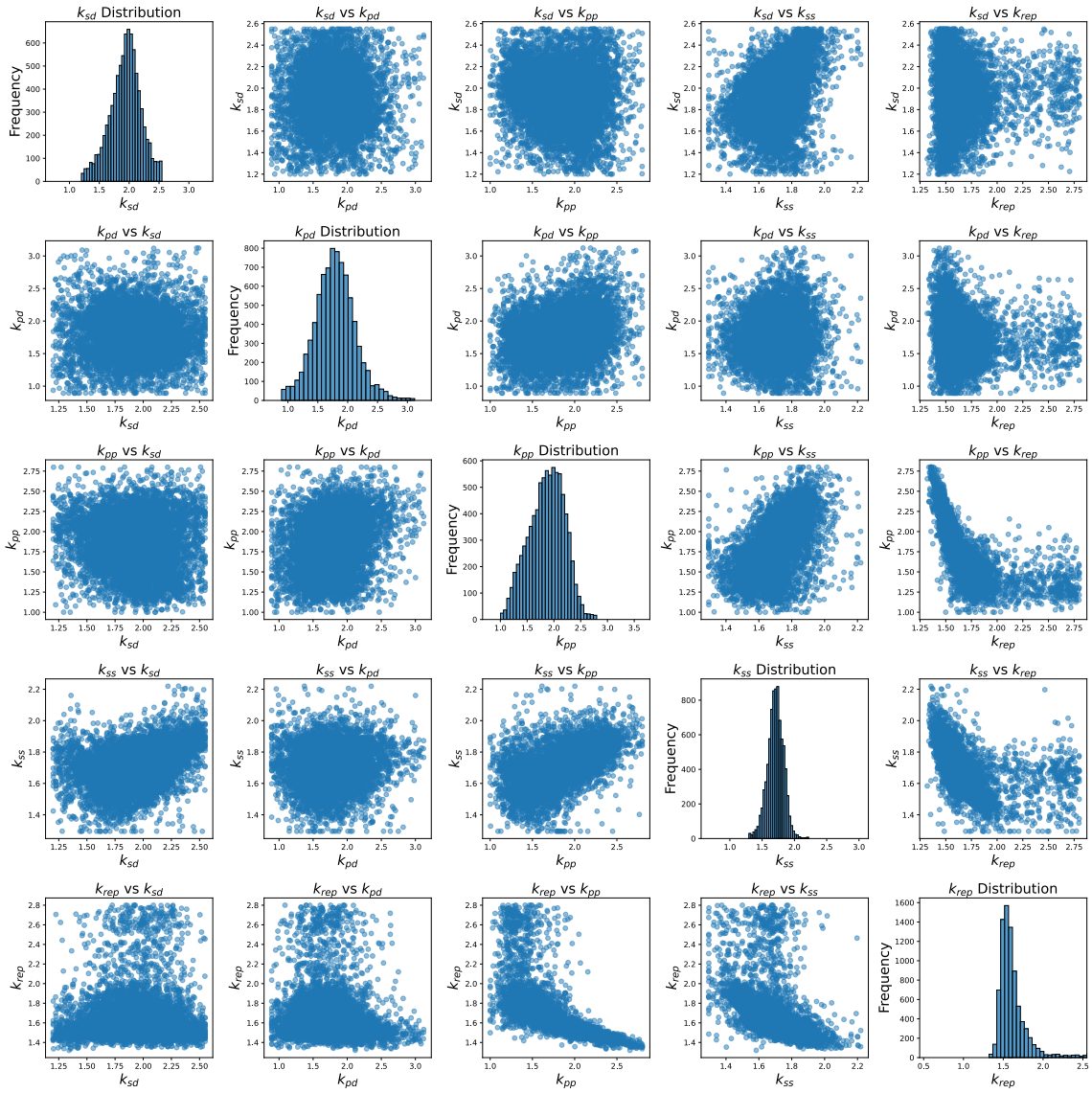

Figure 4.5: Distributions and scatter plots of each pair of parameter distributions. Each row corresponds to a parameter on the y-axis and each column corresponds to a parameter on the x-axis. For on-diagonal plots, the 1D distribution of the parameter is shown. For off-diagonal plots, the 2D scatter plot is shown.

those favored by DFT, reducing the utility of the optimized parameters. However, even with better initialization from GFN-FF, the optimized parameters failed to reduce DFT cost consistently.

To completely eliminate basin mismatch, a third experiment started from DFT-optimized geometries, which were relaxed with xTB (default or optimized) and then re-relaxed with DFT. This ensured that both xTB and DFT operated within the same local region of the potential energy surface. In this idealized setting, optimized parameters should produce smaller deviations from DFT geometries, requiring fewer SCF cycles to reconverge. Surprisingly, even here, the default parameters generally outperformed the optimized ones.

These results suggest that minimizing the maximum atomic force at the end of xTB relaxation is not a good surrogate for reducing DFT relaxation effort.

To further investigate, I examined whether molecules that experienced the largest reduction in maximum atomic force also required fewer DFT cycles. Figure 4.7 plots the change in maximum atomic force (relative to default) against the change in DFT SCF cycles. While the first two molecules showed a strong correlation—achieving both lower forces and lower DFT cost—this trend broke down for the rest of the dataset. Overall, there was no consistent relationship.

Still, one point of optimism is the molecule `CPxSi3H7`, which achieved a perfect outcome: it required zero DFT cycles after relaxation with optimized xTB parameters.

4.4 Machine Learning

Figure 4.8 shows the learning curves for each parameter across a range of regressors and molecular representations.

No single model-representation pair consistently outperforms others across all five parameters. Instead, performance appears to vary depending on the parameter being predicted. This variability suggests that each parameter may require different inductive biases or may be sensitive to different features of the molecular representation. It also highlights the challenge of building a unified model capable of learning the full parameter vector accurately.

Although model performance improves steadily with increasing training set size (up to 6400 molecules), prediction errors remain large relative to the label scale (approximately 1-3x). This is particularly evident in Figure 4.9, which plots the downstream performance of learned parameters. A substantial gap remains between the maximum atomic force achievable with the learned parameters and that obtained using the ground-truth labels, even as the training set grows. This indicates that current regressors have not yet captured the complex, nonlinear mapping from molecular structure to optimal parameter values.

This gap underscores several limitations. First, it suggests that the parameter landscape is difficult to learn directly, particularly with the limited supervision available—each molecule is annotated with just one optimal parameter set. Second, it raises the question of whether learning optimal parameters directly is the right approach. Instead, a more scalable strategy may be to learn the relationship between parameters and forces, enabling training on suboptimal parameter-force pairs. Such a model could then be used at inference time to predict the parameters that would produce minimal forces—effectively performing parameter optimization via a learned surrogate.

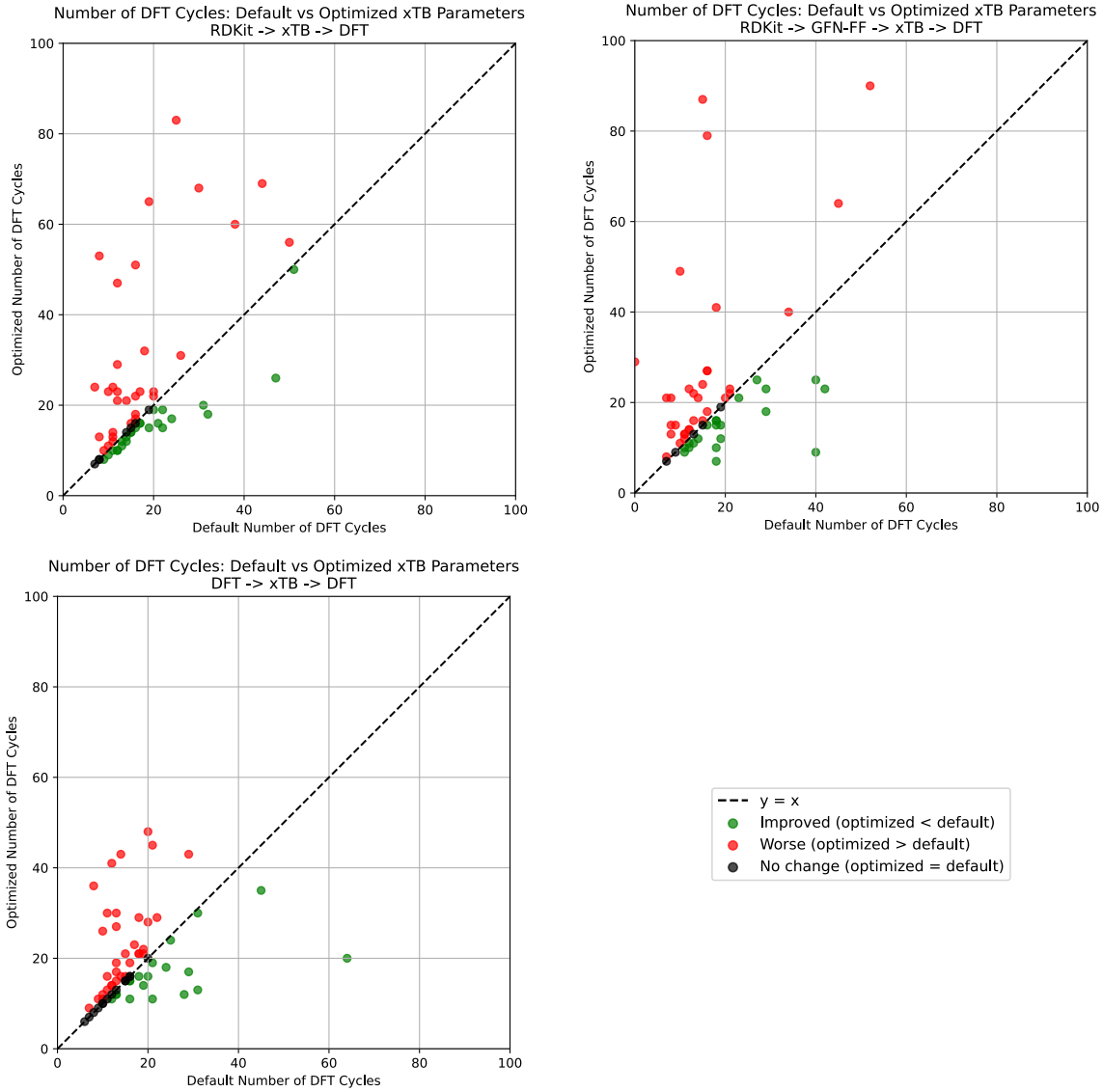

Figure 4.6: DFT cost comparisons between relaxed xTB geometries generated using optimized versus default force parameters. Points above $y = x$ indicate more DFT cycles were required to converge compared to the default; points below indicate fewer cycles. Three experiments are shown, left to right, top to bottom: (1) RDKit → xTB → DFT relaxation, (2) RDKit → GFN-FF → DFT relaxation, (3) DFT → xTB → DFT relaxation.

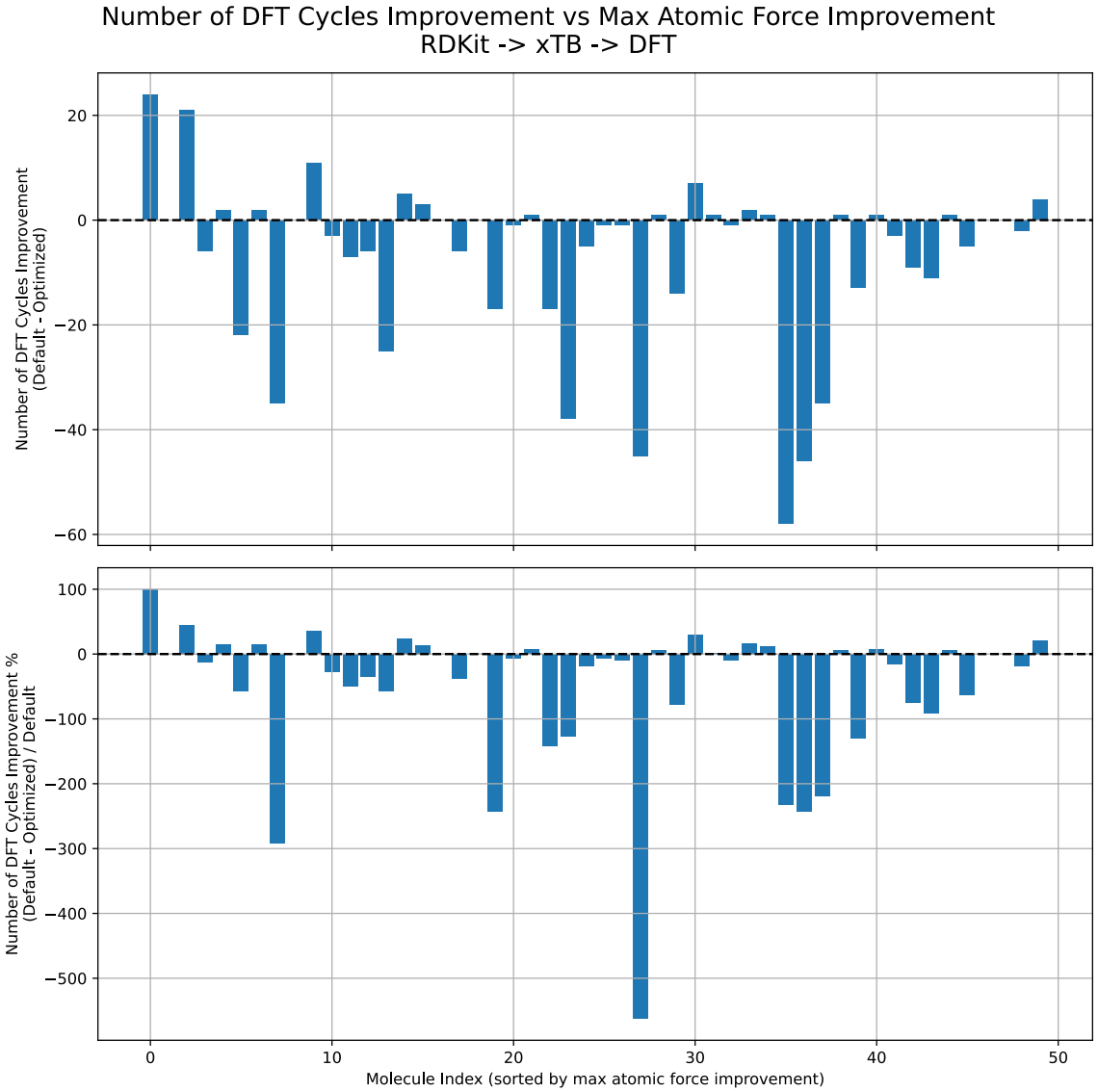

Figure 4.7: Relationship between the change in atomization energy (relative to default xTB parameters) and the change in the number of DFT relaxation cycles. Each bar represents a molecule. Molecules are sorted by the reduction in maximum atomic force with the greatest reduction on the left. Negative values on the y-axis indicate more DFT cycles required. The plot assesses whether optimizing for forces is proportional to improvements in geometric convergence.

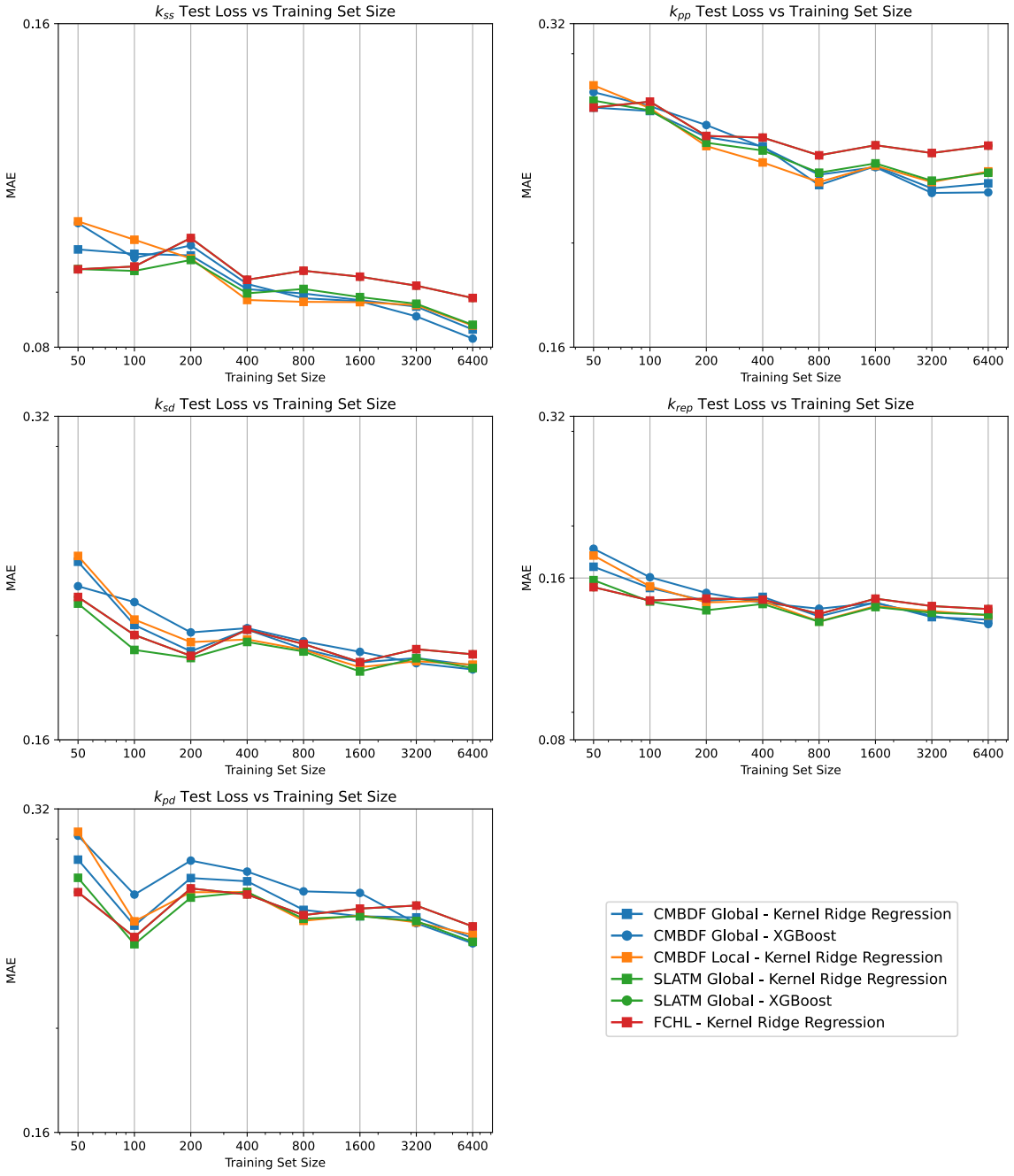

Figure 4.8: Test loss versus training set size for optimized parameters using different representations and regressors. From left to right, top to bottom, the subplots correspond to k_{ss}, k_{pp}, k_{sd}, k_{pd}, and k_{rep}. Results were obtained using 4-fold nested cross-validation with a fixed test set size of 200.

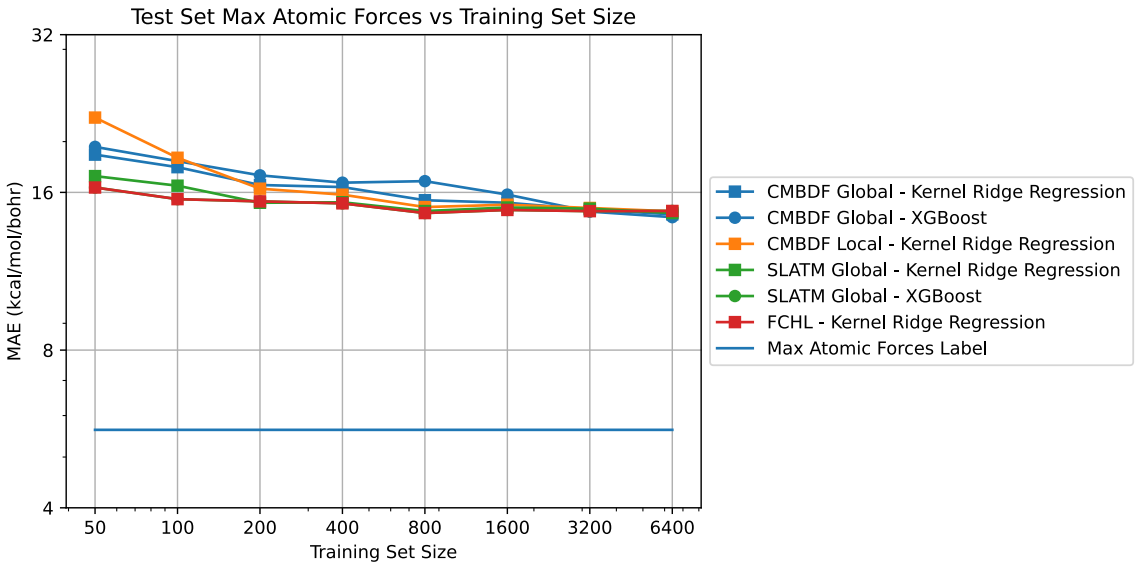

Figure 4.9: Test set maximum atomic force versus training set size using different representations and regressors. The maximum atomic force calculated by the label parameters is indicated which represents the performance expectations of a perfect regressor. Results were obtained using 4-fold nested cross-validation with a fixed test set size of 200.

Chapter 5

Conclusion

In this thesis, I explored the use of adaptive parameter tuning to improve the accuracy of the semi-empirical quantum chemistry method GFN2-xTB. Specifically, I focused on learning system-specific global parameters that reduce the error in atomic forces on DFT-relaxed geometries.

Results showed that optimizing xTB parameters with respect to maximum atomic force substantially reduced force magnitudes across molecules in the VQM24 dataset. In contrast, optimizing for structural displacement led to unstable parameter configurations, often increasing force magnitudes and producing outliers. Analysis of parameter distributions and correlations revealed that certain parameters, such as k_{ss} and k_{rep}, are coupled, underscoring the value of joint optimization approaches.

Despite force-based improvements, these gains did not translate into reduced DFT relaxation costs. Across multiple experimental setups, optimized xTB geometries often required more DFT cycles to converge than those generated with default parameters. This result suggests that minimizing maximum atomic force is not an effective surrogate for reducing downstream compute time. Structural deviation or energy-based metrics may provide better optimization targets for practical use.

Lastly, the use of machine learning to predict optimal parameters proved challenging. While learning curves showed improvement with scale, prediction errors remained large relative to the label range, and no model or representation emerged as clearly superior. This highlights the complexity of the underlying parameter-observable relationship and the need for more expressive models and perhaps larger, more chemically diverse datasets.

In summary, while adaptive parameter tuning can reduce force errors in xTB, translating these improvements into practical gains—such as reduced DFT compute time—remains an open challenge. Future work should explore alternate optimization objectives, better molecular representations, and more advanced models that can account for parameter interdependencies.

5.1 Future Work

There are several promising directions to extend this work beyond its current limitations.

First, a key bottleneck in scaling the dataset has been the computational expense of Bayesian optimization. As each optimization requires numerous expensive evaluations, this approach becomes

increasingly infeasible at scale. A more scalable alternative would be to shift from learning optimal parameters directly to learning the relationship between atomic forces and the xTB parameters. By modeling this relationship, it would be possible to train on suboptimal parameter-force pairs. At deployment time, one could then query the learned model to infer the parameter set that yields zero or near-zero atomic forces, bypassing the need for explicit optimization altogether.

Second, the study was limited to two geometry-based optimization objectives: maximum atomic force and a displacement-based metric. While maximum atomic force was chosen under the intuition that the atom experiencing the largest force dictates convergence behavior in DFT relaxations, our results show that this metric may not be sufficient. Even after optimization, the maximum atomic force remained far above typical DFT convergence thresholds (on the order of 0.01 kcal/mol/bohr). This suggests that alternative or more comprehensive objectives—such as minimizing the full force vector norm, total force RMS, or direct measures of geometric deviation—may lead to better alignment with DFT convergence behavior. Future work should explore these alternative loss functions to better align optimization objectives with downstream utility.

Third, the machine learning models in this study were limited to predicting each parameter independently. However, as shown in Figure 4.5, there are significant correlations between certain parameters—such as between k_{ss}, k_{pp}, and k_{rep}-suggesting underlying dependencies. This motivates the use of models that can capture joint parameter distributions. Future work should explore multivariate regressors or structured models—such as multi-output neural networks or message-passing architectures—that can explicitly model parameter interdependencies.

Appendix A

Code

The full code used for optimization, machine learning model training and evaluation, and the generation of figures is available at: `https://github.com/Jonathan-Woo/xtb-optimization`

Appendix B

Training Dataset Generation

To generate the training dataset of 10000 optimized parameter sets for 10000 unique molecules, the optimal parameter bounds and number of iterations was determined systematically.

B.1 Parameter Bounds Selection

In selecting parameter bounds, it was necessary to minimize the parameter search space while ensuring that the minima for each molecule was included. As a result, an iterative process was followed where an optimization run was executed given a fixed number of iterations and parameter bounds. Then, if any of the optimized parameters was at a boundary, then the bounds for each parameter were increased by a factor of 10% of the default value and another optimization run was executed. By increasing the search space, there were situations where the found parameters performed poorer than the previous run so the number of iterations was increased by 500 steps and then another optimization run was executed. Otherwise, the process was completed and the parameter bounds were chosen by inspection.

B.2 Number of Iterations Selection

In the process of determining the parameter bounds, when any optimized parameter hit the boundary, all parameter bounds were increased. As a result, there was redundancy in some of the parameter bounds. So, after reducing the parameter bounds to the minimum set in the previous section, I finally decided on the minimum number of iterations required by trying various options in increments of 500. Based on Figure B.2, 2500 iterations was chosen as the loss began to plateau beyond that.

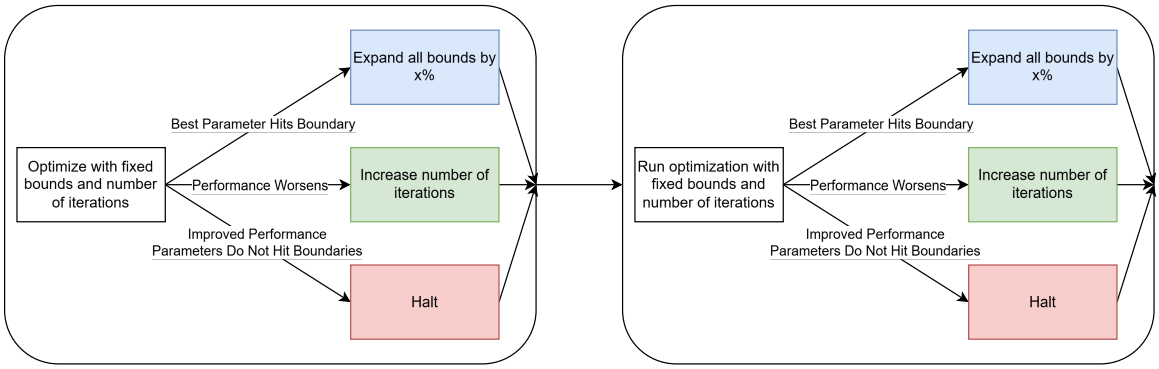

Figure B.1: Systematic process for selecting parameter bounds. For each optimization run, one of three outcomes was possible. (1) optimal parameter is on the parameter boundary indicating that the parameter may be further minimized given the opportunity to explore further in that direction so the parameter bounds were increased. (2) the performance of this optimization run was worse than the best one so far likely due to an expansion in the parameter space. Larger parameter space with the same number of iterations results in a lower effective search resolution. (3) best performance with optimized parameters that don't hit the boundaries. For cases (1) and (2), a new optimization run was executed and the process was repeated based on the new results.

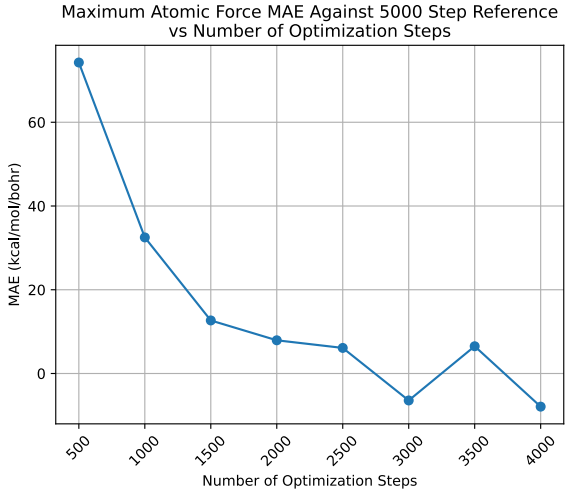

Figure B.2: Maximum atomic force MAE compared to the parameter bounds search dataset as a function of the number of optimization steps. The reference dataset was generated using 5000 optimization steps across a broad and highly redundant parameter space. After filtering out redundant regions, a performance plateau is observed around 2500 optimization steps, suggesting this as an effective step count for the reduced parameter space.

Appendix C

Hyperparameters

Table C.1: Hyperparameters tested for Kernel Ridge Regression.

Hyperparameter	**Value**
Length scale (σ^2)	10^3, 10^2, 10^1, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-6}, 10^{-9}, 10^{-10}
Regularization strength (λ)	0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4, 12.8, 25.6, 51.2, 102.4, 204.8, 409.6, 819.2
Kernel	Gaussian

Table C.2: Final selected hyperparameters for Kernel Ridge Regression across different molecular representations.

Representation	**Length Scale (σ^2)**	**Regularization Strength (λ)**	**Kernel**
cMBDF Global	1.0	409.6	Gaussian
cMBDF Local	10^2	12.8	Gaussian
SLATM Global	1.0	409.6	Gaussian
FCHL	1.0	12.8	Gaussian

Table C.3: Hyperparameters tested for XGBoost.

Hyperparameter	**Value**
Number of estimators (n_estimators)	100, 200, 300
Maximum tree depth (max_depth)	3, 5, 7
Learning rate (learning_rate)	0.01, 0.1, 0.2
Subsample ratio (subsample)	0.8, 1.0
Column subsample ratio (colsample_bytree)	0.8, 1.0

Table C.4: Final selected hyperparameters for XGBoost across different molecular representations.

Representation	**n_estimators**	**max_depth**	**learning_rate**	**subsample**	**colsample_bytree**
cMBDF Global	200	3	0.1	1.0	0.8
SLATM Global	100	3	0.01	1.0	0.8

References

[1] Seoin Back et al. "Accelerated chemical science with AI". In: *Digital Discovery* 3 (1 2024), pp. 23–33. DOI: 10.1039/D3DD00213F. URL: http://dx.doi.org/10.1039/D3DD00213F.

[2] Alex Zunger. "Inverse design in search of materials with target functionalities". In: *Nature Reviews Chemistry* 2.4 (2018), p. 0121.

[3] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik. "Inverse molecular design using machine learning: Generative models for matter engineering". In: *Science* 361.6400 (2018), pp. 360–365.

[4] Gerbrand Ceder. "Predicting properties from scratch". In: *Science* 280.5366 (1998), pp. 1099–1100.

[5] Attila Szabo and Neil S Ostlund. *Modern quantum chemistry: introduction to advanced electronic structure theory*. Courier Corporation, 1996.

[6] Walter Kohn and Lu Jeu Sham. "Self-consistent equations including exchange and correlation effects". In: *Physical review* 140.4A (1965), A1133.

[7] Axel D Becke. "Perspective: Fifty years of density-functional theory in chemical physics". In: *The Journal of chemical physics* 140.18 (2014).

[8] James JP Stewart. "Optimization of parameters for semiempirical methods I. Method". In: *Journal of computational chemistry* 10.2 (1989), pp. 209–220.

[9] 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 (2019), pp. 1652–1671.

[10] Philipp Pracht et al. "A robust non-self-consistent tight-binding quantum chemistry method for large molecules". In: (2019).

[11] Stefan Grimme, Christoph Bannwarth, and Philip Shushkov. "A robust and accurate tight-binding quantum chemical method for structures, vibrational frequencies, and noncovalent interactions of large molecular systems parametrized for all spd-block elements (Z= 1–86)". In: *Journal of chemical theory and computation* 13.5 (2017), pp. 1989–2009.

[12] Danish Khan, Maximilian L Ach, and O Anatole von Lilienfeld. "Adaptive atomic basis sets". In: *arXiv preprint arXiv:2404.16942* (2024).

[13] Danish Khan et al. "Adapting hybrid density functionals with machine learning". In: *Science Advances* 11.5 (2025), eadt7769.

[14] Axel D Becke. “Becke’s three parameter hybrid method using the LYP correlation functional”. In: *J. Chem. Phys* 98.492 (1993), pp. 5648–5652.

[15] Carlo Adamo and Vincenzo Barone. “Toward reliable density functional methods without adjustable parameters: The PBE0 model”. In: *The Journal of chemical physics* 110.13 (1999), pp. 6158–6170.

[16] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical bayesian optimization of machine learning algorithms”. In: *Advances in neural information processing systems* 25 (2012).

[17] Danish Khan and O Anatole von Lilienfeld. “Generalized convolutional many body distribution functional representations”. In: *arXiv preprint arXiv:2409.20471* (2024).

[18] Danish Khan et al. “Quantum mechanical dataset of 836k neutral closed shell molecules with upto 5 heavy atoms from CNOFSiPSClBr”. In: *arXiv preprint arXiv:2405.05961* (2024).

[19] Frank Jensen. *Introduction to computational chemistry*. John wiley & sons, 2017.

[20] Shuhei Watanabe. “Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance”. In: *arXiv preprint arXiv:2304.11127* (2023).

[21] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[22] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5 (1998), pp. 1299–1319.

[23] Matthias Rupp et al. “Fast and accurate modeling of molecular atomization energies with machine learning”. In: *Physical review letters* 108.5 (2012), p. 058301.

[24] Katja Hansen et al. “Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space”. In: *The journal of physical chemistry letters* 6.12 (2015), pp. 2326–2331.

[25] Albert P Bartók, Risi Kondor, and Gábor Csányi. “On representing chemical environments”. In: *Physical Review B—Condensed Matter and Materials Physics* 87.18 (2013), p. 184115.

[26] Jörg Behler. “Atom-centered symmetry functions for constructing high-dimensional neural network potentials”. In: *The Journal of chemical physics* 134.7 (2011).

[27] Anders S Christensen et al. “FCHL revisited: Faster and more accurate quantum machine learning”. In: *The Journal of chemical physics* 152.4 (2020).

[28] Bing Huang and O Anatole von Lilienfeld. “Quantum machine learning using atom-in-molecule-based fragments selected on the fly”. In: *Nature chemistry* 12.10 (2020), pp. 945–951.

[29] Marcel Müller, Andreas Hansen, and Stefan Grimme. “ωB97X-3c: A composite range-separated hybrid DFT method with a molecule-optimized polarized valence double-ζ basis set”. In: *The Journal of Chemical Physics* 158.1 (2023).

[30] Daniel GA Smith et al. “PSI4 1.4: Open-source software for high-throughput quantum chemistry”. In: *The Journal of chemical physics* 152.18 (2020).

[31] Greg Landrum. "RDKit: Open-Source Cheminformatics Software". In: (2016). URL: https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4.

[32] Julian D Gale et al. "A universal force field for materials, periodic GFN-FF: implementation and examination". In: *Journal of Chemical Theory and Computation* 17.12 (2021), pp. 7827–7849.

[33] Leonid Komissarov and Toon Verstraelen. "Improving the Silicon Interactions of GFN-xTB". In: *Journal of Chemical Information and Modeling* 61.12 (2021). PMID: 34890199, pp. 5931–5937. DOI: 10.1021/acs.jcim.1c01170. eprint: https://doi.org/10.1021/acs.jcim.1c01170. URL: https://doi.org/10.1021/acs.jcim.1c01170.