

Development and Application of Machine Learning Potentials for QM/MM Simulations of Biomolecular Systems

Zur Erlangung des akademischen Grades eines
DOKTORS DER NATURWISSENSCHAFTEN
(Dr. rer. nat.)

von der KIT-Fakultät für Chemie und Biowissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Lukas Petersen
aus Rendsburg

Tag der mündlichen Prüfung: 16.04.2026

Referent: Prof. Dr. Marcus Elstner

Korreferent: Prof. Dr. Pascal Friederich

Abstract

Enzymatic reactions are fundamental to life, and understanding their mechanisms poses a crucial step toward rational enzyme design with wide-ranging applications in drug design and biotechnology. Molecular simulations offer a powerful tool for investigating these reactions at the atomic level, but they span a wide range of spatial and temporal scales. The hybrid QM/MM approach is a widely used method to bridge this gap, treating the active site with Quantum Mechanical (QM) methods while modeling the surrounding environment with Molecular Mechanics (MM). Semi-empirical QM methods provide further avenues to access longer timescales but come with trade-offs in accuracy. The advent of Machine Learning Potentials (MLPs) in recent years gives the prospect of achieving QM accuracy at MM cost. However, several challenges remain in applying MLPs to QM/MM simulations, most notably the accurate representation of electrostatic interactions, since most MLPs are designed for vacuum.

This thesis systematically develops, compares, and validates ML/MM approaches for enzymatic reactions, from training data generation through architecture selection to production simulations of protein reactions. The ultimate goal of this thesis is to reach QM accuracy at MM cost for enzymatic reactions, enabling routine simulations of enzymatic catalysis with favorable accuracy and efficiency.

In Chapter 4, different sampling methods for generating MLP training data are compared. Potential energy surface scans, enhanced sampling techniques, and uncertainty-based sampling are evaluated with respect to their configurational space coverage and MLP performance in a cross-validation setting.

In Chapter 5, the focus lies on reproducing the electrostatic potential of the continuous charge distribution using partitioning schemes that assign partial charges. Each scheme's performance is evaluated by its ability to reproduce the electrostatic potential and its suitability for MLP training.

In Chapter 6, model-specific hyperparameters of ML/MM-capable MLPs are optimized, and data efficiency is assessed by training on increasing dataset sizes. Multiple independent models are trained with matched data splits in vacuum and solvent to enable a controlled comparison of the ML/MM training. Finally, the trained models are validated in metadynamics simulations using stability criteria, Potential of Mean Force (PMF) reconstruction, and reference recalculations of configurations encountered during the simulations.

Zusammenfassung

Enzymatische Reaktionen sind von lebensnotwendiger Bedeutung, und das Verständnis ihrer Mechanismen ist entscheidend für das rationale Enzymdesign. Molekulare Simulationen bieten ein leistungsfähiges Werkzeug, um diese Reaktionen auf atomarer Ebene zu untersuchen, doch mit diesen multiskaligen Simulationen geht eine enorme Diskrepanz zwischen räumlichen und zeitlichen Skalen einher. Der QM/MM-Ansatz schließt diese Lücke, indem das aktive Zentrum mit Quantum Mechanical (QM)-Methoden und die Umgebung mit Molecular Mechanics (MM)-Methoden behandelt wird. Mit weniger präzisen semi-empirischen QM-Methoden lassen sich längere Zeitskalen erreichen. Mit Machine Learning Potentials (MLPs) lässt sich die Genauigkeit von QM zu MM-Kosten erreichen. Allerdings sind MLPs in QM/MM-Simulationen noch herausfordernd, insbesondere bei elektrostatischen Wechselwirkungen, da die meisten davon auf Vakuum ausgelegt sind. Weiterhin sind effektive Trainingsdatenerzeugung und Stabilitätsgewährleistung keine trivialen Aufgaben.

Diese Arbeit entwickelt, vergleicht und validiert systematisch ML/MM-Ansätze für enzymatische Reaktionen, von der Trainingsdaten-Generierung über die Auswahl der Architektur bis hin zu Produktionssimulationen. Das letztendliche Ziel dieser Arbeit ist es, für enzymatische Reaktionen die Genauigkeit von QM zu den Kosten von MM zu erreichen, um Routinesimulationen der enzymatischen Katalyse mit hoher Genauigkeit und Effizienz zu ermöglichen.

In Kapitel 4 werden verschiedene Stichprobenverfahren zur Generierung von MLP-Trainingsdaten verglichen. Potentialhyperflächen-Scans, Metadynamik und unsicherheitsbasiertes Sampling werden anhand ihrer Konfigurationsraumabdeckung und ihrer MLP-Leistung in einer Kreuzvalidierung bewertet.

In Kapitel 5 liegt der Schwerpunkt auf der Reproduktion des elektrostatischen Potentials für eine kontinuierliche Ladungsverteilung mithilfe von Partitionierungsschemata, die Partialladungen zuweisen. Die Leistung jedes Schemas wird anhand seiner Fähigkeit zur Reproduktion des elektrostatischen Potentials sowie seiner Eignung für das MLP-Training bewertet.

In Kapitel 6 werden modellspezifische Hyperparameter optimiert und die Dateneffizienz bei zunehmender Datensatzgröße bewertet. Es werden mehrere unabhängige Modelle mit angepassten Datensplits im Vakuum und im Lösungsmittel trainiert, um den Vergleich des ML/MM-Trainings zu ermöglichen. Schließlich werden die trainierten Modelle in Metadynamik-Simulationen mittels Stabilitätskriterien, Potential of Mean Force (PMF)-Rekonstruktionen und Referenz-Neuberechnungen validiert.

Contents

I. Introduction	1
1. Motivation	3
1.1. Scope of This Work	4
2. Theoretical Background	5
2.1. Molecular Mechanics	5
2.1.1. Force Fields	5
2.2. Molecular Dynamics	6
2.2.1. Sampling Methods for Thermodynamic Properties	14
2.2.2. Sampling Methods for Machine Learning Potentials	18
2.3. Quantum Mechanics	23
2.3.1. Schrödinger Equation	23
2.3.2. Density Functional Theory	25
2.3.3. Basis Sets	28
2.3.4. Semi-Empirical Methods	29
2.3.5. Population Analysis	32
2.4. Quantum Mechanics/Molecular Mechanics	36
2.5. Machine Learning	37
2.5.1. Artificial Neural Networks	37
2.5.2. Machine Learning Potentials	41
3. Data and Methods	59
3.1. Workflow	59
3.2. Sampling Methods	60
3.3. Reference Methods	64
3.4. Machine Learning Potentials — Training	66
3.5. Machine Learning Potentials — Simulation	71
3.5.1. Von Willebrand Factor Simulation	73
3.6. Software	73
3.7. Implementation	74

II. Results and Discussion	79
4. Sampling Methods	81
4.1. Alanine Dipeptide	82
4.1.1. Geometry Comparison	83
4.1.2. Property Comparison	85
4.1.3. Cross-validation	85
4.1.4. Conclusion	87
4.2. Thiol Disulfide Exchange Reaction	88
4.2.1. Geometry Comparison	88
4.2.2. Property Comparison	92
4.2.3. Cross-validation	92
4.2.4. Conclusion	94
5. Electrostatic Interactions in QM/MM Simulations	95
5.1. Methodology	96
5.2. Alanine Dipeptide	97
5.2.1. Partial Charge Method Comparison	97
5.2.2. Environment Comparison	97
5.2.3. Electrostatic Potential Reproduction	99
5.2.4. Charge Training	100
5.2.5. Conclusion	101
5.3. Thiol Disulfide Exchange Reaction	102
5.3.1. Partial Charge Method Comparison	102
5.3.2. Environment Comparison	102
5.3.3. Electrostatic Potential Reproduction	104
5.3.4. Conclusion	104
6. Machine Learning Potentials for QM/MM	105
6.1. Training Alanine Dipeptide in Vacuum	105
6.1.1. Hyperparameter Optimization	106
6.1.2. Dataset Size Dependence	111
6.1.3. Training in Vacuum	112
6.1.4. Metadynamics Simulation in Vacuum	115
6.2. Training Thiol Disulfide Exchange Reaction in Vacuum	120
6.2.1. Hyperparameter Optimization	120
6.2.2. Training in Vacuum	120
6.2.3. Metadynamics Simulation in Vacuum	121

6.3. Training Alanine Dipeptide in Water	127
6.3.1. Environment Comparison	128
6.3.2. Training in Water	129
6.3.3. Metadynamics Simulation in Water	130
6.3.4. Conclusion	137
6.4. Training Thiol Disulfide Exchange Reaction in Environment	138
6.4.1. Environment Comparison	138
6.4.2. Training in Environment	138
6.4.3. Metadynamics Simulation in Water	140
6.4.4. Metadynamics Simulation in Protein	143
6.5. Δ -Learning in Vacuum and Solvent	148
6.5.1. Reference Comparison	148
6.5.2. Δ -Learning Performance	152
6.5.3. Metadynamics Simulation	154
6.5.4. Δ -Learning Thiol Disulfide Exchange Reaction	154
6.5.5. Conclusion	154
6.6. Computational Cost	155
 III. Conclusions and Outlook	 159
 A. List of Figures	 169
 B. List of Tables	 173
 C. Bibliography	 175
 D. Appendix	 189
D.1. Sampling Methods	189
D.2. Electrostatic Potential Reproduction	195
D.3. Hyperparameter Search — Alanine Dipeptide	207
D.4. Hyperparameter Search — Thiol Disulfide Exchange Reaction	227
D.5. Dataset Size Dependence — Thiol Disulfide Exchange Reaction	235
D.6. Δ -Learning — Alanine Dipeptide	236
D.7. Δ -Learning — Thiol Disulfide Exchange Reaction	237

Part I.

Introduction

1. Motivation

Recent advances in neural networks have transformed many fields, including natural language processing, computer vision, medical image analysis, and also chemistry. Their scientific impact was highlighted by the 2024 Nobel Prize in Chemistry, which recognized breakthroughs in computational protein science, including structure prediction with AlphaFold¹. This progress demonstrates that data-driven models can solve highly complex, high-dimensional problems that were previously out of reach for conventional approaches.

In computational chemistry, this development opens an important opportunity. Accurate Quantum Mechanical (QM) electronic-structure methods are still too expensive for routine long-timescale Molecular Dynamics (MD) simulations of large molecular systems, while Molecular Mechanics (MM) methods are efficient but approximate. Machine Learning Potentials (MLPs) promise to bridge this gap by learning the QM potential energy surface and forces from reference data, with the objective of approaching QM-level accuracy at near-MM computational cost.

This promise becomes particularly relevant in solvated and biomolecular systems, where environmental effects are essential. Solvent and protein surroundings can strongly alter relative conformer stabilities, barrier heights, and reaction pathways through electrostatic and polarization effects. A framework to address this multiscale problem is the Quantum Mechanics/Molecular Mechanics (QM/MM)⁴ method, in which the reactive region is treated quantum-mechanically and the environment classically. The 2013 Nobel Prize in Chemistry recognized the significance of this approach. Still, many processes of interest, such as chemical

reactions, conformational rearrangements, ligand binding events, or rare reactive transitions, occur on nanosecond-to-millisecond timescales or beyond. MD trajectories need millions to billions of integration steps for a single such simulation. Even with QM/MM and semi-empirical methods, routinely accessible

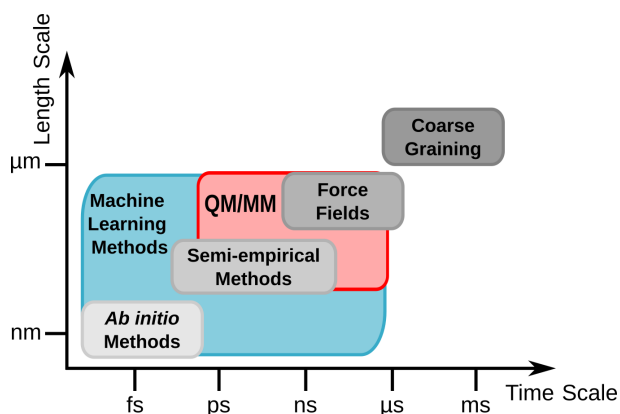


Figure 1.1.: Accessible time and length scales of methods in molecular dynamics simulations.^{2,3}

trajectory lengths are often limited to the picosecond-to-nanosecond range. Closing this timescale gap without sacrificing accuracy is a central motivation for the development of efficient ML/MM methods.

However, transferring MLPs to this setup is non-trivial: most models are designed for application to molecules in vacuum, while QM/MM requires robust treatment of environment-dependent electrostatics, transferability across varying local environments, and stable long-time dynamics beyond static test-set accuracy.^{5,6}

1.1. Scope of This Work

This thesis addresses this challenge by developing, training, and validating ML/MM workflows that retain the efficiency of MLPs while remaining reliable for realistic condensed-phase simulations. Two representative model systems are studied throughout: alanine dipeptide, a standard benchmark for conformational sampling, and thiol disulfide exchange, an enzymatic reaction catalyzed by glutaredoxin proteins⁷.

Sampling strategies (Chapter 4). Different data-generation strategies are compared with respect to configurational coverage, physical plausibility, and transferability across heterogeneous test sets. The goal is to identify sampling protocols that provide robust training data for stable MLP behavior.

Electrostatic interactions (Chapter 5). The error introduced by replacing the continuous quantum-mechanical electron density with atom-centered partial charges is quantified directly on the electrostatic potential and electrostatic energies. Different charge partitioning schemes are evaluated with respect to accuracy, learnability, and chemical reasonability to guide the choice of charge targets for ML/MM models.

Architecture benchmark in vacuum. A broad architecture benchmark under harmonized training conditions is performed on molecules in vacuum to assess which MLP architectures yield not only low test-set errors but also stable long-timescale dynamics in metadynamics simulations. This benchmark serves as a baseline for the subsequent evaluation of ML/MM training.

Training in condensed-phase environments. Models are retrained and evaluated in explicit solvent and protein-like environments, to assess whether environment-aware electrostatic treatments improve transferability and free-energy reproduction compared to environment-agnostic baselines.

Δ -learning. Δ -learning⁸ strategies that combine a fast semi-empirical baseline with a learned correction are explored as an alternative route to accurate ML/MM simulations.

2. Theoretical Background

2.1. Molecular Mechanics

Biomolecular systems typically consist of thousands to millions of atoms. QM methods are computationally too expensive to simulate such large systems.⁹ Therefore, classical MM methods are employed, which approximate the interactions between atoms using empirical force fields. These force fields describe the potential energy of a system as a function of the atomic positions, neglecting electronic degrees of freedom. This allows for the simulation of large biomolecular systems over extended timescales, making MM methods a powerful tool in computational biology and chemistry.¹⁰

2.1.1. Force Fields

Force fields describe the potential energy of a molecular system via terms that account for bonded and non-bonded interactions. Bonded interactions include bond stretching, angle bending, and dihedral torsions, while non-bonded interactions comprise van der Waals (vdW) and electrostatic interactions. Bond stretching and angle bending employ harmonic potentials, whereas dihedral torsions are often modeled using periodic cosine functions. Improper torsions, which are angles between two planes without a coincident chemical bond, can be used to maintain planarity in certain molecular groups, such as aromatic rings. Van der Waals interactions are typically modeled using Lennard-Jones potentials, while Coulomb's law describes electrostatic interactions. 1–4 interactions are frequently scaled to account for their partial inclusion in the bonded terms. The parameters for these potentials are derived from experimental data and high-level QM calculations. The total potential energy V_{total} of a system with N atoms can be expressed as:⁹

$$\begin{aligned}
V_{\text{total}}(\mathbf{r}^N) &= \underbrace{V_{\text{str}} + V_{\text{bend}} + V_{\text{tor}}}_{V_{\text{bonded}}} + \underbrace{V_{\text{vdW}} + V_{\text{elec}}}_{V_{\text{non-bonded}}} \\
&= \sum_{\text{bonds}} \frac{1}{2} k_b (r - r_0)^2 + \sum_{\text{angles}} \frac{1}{2} k_\theta (\theta - \theta_0)^2 + \sum_{\text{dihedrals}} \frac{1}{2} V_n (1 + \cos[n\phi - \gamma]) \\
&\quad + \sum_i \sum_{j>i} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_i \sum_{j>i} \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}}
\end{aligned} \tag{2.1}$$

r, θ, ϕ : Bond length, angle, and dihedral angle

k_b, k_θ : Force constants for bond and angle

r_0, θ_0 : Equilibrium bond length and angle

V_n, n, γ : Amplitude, multiplicity, and phase of dihedral

$\epsilon_{ij}, \sigma_{ij}$: Lennard-Jones parameters, well depth and equilibrium distance

ϵ_0 : Vacuum permittivity

q_i, q_j : Partial charges

r_{ij} : Distance between atoms i and j

Commonly used biomolecular force fields include Assisted Model Building with Energy Refinement (AMBER)¹¹, Chemistry at HARvard Macromolecular Mechanics (CHARMM)¹², and Optimized Potentials for Liquid Simulations (OPLS)¹³.

While MM methods enable the simulation of large biomolecular systems, they have limitations. The accuracy of MM simulations depends on the quality of the force field parameters, which may not capture all relevant interactions, particularly in complex environments. Additionally, MM methods cannot describe electronic effects such as bond formation and breaking, charge transfer, and polarization. Specialized reactive force fields, such as ReaxFF¹⁴, polarizable force fields, such as AMOEBA¹⁵, and QM/MM methods^{4,16} have been developed to address some of these limitations. Still, they entail increased computational cost and complexity.

Further abstraction levels, such as coarse-grained models like Martini¹⁷, reduce the system's degrees of freedom by grouping atoms into larger particles, so-called beads. This enables the simulation of even larger systems and longer timescales, but at the cost of losing atomistic detail.¹⁸

2.2. Molecular Dynamics

MD simulations model the time evolution of a molecular system by integrating Newton's equations of motion in time steps. The resulting trajectories specify the positions and velocities of all atoms in the

system as a function of time. They can be used to compute thermodynamic and kinetic properties of the system, such as free energies, diffusion coefficients, and reaction rates.⁹

MD simulations enable the exploration of the conformational space, equilibrium states, and dynamic behavior of biomolecules and complex systems over time. This can provide insights into reaction barriers, protein folding, ligand binding, and conformational changes, which are essential for understanding biological function and drug design.¹⁰

Newton's Equations of Motion

Nuclei behave, in good approximation, like classical particles due to their relatively large masses. Newton's equations of motion can describe their motion:

$$m_i \frac{d^2 \mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(t) = -\nabla_i V_{\text{total}}(\mathbf{r}^N(t)) \quad (2.2)$$

$\mathbf{r}_i(t)$: Position of atom i at time t

$\mathbf{F}_i(t)$: Force acting on atom i at time t

∇_i : Gradient with respect to the position of atom i , $\frac{\partial}{\partial \mathbf{r}_i}$

In the case of MM simulations, the forces $\mathbf{F}_i(t)$ are derived from the empirical force field $V_{\text{total}}(\mathbf{r}^N)$ as described in Eq. (2.1). Analytical gradients of the potential energy function are used to compute the forces acting on each atom at every time step. The equations of motion are then integrated numerically using algorithms such as the Verlet algorithm¹⁹, leapfrog algorithm²⁰, or velocity Verlet algorithm²¹ to update the positions and velocities of the atoms over time.

Verlet Algorithm

The Verlet algorithm efficiently integrates Newton's equations of motion by using positions from the current and previous time steps to calculate the new positions. The basic Verlet algorithm is given by:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{1}{2}\mathbf{a}_i(t)(\Delta t)^2 + \dots \quad (2.3)$$

$$\mathbf{r}_i(t - \Delta t) = \mathbf{r}_i(t) - \mathbf{v}_i(t)\Delta t + \frac{1}{2}\mathbf{a}_i(t)(\Delta t)^2 + \dots \quad (2.4)$$

$$\mathbf{r}_i(t + \Delta t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \mathbf{a}_i(t)(\Delta t)^2 + \dots \quad (2.5)$$

Eq. (2.5) shows that the new position $\mathbf{r}_i(t + \Delta t)$ can be computed independently of the velocity $\mathbf{v}_i(t)$ by using the current position $\mathbf{r}_i(t)$, the previous position $\mathbf{r}_i(t - \Delta t)$, and the acceleration $\mathbf{a}_i(t) = \mathbf{F}_i(t)/m_i$. The velocity can be estimated from the positions at the current and previous time steps:

$$\mathbf{v}_i(t) = \frac{\mathbf{r}_i(t + \Delta t) - \mathbf{r}_i(t - \Delta t)}{2\Delta t} \quad (2.6)$$

At each time step, positions and velocities are updated, and forces are recalculated using the new positions.^{9,19,22}

Leap-Frog Algorithm

The leap-frog algorithm is a variant of the Verlet algorithm that updates positions and velocities in a staggered manner. The position and velocity updates ‘leap-frog’ each other as the phase is shifted by half a time step. The leap-frog algorithm is given by:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i\left(t + \frac{\Delta t}{2}\right) \Delta t \quad (2.7)$$

$$\mathbf{v}_i\left(t + \frac{\Delta t}{2}\right) = \mathbf{v}_i\left(t - \frac{\Delta t}{2}\right) + \mathbf{a}_i(t) \Delta t \quad (2.8)$$

The leap-frog algorithm has two advantages over the basic Verlet algorithm: velocities are included directly in the integration scheme, and it is numerically more stable because truncation errors accumulate more quickly in the basic Verlet algorithm due to the $(\Delta t)^2$ term. A disadvantage is that the positions and velocities are not available at the same time, which can complicate the calculation of certain properties.^{9,20,22}

Velocity Verlet Algorithm

The velocity-Verlet algorithm combines the advantages of the basic Verlet and leapfrog algorithms by updating positions and velocities simultaneously within a single time step. The velocity Verlet algorithm is given by:

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{1}{2}\mathbf{a}_i(t)(\Delta t)^2 \quad (2.9)$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{1}{2}[\mathbf{a}_i(t) + \mathbf{a}_i(t + \Delta t)]\Delta t \quad (2.10)$$

Eq. (2.10) shows that the new velocity $\mathbf{v}_i(t + \Delta t)$ is computed using the average of the current acceleration $\mathbf{a}_i(t)$ and the new acceleration $\mathbf{a}_i(t + \Delta t)$, which is calculated from the potential after updating the positions via Eq. (2.9). This approach improves the accuracy of the velocity update and maintains numerical stability.^{9,21,22}

Maxwell-Boltzmann Distribution

The Maxwell-Boltzmann distribution describes the statistical distribution of particle velocities in a gas at thermal equilibrium. It is used to initialize the velocities of atoms in MD simulations according to a specified temperature T . The probability density function for the velocity component v_i in one dimension and the velocity magnitude v in three dimensions of a particle with mass m are given in Eq. (2.11) and Eq. (2.12). A visualization of the distributions is shown in Figure 2.1.

$$p(v_i) = \sqrt{\frac{m}{2\pi k_B T}} \exp\left(-\frac{mv_i^2}{2k_B T}\right) \quad (2.11)$$

$$p(v) = \left(\frac{m}{2\pi k_B T}\right)^{3/2} 4\pi v^2 \exp\left(-\frac{mv^2}{2k_B T}\right) \quad (2.12)$$

$p(v_i)$: Probability density function for velocity component $v_i, i = x, y, z$

$p(v)$: Probability density function for velocity magnitude $v = |\mathbf{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}$

k_B : Boltzmann constant

By sampling velocities from the Maxwell-Boltzmann distribution, the system can be initialized to reflect the desired temperature while ensuring that the total momentum is zero. This is typically done by first sampling the velocities from the distribution and then subtracting the center-of-mass velocity from each atom's velocity.⁹

Thermostat

MD simulations require a thermostat to control the system temperature. Thermostats adjust particle velocities to maintain a target temperature. The general idea of a simple thermostat is to scale the velocities

2. Theoretical Background

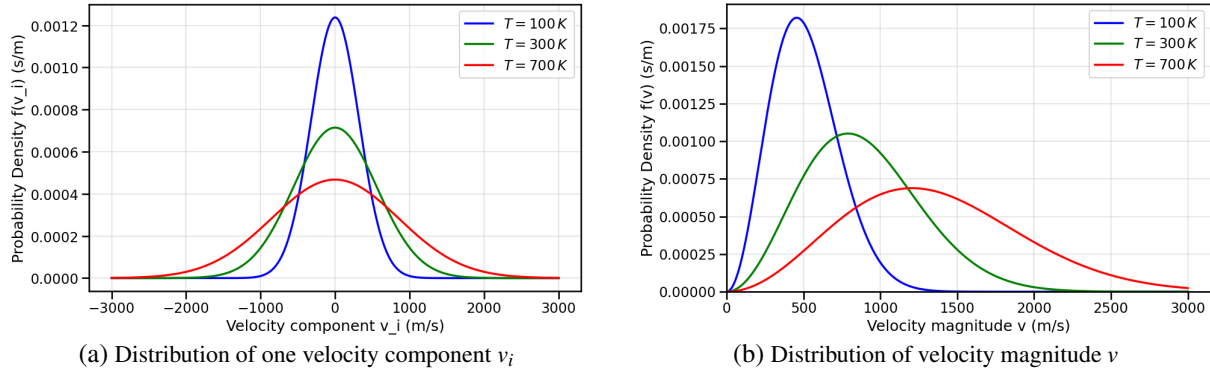


Figure 2.1.: Maxwell-Boltzmann distribution of (a) one velocity component v_i and (b) velocity magnitude v for a particle with mass $m = 8$ amu at different temperatures T .

of the particles based on the difference between the current temperature T and the target temperature T_{ref} by a scaling factor λ . The temperature difference ΔT and the scaling factor λ are given by:

$$\Delta T = T_{\text{ref}} - T \quad (2.13)$$

$$\Delta T = \sum_{i=1}^N \frac{\frac{1}{2} m_i (\lambda \mathbf{v}_i)^2}{\frac{3}{2} N k_B} - \sum_{i=1}^N \frac{\frac{1}{2} m_i \mathbf{v}_i^2}{\frac{3}{2} N k_B} \quad (2.14)$$

$$\Delta T = \lambda^2 T - T \quad (2.15)$$

$$\lambda = \sqrt{1 + \frac{\Delta T}{T}} \quad (2.16)$$

Commonly applied thermostats include the Berendsen thermostat²³, the Bussi-Parrinello thermostat²⁴, and the Nosé-Hoover thermostat^{25,26}.

Stochastic thermostats, such as the Andersen thermostat²⁷ and Langevin dynamics, regulate the temperature differently. They include a stochastic element that mimics collisions with an implicit heat bath.

Berendsen Thermostat The Berendsen thermostat adjusts the velocities of the particles smoothly based on the difference between the current temperature T and the target temperature T_{ref} by using a time constant τ_T that controls the strength of the coupling to the heat bath, typically in the range of 0.1 ps to 10 ps. A smaller τ_T leads to a faster response to temperature fluctuations. This results in an exponential relaxation of the system's temperature towards the target temperature.

$$\frac{dT}{dt} = \frac{1}{\tau_T} (T_{\text{ref}} - T) \quad (2.17)$$

$$\Delta T = \frac{\Delta t}{\tau_T} (T_{\text{ref}} - T) \quad (2.18)$$

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau_T} \left(\frac{T_{\text{ref}}}{T} - 1 \right)} \quad (2.19)$$

τ_T : Time constant for temperature coupling

This thermostat is computationally efficient and easy to implement. Still, it does not generate the correct canonical ensemble, as it gradually transfers energy from the fastest modes, the vibrations, to the slowest modes, the translational modes, a phenomenon known as the ‘flying ice cube’ effect.^{9,23,28}

Bussi-Parrinello Thermostat The Bussi-Parrinello thermostat, also known as the stochastic velocity rescaling thermostat, adds a correction to the Berendsen thermostat to ensure that the system samples the correct canonical ensemble. This correction comes in the form of a stochastic term that introduces random fluctuations to the temperature, ensuring that the system’s temperature distribution follows the Maxwell-Boltzmann distribution.²⁴

$$\Delta T = \frac{\Delta t}{\tau_T} (T_{\text{ref}} - T) + \frac{2\Delta W}{\sqrt{\tau_T}} \sqrt{\frac{T_{\text{ref}} T}{N_{DoF}}} \quad (2.20)$$

$$\lambda = \sqrt{1 + \frac{\Delta t}{\tau_T} \left(\frac{T_{\text{ref}}}{T} - 1 \right) + \frac{2\Delta W}{\sqrt{\tau_T}} \sqrt{\frac{T_{\text{ref}}}{TN_{DoF}}}} \quad (2.21)$$

ΔW : Sequence of random numbers, Wiener process

N_{DoF} : Number of degrees of freedom

Langevin Dynamics Langevin dynamics are a stochastic approach to simulate the motion of particles in a fluid. Three contributions act on each particle: the deterministic force derived from the potential energy, a frictional force that models drag from the implicit solvent, and a random force that represents thermal interactions with solvent molecules. These contributions are combined with Newton’s equations of motion to yield the Langevin equation:

$$m_i \frac{d^2 \mathbf{r}_i(t)}{dt^2} = \mathbf{F}_i(\{\mathbf{r}^N(t)\}) - m_i \gamma_i \frac{d\mathbf{r}_i(t)}{dt} + \sqrt{\frac{2m_i \gamma_i k_B T_{\text{ref}}}{\Delta t}} \mathbf{R}_i(t) \quad (2.22)$$

$\mathbf{F}_i(\{\mathbf{r}^N(t)\})$: Force from potential energy

γ_i : Friction coefficient for atom i

T_{ref} : Reference temperature

Δt : Time step

$\mathbf{R}_i(t)$: Random force contribution

Langevin dynamics are particularly useful for simulating systems in the absence of an explicit solvent. The temperature of the system is controlled by the friction and random forces, which act as a thermostat. The friction coefficient γ_i can be adjusted to tune the coupling strength to the heat bath, with higher values leading to stronger damping of the particle motion. The random forces $\mathbf{R}_i(t)$ are typically drawn from a Gaussian distribution with zero mean and $2m_i \gamma_i k_B T_{\text{ref}} / \Delta t$ variance to ensure that the system samples the correct Boltzmann distribution at the reference temperature.⁹

Andersen Thermostat The Andersen thermostat is also stochastic. It randomly selects particles to undergo collisions, reassigning their velocities according to the Maxwell-Boltzmann distribution (see Eq. (2.11) and Eq. (2.12)) at the target temperature T_{ref} . This mimics the effect of collisions with an implicit heat bath. The collision frequency ν must be chosen carefully, as a high collision frequency can disrupt the underlying equations of motion, whereas a low collision frequency may not effectively control the temperature. The mean collision rate is given by:

$$\nu = \frac{2\alpha\kappa}{3k_B \mathcal{N}^{1/3} N^{2/3}} \quad (2.23)$$

α : Dimensionless constant

κ : Thermal conductivity of the system

$\mathcal{N}$: Particle number density

N : Number of particles

A disadvantage of the Andersen thermostat is that it does not provide a continuous trajectory for the particles, as their velocities are randomly reassigned during collisions. This can affect the calculation of dynamic properties, such as diffusion coefficients and viscosity.^{9,27}

Barostat

A barostat is used in MD simulations to control the system pressure. Pressure control is relevant for generating the isothermal-isobaric (NPT) ensemble, which is prevalent under most experimental conditions. Barostats adjust the volume of the simulation box to maintain a target pressure P_{ref} . Microscopically, pressure is defined as the force per unit area exerted by the particles on the container walls. In MD simulations, pressure can be calculated using the virial theorem, which relates the pressure to the kinetic energy and the forces acting on the particles. The instantaneous pressure P of a system with volume V containing N particles is given by:

$$P = \frac{1}{3V} \left(\sum_{i=1}^N m_i v_i^2 + \sum_{i=1}^N \sum_{j>i}^N \mathbf{r}_{ij} \cdot \mathbf{F}_{ij} \right) \quad (2.24)$$

Both the Berendsen barostat²³ and the Parrinello-Rahman barostat²⁹ are frequently used in MD simulations.

Berendsen Barostat The Berendsen barostat uses a similar approach to the Berendsen thermostat. The change of pressure over time is proportional to the difference between the target pressure P_{ref} and the current pressure P , controlled by a coupling constant τ_p . The volume V of the simulation box is scaled by the respective scaling factor λ , which is equivalent to scaling the atomic coordinates by $\lambda^{1/3}$ in three dimensions.

$$\frac{dP}{dt} = \frac{1}{\tau_p} (P_{\text{ref}} - P) \quad (2.25)$$

$$\Delta P = \frac{\Delta t}{\tau_p} (P_{\text{ref}} - P) \quad (2.26)$$

$$\lambda = 1 - \kappa_T \frac{\Delta t}{\tau_p} (P_{\text{ref}} - P) \quad (2.27)$$

$$r'_i = \lambda^{1/3} r_i \quad (2.28)$$

τ_p : Time constant for pressure coupling

κ_T : Isothermal compressibility of the system

Usually, the scaling is applied isotropically. However, for certain 2D systems, such as membranes or surfaces, anisotropic scaling can be applied to allow for independent scaling in different spatial directions.^{9,23}

Periodic Boundary Conditions

Periodic boundary conditions create an infinite lattice of simulation boxes by replicating the primary simulation box in all spatial directions. This approach minimizes surface effects and mimics bulk behavior in finite-sized systems, thereby improving computational efficiency. When a particle moves out of the primary simulation box, it re-enters from the opposite side, maintaining a constant number of particles and density within the box. The simulation box can have different shapes. The simplest shape is a cubic box, but other space-filling shapes, such as the hexagonal prism or the truncated octahedron, can further reduce the number of solvent molecules. However, they are more complex to implement.

The Minimum Image Convention (MIC) is often employed, in which each particle interacts only with the nearest image of every other particle, thereby reducing computational cost while preserving accuracy. This is particularly important for electrostatic interactions, which are long-ranged and decay as $1/r$. Directly calculating all pairwise electrostatic interactions scales as $O(N^2)$, which becomes computationally prohibitive for large systems.

Another approach is a hard cutoff radius, in which interactions are computed only for particles within a specified distance of each other. This further reduces computational cost but can introduce artifacts at the cutoff boundary as gradients become discontinuous. To mitigate these artifacts, switching or shifting functions can be applied to smoothly reduce the interaction potential to zero at the cutoff distance.

A way to efficiently compute all long-range electrostatic interactions across periodic boundaries is the Ewald summation method³⁰. It splits the electrostatic potential into a short-range part, computed in real space, and a long-range part, computed in reciprocal space via Fourier transforms. The Particle Mesh Ewald (PME) method³¹ is a popular variant that uses a grid to discretize the charge distribution, enabling efficient computation of long-range interactions via fast Fourier transforms (FFT). PME scales as $O(N \log N)$, making it suitable for large systems.^{9,22}

2.2.1. Sampling Methods for Thermodynamic Properties

Statistical Mechanics

While experiments are performed at the macroscopic scale, computational simulations operate at the microscopic scale. Statistical mechanics is a tool for connecting these two scales by averaging over a large ensemble of microscopic states to obtain macroscopic properties. The experimental conditions determine the ensemble at hand. The most common ensembles are the microcanonical (NVE), canonical (NVT), and isothermal-isobaric (NPT) ensembles. In the NVE ensemble, the number of particles N , volume V , and energy E are constant. In the NVT ensemble, the number of particles N , volume V , and

temperature T are constant. In the NPT ensemble, the number of particles N , pressure P , and temperature T are constant.

The average $\langle A \rangle$ of a property A can be expressed as the sum over all possible discrete states i of the property A_i weighted by their respective probabilities p_i . The probabilities are given by the Boltzmann distribution for each state i with energy E_i .

$$\langle A \rangle = \frac{1}{M} \sum_i^M A_i = \sum_i A_i p_i \quad (2.29)$$

$$p_i = \frac{\exp\left(-\frac{E_i}{k_B T}\right)}{\sum_j \exp\left(-\frac{E_j}{k_B T}\right)} = \frac{\exp\left(-\frac{E_i}{k_B T}\right)}{Q} \quad (2.30)$$

$$Q = \sum_j \exp\left(-\frac{E_j}{k_B T}\right) \quad (2.31)$$

Here, Q is the canonical partition function, a recurring term in statistical mechanics that serves as a normalization factor for the probabilities. It's a central quantity from which many thermodynamic properties can be derived.

Another underlying assumption in statistical mechanics is that the time average of a property over a sufficiently long trajectory is equivalent to the ensemble average over all possible states within the continuous $6N$ -dimensional phase space consisting of the positions $\mathbf{r}^N$ and momenta $\mathbf{p}^N$ of N particles. This assumption is known as the ergodic hypothesis. This means that instead of generating and calculating all possible configurations in the phase space, a single long MD trajectory can be used to compute the property of interest.

Mechanical properties, such as internal energy U , pressure P , and heat capacity C_V , can be easily obtained from MD trajectories. They only depend on derivatives of the partition function Q . High-energy configurations contribute little to the average as their probabilities are low due to the exponential term with a negative exponent. Thermodynamic properties, such as free energy G , chemical potential μ , and entropy S , on the other hand, depend on the logarithm of the partition function Q . Therefore, despite their low probabilities, high-energy configurations can have a significant impact on the average as they enter the equation with a positive exponent. This is illustrated exemplarily for the internal energy U and the free energy F :

$$U = k_B T^2 \frac{\partial \ln Q}{\partial T} = \int \int E(\mathbf{r}^N, \mathbf{p}^N) \frac{\exp\left(-\frac{E(\mathbf{r}^N, \mathbf{p}^N)}{k_B T}\right)}{Q} d\mathbf{r}^N d\mathbf{p}^N \quad (2.32)$$

$$F = -k_B T \ln Q \propto \ln \left[\int \int \exp\left(+\frac{E(\mathbf{r}^N, \mathbf{p}^N)}{k_B T}\right) \rho(\mathbf{r}^N, \mathbf{p}^N) d\mathbf{r}^N d\mathbf{p}^N \right] \quad (2.33)$$

This makes the calculation of thermodynamic properties more challenging, as they require extensive sampling of phase space to capture the contributions from all relevant configurations accurately.^{9,32,33}

Enhanced sampling methods have been developed for this reason. They aim to improve phase-space exploration by biasing sampling toward rare events or high-energy configurations. Examples of enhanced sampling methods include umbrella sampling³⁴ and metadynamics³⁵.

Collective Variables

Collective Variables (CVs) are functions of the atomic coordinates that describe the system's state during a reaction or conformational change. They can be simple geometric properties, such as distances, angles, and dihedrals, or more complex composite properties, such as combinations of bonds or derived properties, such as the Root Mean Square Deviation (RMSD) to a reference structure. The projection of the high-dimensional phase space onto a low-dimensional CV space is known as Potential of Mean Force (PMF).⁹

The value of the PMF at a specific value of the CV ξ_0 and the difference in PMF between two values of the CV ξ_0 and ξ_1 can be extracted with a Delta function:

$$F(\xi = \xi_0) = -k_B T \ln(Q) - k_B T \ln\langle\delta(\xi - \xi_0)\rangle \quad (2.34)$$

$$F(\xi = \xi_0) - F(\xi = \xi_1) = -k_B T \ln \frac{\langle\delta(\xi - \xi_0)\rangle}{\langle\delta(\xi - \xi_1)\rangle} = -k_B T \ln \frac{p(\xi_0)}{p(\xi_1)} \quad (2.35)$$

The probabilities $p(\xi_0)$ and $p(\xi_1)$ can be estimated from the histogram of the CV values sampled during an MD simulation, as long as the CV is sufficiently sampled.

Umbrella Sampling

In umbrella sampling, a biasing potential is applied to the system to enhance the sampling of specific regions in the CV space. The biasing potential is typically harmonic and centered in intervals along the CV. When plotted against the CV, the biasing potentials resemble umbrellas, hence the name umbrella sampling. The biased potential energy function $V_{\text{biased}}(\mathbf{r}^N)$ is given by:

$$V_{\text{biased}}(\mathbf{r}^N) = V_{\text{unbiased}}(\mathbf{r}^N) + V_{\text{bias}}(\xi(\mathbf{r}^N)) \quad (2.36)$$

$$V_{\text{bias}}(\xi(\mathbf{r}^N)) = \frac{1}{2}k(\xi(\mathbf{r}^N) - \xi_0)^2 \quad (2.37)$$

$\xi(\mathbf{r}^N)$: Collective variable as a function of atomic coordinates

With multiple overlapping simulation windows along the CV, the unbiased PMF can be reconstructed using techniques such as the weighted histogram analysis method (WHAM)³⁶ or umbrella integration³⁷.

Metadynamics

Metadynamics is an iterative enhanced sampling method that discourages the system from revisiting previous CV values by periodically adding small Gaussian potentials, called hills, centered at the current CV value to the biasing potential. The total biasing potential $V_{\text{bias}}(\xi(\mathbf{r}^N), t)$ at time t is given by the sum of all previously added Gaussian hills. Barriers are gradually overcome as the biasing potential fills the free energy wells, thereby pushing the system to explore new regions. The unbiased PMF can be obtained by negating the biasing potential once the simulation has converged.

$$V_{\text{biased}}(\mathbf{r}^N, t) = V_{\text{unbiased}}(\mathbf{r}^N) + V_{\text{bias}}(\xi(\mathbf{r}^N), t) \quad (2.38)$$

$$V_{\text{bias}}(\xi(\mathbf{r}^N), t) = \sum_{t'=\Delta t, 2\Delta t, \dots}^{t' \leq t} w \exp\left(-\frac{(\xi(\mathbf{r}^N) - \xi(t'))^2}{2\sigma^2}\right) \quad (2.39)$$

$$F = -\lim_{t \rightarrow \infty} V_{\text{bias}}(\xi(\mathbf{r}^N), t) + \text{const} \quad (2.40)$$

Δt : Time interval between adding new hills

w : Height of the Gaussian hills

σ : Width of the Gaussian hills

Variants of metadynamics include well-tempered metadynamics³⁸, which gradually decreases the height of the added hills over time to improve convergence, bias-exchange metadynamics³⁹, which combines the two concepts of metadynamics and replica exchange, and On-the-fly Probability Enhanced Sampling (OPES)^{40–42}, which estimates the required probability distribution to sample from the target uniform distribution on-the-fly.

2.2.2. Sampling Methods for Machine Learning Potentials

While the sampling methods discussed in Section 2.2.1 are designed to compute thermodynamic properties by generating the respective ensemble, they can also be used to generate training data for MLPs. However, these methods aim to sample the high-dimensional phase space according to the Boltzmann distribution. The resulting training data may be suboptimal for training MLPs, as local minima are over-sampled and redundant, whereas high-energy configurations and transition states are underrepresented. This can lead to poor generalization of the MLP model, resulting in large errors in predicted energies and forces at critical points in phase space and ultimately causing instability during MD simulations. To address this issue, specialized sampling methods have been developed to generate more diverse and informative training data for MLPs. Alternative sampling methods for training data generation include normal mode sampling⁴³ or active learning techniques, such as adaptive sampling^{44–48} and uncertainty sampling^{49–51}.

Normal Mode Sampling

Normal Mode Sampling (NMS) generates training data by distorting starting structures along their vibrational normal modes to create new configurations. The normal modes are obtained by diagonalizing the Hessian matrix, which contains the second derivatives of the potential energy with respect to the atomic coordinates. The distortion amplitude can be scaled to a target temperature T to match thermally accessible configurations. NMS is computationally efficient and can generate a large number of diverse configurations from a small number of starting structures. However, it is unlikely to generate transition states or rotational barriers, as it only explores the local vicinity of the starting structures. Variations of NMS include the use of Principal Component Analysis (PCA) to replace the diagonalization of the Hessian matrix with a covariance matrix obtained from an MD trajectory, and Wigner sampling, which generates configurations based on the Wigner distribution of the quantum harmonic oscillator.^{52,53}

$$P_W(\mathbf{Q}, \mathbf{P}) = \prod_{i=1}^{N_F} \frac{1}{\pi \hbar} \exp \left(-\frac{1}{\hbar \omega_i} (\omega_i^2 Q_i^2 + P_i^2) \right) \quad (2.41)$$

$P_W(\mathbf{q}, \mathbf{p})$: Wigner distribution

N_F : Number of vibrational degrees of freedom

Q_i, P_i : Mass-scaled normal mode coordinates and momenta for mode i

ω_i : Frequency of mode i

Adaptive Sampling

Adaptive sampling is an active learning technique that iteratively improves an MLP. Its aim is not to construct a PMF, but to identify and sample underrepresented or high-uncertainty regions in the conformational space to improve the MLP model. The general workflow of adaptive sampling is depicted in Figure 2.2. It starts with an initial training set. An initial ensemble of models is trained on this training set and used to run multiple ML-driven MD simulations. During the simulations, significant conformations are identified using uncertainty metrics, such as ensemble disagreement. These conformations are then evaluated with the reference method, added to the training set, and the ensemble of models is retrained. This process is repeated until the ensemble no longer encounters sufficient new significant conformations.^{44–48}

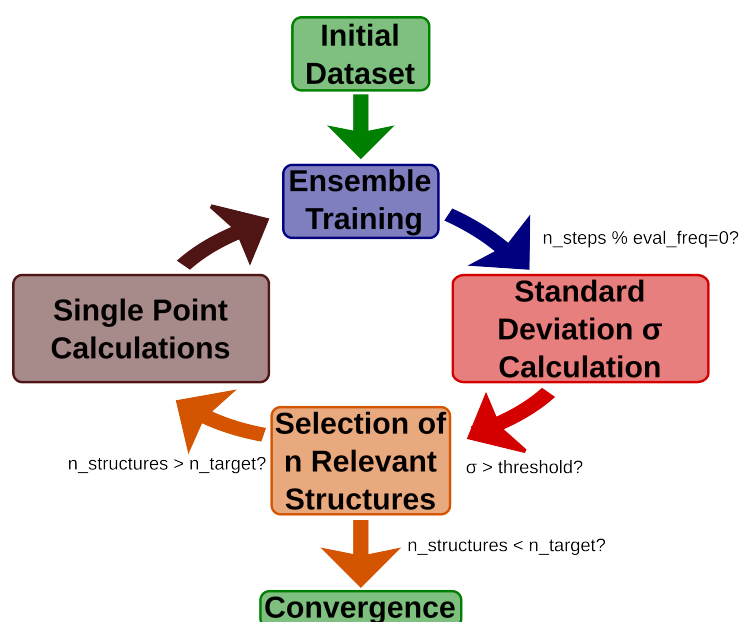


Figure 2.2.: Schematic workflow of adaptive sampling. Starting from an initial training set, an ensemble of models is trained and used to run multiple ML-driven MD simulations. Significant conformations are identified using uncertainty metrics, evaluated with the reference method, added to the training set, and the ensemble of models is retrained. This process is repeated until convergence.

Uncertainty Sampling

Uncertainty sampling is another active learning technique to improve the training of MLPs. In contrast to adaptive sampling, it doesn't passively run ML-driven MD simulations until significant conformations are found. Instead, it actively biases the sampling towards high-uncertainty regions. This can be achieved directly by basing the ensemble disagreement σ_E^2 .^{49,50}

$$\sigma_E^2 = \frac{1}{2} \sum_{m=1}^{N_M} (\hat{E}_m - \hat{E})^2 \quad (2.42)$$

$$V_{\text{bias}}(\sigma_E^2) = A \left[\exp \left(\frac{\sigma_E^2}{N_M N_A B^2} \right) - 1 \right] \quad (2.43)$$

$$-\frac{\partial}{\partial \mathbf{r}_i} V_{\text{bias}}(\sigma_E^2) = -\frac{\partial V_{\text{bias}}(\sigma_E^2)}{\partial \sigma_E^2} \frac{\partial \sigma_E^2}{\partial \mathbf{r}_i} \quad (2.44)$$

$$-\frac{\partial \sigma_E^2}{\partial \mathbf{r}_i} = -\sum_m^{N_M} (\hat{E}_m - \hat{E}) \frac{\partial}{\partial \mathbf{r}_i} (\hat{E}_m - \hat{E}) = -\sum_m^{N_M} (\hat{E}_m - \hat{E})(\hat{\mathbf{F}}_m - \hat{\mathbf{F}}) \quad (2.45)$$

$\hat{E}_m, \hat{\mathbf{F}}_m$: Predicted energy and forces from model m

$\hat{E}, \hat{\mathbf{F}}$: Mean predicted energy and forces from all models

N_M, N_A : Number of models in the ensemble and number of atoms

A, B : Hyperparameters to tune the strength and width of the biasing potential

To avoid the costly computation of ensemble disagreement, singular model uncertainty using batch selection algorithms⁵⁴, last-layer uncertainty⁵⁵, evidential deep learning⁵⁶, or Gaussian Mixture Models (GMMs)⁵¹ can be used instead.

Gaussian Mixture Model In the GMM approach by Dietschreit et al.⁵¹, subpopulations in the D -dimensional feature space are described by a weighted mixture of K Gaussian distributions for a data point $\mathbf{x}$ as follows:

$$p(\mathbf{x}|\mathbf{w}, \mu, \Sigma) = \sum_{k=1}^K w_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k) \quad (2.46)$$

w_k : Weight of Gaussian k , $\sum_{k=1}^K w_k = 1$

μ_k : Mean vector of Gaussian k , $\mu_k \in \mathbb{R}^{D \times 1}$

Σ_k : Covariance matrix of Gaussian k , $\Sigma_k \in \mathbb{R}^{D \times D}$

Covariance matrices Σ_k and mean vectors μ_k are predetermined using k-means clustering, and weights w_k are fitted to describe the latent feature vectors of a graph neural network potential (see Section 2.5.2). The negative log-likelihood of a new data point $\mathbf{x}_{\text{test}}$ describes how well it is represented by the k Gaussian distribution clusters found in the calibration data set $\mathbf{X}_{\text{calib}}$:

$$\text{NLL}(\mathbf{x}_{\text{test}}|\mathbf{X}_{\text{calib}}) = -\log \left(\sum_{k=1}^K w_k \mathcal{N}(\mathbf{x}_{\text{test}}|\mu_k, \Sigma_k) \right) \quad (2.47)$$

Extended Adaptive Biasing Force The extended Adaptive Biasing Force (eABF) method enables the use of GMM uncertainty as a CV to drive sampling toward high-uncertainty regions. The eABF method is an improvement of the Adaptive Biasing Force (ABF) method, which computes the mean of instantaneous forces $\langle F_{\text{inst}} \rangle_{\xi}$ acting on a CV ξ on-the-fly and applies this mean as a biasing force to cancel itself out, overcoming energy barriers to achieve uniform sampling along the CV.^{57–59} In the original formulation of ABF this requires an analytical expression for the instantaneous forces acting on the CV, which might be challenging to obtain for an arbitrary complex CV, such as the GMM uncertainty, and it requires all CVs and constraints to be orthogonal. The eABF method overcomes these limitations by introducing a fictitious particle with a corresponding fictitious coordinate λ that is harmonically coupled to the CV. In this case, the instantaneous force acting on the real coordinate $\mathbf{r}$ and the fictitious coordinate λ can be computed easily:^{59–61}

$$V_{\text{ext}}(\mathbf{r}, \lambda) = V(\mathbf{r}) + \frac{1}{2\beta\sigma^2}(\xi(\mathbf{r}) - \lambda)^2 \quad (2.48)$$

$$F_{\text{inst}}(\mathbf{r}, \lambda) = -\frac{\partial V_{\text{ext}}(\mathbf{r}, \lambda)}{\partial \mathbf{r}} = -\frac{\partial V(\mathbf{r})}{\partial \mathbf{r}} - \frac{1}{\beta\sigma^2}(\xi(\mathbf{r}) - \lambda) \frac{\partial \xi(\mathbf{r})}{\partial \mathbf{r}} \quad (2.49)$$

σ : Thermal coupling constant between ξ and λ

The mean force $\langle F_{\text{inst}} \rangle_{\lambda}$ is then computed by averaging the instantaneous forces in bins along the fictitious coordinate λ . This running average is used as an adaptive biasing force to cancel forces acting along the CV ξ and achieve uniform sampling across the CV space. The biased forces acting on the atomic coordinates $\mathbf{r}$ are given by:

$$F_{\text{bias}}(\lambda) = -\langle F_{\text{inst}} \rangle_{\lambda} \frac{\partial \xi(\mathbf{r})}{\partial \mathbf{r}} \quad (2.50)$$

$$F_{\text{total}}(\mathbf{r}, \lambda) = F_{\text{inst}}(\mathbf{r}, \lambda) + F_{\text{bias}}(\lambda) \quad (2.51)$$

The fictitious particle also gets assigned a mass and is propagated, receiving an instantaneous spring force and a corresponding adaptive biasing force, both directly computable without the partial derivatives of ξ , as λ is only harmonically coupled to ξ .

Gaussian Accelerated Molecular Dynamics The uncertainty sampling implementation from Dietschreit et al.⁵¹ employs Gaussian accelerated Molecular Dynamics (GaMD) as an additional accelerating biasing force. It allows the system to escape local minima more easily by adding a harmonic boost potential to the original potential energy surface whenever the system’s potential energy falls below a certain threshold energy E_{thresh} .⁶²

$$V_{\text{GaMD}}(V(\mathbf{r})) = \begin{cases} \frac{1}{2}k(E_{\text{thresh}} - V(\mathbf{r}))^2, & V(\mathbf{r}) < E_{\text{thresh}} \\ 0, & V(\mathbf{r}) \geq E_{\text{thresh}} \end{cases} \quad (2.52)$$

$$V_{\text{ext}}(\mathbf{r}) = V(\mathbf{r}) + V_{\text{GaMD}}(V(\mathbf{r})) \quad (2.53)$$

The threshold energy E_{thresh} and the harmonic force constant k are determined based on the average and standard deviation of the system’s potential energy during an initial equilibration phase.

Uncertainty Sampling Iterations In order to run uncertainty-driven MD simulations, the biasing potentials from the eABF method and GaMD are combined with the original potential energy function $V(\mathbf{r})$ to obtain the total potential energy function $V_{\text{total}}(\mathbf{r})$ that governs the dynamics of the system:

$$V_{\text{total}}(\mathbf{r}) = V(\mathbf{r}) + V_{\text{eABF}}(\xi(\mathbf{r})) + V_{\text{GaMD}}(V(\mathbf{r})) \quad (2.54)$$

To initialize the uncertainty sampling, an initial training set is generated and labeled with the reference method. A single model is trained on this initial training set, and then the GMM is fitted to its latent feature vectors and calibrated to yield meaningful uncertainty estimates. In an active learning cycle, uncertainty sampling is then performed for multiple iterations using the MLP model as the underlying potential energy function. It is biased by the GaMD and eABF methods, with the GMM-provided uncertainty used as CV. The resulting trajectory is analyzed to identify high-uncertainty configurations above a certain threshold. A predefined number of these configurations with the highest structural diversity, quantified by the cosine distance between their latent feature vectors, is selected and labeled using the reference method. The model is retrained on the updated training set, and the GMM is refitted. This process is repeated until the number of generated configurations no longer exceeds the predefined threshold, indicating that the model has sufficiently learned the relevant regions of the phase space, or until a predefined maximum number of iterations is reached.

2.3. Quantum Mechanics

2.3.1. Schrödinger Equation

In contrast to classical mechanics, quantum mechanics describes the behavior of matter at the atomic scale, where particles exhibit wave-like properties expressed by the many-body wavefunction $\Psi(\mathbf{r}, t)$. The time-dependent Schrödinger equation describes the evolution of the wavefunction over time. For stationary states, the time-independent Schrödinger equation is used, which separates the spatial and temporal parts of the wavefunction:

$$\hat{H}(\mathbf{r}, t)\Psi(\mathbf{r}, t) = i\hbar \frac{\partial}{\partial t}\Psi(\mathbf{r}, t) \quad \text{Time-dependent Schrödinger equation} \quad (2.55)$$

$$\hat{H}(\mathbf{r})\Psi(\mathbf{r}) = E\Psi(\mathbf{r}) \quad \text{Time-independent Schrödinger equation} \quad (2.56)$$

$$\Psi(\mathbf{r}, t) = \Psi(\mathbf{r}) \exp\left(-\frac{iEt}{\hbar}\right) \quad \text{Separation of spatial and temporal part} \quad (2.57)$$

Here, $\hat{H}$ is the Hamiltonian operator, which is comprised of the kinetic energy operator $\hat{T}$ and the potential energy operator $\hat{V}$ terms for all particles in the system:

$$\hat{H} = \underbrace{-\frac{1}{2} \sum_A \frac{1}{M_A} \nabla_A^2}_{\hat{T}_N} - \underbrace{\frac{1}{2} \sum_i \nabla_i^2}_{\hat{T}_e} - \underbrace{\sum_A \sum_i \frac{Z_A}{r_{Ai}}}_{\hat{V}_{Ne}} + \underbrace{\sum_{i < j} \frac{1}{r_{ij}}}_{\hat{V}_{ee}} + \underbrace{\sum_{A < B} \frac{Z_A Z_B}{r_{AB}}}_{\hat{V}_{NN}} \quad (2.58)$$

Given the large mass difference between nuclei and electrons, the Born-Oppenheimer approximation allows separation of the nuclear and electronic wavefunctions. The electronic Schrödinger equation can then be solved for fixed nuclear positions.

Hartree-Fock

Ab initio methods are computational methods that solve the time-independent electronic Schrödinger equation from first principles without relying on empirical parameters. In contrast, semi-empirical methods use experimental data to simplify calculations. The Hartree-Fock method is an *ab initio* method that approximates the many-electron wavefunction using a single Slater determinant constructed from orthogonal one-electron molecular orbitals. The use of a Slater determinant instead of the simple product approach ensures the antisymmetry of the wavefunction and compliance with the Pauli exclusion principle. The molecular orbitals are expressed as Linear Combination of Atomic Orbitals (LCAO), which

are in turn expanded using a finite set of basis functions. The coefficients of the linear combination are optimized iteratively using the Roothaan-Hall equations and the Self-Consistent Field (SCF) procedure to minimize the total energy of the system. Here, starting from an initial guess for the density matrix containing the basis function coefficients, the Fock matrix is constructed. The Fock matrix is then diagonalized to obtain new molecular orbitals and a new density matrix. This process is repeated until convergence.

The Hartree-Fock method accounts for the average electron-electron repulsion but neglects electron correlation effects, since a single Slater determinant cannot capture instantaneous electron-electron interactions. This limitation can lead to significant errors in the calculated energies and properties. Post-Hartree-Fock methods have been developed to incorporate electron-correlation effects.²²

Configuration Interaction

In post-Hartree-Fock methods, multiple Slater determinants Φ are combined to approximate the many-electron wavefunction better Ψ and account for electron correlation effects. The determinants are generated by exciting electrons from occupied to virtual molecular orbitals in the Hartree-Fock reference determinant Φ_{HF} . The methods differ in the types and number of excitations they include.

$$\Psi = a_0 \Phi_{HF} + \sum_i a_i \Phi_i \quad (2.59)$$

Full Configuration Interaction (CI) includes all possible excitations within a given basis set. However, it is computationally expensive and scales factorially with the number of electrons and the number of basis functions. Truncated CI methods, such as CIS (singles), and CISD (singles and doubles), offer a compromise between accuracy and computational cost by limiting the excitations to a certain level. Still, they are not size-consistent, meaning that the energy does not scale correctly with the number of particles in the system. Additionally, according to Brillouin's theorem, single excitations do not contribute to the correlation energy when starting from a Hartree-Fock reference, making CIS pointless for ground state energy calculations, as it gives the same energy as Hartree-Fock.²²

Møller-Plesset Perturbation Theory

Møller-Plesset perturbation theory⁶³ is a post-Hartree-Fock method that treats electron correlation as a perturbation to the Hartree-Fock reference state. Energy and wavefunction are expanded in a perturbation series, with the Hartree-Fock energy as the zeroth-order term. Terms in the expansion are collected based on their order in the perturbation parameter. The first-order correction is zero due to Brillouin's theorem.

In contrast, the second-order correction (MP2) captures a significant portion of the correlation energy and is the most commonly used order. It scales as $O(N^5)$. Higher-order corrections provide increased accuracy but scale as $O(N^6)$ (MP3) and $O(N^7)$ (MP4).²²

Coupled Cluster

Coupled cluster methods^{64,65} provide a highly accurate approach to account for electron correlation by expressing the many-electron wavefunction Ψ using an exponential ansatz of excitation operators $\mathbf{T}_n$ acting on the Hartree-Fock reference determinant Φ_{HF} . The excitation operators generate all possible n -fold excited Slater determinants by promoting electrons from occupied to virtual molecular orbitals.

$$\mathbf{T} = \mathbf{T}_1 + \mathbf{T}_2 + \mathbf{T}_3 + \dots \quad (2.60)$$

$$\mathbf{T}_1 \Phi_{HF} = \sum_i^{\text{occ}} \sum_a^{\text{virt}} t_i^a \Phi_i^a \quad (2.61)$$

$$\mathbf{T}_2 \Phi_{HF} = \sum_{i < j}^{\text{occ}} \sum_{a < b}^{\text{virt}} t_{ij}^{ab} \Phi_{ij}^{ab} \quad (2.62)$$

$$\Psi_{CC} = e^{\mathbf{T}} \Phi_{HF} \quad (2.63)$$

$$e^{\mathbf{T}} = 1 + \mathbf{T} + \frac{1}{2!} \mathbf{T}^2 + \frac{1}{3!} \mathbf{T}^3 + \dots \quad (2.64)$$

Truncating the excitation operators to include only singles and doubles (CCSD) provides a good balance between accuracy and computational cost, scaling as $O(N^6)$. The inclusion of perturbative triples (CCSD(T)) further improves accuracy. Combined with complete basis set extrapolation, CCSD(T) is often referred to as the ‘gold standard’ of quantum chemistry.²²

2.3.2. Density Functional Theory

In contrast to wavefunction-based methods, Density Functional Theory (DFT) describes the electronic structure of a many-body system using the electron density $\rho(\mathbf{r})$ as the central quantity. The electron density is a three-dimensional function that gives the probability of finding an electron at a specific point in space $\mathbf{r}_1$. It is obtained by integrating the square of the many-electron wavefunction $\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ over all but one of the electron coordinates. The electron density is normalized to the total number of electrons N in the system:

$$\rho(\mathbf{r}_1) = N \int |\Psi(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_N)|^2 d\mathbf{r}_2 d\mathbf{r}_3 \dots d\mathbf{r}_N \quad (2.65)$$

$$\int \rho(\mathbf{r}_1) d\mathbf{r}_1 = N \quad (2.66)$$

According to the Hohenberg-Kohn mapping theorem⁶⁶, the ground-state electron density uniquely determines the properties of the system, including the total energy. Building on an orbital-free formulation by Thomas and Fermi, Kohn and Sham⁶⁷ reintroduced orbitals into DFT. This increases the complexity from 3 to 3N dimensions, but allows the exact treatment of the kinetic energy of non-interacting electrons. The Kohn-Sham equations describe a system of non-interacting electrons moving in an effective potential and the electron-electron interactions. They contain terms for the kinetic energy of non-interacting electrons $T_s[\rho]$, the electron-nucleus interaction energy $E_{\text{Ne}}[\rho]$, the classical Coulomb repulsion energy $J[\rho]$, and the exchange-correlation energy $E_{\text{xc}}[\rho]$, which accounts for the remaining quantum mechanical effects of electron-electron interactions, such as kinetic correlation energy, potential correlation and exchange energy, and self-interaction corrections:²²

$$E_{\text{KS}}[\rho] = T_s[\rho] + E_{\text{Ne}}[\rho] + J[\rho] + E_{\text{xc}}[\rho] \quad (2.67)$$

$$T_s = \sum_i^N \langle \psi_i | -\frac{1}{2} \nabla^2 | \psi_i \rangle \quad (2.68)$$

$$E_{\text{Ne}} = - \sum_A^M \int \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \rho(\mathbf{r}) d\mathbf{r} \quad (2.69)$$

$$J[\rho] = \frac{1}{2} \int \int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' \quad (2.70)$$

The exact form of the exchange-correlation functional $E_{\text{xc}}[\rho]$ is unknown and must be approximated. The approach for this approximation defines the different DFT functionals. Common approximations include the Local Density Approximation (LDA), which assumes a uniform electron density, the Generalized Gradient Approximation (GGA), which includes the gradient of the electron density, and hybrid functionals, which incorporate a portion of exact exchange from Hartree-Fock theory. Although DFT is not systematically improvable like wavefunction-based methods, the above methods can be viewed as a hierarchy of increasing accuracy and computational cost, known as Jacob's ladder⁶⁸ of DFT functionals.

Local Density Approximation The uniform electron gas model forms the basis of the LDA, which assumes that the exchange-correlation energy at each point in space depends only on the local electron

density $\rho(\mathbf{r})$. The exchange-correlation energy functional can be expressed in terms of the energy per particle $\epsilon_{xc}(\rho)$:

$$E_{xc}^{LDA}[\rho] = \int \rho(\mathbf{r}) \epsilon_{xc}[\rho(\mathbf{r})] d\mathbf{r} \quad (2.71)$$

$$\epsilon_{xc}[\rho] = \epsilon_x(\rho) + \epsilon_c(\rho) \quad (2.72)$$

$$\epsilon_x[\rho] = -\frac{3}{4} \left(\frac{3}{\pi} \right)^{1/3} \rho^{1/3} \quad (2.73)$$

The correlation part is often derived from quantum Monte Carlo simulations of the uniform electron gas. Suitable parameterizations include the Vosko-Wilk-Nusair⁶⁹ and Perdew-Wang⁷⁰ functionals. LDA performs well for solid-state systems with slowly varying electron densities but gives poor results for molecular systems with rapidly changing electron densities.²²

Generalized Gradient Approximation The GGA improves upon the LDA by incorporating the gradient of the electron density $\nabla\rho(\mathbf{r})$ into the exchange-correlation functional. This allows the functional to more accurately account for non-uniform electron densities. The general form of the GGA exchange-correlation functional is given by:

$$E_{xc}^{GGA}[\rho] = \int f(\rho(\mathbf{r}), \nabla\rho(\mathbf{r})) d\mathbf{r} \quad (2.74)$$

An early representative of GGA functionals is Becke's 1988 exchange functional (B88)⁷¹. A gradient-dependent correction is added to the LDA exchange energy:

$$\epsilon_x^{B88}[\rho] = \epsilon_x^{LDA}[\rho] - \Delta\epsilon_x^{B88}[\rho, \nabla\rho] \quad (2.75)$$

$$\Delta\epsilon_x^{B88}[\rho, \nabla\rho] = \beta \rho^{1/3} \frac{x^2}{1 + 6\beta x \sinh^{-1}(x)} \quad (2.76)$$

$$x = \frac{|\nabla\rho|}{\rho^{4/3}} \quad (2.77)$$

The Lee-Yang-Parr (LYP) correlation functional⁷² is often combined with B88 to form the BLYP functional. Another popular GGA functional is Perdew-Burke-Ernzerhof (PBE)⁷³. Climbing the Jacob's ladder, meta-GGA functionals include additional terms, such as the kinetic energy density or the Lapla-

cian of the electron density, to further improve accuracy. Examples of meta-GGA functionals include TPSS⁷⁴ and SCAN^{75, 22}

Hybrid Functionals A hybrid functional combines DFT exchange with a portion of exact exchange from Hartree-Fock theory to improve accuracy, e.g., the BHHLYP functional⁷⁶ uses 50% exact exchange, while the popular B3LYP functional^{77, 78} uses 20%. Other widely used hybrid functionals include PBE0⁷⁹ and the range-separated hybrid functional ω B97X^{80, 81}, which separate the exchange interaction into short-range and long-range components and treat them differently to improve accuracy for systems with varying electron-correlation lengths. An existing hybrid variant of a meta-GGA functional is TPSSH^{82, 83} which adds 10% exact exchange to the TPSS functional.

2.3.3. Basis Sets

Ab initio and DFT methods require the use of basis sets to represent molecular orbitals in LCAO form. Common types of basis functions include Slater-Type Orbitals (STOs) and Gaussian-Type Orbitals (GTOs).

$$\chi_{\text{STO}}(r) \propto e^{-\zeta r} \quad (2.78)$$

$$\chi_{\text{GTO}}(r) \propto e^{-\zeta r^2} \quad (2.79)$$

STOs accurately represent the behavior of atomic orbitals near the nucleus and at long distances, but are computationally expensive. GTOs are computationally more efficient, as the product of two Gaussian functions centered on different atoms can be expressed as a single Gaussian function centered at an intermediate point. However, they do not accurately capture the cusp behavior of atomic orbitals at the nucleus and decay too rapidly at long distances. To mitigate this, GTOs are often linearly combined to better approximate STOs. Different families of basis sets have been developed with their members varying in size and complexity to balance accuracy and computational cost. Examples include Pople basis sets^{84, 85} (e.g., STO-3G, 3-21G, 6-31G, 6-311G), Dunning's correlation-consistent basis sets⁸⁶ (e.g., cc-pVDZ, cc-pVTZ, cc-pVQZ), and the Karlsruhe def2 basis sets developed by Ahlrichs⁸⁷ (e.g., def2-SVP, def2-TZVP, def2-QZVP). Additional functions, such as polarization and diffuse functions, can be added to improve the flexibility of the basis set and better describe electron distribution in molecules.^{22, 88}

2.3.4. Semi-Empirical Methods

Quantum mechanical methods that use empirical parameters to simplify calculations and reduce computational cost are known as semi-empirical methods. They are often used for larger systems or when many calculations are required, such as in molecular dynamics simulations. Semi-empirical methods typically involve approximating the electronic Hamiltonian and using parameterized integrals derived from experimental data or high-level quantum-mechanical calculations. Examples of semi-empirical methods include the Neglect of Diatomic Differential Overlap (NDDO) family of methods, which includes MNDO⁸⁹, AM1⁹⁰, and PM3⁹¹, as well as the Density Functional Tight Binding (DFTB)^{92–96} and xTB^{97,98} methods, which are based on a tight-binding approximation and use parameters derived from DFT calculations.

DFTB

DFTB approximates the Kohn-Sham DFT (see Eq. (2.67)). Instead of finding the energy-minimizing electron density ρ self-consistently, DFTB approximates it from a reference density ρ_0 that is constructed from a superposition of atomic densities centered at the positions of the atoms in the molecule. A charge fluctuation term $\delta\rho$ is then added to account for the redistribution of electron density due to chemical bonding and interactions.

It is constructed as a superposition of atomic densities ρ_A^{atomic} centered at the positions of the atoms in the molecule, and a charge fluctuation term $\delta\rho$ caused by the redistribution of electron density due to chemical bonding and interactions:

$$\rho_0(\mathbf{r}) = \sum_A^M \rho_A^{atomic}(\mathbf{r}) \quad (2.80)$$

$$\rho(\mathbf{r}) \approx \rho_0(\mathbf{r}) + \delta\rho(\mathbf{r}) \quad (2.81)$$

The Kohn-Sham energy is expressed as a functional of the reference density ρ_0 and the charge fluctuation $\delta\rho$. The expansion of the exchange-correlation functional in a Taylor series around ρ_0 at different orders leads to different variants of DFTB. The first-order expansion, known as DFTB1⁹², includes only the zeroth and first-order terms. The second-order expansion, known as DFTB2⁹³, includes up to the second-order term, enabling the treatment of charge transfer and polarization effects. The third-order expansion, DFTB3⁹⁵, includes up to the third-order term, which includes changes in the chemical hardness.

$$\begin{aligned}
 E[\rho^0 + \delta\rho] = & \underbrace{\sum_i n_i \langle \psi_i | \hat{H}[\rho^0] | \psi_i \rangle}_{E_{\text{el}}} \\
 & - \underbrace{\frac{1}{2} \iint \frac{\rho^0(\mathbf{r}) \rho^0(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r} d\mathbf{r}' - \int V^{\text{xc}}[\rho^0] \rho^0(\mathbf{r}) d\mathbf{r} + E^{\text{xc}}[\rho^0] + E^{\text{nn}}}_{E_{\text{rep}}} \\
 & + \underbrace{\frac{1}{2} \iint \left(\frac{1}{|\mathbf{r} - \mathbf{r}'|} + \frac{\delta^2 E^{\text{xc}}[\rho]}{\delta\rho(\mathbf{r}) \delta\rho(\mathbf{r}')} \bigg|_{\rho^0} \right) \delta\rho(\mathbf{r}) \delta\rho(\mathbf{r}') d\mathbf{r} d\mathbf{r}'}_{\text{DFTB2}} \\
 & + \underbrace{\frac{1}{6} \iiint \frac{\delta^3 E^{\text{xc}}[\rho]}{\delta\rho(\mathbf{r}) \delta\rho(\mathbf{r}') \delta\rho(\mathbf{r}'')} \bigg|_{\rho^0} \delta\rho(\mathbf{r}) \delta\rho(\mathbf{r}') \delta\rho(\mathbf{r}'') d\mathbf{r} d\mathbf{r}' d\mathbf{r}''}_{\text{DFTB3}} \quad (2.82)
 \end{aligned}$$

DFTB1 only contains the electronic energy $E_{\text{el}}[\rho^0]$ and a repulsive energy term E_{rep} . The electronic energy is calculated from the Kohn-Sham equations using a LCAO approach with a minimal confined STOs basis and the variational principle to minimize the energy with respect to the orbital coefficients $c_{\mu i}$. No iterative steps until self-consistency are needed, as the reference density ρ_0 is fixed and does not depend on the orbital coefficients. Applying a one-center approximation for the diagonal elements and a two-center approximation for the off-diagonal elements of the Hamiltonian matrix, the electronic energy can be tabulated pairwise for each combination of elements as a function of the interatomic distance in the form of Slater-Koster tables. The tabulation makes the method computationally efficient, as E_{el} can be obtained from simple lookups. For E_{rep} , combining the nuclear-nuclear repulsion and the remaining terms of the Kohn-Sham energy, a pairwise potential is fitted to the difference between the reference energy and the E_{el} for a set of reference structures in the form of a distance-dependent spline function. The DFTB1 energy is then given by the sum of the electronic energy and the repulsive energy.^{94,96,99,100}

$$E_{\text{DFTB1}} = \sum_i \sum_{\mu\nu} n_i c_{\mu i} c_{\nu i} H_{\mu\nu}^0 + \frac{1}{2} \sum_{A \neq B} V_{AB}^{\text{rep}}(R_{AB}) \quad (2.83)$$

DFTB2, also known as SCC-DFTB, includes the second-order term in the Taylor expansion, which contains the charge fluctuation $\delta\rho$. It is modeled via a sum of atomic contributions $\delta\rho_a$ that use a Slater function scaled by Mulliken charges q_a (see Section 2.3.5) and a width parameter τ_a that is fitted to reproduce the atomic hardness quantified by the Hubbard parameter U :

$$\delta\rho(\mathbf{r}) = \sum_a^M \delta\rho_a(\mathbf{r}) \quad (2.84)$$

$$\delta\rho_a(\mathbf{r}) = q_a \frac{\tau_a^3}{8\pi} e^{-\tau_a r} \quad (2.85)$$

$$\tau_a = \frac{16}{5} U_a \quad (2.86)$$

The second-order term is approximated with an analytical function $\gamma_{ab}(R_{ab})$ that deviates from the $\frac{1}{R_{ab}}$ Coulomb interaction at short distances. At $R_{ab} \rightarrow 0$, it approaches the Hubbard parameter U_a for $a = b$. Using the γ function, the DFTB2 energy is given by:

$$E_{\text{DFTB2}} = E_{\text{DFTB1}} + \frac{1}{2} \sum_{ab} \gamma_{ab}(R_{ab}) q_a q_b \quad (2.87)$$

When applying the variational principle to optimize the orbital coefficients $c_{\mu i}$, the resulting Kohn-Sham-like equations for DFTB2 contain an additional term dependent on the Mulliken charge q_a , which in turn depends on the orbital coefficients, making the equations self-consistent.^{93,96,100}

DFTB3 includes the third-order term in the Taylor expansion, which accounts for changes in the chemical hardness: $\frac{\partial U_a}{\partial q_a}$. This allows anions to differ in their hardness from their neutral counterparts. The same approximations for the charge fluctuation $\delta\rho$ are used as in DFTB2, resulting in the following energy expression:

$$E_{\text{DFTB3}} = E_{\text{DFTB2}} + \frac{1}{3} \sum_{ab} q_a^2 q_b \Gamma_{ab} \quad (2.88)$$

The Γ function contains an analytical expression for the derivative of γ w.r.t. to the hardness U and the change in hardness with respect to the charge, which can be obtained from calculations of an atom and its ions. The resulting Kohn-Sham-like equations for DFTB3 still contain orbital coefficient-dependent terms, so they are also solved self-consistently.^{95,96,100}

Common parameterizations for DFTB include the mio set⁹³ and the 3OB set^{101–104}, which cover a wide range of elements and are suitable for organic and biomolecular systems.

2.3.5. Population Analysis

Population analysis refers to methods for assigning partial atomic charges to atoms in a molecule based on electronic structures obtained from quantum-mechanical calculations. These partial charges are useful for understanding the distribution of electron density within a molecule and for use in molecular simulations, such as MD with classical force fields. Different population analysis methods use different approaches to partition the electron density among the atoms in a molecule. Methods can be broadly categorized into wavefunction-based and electrostatic potential-based approaches. The description in Section 2.3.1 is relevant for wave-function-based methods.

$$\rho_i(\mathbf{r}) = \phi_i^2(\mathbf{r}) \quad \text{Electron density from atomic orbital } i \quad (2.89)$$

$$\phi_i = \sum_{\mu}^{M_{\text{basis}}} c_{i\mu} \chi_{\mu} \quad \text{Atomic orbital expansion} \quad (2.90)$$

$$\phi_i^2 = \sum_{\mu}^{M_{\text{basis}}} \sum_{\nu}^{M_{\text{basis}}} c_{i\mu} c_{i\nu} \chi_{\mu} \chi_{\nu} \quad (2.91)$$

$$\sum_i^{N_{\text{orb}}} n_i \int \phi_i^2 d\mathbf{r} = \sum_{\alpha\beta}^{M_{\text{basis}}} D_{\alpha\beta} S_{\alpha\beta} = N_{\text{elec}} \quad (2.92)$$

$$D_{\alpha\beta} = \sum_i^{N_{\text{orb}}} n_i c_{\alpha i} c_{\beta i} \quad \text{Density matrix element} \quad (2.93)$$

$$S_{\alpha\beta} = \int \chi_{\alpha} \chi_{\beta} d\mathbf{r} \quad \text{Overlap matrix element} \quad (2.94)$$

Mulliken

The Mulliken population analysis¹⁰⁵ is the earliest population analysis method and takes the simplest approach to partition the electron density as derived from the molecular wavefunction. It divides the electron density equally between the two atoms involved in each basis function overlap. The partial atomic charge q_A on atom A is calculated as follows:

$$\rho_A = \sum_{\alpha \in A} \sum_{\beta}^{M_{\text{basis}}} D_{\alpha\beta} S_{\alpha\beta} \quad (2.95)$$

$$q_A = Z_A - \rho_A \quad (2.96)$$

Löwdin

The Löwdin population analysis¹⁰⁶ improves upon the Mulliken method by using symmetric orthogonalization of the basis functions. The density matrix is transformed using the overlap matrix S to obtain an orthogonalized density matrix D' . The partial atomic charge q_A on atom A is then calculated using the diagonal elements of the orthogonalized density matrix:

$$D' = S^{1/2} D S^{1/2} \quad (2.97)$$

$$\rho_A = \sum_{\alpha \in A} D'_{\alpha\alpha} \quad (2.98)$$

Hirshfeld

Hirshfeld population analysis¹⁰⁷ is inspired by the concept of Atoms In Molecules (AIM), which partitions the electron density according to its topology rather than partitioning the basis functions. It constructs a promolecular electron density by superimposing the spherically averaged electron densities of isolated atoms ($\rho_A^{atomic}(\mathbf{r})$) at their positions in the molecule. The ratio of the atomic electron density to the promolecular density at each point in space is then used as a weight function to partition the total electron density $\rho(\mathbf{r})$ among the atoms:

$$\rho_{pro}(\mathbf{r}) = \sum_A^M \rho_A^{atomic}(\mathbf{r}) \quad (2.99)$$

$$w_A(\mathbf{r}) = \frac{\rho_A^{atomic}(\mathbf{r})}{\rho_{pro}(\mathbf{r})} \quad (2.100)$$

$$q_A = Z_A - \int w_A(\mathbf{r}) \rho(\mathbf{r}) d\mathbf{r} \quad (2.101)$$

Minimal-Basis Iterative Stockholder

Minimal-Basis Iterative Stockholder (MBIS) population analysis¹⁰⁸ is a variant of the Hirshfeld method that uses a minimal basis set of s-type Slater functions to represent the atomic electron densities. The parameters of the Slater functions are optimized iteratively best to reproduce the total electron density of the molecule. The optimization is performed based on the Kullback-Leibler divergence between the molecular electron density and the promolecular density constructed from the Slater functions. The partial atomic charges are then calculated similarly to the Hirshfeld method, using the optimized atomic electron densities.

Merz-Kollman and CHELPG

Electrostatic potential-based methods, such as the Merz-Kollman (MK) method^{109,110} and the CHarges from ELectrostatic Potentials using a Grid-based method (CHELPG) method¹¹¹, attempt to reproduce the molecular ElectroStatic Potential (ESP) by fitting point charges located on the atomic nuclei. The ESP is calculated at a set of grid points surrounding the molecule, and the partial atomic charges are optimized to minimize the difference between the quantum mechanically calculated ESP and the ESP generated by the fitted point charges.

$$\Phi(\mathbf{r}) = \sum_A^{nuclei} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}' \quad \text{Molecular ESP} \quad (2.102)$$

$$\Phi_{\text{model}}(\mathbf{r}) = \sum_A^{nuclei} \frac{q_A}{|\mathbf{r} - \mathbf{R}_A|} \quad \text{Model ESP from point charges} \quad (2.103)$$

$$\mathcal{L} = \frac{1}{N_{\text{grid}}} \sum_i^{N_{\text{grid}}} (\phi(\mathbf{r}_i) - \phi_{\text{model}}(\mathbf{r}_i))^2 \quad \text{Loss function to minimize} \quad (2.104)$$

MK places the grid points outside of the van-der-Waals surface of the molecule, while CHELPG uses a uniform grid surrounding the molecule. Both methods can include constraints to ensure that the total charge of the fitted point charges matches the total molecular charge.

RESP

The Restrained Electrostatic Potential (RESP) method¹¹² is an extension of the MK method that includes an additional hyperbolic regularization term in the fitting procedure to prevent overfitting and ensure chemically reasonable charges. The regularization penalizes large deviations of the fitted charges from zero. The loss function to be minimized in RESP includes both the difference between the quantum mechanical and model ESP and the regularization term:

$$\mathcal{L} = \frac{1}{N_{\text{grid}}} \sum_i^{N_{\text{grid}}} (\phi(\mathbf{r}_i) - \phi_{\text{model}}(\mathbf{r}_i))^2 + a \sum_A^{nuclei} \sqrt{q_A + b^2} - b \quad (2.105)$$

As a consequence, buried atoms with less influence on the ESP tend to have charges closer to zero compared to MK charges.

Electronegativity-Equalization Charges

Electronegativity-Equalization Method (EEM) is based on three assumptions: (1) Each atom has an effective charge-dependent electronegativity χ_i . (2) The effective electronegativity equalizes throughout the molecule. (3) The total charge of the molecule is conserved.^{113–115}

The effective electronegativity of an atom i can be expressed as a function of its partial atomic charge q_i and the partial atomic charges of all other atoms j in the molecule:

$$\chi_i = (\chi_i^0 + \delta\chi_i) + 2(\eta_i^0 + \delta\eta_i)q_i + \kappa \sum_{j \neq i} \frac{q_j}{R_{ij}} \quad (2.106)$$

$$\chi_1 = \chi_2 = \dots = \chi_{\text{molecule}} \quad (2.107)$$

$$\sum_i q_i = Q_{\text{total}} \quad (2.108)$$

χ_i : Effective electronegativity of atom i

χ_i^0 : Electronegativity of isolated atom i

$\delta\chi_i$: Electronegativity change due to chemical environment

η_i^0 : Hardness of isolated atom i

$\delta\eta_i$: Hardness change due to chemical environment

κ : Scaling parameter for Coulomb interaction

The parameters χ_i^0 , $\delta\chi_i$, η_i^0 , $\delta\eta_i$, and κ are typically determined empirically by fitting to a reference set of partial atomic charges obtained from quantum mechanical calculations. For χ_i^0 and η_i^0 , values based on Sanderson's electronegativity and Pearson's hardness concepts are used. MK charges are used as the reference charges for the fitting procedure for the parameters used in this work.¹¹⁴

The resulting system of linear equations can be solved to obtain the partial atomic charges q_i for all atoms in the molecule:

$$\begin{bmatrix} B_1 & \frac{\kappa}{R_{12}} & \dots & \frac{\kappa}{R_{1N}} & -1 \\ \frac{\kappa}{R_{21}} & B_2 & \dots & \frac{\kappa}{R_{2N}} & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\kappa}{R_{N1}} & \frac{\kappa}{R_{N2}} & \dots & B_N & -1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \chi_{\text{molecule}} \end{bmatrix} = \begin{bmatrix} -A_1 \\ -A_2 \\ \vdots \\ -A_N \\ Q_{\text{total}} \end{bmatrix} \quad (2.109)$$

2.4. Quantum Mechanics/Molecular Mechanics

While quantum mechanical methods provide accurate descriptions of molecular systems, their computational cost scales steeply with system size, making them impractical for large systems. Classical molecular mechanics methods, on the other hand, are computationally efficient but cannot accurately describe electronic structure and chemical reactions. Hybrid QM/MM methods serve as a compromise between these two approaches.^{4,16,116,117} They bridge the scale gap between feasible quantum-mechanical calculations and large-scale molecular simulations by combining quantum-mechanical and classical molecular-mechanics approaches. The system is partitioned into a QM region, where electronic-structure calculations are performed, and a MM region, which is treated with classical force fields. The total energy of the system is expressed as:

$$E_{\text{total}} = E_{\text{QM}} + E_{\text{MM}} + E_{\text{QM/MM}} \quad (2.110)$$

$$E_{\text{QM/MM}} = E_{\text{QM/MM}}^{\text{elec}} + E_{\text{QM/MM}}^{\text{vdW}} + E_{\text{QM/MM}}^{\text{covalent}} \quad (2.111)$$

E_{QM} : Energy of the QM region calculated using quantum mechanics

E_{MM} : Energy of the MM region calculated using molecular mechanics

$E_{\text{QM/MM}}$: Interaction energy between the QM and MM regions

The interaction energy $E_{\text{QM/MM}}$ can include electrostatic interactions $E_{\text{QM/MM}}^{\text{elec}}$, van der Waals interactions $E_{\text{QM/MM}}^{\text{vdW}}$, and covalent interactions $E_{\text{QM/MM}}^{\text{covalent}}$ between the QM and MM regions. Van der Waals interactions are typically modeled using Lennard-Jones potentials, while covalent interactions are treated using link atoms to cap dangling bonds at the QM/MM interface. Both of these interaction types are handled by the MM force field. Electrostatic interactions can be treated in different ways, depending on whether the MM point charges are included in the QM Hamiltonian (electrostatic embedding) or not (mechanical embedding).

Mechanical embedding treats the QM and MM regions as separate entities; the electrostatic interactions between them are calculated after the individual energies have been computed. The QM region is not influenced by the MM environment during the quantum mechanical calculation.

Electrostatic embedding enables polarization of the QM region by the MM environment, yielding more accurate results. The polarization of the QM region is achieved by including the MM point charges in the QM Hamiltonian. During the SCF procedure, the electron density of the QM region is optimized in the presence of the electrostatic potential generated by the MM point charges. This allows the QM electron density to respond to the local environment created by the MM region. E_{QM} and $E_{\text{QM/MM}}^{\text{elec}}$ can then be obtained as eigenvalues of the effective QM Hamiltonian.^{45,118}

$$E_{\text{QM}} + E_{\text{QM/MM}}^{\text{elec}} = \langle \Psi | \hat{H}_{\text{QM}} + \sum_j^{MM} q_j v_{MM}(\mathbf{r}_j) | \Psi \rangle \quad (2.112)$$

$$v_{MM}(\mathbf{r}_j) = \kappa \sum_i^{QM} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}_j|} - \kappa \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_j|} d\mathbf{r}' \quad (2.113)$$

$v_{MM}(\mathbf{r}_j)$: Electrostatic potential at MM point charge j due to QM zone

Electrostatic embedding is the most commonly used embedding scheme in QM/MM simulations, as the additional computational cost of the QM calculation is minimal relative to the improved accuracy.

A third embedding scheme, polarizable embedding, goes beyond electrostatic embedding by allowing for mutual polarization between the QM and MM regions. However, this approach is more complex and computationally demanding, and it is less commonly used in practice.

2.5. Machine Learning

The term ‘machine learning’ encompasses a broad range of algorithms and techniques that enable computers to learn patterns from data and make predictions for new, unseen data. Methods range from simple linear regression to complex deep learning architectures, large language models, and generative models. In computational chemistry, machine learning can be used to predict molecular properties or to generate new molecular structures. The following sections provide a brief introduction to the fundamental concepts of machine learning, with a focus on artificial neural networks, which form the basis for the MLP models used in this work.

2.5.1. Artificial Neural Networks

Neurons Neurons or nodes are the fundamental building blocks of neural networks. They receive input signals x_i , apply a linear transformation, and then pass the result through a non-linear activation function σ to produce an output signal a . The output of a neuron can be expressed mathematically as a weighted sum of the inputs plus a bias term, followed by the activation function:

$$z = \sum_{i=1}^n w_i x_i + b \quad (2.114)$$

$$a = \sigma(z) \quad (2.115)$$

The weights w_i and bias b are learnable parameters adjusted during training to minimize the loss function.

Layers Multiple neurons are organized into layers, with the output of one layer serving as the input to the next layer. The first layer is called the input layer, the last layer is called the output layer, and any layers in between are called hidden layers. The mathematical operation performed by a layer in vectorized form can be expressed as:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l-1)} \mathbf{a}^{(l-1)} \quad (2.116)$$

$$\mathbf{a}^{(l)} = \sigma(\mathbf{z}^{(l)}) \quad (2.117)$$

The dimensions of the weight matrix $\mathbf{W}^{(l-1)}$ depend on the number of neurons in the current layer l and the previous layer $l-1$: $\mathbf{W}^{(l-1)} \in \mathbf{R}^{n_l \times (n_{l-1}+1)}$, where n_l is the number of neurons in layer l and n_{l-1} is the number of neurons in layer $l-1$. The bias term is typically included as an additional column in the weight matrix. The input vector $\mathbf{a}^{(l-1)} \in \mathbf{R}^{n_{l-1}+1}$ includes an additional element with a value of 1 to account for the bias.

Activation Functions Activation functions introduce non-linearity into the neural network, allowing it to learn complex patterns in the data. Common activation functions include the sigmoid function, hyperbolic tangent (tanh) function, Rectified Linear Unit (ReLU)/Exponential Linear Unit (ELU)/Sigmoid Linear Unit (SiLU) function. An activation function must be smooth and differentiable to enable the use of gradient-based optimization methods during training. Vanishing gradients can arise when using saturating activation functions (e.g., sigmoid and tanh), preventing the network from learning effectively, especially in deeper networks.¹¹⁹ Non-saturating activation functions (e.g., ReLU) can help mitigate this issue, although ReLU has a discontinuous derivative at $z=0$. Variants of ReLU, such as Leaky ReLU and ELU, address this issue by allowing for a small non-zero gradient when $z < 0$. A summary of common activation functions and their derivatives is provided in Table 2.1.¹¹⁹

Loss The loss function measures the difference between the neural network's predicted output and the true target values. The choice of loss function depends on the type of problem being solved. For regression problems, common loss functions include Mean Squared Error (MSE) and Mean Absolute Error (MAE), whereas for classification problems, common loss functions include cross-entropy/log loss.

Table 2.1.: Common activation functions, their gradients, and output ranges.

Name	Definition	Derivative	Range
Linear	$\sigma(z) = z$	$\sigma'(z) = 1$	$(-\infty, \infty)$
Sigmoid	$\sigma(z) = 1/(1 + e^{-z})$	$\sigma'(z) = \sigma(z)(1 - \sigma(z))$	$(0, 1)$
Tanh	$\sigma(z) = \tanh(z)$	$\sigma'(z) = 1 - \tanh^2(z)$	$(-1, 1)$
ReLU	$\sigma(z) = \max(0, z)$	$\sigma'(z) = \begin{cases} 0 & z < 0 \\ 1 & z > 0 \end{cases}$	$(0, \infty)$
Leaky ReLU	$\sigma(z) = \begin{cases} z & z > 0 \\ \alpha z & z < 0 \end{cases}$	$\sigma'(z) = \begin{cases} 1 & z > 0 \\ \alpha & z < 0 \end{cases}$	$(-\infty, \infty)$
ELU	$\sigma(z) = \begin{cases} z & z > 0 \\ \alpha(e^z - 1) & z < 0 \end{cases}$	$\sigma'(z) = \begin{cases} 1 & z > 0 \\ \alpha e^z & z < 0 \end{cases}$	$(-\alpha, \infty)$
Swish/SiLU ¹²⁰	$\sigma(z) = z/(1 + e^{-z})$	$\sigma'(z) = \sigma(z) + (1 - \sigma(z))/(1 + e^{-z})$	$(-0.28, \infty)$
Softplus	$\sigma(z) = \ln(1 + e^z)$	$\sigma'(z) = 1/(1 + e^{-z})$	$(0, \infty)$

$$\text{MSE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.118)$$

$$\text{MAE}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.119)$$

Another useful metric, in addition to the loss, is the R^2 score, which quantifies the ratio of the error range to the data range. An R^2 score of 1 means perfect correlation, 0 means the prediction is as good as simply taking the mean, and a negative R^2 score implies that the prediction is worse than taking the mean:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (2.120)$$

Backpropagation Backpropagation¹²¹ is the heart of the training process for neural networks. It is an efficient algorithm for computing the gradients of the loss function with respect to the network's weights and biases. The algorithm works by applying the chain rule to propagate errors from the later layers back to the earlier layers, since the gradients in the earlier layers depend on those in the later layers. For a simple feedforward neural network with two layers, as shown in Figure 2.3, the backpropagation algorithm can be expressed mathematically as follows:

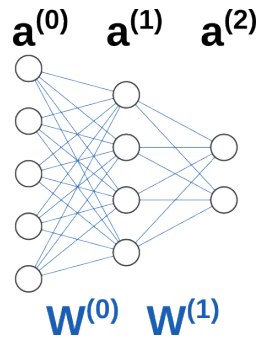


Figure 2.3.: Schematic representation of a neural network.

$$\begin{aligned}
 \mathbf{a}^{(2)} &= \sigma(\mathbf{z}^{(2)}) = \sigma(\mathbf{W}^{(1)}\mathbf{a}^{(1)}) = \sigma(\mathbf{W}^{(1)}\sigma(\mathbf{z}^{(1)})) = \sigma(\mathbf{W}^{(1)}\sigma(\mathbf{W}^{(0)}\mathbf{a}^{(0)})) \\
 \mathcal{L} &= \text{MSE}(\mathbf{a}^{(2)}, y) \\
 \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(2)}} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{z}^{(2)}} \cdot \frac{\partial \mathbf{z}^{(2)}}{\partial \mathbf{W}^{(1)}} \\
 \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(0)}} &= \frac{\partial \mathcal{L}}{\partial \mathbf{a}^{(2)}} \cdot \frac{\partial \mathbf{a}^{(2)}}{\partial \mathbf{z}^{(2)}} \cdot \frac{\partial \mathbf{z}^{(2)}}{\partial \mathbf{a}^{(1)}} \cdot \frac{\partial \mathbf{a}^{(1)}}{\partial \mathbf{z}^{(1)}} \cdot \frac{\partial \mathbf{z}^{(1)}}{\partial \mathbf{W}^{(0)}} \\
 &= \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(1)}} \cdot \frac{\partial \mathbf{a}^{(1)}}{\partial \mathbf{z}^{(1)}} \cdot \frac{\partial \mathbf{z}^{(1)}}{\partial \mathbf{W}^{(0)}}
 \end{aligned}$$

The gradients are then used to update the weights and biases in mini-batches over multiple epochs using an optimization algorithm such as stochastic gradient descent or Adam¹²² with a specified learning rate η :

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} \quad (2.121)$$

Best Practice The data used to train a neural network are typically split into three subsets: the training set, the validation set, and the test set. The training set is used to optimize the network weights and biases, while the validation set is used to monitor the loss during training and select the epoch with the best performance. By stopping training early, this practice helps prevent overfitting, a situation in which the model learns to fit the training data very well but fails to generalize to new, unseen data. The test set is used to evaluate the final performance of the trained model, completely independent of the training process.¹¹⁹

To further mitigate overfitting, regularization techniques can be employed. Regularization adds a penalty term to the loss function that penalizes overfitting to the training data. Common regularization techniques include L1 regularization (Lasso), which adds a penalty proportional to the absolute value of the weights, and L2 regularization (Ridge), which adds a penalty proportional to the squared weights.¹¹⁹

Scaling the input features and target values can be beneficial for training neural networks, as it can improve convergence and prevent issues arising from different scales in the input data. Common scaling techniques include min-max scaling and standardization. The former rescales the features to a specified range, typically [0, 1], whereas the latter transforms the features to have a mean of 0 and a standard deviation of 1.

2.5.2. Machine Learning Potentials

Many different MLP architectures have been developed in recent years.^{43,123–131} They predict molecular properties such as energies, forces, and dipole moments based on the atomic positions and elements. They differ in their molecular and atomic representations, model architectures, available properties, and application domains. The following sections provide an overview of the representations and model architectures used in MLP development, with a focus on those employed in this work.

Molecular Representation

Molecules require a numerical representation that captures relevant information about their structure and composition to serve as input to MLP models. The fixed size of the input representation is a requirement for neural networks, which causes a challenge for representing molecules of varying sizes. Different approaches have been developed to address this issue.

Cartesian Coordinates Cartesian coordinates are the most straightforward molecular representation, but they suffer from transferability issues due to the fixed size of the input representation. They are not invariant to translations and rotations of the molecule, which can lead to different predictions for the same molecule in different orientations. Therefore, Cartesian coordinates are a poor choice for representing molecules.

Internal Coordinates Internal coordinates, such as bond lengths, bond angles, and dihedral angles, provide a more general representation of a molecule that is invariant to translations and rotations. However, they can still suffer from transferability issues due to the fixed size of the input representation. Internal coordinates are also not unique: different sets of internal coordinates can represent the same molecule, leading to ambiguity in the representation.

Coulomb Matrix The Coulomb matrix represents the pairwise interactions between atoms in a molecule. It is constructed by calculating the Coulomb interactions between all pairs of atoms, resulting in a square matrix:¹³²

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4} & i = j \\ \frac{Z_i Z_j}{|\mathbf{R}_i - \mathbf{R}_j|} & i \neq j \end{cases} \quad (2.122)$$

The upper-triangular part of the Coulomb matrix can be flattened to form a fixed-size input vector for the neural network. It is invariant to translations and rotations of the molecule. Still, it is not invariant under permutations of the atomic indices, which can lead to different predictions for the same molecule under different atom orderings. The closely related inverse distance matrix, which uses the inverse of interatomic distances rather than Coulomb interactions, suffers from the same issues.

Bag of Bonds The Bag of Bonds (BoB) representation is inspired by the natural language processing technique of bag-of-words and the Coulomb matrix. It groups the Coulomb interactions between atoms into ‘bags’ according to the elements involved. The bags are then sorted, padded to a fixed (dataset-dependent) size to accommodate molecules of varying sizes, and concatenated to create a fixed-size input vector for the neural network. This representation is invariant under translations, rotations, and permutations of the atom indices, owing to the sorting and grouping of the interactions.¹³³

Many-Body Tensor Representation The final product of the Many-Body Tensor Representation (MBTR) is a histogram. For each many-body term, a distribution of broadened geometric feature (such as interatomic distances for 2-body terms or angles for 3-body terms) is constructed for each element combination $(z_1, \dots, z_k)$:

$$f_k(x, z_1, \dots, z_k) = \sum_{i_1, \dots, i_k} w_k(i_1, \dots, i_k) \mathcal{N}(x | g_k, \sigma) \prod_{j=1}^k \delta(z_j, Z_{i_j}) \quad (2.123)$$

The distributions get discretized into a fixed number of bins to create a fixed-size input vector. The MBTR representation is invariant to translations, rotations, and permutations of the atom indices.^{134,135}

Smooth Overlap of Atomic Positions The Smooth Overlap of Atomic Positions (SOAP) representation is more closely related to quantum mechanical methods, as it is based on the idea of representing the local atomic environment as a continuous density of neighboring atoms. The local atomic density is expanded in a basis of radial basis functions $g_n(\mathbf{r})$ and angular spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$. The expansion coefficients c_{nlm}^i are obtained by projecting the local atomic density onto the basis functions. They are then used to construct rotationally invariant descriptors by taking the power spectrum of the expansion coefficients.^{135,136}

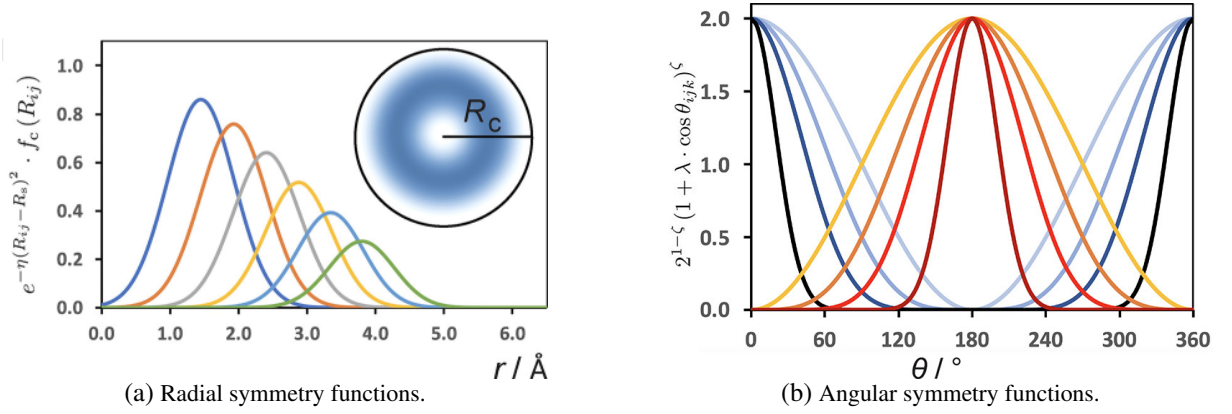


Figure 2.4.: Radial and angular symmetry functions with different parameters. Reprinted (adapted) with permission from Jörg Behler. „Four Generations of High-Dimensional Neural Network Potentials“. In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 10037–10072. DOI: 10.1021/acs.chemrev.0c00868. Copyright 2026 American Chemical Society.

$$\rho_i(\mathbf{r}) = \sum_{n,l,m} c_{nlm}^i g_n(\mathbf{r}) Y_{lm}(\hat{\mathbf{r}}) \quad (2.124)$$

$$c_{nlm}^i = \langle g_n Y_{lm} | \rho_i \rangle \quad (2.125)$$

$$p_{nn'l}^i = \sum_{m=-l}^l c_{nlm}^i (c_{n'l m}^i)^* \quad (2.126)$$

Atom-Centered Symmetry Functions The Atom-Centered Symmetry Functions (ACSF) representation is based on the idea of describing the local environment of each atom in a molecule using a set of Gaussian and cosine functions to capture the radial and angular distribution of neighboring atoms. A cutoff function ensures smooth decay to zero of the contributions from atoms beyond a certain distance:

$$G_{i,m,Z}^{\text{radial}} = \sum_{j \neq i}^{N_Z} e^{-\eta_m (R_{ij} - R_{s,m})^2} f_c(R_{ij}) \quad (2.127)$$

$$G_{i,m,Z_1,Z_2}^{\text{angular}} = 2^{1-\zeta_m} \sum_{j \neq i}^{N_{Z_1}} \sum_{k \neq i,j}^{N_{Z_2}} (1 + \lambda_m \cos \theta_{ijk})^{\zeta_m} e^{-\eta_m (R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \quad (2.128)$$

$$f_c(R) = \begin{cases} 0.5 \left[\cos \left(\frac{\pi R}{R_c} \right) + 1 \right] & R \leq R_c \\ 0 & R > R_c \end{cases} \quad (2.129)$$

Different combinations of the parameters η , R_s , λ , and ζ can be used to create a set m of symmetry functions that capture different aspects of the local atomic environment. Similar to the BoB, the ACSF

representation sorts the symmetry functions based on the involved element Z (radial part) or element pair Z_1Z_2 (angular part) and concatenates them. The input vector for an atom with index 0 in a dataset containing hydrogen, carbon, and oxygen could look like:

$$\mathbf{G}_0 = [\mathbf{G}_{0,H} || \mathbf{G}_{0,C} || \mathbf{G}_{0,O} || \mathbf{G}_{0,H,H} || \mathbf{G}_{0,H,C} || \mathbf{G}_{0,H,O} || \mathbf{G}_{0,C,C} || \mathbf{G}_{0,C,O} || \mathbf{G}_{0,O,O}]$$

$||$ denotes concatenation. Each $\mathbf{G}_{0,Z}$ and $\mathbf{G}_{0,Z_1,Z_2}$ is a vector of symmetry function values for the different parameter combinations m . Due to the summation over neighboring atoms, the ACSF representation is invariant to translations, rotations, and permutations of the atom indices.

Graph Representation In a molecular graph representation, atoms are nodes, and bonds between atoms are edges.¹³⁸ Properties can be assigned at the graph (total energy, total charge, total spin, total dipole moment), node (position, element, atomic charge), and edge level (bond type, bond length, bond order). Graph-based MLP models typically use message passing to propagate information stored in the feature vector $\mathbf{h}_i$ for each node i throughout the graph. Initially, the feature vector can be initialized using an embedding of the corresponding atom’s atomic number z_i . Over T message passing steps, the feature vector gets updated based on the information from neighboring nodes $j \in \mathcal{N}(i)$. In the simplest case, the message passing function $\Phi_{M,t}$ and the update function $\Phi_{U,t}$ can be simple weighted sums, but in most cases, at least a distance-based weighting is included. The final energy prediction for each node i is obtained by applying a readout function $\mathcal{R}_t$ to the node’s feature vector at each message passing step and summing over all steps. The total energy prediction for the molecule is then obtained by summing over the energy predictions for all nodes.¹³¹

$$\sigma_i^{(t)} = (\mathbf{r}, z_i, \mathbf{h}_i^{(t)}) \quad \text{State of node } i \text{ at step } t \quad (2.130)$$

$$\mathbf{h}_i^{(0)} = \sum_k^{N_{\text{elements}}} \mathbf{W}_k \delta(z_i, k) \quad \text{Initial node feature vector} \quad (2.131)$$

$$\mathbf{m}_i^{(t)} = \bigoplus_{j \in \mathcal{N}(i)} \Phi_{M,t}(\sigma_i^{(t)}, \sigma_j^{(t)}) \quad \text{Message passing} \quad (2.132)$$

$$\mathbf{h}_i^{(t+1)} = \Phi_{U,t}(\sigma_i^{(t)}, \mathbf{m}_i^{(t)}) \quad \text{Feature vector update} \quad (2.133)$$

$$E_i = \sum_{t=1}^T \mathcal{R}_t(\sigma_i^{(t)}) \quad \text{Energy prediction for node } i \quad (2.134)$$

$$E_{\text{total}} = \sum_i E_i \quad \text{Total energy prediction} \quad (2.135)$$

The exact form of the message passing function M_t , the update function U_t , and the readout function $\mathcal{R}_t$ can vary between different graph-based MLP models, leading to different architectures and performance characteristics. The graph representation is invariant to translations, rotations, and permutations of the atom indices, due to message passing and the fact that the graph structure itself does not depend on the ordering of the nodes.

Equivariance Graph-based MLP models can be designed to be equivariant to certain transformations, such as rotations and translations. A representation is equivariant to a transformation if the output of the representation transforms in the same way as the input when the transformation is applied. In contrast, a representation is invariant to a transformation if the output does not change when the transformation is applied to the input. For example, two points in 3D space can be represented by their distance, which is invariant under rotations and translations, since the result is the same whether one measures the distance before or after the rotation or translation. However, the distance vector between the two points is equivariant to rotations and translations, as it will change in the same way as the points when a rotation or translation is applied. The mathematical definition of equivariance of a feature vector $\mathbf{h}$ to a transformation Q can be expressed as:^{130,131,139}

$$\mathbf{h}(Q\mathbf{r}) = D(Q)\mathbf{h}(\mathbf{r}) \quad (2.136)$$

Q : Transformation applied to the input

$D(Q)$: Corresponding transformation applied to the output

Equivariant models embed directional information into feature vectors and preserve it during message passing, enabling them to capture geometric relationships between atoms more effectively. This can lead to improved performance, especially for properties that are sensitive to the spatial arrangement of atoms, such as forces and dipole moments. An especially important group of transformations for molecular systems is the $SO(3)$ group, which comprises all possible rotations in three-dimensional space. Models that are equivariant to the $SO(3)$ group can capture the full rotational symmetry of molecular systems.^{130,131,139}

Model Architectures in Vacuum

The development of MLP models has been particularly focused on gas-phase and periodic systems, where the absence of an explicit solvent environment enables a more straightforward application. The following sections provide an overview of some of the most prominent MLP architectures developed for gas-phase.

High-Dimensional Neural Network Potentials Behler and Parrinello pioneered the MLP scene with the High-Dimensional Neural Network Potential (HDNNP) architecture. It is based on the idea of decomposing the total energy of a molecule into a sum of atomic contributions, each predicted by an individual elemental neural network Φ_{Z_i} for each atom. The input of each atom to each elemental neural network is a set of ACSF values that describe the local environment of the atom. The total energy of the molecule is then obtained by summing the contributions from all atoms:^{123,128,137}

$$E_{\text{total}} = \sum_{i=1}^N E_i = \sum_{i=1}^N \Phi_{Z_i}(\mathbf{G}_i) \quad (2.137)$$

The HDNNP architecture has been further developed over generations, with the first generation referring to an earlier use of an MLP¹⁴⁰. The description so far corresponds to the 2nd generation HDNNP, which is still widely used and serves as the basis for many other MLP architectures. Its architecture is depicted in Figure 2.5a.¹²³ A charge prediction is added in the 3rd generation HDNNP¹⁴¹ to allow for the calculation of long-range electrostatic interactions beyond the cutoff radius of the ACSF representation, which is particularly important for charged and polar systems. A second set of elemental neural networks is used to predict the atomic charges q_i from the same ACSF input as in the energy prediction. The total energy can then be obtained by summing the electrostatic energy E_{elec} and the atomic contributions from the elemental neural networks, now considered as short-range energy E_{short} . The electrostatic energy is calculated using Gaussian charge densities centered on the atoms with the predicted charges q_i and covalent radii as the width σ_i .¹²⁸

$$E_{\text{total}} = E_{\text{short}} + E_{\text{elec}} \quad (2.138)$$

$$E_{\text{short}} = \sum_{i=1}^N \Phi_{Z_i}(\mathbf{G}_i) \quad (2.139)$$

$$E_{\text{elec}} = \sum_{i=1}^N \sum_{j>i}^N \kappa \frac{\text{erf}\left(\frac{R_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{R_{ij}} q_i q_j + \sum_{i=1}^N \kappa \frac{q_i^2}{2\sqrt{\pi}\sigma_i} \quad (2.140)$$

$$\gamma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2}; \quad \kappa = \frac{1}{4\pi\epsilon_0} \quad (2.141)$$

A method for charge conservation is introduced in the 4th Gen. High-Dimensional Neural Network Potential (4G-HDNNP)¹²⁸ (see Figure 2.5b). The mechanism uses the Charge Equilibration via neural Network Technique (CENT)^{142,143}. Instead of using the second set of elemental neural networks to predict the atomic charges, the CENT uses it to predict an intermediate quantity interpreted as the electronegativity of each atom χ_i . By tailoring the energy term in the short-range elemental neural networks

to the charges, these electronegativity values are then used to determine the optimal charge distribution that minimizes the electrostatic energy. The parameters for hardness J_{ii} and the Gaussian width σ_i can be fixed values or optimized as weights during training. The total charge of the system is conserved with a Lagrange multiplier in the resulting linear equation system:¹²⁸

$$E_{\text{total}} = E_{\text{elec}} + \sum_{i=1}^N (E_i^0 + \chi_i q_i + \frac{1}{2} J_{ii} q_i^2) \quad (2.142)$$

$$\frac{\partial E_{\text{total}}}{\partial q_k} = 0 \Rightarrow \chi_k + \sum_{i=1}^N A_{ki} q_i = 0 \quad (2.143)$$

$$A_{ki} = \begin{cases} J_{kk} + \kappa \frac{1}{\sqrt{\pi} \sigma_k} & k = i \\ \kappa \frac{\text{erf}\left(\frac{R_{ki}}{\sqrt{2} \gamma_{ki}}\right)}{R_{ki}} & k \neq i \end{cases} \quad (2.144)$$

$$\sum_{i=1}^N q_i = Q_{\text{total}} \quad (2.145)$$

$$\left(\begin{array}{cccc|c} A_{11} & A_{12} & \cdots & A_{1N} & 1 \\ A_{21} & A_{22} & \cdots & A_{2N} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} & 1 \\ \hline 1 & 1 & \cdots & 1 & 0 \end{array} \right) \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \lambda \end{pmatrix} = \begin{pmatrix} -\chi_1 \\ -\chi_2 \\ \vdots \\ -\chi_N \\ Q_{\text{total}} \end{pmatrix} \quad (2.146)$$

SchNet SchNet¹²⁴ is one of the first popular MLP architectures that use a graph representation of molecules, introduced in Section 2.5.2. It uses continuous-filter convolutional layers to perform message passing between nodes in the graph. These convolutional layers generalize the discrete convolutional layers used in image processing. They work initially very similarly to radial ACSFs by using Gaussian radial basis functions to expand the interatomic distances with a uniform grid of centers μ , a width γ , a cutoff radius r_c , and an offset μ_{off} . The expanded distances $\mathbf{e}_{ij}$ get passed through a neural network Φ_{filter} to create a continuous filter, and the filter gets applied to the representation of neighboring nodes during message passing:¹⁴⁴

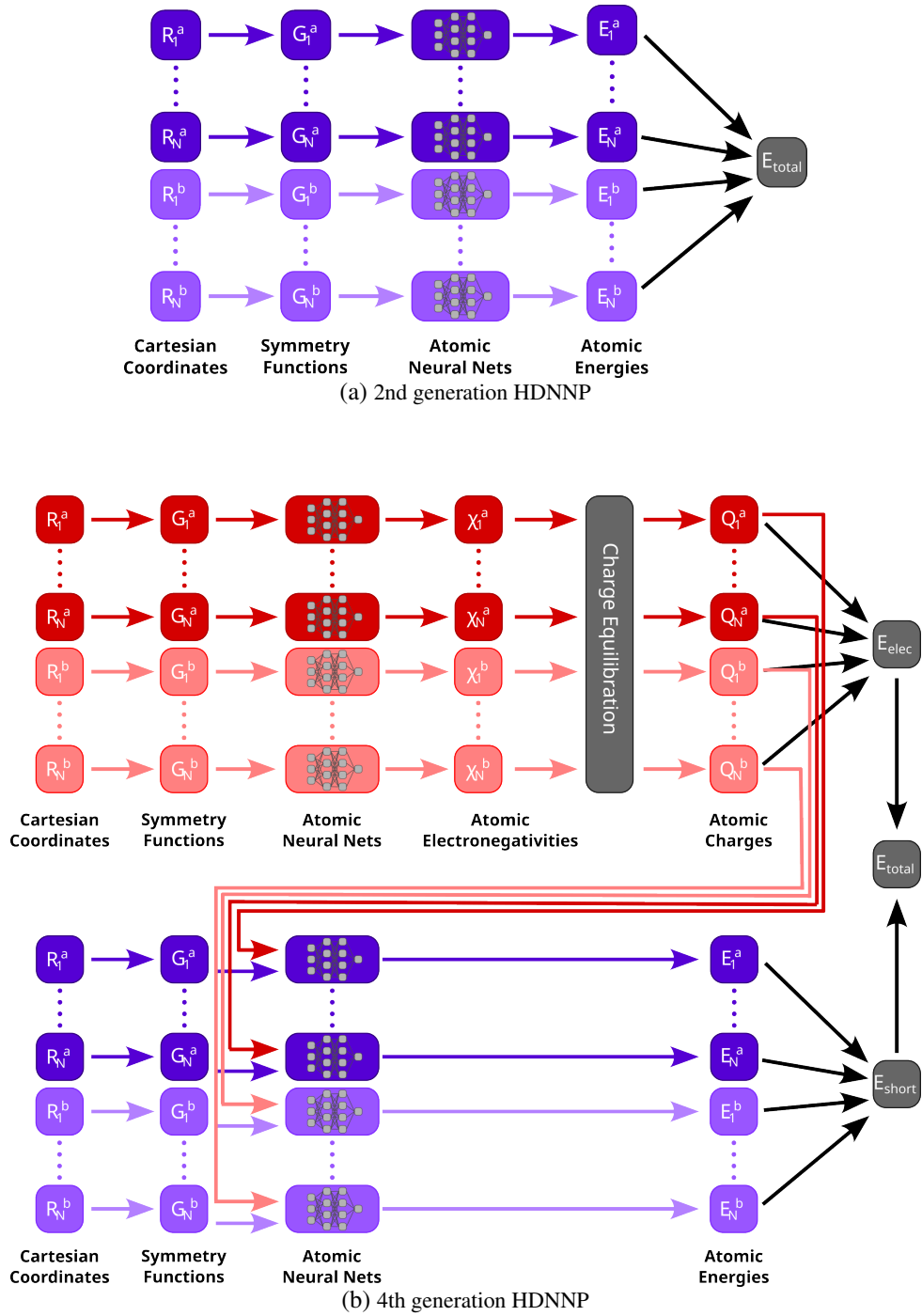


Figure 2.5.: Schematic representation of the 2nd and 4th generation HDNNP architectures.

$$\mu = \left[0, \frac{r_c}{n}, \frac{2r_c}{n}, \dots, r_c \right] \quad \text{Uniform grid with } n \text{ bins} \quad (2.147)$$

$$e_k(r_{ij}) = \exp(-\gamma(r_{ij} - \mu_{off} - \mu_k)^2) \quad \text{Gaussian radial basis functions} \quad (2.148)$$

$$\mathbf{W}^{(t)}(r_{ij}) = \Phi_{\text{filter}}(\mathbf{e}_{ij}) \quad \text{Continuous filter} \quad (2.149)$$

$$\mathbf{m}_i^{(t)} = \sum_{j \in \mathcal{N}(i)} \mathbf{W}^{(t)}(r_{ij}) \odot \Phi_M(\mathbf{h}_j^{(t)}) \quad \text{Message passing} \quad (2.150)$$

$$\mathbf{h}_i^{(t+1)} = \mathbf{h}_i^{(t)} + \Phi_U(\mathbf{m}_i^{(t)}) \quad \text{Feature vector update} \quad (2.151)$$

$$E_{\text{total}} = \sum_{i=1}^N \mathcal{R}(\mathbf{h}_i^{(T)}) \quad \text{Energy prediction} \quad (2.152)$$

$\odot$ denotes element-wise multiplication.

DimeNet The expansion with Gaussian radial basis functions on a uniform grid requires a relatively large number of bins to capture distance information with sufficient resolution. DimeNet¹³⁹ addresses this issue by using a Bessel basis for the distance expansion, which needs fewer bins to capture the same amount of information. An envelope function f_c with an envelope exponent p is used as a cutoff function to ensure smooth decay to zero. The cutoff function is also twice continuously differentiable at the cutoff radius, which is important for the calculation of forces and other derivatives. Additionally, DimeNet incorporates angular information into the message-passing process by using spherical harmonics to expand the angles between neighboring atoms.

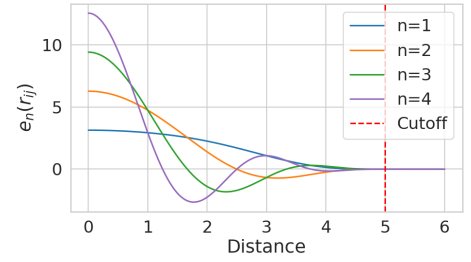


Figure 2.6.: Bessel basis functions with $r_c = 6$ and $p = 5$.

$$e_n(r_{ij}) = \sqrt{\frac{2}{r_c}} \frac{\sin\left(\frac{n\pi r_{ij}}{r_c}\right)}{r_{ij}} f_c(r_{ij}) \quad \text{Bessel basis function} \quad (2.153)$$

$$f_c(r) = 1 - \frac{(p+1)(p+2)}{2} r^p + p(p+2) r^{p+1} - \frac{p(p+1)}{2} r^{p+2} \quad \text{Cutoff function} \quad (2.154)$$

PaiNN PaiNN¹²⁹ is the equivariant extension of SchNet. It uses a similar architecture to SchNet, but it employs scalar $\mathbf{h}_{i,s}$ and vector node features $\mathbf{h}_{i,v}$ and uses continuous-filter convolutional layers that are designed to preserve the equivariance of the features during message passing. The equivariant filters are derived as gradients of the invariant filters used in SchNet. Vector messages are constructed from two terms: scaled vector features with invariant filters and scalar features with equivariant filters:^{129,144}

$$\nabla W_{vs}(r_{ij}) = W'_{vs}(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \quad \text{Equivariant filters} \quad (2.155)$$

$$\mathbf{m}_{i,v}^{(t)} = \sum_{j \in \mathcal{N}(i)} h_{j,v}^{(t)} \odot \Phi_{vv}(h_{j,s}^{(t)}) \odot W_{vv}(r_{ij}) + \sum_{j \in \mathcal{N}(i)} \Phi_{vs}(h_{j,s}^{(t)}) \odot W'_{vs}(r_{ij}) \frac{\mathbf{r}_{ij}}{r_{ij}} \quad \text{Vector messages} \quad (2.156)$$

MACE MACE¹³¹ is an equivariant graph neural network architecture that is inspired by Neural equivariant Interatomic Potential (NequIP)¹³⁰ and other equivariant models. Its main feature is higher-order message passing, which effectively passes messages between more than just pairs of nodes.

$$\mathbf{m}_i^{(t)} = \sum_j \mathbf{u}_1(\sigma_i^{(t)}, \sigma_j^{(t)}) + \sum_{j_1, j_2} \mathbf{u}_2(\sigma_i^{(t)}, \sigma_{j_1}^{(t)}, \sigma_{j_2}^{(t)}) + \cdots + \sum_{j_1, \dots, j_k} \mathbf{u}_k(\sigma_i^{(t)}, \sigma_{j_1}^{(t)}, \dots, \sigma_{j_k}^{(t)}) \quad (2.157)$$

This reduces the number of message-passing steps required to reach a given interaction order. A reduced number of message-passing steps improves the model's computational efficiency, as message passing must be performed iteratively, whereas higher-order message passing can be parallelized. Given its importance to this work, the MACE architecture is described in more detail here.

As an equivariant model, MACE embeds higher-dimensional features into the node feature vectors. As with spherical harmonics, the features are indexed by a rotation order l and a rotation index m . Scalar features with $l = 0$ are invariant to rotations, while vector features with $l = 1$ and higher-order features with $l > 1$ are equivariant to rotations. In its irreducible representation, a 3D matrix with $l = 2$ has 5 features corresponding to the 5 independent components of a rank-2 tensor in three-dimensional space. Equivariant message passing is achieved by using tensor products and Clebsch-Gordan coefficients to combine the features of neighboring nodes in a way that preserves the equivariance of the resulting features. During the initialization of each node feature vector $\mathbf{h}_i^{(0)}$, the element z_i of the corresponding node i is linearly embedded into the scalar features with $l = 0$ (and $m = 0$) in each index k (channel) of the feature vector. The higher-order features with $l > 0$ are initialized to zero and are populated with a weighted mixture of the scalar features at each message-passing step t .

$$h_{i,k00}^{(0)} = \sum_z^{N_{\text{elements}}} W_{kz} \delta(z_i, z) \quad \text{Initialization of scalar features} \quad (2.158)$$

$$\bar{h}_{i,kl_2m_2}^{(t)} = \sum_{\bar{k}} W_{\bar{k}kl_2}^{(t)} h_{i,\bar{k}l_2m_2}^{(t)} \quad \text{Weighted mixing of features} \quad (2.159)$$

To include the distance information in the message passing, a radial basis R is constructed from a neural network Φ_{Radial} that takes an embedding of the interatomic distance r_{ij} as input. The distance embedding $\mathbf{j}_0$ is obtained by applying a set of radial basis functions to the interatomic distance, typically Bessel functions with different frequencies n and a cutoff function to ensure smooth decay to zero of the contributions from atoms beyond a certain distance. To construct the message from node j to node i , a tensor product is applied to the features vector $\mathbf{h}_j$ and the radial basis R_{ij} . Its indices will be outlined in the next paragraph.

$$e_n(r_{ij}) = \sqrt{\frac{2}{r_c}} \frac{\sin\left(\frac{n\pi r_{ij}}{r_c}\right)}{r_{ij}} f_c(r_{ij}) \quad \text{Distance embedding} \quad (2.160)$$

$$R_{k\eta_1 l_1 l_2 l_3}^{(t)} = \Phi_{Radial}(\mathbf{e}_{ij}) \quad \text{Radial basis} \quad (2.161)$$

Clebsch-Gordan coefficients C and spherical harmonics Y are used to combine the features of neighboring nodes in a way that preserves the equivariance of the resulting features. Different paths are possible when multiplying tensors. The result of a tensor product might preserve or change the rotation order l , e.g., the tensor product of two scalar features with $l = 0$ is a scalar feature with $l = 0$. In comparison, the dot product of two vector features with $l = 1$ will result in a scalar feature with $l = 0$, and the cross product of two vector features will result in a vector feature with $l = 1$. The indices η_1 and l_3 enumerate the path and the resulting rotation order of the resulting features.^{131,145}

$$\phi_{ij,k\eta_1 l_3 m_3}^{(t)} = \sum_{l_1 l_2 m_1 m_2} C_{\eta_1, l_1 m_1 l_2 m_2}^{l_3 m_3} R_{k\eta_1 l_1 l_2 l_3}^{(t)}(r_{ij}) \otimes Y_{l_1}^{m_1}(\hat{\mathbf{r}}_{ij}) \bar{h}_{j,kl_2 m_2}^{(t)} \quad \text{One-Particle Basis} \quad (2.162)$$

The one-particle basis ϕ is used to construct another intermediate basis called the atomic basis A by applying a linear transformation that also sums the various paths η_1 and the features from different neighboring nodes j . The core of the MACE architecture, the higher-order message passing, is applied to the atomic basis A by taking the tensor product with itself up to a body order v . The resulting product basis $\mathbf{A}$ contains many-body interactions between the neighboring nodes with the rotation order array $\mathbf{l}$ and $\mathbf{m}$. The multiple paths to combine the rotation order arrays from the tensor product indexed by η_v are again symmetrically combined using general Clebsch-Gordan coefficients C , resulting in the fully symmetric basis $\mathbf{B}$.

$$A_{i,kl_3m_3}^{(t)} = \sum_{\tilde{k}, \eta_1} W_{k\tilde{k}\eta_1l_3}^{(t)} \sum_{j \in \mathcal{N}(i)} \phi_{ij,k\eta_1l_3m_3}^{(t)} \quad \text{Atomic Basis} \quad (2.163)$$

$$\mathbf{A}_{i,k\mathbf{lm}}^{(t),\nu} = \prod_{\xi=1}^{\nu} A_{i,kl_{\xi}m_{\xi}}^{(t)} \quad \text{Product Basis from } \nu \text{ Tensor Products} \quad (2.164)$$

$$\mathbf{B}_{i,\eta_{\nu}kLM}^{(t),\nu} = \sum_{\mathbf{lm}} C_{\eta_{\nu}\mathbf{lm}}^{LM} A_{i,k\mathbf{lm}}^{(t),\nu} \quad \text{Fully Symmetric Basis} \quad (2.165)$$

With the fully symmetric basis $\mathbf{B}$, the message passing can be performed by applying a linear transformation to the features in the basis and summing over the different body orders ν and paths η_{ν} . The resulting message $\mathbf{m}_i$ and the current node features $\mathbf{h}_i$ are then used to update the node features for the next message passing step $t + 1$. The energy prediction for each node i at each message passing step t is obtained by applying a readout function $\mathcal{R}$ to the scalar features with $l = 0$ (and $m = 0$) in the node feature vector. For the final energy prediction at the last message-passing step, a network Φ_{readout} is used to combine the scalar features from all channels k rather than a simple linear transformation. The total energy prediction for each node i is then obtained by summing over the energy predictions from all T message passing steps, and the total energy prediction for the molecule is obtained by summing over the energy predictions for all nodes. The forces are calculated as the negative gradient of the total energy with respect to the atomic positions:¹³¹

$$m_{i,kLM}^{(t)} = \sum_{\nu} \sum_{\eta_{\nu}} W_{z_i\eta_{\nu}kL}^{(t),\nu} \mathbf{B}_{i,\eta_{\nu}kLM}^{(t),\nu} \quad \text{Message construction}$$

$$h_{i,kLM}^{(t+1)} = \sum_{\tilde{k}} W_{kL,\tilde{k}}^{(t)} m_{i,\tilde{k}LM}^{(t)} + \sum_{\tilde{k}} W_{kz_iL,\tilde{k}}^{(t)} h_{i,\tilde{k}LM}^{(t)} \quad \text{Update}$$

$$E_i^{(t)} = \mathcal{R}(\mathbf{h}_i^{(t)}) = \begin{cases} \sum_{\tilde{k}} W_{\tilde{k}}^{(t)} h_{i,\tilde{k}00}^{(t)} & , \text{if } t < T \\ \Phi_{\text{readout}}^{(t)}(\{h_{i,\tilde{k}00}^{(t)}\}_k) & , \text{if } t = T \end{cases} \quad \text{Energy Readout}$$

$$E_i = \sum_{t=0}^T E_i^{(t)}$$

$$E = \sum_i E_i$$

$$\mathbf{F} = -\nabla \sum_i E_i$$

Model Architectures in Condensed Phase

Most MLPs are developed solely for vacuum applications, but more and more research is being done to extend their applicability to condensed-phase systems. Different approaches have been proposed to achieve this.^{6,146}

1. Full QM treatment: Both solute and solvent are treated at the same QM level of theory. This is the most straightforward approach, but generating training data can be computationally expensive, especially for larger solvent systems.^{47,147}
2. Δ -learning⁸: An MLP is trained to learn the difference between a cheaper method (e.g., DFTB) and a more accurate method (e.g., DFT) in vacuum, and then applied to the solvent system. This is based on the assumption that the solvent effect is captured well by the cheaper method, and the MLP only needs to learn the vacuum correction.^{118,148,149}
3. Mechanical embedding: The MLP interacts with the solvent through classical electrostatic interactions with fixed partial charges. Polarization effects are not captured via this approach.^{150–152}
4. Polarizable-Corrected Mechanical Embedding (PCME): Just like the mechanical embedding, the MLP is trained on vacuum data, but a *post hoc* correction is added to approximate environment-induced polarization effects on the solute. This can be done largely physically motivated, which aids transferability.^{153,154}
5. Environment-Integrated Embedding (EIE): The environment is directly integrated into the MLP training by embedding the solvent molecules in the input representation resembling standard QM/MM. This allows the MLP to learn the solvent effect directly from the data. Transferability might only be given for similar solvent environments.^{155,156}

In this thesis, the focus is on the EIE approach, as it enables the MLP to learn the solvent effect directly from data.

High-Dimensional Neural Network Potential Charge prediction in the 4G-HDNNP serves as the entry point for environmental interaction. As part of this thesis, the architecture is extended to incorporate the solvent environment into charge prediction via ESP and its gradient, which is the negative of the electric field. This leads to a modified total energy expression and, consequently, a modified CENT to ensure charge prediction in the presence of environmental polarization. The key addition is the addition of a term for the electrostatic interaction between the predicted atomic charges q_i and the environment-induced ESP ϕ via Coulomb interaction. As the ESP is a function of the atomic positions $\mathbf{r}$, but the model directly receives the ESP values at the atomic positions as input, the forces are calculated using the chain rule to account for the implicit dependence of the ESP on the atomic positions. This is achieved

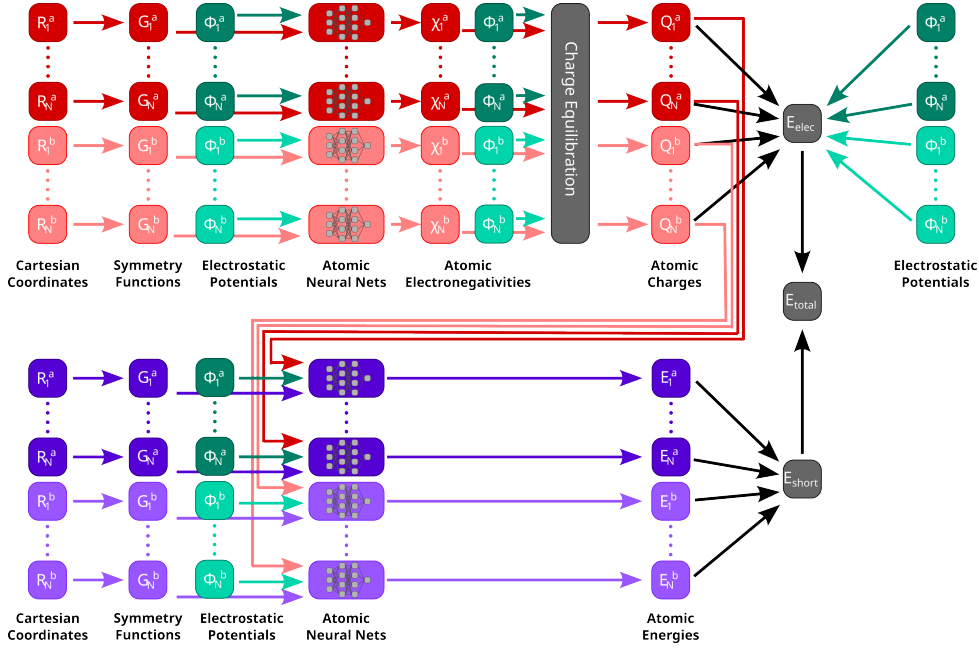


Figure 2.7.: Schematic representation of the 4th generation HDNNP architecture with the addition of the environment-induced ESP as input to the charge prediction.

by also providing the gradient of the ESP w.r.t. atomic positions as input to the model. The ESP is also concatenated to the ACSF input to the elemental neural networks. The updated architecture is schematically depicted in Figure 2.7. The total energy expression is modified as follows:

$$\begin{aligned}
 E_{total}(\mathbf{r}, \phi) &= E_{short} + E_{elec, QM} + E_{elec, QMMM} \\
 E_{elec, QMMM} &= \sum_{i \in QM}^{N_{QM}} q_i \phi_i = \sum_{i \in QM}^{N_{QM}} q_i \sum_{j \in MM}^{N_{MM}} \frac{q_j}{|r_i - r_j|} \\
 \frac{\partial E_{total}}{\partial q_k} &= 0 \rightarrow \chi_k + \phi_k + \sum_{i=1}^N A_{ik} q_i = 0 \\
 \sum_{i=1}^N q_i &= Q_{total} \\
 \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NN} \\ \hline 1 & 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \\ \hline \lambda \end{pmatrix} &= \begin{pmatrix} -\chi_1 - \phi_1 \\ -\chi_2 - \phi_2 \\ \vdots \\ -\chi_N - \phi_N \\ \hline Q_{total} \end{pmatrix} \\
 -\mathbf{F} &= \frac{\partial E_{total}(\mathbf{r}, \phi(\mathbf{r}))}{\partial \mathbf{r}} = \frac{\partial E_{total}}{\partial \mathbf{r}} + \frac{\partial E_{total}}{\partial \phi} \cdot \frac{\partial \phi}{\partial \mathbf{r}}
 \end{aligned}$$

Charge Equilibration MACE MACE is one of the most popular choices for MLP architectures but lacks an easily available implementation for condensed-phase systems, so a charge-equilibration mechanism similar to that in the 4G-HDNNP is implemented in the MACE architecture. It is developed as part of this thesis under the name Charge-Equilibration MACE (QEq-MACE). The charge equilibration mechanism is implemented as additional readout functions $\mathcal{R}_{\text{CENT}}$ that work exactly like the energy readout functions. They are applied to the scalar features $h_{i,k00}$ from each message passing step t to predict the electronegativity χ_i for each atom i . The predicted electronegativity values are then used to determine the optimal charge distribution, and the resulting charges are used to calculate the electrostatic energy from Gaussian charge densities, as described for the 4G-HDNNP, and to add it to the total energy prediction. The parameters for the width σ of the Gaussian charge densities are initialized as the covalent radii of each element. The chemical hardness J_{ii} is randomly initialized with a positive value between 0 and $10 \text{ eV}/\text{e}^2$. Both parameters are optimized during training.

In contrast to the 4G-HDNNP, the ESP is only part of the right-hand side of the linear equation system for charge equilibration, but not embedded in the input representation of the model. The ESP embedding was included as an additional weighted contribution during scalar feature initialization, and it was found to be neither beneficial nor detrimental to model training. The decision to not include the ESP in the input representation was made to keep the architecture more transferable and physically motivated.

FieldSchNet FieldSchNet¹⁵⁵ is an extension of SchNet that incorporates environmental effects by including the interaction between predicted atomic dipoles and external fields ϵ_{ext} in the energy prediction. For this purpose, an additional three-dimensional feature vector is constructed for each node i representing an atomic dipole moment μ_i , which also undergoes message passing and updates. Traditional SchNet messages m_i are interpreted as charge-charge interaction. The interactions are inspired by electrostatic physics, but no physical interaction term contributes directly to the prediction of the total energy. Response properties next to the forces as derivatives of the energy are available this way, e.g., the molecular dipole moment as the derivative of the energy w.r.t. the external electric field:¹⁵⁵

$\mu_i^{(0)} = \mathbf{0}$	Initialization of dipole features
$\mu_i^{(t+1)} = \mu_i^{(t)} + \sum_{j \in \mathcal{N}(i)} \Phi_q^{(t)}(\mathbf{h}_j^{(t)}) \mathbf{r}_{ij} f_c(r_{ij})$	Dipole feature update
$\mathbf{v}_i^{(t)} = \Phi_\mu^{(t)} \left(\sum_{j \in \mathcal{N}(i)} \mu_j^{(t)} \mathbf{T}^{(t)}(\mathbf{r}_{ij}) \mu_j^{(t)} \right)$	Dipole-dipole interaction
$\mathbf{T} = \frac{3\mathbf{r}_{ij}^2 \mathbf{I} - \mathbf{r}_{ij} \otimes \mathbf{r}_{ij}}{r_{ij}^5} \mathbf{W}_\mu^{(t)}(r_{ij})$	Dipole-dipole interaction tensor
$\mathbf{u}_i^{(t)} = \Phi_\epsilon^{(t)} \left(\mu_i^{(t)} \epsilon_{\text{ext}}(\mathbf{r}_i) \right)$	Dipole-field interaction
$\mathbf{h}_i^{(t+1)} = \mathbf{h}_i^{(t)} + \mathbf{m}_i^{(t)} + \mathbf{v}_i^{(t)} + \mathbf{u}_i^{(t)}$	Feature update
$E_{\text{total}} = \sum_{i=1}^N \mathcal{R}(\mathbf{h}_i^{(T)})$	Energy prediction

Anisotropic Message Passing The Anisotropic Message Passing (AMP) architecture^{156–158} combines equivariant message passing with a physically motivated electrostatic interaction via multipole expansion. To achieve this combination, the model uses typical node $\mathbf{h}_i$ and edge feature vectors $\mathbf{e}_{ij}$, message passing Φ_m , feature vector update Φ_u , and energy readout $\mathcal{R}$ functions to predict part of the total energy, denoted as V_{AMP} . In addition, the model adds a long-range electrostatic energy term $V_{\text{ES,QM}}$ from electrostatic interaction within the QM-zone, but only beyond the cutoff. This calculation is a Coulomb interaction of predicted partial charges/atomic monopoles:¹⁵⁶

$$V_{\text{QM}} = V_{\text{AMP}} + V_{\text{ES,QM}} \quad (2.166)$$

$$V_{\text{AMP}} = \sum_{i=1}^N \mathcal{R}(\mathbf{h}_i^{(T)}) \quad (2.167)$$

$$\mathbf{h}_i^{(t+1)} = \Phi_u \left(\mathbf{h}_i^{(t)}, \sum_{j \in \mathcal{N}(i)} \Phi_m(\mathbf{h}_i^{(t)}, \mathbf{h}_j^{(t)}, \mathbf{e}_{ij}) \right) \quad (2.168)$$

$$V_{\text{ES,QM}} = \sum_{i>j} (1 - f_{\text{switch}}) \cdot \kappa \frac{q_i q_j}{r_{ij}} \quad (2.169)$$

The prediction of the atomic monopoles q_i and higher multipoles is achieved by constructing many radial basis functions R_{ij} from k -th order tensor products of interatomic distances r_{ij} . The combined radial basis functions of all neighboring nodes j with a predicted coefficient form the atomic multipole moments M_i^k for each atom i and order k :

$$\mathbf{R}_{ij}^k = \hat{r}_{ij} \otimes \hat{r}_{ij} \otimes \dots$$

Tensor product of order k

$$\mathbf{R}_{ij}^2 = \begin{pmatrix} x^2 & xy & xz \\ xy & y^2 & yz \\ xz & yz & z^2 \end{pmatrix}$$

Local basis of order 2

$$\mathbf{M}_i^k = \sum_{j \in \mathcal{N}(i)} c_{ij}^k \mathbf{R}_{ij}^k$$

Atomic multipole construction

$$c_{ij}^k = \phi_{M(k)}(h_i \mid h_j \mid \mathbf{e}_{ij})$$

Contribution to the polarization of node i

The environment-dependent contribution to the polarization is added as a function of the node's polarizability α_i , an orientation-dependent factor b_{ij} , and the electric field of the environment at the position of node i . The model predicts α_i and b_{ij} as functions of the node, Bessel-function-embedded edge features e_{ij} , and multipole coefficients G_{ij} , which are explained in the next paragraph. The environment-induced polarization is then added to QM multipole moments M_i^k to get the final multipole moments $M_i^{k,final}$:

$$\mathbf{M}_{i, \text{QM-MM}}^k = \sum_i^{N_{QM}} \sum_j^{N_{MM}} \alpha_i b_{ij} \mathbf{R}_{ij}^k \frac{\mathbf{M}_j^0}{r_{ij}^2}$$

MM-multipole contribution

$$\alpha_i = \Phi_{\alpha_i^k}(h_i)$$

Per-atom polarizability

$$b_{ij} = \Phi_b(\mathbf{e}_{ij}, G_{ij})$$

Distance and orientation dependency

$$\mathbf{M}_{i, \text{final}}^k = \mathbf{M}_i^k + \mathbf{M}_{i, \text{QM-MM}}^k$$

Multipole adjustment due to MM-polarization

The electrostatic QM/MM interaction is calculated using multipole interaction between the QM multipole moments $M_i^{k,final}$ and the MM monopoles M_j^0 :

$$\begin{aligned}
 V_{\text{ES,QMMM}} &= \kappa \sum_{l=0}^2 \sum_i^{N_{\text{QM}}} \sum_j^{N_{\text{MM}}} B_{ij}^l G_{ij}^l && \text{Monopole-multipole interaction up to order 2} \\
 B_{ij}^l &= \frac{(2l-1)!!}{r_{ij}^{l+1}} && \text{Distance dependency} \\
 G_{ij}^0 &= M_i^0 M_j^0 && \text{Monopole-monopole interaction} \\
 G_{ij}^1 &= (M_i^{1,\alpha}, R_{ij}^\alpha) M_j^0 && \text{Monopole-dipole interaction} \\
 G_{ij}^2 &= (M_i^{2,\alpha\beta}, R_{ij}^{\alpha\beta}) M_j^0 && \text{Monopole-quadrupole interaction} \\
 (M_i^{2,\alpha\beta}, R_{ij}^{\alpha\beta}) &= \sum_\alpha \sum_\beta M_i^{2,\alpha\beta} R_{ij}^{\alpha\beta} && \text{Tensor contraction}
 \end{aligned}$$

FieldMACE FieldMACE¹⁵⁹ combines the ideas of FieldSchNet with the MACE architecture, but implements it differently from FieldSchNet. Instead of including field information, the MM atoms get featurized into k' channels (typically $k' < k$) in a similar way as the QM atoms. The QM features $\mathbf{h}_{i,klm}^{\text{QM}}$ are projected into the same feature space as the MM features $\mathbf{h}_{j,k'lm}^{\text{MM}}$ via a linear transformation. A pairwise attention network Φ_ξ is then applied to the L1-normalized difference features of the QM and MM features to get an attention score $\kappa_{ijk'}$ for each pair of QM and MM features. The score is then used to weight the contribution of multipole interactions between the multipolar QM features and the monopole MM charges to form a $\mathbf{m}_{i,k'lm}^{\text{QM-MM}}$ interaction message. The multipole expansion used here expresses Coulomb's law with the distance dependence $\frac{1}{r_{ij}}$ in terms of spherical harmonics using the discrete MM charges. The details about this step are explained in the original paper¹⁵⁹. The resulting QM-MM message is then added to the QM message from the MACE architecture to incorporate environmental effects into message passing, and the model proceeds to the usual update and readout steps. No explicit electrostatic interaction term is added to the total energy prediction; the environment effect only enters the model via featurization:¹⁵⁹

$$\begin{aligned}
 \mathbf{h}_{j,k'lm}^{\text{MM}} &= \mathbf{W}_{k'lm} Y_l^m && \text{Featurization of MM atoms} \\
 \mathbf{h}_{i,k'lm}^{\text{QM}} &= \mathbf{W}_{k'lm} \mathbf{h}_{i,klm}^{\text{QM}} && \text{Projection of QM features to MM feature space} \\
 \kappa_{ijk'}^{(t)} &= \Phi_\xi \left(\left| \mathbf{h}_{i,k'lm}^{\text{QM}} - \mathbf{h}_{j,k'lm}^{\text{MM}} \right| \right) && \text{Attention score for QM-MM feature pairs} \\
 \mathbf{m}_{i,klm}^{(t)} &= \mathbf{m}_{i,klm}^{(t),\text{QM}} + W_{kk'lm}^{(t)} \mathbf{m}_{i,k'lm}^{\text{QM-MM}} && \text{Message update with QM-MM features}
 \end{aligned}$$

Other ML/MM Architectures A comprehensive review about other ML/MM architecture, such as BurNN¹⁶⁰, EMLE¹⁵⁴ or the approaches by Lei, Yagi, and Sugita¹⁶¹ or Semelak et al.¹⁶² has been published by Grassano et al.⁶.

3. Data and Methods

3.1. Workflow

This section outlines the general Workflow used to develop neural network potentials for Quantum Mechanics/Molecular Mechanics (QM/MM) simulations. Substantial work has been invested in establishing an efficient and flexible workflow that enables the generation of diverse training datasets, the training of various neural network architectures, and the application of these models to QM/MM simulations. The general workflow comprises four main steps: data generation, data preprocessing, training of neural network potentials, and model application. An overview of the workflow is depicted in Figure 3.1.

In the data generation step, various sampling methods are employed to generate diverse molecular structures for two different systems: alanine dipeptide and thiol disulfide exchange reactions, as described in Section 3.2. Each structure is then subjected to single-point calculations using different reference methods, including Density Functional Theory (DFT) and Density Functional Tight Binding (DFTB), to obtain ground-truth data for training neural network potentials, as detailed in Section 3.3. Extracting the relevant information from the raw calculation outputs, primarily geometries, energies, and forces, yields the raw datasets.

In the data preprocessing step, the raw data sets are processed to prepare them for training the neural network potentials. For Δ -learning models, it is necessary to assert the convergence of corresponding calculations. Unit standardization is also an important aspect of data preprocessing. While vacuum Machine Learning Potentials (MLPs) are rather resistant to different unit systems, ML/MM outputs a sum of physical energies, which requires consistent units throughout the workflow. This work employs different model architectures, each requiring specific input formats (see Section 3.6 for details on the software used). Models trained via the Keras Convolutional Graph Neural Network (KGCNN)¹⁴⁴ require pickle files, Base MACE¹³¹, QEq-MACE, and FieldMACE¹⁵⁹ use extxyz files, SchNetPack¹⁶³/FieldSchNet¹⁵⁵ use SQL-like db format from Atomic Simulation Environment (ASE), and Anisotropic Message Passing (AMP)¹⁵⁶ uses the HDF5 format for storing and NumPy npy format for loading data. The data preprocessing step converts the raw data sets into the required formats for each model architecture. For Δ -learning models, the differences between the reference method and the baseline method (DFTB) are calculated and stored as target values for training.

For the model training step, neural network potentials are trained on the preprocessed datasets. Different architectures are explored, including 2nd Gen. High-Dimensional Neural Network Potential (2G-HDNNP), 4th Gen. High-Dimensional Neural Network Potential (4G-HDNNP), SchNet, PaiNN, Base MACE, QEq-MACE, AMP, and FieldMACE. The impact of different hyperparameters on model performance is investigated during a hyperparameter search. The training procedures and hyperparameter choices are detailed in Section 3.4.

Finally, the trained neural network potentials are applied to perform QM/MM simulations of the alanine dipeptide and thiol disulfide exchange reactions in their respective environments. The performance of the models is evaluated by their accuracy in reproducing reference energies and forces, as well as by their computational efficiency. The details of the simulation setups are discussed in Section 3.5.

3.2. Sampling Methods

Alanine Dipeptide

Metadynamics Simulations Alanine dipeptide was simulated in a cubic box with an edge length of 3.0 nm, containing ~ 900 water molecules, using the Amber99SB-ILDN force field¹⁶⁴ and the TIP3P water model¹⁶⁵. QM/MM calculations were performed using DFTB3^{93,101} with 3OB parameters¹⁰¹ for the alanine dipeptide. Non-bonded interactions were treated with a cutoff of 1.0 nm and Particle Mesh Ewald (PME) with a MM charge scale factor of 1.0 for long-range electrostatics.

The system was optimized for 50000 steps using the steepest-descent algorithm, followed directly by the production metadynamics simulations. The simulations were performed for 10 ns with a time step of 1 fs in the NVT ensemble at a temperature of 300 K in one simulation and 500 K for a second simulation using the Nose-Hoover thermostat. The backbone dihedral angles ϕ and ψ were biased using metadynamics with a hill height of 0.5 kJ/mol and a hill width of 0.02 rad deposited every 200 steps. A total of 16 walkers were used for the simulation.

10000 frames were randomly sampled from each simulation to generate training data. The simulation box was enlarged to 10 nm (without centering on the solute) to achieve an isolated water cluster in vacuum. Non-bonded interactions were treated with a cutoff of 4.0 nm and a MM charge scale factor of 1.0 for these structures during the single-point calculations.

Modified Force Field Parameter Simulation The solvated, but unoptimized, alanine dipeptide system described above was subjected to a classical Molecular Dynamics (MD) simulation using a modified Amber99SB-ILDN force field. The modification affects the force constant for bonds k_b (see Eq. (2.1)). Force constants involving hydrogen atoms were reduced by 50% to increase bond length fluctuations.

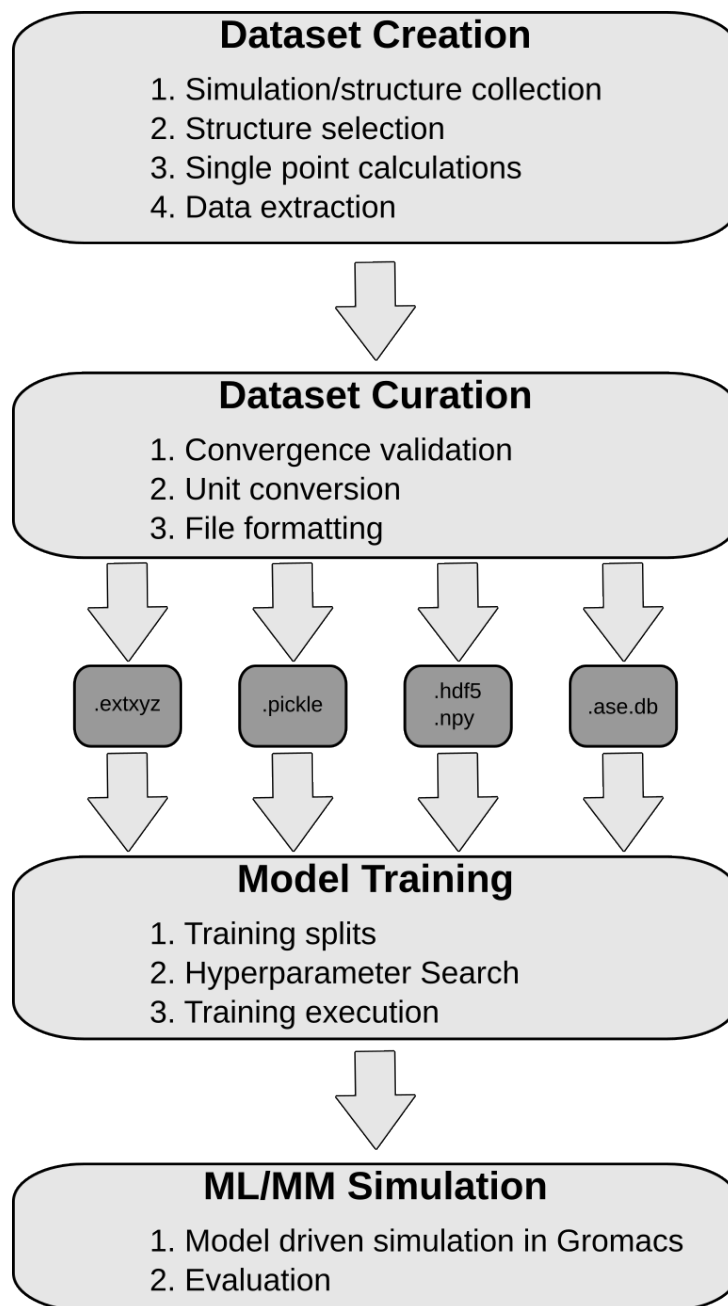


Figure 3.1.: Overview of the general Workflow employed in this work for the development of neural network potentials for QM/MM simulations.

The production metadynamics simulation was performed for 10 ns with a time step of 0.5 fs in the NVT ensemble at a temperature of 300 K using the Nose-Hoover thermostat. The non-bonded interactions were treated with a cutoff of 1.0 nm and PME for long-range electrostatics. The backbone dihedral angles ϕ and ψ were biased using metadynamics with a hill height of 0.5 kJ/mol and a hill width of 0.02 rad deposited every 200 steps. A total of 16 walkers were used for the simulation.

10000 frames were randomly sampled from the simulation to generate training data. Non-bonded interactions were treated with a cutoff of 1.0 nm and a MM charge scale factor of 1.0 for these structures during the single-point calculations.

Thiol Disulfide Exchange Reaction

Potential Energy Scan The gas-phase structures were adopted from Gómez Flores [99]. Here, an unrelaxed potential-energy scan of methylthiol attacking dimethyl disulfide was performed. The reaction coordinate was defined as the distance between the two sulfur atoms, ranging from 1.9 Å to 3.3 Å. The structures were solvated in a 4.0 nm water sphere in a 22.0 nm cubic box. A short 1000-step QM/MM-like optimization was performed where the reacting molecules were treated with DFTB3/3OB while the water molecules were treated with the Amber99SB-ILDN force field and TIP3P water model. Despite the QM/MM setup, only the solvent molecules were allowed to move during optimization; the QM region was kept frozen to ensure correctly polarized structures.

5112 solvated structures were obtained from the potential energy scan to generate training data. Non-bonded interactions were treated with a cutoff of 5.0 nm and a MM charge scale factor of 0.7 for these structures during the single-point calculations.

Water Simulation The gas-phase structures were adopted from Gómez Flores [99] and Maag [166]. Originally, the structures were sampled from a QM/MM metadynamics simulation at the DFTB/3OB level of the thiol disulfide exchange reaction in a mutated immunoglobulin I27 domain. Only gas-phase structure data were available from these works. These structures were treated in the same way as described above for the potential energy scan to generate solvated structures.

8436 solvated structures were obtained from the water simulation to generate training data. Non-bonded interactions were treated with a cutoff of 5.0 nm and a MM charge scale factor of 0.7 for these structures during the single-point calculations.

Protein Simulation Thiol disulfide exchange reaction motifs derived from the Human Glutaredoxin 1 (HsGrx1) and the N-terminal Heavy Metal transporting domain of the ATPase from *Arabidopsis thaliana* (HMA4n) were used for classical MD simulations. The dithiol and monothiol forms of HsGrx1 were

constructed from the PDB structure 1jhb. In both cases, Cys23 was modeled in its deprotonated thiolate state, and the monothiol variant additionally included a Cys26Ser mutation. The HMA4n-based model was generated from PDB structure 2kkh after removal of the crystallographic zinc ion. The resulting simulation systems were defined according to their redox state and protein composition. Four different systems were constructed from the two proteins and used for classical MD simulations: the monothiol glutaredoxin (OxMo1), the dithiol glutaredoxin in its intramolecular disulfide form (ReDi1), a mixed disulfide complex of monothiol glutaredoxin covalently linked to HMA4n via Cys27 (ReMo2X7), and a mixed disulfide complex of dithiol glutaredoxin covalently linked to HMA4n via Cys27 (ReDi2X7d).

The systems were simulated in cubic boxes with edge lengths between 7.2 nm up to 12.5 nm containing the proteins, water, and counter ions using the Amber99SB-ILDN force field¹⁶⁴ and TIP3P water model¹⁶⁵. For each protein, a 10 ns NVT equilibration at 500 K using the V-rescale thermostat was performed, followed by a 10 ns NPT equilibration at 1 bar using the Parrinello-Rahman barostat. Finally, a 125 ns with a time step of 0.5 fs production simulation was performed in the NVT ensemble at 500 K and 1 bar. The non-bonded interactions were treated with a cutoff of 1.0 nm and PME for long-range electrostatics. Respective thiolate and disulfide groups in each protein were constrained to stay within a distance of 0.5 nm.

About 10,000 frames, evenly distributed across the four protein simulations, were randomly extracted for training data generation. Non-bonded interactions were treated with a cutoff of 2.0 nm and a MM charge scale factor of 0.7 for these structures during the single-point calculations after performing the previously described solvation of the gas-phase structures.

Modified Force Field Parameter Simulations Analogous to the alanine dipeptide simulations, modified force field parameter simulations were performed with an even stronger reduction of the bond force constants k_b by 75% for all bonds involving hydrogen atoms to increase bond length fluctuations.

The production well-tempered metadynamics simulation was performed for 10 ns with a time step of 0.5 fs in the NVT ensemble at a temperature of 300 K using the Nose-Hoover thermostat. The non-bonded interactions were treated with a cutoff of 1.0 nm. The distances between the respective thiolate sulfur and disulfide sulfur atoms were biased using metadynamics with a hill height of 1.0 kJ/mol and a hill width of 0.02 nm deposited every 500 steps. The bias factor for the well-tempered metadynamics was set to 21. A total of 16 walkers were used for the simulation.

The 10000 structures exhibiting the largest hydrogen-involving bond lengths were extracted from the simulation to generate training data. Non-bonded interactions were treated with a cutoff of 1.0 nm and a MM charge scale factor of 0.7 for these structures during the single-point calculations.

Uncertainty Sampling Uncertainty sampling⁵¹ as described in Section 2.2.2 was performed to further enhance the training data set. 10 iterations of uncertainty sampling were performed starting from an initial data set of 100 structures from the previous data generation steps. In each iteration, the methods Gaussian Mixture Model (GMM) based on MACE¹³¹, extended Adaptive Biasing Force (eABF), and Gaussian accelerated Molecular Dynamics (GaMD) were combined to sample new structures. The ground truth for the MACE training was calculated using DFTB3/3OB. The uncertainty-driven simulations were performed using Langevin dynamics at 300 K with a time step of 0.5 fs for an initial 50000 steps, increasing the time step each iteration by 20%. The threshold for the cosine distance in the latent space was adjusted dynamically to yield 2000 new structures per iteration. The thermal coupling constant σ was set to 10 to balance exploration and stability. The hyperparameters for the MACE model used in uncertainty sampling are summarized in Table 3.1.

Table 3.1.: MACE hyperparameters for uncertainty-driven sampling

Hyperparameter	Value
Model	MACE
$r_{\max}$	5.0 Å
Radial type	Bessel
Number of radial basis functions	8
Number of cutoff basis functions	5
Radial MLP layers	[64, 64, 64]
Interaction blocks	2
Correlation order	3
Maximum ℓ	3
Hidden Irreps	128x0e + 128x1o
Gate activation	SiLU

Approximately 8300 structures were obtained via uncertainty sampling to generate training data. Non-bonded interactions were treated with a cutoff of 5.0 nm and a MM charge scale factor of 0.7 for these structures during the single-point calculations.

3.3. Reference Methods

The structures obtained from the sampling methods described above were subjected to single-point calculations using different reference methods to obtain ground-truth data for training neural network potentials. Each structure was treated once in vacuum and once within its environment (solvent or protein) using a QM/MM setup. Reference energies were subtracted from the total energies for the training. Reference energies were taken from optimized structures in the gas phase or in solution, or from atomic energies, depending on the data set. The following reference methods were applied:

DFT The hybrid functional B3LYP^{72,77} in combination with the augmented correlation-consistent triple-zeta basis set aug-cc-pVTZ⁸⁶ was used for all DFT calculations using the ORCA program¹⁶⁷. While B3LYP is not the most accurate functional for the systems studied here, it shows consistent average performance across a range of properties¹⁶⁸. Calculations were performed in the electronic ground state. No dispersion corrections were applied. Energies, forces, atomic partial charges, molecular dipole moments, and quadrupole moments were calculated and stored. The ElectroStatic Potential (ESP) and its gradient caused by the Molecular Mechanics (MM) point charges were calculated at the position of each Quantum Mechanical (QM) atom for the QM/MM structures based on ORCA's point charge files using a custom script.

Mulliken, Löwdin, and Hirshfeld charge partitioning methods were applied to obtain atomic partial charges for all the structures. For a subset of 100 structures per sampling method, additional charge partitioning methods (CHELPG, MBIS, RESP, EEM, MBIS, Merz-Kollman) were applied using the Multiwfn program¹¹⁵.

The reference energies used for training data generation are summarized in Table 3.2 and Table 3.3.

Table 3.2.: DFT reference energies for different systems and environments.

System	Environment	Reference Energy (eV)
Alanine dipeptide	Vacuum	-495.751901
Alanine dipeptide	Water	-495.811528
Thiol disulfide exchange reaction	Vacuum	-1314.213707
Thiol disulfide exchange reaction	Water	-1314.594625

Table 3.3.: DFT atomic energies used for atomization energy calculations.

Element	$E_{0,\text{DFT}}$ (eV)
H	-13.575
C	-1029.617
N	-1485.141
O	-2042.617
S	-10832.265

DFTB The semi-empirical method DFTB3⁹³ was used for all DFTB calculations. The 3OB parameter set¹⁰¹ was employed for the alanine dipeptide structures. In contrast, a modified 3OB parameter set with improved sulfur interactions (3OB_{Mod S-S})⁹⁹ was used for the thiol disulfide exchange reaction structures. The calculations were performed in the electronic ground state, using Fermi-Dirac smearing to aid convergence. The Fermi temperature was set to 300 K for the alanine dipeptide structures and 500 K for the thiol disulfide exchange reaction structures. No dispersion corrections were applied. Energies, forces, and Mulliken atomic partial charges were calculated and stored.

Δ energies ΔE_0 were calculated as the difference between the atomization energies E_0 obtained from DFT and DFTB calculations. Atomization energies were calculated as the difference between the total energy of the molecule and the sum of the atomic energies of its atoms $E_0^{(i)}$:

$$\Delta E_0 = E_{0,\text{DFT}} - E_{0,\text{DFTB}} \quad (3.1)$$

$$\Delta E_0 = (E_{\text{DFT}} - \sum_i^{N_{\text{elements}}} n_i E_{0,\text{DFT}}^{(i)}) - (E_{\text{DFTB}} - \sum_i^{N_{\text{elements}}} n_i E_{0,\text{DFTB}}^{(i)}) \quad (3.2)$$

$$\Delta \mathbf{F} = \mathbf{F}_{\text{DFT}} - \mathbf{F}_{\text{DFTB}} \quad (3.3)$$

The reference energies used for training data generation are summarized in Table 3.4 and Table 3.5.

Table 3.4.: DFTB reference energies for different systems and environments.

System	Environment	Reference Energy (eV)
Alanine dipeptide	Vacuum	-26.227679
Alanine dipeptide	Water	-26.524517
Thiol disulfide exchange reaction	Vacuum	-15.868835
Thiol disulfide exchange reaction	Water	-16.299172

Table 3.5.: DFTB atomic energies and Δ atomic energies ($\Delta E_0 = E_{0,\text{DFT}} - E_{0,\text{DFTB}}$) used for atomization energy calculations.

Element	$E_{0,\text{DFTB}}$ (eV)	ΔE_0 (eV)
H	-7.609986	-5.965049
C	-39.292500	-990.324862
N	-60.326270	-1424.814794
O	-85.497297	-1957.120012
S	-63.221147	-10769.044186

3.4. Machine Learning Potentials — Training

The final datasets for the two systems consist of two parts: one for vacuum calculations and one for QM/MM simulations in water. Each part contains corresponding structures sampled as described in Section 3.2, 30 000 structures for the alanine dipeptide and 41 836 structures for the thiol disulfide exchange reaction. Each structure was calculated using DFT at the B3LYP/aug-cc-pVTZ level and DFTB using the 3OB or 3OB_{Mod S-S} parameter set. Information on the elements, atomic positions, electrostatic potentials, and electric fields was used as input features. The output targets include energies, forces, atomic

partial charges, dipole moments, and quadrupole moments, although not every MLP architecture uses all of these features and targets.

The final datasets were split into two parts: 10% of the data was set aside as a test set to evaluate the final model performance. The remaining 90% was further divided into three separate subsets to train models on independent data. In total, 3 models were trained each time. One of each non-overlapping subset was used for training, and the remaining two for validation. This procedure was applied during all model training unless otherwise stated.

The crucial hyperparameters for each model depend on the architecture used. They were determined in different ways, either by hyperparameter optimization or by adapting literature values.

Hyperparameter Optimization

A grid search and a hyperband search were used to optimize hyperparameters across different model architectures. For the KGCNN models, KerasTuner¹⁶⁹ was used to perform an extensive hyperparameter search. For the Base MACE models, Optuna¹⁷⁰ was employed for hyperparameter optimization. The hyperband search was used for both systems, the alanine dipeptide and thiol disulfide exchange reaction. The grid search was only used for the alanine dipeptide system. The results of the hyperparameter optimization for both model architectures are presented in Section 6.1.1.

Grid Search The grid search was performed on a subset of 10000 data points for the alanine dipeptide system, using 50 epochs for the KGCNN models. Base MACE models used a subset of 1000 data points for 67 epochs. Details about the search space are provided in Appendix D.3

Hyperband Search The hyperband search was performed on a subset of 10000 data points for the alanine dipeptide system and on the full thiol disulfide exchange reaction data set, using 200 epochs and an early stopping patience of 25 epochs for the KGCNN models. The hyperband factor was set to 2. Base MACE models used a subset of 2500 data points for 67 epochs without early stopping and a hyperband factor of 3.

Vacuum Machine Learning Potentials

KGCNN Models Models with the 2G-HDNNP, 4G-HDNNP, SchNet, and PaiNN architectures were trained using KGCNN¹⁴⁴. An extensive hyperparameter optimization using KerasTuner¹⁶⁹ was performed for these architectures. The results are presented in Section 6.1.1. The hardness and width

parameters for 4G-HDNNP models are fixed to literature hardness values¹⁷¹ and covalent radii¹⁷², respectively, without further optimization.

Base MACE MACE was trained using a customized MACE package¹³¹. Base MACE hyperparameters were investigated using Optuna¹⁷⁰ for hyperparameter optimization. The results are presented in Section 6.1.1. Alanine dipeptide models were trained using typical hyperparameters without further optimization. Thiol disulfide exchange reaction models were trained with hyperparameters optimized using Optuna.

Table 3.6.: Base MACE hyperparameters used for training of alanine dipeptide.

Hyperparameter	Value
Model	ScaleShiftMACE
$r_{\max}$	5.0 Å
Radial type	Bessel
Number of radial basis functions	8
Number of cutoff basis functions	5
Number of interaction blocks	2
Correlation order	3
Maximum ℓ	3
Hidden irreps	64x0e + 64x1o
MLP irreps	16x0e
Radial MLP layers	[64, 64, 64]
Gate activation	SiLU
Energy weight	1
Forces weight	100
Charges weight	50
SWA energy weight	1000
SWA forces weight	100
Optimizer	Adam
Learning rate	0.01
SWA learning rate	0.001
Weight decay	5×10^{-7}
Scheduler	ReduceLROnPlateau
LR factor	0.8
Scheduler patience	3
Gradient clipping	10.0
Batch size	10
Maximum epochs	100
Patience	15

The training loss was calculated as a weighted sum of energy, forces, and atomic partial charges:

$$\mathcal{L} = 1 \cdot \mathcal{L}_E + 100 \cdot \mathcal{L}_F + 50 \cdot \mathcal{L}_q \quad (3.4)$$

Environment-Dependent Machine Learning Potentials

Charge-Equilibration MACE Charge-Equilibration MACE (QEq-MACE) models were trained using a customized MACE package¹³¹. The hyperparameters were adapted identically to the Base MACE models, except for the charge weight in the loss function. Parameters for the hardness J_{ii} and the Gaussian width σ_i were freely optimized during training.

$$\mathcal{L} = 1 \cdot \mathcal{L}_E + 100 \cdot \mathcal{L}_F + 50 \cdot \mathcal{L}_q \quad (3.5)$$

FieldSchNet FieldSchNet models were trained using the SchNetPack library¹⁶³. Typical hyperparameters from the literature were used for training. In contrast to other model training, only one 80:10:10 data split was used, rather than three independent splits. The hyperparameters used are summarized in Table 3.7.

Table 3.7.: FieldSchNet hyperparameters used for training.

Hyperparameter	Value
$r_{\max}$	10.0 Å
Radial basis functions	Gaussian
Number of radial basis functions	60
Cutoff function	Cosine
Atom basis size	64
Number of filters	256
Interaction blocks	6
Shared interactions	False
External fields	Electric field
Output hidden layers	256
Output activation	Tanh
Output layers	2
Energy weight	0.1
Forces weight	0.85
Charges weight	0.05
Optimizer	Adam
Learning rate	0.0001
Weight decay	0.0
Scheduler	ReduceLROnPlateau
LR factor	0.8
Scheduler patience	5
Gradient clipping	10.0
Batch size	16
Maximum epochs	150
Early stopping patience	50
EMA decay	0.995

The training loss was calculated as a weighted sum of energy, forces, and atomic partial charges:

$$\mathcal{L} = 0.1 \cdot \mathcal{L}_E + 0.85 \cdot \mathcal{L}_F + 0.05 \cdot \mathcal{L}_q \quad (3.6)$$

FieldMACE FieldMACE models were trained using the modified MACE package published by Barrett, Dietschreit, and Westermayr¹⁵⁹. Typical literature hyperparameters were adapted for training, following those used for Base MACE.

Table 3.8.: FieldMACE hyperparameters used for training.

Hyperparameter	Value
Model	AttentionFieldEMACE
$r_{\max}$	8.0 Å
Radial type	Bessel
Number of radial basis functions	8
Number of cutoff basis functions	5
Hidden irreps	64x0e + 64x1o
Field irreps	16x0e + 16x1o + 16x2e
Maximum ℓ	3
Correlation order	3
Interaction blocks	2
Gate activation	SiLU
Energy weight	1
Forces weight	100
Dipoles weight	10
NACs weight	0
SWA energy weight	1000
SWA forces weight	100
SWA dipoles weight	10
SWA NACs weight	0
Optimizer	Adam
Learning rate	0.01
SWA learning rate	0.001
Scheduler	ReduceLROnPlateau
LR factor	0.8
Scheduler patience	3
Gradient clipping	10.0
Batch size	8
Maximum epochs	100
EMA decay	0.99

The training loss was calculated as a weighted sum of energy, forces, and dipole moments:

$$\mathcal{L} = 1 \cdot \mathcal{L}_E + 100 \cdot \mathcal{L}_F + 10 \cdot \mathcal{L}_\mu + 0 \cdot \mathcal{L}_{\text{NAC}} \quad (3.7)$$

AMP AMP models were trained using a customized AMP package¹⁵⁶. The hyperparameters for the model were adapted exactly as described for the smaller 600.000 parameter model in Pultar et al. [156]. The training hyperparameters are summarized in Table 3.9.

Table 3.9.: AMP hyperparameters used for training.

Hyperparameter	Value
Model architecture	minimal
Node size	128
Number of channels	32
Number of kernels	8
Number of QM/MM kernels	8
Order	2
Cutoff	10.0 Å
Cutoff ESP	500.0 Å
Cutoff long-range	1000.0 Å
Learning rate	0.0005
Batch size	16
Maximum gradient norm	10.0
Number of epochs	100
Optimizer	Adam
Scheduler	ExponentialLR
Decay factor	0.05

The training loss was calculated as a weighted sum of energy, forces (both QM and MM), dipole moments, and quadrupole moments:

$$\mathcal{L} = (1 - 0.99) \cdot \mathcal{L}_E + \frac{0.99}{3} \cdot \mathcal{L}_{F_{QM}} + \frac{0.99 \cdot 1}{3} \cdot \mathcal{L}_{F_{MM}} + \frac{1}{3} \cdot \mathcal{L}_{\text{dipole}} + \frac{1}{6} \cdot \mathcal{L}_{\text{quadrupole}} \quad (3.8)$$

3.5. Machine Learning Potentials — Simulation

In preparation for the production runs using the trained neural network potentials, the alanine dipeptide and thiol disulfide exchange reaction systems were equilibrated in a 2500 fs NVT simulation with a time step of 0.5 fs at 300 K. Vacuum simulations used Langevin dynamics, while QM/MM simulations employed the Nose-Hoover thermostat. The box sizes for the vacuum simulations were set to 3.0 nm for the alanine dipeptide and 20.0 nm for the thiol disulfide exchange reaction. The increased box size for the thiol disulfide exchange reaction was necessary to avoid periodic image selection errors with large thiol disulfide distances. The solvated alanine dipeptide was placed in a cubic box with an edge length of 3.0 nm, containing ~ 900 water molecules. The solvated thiol disulfide exchange reaction used a cubic box with an edge length of 20.0 nm with ~ 11000 water molecules in a 4.0 nm sphere around the

reactants to limit the computational cost while still providing a sufficient solvation environment for the reaction.

A DFTB QM/MM simulation was performed as a reference for the ML/MM simulations. 2G-HDNNP, 4G-HDNNP, SchNet, PaiNN, Base MACE, QEq-MACE, and AMP MLPs were used for the production vacuum simulations. 4G-HDNNP, Base MACE (as a baseline), QEq-MACE, and AMP MLPs were used for the production ML/MM simulations. The simulations were performed for 0.75 ns with a time step of 0.5 fs in the NVT ensemble at a temperature of 300 K. Vacuum simulations used Langevin dynamics, while ML/MM simulations employed the Nose-Hoover thermostat. In case of alanine dipeptide, the electrostatics were treated with a cutoff of 1.45 nm with a MM charge scale factor of 1.0. In case of the thiol disulfide exchange reaction, the electrostatics were treated with a cutoff of 1.00 nm with a MM charge scale factor of 0.7. Constraints via the LINCS algorithm were not applied; instead, the trained neural network potentials were relied upon to maintain bond lengths.

Simulations were enhanced using metadynamics. The backbone dihedral angles ϕ and ψ of alanine dipeptide were biased using metadynamics with a hill height of 0.5 kJ/mol and a hill width of 0.05 rad deposited every 100 steps. Two of the sulfur distances in the thiol disulfide exchange reaction were biased using metadynamics with a hill height of 1.25 kJ/mol and a hill width of 0.008 nm deposited every 500 steps. While the thiol disulfide exchange reaction used well-tempered metadynamics with a bias factor of 20 for the entire duration of the simulation, the alanine dipeptide simulation employed standard metadynamics for the first 0.75 ns followed by well-tempered metadynamics with a bias factor of 21 in a second 0.25 ns simulation to reach a total of 1.0 ns simulation time.

The two biased distances in the thiol disulfide exchange reaction were restrained using harmonic restraints with a force constant of $100\,000\text{ kJ}/(\text{mol nm}^2)$ to stay below a distance of 1.0 nm. Additionally, the attack angle formed by the three sulfur atoms was restrained to stay above 130° using a harmonic restraint with a force constant of $100\,000\text{ kJ}/(\text{mol rad}^2)$.

The ML/MM simulations were monitored for stability by tracking the bond distances and the standard deviations of energy predictions across an ensemble of models. The simulation was terminated once any bond distance exceeded $2.5\text{ \AA}$ or the standard deviation of the energy predictions exceeded 0.015 eV. The standard deviation threshold was chosen based on the observed fluctuations in the test set predictions: $\sigma_{\text{threshold}} = \sigma_{\text{mean}} + 10 \times \sigma_{\text{std}}$. An ensemble of three models, each trained on an independent data split, was used to compute the standard deviation of the energy predictions. To limit the additional computational cost of evaluating multiple models during the live simulation, the standard deviation was only calculated every 100 steps.

3.5.1. Von Willebrand Factor Simulation

For the von Willebrand factor case study, ML/MM metadynamics simulations were performed for a QM/MM setup containing a single QM region (15 atoms) embedded in a fully solvated protein system (71662 atoms in total, neutralized with NaCl). The production setup used a time step of 1.0 fs for 750 000 steps, PME electrostatics, and real-space cutoffs of 1.0 nm for both Coulomb and van der Waals interactions. Temperature coupling was applied with the V-rescale thermostat at 300 K, and pressure coupling used the isotropic Parrinello-Rahman barostat at 1 bar. The MM charge scaling factor in the QM/MM coupling was set to 0.75.

The biased coordinates were sulfur-sulfur distances between Cys79–Cys36 (d_1) and Cys36–Cys78 (d_3), making Cys36 the central group, while Cys79–Cys78 (d_2) was monitored. Standard multi-walker metadynamics was employed with 24 walkers, hill height 1.0 kJ/mol, Gaussian widths $\sigma_{d_1} = \sigma_{d_3} = 0.008$ nm, and a deposition stride of 500 steps. The bias grid was defined in the range 0.15 nm to 0.70 nm for both d_1 and d_3 with a grid spacing of 0.005 nm. After 500 ps, well-tempered settings were enabled in this setup with a bias factor of 20. The metadynamics setting differ from the source settings¹⁶⁶, which used Gaussian widths $\sigma_{d_1} = \sigma_{d_3} = 0.02$ nm, which lead to extreme biases breaking both S—S bonds.

To keep sampling within chemically meaningful reactive geometries, additional restraints were applied. Upper walls were imposed on d_1 and d_3 at 0.60 nm with $\kappa=50\,000$ kJ/mol/nm². The S—S attack angle (Cys79–Cys36–Cys78) was restrained by a lower wall at 2.80 rad ($\approx 160^\circ$) with $\kappa=100\,000$ kJ/mol/rad². In addition, coordination-number-based lower-wall restraints were applied to the three cysteine C_β atoms to preserve local bonding patterns during enhanced sampling. The coordination should stay above 1.5 with $\kappa=50\,000$ kJ/mol/nm².

The termination criteria were kept up, but the standard deviation threshold was increased to 0.070 eV for the AMP model, as runs with the default threshold of 0.015 eV were terminated prematurely due to the increased fluctuations in the energy predictions for this system.

3.6. Software

Molecular Dynamics and Quantum Mechanics MD simulations with QM/MM capabilities were performed using Gromacs¹⁷³ (Development-Version 2022) with QM/MM interfaces to DFTB+^{174,175}, ORCA, and own developments to interface with TensorFlow and PyTorch.¹⁷⁵ Metadynamics and biasing of MD simulations were implemented using PLUMED (Version 2.9).^{176,177} DFTB calculations were carried out with DFTB+ (Version 24.1).^{178,179} DFT calculations were performed using ORCA (Version 5.0.3).¹⁶⁷ Wavefunction analysis of ORCA QM calculations, calculation of electrostatic potentials, and charge partitioning methods (CHarges from ELectrostatic Potentials using a Grid-based

method (CHELPG), Minimal-Basis Iterative Stockholder (MBIS), Restrained Electrostatic Potential (RESP), Electronegativity-Equalization Method (EEM), MBIS, Merz-Kollman) were performed with Multiwfn^{115,180} (Version 3.8).

Neural Network Potentials and Machine Learning Implementation and training of neural network potentials were conducted using TensorFlow¹⁸¹ (Version 2.15) with Keras (Version 2.15)¹⁸² as well as PyTorch¹⁸³ (Version 2.3) with PyTorch Lightning.¹⁸⁴ The 2G-HDNNP, 4G-HDNNP, SchNet, and PaiNN neural network potentials were implemented using KGCNN¹⁴⁴ (Version 2.2.3). The MACE neural network potential was implemented using the MACE package¹³¹ (Version 0.3.5). The AMP neural network potential was implemented using the AMP package¹⁵⁶ (Version 0.1). The FieldSchNet neural network potential was implemented using the SchNetPack library¹⁶³ (Version 2.1.1).

KerasTuner¹⁶⁹ (Version 1.4.7) and Optuna¹⁷⁰ (Version 4.5.0) were used for hyperparameter optimization of neural network potentials.

Data Management and Analysis ASE¹⁸⁵ (Version 3.23.0) was used for dataset management and interfacing with different software packages. MDAnalysis^{186,187} (Version 2.7.0) was used for trajectory analysis and data processing.

Weights & Biases¹⁸⁸ was used for experiment tracking, visualization of training processes, and overview of results.

GNU Parallel¹⁸⁹ was used to parallelize calculations on high-performance computing clusters.

3.7. Implementation

To utilize MLPs within MD simulations using GROMACS, interfaces for TensorFlow and PyTorch were developed to enable seamless integration of the models into the simulation workflow. These interfaces can load multiple models for uncertainty estimation, enabling active learning and allowing termination criteria based on ensemble uncertainty during simulations. To reduce the computational cost associated with evaluating multiple models, the frequency of uncertainty evaluation can be adjusted. A different use case for the ensemble of models is to improve prediction accuracy by averaging the outputs of multiple models.

The interfaces are designed to facilitate modular preparation of inputs from GROMACS for the MLP and the return of outputs from the MLP to GROMACS. Currently, implementations of the 2G-HDNNP, 4G-HDNNP, SchNet, and PaiNN architectures are available in TensorFlow, while Base MACE, QEq-MACE, and AMP are implemented in PyTorch. Using model wrappers that translate inputs and outputs to the

expected format, other architectures can be implemented easily. The native model structure, as saved in TensorFlow and PyTorch, remains intact, ensuring no loss of performance in the model call from the GROMACS side. The interfaces also support delta learning models. For this, both the base method (e.g., DFTB) and the MLP are called at each time step, and the final energy and forces are obtained by adding the predictions of both methods.

User input for the interfaces is handled via environment variables, enabling easy configuration of the MLP models. An overview of the available environment variables and their descriptions is provided in Table 3.10.

Table 3.10.: Environment variables for configuring the MLP interfaces in Gromacs.

Variable	Description
GMX_N_MODELS	Number of neural network models to load. Default: 1
GMX_MODEL_PATH_PREFIX	Prefix of the path to the neural network models. The models are expected to be named as <prefix>0.pt, <prefix>1.pt, ..., for PyTorch models and <prefix>0, <prefix>1, ... for TensorFlow models. Default: "
GMX_ENERGY_PREDICTION_STD_THRESHOLD	Threshold for the energy standard deviation between models for a structure to be considered relevant, in units of output of model. A negative value disables this criterion. Default: -1
GMX_FORCE_PREDICTION_STD_THRESHOLD	Threshold for the force standard deviation between models for a structure to be considered relevant, in units of output of model. A negative value disables this criterion. If both energy and force thresholds are set, the energy threshold has priority. Default: -1
GMX_NN_EVAL_FREQ	Frequency of neural network evaluation, 1 means every step. Default: 1
GMX_NN_ARCHITECTURE	Neural network architecture. Implemented choices are hdnnp2nd, hdnnp4th, schnet, and painn for TensorFlow models, and mace, maceqeq, and amp for PyTorch models. Default: 'maceqeq'
GMX_NN_SCALER	Scaler file for the neural network, optional, TensorFlow specific. Default: "
GMX_NN_R_MAX	Cutoff radius for the neural network, in Bohr. TensorFlow specific. Default: 20.0
GMX_NN_EXTXYZ	Frequency of writing the extended xyz file, 1 means every step, 0 means never. Default: 0

Proof of Implementation

As an assurance of the correct implementation of the MLP architectures within the TensorFlow and PyTorch interfaces, the agreement of energy, forces, and charges (where applicable) predictions between Gromacs and standalone Python scripts was verified. For this purpose, a short 100-step MD simulation of alanine dipeptide in water was performed using each architecture. The extxyz-format output, as a feature of the interfaces, was used to store the inputs and outputs of each step. The extxyz files were then converted to the input format for each architecture, and predictions were obtained again for the same structures, this time using standalone Python scripts. The predictions from Gromacs and the Python scripts were compared, and perfect agreement was found in all cases, confirming the correctness of the implementations. The results are shown in Figure 3.2 for the TensorFlow implementations and in Figure 3.3 for the PyTorch implementations.

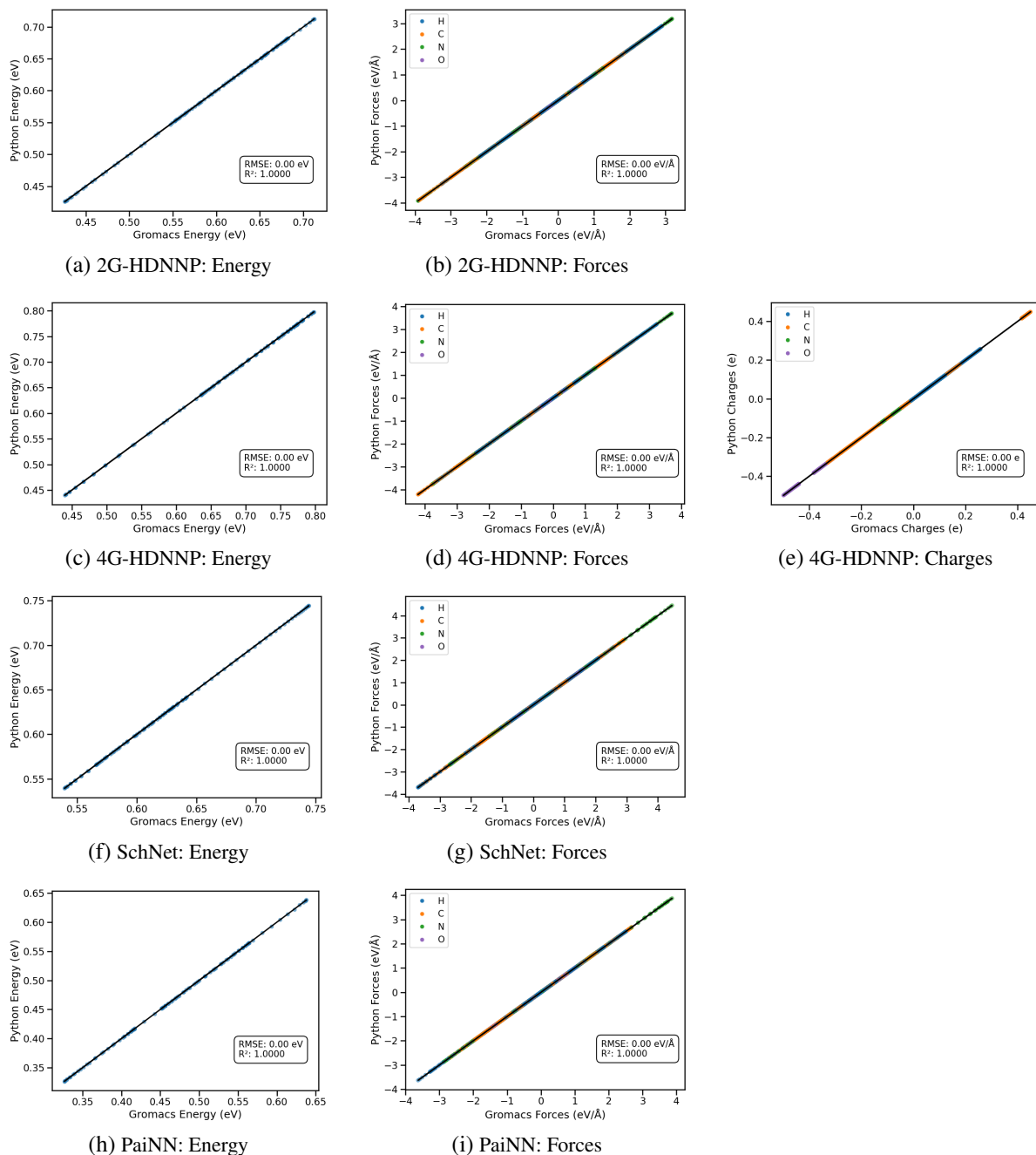


Figure 3.2.: Proof of implementation for all MLP architectures implemented using TensorFlow. Each row shows one architecture, with available predictions in GROMACS and Python (energy, forces, and charges, where applicable). The correlations are perfect in each case, indicating that the implementations are correct.

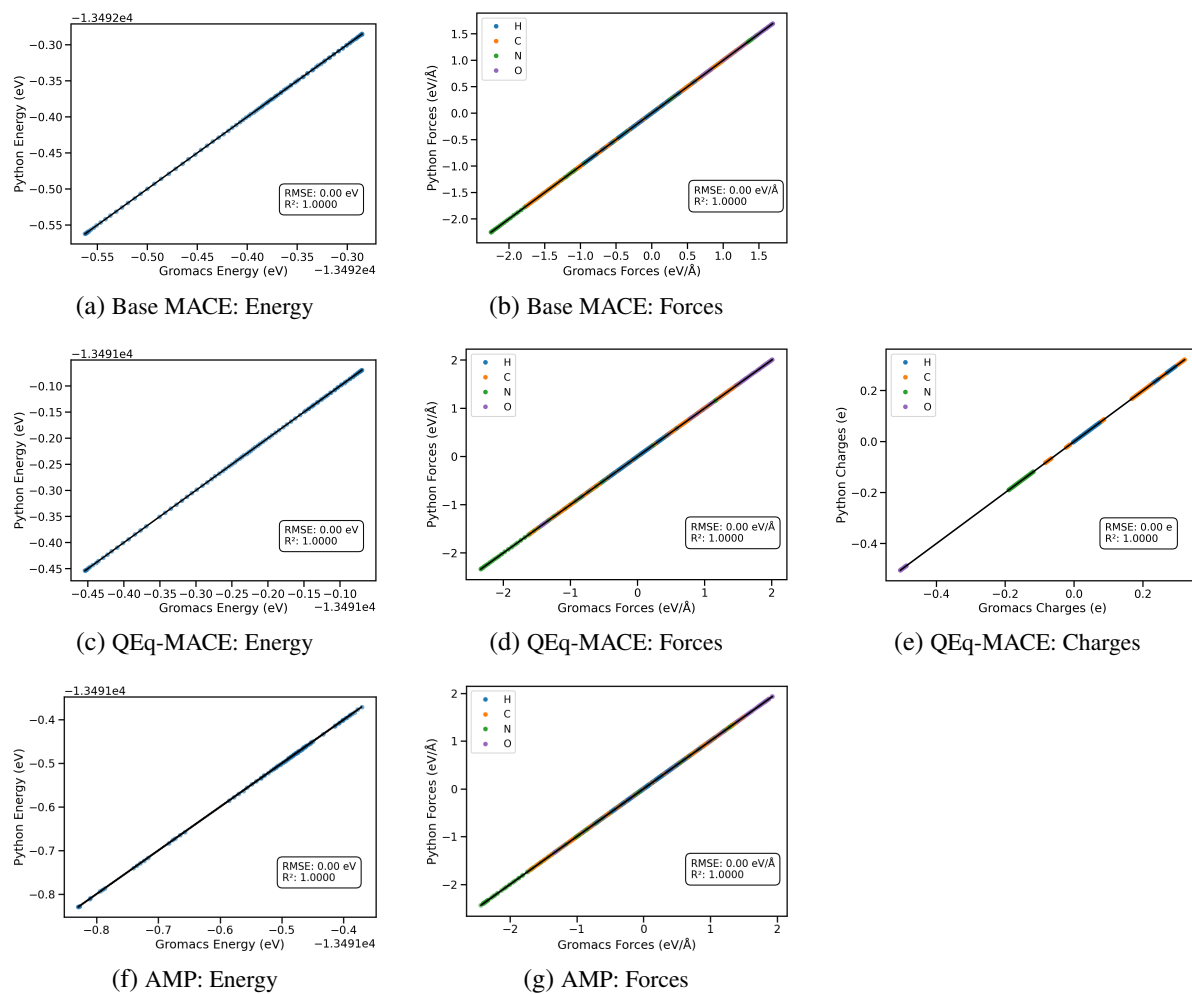


Figure 3.3.: Proof of implementation for all MLP architectures implemented using PyTorch. Each row shows one architecture, with available predictions in GROMACS and Python (energy, forces, and charges, where applicable). The correlations are perfect in each case, indicating that the implementations are correct.

Part II.

Results and Discussion

4. Sampling Methods

When choosing a sampling method to generate training data for Machine Learning Potentials (MLPs), one must balance several factors. The computational cost must be weighed against the desired accuracy and the size of the training dataset. The sampling method must provide sufficient coverage of the configurational space while remaining relevant to the target application. Critical regions, such as transition states, must be adequately represented in the training data to ensure reliable predictions. During the Machine Learning (ML)-driven simulation, the model must make accurate predictions even in the fringes of the configurational space. Errors in force prediction can induce instabilities during Molecular Dynamics (MD) simulations, ultimately leading to simulation failure.

Examples of sampling methods include MD simulations at different temperatures, enhanced sampling techniques (umbrella sampling, metadynamics, replica exchange MD), normal mode sampling, potential energy scans, and uncertainty-based sampling. An introduction to these sampling methods is provided in Section 2.2.1.

The purpose of this chapter is to compare different sampling methods for generating training data for MLPs for two example systems: alanine dipeptide and a thiol disulfide exchange reaction. MD simulations at different temperatures, metadynamics, potential energy scans, and uncertainty-based sampling are applied to generate diverse datasets for both systems. The generated datasets are compared with respect to structural diversity and property distributions. Furthermore, MLPs are trained on different datasets, and their performance is evaluated using cross-validation.

Author Contributions The work presented in this chapter was in parts adapted from the earlier works of Gómez Flores⁹⁹ and Maag¹⁶⁶ and supported by Dr. Christian C. Schmidt, Dr. Julian Böser, M.Sc. Sebastian Will, and M.Sc. Pascal Schmidt. Potential energy scan and water metadynamics simulations for vacuum thiol disulfide exchange reaction structure generation were performed by Gómez Flores and Maag and reused in this thesis. Dr. Julian Böser performed classical protein metadynamics simulations of the thiol disulfide exchange reaction. Alanine dipeptide structures in vacuum and solution from 300 K and 500 K simulations were generated by Dr. Christian C. Schmidt. M.Sc. Sebastian Will performed initial simulations of the thiol disulfide exchange reaction with different scaling factors for the electrostatic interactions. M.Sc. Pascal Schmidt generated vacuum structures from uncertainty sampling. Solvation,

whenever required, reference method calculations, model training, and the analysis of the results were performed by the author of this thesis.

4.1. Alanine Dipeptide

Alanine dipeptide (N-acetyl-alanine-N'-methylamide, Ace-Ala-Nme) is a small biomolecule that is frequently used as a model system in computational chemistry and molecular simulations.^{51,156,190–196} It consists of one alanine amino acid connected by peptide bonds to acetyl and methylamide groups capping the N- and C-termini. The peptide bond allows for rotation around the ϕ and ψ dihedral angles (see Figure 4.1), which are related to the secondary structure of proteins, like α -helices and β -sheets. Several local energy minima can be observed

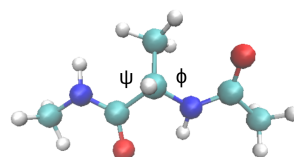


Figure 4.1.: Structure of alanine dipeptide.

in the CV space of the ϕ and ψ dihedral angles, known as the Ramachandran plot¹⁹⁷. The three most stable ones are referred to as $C7_{ax}$, $C7_{eq}$, and C_5 in literature.^{194,198} They possess intramolecular hydrogen bonds stabilizing the structures in vacuum. In solution, intermolecular hydrogen bonds with the solvent are formed instead, which leads to differences in the Potential Energy Surface (PES) compared to the gas-phase.^{191,194,198} In aqueous solution, the stable conformations correspond to the well-known α , β , and α_L conformations in the Ramachandran plot. Figure 4.2 shows the Potential of Mean Force (PMF) of alanine dipeptide in vacuum and water at the DFTB3/3OB level of theory with the most stable conformations annotated.

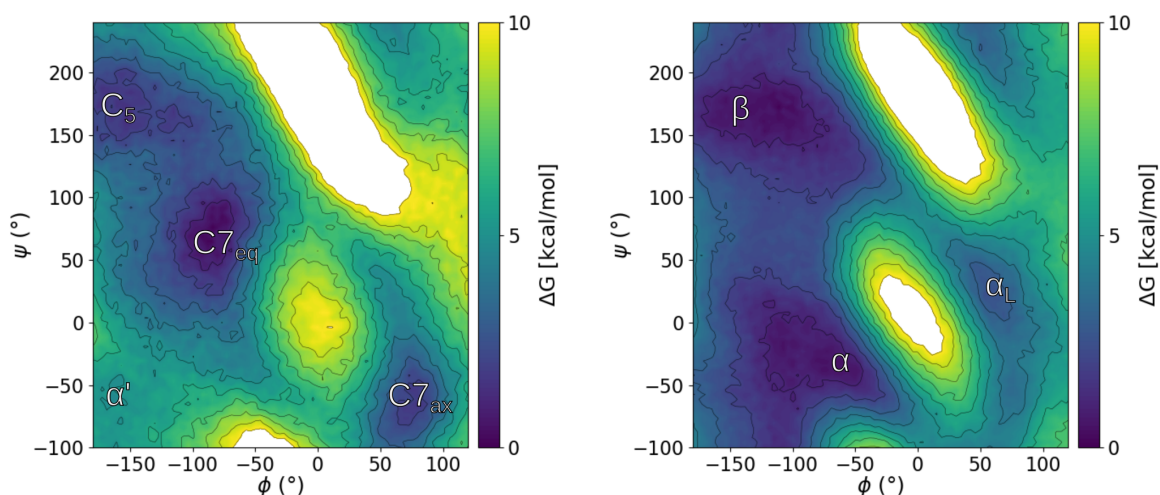


Figure 4.2.: PMF of alanine dipeptide in vacuum (left) and water (right) at the DFTB3/3OB level of theory. The most stable conformations in vacuum and water are annotated.

4.1.1. Geometry Comparison

Three sampling methods are compared for generating training data for alanine dipeptide: metadynamics simulations at 300 K and 500 K in water using Density Functional Tight Binding (DFTB), and at 300 K in water using modified force field parameters with weakened hydrogen bonds. The modified force field parameters are designed to include extreme hydrogen-bond lengths in the dataset to improve the robustness of the trained MLPs. Bonds involving hydrogen are the fastest oscillating degrees of freedom in molecular systems, so it is crucial to ensure that the MLP can accurately predict forces in the extrema of these bonds across all configurations.

Dimensionality reduction methods such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (tSNE), and Uniform Manifold Approximation and Projection (UMAP) can visualize the 3N-dimensional configurational space of a molecular system in lower dimensions. This is used as a preliminary qualitative comparison of the structural diversity of datasets generated by different sampling methods, without requiring knowledge of a suitable set of Collective Variables (CVs). Neighboring points in tSNE and UMAP plots are similar in the original high-dimensional space. An isolated cluster in the tSNE or UMAP plot indicates a distinct region of the configurational space, while an overlap of clusters indicates similar structures in the different datasets.

Here, tSNE is applied to the bond distances of the different datasets to compare the cluster formations and overlaps of the different sampling methods (Figure 4.3). Three distinct clusters can be observed, which can be attributed to $-\text{CH}_3$ rotations as only one cluster is present as soon as bonds involving hydrogen are excluded from the analysis. No distinction between the 300 K (blue) and 500 K (orange) simulations is visible, indicating that simulations at both temperatures cover similar regions of the configurational space. A higher temperature results in higher kinetic energy in the system, so encountering higher-energy structures is more likely. However, the dataset's structural diversity is not necessarily increased; the effect of temperature at this scale is not significant enough to yield a clear separation in the tSNE plot. On the other hand, the modified force-field parameter simulation (green) stands out with a significant fraction of structures located at the fringes of the clusters, indicating a greater presence in regions where the other sampling methods have few structures.

A more detailed comparison of the sampling methods can be achieved by comparing the location of the structures in the CV space of the ϕ and ψ dihedral angles (see Figure 4.4). The 300 K and 500 K simulations show a similar sampling of the CV space. The distribution of the sampled structures meets the expectation from the PMF in water in Figure 4.2, with the most stable conformations being the most populated ones. The modified force field parameters lead to increased sampling of otherwise unvisited regions of the CV space, consistent with the observation from the tSNE plot.

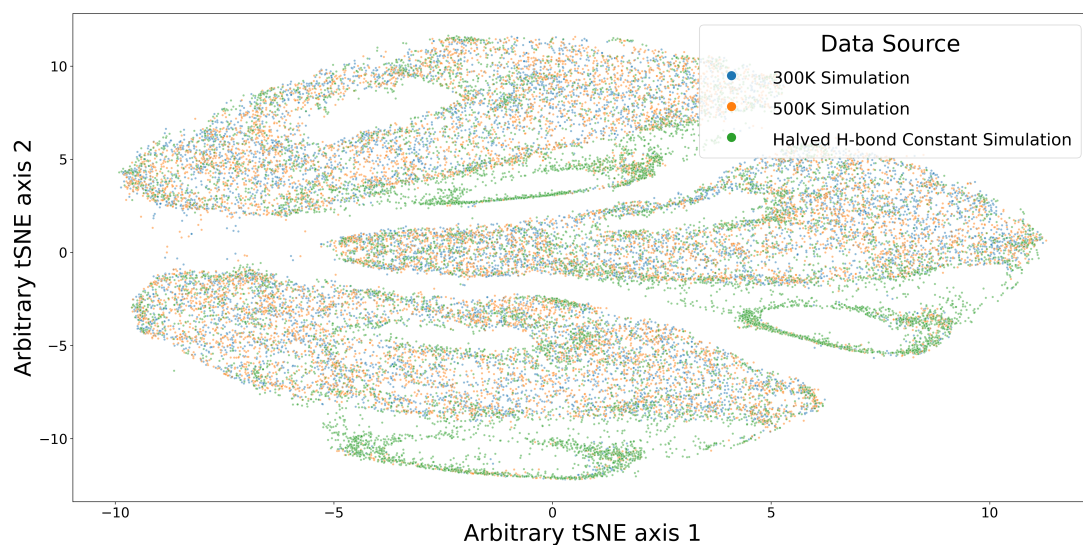


Figure 4.3.: Comparison of sampling methods for alanine dipeptide system using tSNE dimensionality reduction. No distinction between the 300 K and 500 K simulations is visible. Modified force-field parameters (green) lead to an increased location at the cluster fringes.

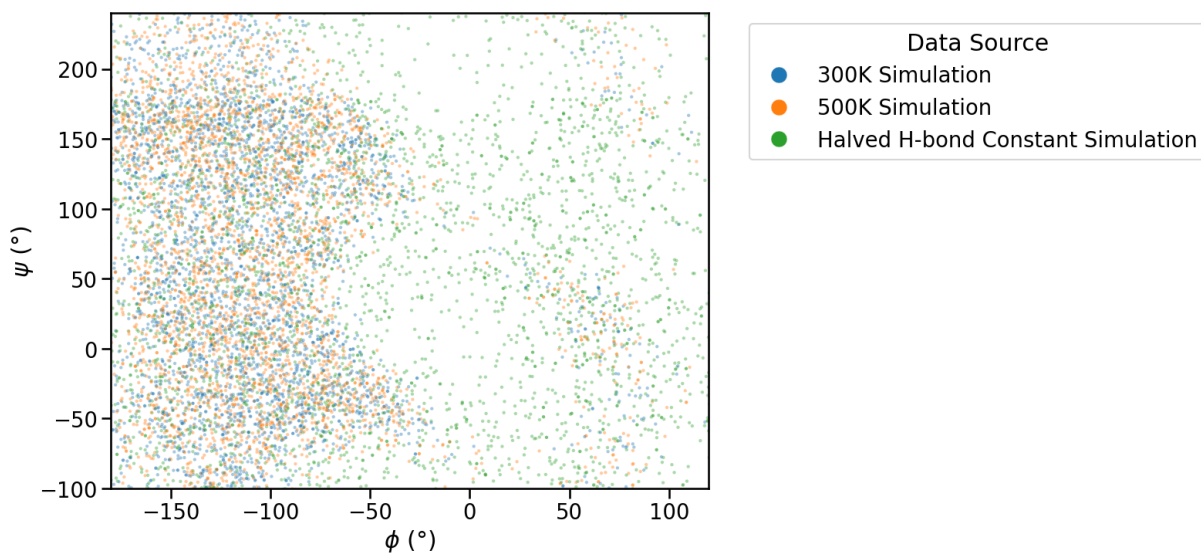


Figure 4.4.: Comparison of sampling methods for alanine dipeptide system in CV space of the two dihedral angles ϕ and ψ . No distinction between the 300 K and 500 K simulations is visible. Modified force-field parameters (green) enable sampling of otherwise unvisited regions.

Having observed the significant effect of the modified force-field parameters on the dataset’s structural diversity, the physical plausibility of the generated structures is now under scrutiny. The distribution of bond lengths of bonds involving hydrogen is compared across the different sampling methods in Figure 4.5. Both the 300 K and 500 K simulations show a range of C-H bond lengths between 1.0 Å and 1.2 Å. The modified force-field parameters lead to a slightly broader distribution of bond lengths; only a small fraction of structures have bond lengths exceeding 1.2 Å or below 1.0 Å. A similar observation can be made for the N-H bonds, with bonds ranging between 0.95 Å and 1.1 Å for the 300 K and 500 K simulations, whereas a noticeable fraction of the modified force field parameters N-H bonds fall below 0.95 Å.

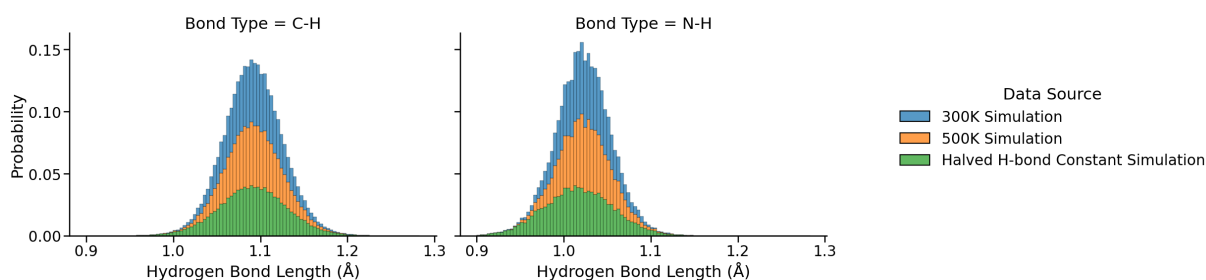


Figure 4.5.: Distribution of bond lengths of bonds involving hydrogen for the alanine dipeptide system. Modified force field parameters (green) lead to a broader distribution of bond lengths

This concludes that the structures introduced by the modified force field parameters are plausible. The force field itself allowed a wider range of dihedral angle combinations, whereas the modifications to the hydrogen-bond parameters only slightly broadened the distribution of bond lengths.

4.1.2. Property Comparison

Reference method calculations at the B3LYP/aug-cc-pVTZ level of theory in vacuum and in water are performed for all structures generated using the different sampling methods to compare property distributions across the datasets. The energy and force magnitude distributions for the vacuum case are shown in Figure 4.6. The 300 K and 500 K simulations exhibit similar energy and force-magnitude distributions, consistent with their structural similarity. The modified force-field parameters lead to structures with moderately increased energies and insignificantly increased forces in vacuum. A comparable statement holds for the simulations in water. This further supports the physical plausibility of the structures generated with the modified force field parameters.

4.1.3. Cross-validation

To evaluate the performance of MLPs trained on the different datasets, cross-validation is performed for 4G-HDNNP, SchNet, PaiNN, Base MACE, and Charge-Equilibration MACE (QEq-MACE) models.

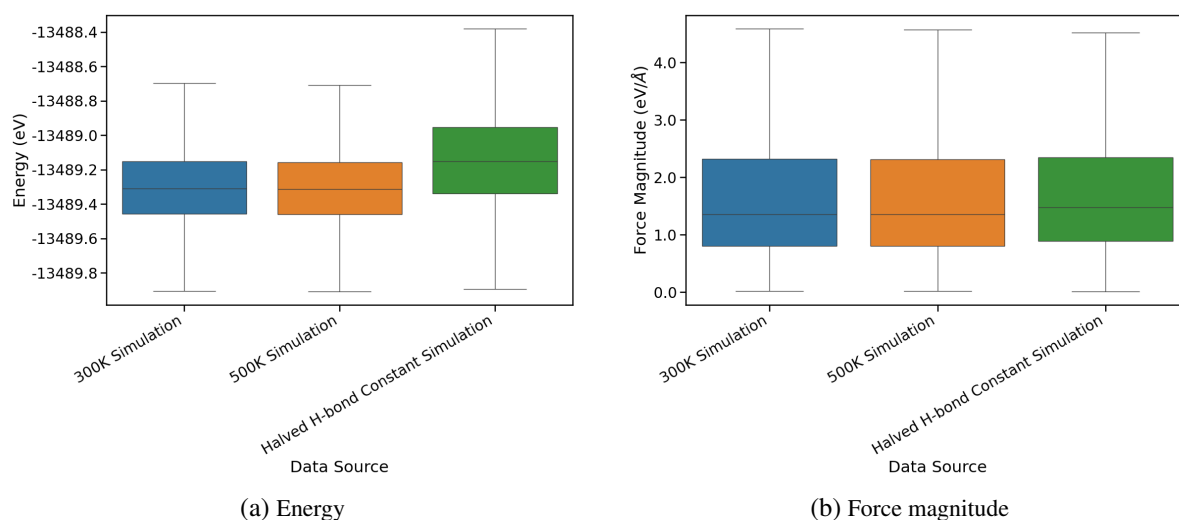


Figure 4.6.: Comparison of sampling methods for alanine dipeptide system regarding (a) energy and (b) forces across different sampling methods. Modified force field parameters (green) lead to structures with moderately increased energies and insignificantly increased forces at the B3LYP/aug-cc-pVTZ level of theory in vacuum. The high-temperature simulation (orange) had no effect.

The models are trained and evaluated on different datasets. For a fair comparison, the training sets were subsampled to 8000 data points across the different sampling methods, and all models reused the test set during the cross-validation. Three models, each trained on non-overlapping splits of the training set, are trained for each dataset source to obtain a statistical estimate of performance. The result of the cross-validation for the 2G-HDNNP models is shown in Figure 4.7 in a confusion matrix. Each row corresponds to a dataset source used for training, and each column to the test set data source used for the evaluation of each model. The diagonal shows the models' performance on the test set of the same dataset used for training. In contrast, the off-diagonal elements show performance on test sets from different dataset sources. The final column shows the overall performance of the models on the combined test set of all dataset sources.

The energy RMSEs are barely affected by the sampling method, with errors ranging between 0.06 eV and 0.10 eV. The force RMSEs range over a wider range between 0.12 eV/Å and 0.17 eV/Å while also displaying a more significant effect of the sampling method. The trend for both metrics mirrors previous observations: models trained on the 300 K and 500 K simulations perform as well on their own test sets as on each other's, consistent with their structural similarity. Both models perform worse on the test set with the modified force field parameters. The modified force field model also shows reduced performance on the 300 K and 500 K simulation test sets compared to its own test set. However, this decline is less pronounced than the performance drop observed for the 300 K and 500 K models when evaluated on the modified force-field test set. This indicates that the modified force field model is more robust across the different sampling methods, which can be attributed to the increased structural diversity of its training dataset.

The complete cross-validation results for 4G-HDNNP, SchNet, PaiNN, Base MACE, and QEq-MACE models trained on different dataset sources of alanine dipeptide in vacuum are shown in Appendix D.1. The results for the different sampling methods show similar trends as discussed here and mainly differ in the errors achieved by the more advanced models.

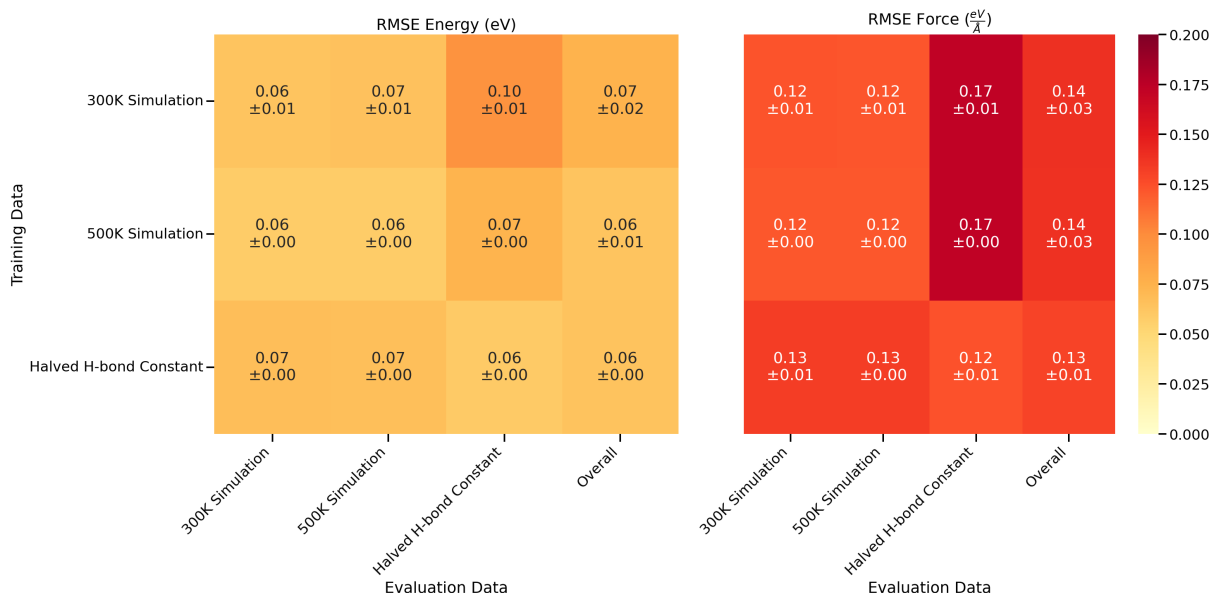


Figure 4.7.: Cross-validation for 2G-HDNNP models trained on different dataset sources of alanine dipeptide in vacuum.

4.1.4. Conclusion

For the alanine dipeptide, metadynamics datasets generated at 300 K and 500 K yield largely overlapping configurational and property distributions, indicating that temperature increases alone do not substantially extend the relevant structural coverage in this setup. In contrast, sampling with modified force field parameters systematically enriches sparsely populated regions in both descriptor and CV space while preserving physical plausibility at the Density Functional Theory (DFT) reference level.

Cross-validation confirms this trend: models trained on modified force-field data transfer more robustly to the other datasets than models trained on the other datasets transfer to the modified force-field model. For this system, the most effective strategy is the deliberate inclusion of edge-case structures through controlled force-field modifications to improve robustness through diversity. The differences are, however, rather insignificant for this system.

4.2. Thiol Disulfide Exchange Reaction

Disulfide bonds are a recurring motif in proteins, where they play a crucial role in folding and stability. The amino acid cysteine contains a thiol group that can form a disulfide bond via post-translational oxidation.^{199,200} Disulfide bonds occur for example in hair keratin^{200,201}, in the blood-clotting protein fibrinogen²⁰², the coenzyme lipoic acid, and the antioxidant glutathione. They play functional roles through cross-links or redox activity.

The thiol disulfide exchange reaction is a S_N2 reaction between a thiolate and a disulfide ($R-S^- + R'-S-S-R' \longrightarrow R-S-S-R' + R'-S^-$, see Figure 4.8).^{168,200,203–205} It is a biologically relevant reaction that plays a crucial role in the rearrangement of disulfide bonds in proteins. The reaction can occur uncatalyzed²⁰⁶, but specialized glutaredoxin enzymes, also known as thioltransferases, catalyze it in vivo to maintain the redox state of the cell.^{207,208}

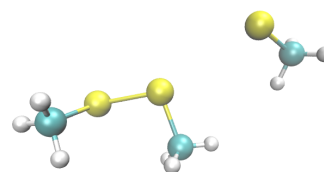


Figure 4.8.: Structure of the thiol disulfide system.

The thiol disulfide exchange reaction exhibits a strong solvent effect, which the Born solvation model can qualitatively explain.^{209–211} The Born solvation model describes the solvation free energy of an ion as a function of its reciprocal effective radius r_0 , $\Delta G_{\text{solv}} \propto \frac{1}{r_0}$. The transition state with a linear S-S-S geometry has a more delocalized charge compared to the reactant and product states and therefore a larger effective radius, which leads to a less favorable solvation free energy. Normally, this results in an increased reaction barrier in solution compared to vacuum. Here, the most extreme case occurs where the transition state is energetically favored in vacuum, while the reactant and product states are more stable in water.²¹²

This is illustrated in Figure 4.9, which shows the PMF of the thiol disulfide exchange reaction in water. The electrostatic interactions are scaled with different scaling factors to weaken or strengthen the solvent effect. Without electrostatic interactions (scaling factor 0.0, the ‘vacuum’ case), the linear configuration around $2.5 \text{ \AA}/2.5 \text{ \AA}$ is energetically favored. As the electrostatic interactions scale, the reactant and product states become more stable than the transition state. This makes the thiol disulfide exchange reaction an interesting test case for training MLPs across different environments.

4.2.1. Geometry Comparison

Five sampling methods are compared for generating training data for the thiol disulfide exchange reaction, as described in Section 2.2.1: an unrelaxed potential energy scan, a DFTB simulation in water, classical protein simulations, a classical simulation with modified force-field parameters, and uncertainty-

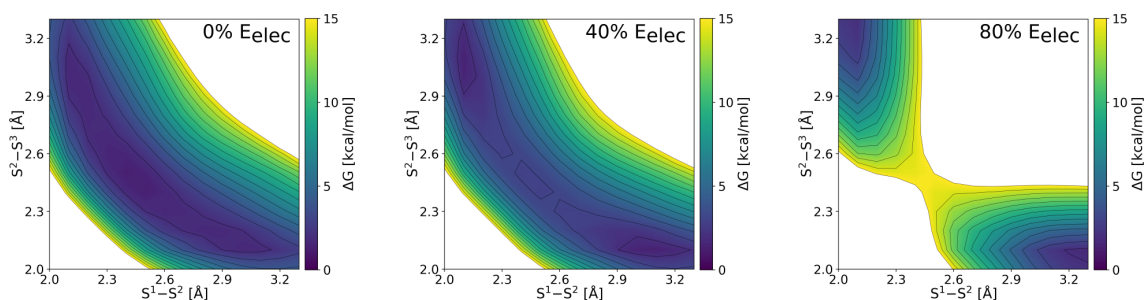


Figure 4.9.: PMF of the thiol disulfide exchange reaction in water at the DFTB3/3OB_{Mod} $S-S$ level of theory with different scaling factors for the electrostatic interactions. The linear configuration is energetically favored in the absence of interactions (left), whereas the reactant and product states are more stable with increased scaling of electrostatic interactions (middle and right).

based sampling. Since halving the force-field parameters in the alanine dipeptide system did not lead to unphysical structures, the parameters are further reduced to 25% of their original values for this reaction.

The tSNE plot of the bond lengths of the different datasets in Figure 4.10 serves as a first qualitative comparison of the structural diversity without requiring knowledge about a suitable set of CVs. The scan (blue) preserves its grid-like structure from the high-dimensional space in the low-dimensional tSNE plot. The scan gradually alters the structures along the reaction coordinate, keeping neighboring structures similar. Only uncertainty sampling (brown) shows some similar structures. As the only other vacuum simulation, it samples the region around the transition state covered by the scan (see Figure 4.9). The classical protein simulation (green) shows three clusters, corresponding to different proteins simulated. The DFTB simulation in water (orange) is the only MD simulation that can undergo a reaction and therefore spreads across the tSNE plot. Less overlap with the scan and uncertainty sampling is observed, as solvent effects lead to different structures compared to vacuum, despite the metadynamics biasing. The classical simulation with modified force-field parameters (purple) exhibits a unique, non-overlapping section in the tSNE plot, but also shows some overlap with the other sampling methods in another section of its cluster.

When comparing the location of the structures in the CV space of the two distances S^1-S^2 and S^2-S^3 in Figure 4.11, the clusters from the tSNE plot can be assigned to different regions of the CV space. The scan spans the transition-state region from 1.8 Å to 3.2 Å for both distances. Uncertainty sampling is also located in the transition-state region, as the linear $S-S-S$ region is favored in vacuum. Classical protein simulations are limited to the reactant and product basins. Still, the definition of the disulfide bond varies across the four simulated proteins, leading to three distinct clusters in the CV space. The third basin at large S^1-S^2 and S^2-S^3 distances corresponds to a structure where the S^1-S^3 disulfide bond formed. The modified force-field parameter simulation visits only one of these basins, but it is less restrictive and spans a wider range of distances. The DFTB simulation in water visits both the reactant and product states, but the transition-state region is visited less frequently despite the metadynamics bias.

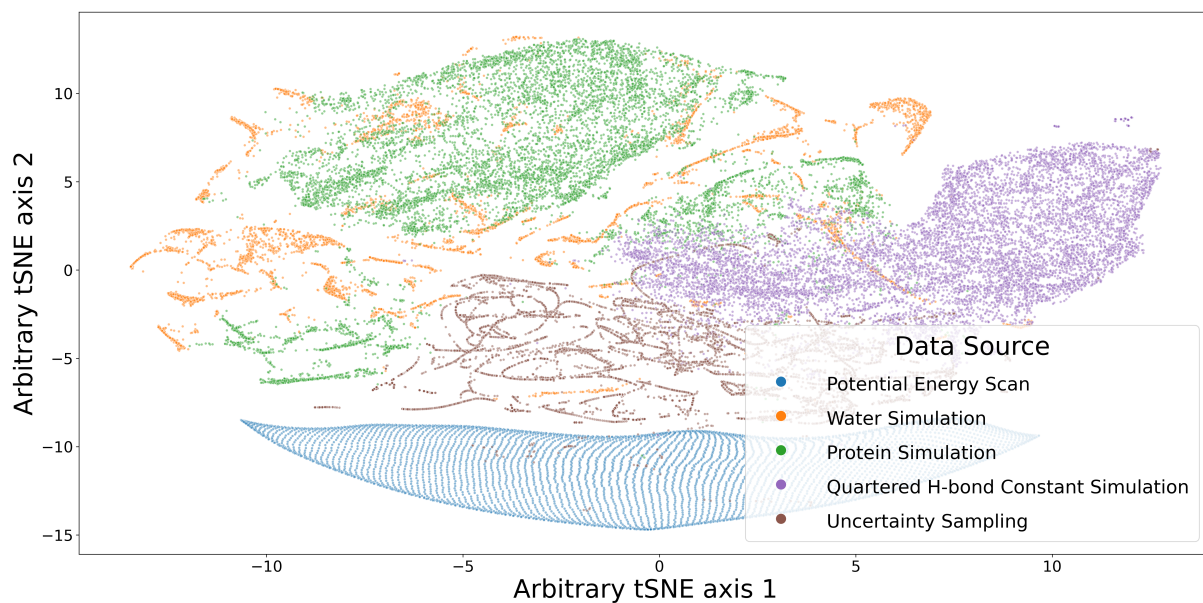


Figure 4.10.: Comparison of sampling methods for thiol disulfide exchange reaction using tSNE dimensionality reduction applied to bond lengths. A clear separation between sampling methods is evident. Clusters can be assigned in accordance with the CV space locations in Figure 4.11.

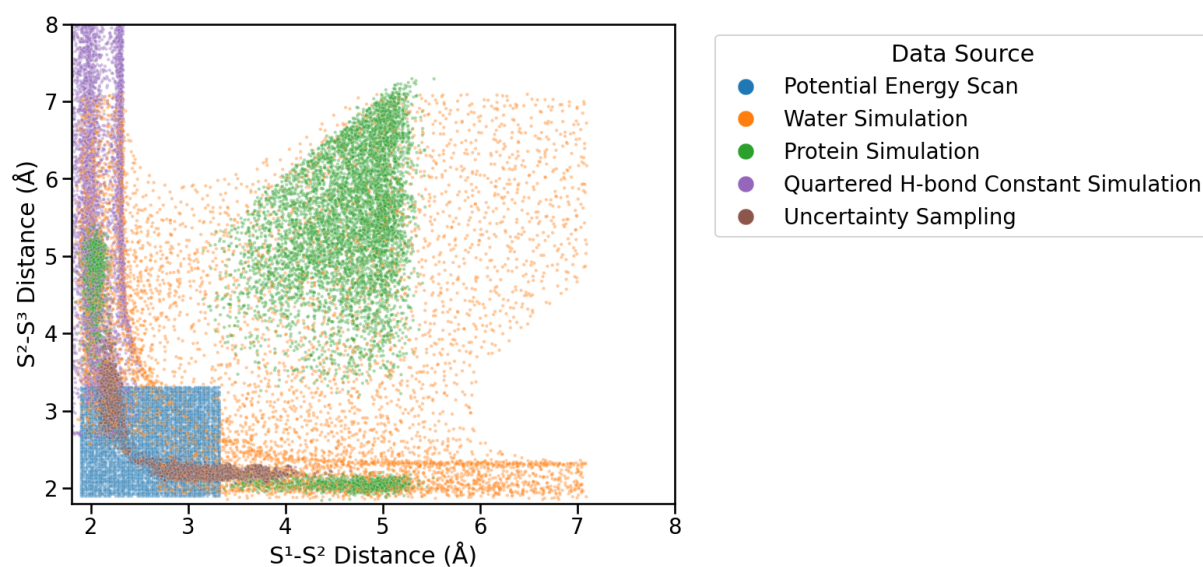


Figure 4.11.: Comparison of sampling methods for thiol disulfide exchange reaction in CV space of the two distances S^1 - S^2 and S^2 - S^3 . Sampling methods cover different regions of the CV space. Force-field methods (green, purple) cannot undergo reactions and are limited to reactant/product basins. DFTB simulations (orange, brown) visit both reactant and product states. The simulation in water (orange) avoids the transition-state region, whereas the potential energy scan (blue) focuses on it.

The plausibility of the structures generated with weakened force-field parameters is assessed by comparing the distribution of C-H bonds across the different sampling methods. This is necessary again because the parameters are further reduced to 1/4 of their original values. Also, as described in Section 2.2.1, only the longest bonds are filtered for the modified force field parameters (purple), which leads to the spike of elongated bond lengths at the tail of the distribution in Figure 4.12 in addition to the broader distribution of bond lengths. The protein simulation, as the only other classical simulation, exhibits a spike at the preset bond length of the linker atom and a narrow distribution compared to the QM/MM simulation in water (orange) and the DFTB-mimicking uncertainty sampling (brown). The scan (blue) did not vary the C-H bond; no extreme values are present.

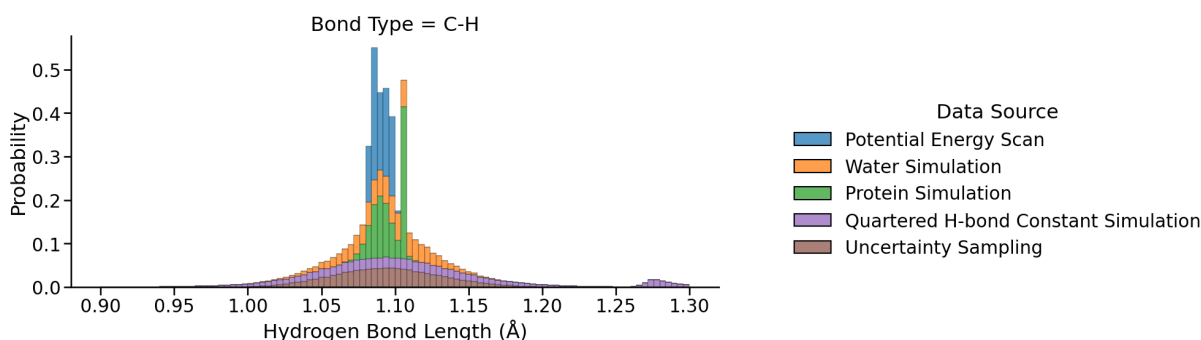


Figure 4.12.: Distribution of bond lengths of bonds involving hydrogen for thiol disulfide exchange reaction. Spike in protein simulation (green) due to the linker atom with preset bond length. Modified force-field parameters (purple) lead to a broader distribution of bond lengths, with a spike in elongated bond lengths due to the specific filtering for the longest bonds.

These observations lead to three conclusions:

1. The structures generated with the modified force field parameters are physically plausible, as the distribution of bond lengths is not significantly broader than the other sampling methods.
2. The unmodified force field parameters are too restrictive in comparison to Quantum Mechanical (QM) methods, as the distribution of bond lengths is significantly narrower than the other sampling methods. A model trained on this dataset would likely struggle to make accurate predictions for structures with elongated bonds.
3. A combination of structure generations in different environments is helpful to cover the relevant configurational space of the thiol disulfide exchange reaction. The metadynamics simulation in water and the potential energy scan complement each other, whereas the modified force-field parameter simulation introduces extreme structures into the dataset.

4.2.2. Property Comparison

The structures generated by the different sampling methods were used as references for calculations at the B3LYP/aug-cc-pVTZ level of theory in vacuum and in water to compare property distributions across the datasets. The energy and force magnitude distributions for the vacuum case are shown in Figure 4.13. The scan (blue) yields structures with the lowest energies and forces, as expected, since the linear S-S-S conformation is energetically favored in vacuum. The modified force-field parameters (purple) yield structures with significantly higher energies and forces. The modified force field parameters yield a median energy approximately 1.0 eV higher than the water simulation (orange), yet fall within 1.5xInterQuartile Range (IQR), a common threshold for identifying outliers indicated by the boxplot whiskers. The median force magnitude of the modified force-field parameter structures also lies within 1.5xIQR of that in the water simulation. Therefore, these structures are considered plausible, especially for training in water, where the reactant and product states are more stable than the transition state.

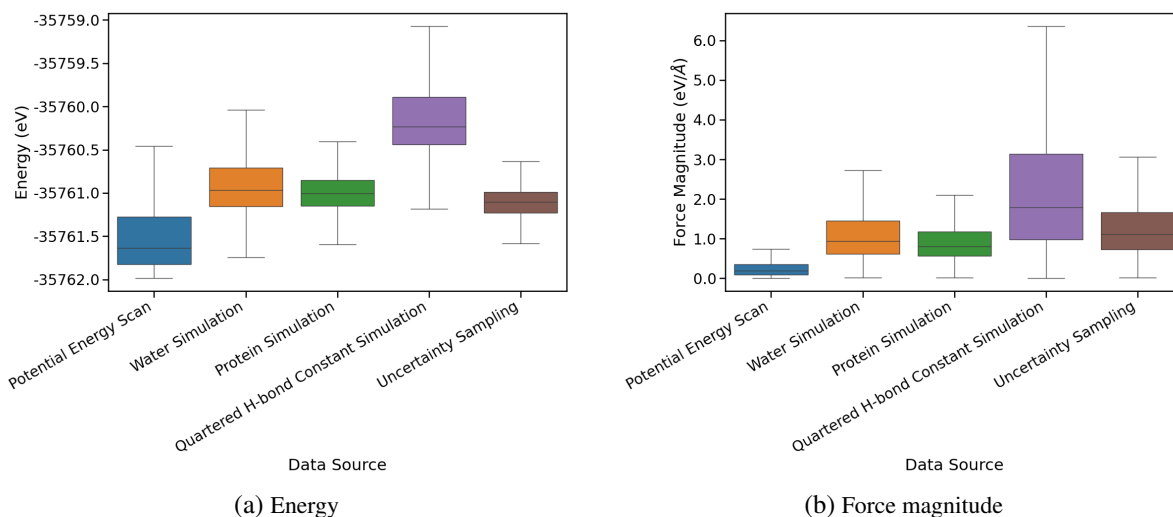


Figure 4.13.: Comparison of sampling methods for thiol disulfide exchange reaction regarding (a) energy and (b) forces across different sampling methods. Typical transition state structures from the potential energy scan (blue) are energetically favored in vacuum. Modified force-field parameters (purple) yield structures with higher energies and forces at the B3LYP/aug-cc-pVTZ level of theory in vacuum.

4.2.3. Cross-validation

Again, MLPs are trained and cross-validated on the different datasets for the thiol disulfide exchange reaction. The same models and training procedure as for the alanine dipeptide are applied; however, the training data subsampling was restricted to 4000 data points, since the scan dataset contains only 5000 data points. The result of the cross-validation for the 2G-HDNNP models is shown in Figure 4.14 in a confusion matrix.

The model trained on the scan dataset (top row) can only make accurate predictions for its own test set, but performs poorly on the other test sets. This is expected, as the scan dataset is limited to the transition-state region. Uncertainty sampling (bottom row) originating from a vacuum simulation displays a similar trend, but with a less pronounced drop in performance. In turn, the model trained on the protein simulation data (third row) yields poor predictions on the scan test set, as the protein simulation lacks structures in the transition-state region. The model trained on the water simulation data (second row) performs well on all test sets except the scan test set and the modified force-field parameter test set.

The test set generated with the modified force-field parameters (fourth row) is the most challenging, even for the model trained on the modified force-field parameter dataset. Among all models evaluated on their respective test sets, this model achieves the lowest performance. However, it performs better on other test sets than on its own, suggesting that data generated with the modified force-field parameters are more diverse and yield models that generalize better across different sampling methods. Notably, this model also yields the best predictions on the scan test set among models not trained on the scan dataset, even though the modified force-field parameter dataset contains very few structures from the transition-state region. This further supports the conclusion that the modified force-field parameters yield a more robust model across different sampling methods.

The complete cross-validation results for 4G-HDNNP, SchNet, PaiNN, Base MACE, and QEq-MACE trained on different dataset sources of the thiol disulfide exchange reaction in vacuum are shown in Appendix D.1. The results for the different sampling methods show similar trends, as discussed here, and mainly differ in the errors achieved by the more advanced models.

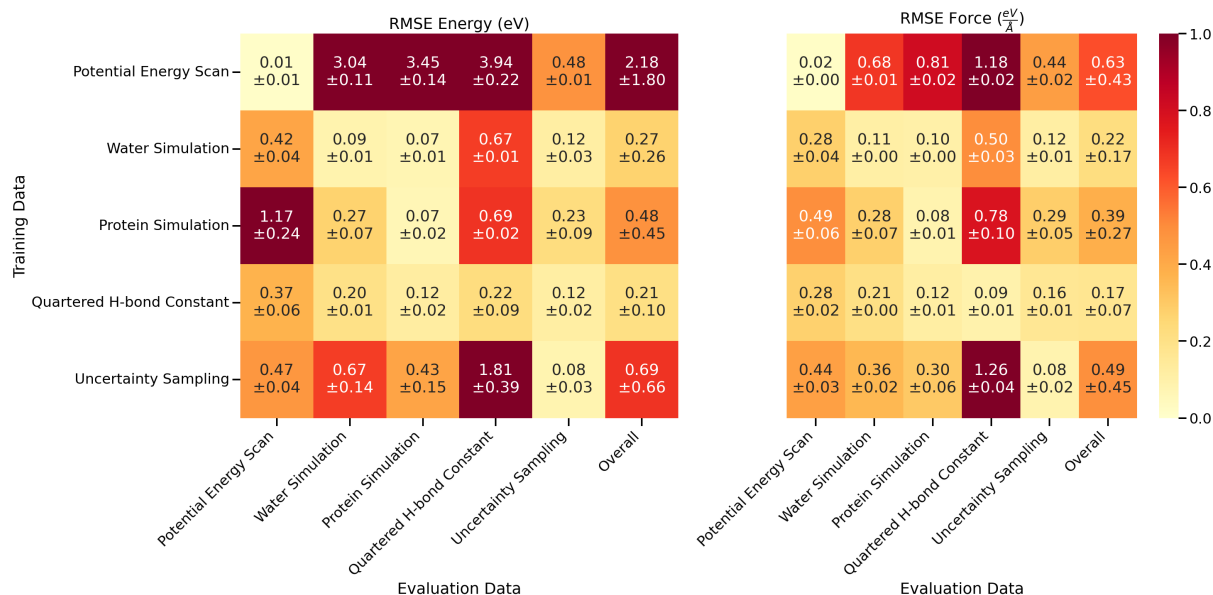


Figure 4.14.: Cross-validation for 2G-HDNNP models trained on different dataset sources of thiol disulfide exchange reaction in vacuum.

4.2.4. Conclusion

For the thiol disulfide exchange reaction, each sampling method covers a distinct part of the configurational landscape, and no single source is sufficient on its own. The potential energy scan and uncertainty sampling focus on transition-state-like vacuum structures, whereas classical protein and water simulations emphasize reactant and product basins under environmental constraints. Modified force-field sampling introduces physically extreme yet plausible structures and broadens bond-length and energy coverage beyond standard classical trajectories.

The cross-validation results demonstrate that dataset diversity is the key determinant of transferability: narrowly focused datasets (scan, protein-only) yield strong in-domain performance but poor generalization, whereas modified force-field data yield models that transfer best across heterogeneous test sets, despite being harder to fit on their own source distribution. Consequently, robust training for this system requires a combined sampling protocol that integrates complementary environments and explicitly includes edge-case configurations.

5. Electrostatic Interactions in QM/MM Simulations

Electrostatic interactions are essential for enzyme function at multiple distance scales. They are substantially weaker than covalent bonds, with non-covalent interaction energies typically ranging from -0.5 kcal/mol to -50 kcal/mol²¹³, but they direct incoming substrates to the active site through long-range effects and facilitate catalysis through specific short-range interactions.²¹⁴

In QM calculations, the unfavorable scaling of computational cost with system size prohibits the inclusion of the entire enzyme and solvent environment. The Quantum Mechanics/Molecular Mechanics (QM/MM) approach (see Section 2.4) allows the active site to be treated at a high level of theory, while the remainder of the system is described with a more efficient Molecular Mechanics (MM) model. However, the accuracy of QM/MM simulations depends critically on the representation of electrostatic interactions between the QM and MM regions.

The electrostatic interaction is the interaction between the MM point charges and the nuclei and electrons of the QM region. The continuous electron density of the QM region is known when employing *ab initio* methods. Therefore, the ElectroStatic Potential (ESP) can be calculated as the sum of the nuclear contributions and the integral over the electron density contributions (also compare Eq. (2.112)). However, even with the QM/MM approach, the time scales of enzymatic reactions are often inaccessible. Extensive sampling is required to capture rare events such as substrate binding, conformational changes, and chemical reactions. Semi-empirical methods, such as DFTB, can reduce computational cost. A common approximation is to represent the QM region with a set of partial charges instead of the full electron density:

$$E_{\text{QM/MM}}^{\text{elec}} = \kappa \sum_j^{MM} q_j \left(\sum_i^{QM} \frac{Z_i}{|\mathbf{R}_i - \mathbf{r}_j|} - \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}' - \mathbf{r}_j|} d\mathbf{r}' \right) \quad (5.1)$$

$$\approx \kappa \sum_j^{MM} \sum_i^{QM} \frac{q_j q_i}{|\mathbf{R}_i - \mathbf{r}_j|} \quad (5.2)$$

Long-range interactions are difficult to capture with commonly applied local descriptors. Some architectures, such as the 4th Gen. High-Dimensional Neural Network Potential (4G-HDNNP), explicitly incorporate long-range interactions, but they still rely on partial-charge assignment to represent the ESP.

Even fewer ML models attempt to predict the electron density directly^{215–217}, instead, most use atom-based descriptors to predict atomic properties such as partial charges or multipole moments. The error introduced by switching from a continuous charge distribution to a discrete set of partial charges must be quantified to assess the limitations of this approximation and to guide the development of more accurate ML potentials for QM/MM simulations.

In this chapter, the reproduction of the ESP using different partitioning schemes for partial charge assignment is investigated. We compare traditional methods such as Hirshfeld, Mulliken, and Löwdin with ESP-based fitting methods such as CHarges from ELectrostatic Potentials using a Grid-based method (CHELPG), Merz-Kollman, Restrained Electrostatic Potential (RESP), and Minimal-Basis Iterative Stockholder (MBIS). The performance of these methods is evaluated based on their ability to reproduce the ESP and electrostatic energy of a continuous charge density for the two model systems, alanine dipeptide and thiol disulfide exchange. We also analyze the effect of the environment (vacuum vs. solvent) on the electrostatic properties and the implications for QM/MM simulations. Environmental changes should lead to a polarization of the electron density, which should be reflected in the partial charges. Finally, the training performance of a 4G-HDNNP model using different partial-charge methods is compared to assess how well these methods can be learned by MLPs. Ultimately, the goal is to decide which partial charge method is best suited for MLP in QM/MM simulations based on the balance between its continuous ESP reproduction accuracy and its learnability by ML models.

5.1. Methodology

100 random structures from each sampling method from the Chapter 4 were randomly selected for both systems. Mulliken, Löwdin, and Hirshfeld partial charges were obtained during the reference method calculations at the B3LYP/aug-cc-pVTZ level with the ORCA software package¹⁶⁷. ORCA output files containing information about the converged basis functions (gbw file) were used as an input to the Multiwfn software¹¹⁵ to calculate CHELPG, Merz-Kollman (MK), Electronegativity-Equalization Method (EEM), RESP, and MBIS partial charges. The EEM method was not used for the thiol disulfide exchange reaction due to a missing parameter for sulfur.

Two approaches were used to calculate the ESP: the integration over the continuous charge density and the summation over discrete point charges. Corresponding vacuum and solvent calculations both used the MM atom positions of the solvent calculation to evaluate the ESP at. By comparing the electrostatic interactions in the vacuum and solvent environments, the solvent-induced polarization of the electron density can be quantified. Multiwfn was used to compute the ESP of the continuous charge density at the respective MM atomic positions, using ORCA's gbw files as input. The ESPs caused by the partial charges at the same set of MM atom positions was calculated using the standard Coulomb's law in a custom Python script. Electrostatic energies were calculated by multiplying the ESP values with the

respective MM point charges and summing over all MM atoms. The Root Mean Square Error (RMSE) between the continuous ESP and the partial charge ESP, as well as the RMSE between the electrostatic energies caused by the two ESPs, were used as metrics to evaluate the performance of each partial charge method.

5.2. Alanine Dipeptide

5.2.1. Partial Charge Method Comparison

Chemical reasonableness is the first aspect to consider when evaluating the performance of a partial-charge method. In Figure 5.1, the distribution of partial charges per element for the alanine dipeptide system in vacuum is shown for different partitioning schemes. Carbon exhibits a very broad distribution of partial charges across all methods, as expected given the heterogeneous chemical environments of carbon atoms in the system. However, the subdistributions for carboxyl carbon, alpha carbon, and methyl carbon are not resolved in the boxplot. Mulliken charges yield the largest values for carbon, with some partial charges even exceeding 1 e and -1 e. Chemical intuition suggests that oxygen should carry the most negative charge. Among the considered charge schemes, only Hirshfeld and Mulliken charges fulfill this expectation in every case. EEM sets nitrogen slightly more negative than oxygen, and the charge fitting methods (CHELPG, MK, RESP) and MBIS have some outliers where nitrogen is more negatively charged than oxygen. Löwdin charges completely fail to reproduce a chemically reasonable charge distribution, yielding positive charges on oxygen and nitrogen. Hirshfeld charges show the smallest spread across all elements. The charge-fitting methods and MBIS have been combined into a single category, as they differ only slightly.

Overall, Hirshfeld charges show the most chemically reasonable distribution, as they consistently reproduce the expected order of element partial charges ($O < N < C$). In contrast, only Löwdin charges fail to reproduce a chemically reasonable charge distribution.

5.2.2. Environment Comparison

The presence of solvent leads to a polarization of the electron density, which is reflected in the ESP, as shown in Figure 5.2, where the ESP from the continuous charge density is mapped onto the van der Waals surface of the molecule. Negative regions of the ESP around the carbonyl oxygen and positive regions around the amide nitrogen, a potential hydrogen bond donor, are present in both environments. However, these features are more pronounced in the solvent environment, as illustrated by the difference plot.

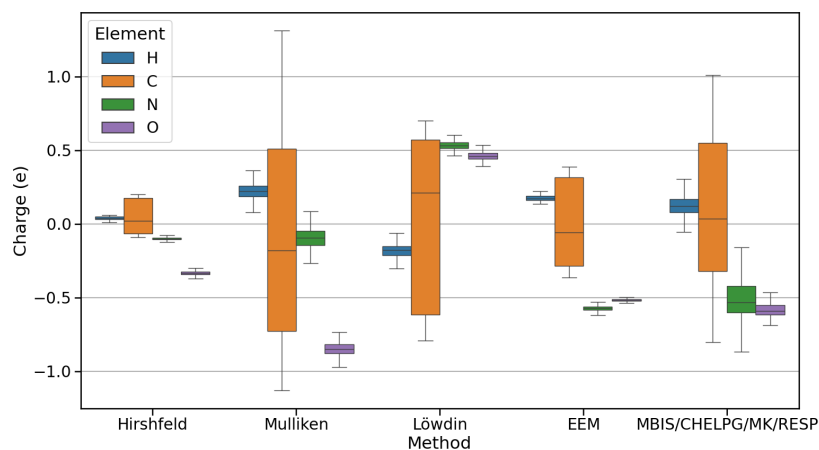


Figure 5.1.: Comparison of partial charge methods for alanine dipeptide system in vacuum. Hirshfeld charges satisfy the expected order ($O < N < C$); Löwdin charges are unreasonable, with positive O and N charges.

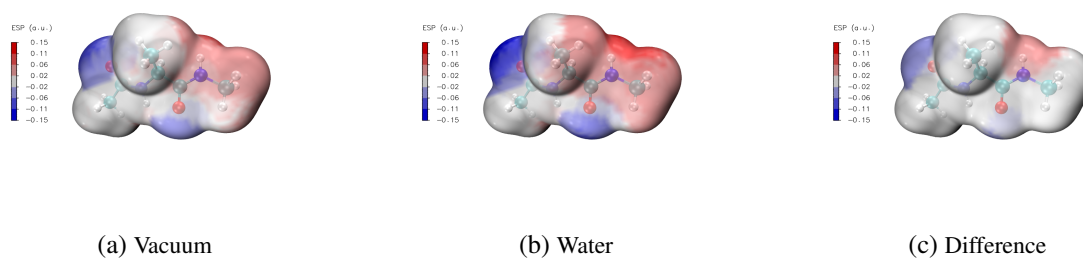


Figure 5.2.: ESP map for alanine dipeptide system in vacuum (a), in water (b), and the difference between both environments (c). The solvent environment leads to a more pronounced ESP with stronger negative regions around the carbonyl oxygen and stronger positive regions around the amide nitrogen.

An ideal partial-charge method should capture this polarization effect, which can be quantified by comparing the difference in partial charges between the solvent and vacuum environments for each method. In Figure 5.3, the difference in partial charges between the two environments is shown for selected methods. Hirshfeld and Löwdin charges exhibit the expected trend: more negative charges for oxygen and more positive charges for nitrogen in the solvent environment, but the magnitude of the change is small. The charge-fitting methods and MBIS exhibit a similar trend, but with a larger magnitude of change, indicating that their fits capture the polarization effect more effectively. However, the median change for the nitrogen is unexpectedly slightly negative for these methods. The behavior of the nitrogen atom is complex, as it can participate in hydrogen bonding and is part of the amide group, which has resonance structures that can affect the charge distribution. A stabilized positive charge on its hydrogen would make the nitrogen more negative. In contrast, a stabilized negative charge on the oxygen would favor the resonance structure with a positive charge on the nitrogen. Changes in Mulliken charges do not follow a clear trend: the overall median change is close to zero, even for oxygen, and the distribution of changes is relatively broad. Because the EEM charges are calculated solely from the QM region, they are insensitive to the environment. As a result, the partial charges remain unchanged between the vacuum and solvent environments, making this approach unsuitable for this application.

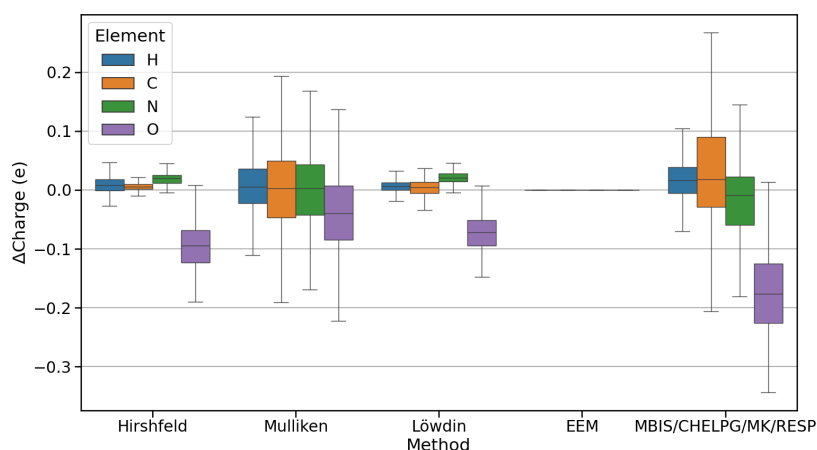


Figure 5.3.: Difference in partial charges for alanine dipeptide system between solvent and vacuum environment for selected methods. Clear trend of more negative charges for oxygen for Hirshfeld, Löwdin, charge fitting methods, and MBIS, with charge fitting methods showing a larger magnitude of change. Mulliken charges show no clear trend and a broad distribution of changes. EEM charges show no change due to insensitivity to the environment.

5.2.3. Electrostatic Potential Reproduction

The objective of this section is to evaluate how well different partial charge methods can reproduce the ESP of the continuous charge density. However, raw ESP RMSEs values are hard to interpret intuitively. Therefore, the electrostatic energy is included to put the error magnitude in perspective relative to the chemical accuracy of 1 kcal/mol (0.043 eV). Still, the electrostatic energy RMSE has the drawback of

error cancellation of positive and negative errors in the ESP, which is why both errors are displayed for the alanine dipeptide system in solution in Figure 5.4.

The ESP fitting methods (CHELPG, MK, RESP) and MBIS significantly outperform traditional QM charge partitioning schemes (Hirshfeld, Mulliken, Löwdin) in reproducing the ESP and electrostatic energy. This is expected, as the ESP fitting methods are designed to reproduce the ESP at specific points in space. The RMSE of the ESP reproduction for the charge fitting methods and MBIS is around 0.02 eV/e, in contrast to 0.15 eV/e for Hirshfeld to 0.18 eV/e for Mulliken. Löwdin charges perform the worst by a large margin. The electrostatic energy reproduction shows a similar trend. Mulliken charges have a slightly better performance than Hirshfeld charges for electrostatic energy reproduction due to error cancellation. Additional data for the ESP reproduction can be found in Appendix D.2.

While the ESP fitting methods show insignificant errors, Mulliken and Hirshfeld charges introduce an error of around a quarter and a half of the chemical accuracy threshold, respectively, simply by switching from a continuous charge distribution to a discrete set of partial charges. Therefore, CHELPG, MK, RESP, and MBIS are a more suitable target for the MLP training.

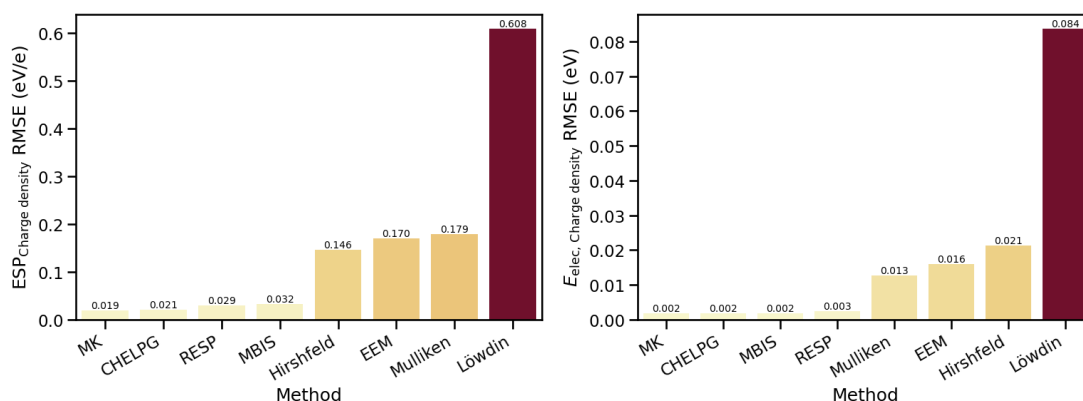


Figure 5.4.: Comparison of charge partitioning methods for ESP and electrostatic energy reproduction of continuous charge distribution for the alanine dipeptide system in solution. ESP fitting methods (CHELPG, Merz-Kollman, RESP) and MBIS significantly outperform traditional partitioning schemes (Hirshfeld, Mulliken, Löwdin).

5.2.4. Charge Training

Another aspect to consider when selecting a partial charge method for MLP training is how well the model can learn the charges from descriptors that represent the local chemical environment. The training performance of a 4G-HDNNP model using different partial charge methods for the alanine dipeptide system in vacuum is shown in Figure 5.5. The charges were the only target property, not the energy or forces, to isolate the learnability of the charge assignment from the overall performance of the MLP. The hyperparameters were adopted from Section 6.1.1.

Hirshfeld, EEM, and MBIS show the best training performance, with a validation RMSE of around 0.03 e. The other methods exhibit significantly worse training performance, with RMSE values twice or even three times higher.

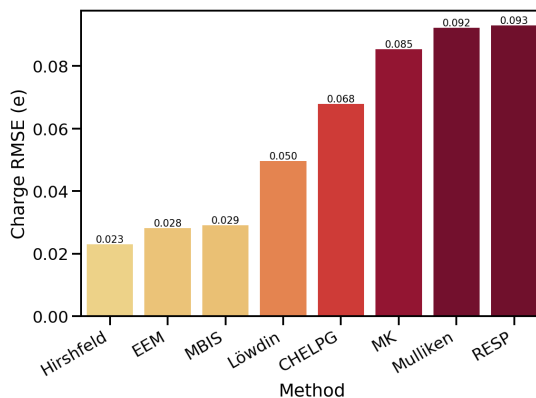


Figure 5.5.: Comparison of training performance for different partial charge methods for alanine dipeptide system in vacuum using the 4G-HDNNP architecture. Hirshfeld, EEM, and MBIS show the best training performance.

5.2.5. Conclusion

Overall, MBIS and Hirshfeld charges exhibit the best balance among chemical reasonability, ESP reproduction, and learnability by the MLP, making them the most suitable targets for MLP training in QM/MM simulations. Charge-fitting methods reproduce ESP well but exhibit poorer training performance, which may be due to noise in the charges arising from fitting an already fitted property. Mulliken charges perform poorly in both ESP reproduction and training, whereas Löwdin charges perform worse in both aspects. Hirshfeld charges will be used for the remainder of the thesis, although MBIS charges could be a promising alternative. An ML/MM approach by Zinovjev^{153,154} selected MBIS charges. Switching from the continuous charge distribution to Hirshfeld charges as charge representation is expected to introduce an electrostatic energy error of 0.021 eV, approximately half the chemical accuracy threshold, as quantified in Figure 5.4

5.3. Thiol Disulfide Exchange Reaction

5.3.1. Partial Charge Method Comparison

The procedure for evaluating the partial charge methods for the thiol disulfide exchange reaction is the same as for the alanine dipeptide system. The distribution of partial charges per element for the thiol disulfide exchange reaction in vacuum is shown in Figure 5.6. Chemical intuition suggests that sulfur should carry the negative charge in the system and exhibit a broad charge range as the charge state changes during the reaction. Carbon may carry a small negative charge inductively.

Nearly all methods show a broad distribution of sulfur charges, except for Löwdin charges, which show a positive sulfur charge. Hirshfeld and MBIS charges are consistent with the expectation of a small negative charge on carbon. In contrast, Mulliken carbon charges are more negative than sulfur charges on average, and the charge-fitting methods indicate a broad positive charge on carbon.

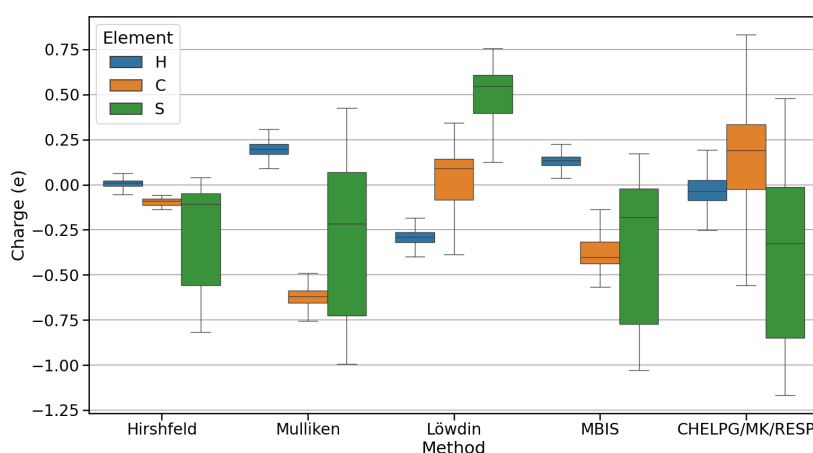


Figure 5.6.: Comparison of partial charge methods for thiol disulfide reaction in vacuum. Hirshfeld and MBIS show an intuitive broad distribution of negative sulfur charges and small negative carbon charges.

5.3.2. Environment Comparison

With sulfur being the most polarizable element in the system, a significant effect of the environment on the charge distribution and ESP can be expected. The effect will differ between the transition state and the reactant/product states and between the disulfide/thiolate sulfurs. The overall ESP is negative due to the negative charge as depicted in Figure 5.7. Figure 5.8 shows the difference in partial charges between the solvent and vacuum environments for selected methods. All methods show a median change of approximately zero across all elements. Mulliken and charge-fitting methods show a broader distribution, with carbon varying the most. However, no clear assignment of the polarization effect to specific elements can be observed overall.

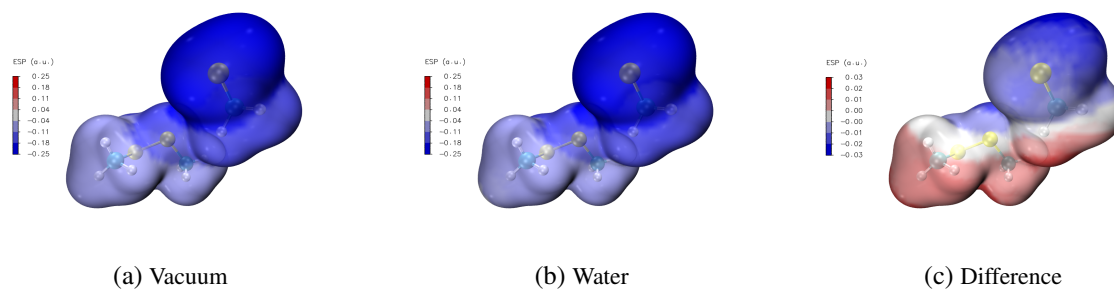


Figure 5.7.: ESP map for thiol disulfide reaction in vacuum (a), in water (b), and the difference between both environments (c).

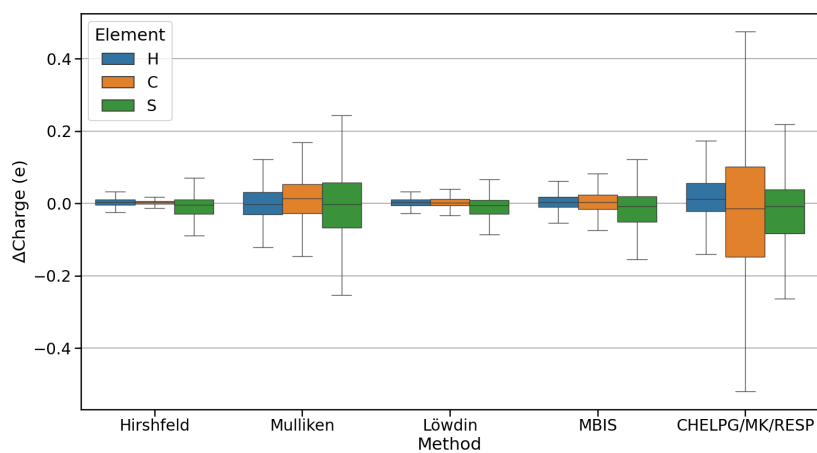


Figure 5.8.: Difference in partial charges for thiol disulfide reaction between solvent and vacuum environment for selected methods. No clear polarization effect for specific elements can be observed.

5.3.3. Electrostatic Potential Reproduction

The comparison of Figure 5.4 and Figure 5.9 shows the worse performance of the charge fitting methods and MBIS for the thiol disulfide exchange reaction relative to the alanine dipeptide system. In contrast, Mulliken and Hirshfeld charges exhibit similar performance in reproducing the ESP. The electrostatic energy performance is improved for the latter methods due to error cancellation. Additional data for the ESP reproduction can be found in Appendix D.2.

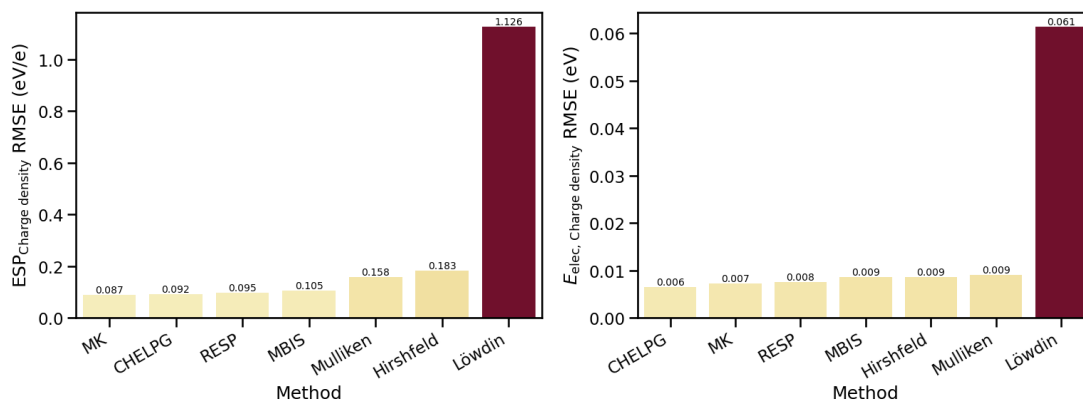


Figure 5.9.: Comparison of charge partitioning methods for ESP and electrostatic energy reproduction of continuous charge distribution for the thiol disulfide exchange reaction in solution.

5.3.4. Conclusion

Overall, the method choice has less impact on the ESP reproduction for the thiol disulfide exchange reaction than for the alanine dipeptide system. Therefore, for consistency, Hirshfeld charges were used for MLP training in this system as well. Errors of 0.009 eV for the electrostatic energy reproduction are introduced by switching from a continuous charge distribution to a discrete set of partial charges, which is approximately a fifth of the chemical accuracy threshold, and therefore acceptable for QM/MM simulations.

6. Machine Learning Potentials for QM/MM

6.1. Training Alanine Dipeptide in Vacuum

Vacuum MLPs have gained significant attention in recent years.^{43,124,128,130,131} They provide a computationally efficient alternative to QM methods for simulating molecular systems while retaining high accuracy. The main focus of this work is the application of MLPs in QM/MM simulations; however, it is crucial to first evaluate their performance in vacuum, where they are expected to perform best. This serves as a baseline for understanding their capabilities and limitations before deciding which MLP architecture to use for the more complex QM/MM simulations.

In this section, we present a comprehensive comparison of various MLP architectures trained on the alanine dipeptide system in vacuum. Some of the MLPs are purely designed for vacuum (2G-HDNNP¹²³, SchNet¹²⁴, PaiNN¹²⁹, Base MACE¹³¹), others have ML/MM capabilities (4G-HDNNP¹²⁸, QEq-MACE, AMP¹⁵⁶, Field-Schnet¹⁵⁵, and Field-MACE¹⁵⁹). These architectures are introduced in Section 2.5.2. They represent a diverse set of state-of-the-art MLP architectures, of which especially Base MACE has gained significant attention in recent years for its high accuracy and efficiency, as seen in the number of citations of the original publication¹³¹ (see Figure 6.1).

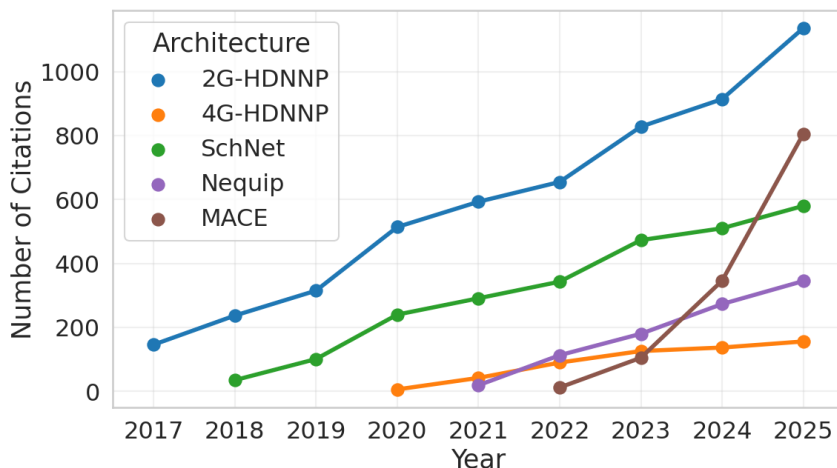


Figure 6.1.: Number of citations of popular MLP architecture publications over time. MACE has gained significant attention in recent years.

Initially, hyperparameter optimization is performed for most architectures to determine optimal settings and assess the influence of different hyperparameters on model performance and training duration. Subsequently, we analyze the dataset size dependence of each architecture to understand how the amount of training data affects their performance. Finally, we perform a metadynamics simulation in vacuum using each MLP to assess its ability to reliably reproduce the reference QM potential energy surface and forces during a dynamic simulation. A special focus is placed on the stability of the simulations. The results of this section provide valuable insights into the strengths and weaknesses of each MLP architecture in vacuum.

Author Contributions Literature implementations of all MLP architectures were used. Martin Vondrak implemented an initial MACE version with a charge equilibration scheme from the group of Prof. Dr. Johannes Margraf in Bayreuth. For both 4G-HDNNP and QEq-MACE, the author of this thesis implemented the ML/MM extension and the interface to GROMACS. The author performed the training, simulation, and analysis.

6.1.1. Hyperparameter Optimization

A hyperparameter is a parameter that is not learned during training but is set before training and can significantly influence the model’s performance. Examples of hyperparameters include the learning rate, the number of layers in a neural network, the number of neurons per layer, the activation function, and the batch size used during training. Hyperparameter optimization is crucial for achieving optimal performance from machine learning models. Various strategies exist for hyperparameter optimization, including grid search, random search, and more sophisticated methods like Bayesian optimization, genetic algorithms, and hyperband²¹⁸. In this work, a combination of grid search and hyperband is employed. Grid search systematically explores a predefined set of hyperparameter values. Only a limited search space can be explored systematically using grid search due to the combinatorial explosion of possible hyperparameter combinations. Hyperband, by contrast, is a more efficient method that allocates resources to promising hyperparameter configurations while quickly discarding poor performers. It works in the following way:

1. A large number of hyperparameter configurations are randomly sampled from the search space and trained for a small number of epochs.
2. The performance of each configuration is evaluated, and only the top-performing fraction $\frac{1}{\eta}$ of configurations is retained. The factor η is a hyperparameter that controls the aggressiveness of the pruning.

3. The retained configurations are then trained for η times more epochs, before the next round of pruning is performed.
4. This process continues until either a single configuration remains or a predefined maximum number of epochs is reached.

Grid search was used to explore the impact of specific hyperparameters in isolation. In contrast, hyperband was employed to efficiently explore a larger hyperparameter space for each MLP architecture, identifying the best-performing configuration. The most impactful hyperparameters for performance and training duration are reported in the main text. An overview of the explored hyperparameter search spaces and strategies, as well as the detailed results of the hyperparameter optimization, is provided in Appendix D.3.

2nd Generation HDNNP

The 2G-HDNNP architecture by Behler and Parrinello¹²³ (see Section 2.5.2) has two sets of architecture-specific hyperparameters: symmetry function hyperparameters and the subnetwork architecture. The hyperparameter arrays R_s , η , ζ , and λ control the shape and range of the symmetry functions encoding the local atomic environment. Their combination determines the size of the feature vector. The subnetwork architecture that predicts atomic contributions to the energy is defined by the number of layers and the number of neurons per hidden layer.

Energy layer size and eta array choice are found to be the most impactful hyperparameters for the 2G-HDNNP architecture, as shown in Figure 6.2. Increasing the energy layer size generally improves model performance with only a small increase in training duration. The performance boost converges quickly. Two small layers outperform a single large layer. The size and range of the η array have an even more significant influence. Increasing the η array size and, especially, its range substantially improves model performance with longer-range information, albeit at the cost of notably longer training durations for a large η array. Otherwise, R_s is found to have an unsystematic influence with an insignificant influence on training duration, as it does not take part in the angular symmetry functions. A larger ζ array size increases the training duration with only a minor influence on performance. The activation function appears to have only a minor influence on performance in the grid search, but only Tanh, Swish, and ELU are among the Top 10 best-performing models during the hyperband search with hundreds of trials.

4th Generation HDNNP

The 4G-HDNNP architecture adds a charge prediction layer to the 2G-HDNNP architecture, but the remaining hyperparameters are the same as for the 2G-HDNNP. In total, the effects of the hyperpa-

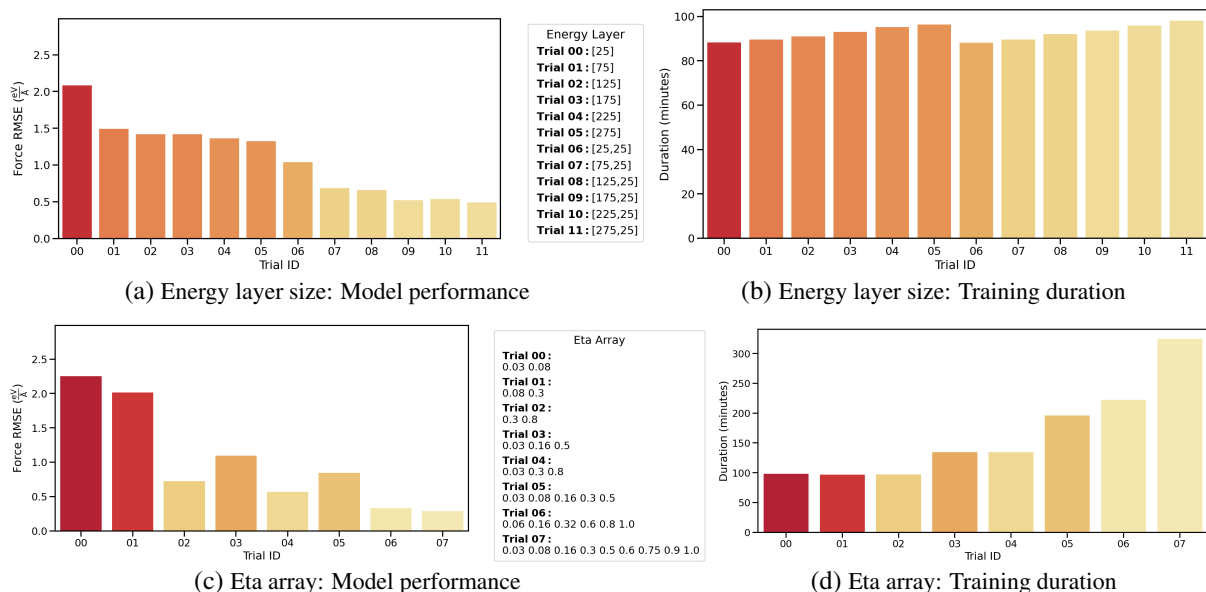


Figure 6.2.: Selected hyperparameter effects on 2G-HDNNP model performance. The architecture of the energy layer and the choice of the η array significantly influence model accuracy and training duration. Increasing the energy layer size improves performance with only a small increase in training time, whereas the choice of η array substantially improves the model's performance with long-range information, but at the cost of notably longer training durations.

rameters R_s , η , ζ , the energy-layer architecture, and the charge-layer architecture on performance and training duration are investigated.

Notably, increasing the number of units is not a guarantee for improved performance, as shown in Figure 6.3. Both the charge network and the energy network influence the performance and tend to perform better with more units, but an optimum is observed within the small search space for the energy layer. When inspecting the larger hyperband search space, most top-performing models have three layers, with higher node counts in the first layer, whereas the next layers vary. A strong influence of the η and ζ array choice is observed. Again, only Tanh, Swish, and ELU appear in the Top 10.

SchNet

SchNet¹²⁴ is a convolutional graph neural network that iteratively constructs feature vectors from a simple embedding of atomic numbers via message passing through interaction blocks. Relevant hyperparameters concern the input embedding, the interaction blocks, the convolutional filters, and the output blocks. The initial input embedding of the atomic numbers can vary in the number of units, which determines the size of the initial feature vector. The interaction blocks internally use dense layers with a certain number of units and an activation function. The depth of the network is determined by the number of interaction blocks, which determines how many times the feature vectors are updated by message passing. The convolutional filters are defined by the number of Gaussian bins with an offset and width

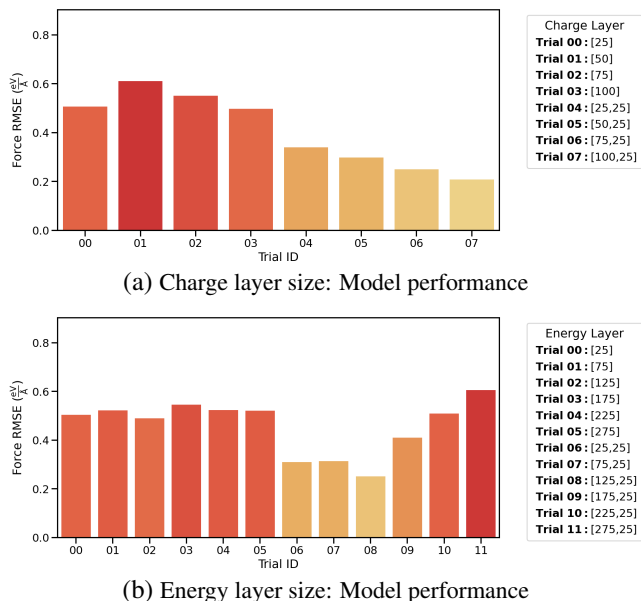


Figure 6.3.: Selected hyperparameter effects on 4G-HDNNP model performance. Larger charge and energy layers improve model accuracy, although a performance optimum was observed for the energy-layer size.

up to the cutoff distance. The output blocks consist of dense layers with a certain number of units and an activation function. Further details of the architecture are provided in Section 2.5.2.

As shown in Figure 6.4, the number of interaction blocks systematically influences model performance. However, increasing the number of interaction block units linearly scales up the training duration. Most other hyperparameters show a minor influence on this small test system. The number of Gaussian bins has a moderate influence on the performance. Surprisingly, the number of interaction blocks has little impact on performance, which might be due to the system’s small size. The best performing models in the hyperband search have 4–8 interaction blocks. Swish and Tanh consistently perform best among activation functions. The majority of the Gauss parameters for the distance expansion showed similar performance, but the cutoff distance must be sufficiently large to capture the relevant interactions ($>5 \text{ \AA}$). The default value was too small during the grid search, leading to a sudden performance boost when increasing it or using an offset (see Figure D.21 in the appendix). No impact of the output network architecture is observed.

PaiNN

PaiNN, as an equivariant version of SchNet, shares its architecture, except for the equivariant features and most hyperparameters. The most striking change in hyperparameters is the use of Bessel functions instead of Gaussian functions for the distance expansion, which is controlled by the (much smaller) number of Bessel bins, an enveloping cutoff function, and the cutoff distance. In total, the influence

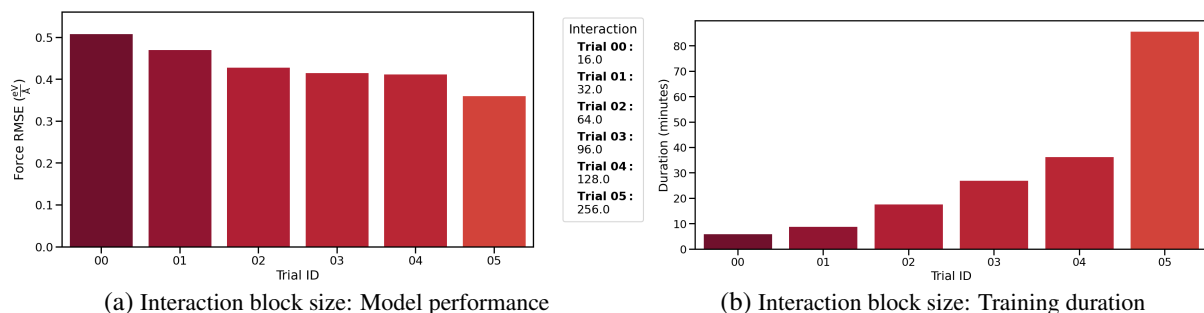


Figure 6.4.: Selected hyperparameter effects on SchNet model performance. An increased number of Gauss bins moderately improves model performance, with an acceptable increase in training duration. In contrast, increasing the number of interaction blocks improves the performance with a linear increase in training duration.

of the input embedding size (also used as the interaction block size), the number of interaction blocks, the number of Bessel bins, the exponent of the enveloping function, the cutoff distance, the output network architecture, and the activation function (both in the interaction blocks and the output blocks) is investigated.

As shown in Figure 6.5, the number of Bessel bins is found to have a strong influence on the performance, with a larger number of bins counterintuitively leading to worse performance. Consistent with the SchNet results, the size of the interaction blocks, which were adjusted together with the input embedding size, also significantly influences performance, with larger blocks yielding systematically better performance at the cost of a linear increase in training duration. For the other hyperparameters, too small cutoffs are detrimental to performance, the number of interaction blocks has little influence, and no impact of the output network architecture is observed. No generally best activation function is observed; 5/6 of the candidate activation functions are present in the Top 10. While the effect of the envelope exponent is not obvious in the grid search, 8/10 of the best-performing models in the hyperband search have an exponent of 5.

Base MACE

The MACE architecture¹³¹ is, like PaiNN, an equivariant message passing neural network. It embeds atomic numbers into a feature vector whose irreducible representation is specified by a hyperparameter (the number of channels and the maximum angular momentum $L_{\max}$). Like PaiNN, it encodes radial information using Bessel functions with a certain number of bins, an enveloping function with a certain exponent, and a cutoff distance. The radial information is then used as input to an intermediate radial Neural Network (NN) with a parameterizable architecture, denoted the radial MLP (Multi Layer Perceptron). During the message construction, a many-body expansion is performed up to a certain correlation order, which is a hyperparameter. Output blocks, here often called Readout blocks, consist of a single dense layer with a certain number of units and an activation/gate function.

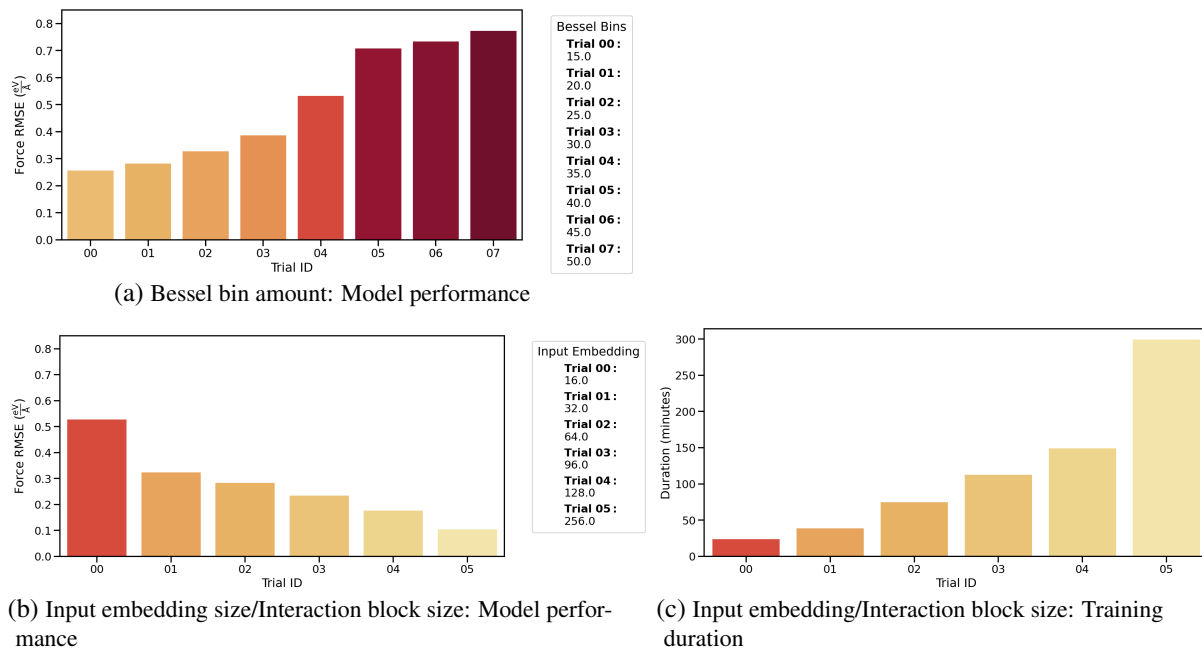


Figure 6.5.: Selected hyperparameter effects on PaiNN model performance. An increased number of Bessel bins is detrimental to model performance. Increasing the input embedding size clearly improves model performance, albeit at the cost of a linear increase in training duration.

MACE models generally achieve a very low error compared to other architectures, but they are also more expensive to train. The most impactful hyperparameter is the correlation order, which significantly improves the performance with higher orders, as shown in Figure 6.6. Higher-order correlations are originally included to reduce the number of message-passing steps required to capture many-body interactions. However, more message passing steps (more interaction blocks/ more depth) did not show a similar improvement compared to the correlation order for this small system (see Appendix D.3). Again, the counterintuitive effect of the number of Bessel bins is observed, with more bins leading to worse performance. A larger number of channels (interaction block size) leads to performance improvements; most other hyperparameters (envelope exponent, radial MLP, output MLP) have only a minor influence on performance.

Due to the generally good performance of the MACE architecture, the typical hyperparameter choices defined in Section 3.4 are used for the training of the Base MACE and QEq-MACE models for the alanine dipeptide dataset instead of the optimal hyperparameters found during the hyperband search.

6.1.2. Dataset Size Dependence

Scope Generating training data is often the most expensive step in developing an MLP. A more data-efficient MLP can achieve good performance with fewer training samples, thereby saving significant computational resources and demonstrating that the MLP can learn the underlying physics of the system

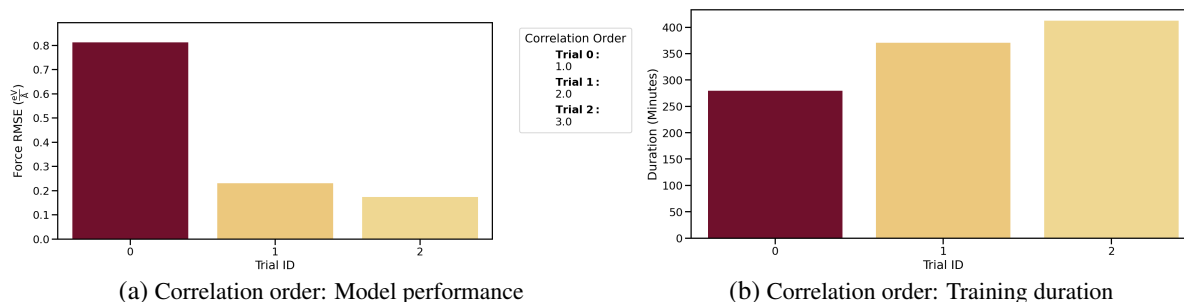


Figure 6.6.: Selected hyperparameter effects on Base MACE model performance. Increasing the correlation order significantly improves model performance, albeit at a substantially longer training time.

more robustly. The dataset-size dependence of different MLP architectures is investigated by training each architecture on subsets of the training data of increasing size and evaluating its performance on the same test set. Three models with the same (unoptimized) hyperparameters are trained independently on a subset of the training set for each training set size to obtain a more robust estimate of the performance and its variability. The results are shown in Figure 6.7.

Results Both generations of the High-Dimensional Neural Network Potential (HDNNP) architecture exhibit a similar dataset-size dependence, with the 4G-HDNNP performing slightly better than the 2G-HDNNP. Still, even with the largest training set size, the HDNNPs fail to reach chemical accuracy of 0.043 eV. The other MLP architectures show a much better performance, with the SchNet and PaiNN models reaching chemical accuracy with around 1000 training samples. They perform as well as MACE for the forces, but MACE’s energy predictions are significantly better. The training procedure for the MACE model involves a second stage with a smaller learning rate, a focus on the energy loss, and Stochastic Weight Averaging (SWA), during which the weights are averaged over multiple epochs to improve generalization. In particular, the focus on energy loss enables the MACE model to converge to a much lower energy error.

Conclusion From these findings, the HDNNPs are expected to perform worse in subsequent MD simulations, as their generalization to unseen configurations is inferior to that of the other MLP architectures.

6.1.3. Training in Vacuum

Scope The key question in this section is how the different MLP architectures perform when trained on the same dataset. Simulation stability and quality are the main criteria for selecting the MLP architecture for the subsequent QM/MM simulations, so particular attention is paid to force predictions, which are more relevant to dynamics than energy predictions. No guarantee of simulation stability can be given

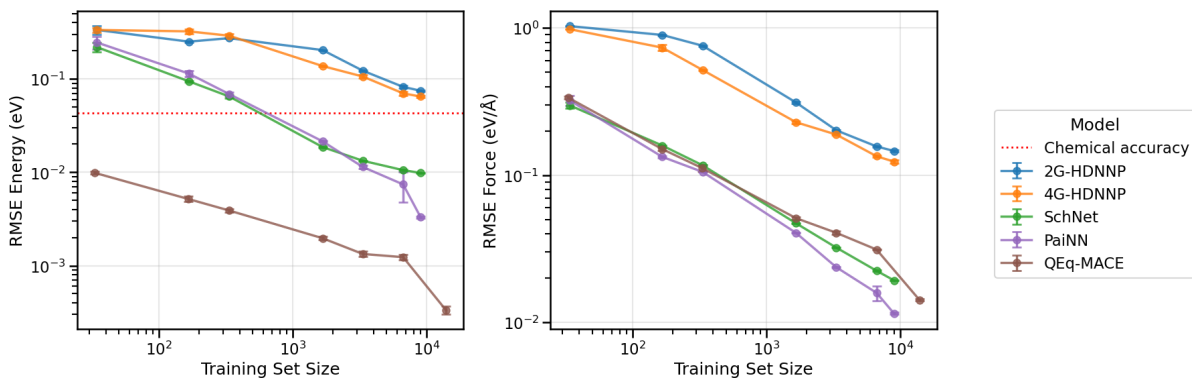


Figure 6.7.: Dataset size dependence of different MLP architectures.

based on test-set performance⁵, but it is a first indication of the model’s ability to generalize to unseen configurations. Three independent training runs are performed for the previously introduced architecture with the identified hyperparameters. Additionally, QEq-MACE is compared with Base MACE as baseline and two more literature MLPs with ML/MM capabilities, Anisotropic Message Passing (AMP)¹⁵⁶, and the MACE-related Field-MACE¹⁵⁹, are added with literature hyperparameters. The results are shown in Table 6.1 and Figure 6.8.

Results Nearly all MLP architectures can achieve chemical accuracy with optimized hyperparameters and the full training set. The performance expectation 2nd Gen. High-Dimensional Neural Network Potential (2G-HDNNP)<4G-HDNNP<SchNet<PaiNN<MACE is confirmed. All MACE models show remarkable performance and reproducibility across the three independent training runs. In contrast, the HDNNPs and SchNet exhibit much greater variability across the three independent training runs. The QEq-MACE model shows a slight improvement over the Base MACE model despite lacking an environment or long-range electrostatic interaction within the training data, for which the charge equilibration scheme is intended. Field-MACE, designed for excited states and ML/MM, performs worse than the baseline Base MACE model. The AMP model performs well but not as well as the MACE models, which may be due to differences in training procedure and the lack of hyperparameter optimization in this system. Only a single model was trained on all the data for the AMP architecture, so that no conclusions can be drawn about the reproducibility of the training.

Conclusion The MACE models and the AMP model are expected to perform well in the subsequent MD simulations. PaiNN appears promising, whereas the HDNNPs and SchNet are expected to perform worse due to larger errors and greater variability.

Table 6.1.: Mean test set performance metrics for different MLP architectures trained on the alanine dipeptide dataset in vacuum across three independent training runs. MACE models and AMP show the best performance.

Method	Energy		Force		Charge	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2	RMSE (e)	R^2
2G-HDNNP	0.034	0.981	0.051	0.998	—	—
4G-HDNNP	0.033	0.983	0.045	0.999	0.043	0.888
SchNet	0.022	0.993	0.032	0.999	—	—
PaiNN	0.010	0.999	0.018	1.000	—	—
Base MACE	0.004	1.000	0.010	0.999	—	—
QEq-MACE	0.003	1.000	0.008	1.000	0.002	1.000
AMP	0.010	0.999	0.011	1.000	—	—
Field-SchNet	0.007	0.999	0.021	1.000	—	—
Field-MACE	0.006	1.000	0.012	1.000	—	—

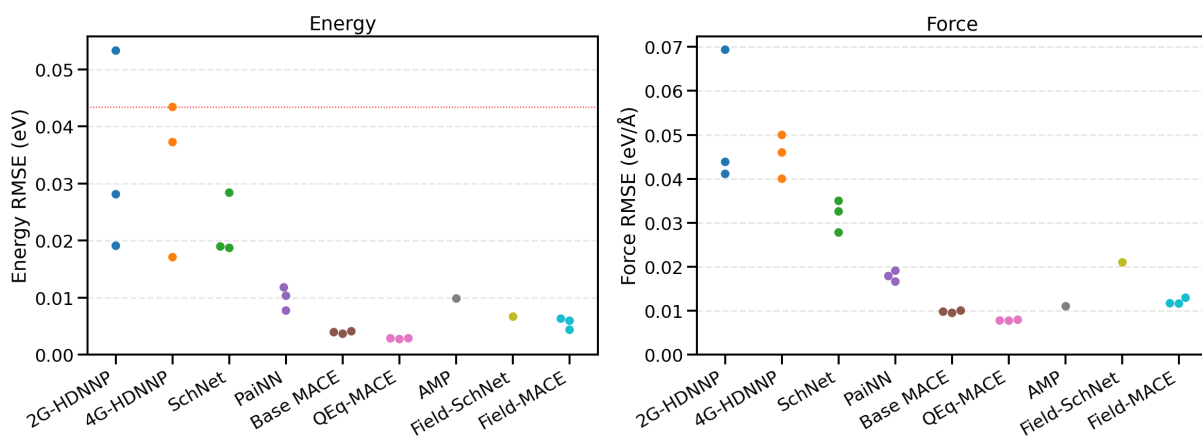


Figure 6.8.: Comparison of test set RMSE for different MLP architectures trained on the alanine dipeptide dataset in vacuum across three independent training runs. MACE models and AMP show the best performance and reproducibility

6.1.4. Metadynamics Simulation in Vacuum

Scope Before the MLP models are applied in the more complex QM/MM simulations, their performance is evaluated in a simpler setup by performing metadynamics simulations of alanine dipeptide in vacuum with the simulation setup described in Section 3.5. As the force prediction performance is not a perfect indicator for the stability of the simulations⁵, the stability of the simulations is evaluated by measuring the completion of the walkers during a 1 ns metadynamics simulation. Additionally, the resulting PMFs are compared to the B3LYP/aug-cc-pVTZ reference to evaluate the quality of the sampling. As an additional metric, the maximum force difference relative to the reference method is evaluated for a subset of the structures encountered during the simulation with QEq-MACE to further validate the PMFs and rank the different MLP architectures.

MLPs are known to be unreliable when extrapolating to configurations outside the training distribution, leading to unphysical forces and simulation instability. To detect this issue, termination criteria are defined for the metadynamics simulations. The simulation is terminated as soon as a bond length or the model ensemble disagreement exceeds a certain threshold. This prevents failed simulations from introducing unphysical structures in the construction of the PMF and eases the comparison of successful simulation runs across different MLP architectures.

Results The AMP, the Base MACE, and the customized QEq-MACE model show a perfect completion of all walkers during the 1 ns metadynamics simulation of alanine dipeptide in vacuum (see Figure 6.9). The other MLP architectures experience instability issues. 4G-HDNNP, SchNet, and PaiNN complete at least a majority of the target simulation time for most walkers, 2G-HDNNP fails to even complete 10% of the simulation. The results for 4G-HDNNP and PaiNN are unexpected. 4G-HDNNP shows a similar force accuracy as 2G-HDNNP during the test set evaluation, but it performs much better in the simulation. PaiNN, on the other hand, shows a very good force accuracy, but it performs worse than expected in the simulation. About a third of the PaiNN walkers complete the full simulation, while many others fail as early as 100 ps.

Enhanced sampling with metadynamics can still be analyzed even if all walkers terminate prematurely. As long as the walkers explore the relevant regions of the free-energy landscape, a reasonable PMF can be constructed and compared with a reference. The resulting PMFs in the ϕ - ψ space for the different MLP architectures are shown in Figure 6.11. The reference PMF at the B3LYP/6-311+G(2d,p) level is shown in Figure 6.11 for comparison¹⁹⁴. Liu, Ensing, and Moore used the quasi-harmonic approximation to correct the relaxed PES to the Free Energy Surface (FES).

Even with only 10% of the target simulation time completed, the 2G-HDNNP model samples at least the minima in the PMF. Higher-energy regions are not sampled because the simulation is terminated

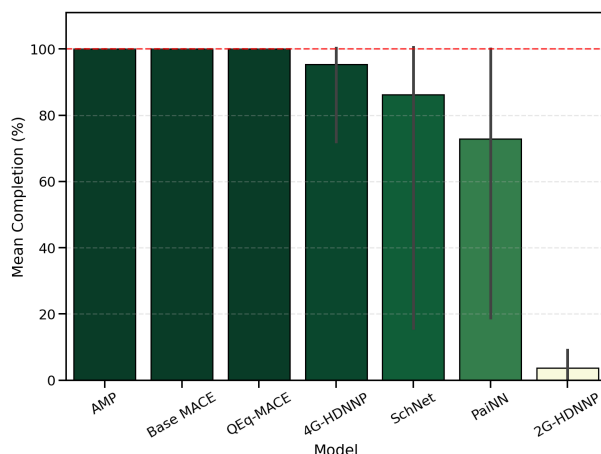


Figure 6.9.: Completion of 16 1 ns walkers during vacuum metadynamics simulation for different MLPs. AMP, QEq-MACE and Base MACE complete all walkers, 2G-HDNNP fails to complete any walker.

early. All other models capture the general shape of the PMF, with the minima and transition regions in the correct locations. The global minimum $C7_{eq}$ is identified as well as the $C7_{ax}$ local minimum. The C_5 minimum is only a clear minimum for the SchNet, Base MACE, QEq-MACE, and AMP models. Transition regions are in the correct locations for all models but the 2G-HDNNP model.

A more detailed analysis of the PMF quality is performed by comparing the maximum force difference to the reference method during the simulation for a subset of 100 structures encountered during the simulation of the Base MACE model, as shown in Figure 6.12. Not only are the MLPs compared, but also DFTB3/3OB as well as PBE0/aug-cc-pVTZ as representatives of semi-empirical and alternative DFT methods to put the MLP performance into perspective. As DFTB3/3OB is not parameterized with the B3LYP functional, it is expected to perform worse than the MLP models, which are trained on B3LYP data. This expectation is met, DFTB shows a median maximum force difference of $1.25 \text{ eV}/\text{\AA}$ to the reference. This deviation is more than twice that of the next-worst method, the PBE0 functional. The difference between all MLPs and the reference B3LYP is smaller than the difference between the two functionals, B3LYP and PBE0 themselves. This is true even for the worst performing MLP model, 2G-HDNNP. Spikes in the deviation of the 2G-HDNNP could be the reason for the early termination of the simulation, although they are still smaller than the deviation of DFTB. The better-performing MLP models exhibit a much smaller deviation from the reference, with the Base MACE, QEq-MACE, and the outstanding AMP models performing best. These results mirror the training results in Figure 6.8 and further deepen the confidence in the quality of the PMF obtained with the MACE models.

An in-depth analysis of the minima and transition states in the PMF of QEq-MACE is performed by comparing these extrema to the reference PMF at the B3LYP/6-311+G(2d,p) level of theory published by Liu, Ensing, and Moore¹⁹⁴. The extrema in the QEq-MACE PMF are assigned to the nearest corresponding extrema in the reference PMF and compared in terms of their relative free energy and their

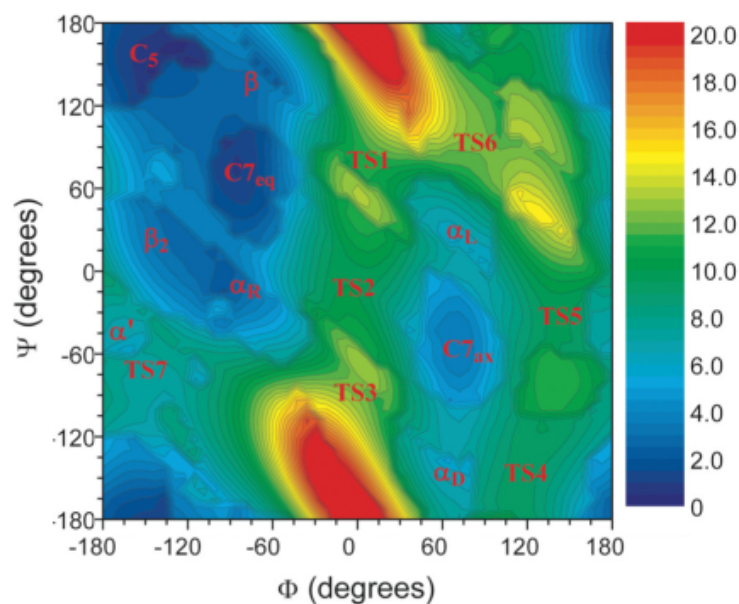


Figure 6.10.: Reference PMF at the B3LYP/6-311+G(2d,p) level of theory in kcal/mol. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the (ϕ - ψ) Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

location in the ϕ - ψ space. Assigning transition states like this from a metadynamics PMF is inherently flawed, as the deposition of the bias potential naturally yields a rougher PMF, and the transition states are not necessarily located at the same positions as in the reference PMF. Still, the relative free energies of the minima and transition states should be comparable. The results are shown in Figure 6.13 and Table 6.2. Both the location of the minima and transition states and their relative free energies show a good agreement between the QEq-MACE PMF and the reference PMF. Locations of the minima and transition states are within 5° of the reference, and the relative free energies are within 1 kcal/mol of the reference. TS2 shows the largest energy deviation of 0.95 kcal/mol.

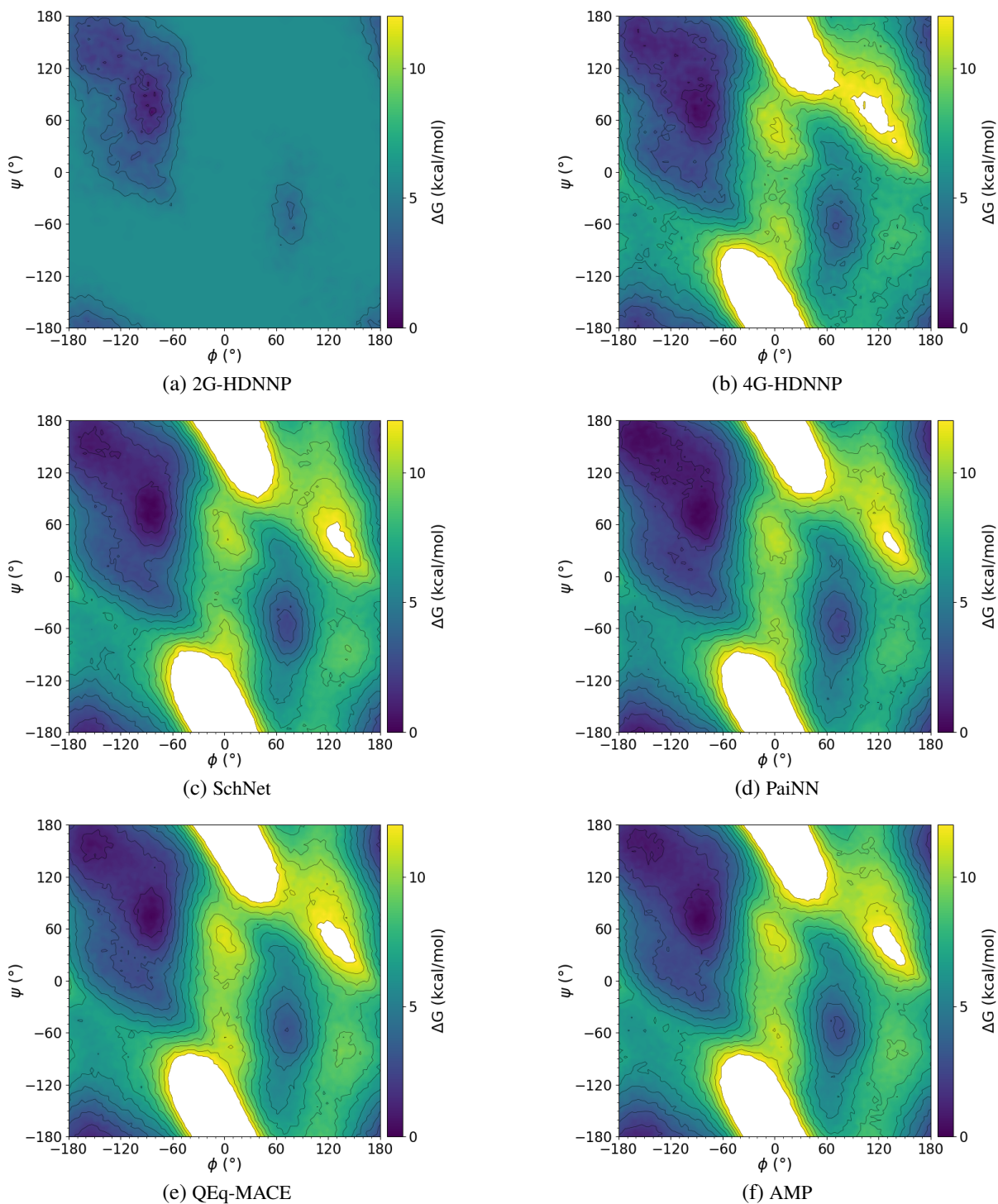


Figure 6.11.: Resulting PMFs after 1.0 ns vacuum metadynamics simulation for different MLPs relative to $C7_{eq}$. Contour lines are drawn at 1.0 kcal/mol intervals. Minima and transition regions are broadly in the correct locations for all models but the 2G-HDNNP.

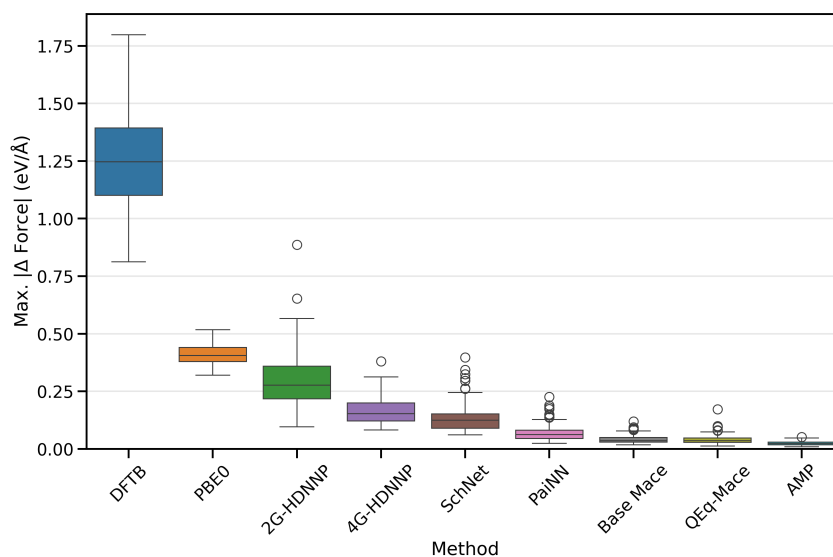


Figure 6.12.: Maximum absolute force difference to the B3LYP/aug-cc-pVTZ reference during 1 ns vacuum simulation of alanine dipeptide for different methods. PBE0 and DFTB included for perspective. Base MACE, QEq-MACE, and AMP show the best performance.

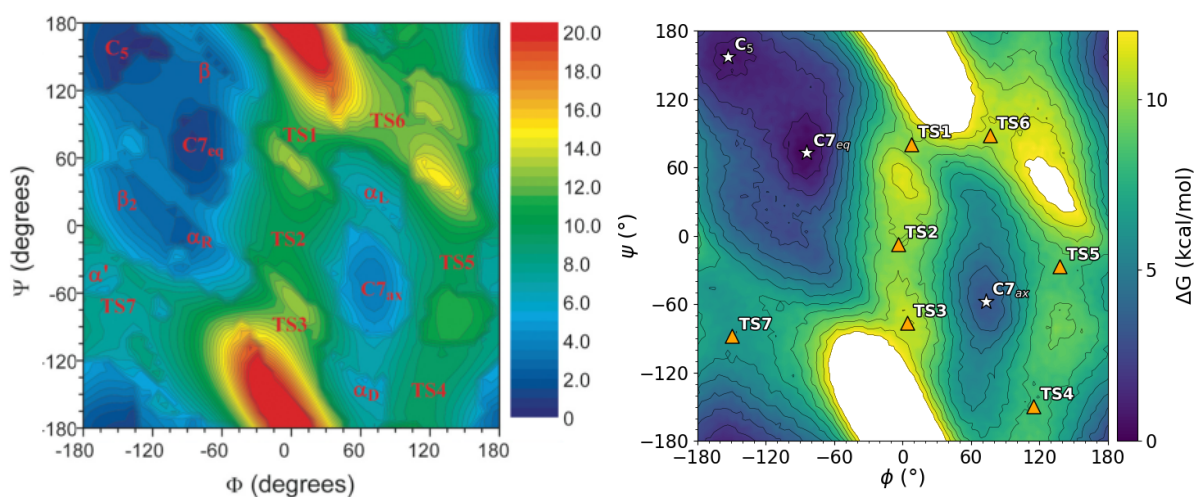


Figure 6.13.: Analysis of the minima and transition states in the reference PMF at the B3LYP/6-311+G(2d,p) level¹⁹⁴ (left) and the QEq-MACE PMF (right). The locations of the corresponding minima and transition states, and their relative free energies, are in good agreement. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the $(\phi-\psi)$ Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

Table 6.2.: Comparison of free energy extrema between the reference at the B3LYP/6-311+G(2d,p) level¹⁹⁴ and the observed extrema from the QEq-MACE PMF analysis. The location of corresponding extrema and their relative free energies are in good agreement. Angles are in degrees, free energies are in kcal/mol. Energies are referenced to the C7_{eq} minimum. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the (ϕ - ψ) Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

Label	ϕ_{ref}	ϕ_{obs}	ψ_{ref}	ψ_{obs}	ΔG_{ref}	ΔG_{obs}	$\Delta(\Delta G)$
C7 _{eq}	-83.10	-84.26	72.60	72.77	0.00	0.00	0.00
C ₅	-158.40	-153.19	164.60	157.02	0.06	0.46	0.40
C7 _{ax}	73.60	72.77	-57.70	-57.45	2.63	2.92	0.29
TS1	5.60	7.66	81.40	80.43	10.31	9.80	-0.51
TS2	-1.40	-3.83	-8.90	-7.66	8.94	9.89	0.95
TS3	2.80	3.83	-77.30	-76.60	10.39	10.96	0.57
TS4	112.80	114.89	-146.70	-149.36	8.24	8.14	-0.10
TS5	135.90	137.87	-26.20	-26.81	8.68	8.48	-0.20
TS6	79.00	76.60	86.40	88.09	11.04	10.56	-0.48
TS7	-149.80	-149.36	-87.30	-88.09	6.27	6.10	-0.17

6.2. Training Thiol Disulfide Exchange Reaction in Vacuum

6.2.1. Hyperparameter Optimization

The general trends and preferences in the hyperparameters of the different MLP architectures mirror those observed for the alanine dipeptide dataset. Long-range η arrays for HDNNP, a large number of interaction block units for SchNet and PaiNN, and fewer Bessel basis functions for the PaiNN and MACE models are the dominant features in the Top 10 hyperparameter sets for the thiol disulfide exchange reaction dataset during the hyperband search. For in-depth information on the hyperparameter search spaces and results, please refer to Appendix D.4. In contrast to the alanine dipeptide system, the training of Base MACE and QEq-MACE models for this system is performed using the optimal hyperparameters found during the hyperband search instead of the typical hyperparameter choices defined in Section 3.4. This decision is based on initial instability issues observed during simulations with the thiol disulfide exchange reaction dataset. Stable simulations are achieved with the optimal hyperparameters and a corrected simulation setup.

6.2.2. Training in Vacuum

Scope In this section, the performance of different MLP architectures trained on the thiol disulfide exchange reaction dataset in vacuum is evaluated. The procedure is similar to that used for training on the alanine dipeptide dataset. Three independent training runs are performed for each MLP architecture

to assess training reproducibility and the stability of the results. The results are compared with the performance of the same MLP architectures trained on the alanine dipeptide dataset in vacuum to assess differences between the two datasets.

Results Trends in the performance of the different MLP architectures are similar to those observed for the alanine dipeptide dataset. The force errors decline in the expected order 4G-HDNNP > SchNet > PaiNN > MACE. The 2G-HDNNP surpasses the expectations by outperforming 4G-HDNNP and SchNet. One model for each architecture performs significantly worse than the other two models. The fact that this issue is consistent across all architectures and that the same test set has been used for all models indicates that the root cause lies in the training set, which appears to lack certain important configurations. The overall performance of the models is worse than for the alanine dipeptide dataset. The exchange reaction contains only 15 atoms, compared to 22 for the alanine dipeptide, and the dataset is almost twice as large. Still, as it is a reaction, it contains a wider range of configurations and properties, making it a more challenging dataset for MLP models. The best-performing models are PaiNN, Base MACE, QEq-MACE, and AMP, which achieve chemical accuracy (energy predictions within 1 kcal/mol). In contrast to the alanine dipeptide dataset, no significant benefit of the second training state using SWA is observed for the MACE models.

Table 6.3.: Mean test set performance metrics for different MLP architectures trained on the thiol disulfide exchange reaction in vacuum across three independent training runs. PaiNN, Base MACE, QEq-MACE, and AMP manage to beat the chemical accuracy threshold. The overall performance is worse than for the alanine dipeptide dataset.

Method	Energy		Force		Charge	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2	RMSE (e)	R^2
2G-HDNNP	0.061	0.993	0.048	0.998	—	—
4G-HDNNP	0.080	0.990	0.064	0.997	0.030	0.968
SchNet	0.064	0.993	0.049	0.998	—	—
PaiNN	0.033	0.998	0.039	0.999	—	—
Base MACE	0.048	0.995	0.026	0.999	—	—
QEq-MACE	0.039	0.997	0.024	1.000	0.021	0.985
AMP	0.027	0.999	0.031	0.999	—	—

6.2.3. Metadynamics Simulation in Vacuum

Scope In this section, the performance of different MLP architectures trained on the thiol disulfide exchange reaction dataset in vacuum is evaluated by performing metadynamics simulations of the exchange reaction using the simulation setup described in Section 3.5. The stability of the simulations is evaluated by measuring the walkers’ completion during a 0.75 ns metadynamics simulation. Additionally, the resulting PMFs are compared to the B3LYP/aug-cc-pVTZ reference to evaluate the quality of the

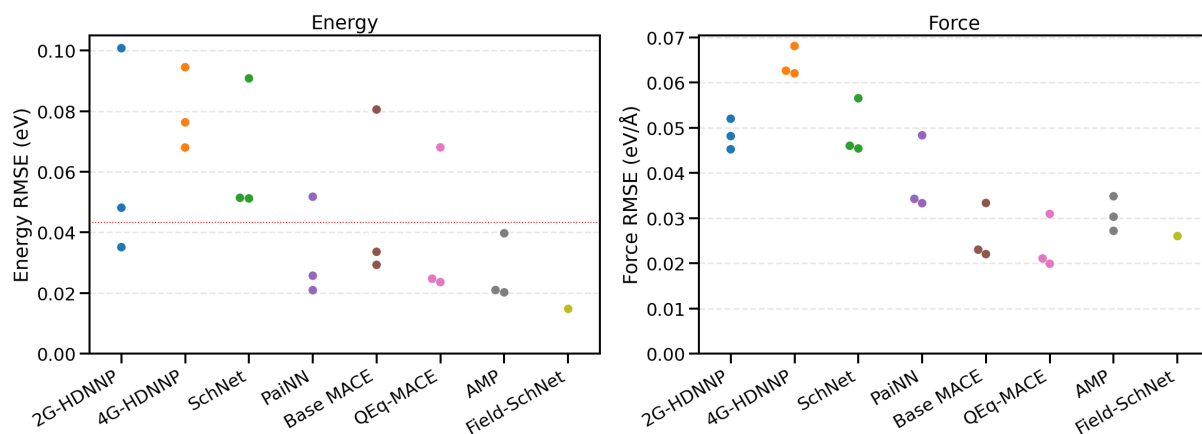


Figure 6.14.: Comparison of test set RMSE for different MLP architectures trained on the thiol disulfide exchange reaction dataset across three independent training runs. PaiNN, Base MACE, QEq-MACE, and AMP manage to beat the chemical accuracy threshold. The overall performance is worse than for the alanine dipeptide dataset.

sampling. As an additional metric, the maximum force difference to the reference method is evaluated for a subset of the structures encountered during the simulation to further validate the PMFs and rank the different MLP architectures.

Results Only the QEq-MACE model manages to complete all walkers during the 0.75 ns metadynamics simulation of the reaction in vacuum, as shown in Figure 6.15. SchNet, PaiNN, and AMP also show good performance, with all walkers completing the full simulation or failing only shortly before the target simulation time. The simulations with 4G-HDNNP rarely complete the full duration, but sample enough of the free energy landscape to construct a reasonable PMF. The Base MACE surprisingly fails to complete any walker, which is unexpected given the model’s good performance during training. A closer inspection of the simulation trajectories reveals that the Base MACE model fails when a methyl group is deprotonated. The methyl group will take a trigonal planar configuration during the deprotonation, which is not well represented in the training set. Deprotonation also occurred in the reference DFTB vacuum simulations, but was not a major issue with the other MLP models. A few active learning cycles and constraints to prevent the deprotonation could solve this issue. As in the alanine dipeptide simulations, the 2G-HDNNP model fails to complete any walker. The unexpectedly good performance in training the 2G-HDNNP model does not translate into good performance in the simulation, which again indicates that force accuracy is not a perfect indicator of simulation stability⁵.

No high-level reference PMF is available for this system, so a Special Reaction Parameterization (SRP) for this system parameterized DFTB3/3OB_{Mod S-S} to the B3LYP functional serves as the reference for the PMF comparison (see Figure 6.16). The simulation subset will be the best evidence for the quality of the PMF obtained with the different MLP architectures. The PMF features three minima, which correspond to the reactant, product, and the linear transition state of the reaction, which instead happens

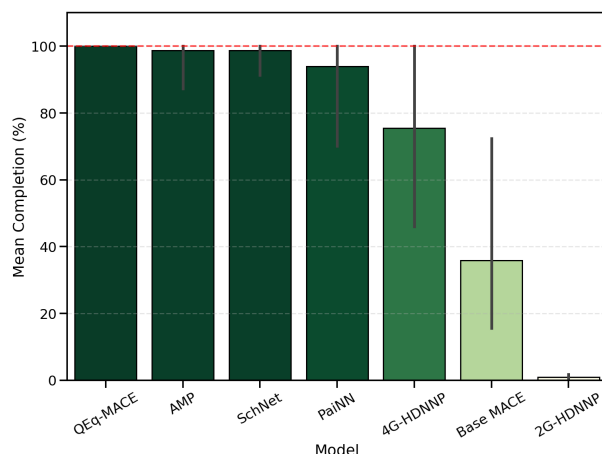


Figure 6.15.: Completion of 16 0.75 ns walkers during vacuum metadynamics simulation for different MLPs. QEq-MACE complete all walkers; SchNet, PaiNN, and AMP complete most walkers. Base MACE and 2G-HDNNP fail to complete any walker.

to be the global minimum in the gas phase. A barrier of about 6 kcal/mol separates the linear transition state from the reactant and product minima.

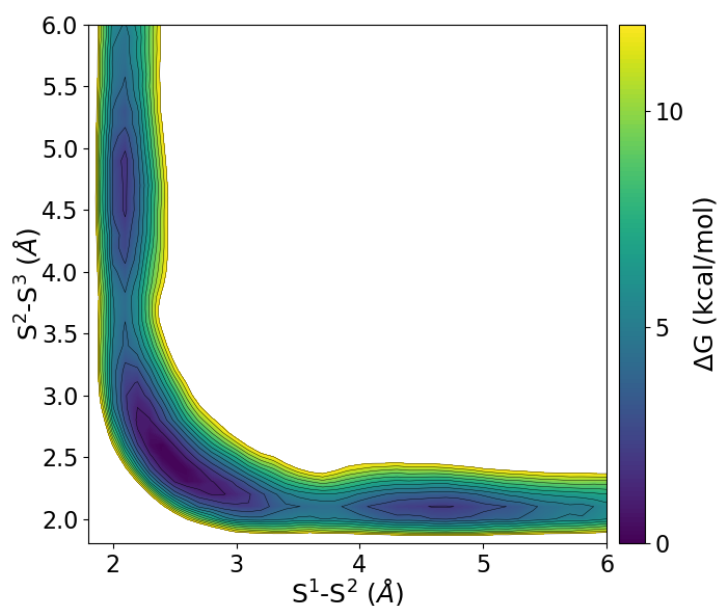


Figure 6.16.: Reference PMF obtained with DFTB3/3OB_{Mod S-S} for the thiol disulfide exchange reaction in vacuum. Contour lines are drawn at 1.0 kcal/mol intervals.

As shown in Figure 6.17, even the 2G-HDNNP model captures the basic features of the reference PMF with its minimal stable simulation time. The convergence is poor, and the transition regions are not sampled, but the minima are in the correct locations. All other models capture the main features of the reference PMF. The minima and transition regions are in the correct locations, and the relative free energies of the minima are in reasonable agreement with the reference. A subset of the simulation structures encountered during the live simulation of the QEq-MACE model is evaluated for the maximum

force difference to the reference method to evaluate the quality of the PMF and to rank the different MLP architectures.

Figure 6.18 shows the ranking of the models plus the alternative DFT functional PBE0/aug-cc-pVTZ and DFTB3/3OB_{Mod S-S} as an orientation of typical deviations to the reference method B3LYP/aug-cc-pVTZ. Here, the QEq-MACE produces the lowest median around 0.1 eV/Å for the maximal force difference. DFTB has lower deviations to B3LYP for this system in comparison to the alanine dipeptide system, dropping from a median of 1.25 eV/Å to 0.5 eV/Å. An SRP fitted to the B3LYP functional is used, which deviates less from B3LYP than the DFTB3/3OB parameterization fitted to the PBE functional. The Base MACE and 2G-HDNNP models are unremarkable in this ranking despite failing to complete any walker during the simulation. Including the critical deprotonation configurations in this evaluation could change the ranking of these models.

Conclusion Vacuum MLPs

MACE and AMP models have proven to be the best-performing MLP architectures for both systems in vacuum, with good training performance and stable simulations. Comparisons with reference methods and analyses of the simulation trajectories provide high confidence in the quality of the obtained PMFs with these models. The other MLP architectures exhibit worse performance during training and less stable simulations, yet still capture the main features of the reference PMF. Field-MACE fails to achieve Base MACE’s performance during training, while the inclusion of the charge equilibration scheme in QEq-MACE enables it to outperform its forerunner slightly. The success of the charge equilibration scheme in QEq-MACE is especially interesting, as the prediction of the atomic charges enables the use of the model in a QM/MM setup, which is the main interest of this thesis.

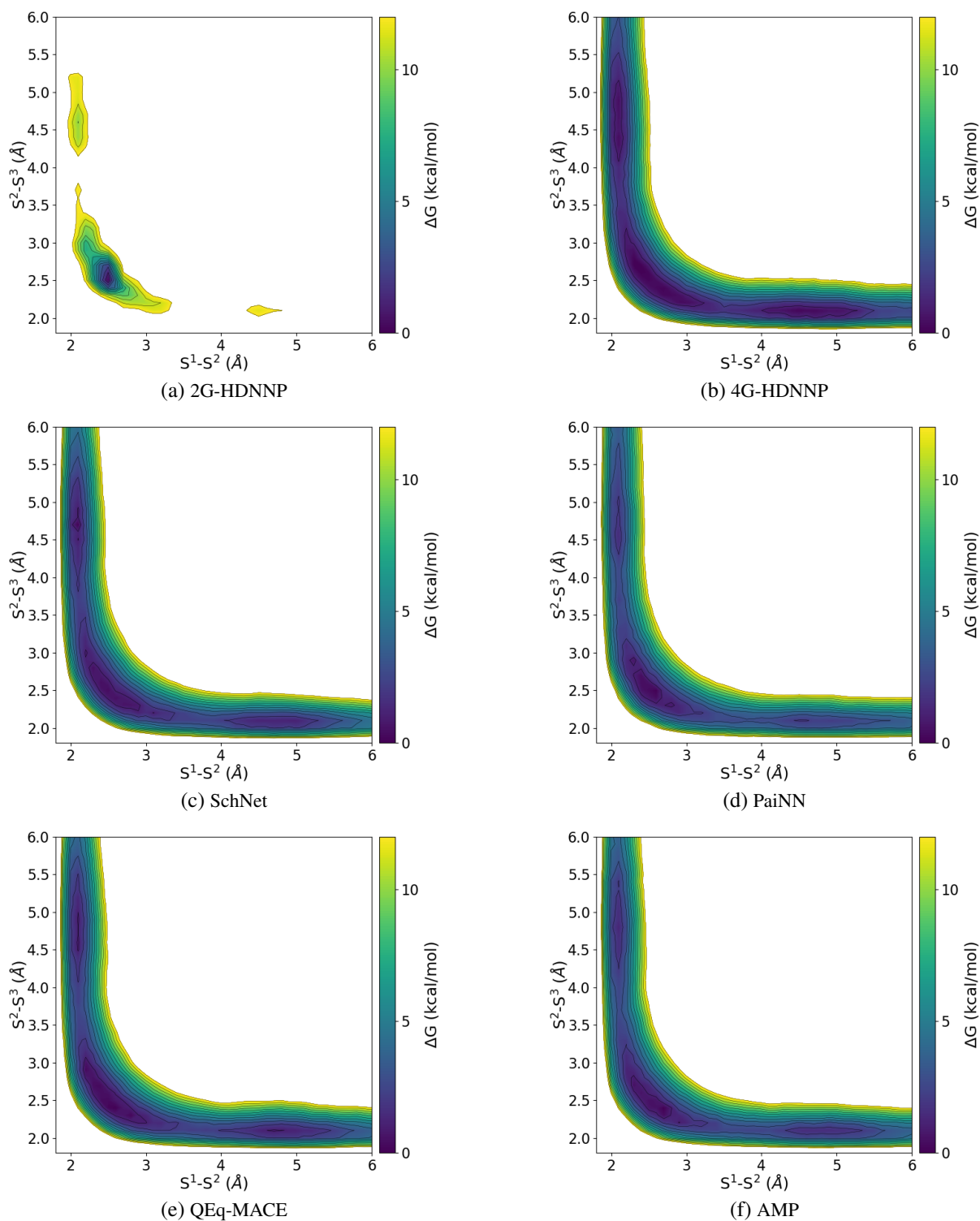


Figure 6.17.: Resulting PMFs after 0.75 ns vacuum metadynamics simulation for different MLPs. Contour lines are drawn at 1.0 kcal/mol intervals. All models except the 2G-HDNNP capture the main features of the reference PMF.

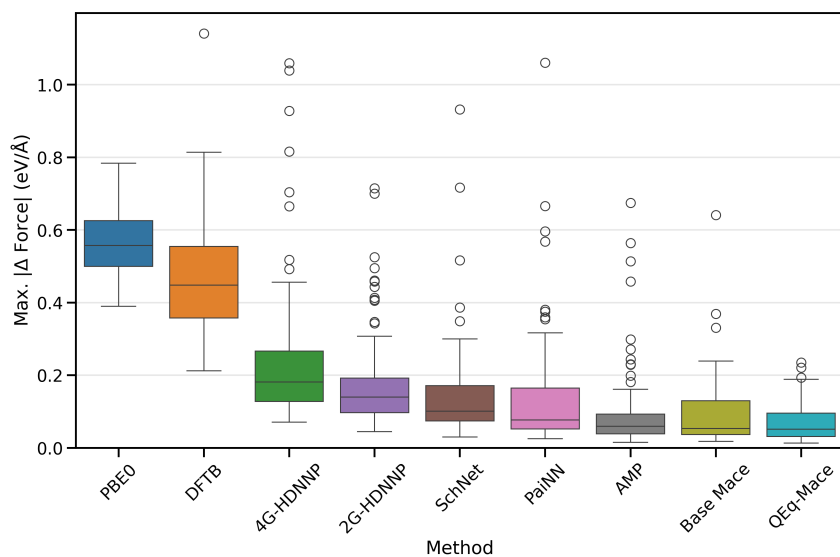


Figure 6.18.: Maximum absolute force difference to the B3LYP/aug-cc-pVTZ reference during 0.75 ns vacuum simulation of the thiol disulfide exchange reaction for different methods.

6.3. Training Alanine Dipeptide in Water

Introduction Most MLPs are developed solely for vacuum applications, but more and more research is being done to extend their applicability to condensed-phase systems. Different approaches have been proposed to achieve this.^{146,162}

1. Full QM treatment: Both solute and solvent are treated at the same QM level of theory. This is the most straightforward approach, but generating training data can be computationally expensive, especially for larger solvent systems.^{47,147,149}
2. Δ -learning⁸: An MLP is trained to learn the difference between a cheaper method (e.g., DFTB) and a more accurate method (e.g., DFT) in vacuum, and then applied to the solvent system. This is based on the assumption that the solvent effect is captured well by the cheaper method, and the MLP only needs to learn the vacuum correction.^{118,148}
3. Mechanical embedding: The MLP interacts with the solvent through classical electrostatic interactions with fixed partial charges. Polarization effects are not captured via this approach.^{150–152}
4. Polarizable-Corrected Mechanical Embedding (PCME): Just like the mechanical embedding, the MLP is trained on vacuum data, but a *post hoc* correction is added to approximate environment-induced polarization effects on the solute. This can be done largely physically motivated, which aids transferability.^{153,154}
5. Environment-Integrated Embedding (EIE): The environment is directly integrated into the MLP training by embedding the solvent molecules in the input representation resembling standard QM/MM. This allows the MLP to learn the solvent effect directly from the data.^{155,156}

This thesis focuses on the last approach, all applied ML/MM-capable MLP architectures (4G-HDNNP, QEq-MACE, AMP, Field-SchNet, Field-MACE) fall into the EIE category, see Section 2.5.2 for an introduction to the theory behind these models. Compared to *ab initio* QM/MM, the ML/MM approach enables many multiscale simulations at much lower computational cost. Still, it has a direct competitor in semi-empirical methods such as DFTB. Therefore, a key question is whether MLPs can achieve better accuracy than DFTB at a similar computational cost.

Scope To answer this question, we apply the ML/MM approach to the two systems: alanine dipeptide and a thiol disulfide exchange reaction. For both systems, we first compare the properties of the solute in vacuum and in solvent to understand the effect of the environment. Then, we train different MLP architectures on the same dataset generated in solvent and compare their performance on a test set. The comparison to their performances in vacuum reveals the additional error introduced by the environment.

Finally, we run metadynamics simulations in solvent with the trained MLPs and compare the resulting PMFs to a reference.

6.3.1. Environment Comparison

Solvation has a significant effect on the properties of the solute, which can be observed by comparing the energies and forces. At the same time, the ESP at each atom's position can be used to quantify the strength of the solvent effect. The alanine dipeptide system is a good example of this, as it exhibits a conformational change in the presence of solvent. Figure 6.19 shows the comparison of differences in energy and forces between solvent and vacuum environments, as well as the ESP values for the different sampling methods investigated in Chapter 4. The solvation stabilizes the molecules mostly between 2.0 eV and 3.0 eV and the force magnitude is adjusted in an IQR between $-0.1 \text{ eV/\AA}$ and $0.05 \text{ eV/\AA}$. These are the ranges of the additional error that the MLP has to learn when trained in solvent instead of vacuum by incorporating the environment information. The ESP, the main focus for incorporation of environment information in this thesis, reveals the different simulation setups as the 300 K and 500 K simulations show ESP IQRs between -1.0 eV/e and 1.0 eV/e , while the simulation with modified force field parameters shows a much higher ESP with an IQR between -2.0 eV/e and 3.0 eV/e . Many structures in the 300 K and 500 K simulations were near the edge of isolated water clusters in vacuum, experiencing only half of the solvent effect.

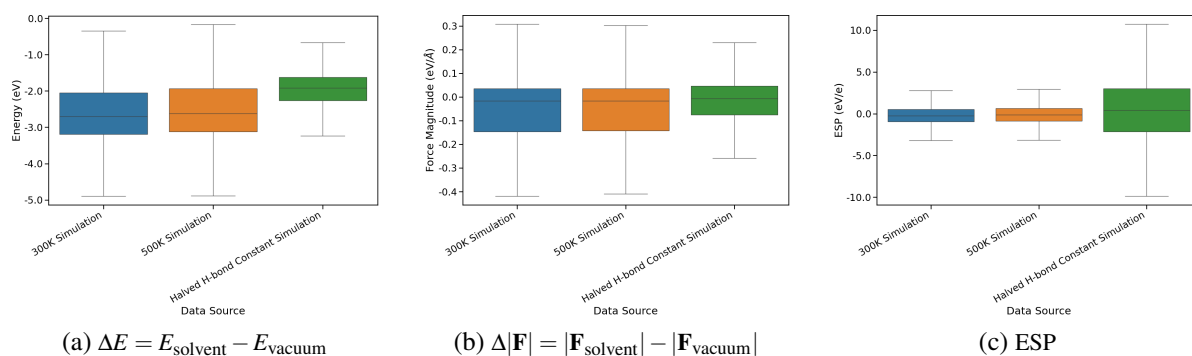


Figure 6.19.: Comparison of environment effect for alanine dipeptide system regarding (a) energy, (b) forces, and (c) ESP across different sampling setups. In solution, energies vary an additional -3.0 eV to -2.0 eV and forces $-0.1 \text{ eV/\AA}$ to $0.05 \text{ eV/\AA}$ compared to vacuum. MLPs need to predict these additional variations by incorporating environmental information.

Different configurations experience a different degree of stabilization by the solvent, which leads to a significant change in the PMF as shown in Figure 6.20 at the B3LYP/6-311+G(2d,p) level of theory using Polarizable Continuum Model (PCM) for solvation and the quasi-harmonic approximation for free energy correction¹⁹⁴. The vacuum minima $C7_{eq}$ and $C7_{ax}$ are seven-membered rings forming an intramolecular hydrogen bond. The $C5$ minimum (also referred to as β in other literature) is a five-membered ring with a different intramolecular hydrogen bond.¹⁹⁸ As they already have a hydrogen bond

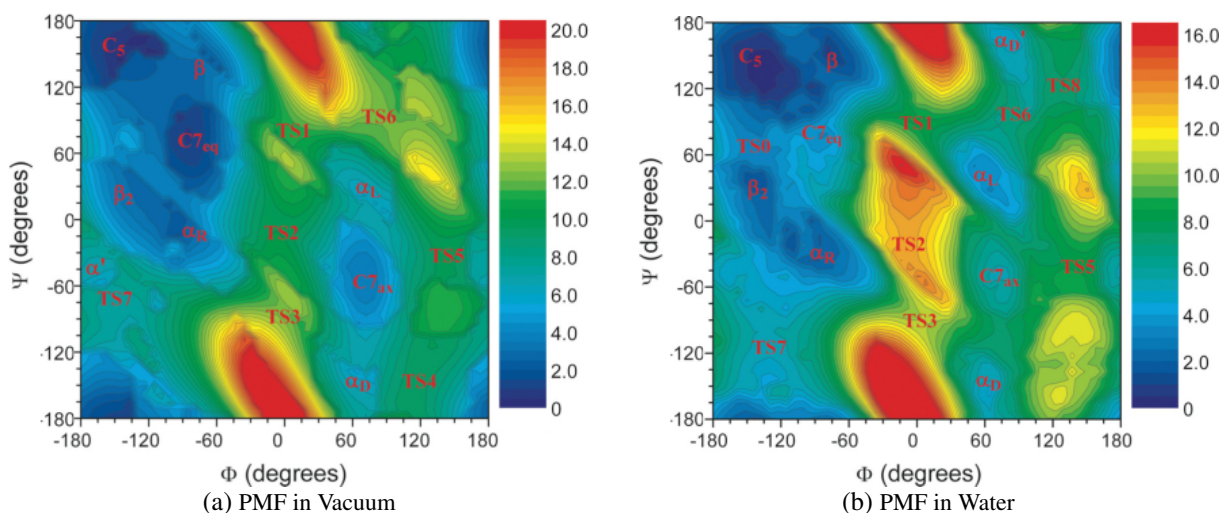


Figure 6.20.: Comparison of PMFs for alanine dipeptide system in (a) vacuum and (b) water environment at the B3LYP/6-311+G(2d,p) level of theory using PCM for solvation. The presence of solvent significantly affects preferred conformations. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the $(\phi-\psi)$ Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

in vacuum, they are not stabilized as much by the solvent as non-hydrogen bonded configurations, such as α_R , α_L , and β (also referred to as Polyproline II (PPII)). The solvent stabilizes these configurations and becomes more favorable than the $C7_{eq}$ and $C7_{ax}$ minima with a more accessible low-energy region around the β minimum. This transformation of the PMF landscape due to solvent effects is a key aspect that MLPs must capture when trained in a solvent, highlighting the importance of incorporating environmental information into the model.

6.3.2. Training in Water

Scope In this section, we train different MLP architectures in the same fashion as in vacuum, but now using the dataset generated in water. This approach is only sensible for MLP architectures that can incorporate environment information. Specifically, we train the 4G-HDNNP, QEq-MACE, Field-SchNet, Field-MACE, and AMP architectures, all of which fall into the EIE category. 4G-HDNNP and QEq-MACE employ a charge-equilibration with a ESP contribution for predicting atomic charges for the electrostatic interactions. Field-SchNet is based on atomic dipole predictions and their interaction with themselves and the surrounding (electric) field caused by the environment.¹⁵⁵ Field-MACE is based on the MACE architecture. It incorporates a multipole expansion with environmental features into the message construction of MACE, without explicitly calculating their physical contributions to energy and force.¹⁵⁹ The multipole expansion is explicitly part of the total energy prediction in AMP and the environment features contributing via an additional weighted term in the multipole expansion.¹⁵⁶ The Base

MACE model, without any environmental awareness, is included as a baseline to evaluate the importance of incorporating environmental information during solvent training. The performance of these models on the test set is compared to their performance when trained in vacuum to evaluate the additional error introduced by the environment. Identical hyperparameters are used for training in vacuum and solvent to ensure a fair comparison.

Results The Base MACE baseline shows an error in a similar range as the energy and force variation introduced by the solvent, which indicates that it is not able to capture the solvent effect at all after being nearly perfect in vacuum. Field-SchNet performs just as poorly, seemingly not benefiting from the environmental information at all, despite receiving the electric field at the atomic positions. Field-MACE shows some effect, but still has a manyfold increase in error compared to vacuum. Even the 4G-HDNNP performs in water better than Field-Schnet and Field-MACE. QEq-MACE improves a lot over the baseline, but still has a significant increase in error compared to vacuum. It fails to achieve chemical accuracy and increases its force error by a factor of 7 compared to vacuum. AMP has a disadvantage in the vacuum-solvent comparison, as only one model was trained on the whole dataset in vacuum, but three independent models were trained on dataset splits in solvent. Still, it performs best in a solvent. It misses chemical accuracy for energy, but only doubles the force error despite seeing only a third of the training data compared to its vacuum counterpart.

Conclusion In Chapter 5 we estimated the error introduced by the discretization of the QM charge density into atomic point charges for the electrostatic interactions to be in the range of 0.02 eV for the electrostatic energy for this system. Both the 4G-HDNNP and QEq-MACE models base their electrostatic interactions on atomic point charges, which could partly explain their increased error in solvent compared to vacuum, but even for QEq-MACE, the error exceeds this estimate by a factor of 2. The success of AMP can be attributed to its explicit inclusion of higher-order multipole interactions, which more accurately capture solvent effects than point charges alone.

6.3.3. Metadynamics Simulation in Water

Scope To evaluate the performance of the trained MLPs in a dynamic simulation, we run multiple walker metadynamics simulations for 1.0 ns with 16 walkers for each model. The simulation completion is compared to evaluate the stability of the simulations. As in the vacuum simulations, the simulation is terminated as soon as a bond length or the model ensemble disagreement exceeds a certain threshold due to the risk of unphysical behavior beyond the interpolation regime of the MLP. The maximum force difference to the DFT reference during the simulation is also compared to evaluate the correct trajectory of the active simulation. The resulting PMFs are compared to the DFT reference from Liu, Ensing,

Table 6.4.: Test set performance metrics for different MLP architectures trained on the alanine dipeptide dataset in water. Values represent mean performance across up to three independent training runs. Field-SchNet and Field-MACE fail to improve on the baseline Base MACE, while 4G-HDNNP and QEq-MACE show some improvement. AMP impresses due to low performance deterioration compared to vacuum.

Method	Energy		Force		Charge	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2	RMSE (e)	R^2
4G-HDNNP	0.186	0.944	0.095	0.991	0.045	0.915
Base MACE	0.730	0.192	0.186	0.969	—	—
QEq-MACE	0.094	0.987	0.056	0.997	0.054	0.878
AMP	0.082	0.992	0.025	0.999	—	—
Field-SchNet	0.712	0.249	0.198	0.966	—	—
Field-MACE	0.276	0.884	0.164	0.975	—	—

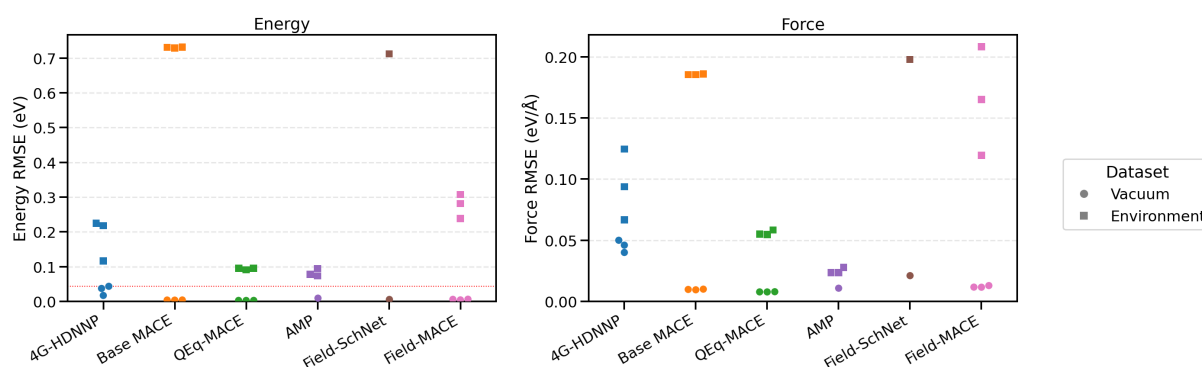


Figure 6.21.: Comparison of test set RMSE for different MLP architectures trained on the same alanine dipeptide dataset in vacuum and water. Field-SchNet and Field-MACE fail to improve on the baseline Base MACE, while 4G-HDNNP and QEq-MACE show some improvement. AMP impresses due to low performance deterioration compared to vacuum.

and Moore¹⁹⁴ to evaluate the accuracy of the simulations. A series of calculations with water bubbles of different sizes around the solute is performed to evaluate the transferability of the trained MLPs to slightly shifted environments.

Results QEq-MACE completes all 16 walkers for the full 1.0 ns simulation, while other models have some premature terminations due to unphysical behavior, as shown in Figure 6.22a. Base MACE also finishes all walkers despite getting no environment information, while 4G-HDNNP performs as poorly as 2G-HDNNP in vacuum. As a result, the PMF from 4G-HDNNP is not converged and only displays a basic topology (see Figure 6.22). Base MACE can capture the general topology even in the absence of electrostatic interactions. With the explicit electrostatic interaction in QEq-MACE, even the shallow $C7_{ax}$ and $C7_{eq}$ minima are observed, which DFTB also misses. AMP also misses some of the shallow minima; convergence in some regions seems incomplete.

Recalculating a subset of 100 structures from the QEq-MACE simulation with the reference method B3LYP/aug-cc-pVTZ and investigating the maximum force difference to it reveals that forces in water deviate for most methods significantly more from the reference than in vacuum, as shown in Figure 6.23. The DFTB3/3OB error increases only marginally, but DFTB remains the worst-performing method because it is parameterized with a different functional. Base MACE, formerly a near-perfect model in vacuum, now exhibits a manyfold increase in deviation and becomes the second-worst-performing method. 4G-HDNNP and QEq-MACE now show a similar error, the simulation-breaking configurations for 4G-HDNNP are not included in the analysis. Typically, hydrogen bonds are the first to break in the simulation, as they oscillate fastest due to the mass ratio. Despite this trend, only C-N bonds break during the 4G-HDNNP simulation, as sampling with weakened hydrogen bonds provided sufficient training data for the correct description of elongated and shortened hydrogen bonds. The only MLP barely affected by the performance drop-off is AMP. As seen in the training performance, it is the only model that maintains a similar level of accuracy in solvent as in vacuum, making it the best-performing model in this selection.

The relative energies and locations of the minima and transition states in the QEq-MACE were analyzed and compared to the reference PMF at the B3LYP/6-311+G(2d,p) level in PCM water¹⁹⁴. The comparison is flawed by the differing basis sets and solvent treatments, resulting in more pronounced free-energy differences. However, it still provides a qualitative picture of the performance of the QEq-MACE simulation. The general topology of the PMF is captured well. The order of relative free energies is in agreement, although the β could not confidently be assigned in the C_5 valley. The relative free energies of the minima are all underestimated, with the largest underestimation of -1.27 kcal/mol for α_R . The location of $C7_{ax}$ and α_L minima are in good agreement, while the C_5 and the other α minima are shifted by up to 27° with α_R being the most affected. The α'_D minimum is not observed in the QEq-MACE simulation; only a shoulder in the PMF is visible. The transition states are again chosen as the clos-

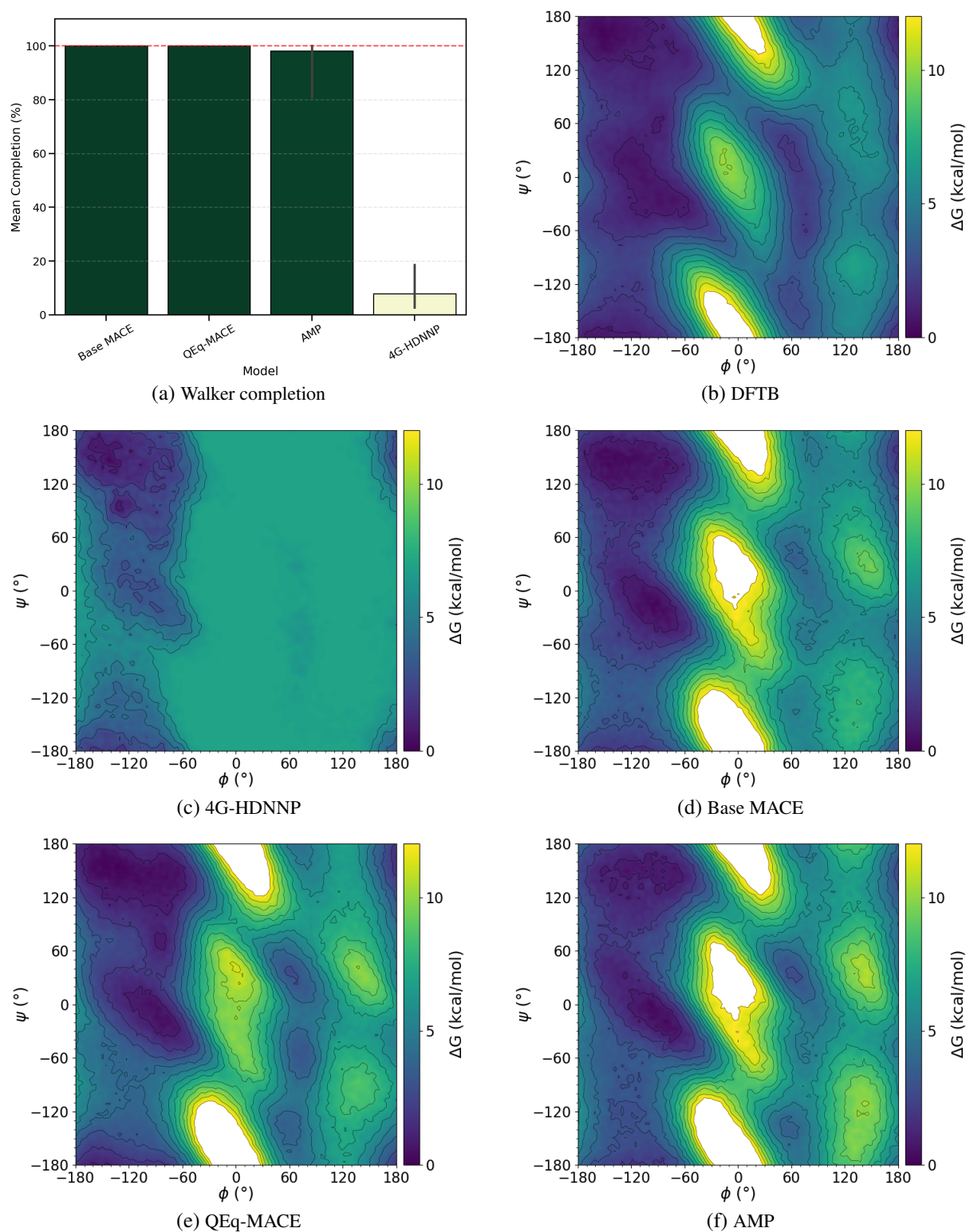


Figure 6.22.: Dipeptide water simulation PMF comparison

Completion of 16 1.0 ns walker metadynamics simulation in water and comparison of resulting PMFs for different MLPs. Contour lines are drawn at 1.0 kcal/mol intervals. No termination occurred for the MACE models. QEq-MACE captures more details of the PMF than the other models.

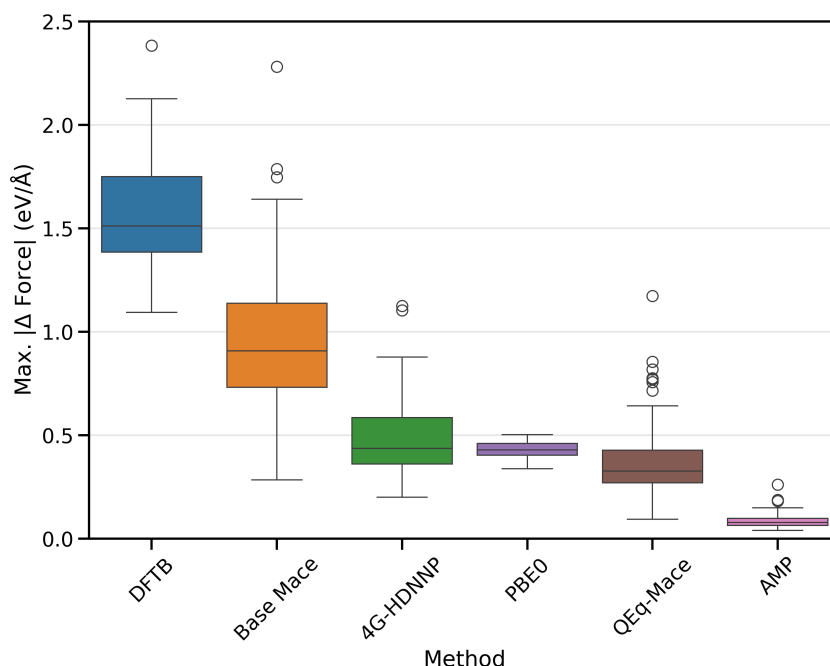


Figure 6.23.: Maximum absolute force difference to the B3LYP/aug-cc-pVTZ reference during 1 ns water simulation for different methods.

est to the reference transition states on the metadynamics PMF. In all cases, close transition states are observed, but the relative free energies are all underestimated by the QEq-MACE simulation, with the largest underestimation of 2.78 kcal/mol for TS7. Transition states are generally more affected than minima.

The analysis so far has focused on the solute properties and the resulting PMF, but the solvent structure around the solute is also an important aspect to evaluate. The Radial Distribution Function (RDF) of water oxygen atoms around the alanine oxygen atom and nitrogen atom during the simulations with different methods is compared in Figure 6.25. DFTB serves as a reference for the expected solvation structure, as it is routinely used for simulations in solvent. DFTB shows a structured solvation shell with a clear density peak around 2.8 Å for the oxygen atom and around 3.0 Å for the nitrogen atom. Only AMP can capture a similar solvation structure, while the other models show a much less structured solvation shell with a later density peak, especially for the nitrogen atom. Base MACE is expected to completely lack electrostatic interactions, leading to an unstructured solvation shell with a relatively late density peak, if any. Environment-aware models are expected to capture solvation structure to some extent. For oxygen, 4G-HDNNP and QEq-MACE show improved agreement with the reference, whereas for nitrogen, the solvation structure remains essentially unchanged from the baseline. The solvation structure is important for the accurate description of the solvent effect on the solute, and failure to capture it can lead to errors in the predicted solute properties and the resulting PMF. AMP is the only model that actively includes point charge gradients in the training. This could be a key factor in its ability to capture

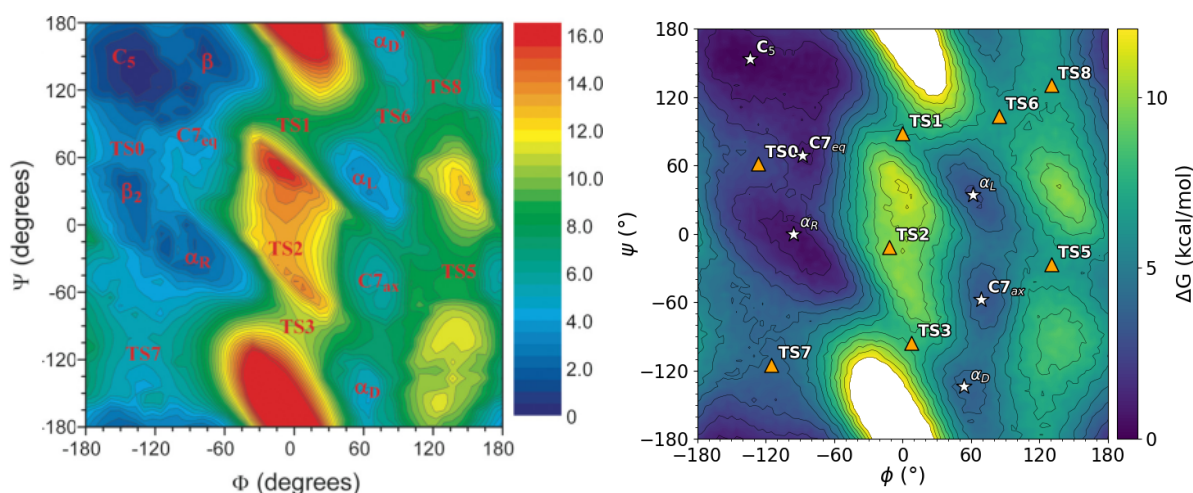


Figure 6.24.: Analysis of the minima and transition states in the reference PMF at the B3LYP/6-311+G(2d,p) level in implicit solvent¹⁹⁴ (left) and the QEq-MACE PMF in explicit water (right). The general topology of the PMF is captured well, but the relative free energies are underestimated. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the $(\phi-\psi)$ Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

Table 6.5.: Comparison of free energy extrema between the reference PMF in implicit solvent at the B3LYP/6-311+G(2d,p) level¹⁹⁴ and the observed extrema from the QEq-MACE water PMF analysis. Angles are in degrees, and free energies are in kcal/mol. Energies are referenced to the C_5 minimum. Reprinted (adapted) with permission from Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the $(\phi-\psi)$ Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n. Copyright 2026 American Chemical Society.

Label	ϕ_{ref}	ϕ_{obs}	ψ_{ref}	ψ_{obs}	ΔG_{ref}	ΔG_{obs}	$\Delta(\Delta G)$
C_5	-151.60	-134.04	147.60	153.19	0.00	0.00	0.00
α_R	-78.10	-95.74	-27.20	0.00	1.60	0.33	-1.27
β	-75.10	—	143.30	—	0.06	—	—
α_L	61.30	61.28	40.90	34.47	3.30	2.64	-0.66
$C7_{\text{ax}}$	73.40	68.94	-53.00	-57.45	4.32	3.13	-1.19
α_D	60.10	53.62	-147.70	-134.04	4.30	3.29	-1.01
TS0	-129.80	-126.38	62.60	61.28	2.63	2.51	-0.12
TS1	0.30	0.00	91.60	88.09	7.88	7.19	-0.69
TS2	-11.00	-11.49	-11.70	-11.49	12.52	9.74	-2.78
TS3	7.30	7.66	-92.10	-95.74	9.99	7.86	-2.13
TS5	132.20	130.21	-28.10	-26.81	7.78	6.44	-1.34
TS6	81.30	84.26	104.20	103.40	6.45	6.42	-0.03
TS7	-114.00	-114.89	-115.60	-114.89	4.99	3.75	-1.24
TS8	127.90	130.21	133.70	130.21	7.09	6.57	-0.52

the solvation structure more accurately than the other models. It might also enforce AMP to predict more physically reasonable atomic mono- and multipole moments, which could contribute to its otherwise elevated performance in solvent compared to the other models.

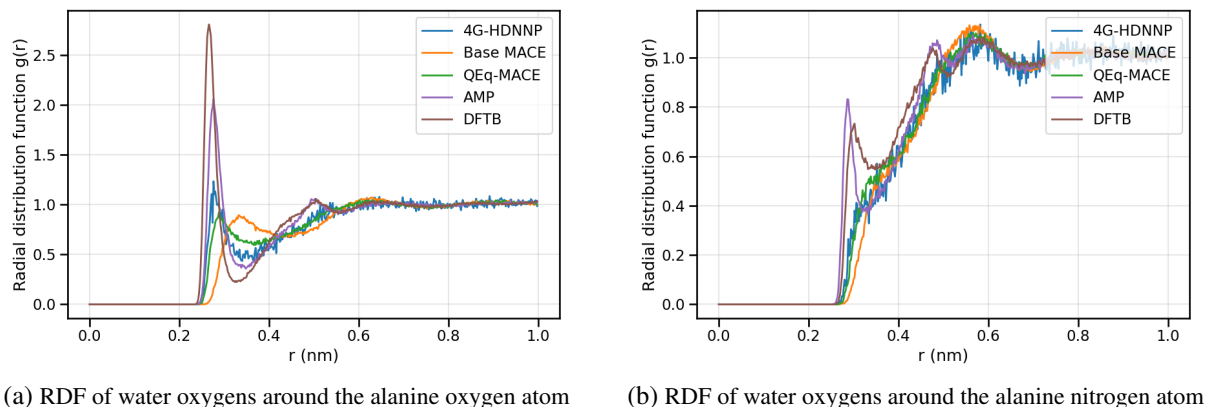


Figure 6.25.: Comparison of RDFs for water oxygens around the alanine oxygen atom and nitrogen atom during simulations with different methods. DFTB and AMP show a similar solvation structure.

Environment Transferability

As an initial test of the transferability of the trained MLPs to slightly shifted environments, a sample of 76 structures from a single walker of the QEq-MACE simulation was recalculated with the reference method B3LYP/aug-cc-pVTZ, PBE0/aug-cc-pVTZ, DFTB3/3OB, and the ML/MM models. To adjust the environment, the cutoff for the environment point charges was varied from 5.0 Å to 30.0 Å, thereby varying the size of the water bubble around the solute. A 0.0 Å cutoff was simulated by setting the cutoff to 1.0 Å and setting the environment point charges to zero, so that only Lennard-Jones interactions with the environment are included. The maximum force difference to the B3LYP/aug-cc-pVTZ reference was calculated for each method and cutoff, and the results are shown in Figure 6.26. DFTB has the highest deviation from the B3LYP/aug-cc-pVTZ reference in agreement with the previous analysis, but it is relatively unaffected by the cutoff. Removing the environment point charges improves the performance of DFTB, as their presence introduces an additional error. In contrast, the environment-agnostic Base MACE performs worse without the environment point charges, as it still attempts to predict mean solvent effects that are absent in that case. The error of the DFT functional PBE0 is completely unaffected by the cutoff, as it adapts to the environment in the same way as B3LYP. 4G-HDNNP and QEq-MACE show an increase in error with decreasing cutoff. Most of the training data were generated with a cutoff of 4.0 Å, so the performance drop-off is expected for smaller cutoffs when evaluating unseen ESP distributions. Removal of the environment point charges improves performance again, as the additional error source is removed, and these models receive information about the environment’s absence through the ESP features. AMP is the only model that is relatively unaffected by the cutoff, which indicates that

it has learned to adapt to different environments and is not heavily reliant on a specific cutoff for its performance. This suggests that AMP has better generalization across different environments than the other models, a desirable property for applications in condensed-phase systems, where the environment can vary significantly.

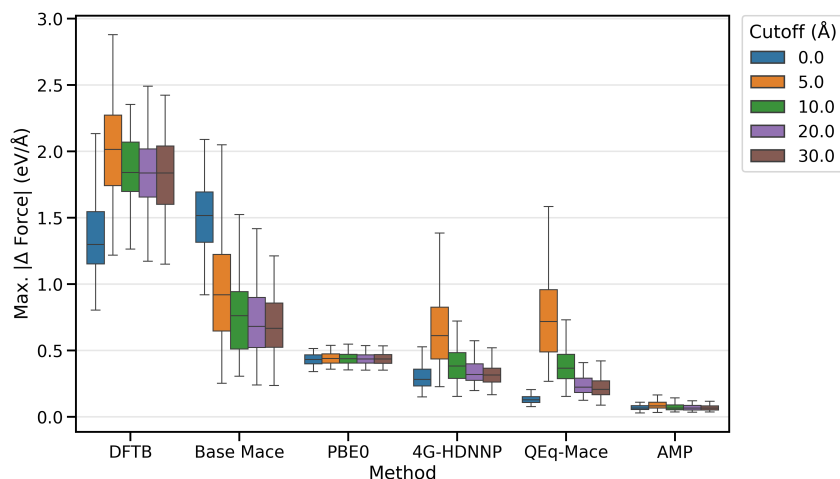


Figure 6.26.: Maximum absolute force difference to the B3LYP/aug-cc-pVTZ reference for different environment cutoffs. DFTB, PBE0, and AMP show high environment transferability.

6.3.4. Conclusion

QEq-MACE shows a significant overall improvement over the MACE baseline. Simulations run stably, the force deviations are reduced, and the topology of the PMF in water is captured in greater detail. Relative free energies of the minima and transition states are underestimated, but the differing reference methods hinder a final assertion of the quality. The 4G-HDNNP model fails to capture the solvent effect and shows a significant increase in error compared to vacuum, which leads to unphysical behavior in the simulations and inaccurate PMFs. AMP is the only model that maintains a similar level of accuracy in solvent as in vacuum, shows a realistic solvation structure, and is relatively unaffected by slightly shifted environments, making it a promising candidate for future applications in condensed phase systems.

6.4. Training Thiol Disulfide Exchange Reaction in Environment

Scope In this section, we apply the same analysis as for the alanine dipeptide system to the thiol disulfide exchange reaction. We first compare the properties of the solute in vacuum and in solvent to understand the effect of the environment. Then, we train different MLP architectures on the same dataset generated in solvent and compare their performance on a test set. The comparison to their performances in vacuum reveals the additional error introduced by the environment. We run metadynamics simulations in the solvent using the trained MLPs and compare the resulting PMFs with a DFTB reference. As a biologically relevant application, we perform a production metadynamics simulation on the C4 domain of von Willebrand Factor (vWF) to investigate the thiol disulfide exchange reaction in the full protein and solvent environment. The vWF is a protein that plays a crucial role in blood clotting, and the thiol disulfide exchange reaction is a key step in its function. Understanding this reaction in the context of the full protein and solvent environment is important for gaining insights into its mechanism and potential therapeutic interventions.

6.4.1. Environment Comparison

The thiol disulfide exchange reaction exhibits a significant solvent effect, as explained in greater detail in Section 4.2. The typical linear S_N2 transition state is actually the global minimum in vacuum. At the same time, the presence of solvent stabilizes the reactant and product states more than the transition state, leading to a reaction barrier in water, as seen in Figure 6.27. This stabilization is only reflected in a subset of the sampling methods investigated in Chapter 4, as seen in Figure 6.28. The protein simulations and the modified force-field simulation show a significant solvent effect, whereas the effect is insignificant for the potential energy scan, the ‘water’ simulation, and the uncertainty sampling. This discrepancy arises from the solvation procedure applied to the configurations. While the protein simulations and the modified force-field simulations were performed in explicit solvent, the other sampling methods were generated in vacuum, solvated using a highly structured solvent box, and relaxed with a short geometry optimization of the solvent. This procedure led to insufficient solvent relaxation, the artificial solvation remained highly symmetrical, and the environment point charges largely canceled each other out, as reflected in the ESP values across the different sampling methods. Nonetheless, all these configurations are included in the training dataset, as they represent a significant portion of the configuration space that the MLP must be able to describe for stable simulation.

6.4.2. Training in Environment

Results Many MLPs already failed to achieve chemical accuracy for the thiol disulfide exchange in vacuum, and the presence of solvent further increases the error for all models, none of which achieve

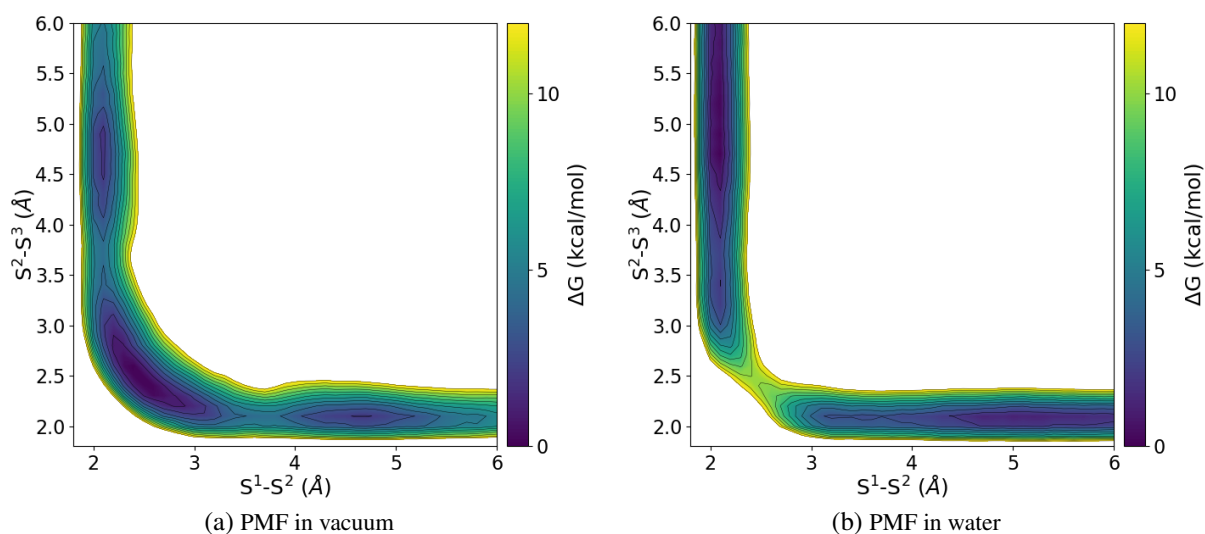


Figure 6.27.: Comparison of PMFs for thiol disulfide exchange reaction in (a) vacuum and (b) water environment at the DFTB3/3OB_{Mod} $S-S$ level of theory. The presence of solvent converts the global minimum in vacuum into a transition state with an 10 kcal/mol barrier in water.

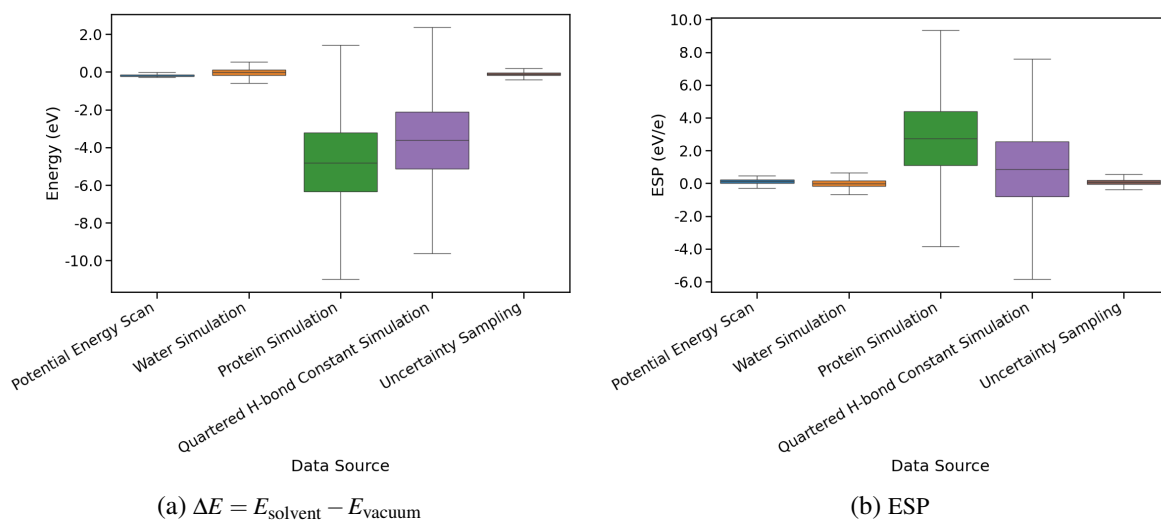


Figure 6.28.: Comparison of environment effect for thiol disulfide exchange reaction regarding (a) energy, and (b) ESP across different sampling methods. Insufficient solvent relaxation for some sampling methods leads to an artificial cancellation of the environment effect.

chemical accuracy in solvent. With the solvent effect being stronger, the environment-agnostic Base MACE baseline performs even worse for the thiol disulfide exchange reaction than for the alanine dipeptide. Field-Schnet also fails to improve on the baseline. 4G-HDNNP and QEq-MACE increase their energy RMSE by about 0.1 eV, which is significantly worse than the estimated error increase of 0.01 eV from the point charge approximation of the electrostatic interactions, but still manage to capture the solvent effect to some extent. AMP shows the best performance in solvent, especially the force RMSE is almost unaffected by the solvent, which is consistent with the results for the alanine dipeptide. The energy RMSE is also the lowest among the tested models, despite not meeting the chemical accuracy threshold.

Table 6.6.: Test set performance metrics for different MLP architectures trained on the thiol disulfide exchange reaction dataset in water. Values represent mean performance across up to three independent training runs. AMP shows the best performance in solution.

Method	Energy		Force		Charge	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2	RMSE (e)	R^2
4G-HDNNP	0.210	0.993	0.098	0.993	0.016	0.992
Base MACE	1.957	0.426	0.181	0.975	—	—
QEq-MACE	0.147	0.997	0.076	0.996	0.042	0.948
AMP	0.133	0.997	0.032	0.999	—	—

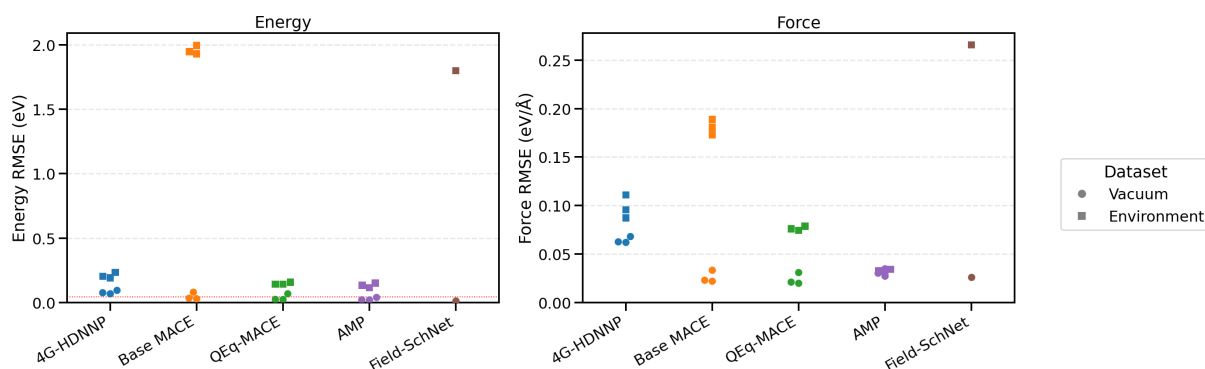


Figure 6.29.: Comparison of test set RMSE for different MLP architectures trained on the thiol disulfide exchange reaction dataset in water. AMP shows the best performance in solvent with almost no increase in force RMSE compared to vacuum.

6.4.3. Metadynamics Simulation in Water

No MLP achieves full simulation completion for all 16 walkers for the full 0.75 ns simulation, but the MACE architectures reach 96% completion. AMP shows its weakest performance in the simulations, with only 82% completion. Most of the terminations were initiated safely by high energy prediction standard deviations when the thiol disulfide broke into three thiols. Still, sudden molecular explosions also

occurred for AMP, terminating the simulation before safety checks could intervene. The 4G-HDNNP model again failed even to complete 20% of the simulation.

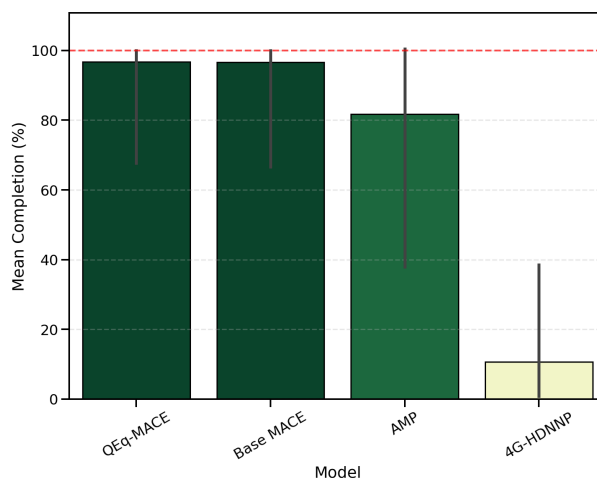


Figure 6.30.: Completion of 16 0.75 ns walker metadynamics simulation in water for different MLP architectures. MACE architectures achieve the best performance, with 96% completion, whereas AMP achieves only 82% completion.

The resulting PMFs from the simulations in Figure 6.31 reveal the failure of the 4G-HDNNP and Base MACE to pass the reaction barrier during the simulation. In the case of 4G-HDNNP, the reason is insufficient convergence due to simulation instability, whereas for Base MACE, the reason is inaccurate predictions due to missing environmental information. QEq-MACE and AMP can pass the reaction barrier and sample the product state. The heights of the reaction barrier with reference to the global minimum amount 7.6 kcal/mol for QEq-MACE, 8.3 kcal/mol for AMP, and 10.0 kcal/mol for the DFTB3/3OB_{Mod S-S} reference. Differences to the DFTB3/3OB_{Mod S-S} reaction barrier of 22 kcal/mol from Gómez-Flores et al.²¹² might arise from the different scaling of environment charges, loosened restraints on the attack angle, or the 3-CV setup in the reference. The PMF should be symmetrical as the reactant and product states are identical, but the QEq-MACE, AMP and DFTB PMFs show a slight asymmetry and different global minima with a free energy difference of 1.8 kcal/mol, 2.5 kcal/mol, and 1.0 kcal/mol, respectively. This asymmetry could be due to insufficient sampling and to convergence issues.

The PMF quality evaluation via maximum force difference to the B3LYP/aug-cc-pVTZ reference in Figure 6.32 fortifies the finding of AMP's better performance in ML/MM simulations, as it shows the lowest force deviation among the tested models, barely increasing from vacuum to solvent. QEq-MACE about doubles its force deviation compared to vacuum and is now on par with the 4G-HDNNP model, but shows a better stability in the simulations. The improvement over the baseline MACE remains significant, particularly for peak force differences, which likely prevent the Base MACE simulations from crossing the reaction barrier. PBE0 and DFTB3/3OB_{Mod S-S} differences stay consistent with the vacuum analysis, as observed in the preliminary environment transferability analysis with different cutoffs.

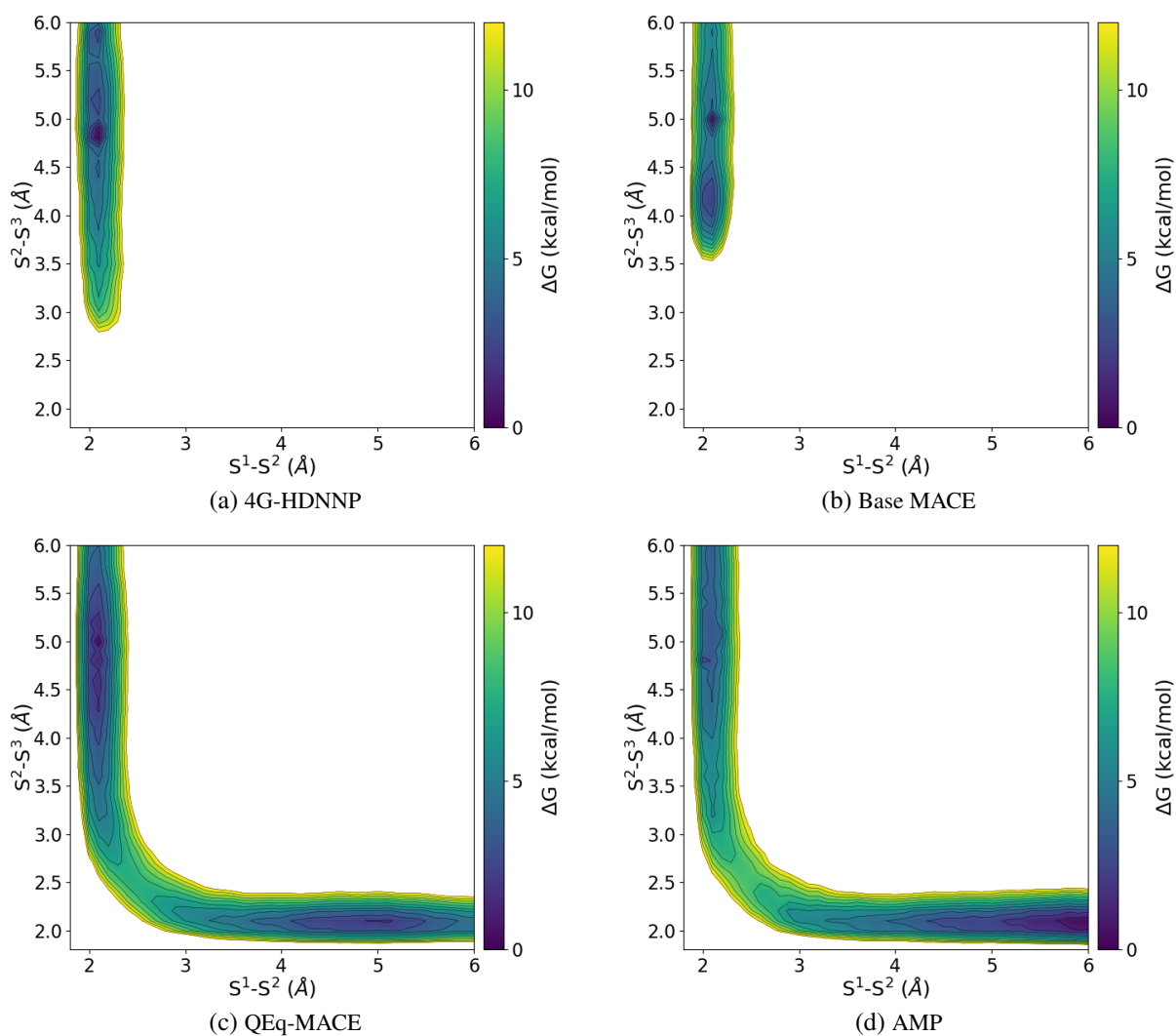


Figure 6.31.: Thiol disulfide water simulation PMF comparison

Resulting PMFs after 0.75 ns metadynamics simulation in water for different MLPs. Contour lines are drawn at 1.0 kcal/mol intervals. QEq-MACE and AMP can pass the reaction barrier and sample the product state, while 4G-HDNNP and Base MACE fail to pass the reaction barrier.

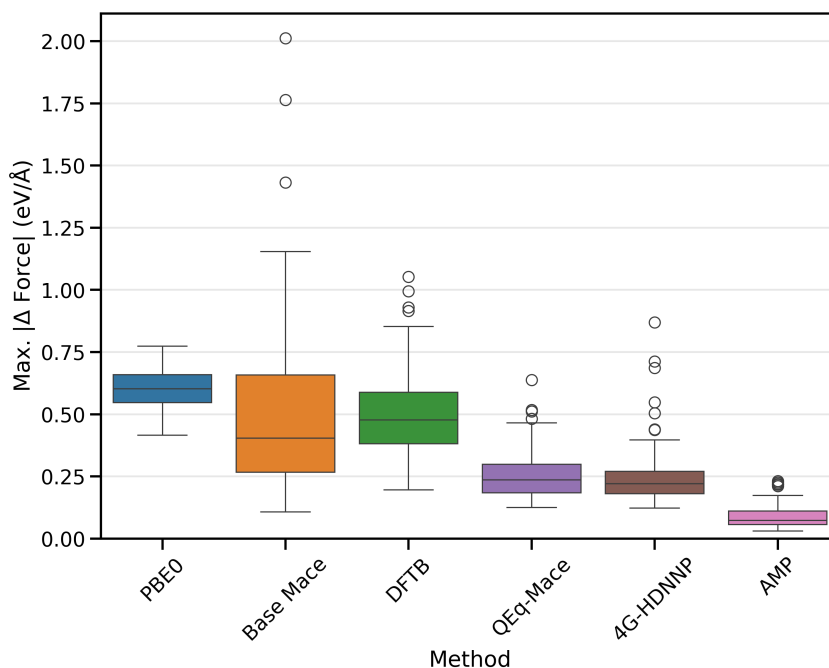


Figure 6.32.: Maximum absolute force difference to the B3LYP/aug-cc-pVTZ reference during 0.75 ns water simulation for different methods. AMP shows the lowest force deviation among the tested models, barely increasing from vacuum to solvent, while QEq-MACE about doubles its force deviation compared to vacuum.

Models incorporating environmental information continue to consistently outperform DFTB in solution, despite being fitted to the same functional.

6.4.4. Metadynamics Simulation in Protein

The big advantage of ML/MM models in contrast to big foundation models such as ANI^{43,219} and MACE-OFF²²⁰ is the reduction of the training data on the specific system of interest in a bigger environment while the environment itself gets featurized without having to train all the complex interactions in the environment. This allows for more efficient training, but requires the MLP to generalize to unseen environments. To test the performance of the trained MLPs in a more complex environment than the previous water bubble transfer, we run metadynamics simulations of the thiol disulfide exchange reaction in the C4 domain of the vWF in explicit solvent (see Figure 6.33). The QM region includes the three cysteine residues involved in the reaction, while the rest of the protein and the solvent are included in the MM region. Only the best-performing MLP from the water simulations, QEq-MACE and AMP, are tested in the protein simulations; DFTB is included as a reference. The simulations are run for 0.75 ns with 24 walkers, and the resulting PMFs are compared to the DFTB3/3OB_{Mod S-S} reference. The simulation setup is described in more detail in Section 3.5.1.

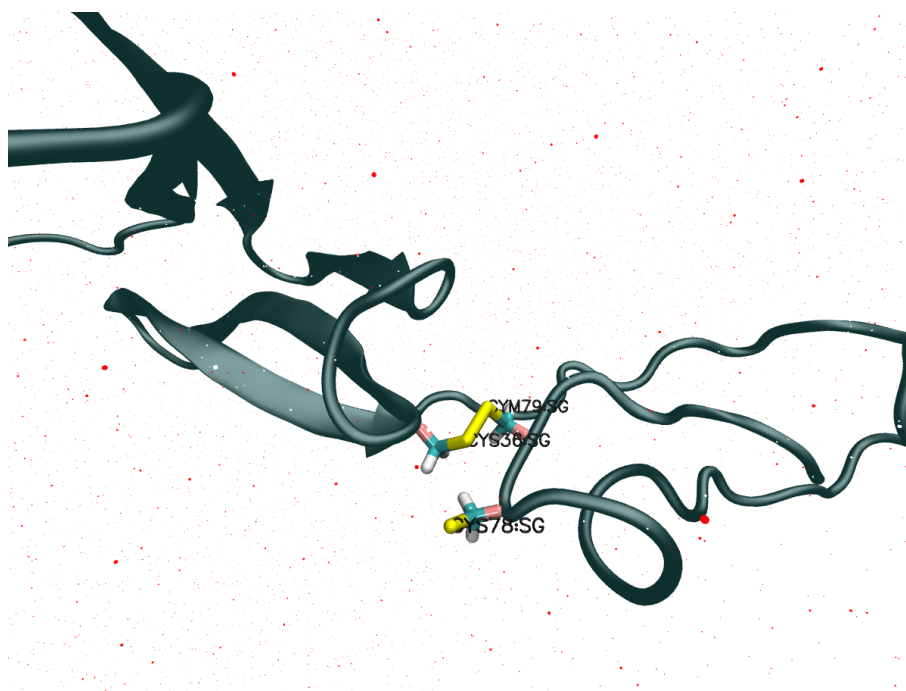


Figure 6.33.: Thiol disulfide exchange reaction in the C4 domain of the vWF using residue 36 as electrophile. The reaction is simulated in the context of the full protein and solvent environment.

Both the walker completion and the resulting PMFs are depicted in Figure 6.34. Both MLPs have a lower walker completion rate than previously observed, but ultimately QEq-MACE still achieves almost 90% completion. In contrast, AMP achieves 75% completion, indicating better stability than other models in less complex environments. The adjusted reaction setup, placed in a more challenging environment, can also contribute to the lower completion rates, as more walkers and a tighter restraint on the sulfur-sulfur distances lead to a faster sampling of the reaction. This might bias the reaction to some extent, so that both $S-S$ bonds are broken. Structures with broken $S-S$ bonds are not included in the training dataset, leading to a higher chance of simulation failure, although the inspection of the CV values shows no major impact of the double bond breaking.

All models capture the expected reaction barrier, the two minima, and show a third minimum at longer S^1-S^2 distances, which is probably an artifact from the maximum distance restraint of $6.0 \text{ \AA}$. DFTB shows a nearly symmetrical PMF with a reaction barrier of 9.5 kcal/mol and reaction free energy ΔG of -0.4 kcal/mol . The reaction barrier and ΔG are very close to the result of the previous study with nearly the same reaction setup by Gómez-Flores et al.²¹² of 8.7 kcal/mol and -0.3 kcal/mol , respectively. The only setup difference is the slightly reduced metadynamics parameters to reduce the total bias at the end of the simulation. The MLPs are in agreement with each other, but differ from DFTB. The PMF is asymmetrical with a more negative ΔG of -2.6 kcal/mol for QEq-MACE and -1.8 kcal/mol for AMP. The reaction barrier is significantly reduced to 5.6 kcal/mol for QEq-MACE and 4.0 kcal/mol for AMP, which is less than half of the DFTB barrier. The reference B3LYP/aug-cc-pVTZ barrier is not feasible to

calculate in the full protein and solvent environment, but so far, MLPs have shown the ability to capture solvent effects on the reaction barrier in water simulations.

The recalculation of a sample of about 100 structures from a single walker again demonstrates the reliability of AMP in the new environment, as it exhibits the lowest force deviation relative to the B3LYP/aug-cc-pVTZ reference among the tested models. QEq-MACE shows a higher error than AMP, but still performs consistently better than DFTB. This proof strengthens confidence in the observed PMF trends toward a lower reaction barrier and a more negative reaction free energy than in the DFTB reference. However, the exact values of the reaction barrier and reaction free energy should be treated with caution because there is no B3LYP/aug-cc-pVTZ reference.

Conclusion MLPs in Environment

Direct training in an environment confirms that explicit solvent information is essential for robust ML/MM simulations. For both alanine dipeptide and thiol disulfide exchange, the environment-agnostic Base MACE degrades substantially from its vacuum performance, whereas models with explicit environment coupling recover a substantial portion of the lost accuracy and stability. This result highlights the importance of incorporating environment information into MLP architectures for accurate simulations in solvent. Across both systems, QEq-MACE consistently improves upon the baseline and reproduces the relevant PMF topology in the solvent, but it still underestimates parts of the free-energy landscape. 4G-HDNNP does not transfer reliably to solvent in this setup and shows frequent simulation instabilities despite acceptable behavior in selected static metrics.

The transferability of the trained MLPs to modified environments via cutoff variation reveals that AMP has the best generalization capability to different environments, as it maintains a similar level of accuracy in solvent as in vacuum and is relatively unaffected by the cutoff. This suggests that AMP has learned a physically meaningful representation of the environment that enables it to adapt across different environments, a desirable property for applications in condensed-phase systems where the environment can vary significantly. In contrast, QEq-MACE exhibited rapid performance deterioration as the cutoff decreased. The learned charges are likely unreliable under unseen electrostatic conditions, and the model relies on familiar electrostatic conditions to perform. AMP was even able to keep its good performance in the protein environment, the most challenging environment tested in this chapter.

AMP is the most promising model in this chapter: it maintains the smallest force degradation from vacuum to solvent, shows favorable transfer behavior to modified environments, and reaches the best overall agreement to reference trends in condensed phase. Still, rare but severe simulation failures indicate that AMP could also benefit from regularization techniques to prevent sudden force spikes by multiple orders of magnitude and a more robust implementation to safeguard against edge cases such as

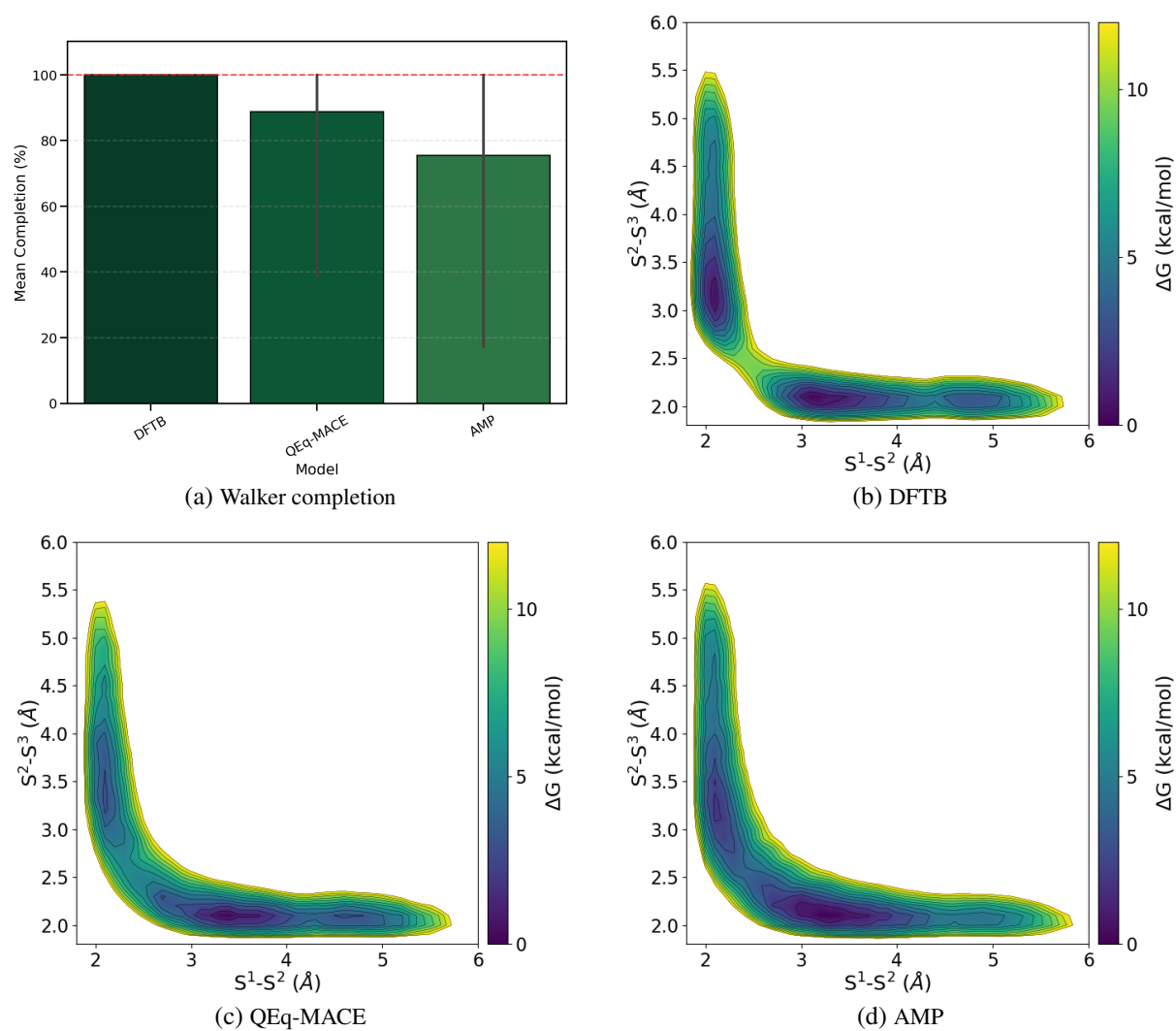


Figure 6.34.: Thiol disulfide protein simulation PMF comparison

Completion of 24 0.75 ns walker metadynamics simulation in protein (a) and comparison of resulting PMF for DFTB, QEq-MACE, and AMP. Contour lines are drawn at 1.0 kcal/mol intervals. QEq-MACE and AMP show a significantly reduced reaction barrier and more negative reaction free energy compared to DFTB.

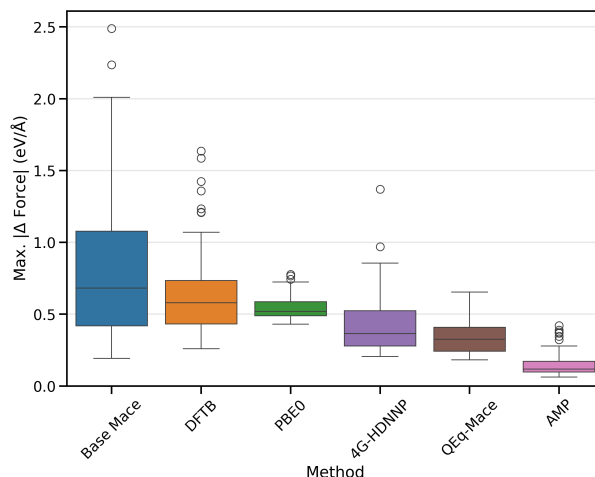


Figure 6.35.: Maximum force difference to the B3LYP/aug-cc-pVTZ reference during 0.75 ns protein simulation for different methods.

missing neighbors or environment point charges for an atom within its cutoff, which occurred in some of the simulation-breaking configurations. QEq-MACE, despite its higher force degradation and sensitivity to unseen electrostatic conditions, still showed a robust stability in the simulations and a good agreement with reference trends. The agreement of the QEq-MACE and AMP PMFs in the protein simulation indicate the advantages of environment-aware MLP architectures for condensed phase simulations over competing semi-empirical QM/MM methods, which is a promising result for future applications of ML/MM simulations in complex environments.

6.5. Δ -Learning in Vacuum and Solvent

A different approach to training MLPs for QM/MM simulations is to train the model on the difference between a high-level reference method and a lower-level baseline method, which is commonly referred to as Δ -learning.⁸ This approach can be advantageous when the baseline method captures a significant portion of the relevant physics, as it allows the MLP to focus on learning the residual error, which is often smaller and smoother than the total energy landscape. In the context of ML/MM simulations, this can be particularly useful when the baseline method is a semi-empirical QM method that already includes some treatment of the environment, as it can help mitigate the challenges associated with learning the complex interactions in condensed-phase systems. In this section, we apply Δ -learning to train MLPs for both the alanine dipeptide and thiol disulfide exchange systems, using B3LYP/aug-cc-pVTZ as the high-level reference and DFTB3/3OB or DFTB3/3OB_{Mod S-S} as the baseline method, respectively. We evaluate the performance of different MLP architectures in this Δ -learning setup and compare it to direct training on the total energy and forces. DFTB3/3OB is not actually fitted to reproduce B3LYP/aug-cc-pVTZ energies, while DFTB3/3OB_{Mod S-S} is a modified version of DFTB3/3OB directly fitted to reproduce B3LYP energies for the thiol disulfide exchange reaction, which is expected to lead to a smaller and smoother error landscape for the MLP to learn.

6.5.1. Reference Comparison

Scope Before training the MLPs on the Δ -learning targets, an initial comparison of the reference and baseline methods is conducted to understand the order of magnitude of the error that the MLP needs to learn and to identify any systematic deviations that could impact the training. This comparison is performed for both energies and forces in vacuum and solvent environments, as well as for partial charges, which are relevant for the ML/MM setup.

Results DFT and DFTB energies are not directly comparable as DFTB only considers valence electrons, not core electrons. A typical approach to make them comparable for datasets containing many different molecules is to use the atomization energy instead of the total energy, which is the energy difference between the molecule and its constituent atoms calculated at the same level of theory. However, since the dataset for the alanine dipeptide system contains only one molecule, we can directly compare the total energies by normalizing them to a single reference point, obtained by subtracting the mean energy of each method from the respective energies. The comparison in Figure 6.36 shows the result for the data from the different sampling methods investigated in Chapter 4. The relatively high energies of the structures from the modified force field parameter sampling are overestimated by DFTB compared to DFT, while the energies of the other structures are underestimated. The total RMSE is 0.12 eV, ranking it

worse than 2G-HDNNP during vacuum training and missing chemical accuracy by a factor of 3. Again, DFTB3/3OB is fitted to a different functional, so this error is not unexpected, but it sets the scale of the error that the MLP needs to learn in the Δ -learning setup.

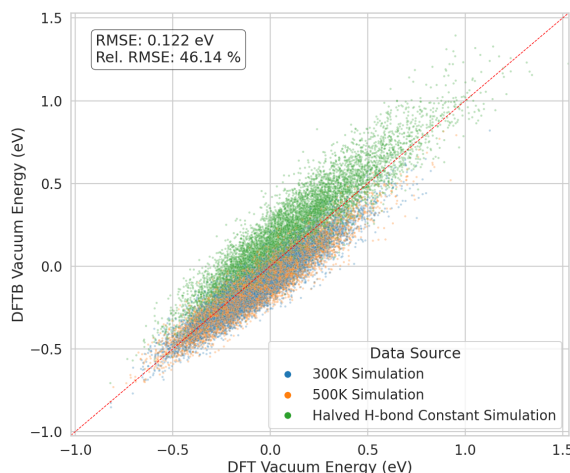


Figure 6.36.: Comparison of B3LYP/aug-cc-pVTZ and DFTB3/3OB energies rectified by their respective mean values for the alanine dipeptide system in vacuum. DFTB overestimates energies of relatively high-energy structures from modified force field parameter sampling (green), and remaining structures get underestimated.

A similar comparison is performed for the force components in Figure 6.37. Here, the relative RMSE quantifies the RMSE relative to the root mean square of the reference data. For hydrogen and carbon, the overall correlation remains close to linear, with a relative RMSE of around 22 %, with some functional-group-specific deviations. For nitrogen and oxygen, the correlation is worse with a relative RMSE around 40 %. Noteworthy is the distribution of the forces on the oxygen atoms for the different sampling methods. The force field sampling with modified parameters constantly overestimates the (absolute) force components with DFTB, while the other sampling methods show an underestimation. There is likely a different displacement range for the C=O bond in the force-field simulation and in the DFTB-driven simulations at different temperatures. Oxygen and nitrogen forces are most likely responsible for the high maximum absolute force error in figures like Figure 6.12.

When investigating which additional error is introduced when going from vacuum to solvent, the energy and force differences in vacuum are plotted against the respective differences in solvent in Figure 6.38. Energies in vacuum and solvent are rectified by each method’s mean energy in vacuum to make them comparable. The correlation is relatively poor, with relative RMSE values of 125 % for energy and 65 % for forces. This implies that learning the error in vacuum does not necessarily translate to learning the error in solvent, which is a main assumption of the Δ -learning approach as a stepping stone to ML/MM training. Even with a perfect Δ -learning model that learns the error in vacuum perfectly, the error in solvent would still be substantially above chemical accuracy. The specific test for vacuum to solvent

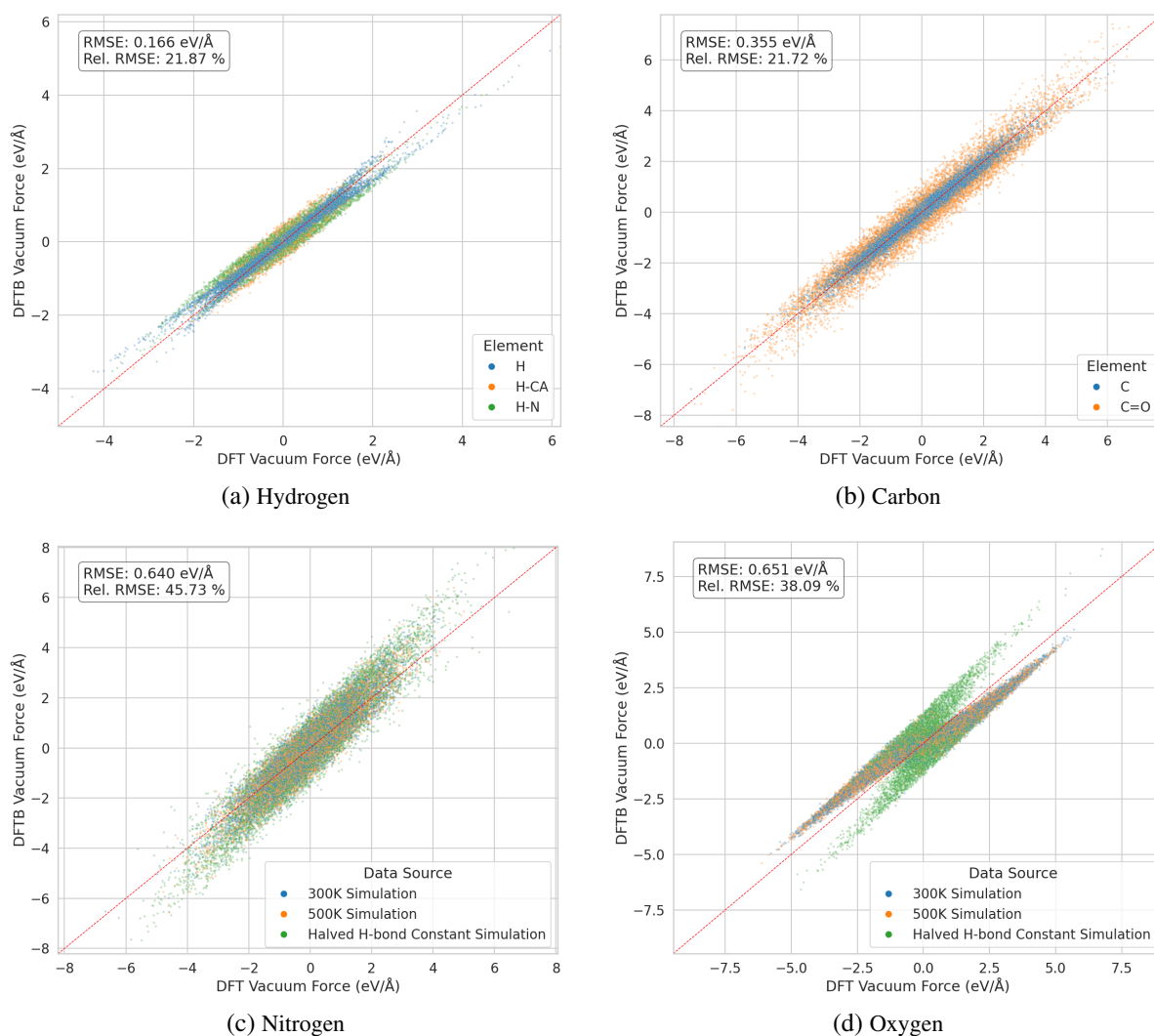


Figure 6.37.: Comparison of B3LYP/aug-cc-pVTZ and DFTB3/3OB force components for alanine dipeptide system in vacuum for different elements. CH_3 hydrogens (a, blue) show the best agreement. Nitrogen (c) and oxygen (d) atoms show larger deviations. Carbonyl carbon (b, orange) performs worse than sp^3 carbons (b, blue).

transferability of the Δ -learning was therefore not attempted in this thesis. Instead, solvent differences were learned directly.

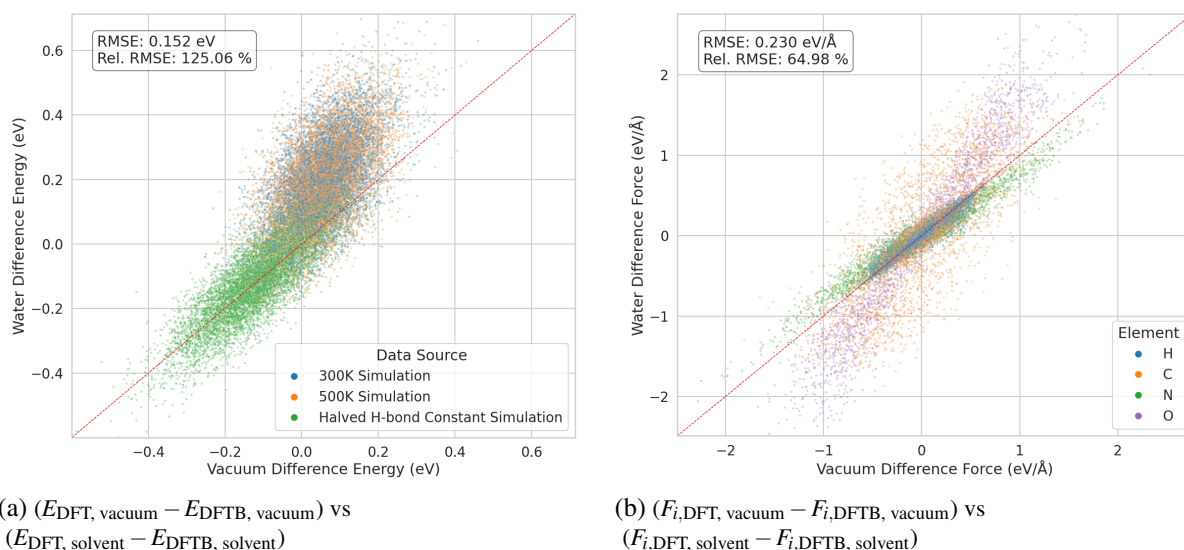


Figure 6.38.: Correlation between vacuum and solvent errors of energy and force components for alanine dipeptide with B3LYP/aug-cc-pVTZ and DFTB3/3OB. The imperfect correlation between the errors in vacuum and solvent indicates that learning the error in vacuum does not necessarily translate to learning the error in solvent, which is a challenge for the Δ -learning approach.

In Δ -learning, only energy differences should be learned, so the absolute values of the energies are not relevant. This has implications for charge-based MLP architectures such as 4G-HDNNP and QEq-MACE, which use atomic charges as an intermediate representation to compute the intermolecular electrostatic energy contribution. If a full partial charge is used as a target, then the MLP will add a full electrostatic contribution to the energy, which is not what we want in Δ -learning. Instead, the MLP should learn the difference in the partial charges between the reference and baseline method, which is expected to be smaller and more suitable for Δ -learning. However, a problem arises when the Mulliken charges from DFT and DFTB show a poor correlation, as shown in Figure 6.39. This poor correlation indicates that the Mulliken charges from DFTB3/3OB are not a good representation of the B3LYP/aug-cc-pVTZ charges, which makes it difficult for the MLP to learn the difference in charges effectively. Therefore, the Δ -learning of Mulliken charges was not attempted in this thesis. Instead, charge targets were set to zero to prevent the model from becoming overly creative. Additionally, the total charge is also considered a Δ total charge, which is always zero for both neutral and charged systems. Balancing the electrostatic and atomic contributions is a major source for overfitting in charge-based MLP architectures without a guiding target charge.

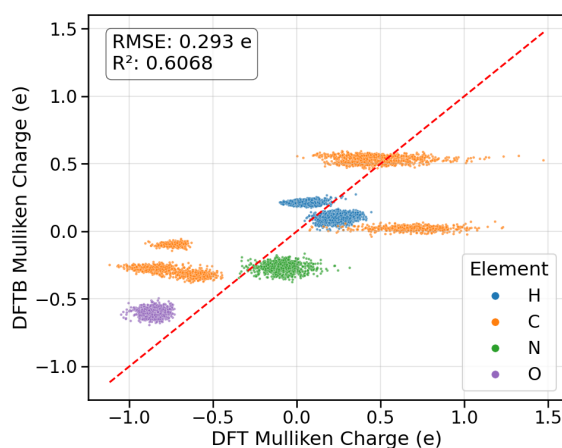


Figure 6.39.: Comparison of B3LYP/aug-cc-pVTZ and DFTB3/3OB Mulliken partial charges for alanine dipeptide system in vacuum. The correlation is poor; DFTB Mulliken charges are not a good representation of DFT charges. Therefore, the Δ -learning of Mulliken charges was not attempted.

6.5.2. Δ -Learning Performance

Scope In this section, several MLP architectures are trained on the Δ -learning targets to answer the question whether Δ -learning can improve the performance of MLPs for QM/MM simulations compared to direct training on the total energy and forces. The Δ -learning data preparation included calculating atomization energies using the atomic energies computed at the same level of theory for the respective method. Their values are reported in Section 3.3. The Δ -learning targets are then calculated as the difference between the reference and baseline method for the atomization energies and forces. Four architectures are trained on the Δ -learning targets for the alanine dipeptide system: 2G-HDNNP, 4G-HDNNP, Base MACE, and QEq-MACE. Three independent training runs, each using separate training data and identical test data, are performed for each architecture and environment. The hyperparameters are inspired by the best-performing hyperparameters from direct training, but reduced in size to account for the expected easier learning task. The exact parameters are reported in Section D.6.

Results The test set performance metrics are reported in Table 6.7 and compared to direct training in vacuum and solvent in Figure 6.40. The HDNNP models improve through Δ -learning both in vacuum and solvent, now clearly surpassing chemical accuracy in vacuum. Δ -4G-HDNNP now performs on a similar level as the direct training of QEq-MACE in solvent for both energy and forces. The Base MACE model shows no improvement in the already near-perfect performance in vacuum, but improves during solvent training. Without environment information, both 2G-HDNNP and Base MACE hit the same error floor in solvent for energy and forces, despite Base MACE’s much better performance in vacuum. The lack of environmental information is therefore the main bottleneck for the performance of the Δ -learning models in solvent, posing a major challenge for the Δ -learning approach as a stepping

stone to ML/MM training. 4G-HDNNP surpasses the bottleneck for the force prediction and performs almost twice as well as the environment-agnostic models. The QEq-MACE model exhibits a different pattern, with a substantial performance deterioration in both the vacuum and the solvent compared to direct training. Especially the negative R^2 for energy prediction in the solvent indicates that the model is unable to learn the Δ -learning targets. The lack of appropriate charge targets leads to major training issues in identifying charge-related parameters, as the optimization converges to local minima with high errors despite fine-tuning the respective hyperparameters. Δ -learning is not a reliable approach for QEq-MACE in this setup.

Table 6.7.: Test set performance metrics for different MLP architectures using Δ -learning on alanine dipeptide. Values represent mean performance across three independent training runs for the vacuum and environment datasets. 2G-HDNNP and Base MACE converge to the same error floor in solvent. QEq-MACE deteriorates substantially in both vacuum and solvent compared to direct training, especially for the energy prediction in solvent.

Δ Vacuum				
Method	Energy		Force	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2
2G-HDNNP	0.018	0.978	0.030	0.993
4G-HDNNP	0.016	0.982	0.028	0.994
Base MACE	0.005	0.998	0.011	0.999
QEq-MACE	0.015	0.985	0.020	0.997

Δ Environment				
Method	Energy		Force	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2
2G-HDNNP	0.101	0.726	0.112	0.941
4G-HDNNP	0.106	0.700	0.060	0.983
Base MACE	0.101	0.726	0.109	0.944
QEq-MACE	0.247	-0.702	0.084	0.967

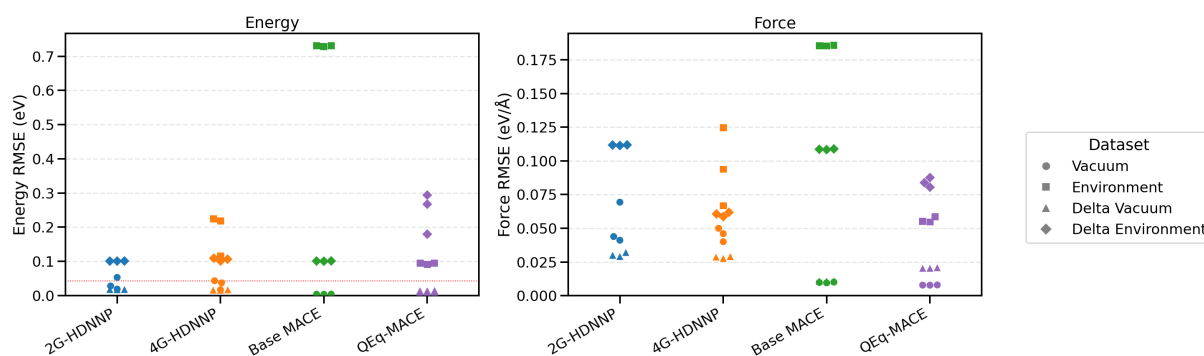


Figure 6.40.: Comparison of test set RMSE for different MLP architectures using Δ -learning on alanine dipeptide in vacuum and water. HDNNP models improve through Δ -learning both in vacuum and solvent, Base MACE only in solvent. QEq-MACE deteriorates substantially in both vacuum and solvent compared to direct training, especially for the energy prediction in solvent.

6.5.3. Metadynamics Simulation

The capability to perform Δ simulations was added to GROMACS by integrating the DFTB and MLP implementations into a single program for both TensorFlow and PyTorch. The implementation was validated, but despite the promising test set performance of the Δ -learning models, the simulations revealed major stability issues, with large force spikes leading to simulation crashes. No new insight was gained from the simulations, so they were not pursued further.

6.5.4. Δ -Learning Thiol Disulfide Exchange Reaction

The Δ -learning approach is also applied to the thiol disulfide exchange reaction system, using B3LYP/aug-cc-pVTZ as the high-level reference and DFTB3/3OB_{Mod S-S} as the baseline method. The references are compared, the same MLP architectures are trained on the Δ -learning targets for this system, and their performance is evaluated and compared to direct training on the total energy and forces. The results and insights coincide with findings from the alanine dipeptide system. The analysis is reported in Appendix D.7.

6.5.5. Conclusion

The reference comparison reveals that the errors in vacuum and solvent are not well correlated. This diminishes the main advantage of the Δ -learning approach for the use in ML/MM training, as the assumption that learning the error in vacuum would directly translate to learning the error in solvent does not hold.

The Δ -learning approach shows promise for improving the performance of weaker MLP architectures, such as 2G-HDNNP and 4G-HDNNP, especially in vacuum, where the error landscape is simpler. The high-level performance of Base MACE in vacuum leaves little room for improvement through Δ -learning, but it does show some improvement in solvent without environment information. Still, good ML/MM models outperform the environment-agnostic Δ -learning models in solvent. In contrast, the QEq-MACE model exhibits substantial performance deterioration in both the vacuum and solvent compared to direct training, because the proper charge targets are missing.

Overall, Δ -learning is not a promising avenue for improving MLP performance for QM/MM simulations anymore. Direct training with specialized ML/MM models, such as QEq-MACE and AMP, for the total energy and forces in solution already achieves good performance without requiring a DFTB calculation during the simulation.

6.6. Computational Cost

Scope MLPs operate in a regime between traditional force fields and QM methods, competing with semi-empirical methods such as DFTB in computational cost. Architectures and implementations are rapidly evolving, and the computational cost of MLPs can vary widely depending on the architecture, system size, and training and inference hardware. In this section, we compare the computational cost of a traditional force field (Amber99SB-ILDN), a semi-empirical method (DFTB), and different MLP architectures for a QM/MM simulation of alanine dipeptide in water. The simulations are performed on a single CPU core (Intel Xeon E6252 Gold) with and without GPU (Nvidia V100S) support. GPU support is available only for the PyTorch-based MLPs (MACE and AMP). Three metrics are used to compare the computational cost of the different methods: the time per nanosecond of simulation for each setup, the scaling of computational performance with the number of parallel processes/walkers, and the scaling of computational performance with system size.

Results A comparison of the computational performance of the different methods for a QM/MM simulation of alanine dipeptide in water using a single CPU core with and without GPU support is shown in Figure 6.41. The QM zone in this setup consists of 22 atoms and the MM zone of ≈ 900 water molecules. The force field Amber99SB-ILDN is, as expected, the uncontested fastest method, being almost an order of magnitude faster than the other methods. DFTB, 2G-HDNNP, and SchNet perform similarly, with DFTB being slightly faster than the simpler MLPs. The more advanced MLPs Base MACE, QEq-MACE, and AMP can reach similar performance with GPU acceleration, but they are significantly slower without GPU support in this setup. The input preparation time for the model execution amounts to about 1% to 10% of the total calculation time; the majority of the time is spent on the model execution itself.

A GPU accelerates the model training significantly, but the effect in the simulation with a single data point per step is less pronounced. Additionally, multiple processes/walkers are used to parallelize the sampling of the free energy landscape during metadynamics simulations. Typically, each walker runs on a separate, readily available CPU core. Access to GPUs is limited; a single GPU would need to be shared during metadynamics with an ML/MM with GPU support. The effect of sharing a single GPU across multiple walkers is investigated by measuring the scaling of computational performance with increased number of parallel processes/walkers for the Base MACE MLP architecture in Figure 6.42. In this setup, the number of CPU cores is adjusted to the number of walkers, but the walkers share a single GPU. The performance drops exponentially with increasing number of walkers until the CPU performance equals the GPU performance. The single data point per step should require only a fraction of the GPU's computing capacity, but because no parallelization across the walkers occurs on the GPU, performance drops significantly with each additional walker, and the GPU becomes the bottleneck. Taking the higher

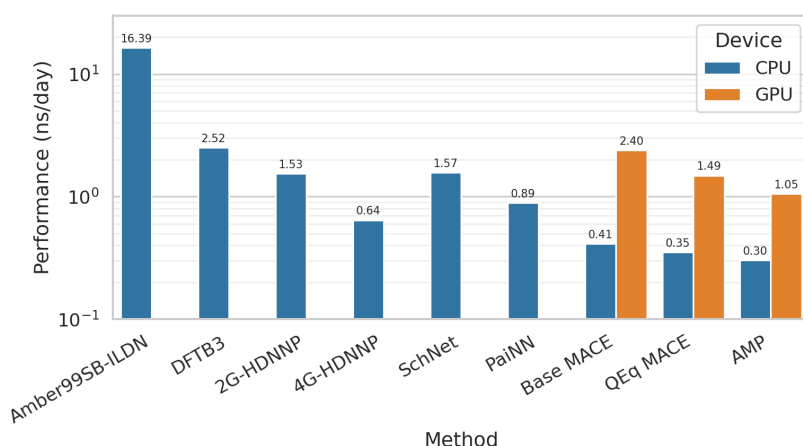


Figure 6.41.: Comparison of method performances for a QM/MM simulation of alanine dipeptide in water using a single CPU core with and without GPU support. The Amber99SB-ILDN force field is the fastest; DFTB, 2G-HDNNP, and SchNet perform similarly. The more advanced MLPs Base MACE, QEq-MACE, and AMP can reach similar performance with GPU acceleration.

energy consumption of the GPU into account (Xeon E6252: 150 W, V100S: 300 W), the break-even point is reached even earlier.

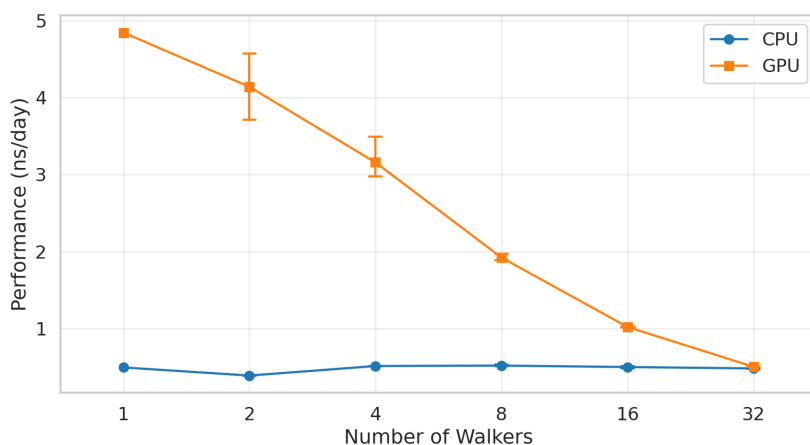


Figure 6.42.: Scaling of computational performance with increased number of parallel processes/walkers for the Base MACE MLP architecture. The number of CPU cores is adjusted to the number of walkers; all walkers share a single GPU. The performance with GPU drops exponentially as the number of walkers increases.

To investigate the scaling of computational performance with increasing system size, simulations of multiple isolated dipeptide molecules in vacuum are performed using different methods. The results are shown in Figure 6.43. The performance for the single dipeptide molecule is better than that of the QM/MM simulation, as the solvent is absent in this setup. DFTB is among the fastest methods for small systems, but it scales comparatively poorly with system size. This setup with isolated molecules favors the MLPs because no interactions are calculated between atoms beyond their cutoff radii, whereas DFTB must calculate interactions between all atoms. However, in the limit of large, dense systems, similar

effects will occur. The MLPs scale much better with system size, with the Base MACE and QEq-MACE models being the fastest methods for larger systems as they profit from GPU support.

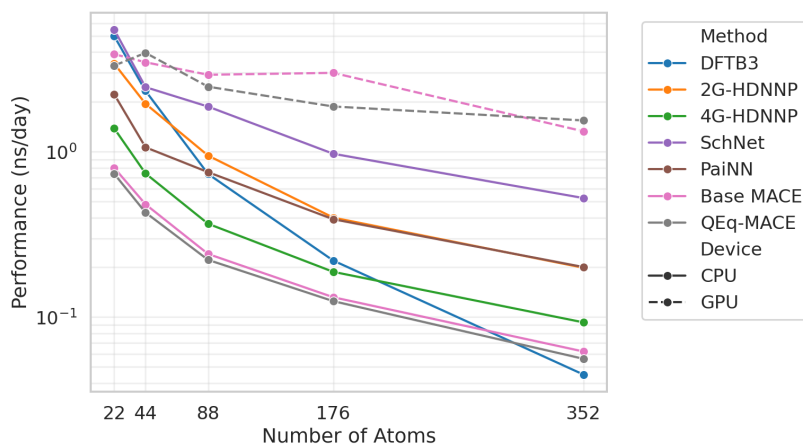


Figure 6.43.: Scaling of computational performance with increased system size for DFTB and different MLP architectures on a single CPU core with and without GPU support. DFTB is among the fastest methods for small systems, but it scales poorly with system size. The MLPs scale favorably. Base MACE and QEq-MACE, which support GPU, are the fastest methods for larger systems.

Conclusion The benchmarks show that no single method is uniformly optimal across all simulation regimes. For small QM regions and single-walker runs on CPU, DFTB, and simpler MLPs (2G-HDNNP, SchNet) provide similar throughput, while advanced MLPs such as Base MACE, QEq-MACE, and AMP require GPU acceleration to become competitive. In multi-walker metadynamics, sharing a single GPU causes a steep performance loss because walker-level parallelization is not exploited on the device, so CPU-only execution can be the more efficient practical choice once several walkers are used.

As system size increases, however, the balance shifts clearly toward MLPs: DFTB scales less favorably, whereas MLPs with GPU support (Base MACE and QEq-MACE) become the fastest options. Overall, MLPs are not universally faster at the smallest scale. Still, they offer the most favorable long-term cost profile for large condensed-phase simulations when hardware usage is matched to the sampling setup.

Part III.

Conclusions and Outlook

This thesis set out to develop, train, and validate ML/MM workflows that combine the accuracy of quantum-mechanical reference methods with the efficiency of machine-learned potentials for condensed-phase simulations. Two representative model systems were studied throughout: alanine dipeptide, a standard benchmark for conformational sampling, and thiol disulfide exchange, an enzymatic reaction in glutaredoxin proteins. The investigation was organized into five stages, each addressing a distinct aspect of the ML/MM pipeline: (i) sampling strategies for training-data generation, (ii) the accuracy of partial-charge representations for electrostatic interactions, (iii) an architecture benchmark in vacuum, (iv) direct training in explicit solvent and protein environments, and (v) Δ -learning as an alternative route to condensed-phase accuracy.

Summary

Sampling strategies Three to five data-generation methods were compared per system with respect to configurational coverage, physical plausibility, and cross-validated transferability. Metadynamics at 300 K and 500 K covered similar regions of configurational space for alanine dipeptide. In contrast, sampling with modified force-field parameters consistently expanded coverage toward low-density regions and bond-length extrema while remaining physically plausible. In cross-validation, models trained on these broader datasets generalized best across heterogeneous test distributions. The results highlight that controlled inclusion of edge-case structures is important for robust MLP behavior.

Electrostatic potential reproduction The systematic error introduced by replacing the continuous quantum-mechanical electron density with atom-centered partial charges was quantified directly on the ESP and electrostatic energies. ESP-fitting schemes and MBIS yielded the best electrostatic reproduction, while Löwdin charges failed entirely. Hirshfeld and MBIS charges emerged as practical targets for ML/MM, balancing electrostatic fidelity, learnability by a 4G-HDNNP, and chemical reasonability. The discretization error amounts to approximately 0.02 eV (0.50 kcal mol⁻¹) for alanine dipeptide and 0.01 eV (0.20 kcal mol⁻¹) for thiol disulfide exchange. These values constitute a non-negligible floor in the overall error budget of any charge-based ML/MM model and must be accounted for in performance benchmarks.

Architecture benchmark in vacuum Nine MLP architectures were trained under harmonized conditions on the same vacuum datasets. The expected performance hierarchy was confirmed, with HDNNPs at the lower end and equivariant models (QEq-MACE, Base MACE, AMP) at the upper end. A central finding was that strong static test-set metrics do not guarantee stable dynamics. For instance, PaiNN exhibited competitive test errors but failed in approximately one-third of the simulations. Conversely, AMP, QEq-MACE, and Base MACE generated stable trajectories of alanine dipeptide. For the thiol disulfide

exchange, Base MACE unexpectedly failed due to deprotonation events, whereas QEq-MACE was the only architecture to complete every simulation without failure. These results confirm that trajectory-level validation through production-length simulations is indispensable for architecture selection.

Direct training in condensed-phase environments Solvation altered both property distributions and PMF landscapes notably. The architectures specifically designed to encode environmental information exhibited a wide range in effectiveness: Field-SchNet and Field-MACE provided negligible improvement over the environment-agnostic baseline, whereas 4G-HDNNP showed moderate benefit. QEq-MACE reduced the vacuum-to-solvent error gap considerably and enabled the stable solvent simulations displaying expected behavior. Nonetheless, its performance deteriorated beyond the estimated error introduced by the point charge representation by a factor of two, indicating room for refinement in the charge-equilibration scheme. AMP stood out as the best-performing architecture in solvent: it nearly preserved its vacuum force accuracy, was the only model to reproduce a structured solvation shell in the RDF, and shows superior robustness at environment transferability. In the protein environment of the vWF C4 domain, both QEq-MACE and AMP yielded plausible reaction barriers. In agreement with each other and DFT single point calculations, they contradicted the DFTB reference PMF by predicting a barrier reduction and a more negative reaction free energy. This highlights an important advantage of ML/MM over semi-empirical baselines.

Δ -Learning Training on the difference between DFT and DFTB improved performance for weaker architectures, whereas higher-level methods did not further improve it. The vacuum and solvent errors of DFTB relative to DFT were poorly correlated, invalidating the central assumption that learning the correction in vacuum would carry over to solvent. In addition, charge-based architectures such as QEq-MACE deteriorated severely under Δ -learning due to the lack of meaningful charge targets. Overall, Δ -learning with a DFTB baseline proved not to be a viable route toward accurate ML/MM simulations, as direct training with specialized solvent-aware architectures already outperforms it without requiring a semi-empirical calculation at inference time.

Computational cost Relative to the DFTB semi-empirical reference, MLPs were not universally faster at the smallest system sizes on a single CPU core. However, they exhibited favorable scaling with system size and achieved substantial speedups through GPU acceleration. Multi-walker metadynamics revealed a practical caveat: current MLP implementations do not parallelize multiple walkers on a single GPU, so CPU-only execution can be more efficient when many walkers share a single device. Nevertheless, the overall cost profile supports the use of MLPs for long-timescale enhanced-sampling simulations of condensed-phase systems.

Taken together, the findings of this thesis demonstrate that reliable ML/MM simulations are achievable today under three conditions: sampling protocols that deliberately cover physically relevant edge regions, electrostatic representations that balance accuracy with learnability, and trajectory-level validation beyond static test-set errors. Under these conditions, modern MLPs can approach QM-level accuracy at a computational cost of semi-empirical methods, thereby enabling simulations of complex chemical processes in condensed-phase. The ultimate goal of this thesis, to reach QM accuracy at MM cost for enzymatic reactions, was therefore only partially achieved. While steps toward QM accuracy were successful, MM cost was out of reach. Only semi-empirical cost was achievable, but favorable scaling and continued advances in GPU-acceleration make it plausible that this gap will narrow in the future.

Limitations

Several limitations of the present work should be acknowledged. They either concern the generalizability of the conclusions or point to specific areas where further methodological development is needed:

Model systems Both systems studied, alanine dipeptide and thiol disulfide exchange, constitute comparatively small QM regions (<25 atoms). Whether the conclusions regarding architecture ranking and electrostatic treatment generalize to larger molecules and more diverse datasets, such as enzymatically catalyzed phosphorylation reactions, remains to be tested.

Point-charge approximation All charge-based ML/MM architectures evaluated here represent the QM region through atom-centered monopoles. The electrostatic energy error of 0.02 eV (0.5 kcal mol⁻¹) sets a systematic accuracy floor that cannot be reduced by improved training alone. Higher-order multipole representations could lower this floor, but were only partially explored through AMP.

Simulation stability Even the best-performing architecture (AMP) exhibited rare but severe force spikes leading to simulation failure in a fraction of simulations. These events, likely caused by missing neighbor configurations or extreme electrostatic conditions not present in the training data, limit unsupervised production use.

Transferability Cutoff-variation tests showed that QEq-MACE degrades rapidly under unfamiliar electrostatic conditions, while AMP is more robust. Data generation in diverse environments should be prioritized to improve transferability, but the extent to which this can be achieved without prohibitive data requirements remains an open question.

Excited states and non-adiabatic effects The present work focuses exclusively on ground-state potential energy surfaces. Extending ML/MM workflows to photochemical processes would require architectures capable of representing multiple electronic states and their couplings. Existing excited state MLPs like SchNarc²²¹, FieldSchNarc²²², or XMACE²²³ in part already provide ML/MM options.

Outlook

Higher-order electrostatic representations AMP’s success demonstrates that physically motivated multipole expansions substantially improve environment transferability. Future ML/MM architectures should move beyond monopole charge equilibration toward dipole and quadrupole terms. Another promising avenue is the PCME approach as implemented in the Electrostatical Machine Learning Embedding (EMLE) framework¹⁵⁴. This approach is expected to be generally more transferable than the EIE approach investigated here.¹⁶²

Active learning and on-the-fly retraining The rare simulation failures observed even for the best-performing models could be mitigated by uncertainty-driven active-learning loops that detect out-of-distribution configurations during production simulations and trigger targeted retraining, reducing the need for exhaustive initial sampling.

Long-timescale production dynamics The simulations performed allow for direct comparison with experimental kinetic observables. The models can be used in production simulations of enzymatic reactions in glutaredoxin, where they can be validated against experimentally measured rate constants and kinetic isotope effects, providing a test of the overall ML/MM approach beyond static benchmarks and free-energy profiles.

Larger and more diverse QM regions. Extending the present workflows to larger active sites, such as in phosphoryl transfer enzymes like creatine kinase, would test whether the architecture rankings and design principles established here transfer to chemically more demanding systems.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Marcus Elstner, for giving me the opportunity to work within the RTG2450 on this exciting topic and for his continuous guidance throughout my PhD. Taking on responsibility for the project and within the RTG made me grow as a person more than I could have imagined when I started my PhD.

I am also very grateful to my co-supervisor, Tomáš Kubař, for all the constructive discussions, helpful feedback, constant support with the requirements for computational resources, and for sharing his expertise whenever I was stuck on a problem.

I would like to thank my long-term office mate, Chris, for the many discussions on the project and physics in general. Sharing the experience and challenges helped me a lot in getting through the PhD.

I am also thankful to the other members of the generation of PhD students I started with, Katharina, Lena, Julian, David, and Alex, for the fun events we organized and enjoyed together, making the PhD experience more entertaining. The same goes for the later additions to the team: Eik, Ebru, Sonali, Quan, Sebastian, and Sani, and the most recent ones, Leonie, Maïke, Kevin, Freddy, and Pascal.

A special thanks goes to the former students I had the chance to supervise: Klara, Sebastian, Anna, Dominik, David, and Pascal. I learned a lot from the experience, and I'm glad I had the chance to pass on the knowledge to the next generation.

Everyone who helped me with the corrections and feedback on the writing of the thesis, Katharina, Freddy, Pascal, Quan, Eik, Julian, Chris, and Tomáš, I am very grateful for the time and effort you put into it, and for the helpful comments that improved the quality of the thesis.

To my friends and family, thank you for providing a safe haven to disconnect and a distraction whenever I needed, and for the support and encouragement throughout the years.

Finally, I would like to thank my girlfriend, Lin, for her patience, support, understanding, and encouragement during the last year, over the long distance and the long hours spent in the office working on the thesis. It meant a lot to me to have you by my side.

ABF	Adaptive Biasing Force
ACSF	Atom-Centered Symmetry Functions
AIM	Atoms In Molecules
AMBER	Assisted Model Building with Energy Refinement
AMP	Anisotropic Message Passing
ANI	Accurate NeurAl networkK engINe for Molecular Energies
ASE	Atomic Simulation Environment
BoB	Bag of Bonds
CENT	Charge Equilibration via neural Network Technique
CHARMM	Chemistry at HARvard Macromolecular Mechanics
CHELPG	CHarges from ELectrostatic Potentials using a Grid-based method
CI	Configuration Interaction
CIS	Configuration Interaction with Singles
CISD	Configuration Interaction with Singles and Doubles
CPU	Central Processing Unit
CV	Collective Variable
DFT	Density Functional Theory
DFTB	Density Functional Tight Binding
eABF	extended Adaptive Biasing Force
EEM	Electronegativity-Equalization Method
EIE	Environment-Integrated Embedding
ELU	Exponential Linear Unit
EMLE	Electrostatical Machine Learning Embedding
ESP	ElectroStatic Potential
FES	Free Energy Surface
GaMD	Gaussian accelerated Molecular Dynamics
GGA	Generalized Gradient Approximation
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GTO	Gaussian-Type Orbital

HDNNP	High-Dimensional Neural Network Potential
2G-HDNNP	2nd Gen. High-Dimensional Neural Network Potential
4G-HDNNP	4th Gen. High-Dimensional Neural Network Potential
HMA4n	N-terminal Heavy Metal transporting domain of the ATPase from <i>Arabidopsis thaliana</i>
HsGrx1	Human Glutaredoxin 1
IQR	InterQuartile Range
KGCNN	Keras Convolutional Graph Neural Network
LCAO	Linear Combination of Atomic Orbitals
LDA	Local Density Approximation
LYP	Lee-Yang-Parr
MACE	MACE
MAE	Mean Absolute Error
MBIS	Minimal-Basis Iterative Stockholder
MBTR	Many-Body Tensor Representation
MD	Molecular Dynamics
MIC	Minimum Image Convention
MK	Merz-Kollman
ML	Machine Learning
MLP	Machine Learning Potential
ML/MM	Machine Learning/Molecular Mechanics
MM	Molecular Mechanics
MP2	second-order Møller-Plesset perturbation theory
MSE	Mean Squared Error
NDDO	Neglect of Diatomic Differential Overlap
NequIP	Neural equivariant Interatomic Potential
NMS	Normal Mode Sampling
NN	Neural Network
OPES	On-the-fly Probability Enhanced Sampling
OPLS	Optimized Potentials for Liquid Simulations
PBE	Perdew-Burke-Ernzerhof

PCA	Principal Component Analysis
PCM	Polarizable Continuum Model
PCME	Polarizable-Corrected Mechanical Embedding
PDB	Protein Data Bank
PES	Potential Energy Surface
PME	Particle Mesh Ewald
PMF	Potential of Mean Force
PPII	Polyproline II
QEq-MACE	Charge-Equilibration MACE
QM	Quantum Mechanical
QM/MM	Quantum Mechanics/Molecular Mechanics
RDF	Radial Distribution Function
ReLU	Rectified Linear Unit
RESP	Restrained Electrostatic Potential
RMSD	Root Mean Square Deviation
RMSE	Root Mean Square Error
SCF	Self-Consistent Field
SiLU	Sigmoid Linear Unit
SOAP	Smooth Overlap of Atomic Positions
SRP	Special Reaction Parameterization
STO	Slater-Type Orbital
SWA	Stochastic Weight Averaging
tSNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
vdW	van der Waals
vWF	von Willebrand Factor
w.r.t.	With Respect To
xTB	extended Tight Binding

A. List of Figures

1.1. Time and length scales of molecular simulations	3
2.1. Maxwell-Boltzmann distributions	10
2.2. Adaptive sampling workflow	19
2.3. Neural network schema	39
2.4. ACSF representation	43
2.5. HDNNP architecture	48
2.6. Bessel basis	49
2.7. 4G-HDNNP with ESP input	54
3.1. Workflow	61
3.2. TensorFlow proof of implementation	77
3.3. PyTorch proof of implementation	78
4.1. Alanine dipeptide structure	82
4.2. Alanine dipeptide PMF	82
4.3. Alanine dipeptide sampling method comparison	84
4.4. Alanine dipeptide CV location data source	84
4.5. Alanine dipeptide hydrogen bond length distribution	85
4.6. Alanine dipeptide sampling method property comparison	86
4.7. Dipeptide 2G-HDNNP cross-validation	87
4.8. Thiol disulfide exchange reaction structure	88
4.9. Thiol disulfide exchange reaction scaled charge PMF	89
4.10. Thiol disulfide sampling method comparison	90
4.11. Thiol disulfide CV location data source	90
4.12. Thiol disulfide hydrogen bond length distribution	91
4.13. Thiol disulfide sampling method property comparison	92
4.14. Thiol disulfide 2G-HDNNP cross-validation	93
5.1. Alanine dipeptide partial charge comparison	98
5.2. Alanine dipeptide ESP difference	98
5.3. Alanine dipeptide partial charge environment effect	99

5.4. Alanine dipeptide partial charge method ESP reproduction comparison	100
5.5. Alanine dipeptide partial charge training	101
5.6. Thiol disulfide partial charge comparison	102
5.7. Thiol disulfide ESP difference	103
5.8. Thiol disulfide partial charge environment effect	103
5.9. Thiol disulfide partial charge method ESP reproduction comparison	104
6.1. Neural network publications over time	105
6.2. 2G-HDNNP hyperparameter effects	108
6.3. 4G-HDNNP hyperparameter effects	109
6.4. SchNet hyperparameter effects	110
6.5. PaiNN hyperparameter effects	111
6.6. Base MACE hyperparameter effects	112
6.7. Dipeptide dataset size dependence	113
6.8. Dipeptide vacuum model comparison	114
6.9. Dipeptide vacuum simulation walker completion	116
6.10. Dipeptide vacuum simulation reference PMF	117
6.11. Dipeptide vacuum simulation PMF comparison	118
6.12. Dipeptide vacuum simulation force difference	119
6.13. Dipeptide vacuum simulation PMF analysis	119
6.14. Thiol disulfide vacuum model comparison	122
6.15. Thiol disulfide vacuum simulation walker completion	123
6.16. Thiol disulfide vacuum simulation reference PMF	123
6.17. Thiol disulfide vacuum simulation PMF comparison	125
6.18. Thiol disulfide vacuum simulation force difference	126
6.19. Alanine dipeptide environment property comparison	128
6.20. Dipeptide DFT PMF comparison	129
6.21. Dipeptide water training model comparison	131
6.22. Dipeptide water simulation PMF comparison	133
6.23. Dipeptide water simulation force difference	134
6.24. Dipeptide water simulation PMF analysis	135
6.25. Dipeptide water simulation RDF comparison	136
6.26. Dipeptide water simulation cutoff comparison	137
6.27. Thiol disulfide potential of mean force comparison	139
6.28. Thiol disulfide environment property comparison	139
6.29. Thiol disulfide exchange reaction water training model comparison	140
6.30. Thiol disulfide water simulation walker completion	141

6.31. Thiol disulfide water simulation PMF comparison	142
6.32. Thiol disulfide water simulation force difference	143
6.33. Thiol disulfide exchange reaction in vWF C4 domain	144
6.34. Thiol disulfide protein simulation PMF comparison	146
6.35. Thiol disulfide protein simulation force difference	147
6.36. Alanine dipeptide DFT vs DFTB vacuum energy comparison	149
6.37. Alanine dipeptide DFT vs DFTB vacuum force comparison	150
6.38. Alanine dipeptide DFT vs DFTB environment effect comparison	151
6.39. Alanine dipeptide DFT vs DFTB Mulliken charge comparison	152
6.40. Alanine dipeptide Δ -learning model comparison	153
6.41. Method performance	156
6.42. Method parallelization scaling	156
6.43. Method scaling performance	157
D.1. Alanine dipeptide vacuum ESP and E_{elec} ranking	195
D.2. Reproduction electrostatic potential alanine dipeptide	196
D.3. Reproduction electrostatic energy alanine dipeptide	197
D.4. Alanine dipeptide environment ESP and E_{elec} ranking	198
D.5. Reproduction electrostatic potential alanine dipeptide	199
D.6. Reproduction electrostatic energy alanine dipeptide	200
D.7. Thiol disulfide vacuum ESP and E_{elec} ranking	201
D.8. Reproduction electrostatic potential thiol disulfide	202
D.10. Thiol disulfide environment ESP and E_{elec} ranking	204
D.11. Reproduction electrostatic potential thiol disulfide	205
D.12. Reproduction electrostatic energy thiol disulfide	206
D.13. 2G-HDNNP grid search hyperparameters	208
D.14. 2G-HDNNP training duration	208
D.15. 2G-HDNNP hyperband search hyperparameters	210
D.16. 2G-HDNNP hyperband trial ranking and duration	210
D.17. 4G-HDNNP grid search hyperparameters	212
D.18. 4G-HDNNP training duration	212
D.19. 4G-HDNNP hyperband search hyperparameters	214
D.20. 4G-HDNNP hyperband trial ranking and duration	215
D.21. SchNet grid search hyperparameters	216
D.22. SchNet training duration	217
D.23. SchNet hyperband search hyperparameters	219
D.24. SchNet hyperband trial ranking and duration	219

D.25. PaiNN grid search hyperparameters	220
D.26. PaiNN training duration	221
D.27. PaiNN hyperband search hyperparameters	222
D.28. PaiNN hyperband trial ranking and duration	223
D.29. MACE grid search hyperparameters	224
D.30. MACE training duration	224
D.31. MACE hyperband search hyperparameters	226
D.32. MACE hyperband trial ranking and duration	226
D.33. 2G-HDNNP hyperband search hyperparameters	227
D.34. 2G-HDNNP hyperband trial ranking and duration	228
D.35. 4G-HDNNP hyperband search hyperparameters	229
D.36. 4G-HDNNP hyperband trial ranking and duration	230
D.37. SchNet hyperband search hyperparameters	231
D.38. SchNet hyperband trial ranking and duration	231
D.39. PaiNN hyperband search hyperparameters	233
D.40. PaiNN hyperband trial ranking and duration	233
D.41. Base MACE hyperband search hyperparameters	234
D.42. Base MACE hyperband trial ranking and duration	235
D.43. Thiol disulfide dataset size dependence	235
D.44. Thiol disulfide DFT vs DFTB vacuum energy comparison	238
D.45. Thiol disulfide DFT vs DFTB vacuum force comparison	239
D.46. Thiol disulfide DFT vs DFTB environment effect comparison	240
D.47. Thiol disulfide Δ -learning model comparison	241

B. List of Tables

2.1. Common activation functions	39
3.1. MACE uncertainty sampling hyperparameters	64
3.2. DFT reference energies	65
3.3. DFT atomic energies	65
3.4. DFTB reference energies	66
3.5. DFTB and Δ atomic energies	66
3.6. Base MACE hyperparameters	68
3.7. FieldSchNet hyperparameters	69
3.8. FieldMACE hyperparameters	70
3.9. AMP hyperparameters	71
3.10. Gromacs environment variables	75
6.1. Alanine dipeptide vacuum test set performance	114
6.2. Dipeptide vacuum PMF extrema: reference vs. QEq-MACE	120
6.3. Thiol disulfide vacuum test set performance	121
6.4. Alanine dipeptide water training test set performance	131
6.5. Dipeptide water PMF extrema: reference vs. QEq-MACE	135
6.6. Thiol disulfide water training test set performance	140
6.7. Alanine dipeptide Δ -learning test set performance	153
D.1. 2G-HDNNP grid search space	207
D.2. 2G-HDNNP hyperband search space	209
D.3. 2G-HDNNP best hyperband configurations	209
D.4. 4G-HDNNP grid search space	211
D.5. 4G-HDNNP hyperband search space	213
D.6. 4G-HDNNP best hyperband configurations	213
D.7. SchNet grid search hyperparameter space	215
D.8. SchNet hyperband hyperparameter space	218
D.9. SchNet best hyperband configurations	218
D.10. PaiNN grid search hyperparameter space	220

D.11. PaiNN hyperband hyperparameter space	221
D.12. PaiNN best hyperband configurations	222
D.13. MACE grid search space	223
D.14. MACE hyperband search space	225
D.15. Base MACE best hyperband configurations	225
D.16. Thiol 2G-HDNNP best hyperband configurations	228
D.17. Thiol 4G-HDNNP best hyperband configurations	230
D.18. Thiol SchNet best hyperband configurations	232
D.19. Thiol PaiNN best hyperband configurations	234
D.20. Thiol Base MACE best hyperband configurations	235
D.21. Δ -training hyperparameters: 2G-HDNNP	236
D.22. Δ -training hyperparameters: 4G-HDNNP	236
D.23. Δ -training hyperparameters: MACE family	237
D.24. Thiol disulfide Δ -learning test set performance	241

C. Bibliography

- [1] John Jumper et al. „Highly accurate protein structure prediction with AlphaFold“. en. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.
- [2] John A. Keith et al. „Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems“. In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 9816–9872. DOI: 10.1021/acs.chemrev.1c00107.
- [3] Musa Ozboyaci et al. „Modeling and simulation of protein–surface interactions: achievements and challenges“. en. In: *Quarterly Reviews of Biophysics* 49 (Jan. 2016), e4. DOI: 10.1017/S0033583515000256.
- [4] A. Warshel and M. Levitt. „Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme“. In: *Journal of Molecular Biology* 103.2 (May 1976), pp. 227–249. DOI: 10.1016/0022-2836(76)90311-9.
- [5] Xiang Fu et al. „Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations“. en. In: *Transactions on Machine Learning Research* (Jan. 2023).
- [6] Juan Santiago Grassano et al. *From QM/MM to ML/MM: A new era in multiscale modeling*. en. Aug. 2025. DOI: 10.26434/chemrxiv-2025-lhwt7.
- [7] Fernando T. Ogata et al. „Glutaredoxin: Discovery, redox defense and much more“. In: *Redox Biology* 43 (July 2021), p. 101975. DOI: 10.1016/j.redox.2021.101975.
- [8] Raghunathan Ramakrishnan et al. „Big Data Meets Quantum Chemistry Approximations: The \Delta-Machine Learning Approach“. In: *Journal of Chemical Theory and Computation* 11.5 (2015), pp. 2087–2096. DOI: 10.1021/acs.jctc.5b00099.
- [9] Andrew R. Leach. *Molecular Modelling: Principles and Applications*. en. 2nd. Pearson Education, 2001.
- [10] Martin Karplus and J. Andrew McCammon. „Molecular dynamics simulations of biomolecules“. eng. In: *Nature Structural Biology* 9.9 (Sept. 2002), pp. 646–652. DOI: 10.1038/nsb0902-646.
- [11] Wendy D. Cornell et al. „A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules“. In: *Journal of the American Chemical Society* 117.19 (May 1995), pp. 5179–5197. DOI: 10.1021/ja00124a002.
- [12] A. D. MacKerell et al. „All-atom empirical potential for molecular modeling and dynamics studies of proteins“. eng. In: *The Journal of Physical Chemistry. B* 102.18 (Apr. 1998), pp. 3586–3616. DOI: 10.1021/jp973084f.
- [13] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. „Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids“. In: *Journal of the American Chemical Society* 118.45 (Nov. 1996), pp. 11225–11236. DOI: 10.1021/ja9621760.
- [14] Adri C. T. van Duin et al. „ReaxFF: A Reactive Force Field for Hydrocarbons“. In: *The Journal of Physical Chemistry A* 105.41 (Oct. 2001), pp. 9396–9409. DOI: 10.1021/jp004368u.
- [15] Jay W. Ponder et al. „Current Status of the AMOEBA Polarizable Force Field“. In: *The Journal of Physical Chemistry B* 114.8 (Mar. 2010), pp. 2549–2564. DOI: 10.1021/jp910674d.

- [16] Hans Martin Senn and Walter Thiel. „QM/MM Methods for Biomolecular Systems“. en. In: *Angewandte Chemie International Edition* 48.7 (2009). Pp. 1198–1229. DOI: 10.1002/anie.200802019.
- [17] Paulo C. T. Souza et al. „Martini 3: a general purpose force field for coarse-grained molecular dynamics“. en. In: *Nature Methods* 18.4 (Apr. 2021), pp. 382–388. DOI: 10.1038/s41592-021-01098-3.
- [18] W. G. Noid. „Perspective: Coarse-grained models for biomolecular systems“. In: *The Journal of Chemical Physics* 139.9 (Sept. 2013), p. 090901. DOI: 10.1063/1.4818908.
- [19] Loup Verlet. „Computer Experiments on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules“. In: *Physical Review* 159.1 (July 1967), pp. 98–103. DOI: 10.1103/PhysRev.159.98.
- [20] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. 1st. Boca Raton: CRC Press, 1988. DOI: 10.1201/9780367806934.
- [21] William C. Swope et al. „A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters“. In: *The Journal of Chemical Physics* 76.1 (Jan. 1982), pp. 637–649. DOI: 10.1063/1.442716.
- [22] Frank Jensen. *Introduction to Computational Chemistry*. en. John Wiley & Sons, Feb. 2017.
- [23] H. J. C. Berendsen et al. „Molecular dynamics with coupling to an external bath“. In: *The Journal of Chemical Physics* 81.8 (Oct. 1984), pp. 3684–3690. DOI: 10.1063/1.448118.
- [24] Giovanni Bussi, Davide Donadio, and Michele Parrinello. „Canonical sampling through velocity rescaling“. eng. In: *The Journal of Chemical Physics* 126.1 (Jan. 2007), p. 014101. DOI: 10.1063/1.2408420.
- [25] Shūichi Nosé. „A molecular dynamics method for simulations in the canonical ensemble“. In: *Molecular Physics* 52.2 (June 1984). Pp. 255–268. DOI: 10.1080/00268978400101201.
- [26] William G. Hoover. „Canonical dynamics: Equilibrium phase-space distributions“. In: *Physical Review A* 31.3 (Mar. 1985), pp. 1695–1697. DOI: 10.1103/PhysRevA.31.1695.
- [27] Hans C. Andersen. „Molecular dynamics simulations at constant pressure and/or temperature“. In: *The Journal of Chemical Physics* 72.4 (Feb. 1980), pp. 2384–2393. DOI: 10.1063/1.439486.
- [28] Stephen C. Harvey, Robert K.-Z. Tan, and Thomas E. Cheatham III. „The flying ice cube: Velocity rescaling in molecular dynamics leads to violation of energy equipartition“. en. In: *Journal of Computational Chemistry* 19.7 (1998). Pp. 726–740. DOI: 10.1002/(SICI)1096-987X(199805)19:7<726::AID-JCC4>3.0.CO;2-S.
- [29] M. Parrinello and A. Rahman. „Polymorphic transitions in single crystals: A new molecular dynamics method“. In: *Journal of Applied Physics* 52.12 (Dec. 1981), pp. 7182–7190. DOI: 10.1063/1.328693.
- [30] P. P. Ewald. „Die Berechnung optischer und elektrostatischer Gitterpotentiale“. en. In: *Annalen der Physik* 369.3 (1921). Pp. 253–287. DOI: 10.1002/andp.19213690304.
- [31] Tom Darden, Darrin York, and Lee Pedersen. „Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems“. In: *The Journal of Chemical Physics* 98.12 (June 1993), pp. 10089–10092. DOI: 10.1063/1.464397.
- [32] Donald A. McQuarrie. *Statistical Mechanics*. English. Sausalito, Calif: University Science Books, 2000.
- [33] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. English. San Diego: Academic Press, 2002.
- [34] G. M. Torrie and J. P. Valleau. „Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling“. In: *Journal of Computational Physics* 23.2 (Feb. 1977), pp. 187–199. DOI: 10.1016/0021-9991(77)90121-8.

- [35] Alessandro Laio and Michele Parrinello. „Escaping free-energy minima“. In: *Proceedings of the National Academy of Sciences* 99.20 (Oct. 2002), pp. 12562–12566. DOI: 10.1073/pnas.202427399.
- [36] Shankar Kumar et al. „THE weighted histogram analysis method for free-energy calculations on biomolecules. I. The method“. en. In: *Journal of Computational Chemistry* 13.8 (1992). Pp. 1011–1021. DOI: 10.1002/jcc.540130812.
- [37] Johannes Kästner. „Umbrella sampling“. en. In: *WIREs Computational Molecular Science* 1.6 (2011). Pp. 932–942. DOI: 10.1002/wcms.66.
- [38] Alessandro Barducci, Giovanni Bussi, and Michele Parrinello. „Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method“. In: *Physical Review Letters* 100.2 (Jan. 2008), p. 020603. DOI: 10.1103/PhysRevLett.100.020603.
- [39] Stefano Piana and Alessandro Laio. „A bias-exchange approach to protein folding“. eng. In: *The Journal of Physical Chemistry. B* 111.17 (May 2007), pp. 4553–4559. DOI: 10.1021/jp0678731.
- [40] Michele Invernizzi and Michele Parrinello. „Rethinking Metadynamics: From Bias Potentials to Probability Distributions“. In: *The Journal of Physical Chemistry Letters* 11.7 (Apr. 2020), pp. 2731–2736. DOI: 10.1021/acs.jpclett.0c00497.
- [41] Michele Invernizzi, Pablo M. Piaggi, and Michele Parrinello. „Unified Approach to Enhanced Sampling“. In: *Physical Review X* 10.4 (Nov. 2020), p. 041034. DOI: 10.1103/PhysRevX.10.041034.
- [42] Michele Invernizzi and Michele Parrinello. „Exploration vs Convergence Speed in Adaptive-Bias Enhanced Sampling“. In: *Journal of Chemical Theory and Computation* 18.6 (June 2022), pp. 3988–3996. DOI: 10.1021/acs.jctc.2c00152.
- [43] J. S. Smith, O. Isayev, and A. E. Roitberg. „ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost“. en. In: *Chemical Science* 8.4 (Mar. 2017), pp. 3192–3203. DOI: 10.1039/C6SC05720A.
- [44] Michael Gastegger, Jörg Behler, and Philipp Marquetand. „Machine learning molecular dynamics for the simulation of infrared spectra“. en. In: *Chemical Science* 8.10 (2017), pp. 6924–6935. DOI: 10.1039/C7SC02267K.
- [45] Lin Shen and Weitao Yang. „Molecular Dynamics Simulations with Quantum Mechanics/Molecular Mechanics and Adaptive Neural Networks“. In: *Journal of Chemical Theory and Computation* 14.3 (2018), pp. 1442–1455. DOI: 10.1021/acs.jctc.7b01195.
- [46] Justin S. Smith et al. „Less is more: Sampling chemical space with active learning“. In: *The Journal of Chemical Physics* 148.24 (May 2018), p. 241733. DOI: 10.1063/1.5023802.
- [47] Manyi Yang et al. „Using metadynamics to build neural network potentials for reactive events: the case of urea decomposition in water“. In: *Catalysis Today. 100 years of CASALE SA: a scientific perspective on catalytic processes* 387 (Mar. 2022), pp. 143–149. DOI: 10.1016/j.cattod.2021.03.018.
- [48] Tom A. Young et al. „Reaction dynamics of Diels–Alder reactions from machine learned potentials“. en. In: *Physical Chemistry Chemical Physics* 24.35 (Sept. 2022), pp. 20820–20827. DOI: 10.1039/D2CP02978B.
- [49] Maksim Kulichenko et al. „Uncertainty-driven dynamics for active learning of interatomic potentials“. en. In: *Nature Computational Science* 3.3 (Mar. 2023). Pp. 230–239. DOI: 10.1038/s43588-023-00406-5.
- [50] Cas van der Oord et al. „Hyperactive learning for data-driven interatomic potentials“. en. In: *npj Computational Materials* 9.1 (Sept. 2023), p. 168. DOI: 10.1038/s41524-023-01104-6.
- [51] Aik Rui Tan, Johannes C. B. Dietschreit, and Rafael Gómez-Bombarelli. „Enhanced sampling of robust molecular datasets with uncertainty-based collective variables“. In: *The Journal of Chemical Physics* 162.3 (Jan. 2025), p. 034114. DOI: 10.1063/5.0246178.

- [52] E. Colomés, Z. Zhan, and X. Oriols. „Comparing Wigner, Husimi and Bohmian distributions: which one is a true probability distribution in phase space?“ en. In: *Journal of Computational Electronics* 14.4 (Dec. 2015), pp. 894–906. DOI: 10.1007/s10825-015-0737-6.
- [53] Max Pinheiro Jr et al. „WS22 database, Wigner Sampling and geometry interpolation for configurationally diverse molecular datasets“. en. In: *Scientific Data* 10.1 (Feb. 2023). P. 95. DOI: 10.1038/s41597-023-01998-3.
- [54] Viktor Zaverkin et al. „Uncertainty-biased molecular dynamics for learning uniformly accurate interatomic potentials“. en. In: *npj Computational Materials* 10.1 (Apr. 2024), p. 83. DOI: 10.1038/s41524-024-01254-1.
- [55] Matthias Kellner and Michele Ceriotti. „Uncertainty quantification by direct propagation of shallow ensembles“. en. In: *Machine Learning: Science and Technology* 5.3 (July 2024), p. 035006. DOI: 10.1088/2632-2153/ad594a.
- [56] Han Xu et al. *Evidential Deep Learning for Interatomic Potentials*. Apr. 2025. DOI: 10.48550/arXiv.2407.13994.
- [57] Eric Darve and Andrew Pohorille. „Calculating free energies using average force“. In: *The Journal of Chemical Physics* 115.20 (Nov. 2001), pp. 9169–9183. DOI: 10.1063/1.1410978.
- [58] Eric Darve, Michael A. Wilson, and Andrew Pohorille. „Calculating Free Energies Using a Scaled-Force Molecular Dynamics Algorithm“. In: *Molecular Simulation* 28.1-2 (Jan. 2002). Pp. 113–144. DOI: 10.1080/08927020211975.
- [59] Adrien Lesage et al. „Smoothed Biasing Forces Yield Unbiased Free Energies with the Extended-System Adaptive Biasing Force Method“. In: *The Journal of Physical Chemistry B* 121.15 (Apr. 2017), pp. 3676–3685. DOI: 10.1021/acs.jpcc.6b10055.
- [60] Jeffrey Comer et al. „The Adaptive Biasing Force Method: Everything You Always Wanted To Know but Were Afraid To Ask“. In: *The Journal of Physical Chemistry B* 119.3 (Jan. 2015), pp. 1129–1151. DOI: 10.1021/jp506633n.
- [61] Haohao Fu et al. „Extended Adaptive Biasing Force Algorithm. An On-the-Fly Implementation for Accurate Free-Energy Calculations“. In: *Journal of Chemical Theory and Computation* 12.8 (Aug. 2016), pp. 3506–3513. DOI: 10.1021/acs.jctc.6b00447.
- [62] Yinglong Miao, Victoria A. Feher, and J. Andrew McCammon. „Gaussian Accelerated Molecular Dynamics: Unconstrained Enhanced Sampling and Free Energy Calculation“. In: *Journal of Chemical Theory and Computation* 11.8 (Aug. 2015), pp. 3584–3595. DOI: 10.1021/acs.jctc.5b00436.
- [63] Chr. Møller and M. S. Plesset. „Note on an Approximation Treatment for Many-Electron Systems“. In: *Physical Review* 46.7 (Oct. 1934), pp. 618–622. DOI: 10.1103/PhysRev.46.618.
- [64] F. Coester and H. Kümmel. „Short-range correlations in nuclear wave functions“. In: *Nuclear Physics* 17 (June 1960), pp. 477–485. DOI: 10.1016/0029-5582(60)90140-1.
- [65] Jiří Čížek. „On the Correlation Problem in Atomic and Molecular Systems. Calculation of Wavefunction Components in Ursell-Type Expansion Using Quantum-Field Theoretical Methods“. In: *The Journal of Chemical Physics* 45.11 (Dec. 1966), pp. 4256–4266. DOI: 10.1063/1.1727484.
- [66] P. Hohenberg and W. Kohn. „Inhomogeneous Electron Gas“. In: *Physical Review* 136.3B (Nov. 1964), B864–B871. DOI: 10.1103/PhysRev.136.B864.
- [67] 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.
- [68] John P. Perdew and Karla Schmidt. „Jacob’s ladder of density functional approximations for the exchange-correlation energy“. In: *AIP Conference Proceedings* 577.1 (July 2001), pp. 1–20. DOI: 10.1063/1.1390175.
- [69] 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.

- [70] John P. Perdew and Yue Wang. „Accurate and simple analytic representation of the electron-gas correlation energy“. In: *Physical Review B* 45.23 (June 1992), pp. 13244–13249. DOI: 10.1103/PhysRevB.45.13244.
- [71] A. D. Becke. „Density-functional exchange-energy approximation with correct asymptotic behavior“. In: *Physical Review A* 38.6 (Sept. 1988), pp. 3098–3100. DOI: 10.1103/PhysRevA.38.3098.
- [72] Chengteh Lee, Weitao Yang, and Robert G. Parr. „Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density“. In: *Physical Review B* 37.2 (Jan. 1988), pp. 785–789. DOI: 10.1103/PhysRevB.37.785.
- [73] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. „Generalized Gradient Approximation Made Simple“. In: *Physical Review Letters* 77.18 (Oct. 1996), pp. 3865–3868. DOI: 10.1103/PhysRevLett.77.3865.
- [74] Jianmin Tao et al. „Climbing the Density Functional Ladder: Nonempirical Meta-Generalized Gradient Approximation Designed for Molecules and Solids“. In: *Physical Review Letters* 91.14 (Sept. 2003), p. 146401. DOI: 10.1103/PhysRevLett.91.146401.
- [75] Jianwei Sun, Adrienn Ruzsinszky, and John P. Perdew. „Strongly Constrained and Appropriately Normed Semilocal Density Functional“. In: *Physical Review Letters* 115.3 (July 2015), p. 036402. DOI: 10.1103/PhysRevLett.115.036402.
- [76] Axel D. Becke. „A new mixing of Hartree–Fock and local density-functional theories“. In: *The Journal of Chemical Physics* 98.2 (Jan. 1993), pp. 1372–1377. DOI: 10.1063/1.464304.
- [77] Axel D. Becke. „Density-functional thermochemistry. III. The role of exact exchange“. In: *The Journal of Chemical Physics* 98.7 (Apr. 1993), pp. 5648–5652. DOI: 10.1063/1.464913.
- [78] P. J. Stephens et al. „Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields“. In: *The Journal of Physical Chemistry* 98.45 (Nov. 1994), pp. 11623–11627. DOI: 10.1021/j100096a001.
- [79] Carlo Adamo and Vincenzo Barone. „Toward reliable density functional methods without adjustable parameters: The PBE0 model“. In: *The Journal of Chemical Physics* 110.13 (Apr. 1999), pp. 6158–6170. DOI: 10.1063/1.478522.
- [80] Jeng-Da Chai and Martin Head-Gordon. „Systematic optimization of long-range corrected hybrid density functionals“. In: *The Journal of Chemical Physics* 128.8 (Feb. 2008), p. 084106. DOI: 10.1063/1.2834918.
- [81] Jeng-Da Chai and Martin Head-Gordon. „Long-range corrected hybrid density functionals with damped atom–atom dispersion corrections“. en. In: *Physical Chemistry Chemical Physics* 10.44 (Nov. 2008), pp. 6615–6620. DOI: 10.1039/B810189B.
- [82] Viktor N. Staroverov et al. „Comparative assessment of a new nonempirical density functional: Molecules and hydrogen-bonded complexes“. In: *The Journal of Chemical Physics* 119.23 (Dec. 2003), pp. 12129–12137. DOI: 10.1063/1.1626543.
- [83] Viktor N. Staroverov et al. „Erratum: “Comparative assessment of a new nonempirical density functional: Molecules and hydrogen-bonded complexes” [J. Chem. Phys. 119, 12129 (2003)]“. In: *The Journal of Chemical Physics* 121.22 (Dec. 2004), p. 11507. DOI: 10.1063/1.1795692.
- [84] W. J. Hehre, R. F. Stewart, and J. A. Pople. „Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals“. In: *The Journal of Chemical Physics* 51.6 (Sept. 1969), pp. 2657–2664. DOI: 10.1063/1.1672392.
- [85] Michael J. Frisch, John A. Pople, and J. Stephen Binkley. „Self-consistent molecular orbital methods 25. Supplementary functions for Gaussian basis sets“. In: *The Journal of Chemical Physics* 80.7 (Apr. 1984), pp. 3265–3269. DOI: 10.1063/1.447079.

- [86] Thom H. Dunning Jr. „Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen“. In: *The Journal of Chemical Physics* 90.2 (Jan. 1989), pp. 1007–1023. DOI: 10.1063/1.456153.
- [87] Florian Weigend and Reinhart Ahlrichs. „Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy“. en. In: *Physical Chemistry Chemical Physics* 7.18 (Aug. 2005), pp. 3297–3305. DOI: 10.1039/B508541A.
- [88] Donald Allan McQuarrie and John Douglas Simon. *Physical chemistry: a molecular approach*. Vol. 1. University science books Sausalito, CA, 1997.
- [89] Michael J. S. Dewar and Walter Thiel. „Ground states of molecules. 38. The MNDO method. Approximations and parameters“. In: *Journal of the American Chemical Society* 99.15 (June 1977), pp. 4899–4907. DOI: 10.1021/ja00457a004.
- [90] Michael J. S. Dewar et al. „Development and use of quantum mechanical molecular models. 76. AM1: a new general purpose quantum mechanical molecular model“. In: *Journal of the American Chemical Society* 107.13 (June 1985), pp. 3902–3909. DOI: 10.1021/ja00299a024.
- [91] James J. P. Stewart. „Optimization of parameters for semiempirical methods I. Method“. en. In: *Journal of Computational Chemistry* 10.2 (1989). Pp. 209–220. DOI: 10.1002/jcc.540100208.
- [92] D. Porezag et al. „Construction of tight-binding-like potentials on the basis of density-functional theory: Application to carbon“. In: *Physical Review B* 51.19 (May 1995), pp. 12947–12957. DOI: 10.1103/PhysRevB.51.12947.
- [93] M. Elstner et al. „Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties“. en. In: *Physical Review B* 58.11 (Sept. 1998), pp. 7260–7268. DOI: 10.1103/PhysRevB.58.7260.
- [94] M. Elstner. „SCC-DFTB: What Is the Proper Degree of Self-Consistency?“ In: *The Journal of Physical Chemistry A* 111.26 (July 2007), pp. 5614–5621. DOI: 10.1021/jp071338j.
- [95] Michael Gaus, Qiang Cui, and Marcus Elstner. „DFTB3: Extension of the Self-Consistent-Charge Density-Functional Tight-Binding Method (SCC-DFTB)“. In: *Journal of Chemical Theory and Computation* 7.4 (Apr. 2011), pp. 931–948. DOI: 10.1021/ct100684s.
- [96] Steve Kaminski et al. „Extended Polarization in Third-Order SCC-DFTB from Chemical-Potential Equalization“. en. In: *Journal of Physical Chemistry A* 116.36 (Sept. 2012), pp. 9131–9141. DOI: 10.1021/jp306239c.
- [97] 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 (May 2017), pp. 1989–2009. DOI: 10.1021/acs.jctc.7b00118.
- [98] 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.
- [99] Claudia Leticia Gómez Flores. „Machine Learning methodology for the study of a disulfide exchange reaction“. English. Dissertation. Karlsruhe: Karlsruhe Institut for Technology, May 2022.
- [100] Julian Böser. „Multiscale Modeling of Protein Systems Integrating QM/MM Simulation, Enhanced Sampling, and Machine Learning“. de. PhD thesis. 2026.
- [101] Michael Gaus, Albrecht Goetz, and Marcus Elstner. „Parametrization and Benchmark of DFTB3 for Organic Molecules“. In: *Journal of Chemical Theory and Computation* 9.1 (Jan. 2013), pp. 338–354. DOI: 10.1021/ct300849w.

- [102] Michael Gaus et al. „Parameterization of DFTB3/3OB for Sulfur and Phosphorus for Chemical and Biological Applications“. In: *Journal of Chemical Theory and Computation* 10.4 (Apr. 2014), pp. 1518–1537. DOI: 10.1021/ct401002w.
- [103] Maximilian Kubillus et al. „Parameterization of the DFTB3 method for Br, Ca, Cl, F, I, K, and Na in organic and biological systems“. eng. In: *Journal of Chemical Theory and Computation* 11.1 (Jan. 2015), pp. 332–342. DOI: 10.1021/ct5009137.
- [104] Xiya Lu et al. „Parametrization of DFTB3/3OB for Magnesium and Zinc for Chemical and Biological Applications“. In: *The Journal of Physical Chemistry B* 119.3 (Jan. 2015), pp. 1062–1082. DOI: 10.1021/jp506557r.
- [105] R. S. Mulliken. „Electronic Population Analysis on LCAO–MO Molecular Wave Functions. I“. In: *The Journal of Chemical Physics* 23.10 (Oct. 1955), pp. 1833–1840. DOI: 10.1063/1.1740588.
- [106] Per-Olov Löwdin. „On the Non-Orthogonality Problem Connected with the Use of Atomic Wave Functions in the Theory of Molecules and Crystals“. In: *The Journal of Chemical Physics* 18.3 (Mar. 1950), pp. 365–375. DOI: 10.1063/1.1747632.
- [107] F. L. Hirshfeld. „Bonded-atom fragments for describing molecular charge densities“. en. In: *Theoretica chimica acta* 44.2 (June 1977), pp. 129–138. DOI: 10.1007/BF00549096.
- [108] Toon Verstraelen et al. „Minimal Basis Iterative Stockholder: Atoms in Molecules for Force-Field Development“. In: *Journal of Chemical Theory and Computation* 12.8 (Aug. 2016), pp. 3894–3912. DOI: 10.1021/acs.jctc.6b00456.
- [109] U. Chandra Singh and Peter A. Kollman. „An approach to computing electrostatic charges for molecules“. en. In: *Journal of Computational Chemistry* 5.2 (1984). Pp. 129–145. DOI: 10.1002/jcc.540050204.
- [110] Brent H. Besler, Kenneth M. Merz Jr., and Peter A. Kollman. „Atomic charges derived from semiempirical methods“. en. In: *Journal of Computational Chemistry* 11.4 (1990). Pp. 431–439. DOI: 10.1002/jcc.540110404.
- [111] Curt M. Breneman and Kenneth B. Wiberg. „Determining atom-centered monopoles from molecular electrostatic potentials. The need for high sampling density in formamide conformational analysis“. en. In: *Journal of Computational Chemistry* 11.3 (1990). Pp. 361–373. DOI: 10.1002/jcc.540110311.
- [112] Christopher I. Bayly et al. „A well-behaved electrostatic potential based method using charge restraints for deriving atomic charges: the RESP model“. In: *The Journal of Physical Chemistry* 97.40 (Oct. 1993), pp. 10269–10280. DOI: 10.1021/j100142a004.
- [113] Wilfried J. Mortier, Karin Van Genechten, and Johann Gasteiger. „Electronegativity equalization: application and parametrization“. In: *Journal of the American Chemical Society* 107.4 (Feb. 1985), pp. 829–835. DOI: 10.1021/ja00290a017.
- [114] Zuzana Jirousková et al. „Electronegativity equalization method: parameterization and validation for organic molecules using the Merz-Kollman-Singh charge distribution scheme“. eng. In: *Journal of computational chemistry* 30.7 (May 2009), pp. 1174–1178. DOI: 10.1002/jcc.21142.
- [115] Tian Lu and Feiwu Chen. „Multiwfn: A multifunctional wavefunction analyzer“. en. In: *Journal of Computational Chemistry* 33.5 (2012). Pp. 580–592. DOI: 10.1002/jcc.22885.
- [116] Dirk Bakowies and Walter Thiel. „Hybrid Models for Combined Quantum Mechanical and Molecular Mechanical Approaches“. In: *The Journal of Physical Chemistry* 100.25 (Jan. 1996), pp. 10580–10594. DOI: 10.1021/jp9536514.
- [117] Jiali Gao and Donald G. Truhlar. „QUANTUM MECHANICAL METHODS FOR ENZYME KINETICS“. en. In: *Annual Review of Physical Chemistry* 53. Volume 53, 2002 (Oct. 2002), pp. 467–505. DOI: 10.1146/annurev.physchem.53.091301.150114.
- [118] Lin Shen, Jingheng Wu, and Weitao Yang. „Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks“. In: *Journal of Chemical Theory and Computation* 12.10 (2016), pp. 4934–4946. DOI: 10.1021/acs.jctc.6b00663.

- [119] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.
- [120] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. *Searching for Activation Functions*. Oct. 2017. DOI: 10.48550/arXiv.1710.05941.
- [121] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. „Learning representations by back-propagating errors“. en. In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0.
- [122] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 2017. DOI: 10.48550/arXiv.1412.6980.
- [123] Jörg Behler and Michele Parrinello. „Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces“. In: *Physical Review Letters* 98.14 (2007), p. 146401. DOI: 10.1103/PhysRevLett.98.146401.
- [124] Kristof T. Schütt et al. „SchNet: A continuous-filter convolutional neural network for modeling quantum interactions“. In: (Dec. 2017). DOI: 10.48550/arXiv.1706.08566.
- [125] Linfeng Zhang et al. „End-to-end Symmetry Preserving Inter-atomic Potential Energy Model for Finite and Extended Systems“. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.
- [126] Oliver T. Unke and Markus Meuwly. „PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges“. In: *Journal of Chemical Theory and Computation* 15.6 (June 2019), pp. 3678–3693. DOI: 10.1021/acs.jctc.9b00181.
- [127] Johannes Gastegger, Florian Becker, and Stephan Günnemann. *GemNet: Universal Directional Graph Neural Networks for Molecules*. June 2024. DOI: 10.48550/arXiv.2106.08903.
- [128] Tsz Wai Ko et al. „A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer“. In: *Nature Communications* 12.1 (2021), p. 398. DOI: 10.1038/s41467-020-20427-2.
- [129] Kristof T. Schütt, Oliver T. Unke, and Michael Gastegger. *Equivariant message passing for the prediction of tensorial properties and molecular spectra*. June 2021. DOI: 10.48550/arXiv.2102.03150.
- [130] Simon Batzner et al. „E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials“. en. In: *Nature Communications* 13.1 (May 2022), p. 2453. DOI: 10.1038/s41467-022-29939-5.
- [131] Ilyes Batatia et al. „MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields“. en. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 11423–11436.
- [132] Matthias Rupp et al. „Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning“. In: *Physical Review Letters* 108.5 (Jan. 2012), p. 058301. DOI: 10.1103/PhysRevLett.108.058301.
- [133] 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 (June 2015), pp. 2326–2331. DOI: 10.1021/acs.jpclett.5b00831.
- [134] Haoyan Huo and Matthias Rupp. *Unified Representation of Molecules and Crystals for Machine Learning*. Jan. 2018. DOI: 10.48550/arXiv.1704.06439.
- [135] Marcel F. Langer, Alex Goeßmann, and Matthias Rupp. „Representations of molecules and materials for interpolation of quantum-mechanical simulations via machine learning“. en. In: *npj Computational Materials* 8.1 (Mar. 2022), pp. 1–14. DOI: 10.1038/s41524-022-00721-x.
- [136] Albert P. Bartók, Risi Kondor, and Gábor Csányi. „On representing chemical environments“. en. In: *Physical Review B* 87.18 (May 2013), p. 184115. DOI: 10.1103/PhysRevB.87.184115.
- [137] Jörg Behler. „Four Generations of High-Dimensional Neural Network Potentials“. In: *Chemical Reviews* 121.16 (Aug. 2021), pp. 10037–10072. DOI: 10.1021/acs.chemrev.0c00868.

- [138] Patrick Reiser et al. „Graph neural networks for materials science and chemistry“. en. In: *Communications Materials* 3.1 (Nov. 2022). Pp. 1–18. DOI: 10.1038/s43246-022-00315-6.
- [139] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. *Directional Message Passing for Molecular Graphs*. Apr. 2022. DOI: 10.48550/arXiv.2003.03123.
- [140] Thomas B. Blank et al. „Neural network models of potential energy surfaces“. In: *The Journal of Chemical Physics* 103.10 (1995), pp. 4129–4137. DOI: 10.1063/1.469597.
- [141] Nongnuch Artrith, Tobias Morawietz, and Jörg Behler. „High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide“. In: *Physical Review B* 83.15 (2011), p. 153101. DOI: 10.1103/PhysRevB.83.153101.
- [142] Somayeh Faraji et al. „High accuracy and transferability of a neural network potential through charge equilibration for calcium fluoride“. In: *Physical Review B* 95.10 (2017), p. 104105. DOI: 10.1103/PhysRevB.95.104105.
- [143] S. Alireza Ghasemi et al. „Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network“. In: *Physical Review B* 92.4 (July 2015), p. 045131. DOI: 10.1103/PhysRevB.92.045131.
- [144] Patrick Reiser, Andre Eberhard, and Pascal Friederich. „Implementing graph neural networks with TensorFlow-Keras“. In: *Software Impacts* 9 (Aug. 2021). P. 100095. DOI: 10.1016/j.simpa.2021.100095.
- [145] David Peter Kovacs et al. „Evaluation of the MACE Force Field Architecture: from Medicinal Chemistry to Materials Science“. In: *The Journal of Chemical Physics* 159.4 (July 2023). P. 044118. DOI: 10.1063/5.0155322.
- [146] Yinuo Yang et al. *Machine Learning of Reactive Potentials*. en. Sept. 2023. DOI: 10.26434/chemrxiv-2023-x82fz.
- [147] Timothée Devergne et al. „Combining Machine Learning Approaches and Accurate Ab Initio Enhanced Sampling Methods for Prebiotic Chemical Reactions in Solution“. eng. In: *Journal of Chemical Theory and Computation* 18.9 (Sept. 2022), pp. 5410–5421. DOI: 10.1021/acs.jctc.2c00400.
- [148] Lennard Bösel, Moritz Thürlmann, and Sereina Riniker. „Machine Learning in QM/MM Molecular Dynamics Simulations of Condensed-Phase Systems“. In: *Journal of Chemical Theory and Computation* 17.5 (May 2021), pp. 2641–2658. DOI: 10.1021/acs.jctc.0c01112.
- [149] Adil Kabylda et al. „Molecular Simulations with a Pretrained Neural Network and Universal Pairwise Force Fields“. In: *Journal of the American Chemical Society* 147.37 (Sept. 2025), pp. 33723–33734. DOI: 10.1021/jacs.5c09558.
- [150] Kai Töpfer, Silvan Käser, and Markus Meuwly. „Double proton transfer in hydrated formic acid dimer: Interplay of spatial symmetry and solvent-generated force on reactivity“. en. In: *Physical Chemistry Chemical Physics* 24.22 (June 2022), pp. 13869–13882. DOI: 10.1039/D2CP01583H.
- [151] Boyi Zhou, Yanzi Zhou, and Daiqian Xie. „Accelerated Quantum Mechanics/Molecular Mechanics Simulations via Neural Networks Incorporated with Mechanical Embedding Scheme“. In: *Journal of Chemical Theory and Computation* 19.4 (Feb. 2023), pp. 1157–1169. DOI: 10.1021/acs.jctc.2c01131.
- [152] João Morado et al. „Does a Machine-Learned Potential Perform Better Than an Optimally Tuned Traditional Force Field? A Case Study on Fluorohydrins“. In: *Journal of Chemical Information and Modeling* 63.9 (May 2023), pp. 2810–2827. DOI: 10.1021/acs.jcim.2c01510.
- [153] Kirill Zinovjev. „Electrostatic Embedding of Machine Learning Potentials“. In: *Journal of Chemical Theory and Computation* 19.6 (Mar. 2023), pp. 1888–1897. DOI: 10.1021/acs.jctc.2c00914.
- [154] Kirill Zinovjev et al. „emle-engine: A Flexible Electrostatic Machine Learning Embedding Package for Multiscale Molecular Dynamics Simulations“. In: *Journal of Chemical Theory and Computation* 20.11 (June 2024), pp. 4514–4522. DOI: 10.1021/acs.jctc.4c00248.

- [155] Michael Gastegger, Kristof T. Schütt, and Klaus-Robert Müller. „Machine learning of solvent effects on molecular spectra and reactions“. en. In: *Chemical Science* 12.34 (Sept. 2021), pp. 11473–11483. DOI: 10.1039/D1SC02742E.
- [156] Felix Pultar et al. „Neural Network Potential with Multiresolution Approach Enables Accurate Prediction of Reaction Free Energies in Solution“. In: *Journal of the American Chemical Society* (Feb. 2025). DOI: 10.1021/jacs.4c17015.
- [157] Moritz Thürlemann and Sereina Riniker. „Anisotropic Message Passing: Graph Neural Networks with Directional and Long-Range Interactions“. en. In: Sept. 2022.
- [158] Moritz Thürlemann, Lennard Bösel, and Sereina Riniker. „Learning Atomic Multipoles: Prediction of the Electrostatic Potential with Equivariant Graph Neural Networks“. In: *Journal of Chemical Theory and Computation* 18.3 (Mar. 2022). Pp. 1701–1710. DOI: 10.1021/acs.jctc.1c01021.
- [159] Rhyen Barrett, Johannes C. B. Dietschreit, and Julia Westermayr. *Incorporating Long-Range Interactions via the Multipole Expansion into Ground and Excited-State Molecular Simulations*. Feb. 2025. DOI: 10.48550/arXiv.2502.21045.
- [160] Bettina Lier et al. „BuRNN: Buffer Region Neural Network Approach for Polarizable-Embedding Neural Network/Molecular Mechanics Simulations“. In: *Journal of Physical Chemistry Letters* 13.17 (May 2022), pp. 3812–3818. DOI: 10.1021/acs.jpclett.2c00654.
- [161] Yao-Kun Lei, Kiyoshi Yagi, and Yuji Sugita. „Efficient Training of Neural Network Potentials for Chemical and Enzymatic Reactions by Continual Learning“. In: *Journal of Chemical Theory and Computation* 21.5 (Mar. 2025), pp. 2695–2711. DOI: 10.1021/acs.jctc.4c01393.
- [162] Jonathan A. Semelak et al. *Advancing Multiscale Molecular Modeling with Machine Learning-Derived Electrostatics*. en. Feb. 2025. DOI: 10.26434/chemrxiv-2024-m9xgc-v3.
- [163] Kristof T. Schütt et al. „SchNetPack 2.0: A neural network toolbox for atomistic machine learning“. In: *The Journal of Chemical Physics* 158.14 (Apr. 2023), p. 144801. DOI: 10.1063/5.0138367.
- [164] Kresten Lindorff-Larsen et al. „Improved side-chain torsion potentials for the Amber ff99SB protein force field“. In: *Proteins* 78.8 (June 2010), pp. 1950–1958. DOI: 10.1002/prot.22711.
- [165] William L. Jorgensen et al. „Comparison of simple potential functions for simulating liquid water“. In: *The Journal of Chemical Physics* 79.2 (July 1983), pp. 926–935. DOI: 10.1063/1.445869.
- [166] Denis Mario Maag. „Enhanced sampling methods applied to chemical reactions in proteins“. de. PhD thesis. 2022. DOI: 10.5445/IR/1000146741.
- [167] Frank Neese et al. „The ORCA quantum chemistry program package“. In: *The Journal of Chemical Physics* 152.22 (June 2020), p. 224108. DOI: 10.1063/5.0004608.
- [168] Rui P. P. Neves et al. „Benchmarking of Density Functionals for the Accurate Description of Thiol–Disulfide Exchange“. In: *Journal of Chemical Theory and Computation* 10.11 (Nov. 2014), pp. 4842–4856. DOI: 10.1021/ct500840f.
- [169] Tom O’Malley et al. *KerasTuner*. 2019.
- [170] Takuya Akiba et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. July 2019. DOI: 10.48550/arXiv.1907.10902.
- [171] Xiao Dong et al. „Electronegativity and chemical hardness of elements under pressure“. In: *Proceedings of the National Academy of Sciences* 119.10 (Mar. 2022), e2117416119. DOI: 10.1073/pnas.2117416119.
- [172] Beatriz Cordero et al. „Covalent radii revisited“. en. In: *Dalton Transactions* 21 (May 2008), pp. 2832–2838. DOI: 10.1039/B801115J.

- [173] Mark James Abraham et al. „GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers“. In: *SoftwareX* 1-2 (Sept. 2015), pp. 19–25. DOI: 10.1016/j.softx.2015.06.001.
- [174] T. Kubař, M. Elstner, and Q. Cui. „Hybrid Quantum Mechanical/Molecular Mechanical Methods For Studying Energy Transduction in Biomolecular Machines“. en. In: *Annual Review of Biophysics* 52. Volume 52, 2023 (May 2023), pp. 525–551. DOI: 10.1146/annurev-biophys-111622-091140.
- [175] Tomáš Kubař. *tomaskubar/gromacs-dftbplus*. Nov. 2025.
- [176] Gareth A. Tribello et al. „PLUMED 2: New feathers for an old bird“. In: *Computer Physics Communications* 185.2 (Feb. 2014), pp. 604–613. DOI: 10.1016/j.cpc.2013.09.018.
- [177] Giovanni Bussi and Gareth A. Tribello. „Analyzing and Biasing Simulations with PLUMED“. eng. In: *Methods in Molecular Biology (Clifton, N.J.)* 2022 (2019), pp. 529–578. DOI: 10.1007/978-1-4939-9608-7_21.
- [178] B. Hourahine et al. „DFTB+, a software package for efficient approximate density functional theory based atomistic simulations“. In: *The Journal of Chemical Physics* 152.12 (Mar. 2020), p. 124101. DOI: 10.1063/1.5143190.
- [179] B. Hourahine et al. „Recent Developments in DFTB+, a Software Package for Efficient Atomistic Quantum Mechanical Simulations“. In: *The Journal of Physical Chemistry A* 129.24 (June 2025), pp. 5373–5390. DOI: 10.1021/acs.jpca.5c01146.
- [180] Tian Lu. „A comprehensive electron wavefunction analysis toolbox for chemists, Multiwfn“. In: *The Journal of Chemical Physics* 161.8 (Aug. 2024), p. 082503. DOI: 10.1063/5.0216272.
- [181] Martín Abadi et al. *TensorFlow: A system for large-scale machine learning*. May 2016. DOI: 10.48550/arXiv.1605.08695.
- [182] François Chollet et al. *Keras*. 2015.
- [183] Adam Paszke et al. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Dec. 2019. DOI: 10.48550/arXiv.1912.01703.
- [184] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Mar. 2019. DOI: 10.5281/zenodo.3828935.
- [185] Ask Hjorth Larsen et al. „The atomic simulation environment—a Python library for working with atoms“. en. In: *Journal of Physics: Condensed Matter* 29.27 (June 2017), p. 273002. DOI: 10.1088/1361-648X/aa680e.
- [186] Naveen Michaud-Agrawal et al. „MDAnalysis: A toolkit for the analysis of molecular dynamics simulations“. en. In: *Journal of Computational Chemistry* 32.10 (2011). Pp. 2319–2327. DOI: 10.1002/jcc.21787.
- [187] Richard J. Gowers et al. „MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations“. en. In: *SciPy 2016* (June 2016). DOI: 10.25080/Majora-629e541a-00e.
- [188] Lukas Biewald. *Experiment Tracking with Weights and Biases*. 2020.
- [189] Ole Tange. *GNU Parallel*. Apr. 2025.
- [190] Peter J. Rossky and Martin Karplus. „Solvation. A molecular dynamics study of a dipeptide in water“. In: *Journal of the American Chemical Society* 101.8 (Apr. 1979), pp. 1913–1937. DOI: 10.1021/ja00502a001.
- [191] Douglas J. Tobias and Charles L. III Brooks. „Conformational equilibrium in the alanine dipeptide in the gas phase and aqueous solution: a comparison of theoretical results“. In: *The Journal of Physical Chemistry* 96.9 (Apr. 1992), pp. 3864–3870. DOI: 10.1021/j100188a054.
- [192] Håkan W. Hugosson et al. „A comparative theoretical study of dipeptide solvation in water“. en. In: *Journal of Computational Chemistry* 27.5 (2006). Pp. 672–684. DOI: 10.1002/jcc.20360.

- [193] Gustavo de M. Seabra, Ross C. Walker, and Adrian E. Roitberg. „Are Current Semiempirical Methods Better Than Force Fields? A Study from the Thermodynamics Perspective“. In: *The Journal of Physical Chemistry A* 113.43 (Oct. 2009), pp. 11938–11948. DOI: 10.1021/jp903474v.
- [194] Zhiwei Liu, Bernd Ensing, and Preston B. Moore. „Quantitative Assessment of Force Fields on Both Low-Energy Conformational Basins and Transition-State Regions of the (ϕ - ψ) Space“. eng. In: *Journal of Chemical Theory and Computation* 7.2 (Feb. 2011), pp. 402–419. DOI: 10.1021/ct100395n.
- [195] Victor L. Cruz, Javier Ramos, and Javier Martinez-Salazar. „Assessment of the Intrinsic Conformational Preferences of Dipeptide Amino Acids in Aqueous Solution by Combined Umbrella Sampling/MBAR Statistics. A Comparison with Experimental Results“. In: *The Journal of Physical Chemistry B* 116.1 (Jan. 2012), pp. 469–475. DOI: 10.1021/jp206757j.
- [196] Alexandar T. Tzanov, Michel A. Cuendet, and Mark E. Tuckerman. „How Accurately Do Current Force Fields Predict Experimental Peptide Conformations? An Adiabatic Free Energy Dynamics Study“. In: *The Journal of Physical Chemistry B* 118.24 (June 2014), pp. 6539–6552. DOI: 10.1021/jp500193w.
- [197] G. N. Ramachandran, C. Ramakrishnan, and V. Sasisekharan. „Stereochemistry of polypeptide chain configurations“. In: *Journal of Molecular Biology* 7.1 (July 1963), pp. 95–99. DOI: 10.1016/S0022-2836(63)80023-6.
- [198] Anders S. Christensen et al. „Semiempirical Quantum Mechanical Methods for Noncovalent Interactions for Chemical and Biochemical Applications“. In: *Chemical Reviews* 116.9 (May 2016), pp. 5301–5337. DOI: 10.1021/acs.chemrev.5b00584.
- [199] Yonathan Lissanu Deribe, Tony Pawson, and Ivan Dikic. „Post-translational modifications in signal integration“. en. In: *Nature Structural & Molecular Biology* 17.6 (June 2010), pp. 666–672. DOI: 10.1038/nsmb.1842.
- [200] Deborah Fass and Colin Thorpe. „Chemistry and Enzymology of Disulfide Cross-Linking in Proteins“. In: *Chemical Reviews* 118.3 (Feb. 2018), pp. 1169–1198. DOI: 10.1021/acs.chemrev.7b00123.
- [201] Christoph Wiedemann et al. „Cysteines and Disulfide Bonds as Structure-Forming Units: Insights From Different Domains of Life and the Potential for Characterization by NMR“. In: *Frontiers in Chemistry* 8 (Apr. 2020), p. 280. DOI: 10.3389/fchem.2020.00280.
- [202] Diego Butera and Philip J. Hogg. „Fibrinogen function achieved through multiple covalent states“. en. In: *Nature Communications* 11.1 (Oct. 2020), p. 5468. DOI: 10.1038/s41467-020-19295-7.
- [203] Rodney A. Bednar. „Reactivity and pH dependence of thiol conjugation to N-ethylmaleimide: detection of a conformational change in chalcone isomerase“. In: *Biochemistry* 29.15 (Apr. 1990), pp. 3684–3690. DOI: 10.1021/bi00467a014.
- [204] Pedro Alexandrino Fernandes and Maria João Ramos. „Theoretical Insights into the Mechanism for Thiol/Disulfide Exchange“. en. In: *Chemistry – A European Journal* 10.1 (2004). Pp. 257–266. DOI: 10.1002/chem.200305343.
- [205] Robert D. Bach, Olga Dmitrenko, and Colin Thorpe. „Mechanism of Thiolate-Disulfide Interchange Reactions in Biochemistry“. In: *The Journal of Organic Chemistry* 73.1 (Jan. 2008), pp. 12–21. DOI: 10.1021/jo702051f.
- [206] Marina Putzu et al. „On the mechanism of spontaneous thiol–disulfide exchange in proteins“. en. In: *Physical Chemistry Chemical Physics* 20.23 (June 2018), pp. 16222–16230. DOI: 10.1039/C8CP01325J.
- [207] Michael P. Washburn and William W. Wells. „The Catalytic Mechanism of the Glutathione-Dependent Dehydroascorbate Reductase Activity of Thioldtransferase (Glutaredoxin),“ in: *Biochemistry* 38.1 (Jan. 1999), pp. 268–274. DOI: 10.1021/bi980480v.
- [208] Julian Böser et al. „Reduction pathway of glutaredoxin 1 investigated with QM/MM molecular dynamics using a neural network correction“. In: *The Journal of Chemical Physics* 157.15 (Oct. 2022), p. 154104. DOI: 10.1063/5.0123089.

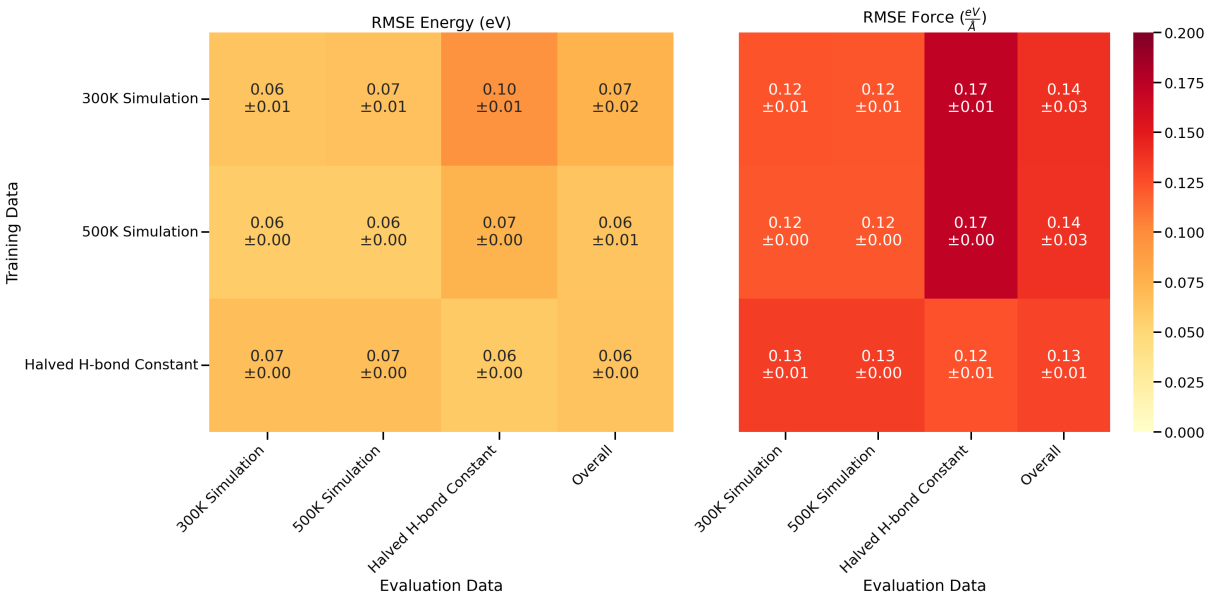
- [209] M. Born. „Volumen und Hydratationswärme der Ionen“. de. In: *Zeitschrift für Physik* 1.1 (Feb. 1920), pp. 45–48. DOI: 10.1007/BF01881023.
- [210] W. Clark Still et al. „Semianalytical treatment of solvation for molecular mechanics and dynamics“. In: *Journal of the American Chemical Society* 112.16 (Aug. 1990), pp. 6127–6129. DOI: 10.1021/ja00172a038.
- [211] Hai Nguyen, Daniel R. Roe, and Carlos Simmerling. „Improved Generalized Born Solvent Model Parameters for Protein Simulations“. In: *Journal of Chemical Theory and Computation* 9.4 (Apr. 2013), pp. 2020–2034. DOI: 10.1021/ct3010485.
- [212] Claudia L. Gómez-Flores et al. „Accurate Free Energies for Complex Condensed-Phase Reactions Using an Artificial Neural Network Corrected DFTB/MM Methodology“. In: *Journal of Chemical Theory and Computation* 18.2 (2022), pp. 1213–1226. DOI: 10.1021/acs.jctc.1c00811.
- [213] Vishal Annasaheb Adhav and Kayarat Saikrishnan. „The Realm of Unconventional Noncovalent Interactions in Proteins: Their Significance in Structure and Function“. In: *ACS Omega* 8.25 (June 2023), pp. 22268–22284. DOI: 10.1021/acsomega.3c00205.
- [214] Dennis R. Livesay et al. „Conservation of Electrostatic Properties within Enzyme Families and Superfamilies“. In: *Biochemistry* 42.12 (Apr. 2003), pp. 3464–3473. DOI: 10.1021/bi026918f.
- [215] Andrea Grisafi et al. „Transferable Machine-Learning Model of the Electron Density“. In: *ACS Central Science* 5.1 (Jan. 2019), pp. 57–64. DOI: 10.1021/acscentsci.8b00551.
- [216] Taoyuze Lv et al. „Deep Charge: Deep learning model of electron density from a one-shot density functional theory calculation“. In: *Physical Review B* 108.23 (Dec. 2023), p. 235159. DOI: 10.1103/PhysRevB.108.235159.
- [217] Patricia Suriana et al. *Beyond Atoms: Evaluating Electron Density Representation for 3D Molecular Learning*. Nov. 2025. DOI: 10.48550/arXiv.2511.21900.
- [218] Lisha Li et al. „Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization“. In: *Journal of Machine Learning Research* 18.185 (2018), pp. 1–52.
- [219] Justin S. Smith et al. „The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules“. In: *Scientific Data* 7 (May 2020), p. 134. DOI: 10.1038/s41597-020-0473-z.
- [220] Dávid Péter Kovács et al. „MACE-OFF: Short-Range Transferable Machine Learning Force Fields for Organic Molecules“. In: *Journal of the American Chemical Society* 147.21 (May 2025), pp. 17598–17611. DOI: 10.1021/jacs.4c07099.
- [221] Julia Westermayr, Michael Gastegger, and Philipp Marquetand. „Combining SchNet and SHARC: The SchNarc Machine Learning Approach for Excited-State Dynamics“. In: *The Journal of Physical Chemistry Letters* 11.10 (May 2020), pp. 3828–3834. DOI: 10.1021/acs.jpcllett.0c00527.
- [222] Maximilian X. Tiefenbacher et al. *Excited-state nonadiabatic dynamics in explicit solvent using machine learned interatomic potentials*. Jan. 2025. DOI: 10.48550/arXiv.2501.16974.
- [223] Rhyan Barrett, Christoph Ortner, and Julia Westermayr. *Transferable Machine Learning Potential X-MACE for Excited States using Integrated DeepSets*. Feb. 2025. DOI: 10.48550/arXiv.2502.12870.

D. Appendix

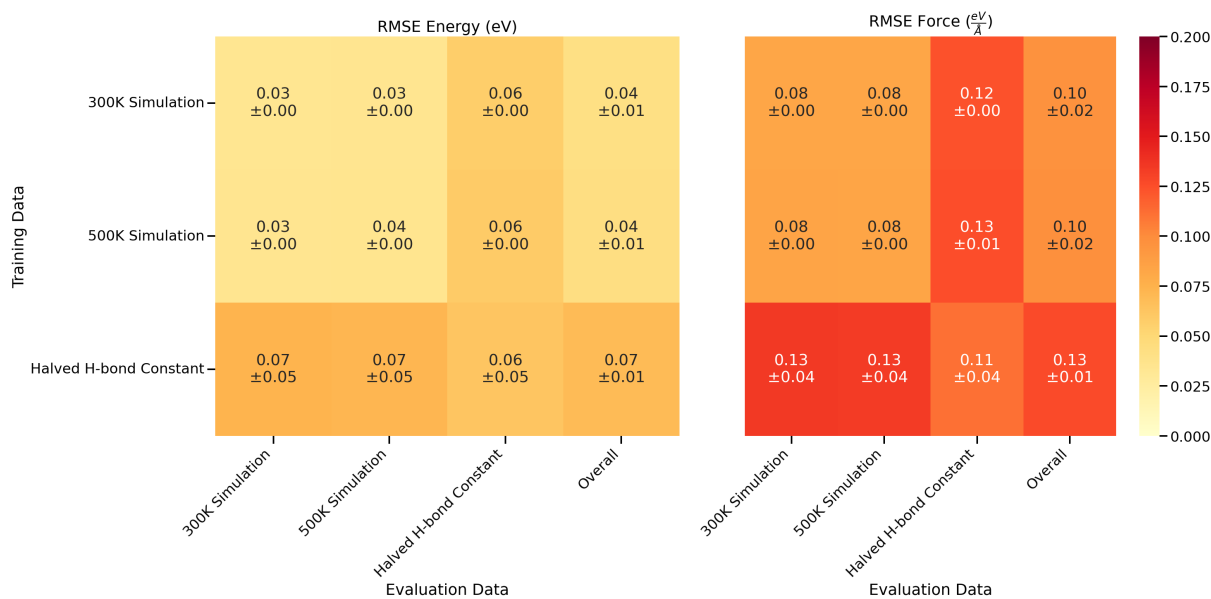
D.1. Sampling Methods

This section contains additional visualizations for the electrostatic potential reproduction discussed in Chapter 4.

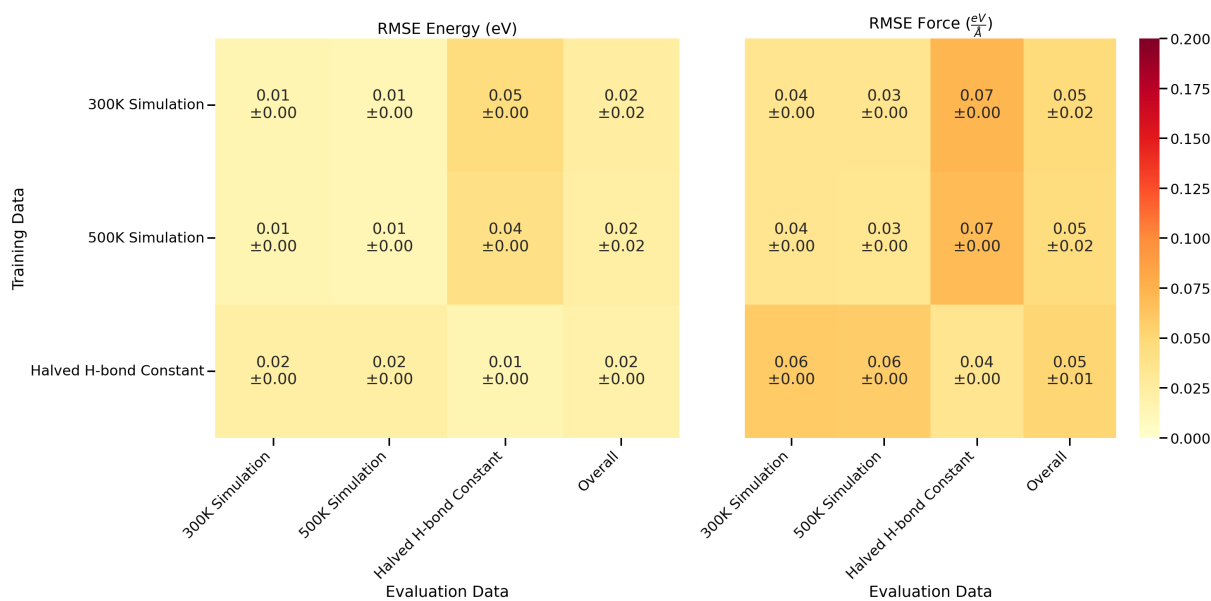
Alanine Dipeptide — Vacuum



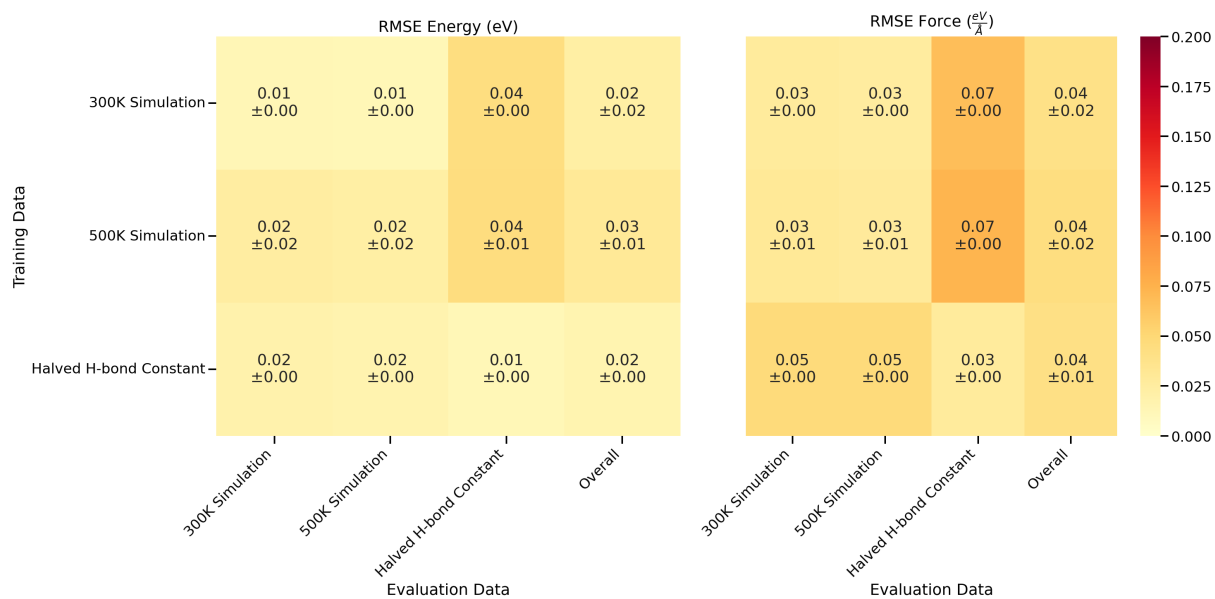
Cross-validation for 2G-HDNNP models trained on different dataset sources of alanine dipeptide in vacuum.



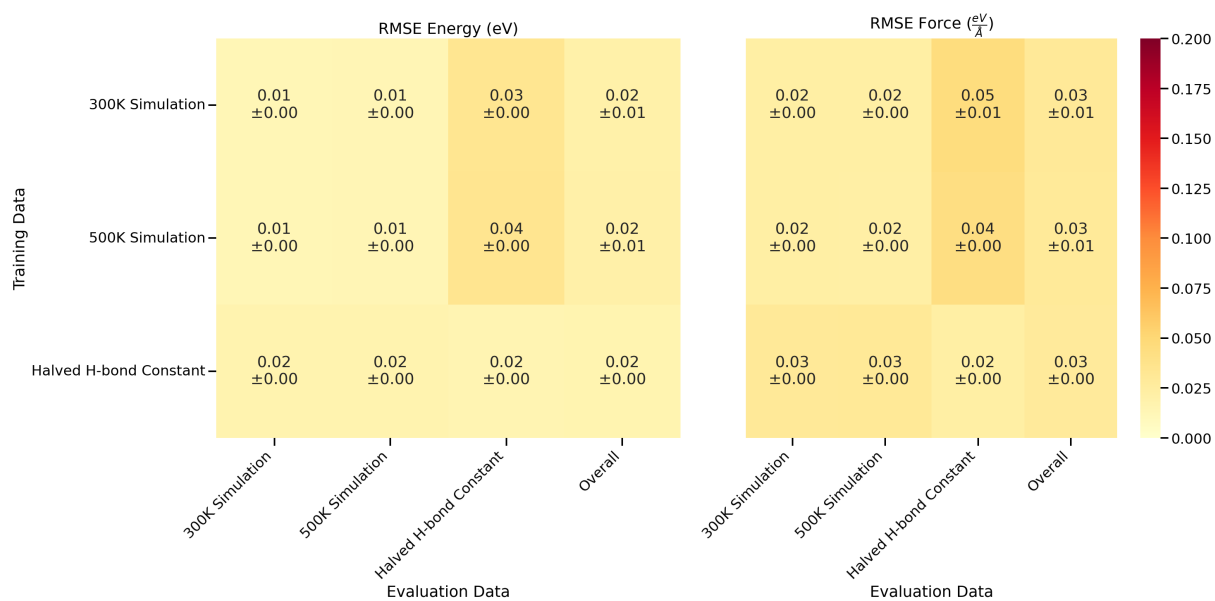
Cross-validation for 4G-HDNNP models trained on different dataset sources of alanine dipeptide in vacuum.



Cross-validation for SchNet models trained on different dataset sources of alanine dipeptide in vacuum.

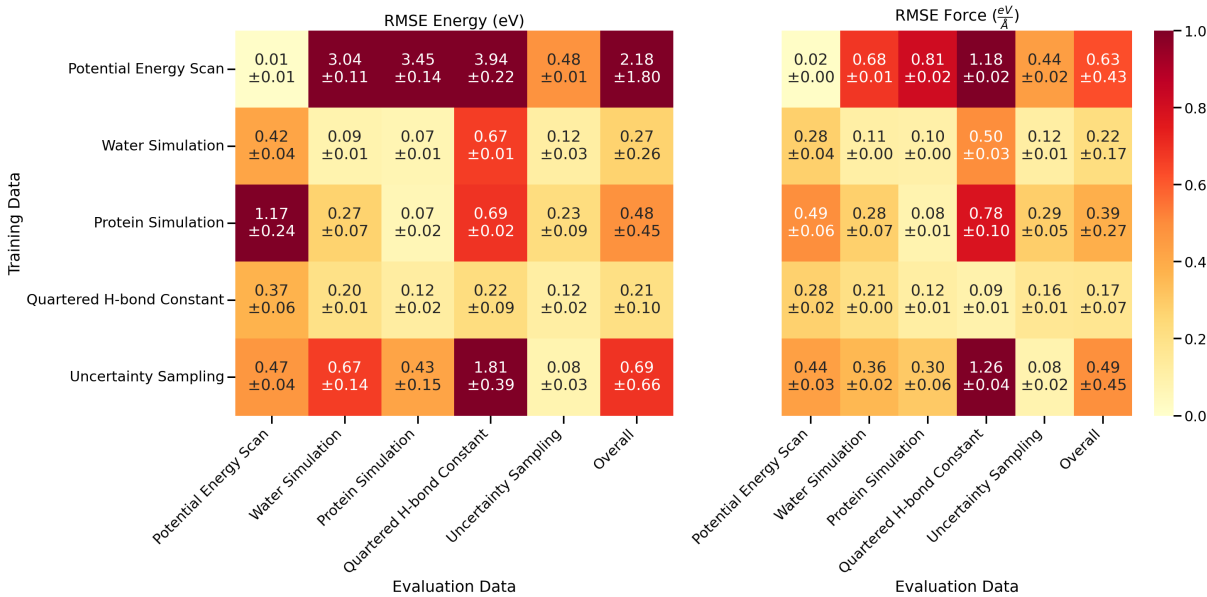


Cross-validation for PaiNN models trained on different dataset sources of alanine dipeptide in vacuum.

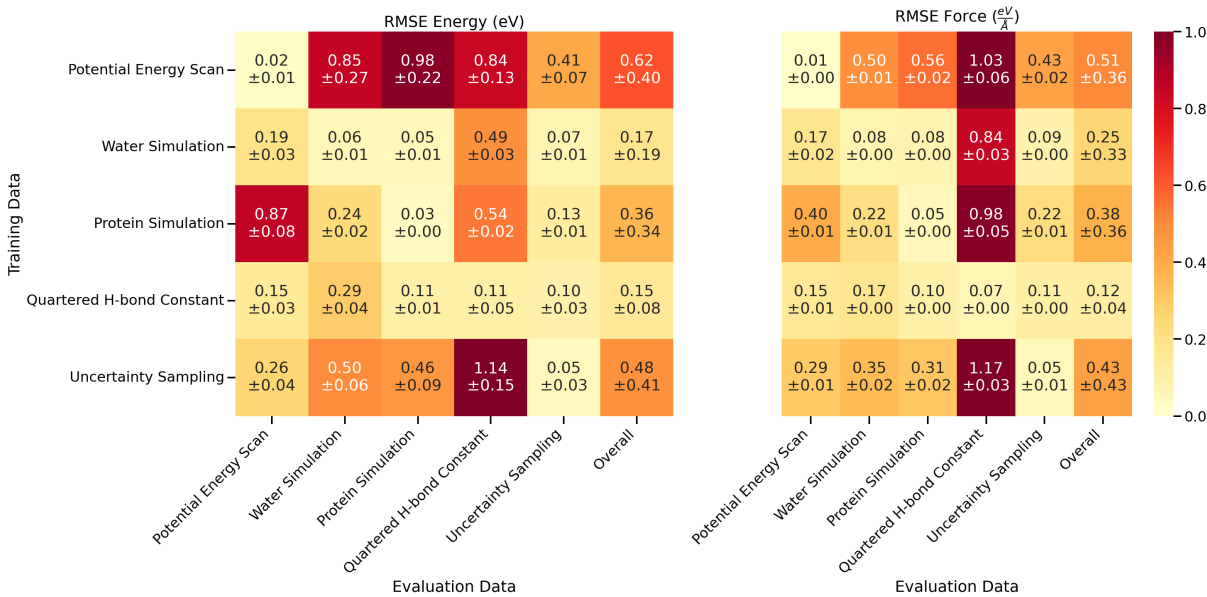


Cross-validation for QEq-MACE models trained on different dataset sources of alanine dipeptide in vacuum.

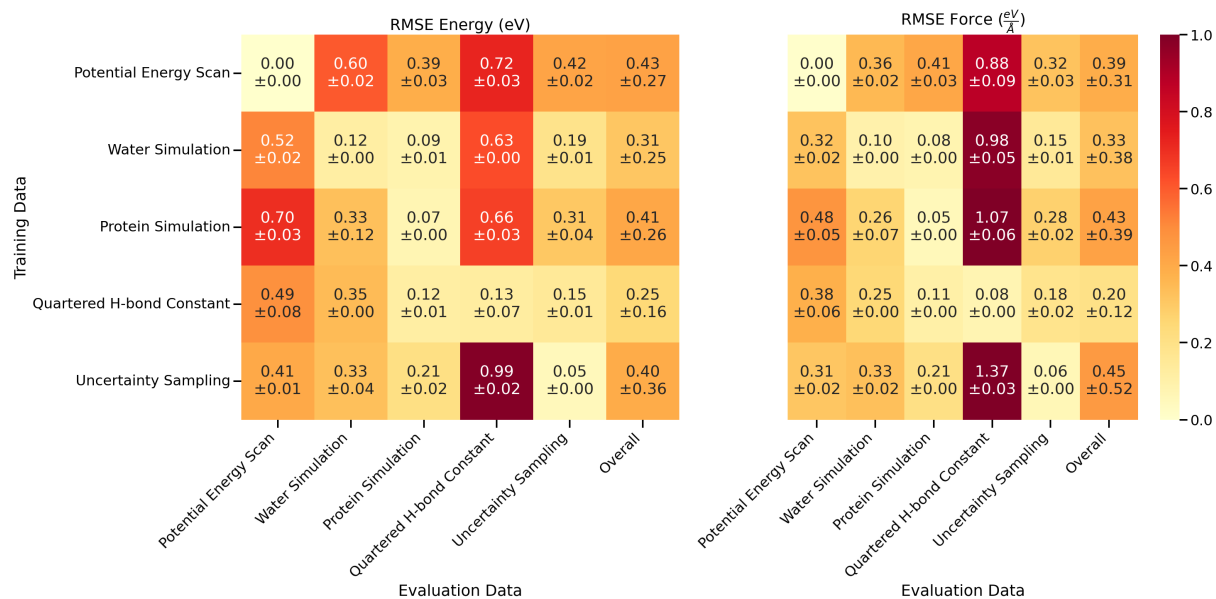
Thiol Disulfide Exchange Reaction — Vacuum



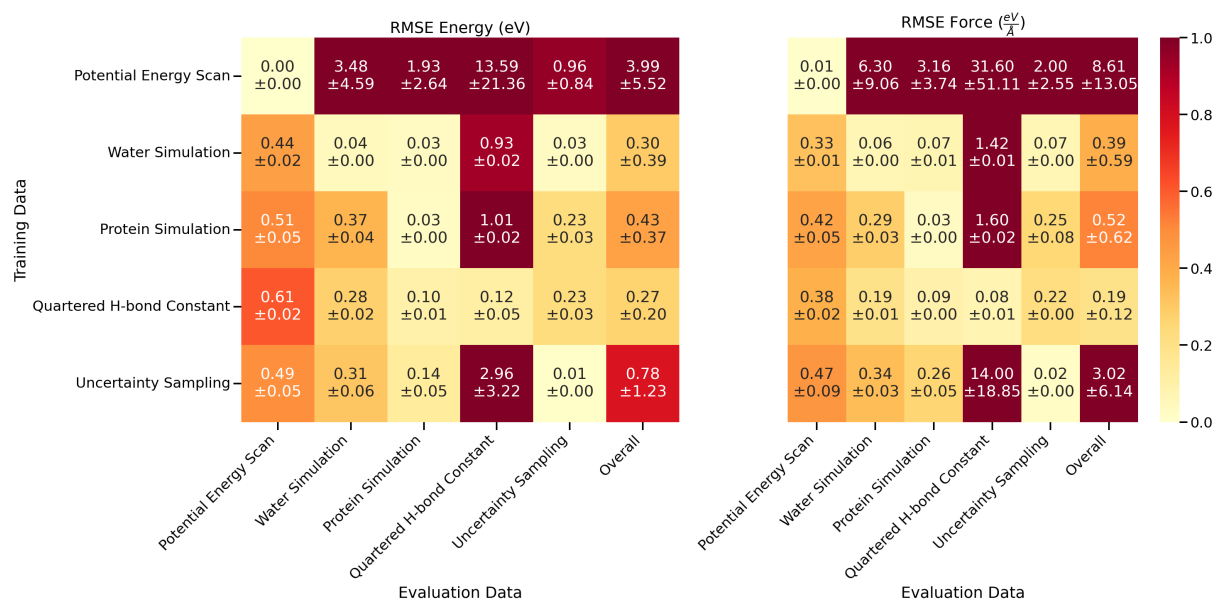
Cross-validation for 2G-HDNNP models trained on different dataset sources of the thiol disulfide exchange reaction in vacuum.



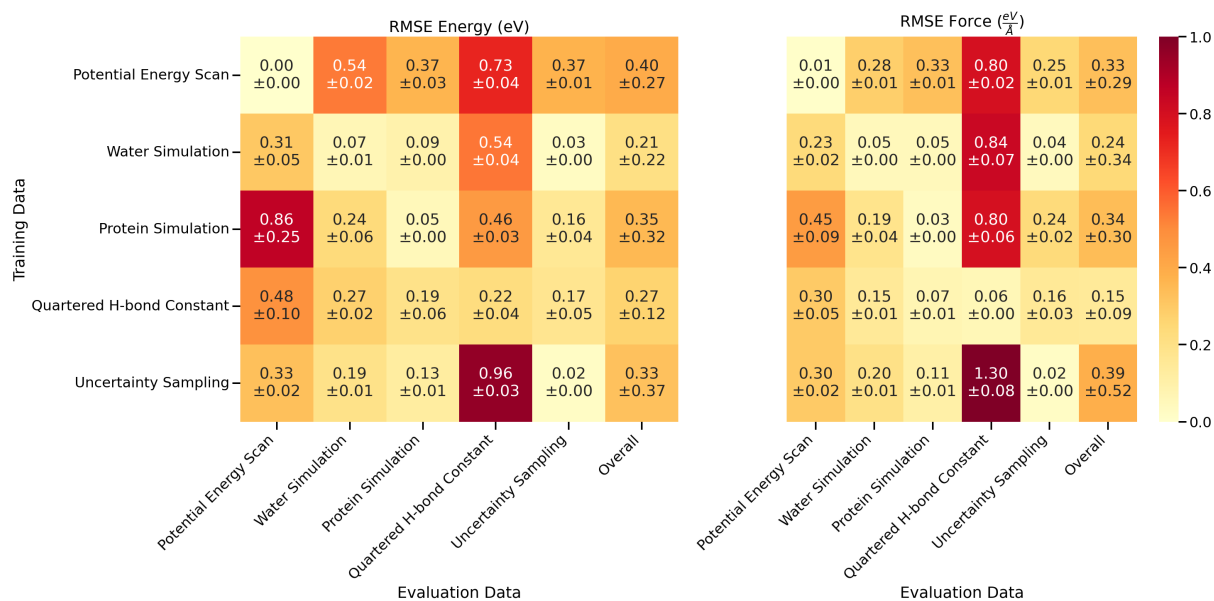
Cross-validation for 4G-HDNNP models trained on different dataset sources of thiol disulfide exchange reaction in vacuum.



Cross-validation for SchNet models trained on different dataset sources of the thiol disulfide exchange reaction in vacuum.



Cross-validation for PaiNN models trained on different dataset sources of the thiol disulfide exchange reaction in vacuum.



Cross-validation for QEq-MACE models trained on different dataset sources of the thiol disulfide exchange reaction in vacuum.

D.2. Electrostatic Potential Reproduction

This section contains additional visualizations for the electrostatic potential reproduction discussed in Section 5.2.3 and Section 5.3.3.

Alanine Dipeptide — Vacuum

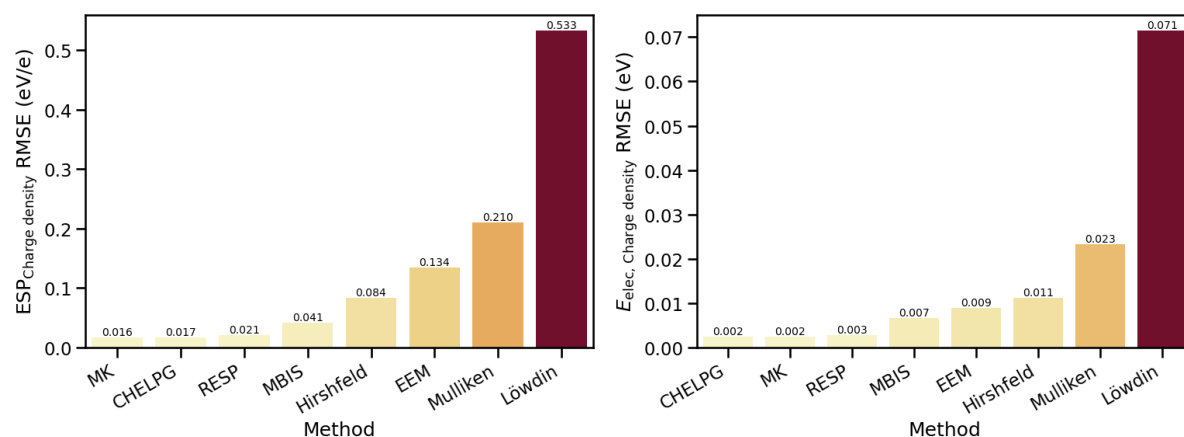


Figure D.1.: Ranking of charge assignment methods for alanine dipeptide in vacuum based on RMSE of electrostatic potential (left) and electrostatic energy (right) reproduction at the positions of the MM point charges in respective environment calculations

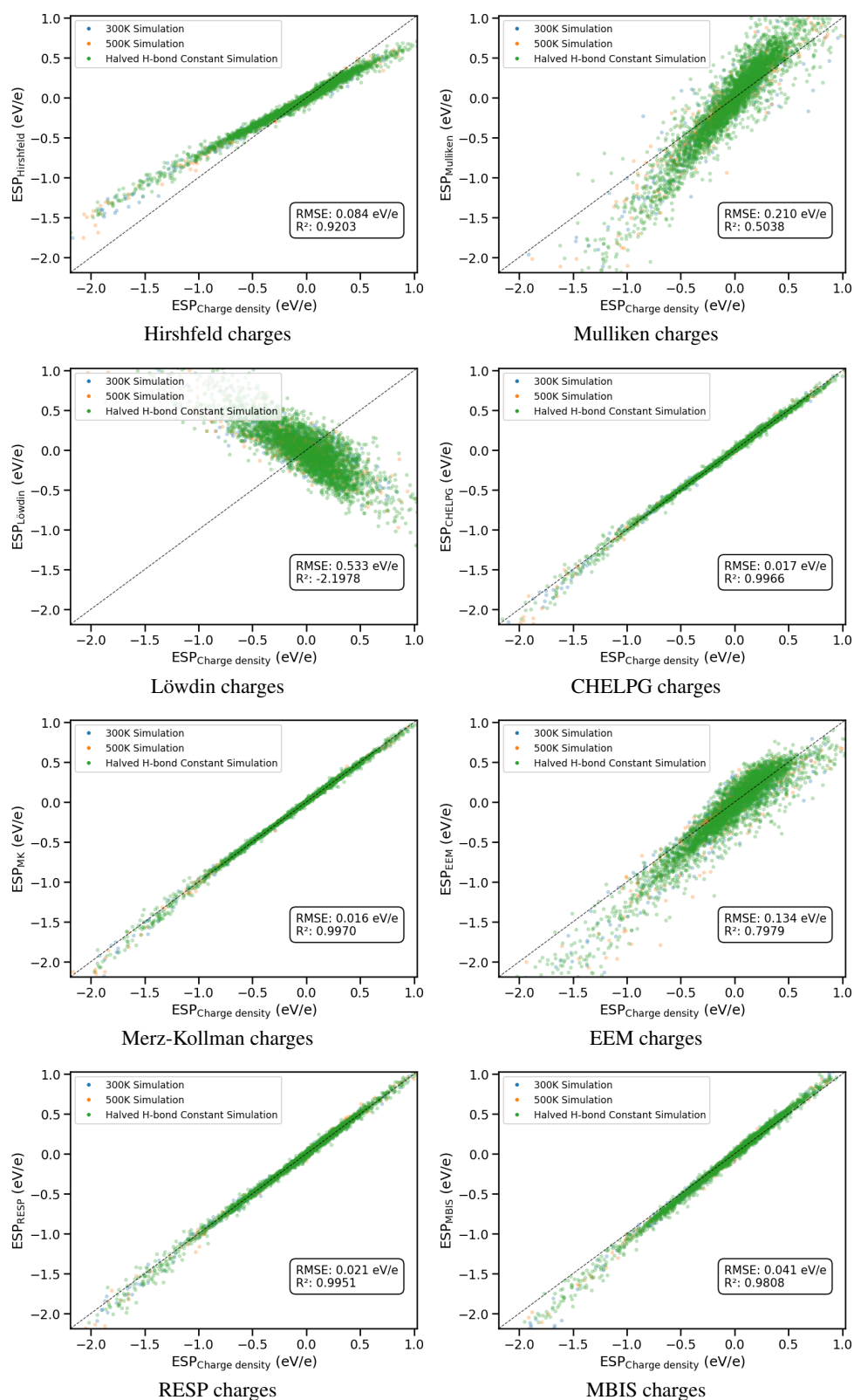


Figure D.2.: Reproduction of electrostatic potential caused by continuous charge distribution of alanine dipeptide in vacuum at the positions of the MM point charges in respective environment calculations

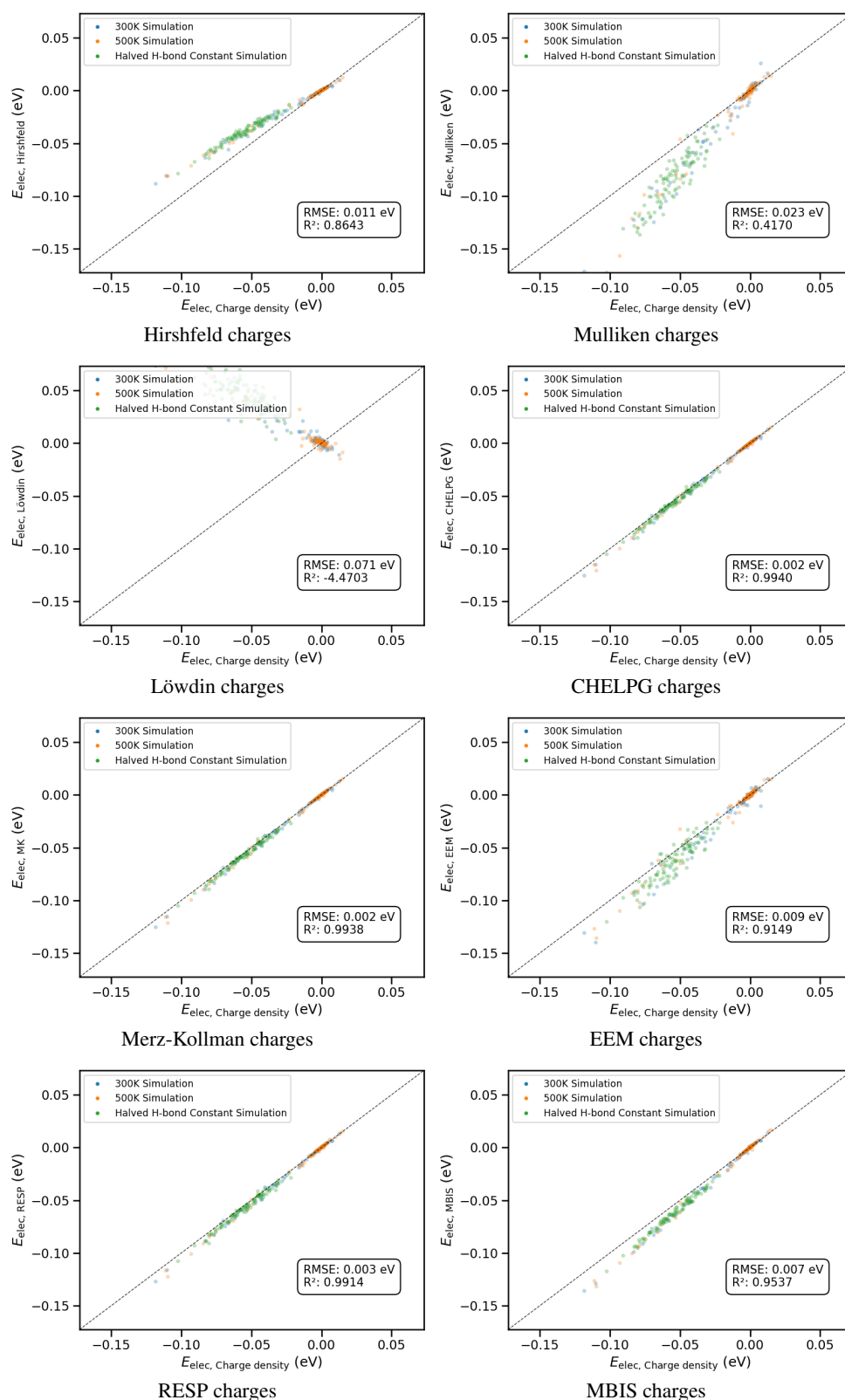


Figure D.3.: Reproduction of electrostatic energy caused by continuous charge distribution of alanine dipeptide in vacuum at the positions of the MM point charges in respective environment calculations

Alanine Dipeptide — Environment

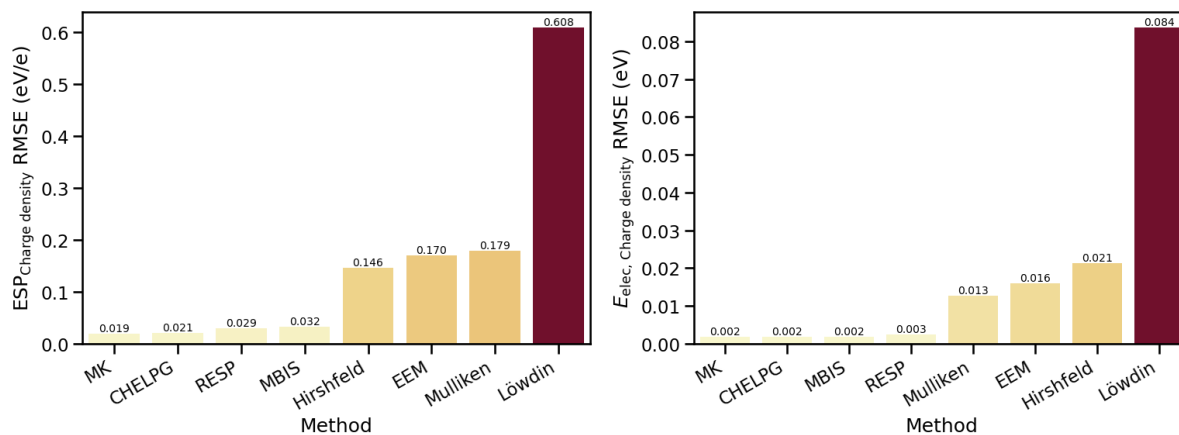


Figure D.4.: Ranking of charge assignment methods for alanine dipeptide in an environment based on RMSE of electrostatic potential (left) and electrostatic energy (right) reproduction at the positions of the MM point charges

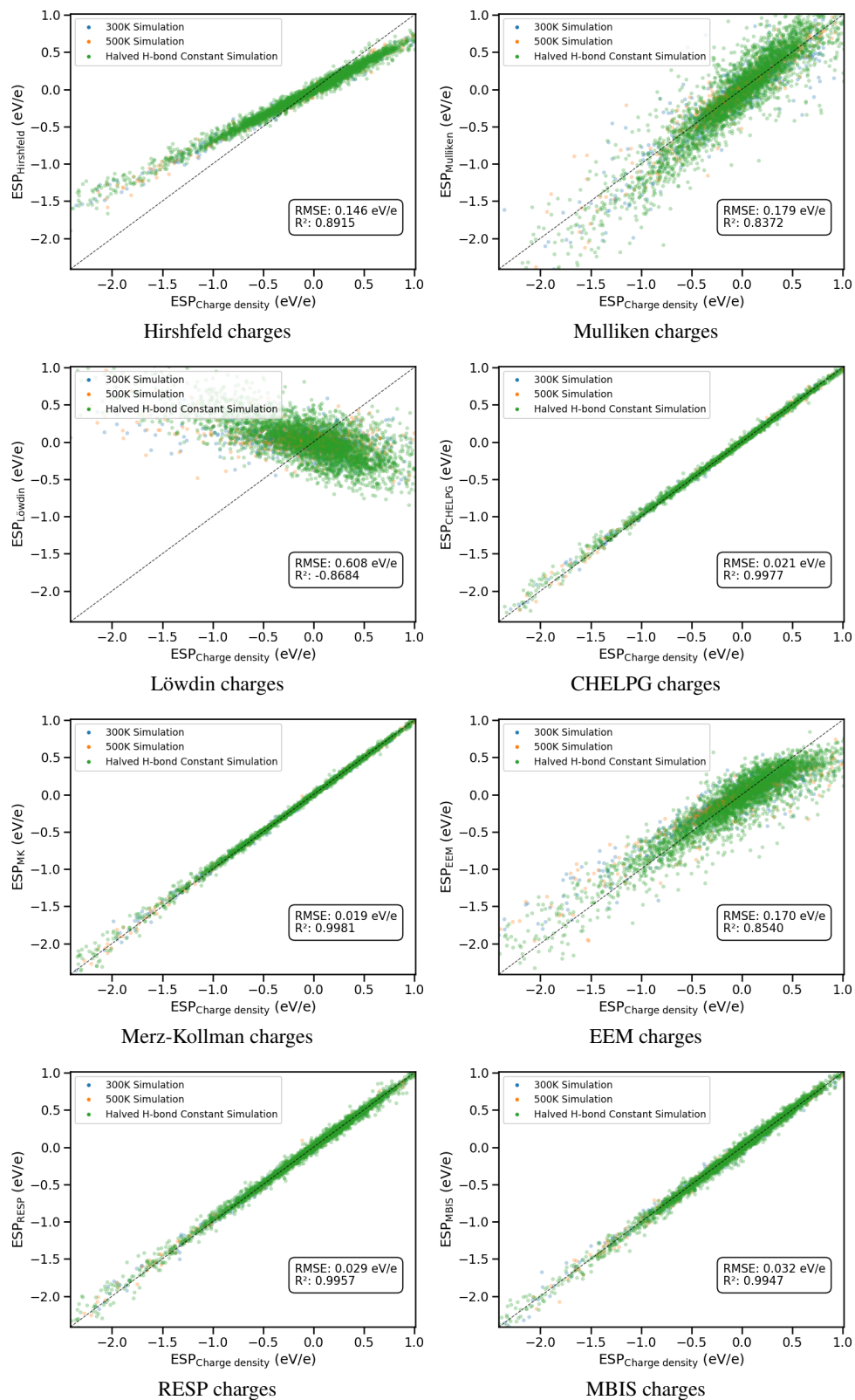


Figure D.5.: Reproduction of electrostatic potential caused by continuous charge distribution of alanine dipeptide in an environment at the positions of the MM point charges

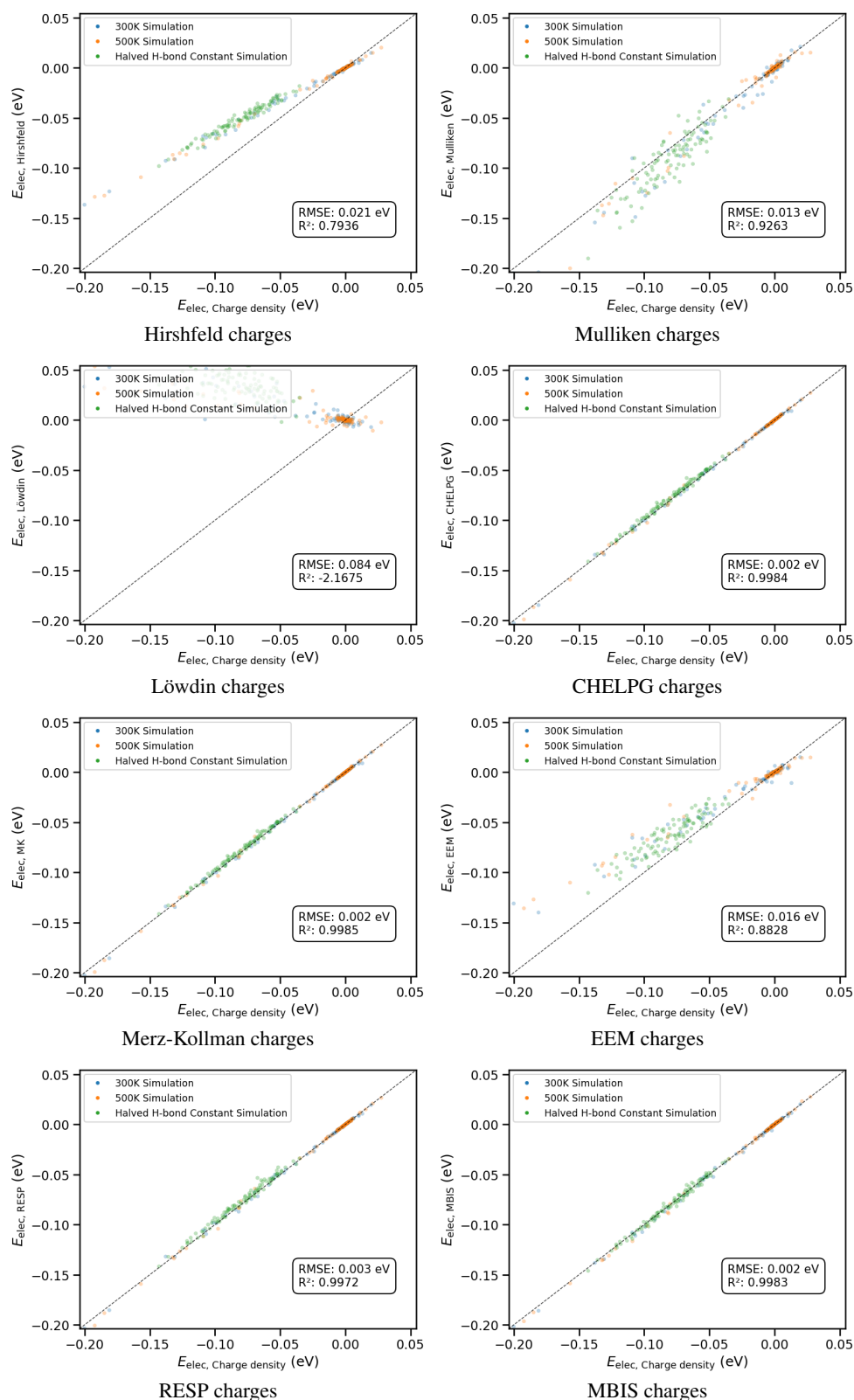


Figure D.6.: Reproduction of electrostatic energy caused by continuous charge distribution of alanine dipeptide in an environment at the positions of the MM point charges

Thiol Disulfide Exchange Reaction — Vacuum

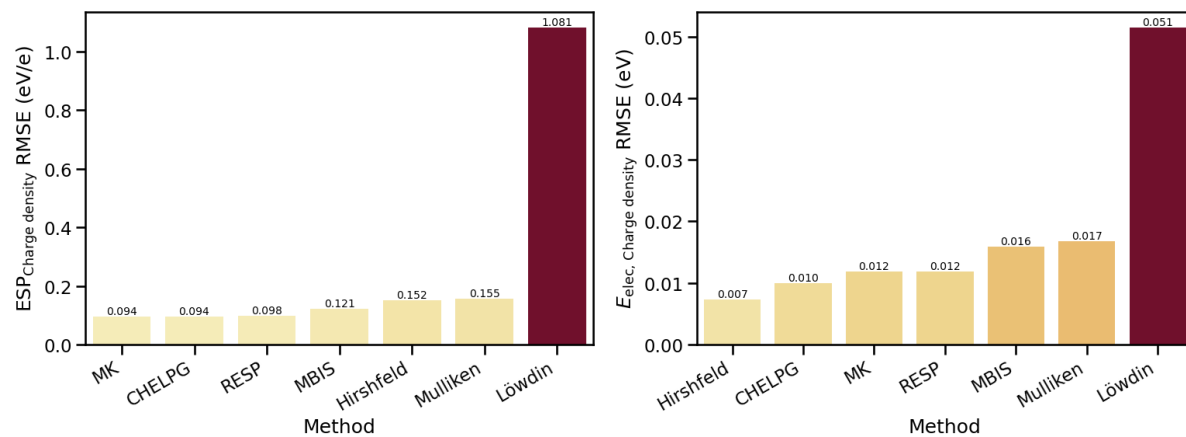


Figure D.7.: Ranking of charge assignment methods for thiol disulfide exchange reaction system in vacuum based on RMSE of electrostatic potential (left) and electrostatic energy (right) reproduction at the positions of the MM point charges of respective environment calculations

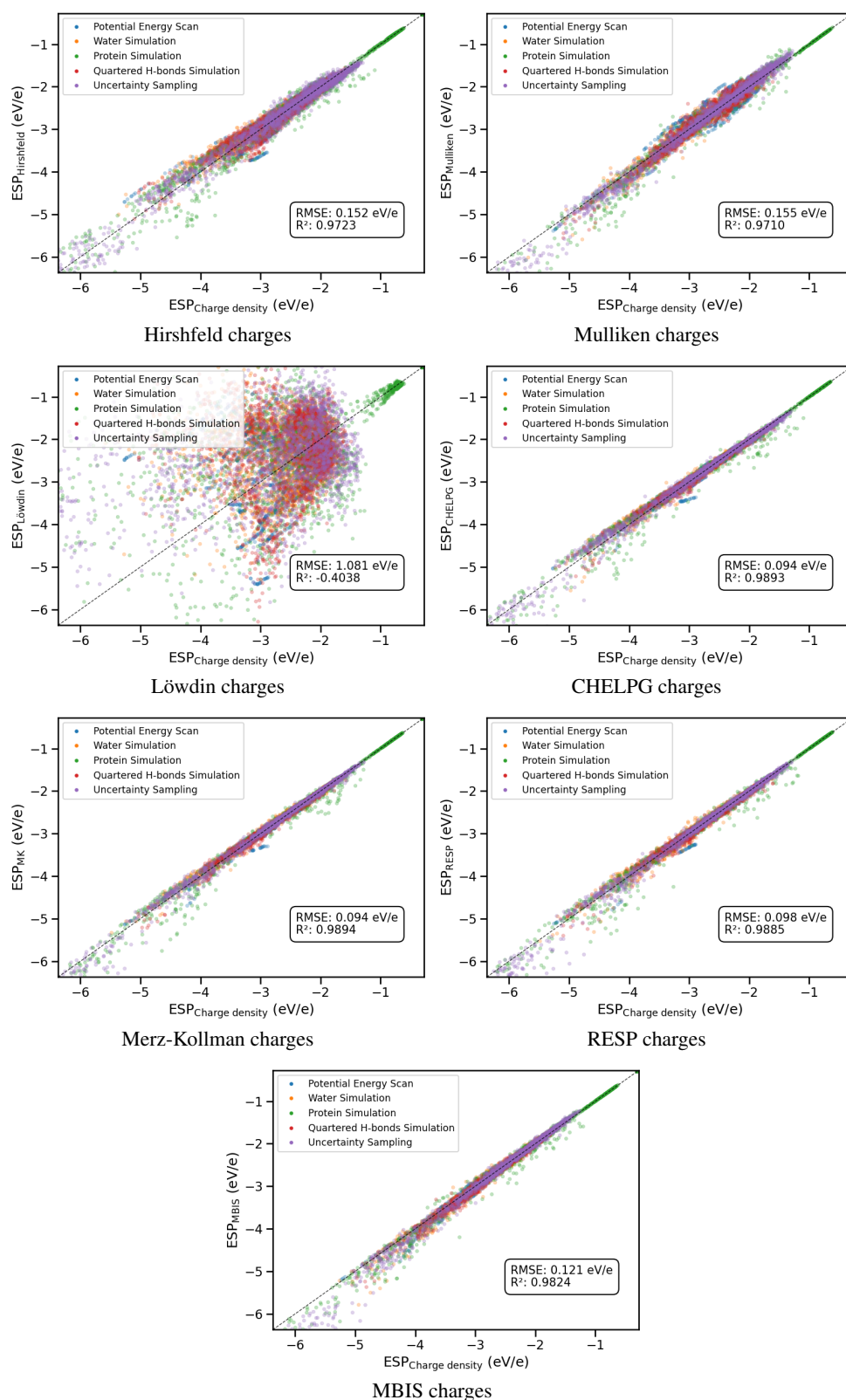
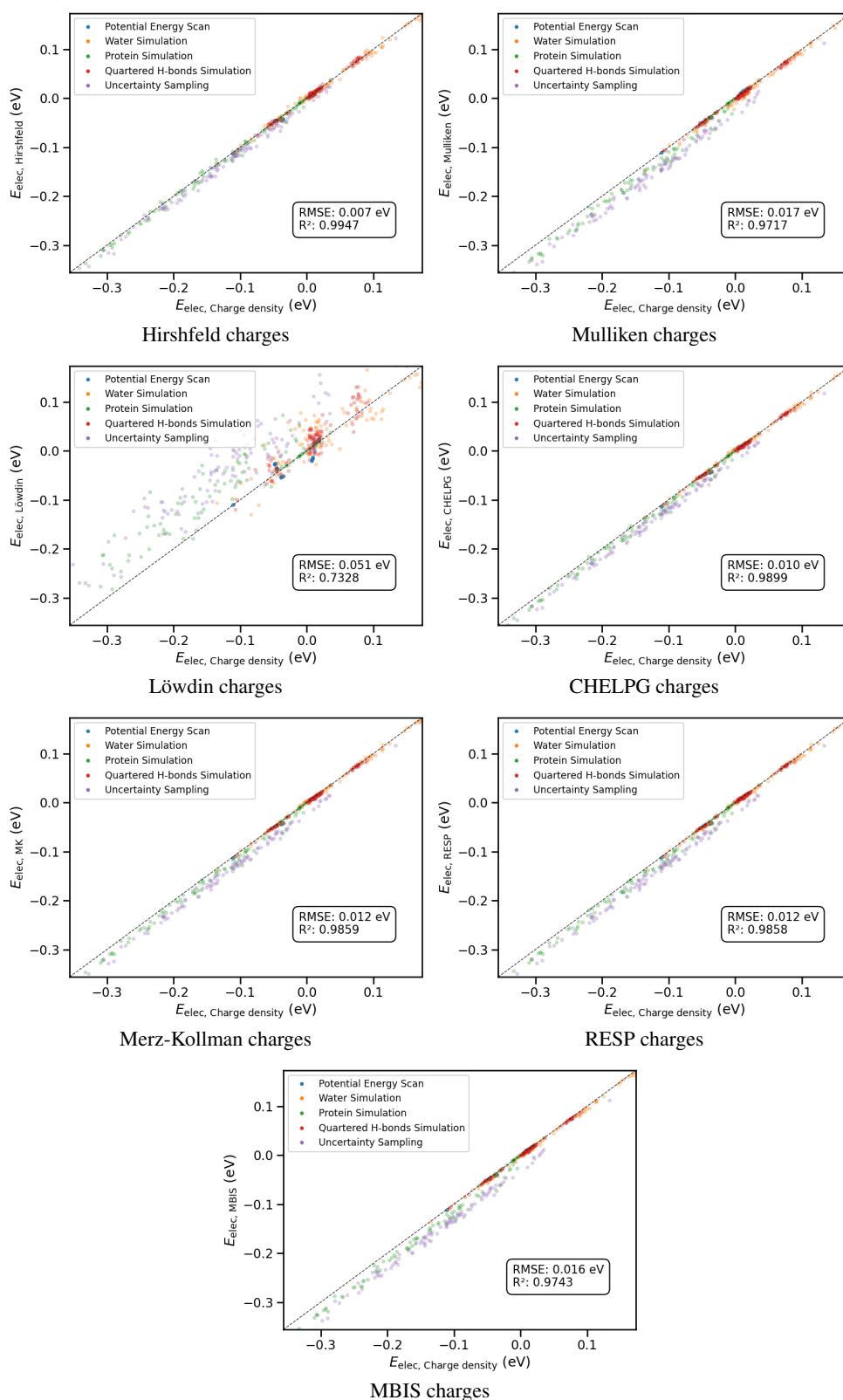


Figure D.8.: Reproduction of electrostatic potential caused by continuous charge distribution of thiol disulfide exchange reaction system in vacuum at the positions of the MM point charges of respective environment calculations



Reproduction of electrostatic energy caused by continuous charge distribution of the thiol disulfide exchange reaction system in vacuum at the positions of the MM point charges of respective environment calculations

Thiol Disulfide Exchange Reaction — Environment

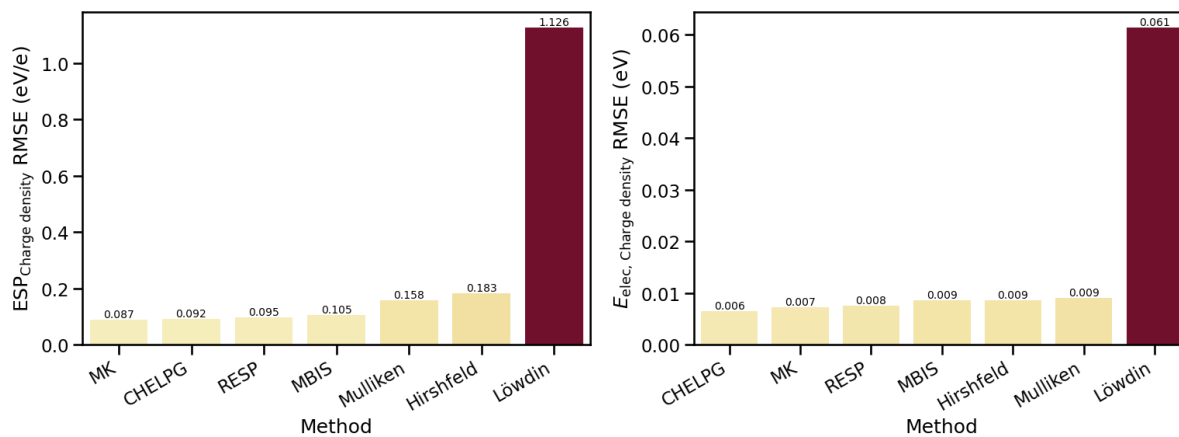


Figure D.10.: Ranking of charge assignment methods for thiol disulfide exchange reaction system in an environment based on RMSE of electrostatic potential (left) and electrostatic energy (right) reproduction at the positions of the MM point charges

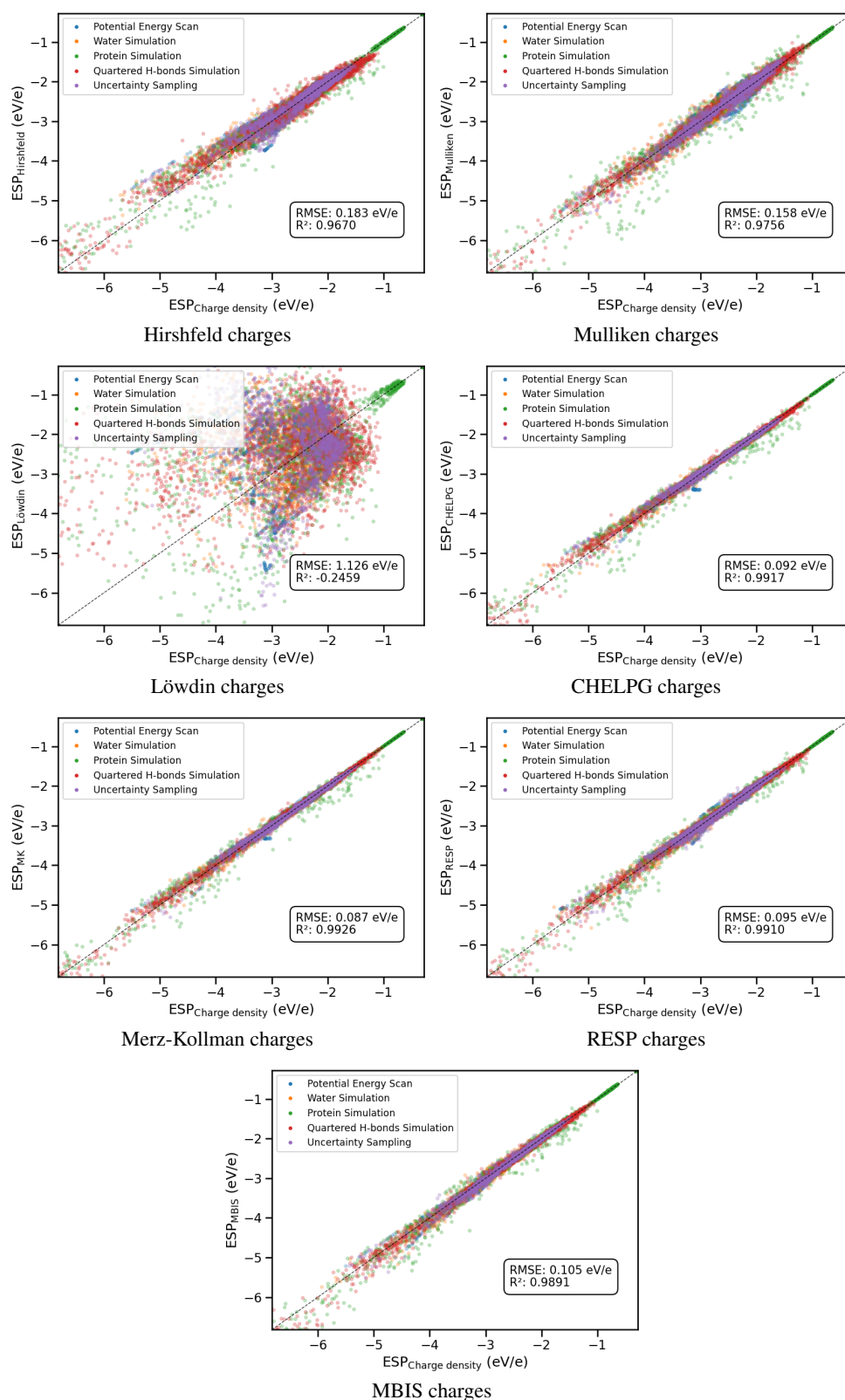


Figure D.11.: Reproduction of electrostatic potential caused by continuous charge distribution of thiol disulfide exchange reaction system in an environment at the positions of the MM point charges

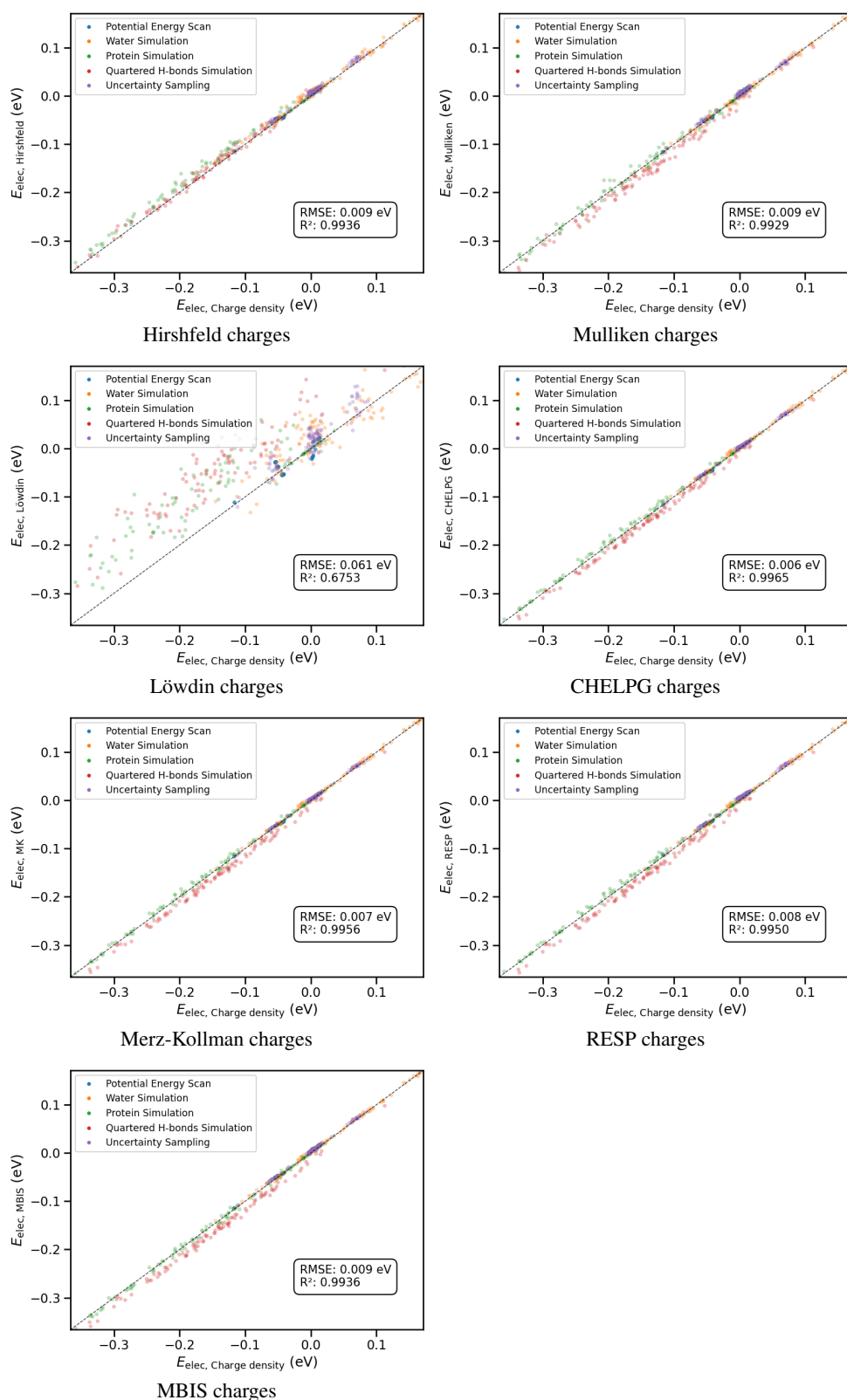


Figure D.12.: Reproduction of electrostatic energy caused by continuous charge distribution of thiol disulfide exchange reaction system in an environment at the positions of the MM point charges

D.3. Hyperparameter Search — Alanine Dipeptide

The hyperparameter search was conducted in two stages. First, a grid search was performed, in which each hyperparameter was varied individually while keeping all others fixed at their default values. Secondly, a hyperband search was used to explore the full search space simultaneously.

2nd Generation HDNNP

Grid Search

Table D.1.: Grid search hyperparameter space for the 2G-HDNNP model.

Hyperparameter	Type	Values	Default
<i>Symmetry function parameters</i>			
R_s array (Bohr)	Choice	{0.0, 1.0, 2.0, 3.0}, {0.0, 2.0, 4.0, 6.0}, {0.0, 3.0, 5.0, 7.0, 9.0}, {0.0, 1.0, 3.0, 4.0, 5.0, 6.0, 7.0}, {0.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0}, {0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 7.0, 10.0, 12.0}	{0.0, 4.0, 6.0, 8.0}
η array (Bohr ⁻²)	Choice	{0.03, 0.08}, {0.08, 0.3}, {0.3, 0.8}, {0.03, 0.16, 0.5}, {0.03, 0.3, 0.8}, {0.03, 0.08, 0.16, 0.3, 0.5}, {0.06, 0.16, 0.32, 0.6, 0.8, 1.0}, {0.03, 0.08, 0.16, 0.3, 0.5, 0.6, 0.75, 0.9, 1.0}	{0.0, 0.08, 0.3}
ζ array	Choice	{1, 2}, {1, 2, 4}, {2, 8, 16}, {1, 4, 8, 16}, {1, 2, 4, 8, 16}, {1, 2, 4, 8, 16, 32}	{2, 8, 16}
<i>Energy subnet architecture</i>			
Hidden layers	Int	1–2	1
Neurons per layer	Int	25–276 (step 50, decreasing ceiling)	25
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus	Tanh

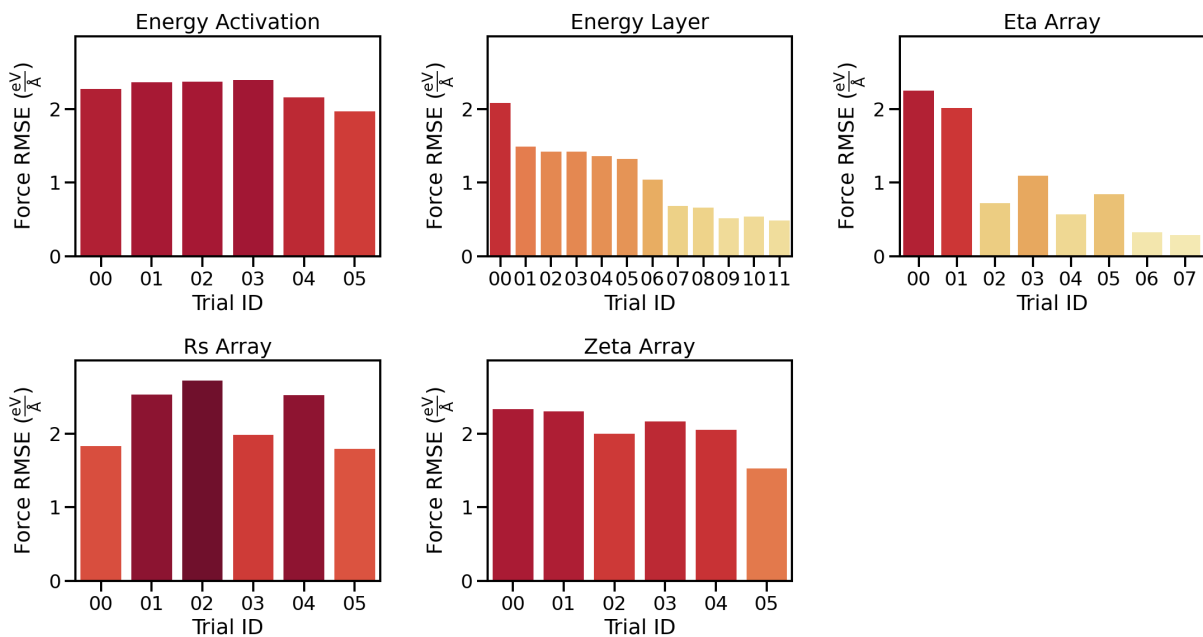


Figure D.13.: Grid search hyperparameter overview for 2G-HDNNP models

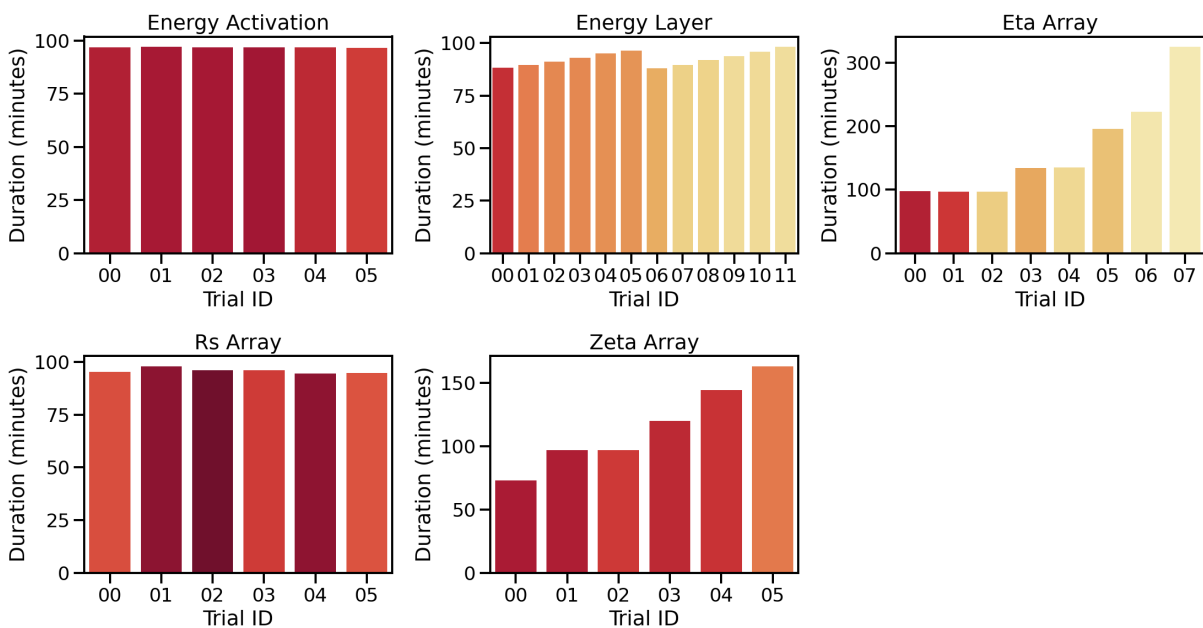


Figure D.14.: Training duration for 2G-HDNNP models with different hyperparameters

Hyperband Search

Table D.2.: Hyperband search hyperparameter space for the 2G-HDNNP model.

Hyperparameter	Type	Values
<i>Symmetry function parameters</i>		
R_s array (Bohr)	Choice	{0.0, 4.0, 6.0, 8.0}, {0.0, 3.0, 5.0, 7.0, 9.0}, {0.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0}, {0.0, 4.0, 6.0, 8.0, 10.0, 12.0, 16.0}
η array (Bohr ⁻²)	Choice	{0.0, 0.08, 0.3}, {0.03, 0.16, 0.5}, {0.0, 0.03, 0.08, 0.16, 0.3, 0.5}, {0.0, 0.06, 0.16, 0.32, 0.6, 0.8, 1.0}
λ array	Choice	{-1, 1}, {-1, 0, 1}
ζ array	Choice	{2, 8, 16}, {1, 4, 8, 16}, {1, 2, 4, 8, 16}, {1, 2, 4, 8, 16, 32}
<i>Energy subnet architecture</i>		
Hidden layers	Int	1-3
Neurons per layer	Int	25-276 (step 50, decreasing ceiling)
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus

Table D.3.: Top 3 hyperparameter configurations from the hyperband search for the 2G-HDNNP model, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
R_s array (Bohr)	{0, 3, 4, 5, 6, 7, 8}	{0, 3, 4, 5, 6, 7, 8}	{0, 3, 4, 5, 6, 7, 8}
η array (Bohr ⁻²)	{.03, .08, .16, .32, .6, .8, 1.0}	{.03, .08, .16, .32, .6, .8, 1.0}	{.03, .08, .16, .32, .6, .8, 1.0}
λ array	{-1, 1}	{-1, 1}	{-1, 1}
ζ array	{2, 8, 16}	{1, 2, 4, 8, 16}	{1, 2, 4, 8, 16, 32}
Hidden layers	225-25-75	175-25	125-25-25
Activation	Tanh	Tanh	ELU

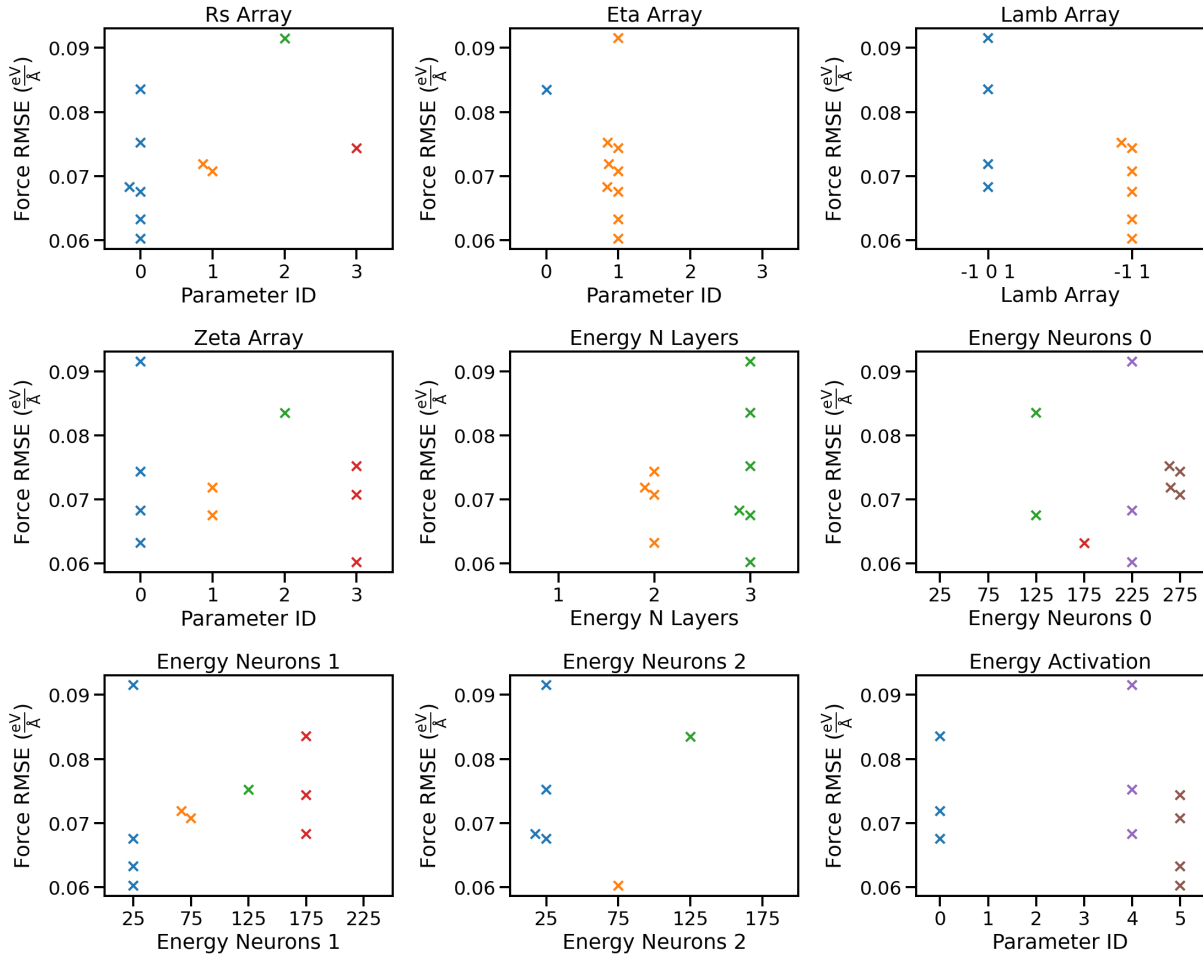


Figure D.15.: Hyperband search hyperparameter overview for 2G-HDNNP models from Top 10 trials

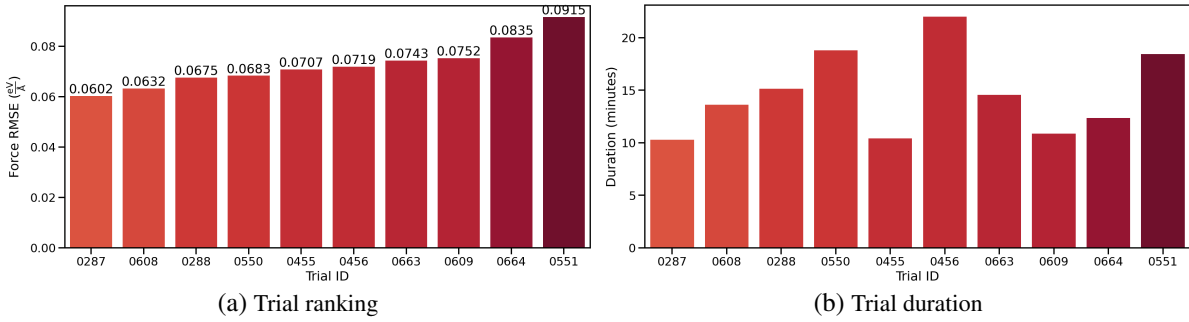


Figure D.16.: Ranking and duration of hyperband search trials for 2G-HDNNP models

4th Generation HDNNP

Grid Search

Table D.4.: Grid search hyperparameter space for the 4G-HDNNP model.

Hyperparameter	Type	Values	Default
<i>Symmetry function parameters</i>			
R_s array (Bohr)	Choice	{0.0, 1.0, 2.0, 3.0}, {0.0, 2.0, 4.0, 6.0}, {0.0, 3.0, 5.0, 7.0, 9.0}, {0.0, 1.0, 3.0, 4.0, 5.0, 6.0, 7.0}, {0.0, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0}, {0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 7.0, 10.0, 12.0}	{0.0, 4.0, 6.0, 8.0}
η array (Bohr ⁻²)	Choice	{0.03, 0.08}, {0.08, 0.3}, {0.3, 0.8}, {0.03, 0.16, 0.5}, {0.03, 0.3, 0.8}, {0.03, 0.08, 0.16, 0.3, 0.5}, {0.06, 0.16, 0.32, 0.6, 0.8, 1.0}, {0.03, 0.08, 0.16, 0.3, 0.5, 0.6, 0.75, 0.9, 1.0}	{0.0, 0.08, 0.3}
ζ array	Choice	{1, 2}, {1, 2, 4}, {2, 8, 16}, {1, 4, 8, 16}, {1, 2, 4, 8, 16}, {1, 2, 4, 8, 16, 32}	{2, 8, 16}
<i>Charge subnet architecture</i>			
Hidden layers	Int	1–2	1
Neurons per layer	Int	25–101 (step 25, decreasing ceiling)	25
Activation function	Fixed	Tanh	Tanh
<i>Energy subnet architecture</i>			
Hidden layers	Int	1–2	1
Neurons per layer	Int	25–276 (step 50, decreasing ceiling)	25
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus	Tanh

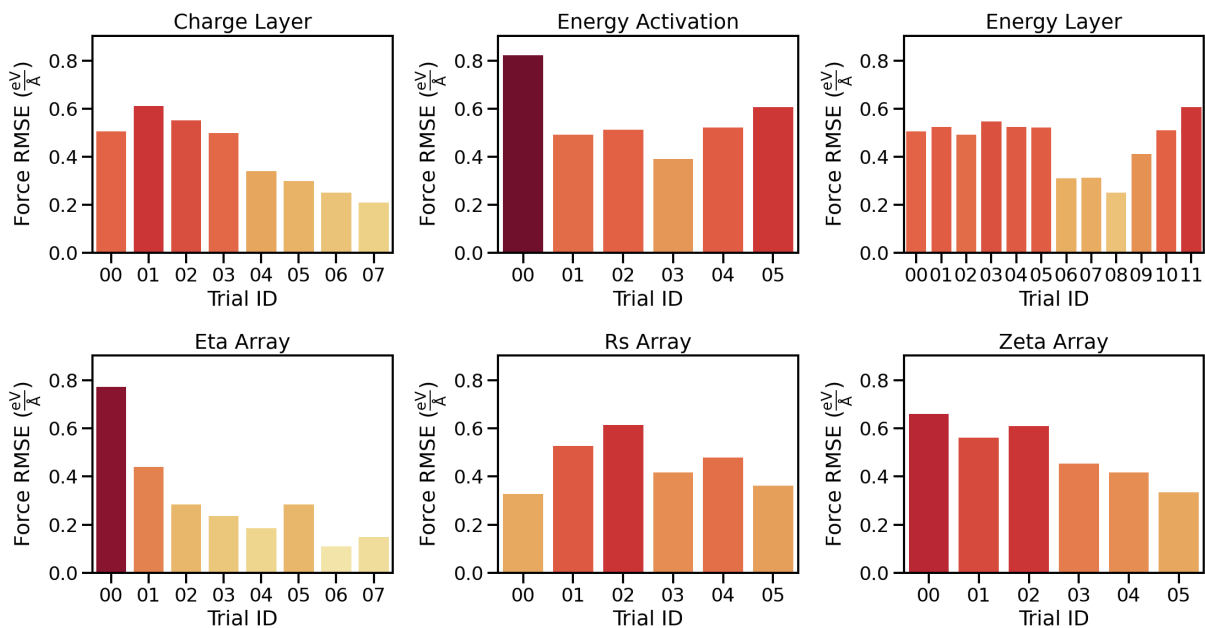


Figure D.17.: Grid search hyperparameter overview for 4G-HDNNP models

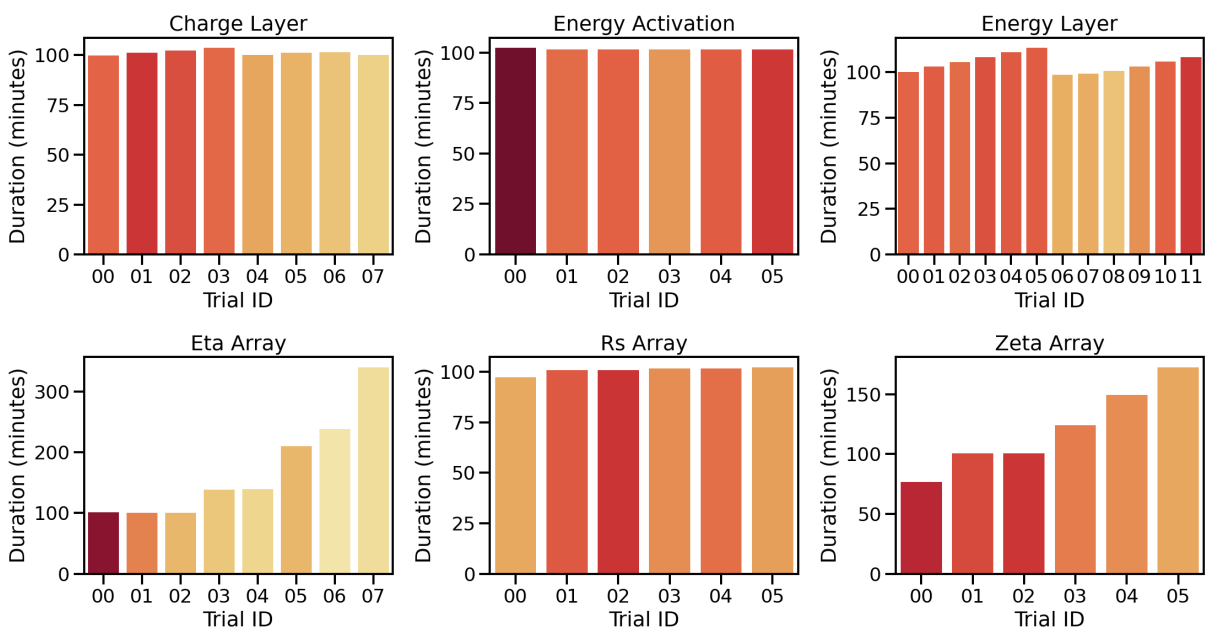


Figure D.18.: Training duration for 4G-HDNNP models with different hyperparameters

Hyperband Search

Table D.5.: Hyperband search hyperparameter space for the 4G-HDNNP model.

Hyperparameter	Type	Values
<i>Symmetry function parameters</i>		
R_s array (Bohr)	Choice	{0.0, 4.0, 6.0, 8.0}, {0.0, 4.0, 6.0, 8.0, 10.0, 12.0, 16.0}
η array (Bohr ⁻²)	Choice	{0.08, 0.3}, {0.06, 0.16, 0.32, 0.6, 0.8, 1.0}
λ array	Fixed	{-1, 1}
ζ array	Choice	{2, 8, 16}, {1, 2, 4, 8, 16, 32}
<i>Charge subnet architecture</i>		
Hidden layers	Fixed	1
Neurons per layer	Int	25–51 (step 25)
Activation function	Fixed	Tanh
<i>Energy subnet architecture</i>		
Hidden layers	Int	1–3
Neurons per layer	Int	25–276 (step 50, decreasing ceiling)
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus

Table D.6.: Top 3 hyperparameter configurations from the hyperband search for the 4G-HDNNP model, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
R_s array (Bohr)	{0, 4, 6, 8}	{0, 4, 6, 8}	{0, 4, 6, 8}
η array (Bohr ⁻²)	{.06, .16, .32, .6, .8, 1.0}	{.06, .16, .32, .6, .8, 1.0}	{.06, .16, .32, .6, .8, 1.0}
λ array	{-1, 1}	{-1, 1}	{-1, 1}
ζ array	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}
Charge layers	50	50	50
Charge activation	Tanh	Tanh	Swish
Energy layers	275-225-25	175-25-25	225-75-125
Energy activation	Swish	Tanh	ELU

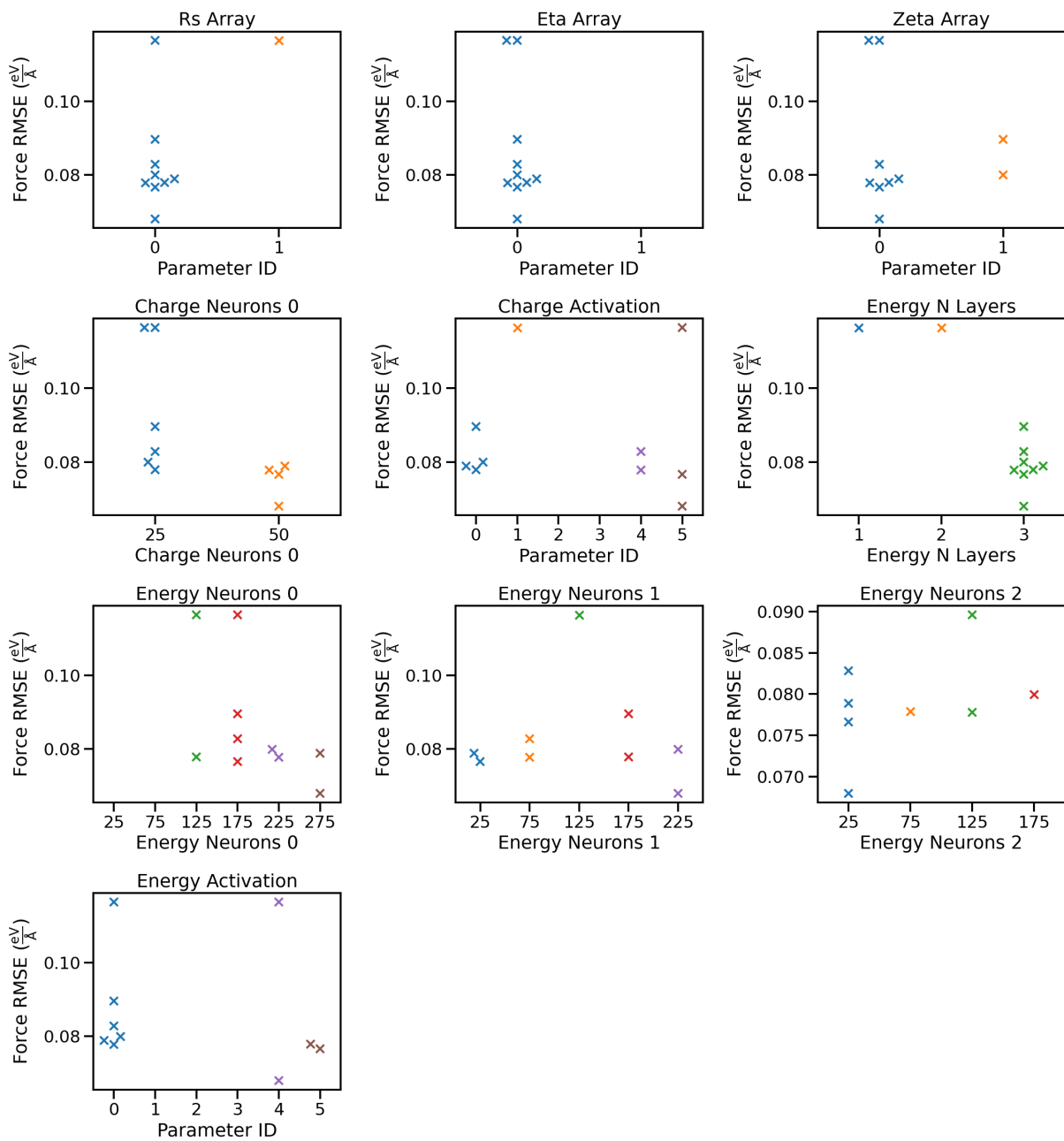


Figure D.19.: Hyperband search hyperparameter overview for 4G-HDNNP models from Top 10 trials

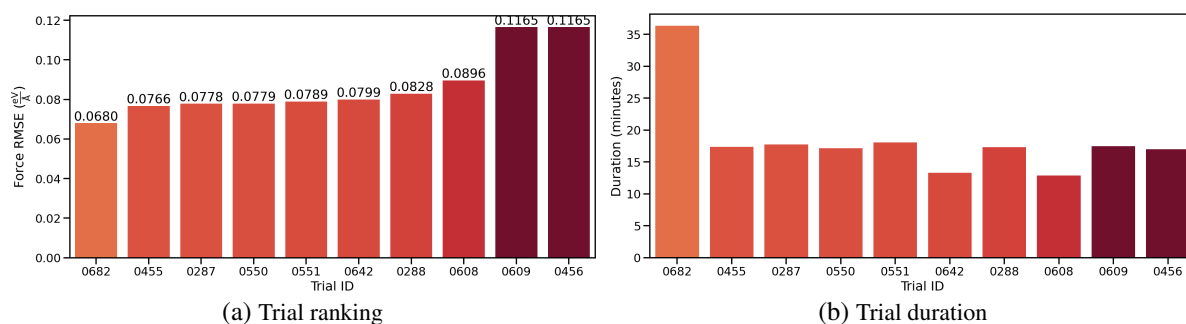


Figure D.20.: Ranking and duration of hyperband search trials for 4G-HDNNP models

SchNet

Grid Search

The default hyperparameter value for cutoff of the Gaussian radial basis function parameters turned out to be suboptimal, leading to significantly worse performance during the other hyperparameter variations. Only the best 12 trials are included in the grid search overview figure, which is why all the high error trials with the default cutoff of 4.0 Bohr are not visible in the overview figure.

Table D.7.: Grid search hyperparameter space for the SchNet model.

Hyperparameter	Type	Values	Default
<i>Model architecture</i>			
Input embedding units	Choice	16, 32, 64, 96, 128, 256	32
Interaction units	Choice	16, 32, 64, 96, 128, 256	32
Interaction blocks	Int	3–8	5
<i>Radial basis functions</i>			
Gaussian bins	Int	20–100 (step 10)	20
Cutoff distance (Bohr)	Choice	4.0, 5.0, 6.0	4.0
Gaussian offset (Bohr)	Choice	0.0, 0.5, 1.0	0.0
Gaussian width (Bohr ⁻²)	Choice	0.2, 0.4, 0.6	0.2
<i>Output MLP</i>			
Layer configuration	Choice	32-1, 64-1, 128-1, 64-32-1, 128-64-1, 256-128-1, 128-128-64-1, 256-128-64-1	128-64-1
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus	Shifted Softplus

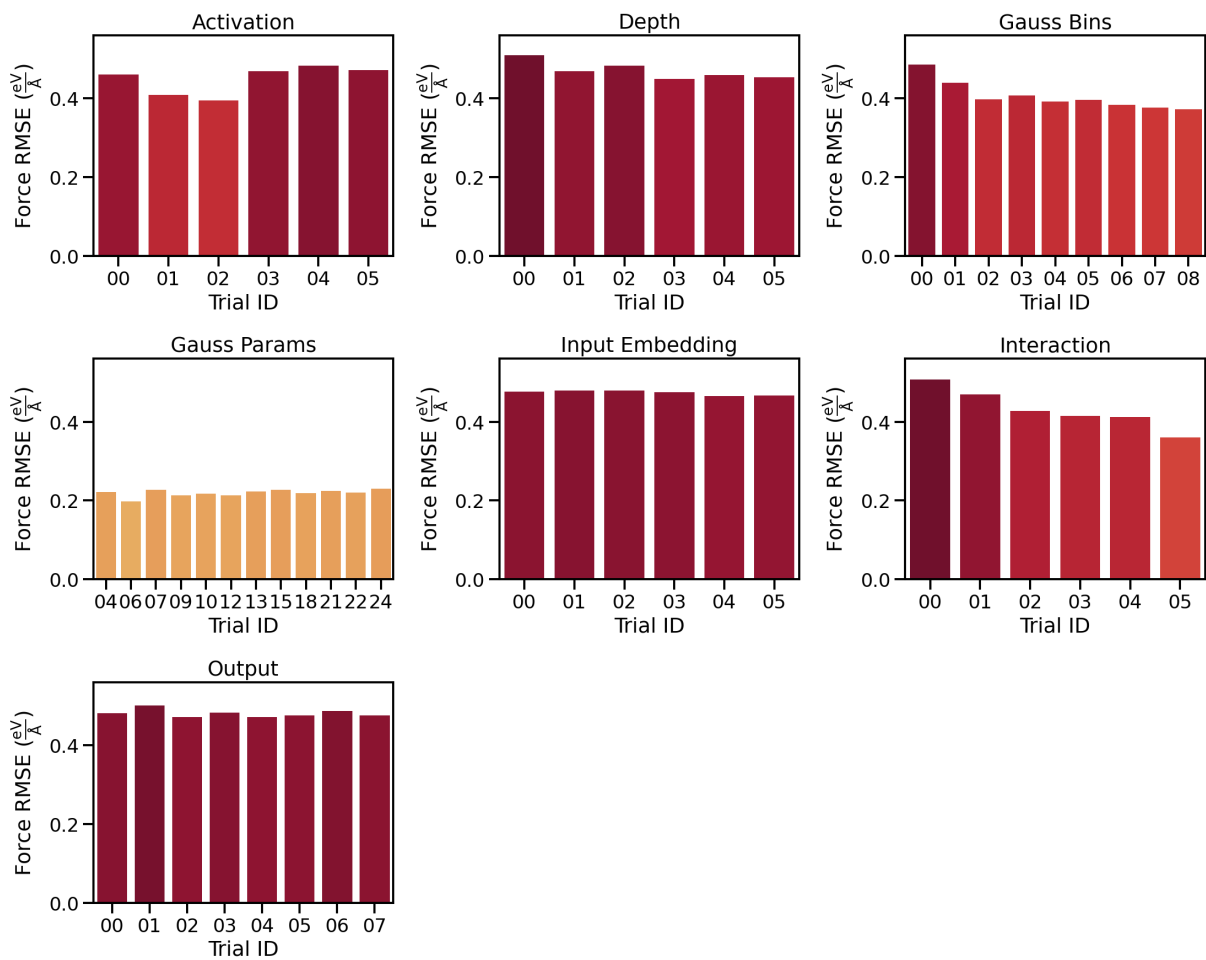


Figure D.21.: Grid search hyperparameter overview for SchNet models

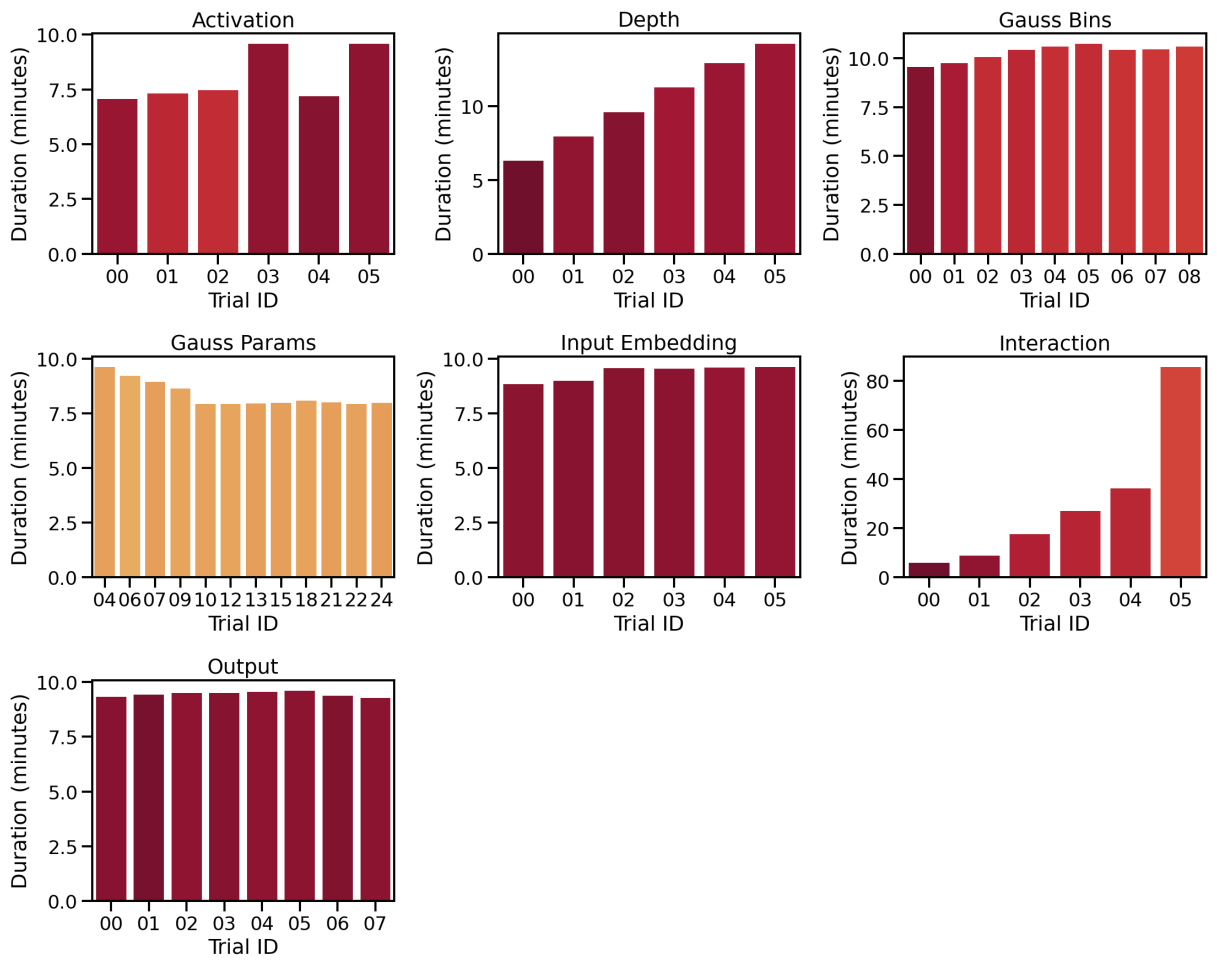


Figure D.22.: Training duration for SchNet models with different hyperparameters

Hyperband Search

Table D.8.: Hyperband search hyperparameter space for the SchNet model.

Hyperparameter	Type	Values
<i>Model architecture</i>		
Input embedding units	Choice	32, 64, 128, 256
Interaction units	Choice	32, 64, 128, 256
Interaction blocks	Int	3–8
<i>Radial basis functions</i>		
Gaussian bins	Int	20–101 (step 10)
Cutoff distance (Bohr)	Choice	4.0, 5.0, 6.0
Gaussian offset (Bohr)	Choice	0.0, 0.5, 1.0
Gaussian width (Bohr ⁻²)	Choice	0.2, 0.4, 0.6
<i>Output MLP</i>		
Layer configuration	Choice	128-64-1, 256-128-1, 128-128-64-1, 256-128-64-1
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus

Table D.9.: Top 3 hyperparameter configurations from the hyperband search for the SchNet model, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
Input embedding units	256	256	64
Interaction units	256	256	256
Interaction blocks	8	8	6
Gaussian bins	60	60	60
Cutoff distance (Bohr)	6.0	6.0	6.0
Gaussian offset (Bohr)	1.0	1.0	0.5
Gaussian width (Bohr ⁻²)	0.4	0.4	0.4
Output MLP	256-128-64-1	256-128-64-1	256-128-1
Activation	Swish	Swish	Tanh

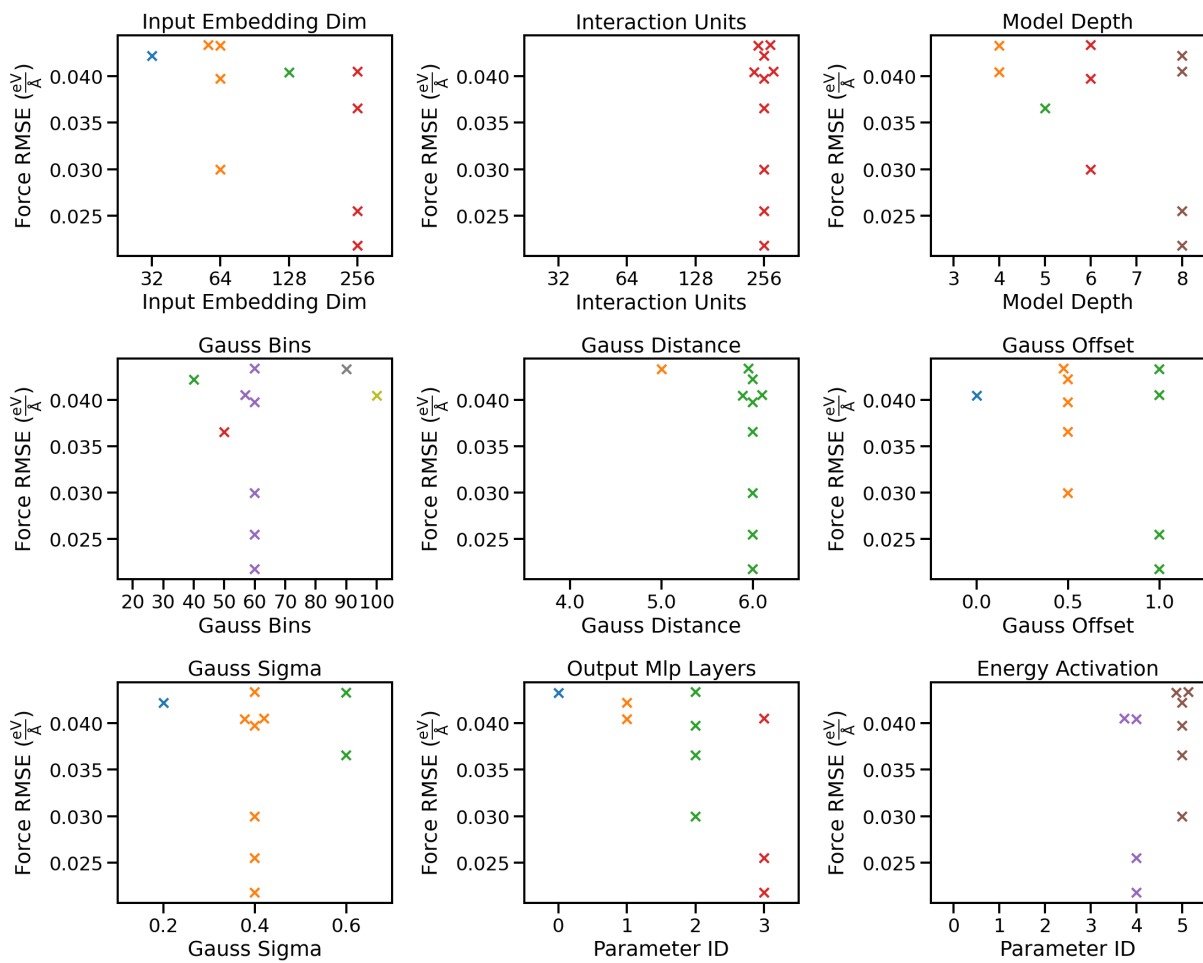


Figure D.23.: Hyperband search hyperparameter overview for SchNet models from Top 10 trials

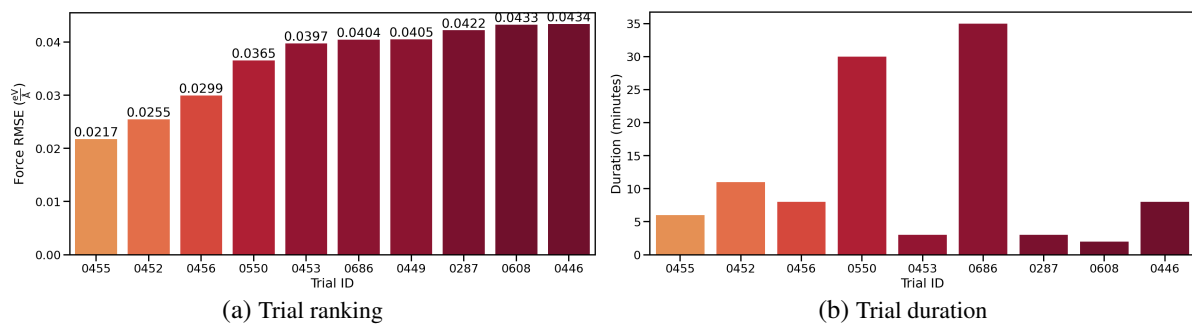


Figure D.24.: Ranking and duration of hyperband search trials for SchNet models

PaiNN

Grid Search

Table D.10.: Grid search hyperparameter space for the PaiNN model.

Hyperparameter	Type	Values	Default
<i>Model architecture</i>			
Input embedding units/Interaction units	Choice	16, 32, 64, 96, 128, 256	64
Interaction blocks	Int	3–8	5
<i>Radial basis functions</i>			
Number of Bessel functions	Int	15–50 (step 5)	15
Cutoff distance (Bohr)	Choice	4.0, 5.0, 6.0	4.0
Envelope exponent	Int	4–6	4
<i>Output MLP</i>			
Layer configuration	Choice	32-1, 64-1, 128-1, 64-32-1, 128-64-1, 256-128-1, 128-128-64-1, 256-128-64-1	128-64-1
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus	Shifted Softplus

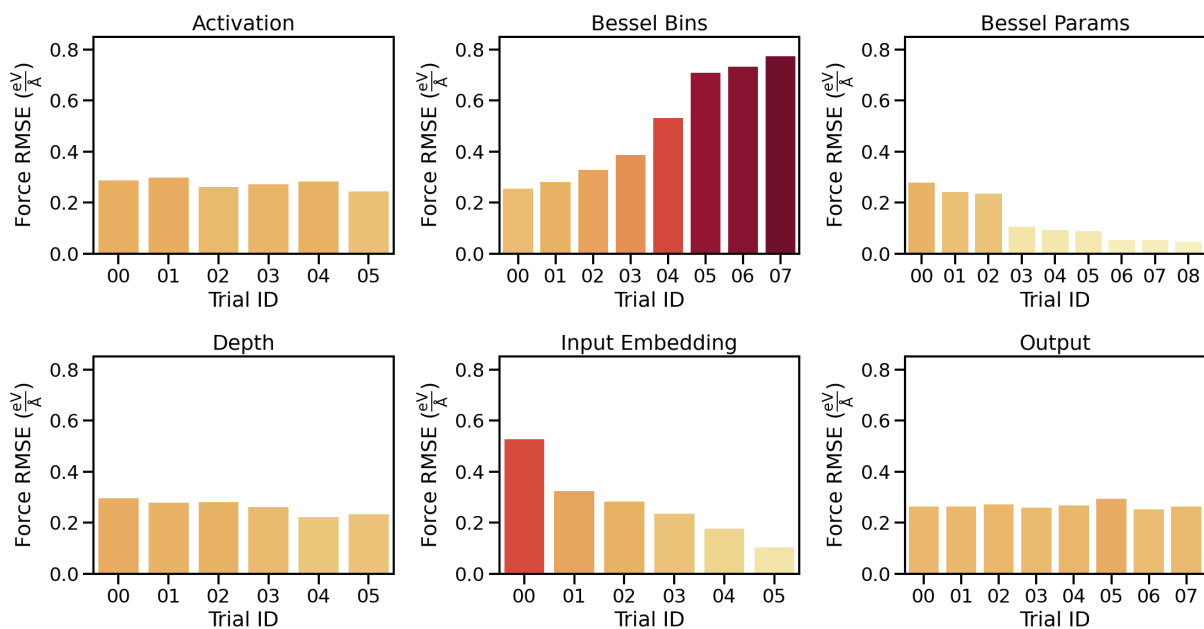


Figure D.25.: Grid search hyperparameter overview for PaiNN models

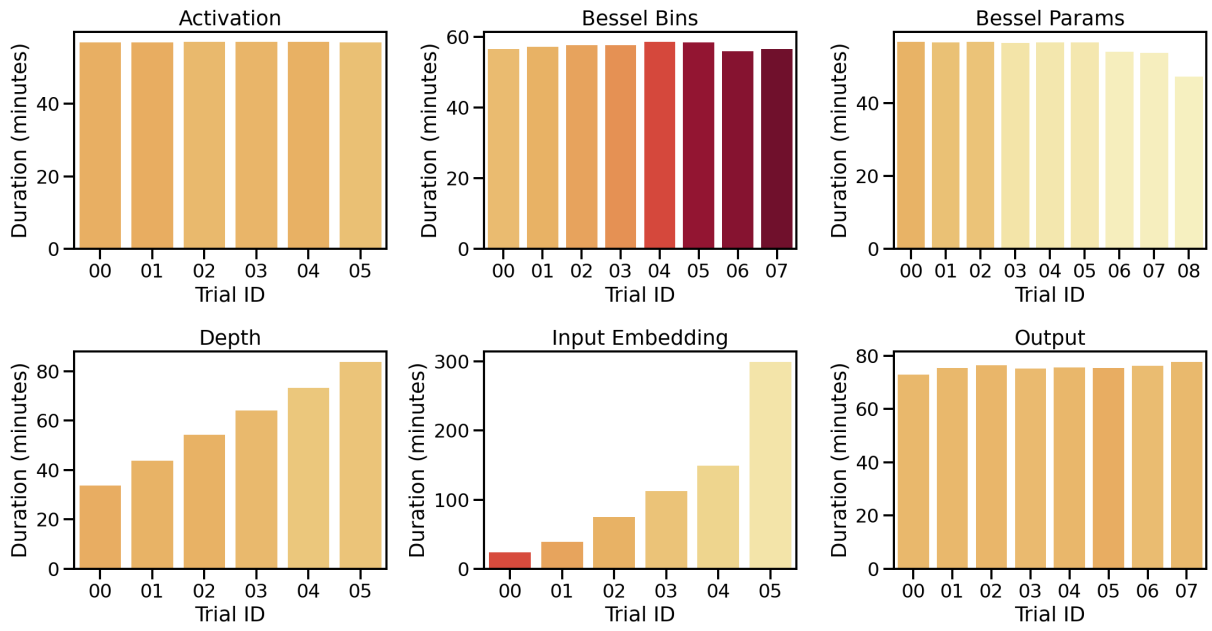


Figure D.26.: Training duration for PaiNN models with different hyperparameters

Hyperband Search

Table D.11.: Hyperband search hyperparameter space for the PaiNN model.

Hyperparameter	Type	Values
<i>Model architecture</i>		
Input embedding units	Choice	64, 128, 256
Interaction blocks	Int	3–8
<i>Radial basis functions</i>		
Number of Bessel functions	Int	15–30 (step 5)
Cutoff distance (Bohr)	Choice	4.0, 5.0, 6.0
Envelope exponent	Int	4–6
<i>Output MLP</i>		
Layer configuration	Choice	128-1, 256-1, 128-64-1, 256-128-1
Activation function	Choice	ReLU, Tanh, ELU, Swish, Leaky ReLU, Shifted Softplus

Table D.12.: Top 3 hyperparameter configurations from the hyperband search for the PaiNN model, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
Input embedding units	256	256	256
Interaction blocks	8	6	5
Bessel functions	15	15	15
Cutoff distance (Bohr)	6.0	6.0	6.0
Envelope exponent	5	5	5
Output MLP	256-128-1	128-1	256-1
Activation	Leaky ReLU	Swish	Tanh

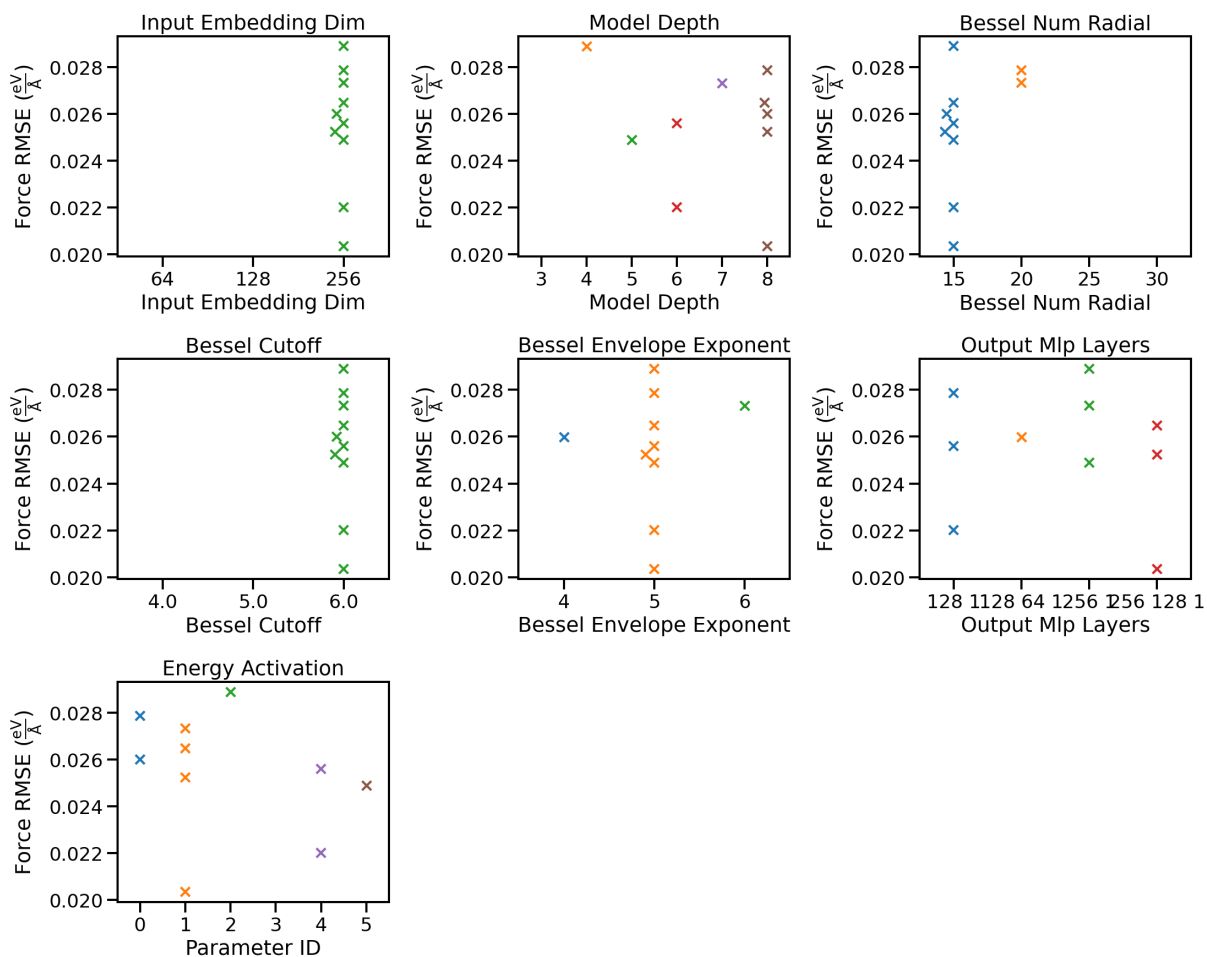


Figure D.27.: Hyperband search hyperparameter overview for PaiNN models from Top 10 trials

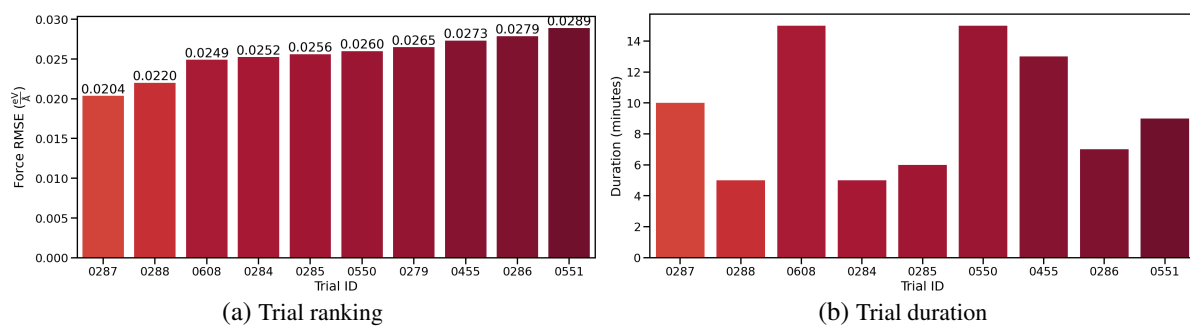


Figure D.28.: Ranking and duration of hyperband search trials for PaiNN models

Base MACE

Grid Search

Table D.13.: Grid search hyperparameter space for the Base MACE model.

Hyperparameter	Type	Values	Default
Number of Bessel functions	Int	4, 12, 20, 28, 36	8
Cutoff distance (Å)	Fixed	5.0	5.0
Envelope exponent	Int	4, 6, 8, 10, 12	5
Correlation order	Int	1, 2, 3	3
Interaction blocks	Int	2, 3, 4, 5	3
MLP irreps	Choice	16x0e, 32x0e, 64x0e, 128x0e	16x0e

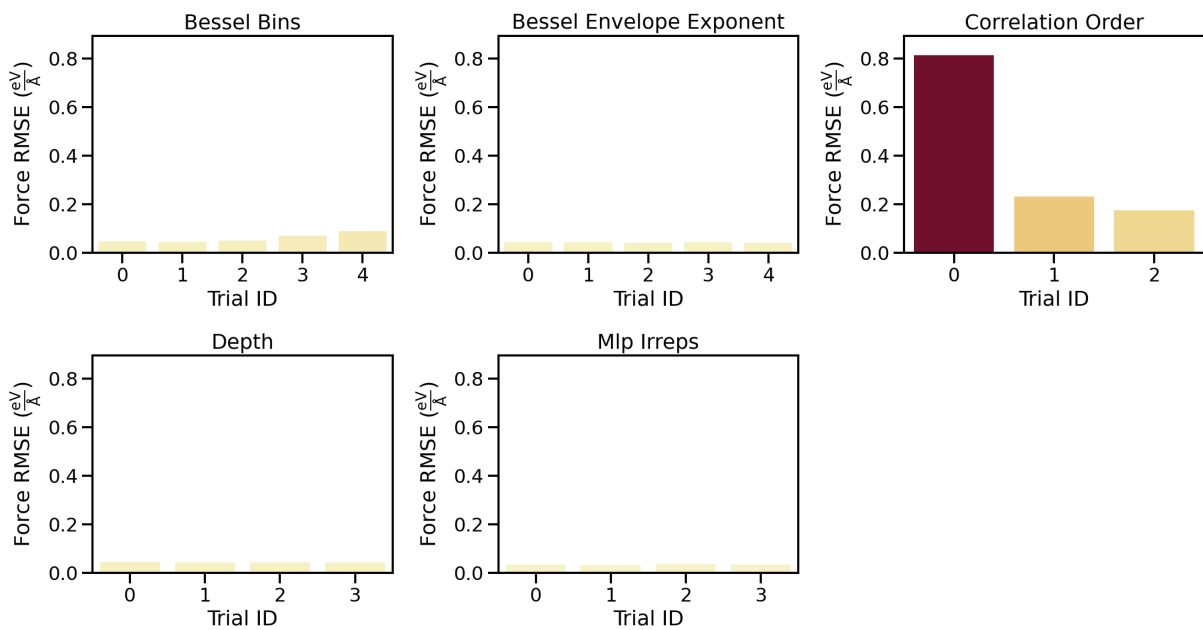


Figure D.29.: Grid search hyperparameter overview for Base MACE models

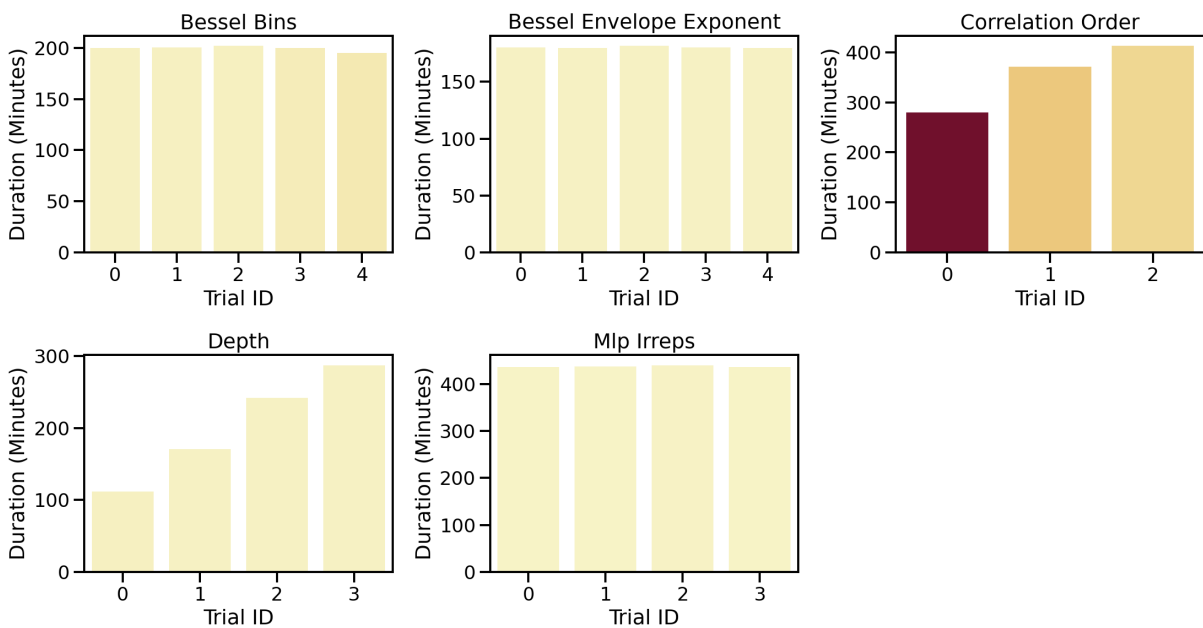


Figure D.30.: Training duration for Base MACE models with different hyperparameters

Hyperband Search

Table D.14.: Hyperband search hyperparameter space for the MACE model.

Hyperparameter	Type	Values
Number of Bessel functions	Int	4, 12, 20, 28, 36
Envelope exponent	Int	4, 6, 8, 10
Cutoff distance (Å)	Fixed	5.0
Correlation order	Int	2, 3
Interaction blocks	Int	2, 3, 4
MLP irreps	Choice	16x0e, 32x0e, 64x0e
Radial MLP	Choice	[32, 32], [64, 64], [128, 128], [32, 32, 32], [64, 64, 64], [128, 64, 32], [128, 128, 64]
Number of channels	Int	32, 64, 96, 128
$L_{\max}$	Int	1, 2, 3
Gate function	Choice	SiLU, tanh

Table D.15.: Top 3 hyperparameter configurations from the hyperband search for the Base MACE model, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
MLP irreps	32x0e	16x0e	32x0e
Correlation order	2	3	3
Gate function	SiLU	Tanh	SiLU
$L_{\max}$	3	2	2
Number of channels	128	128	128
Envelope exponent	6	8	4
Interaction blocks	4	3	3
Bessel functions	12	12	12
Radial MLP	[64, 64, 64]	[128, 64, 32]	[128, 64, 32]

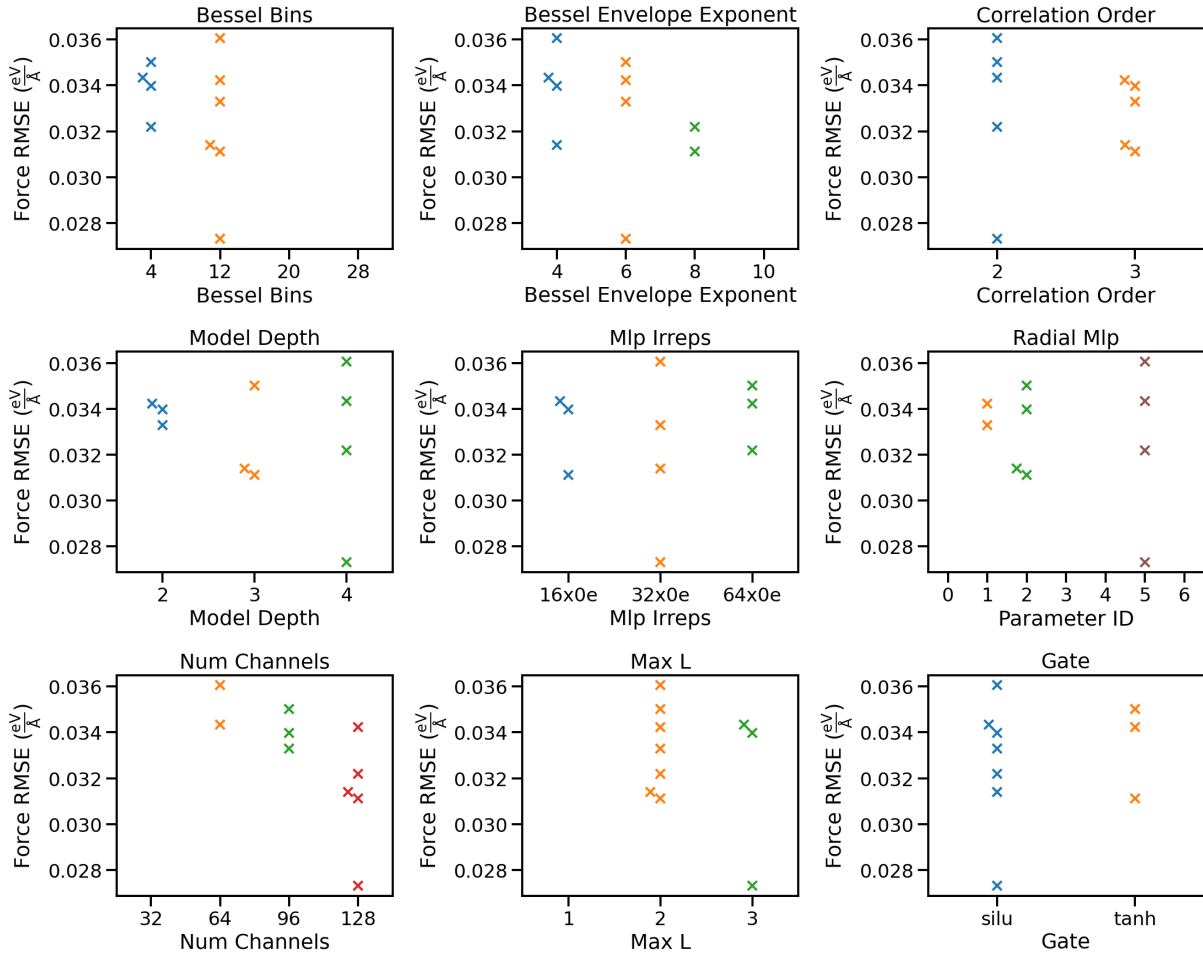


Figure D.31.: Hyperband search hyperparameter overview for Base MACE models from Top 10 trials

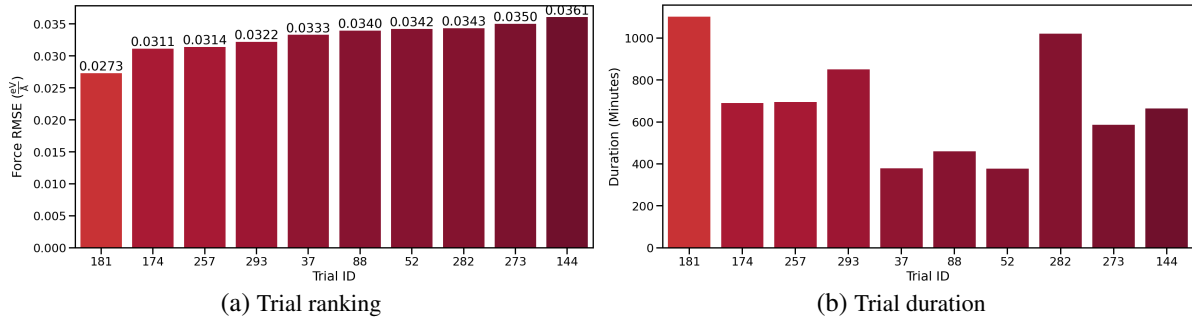


Figure D.32.: Ranking and duration of hyperband search trials for Base MACE models

D.4. Hyperparameter Search — Thiol Disulfide Exchange Reaction

2nd Generation HDNNP

Hyperband Search

The same search space as in Table D.2 was used for the hyperband search of 2G-HDNNP models for the thiol disulfide exchange reaction system.

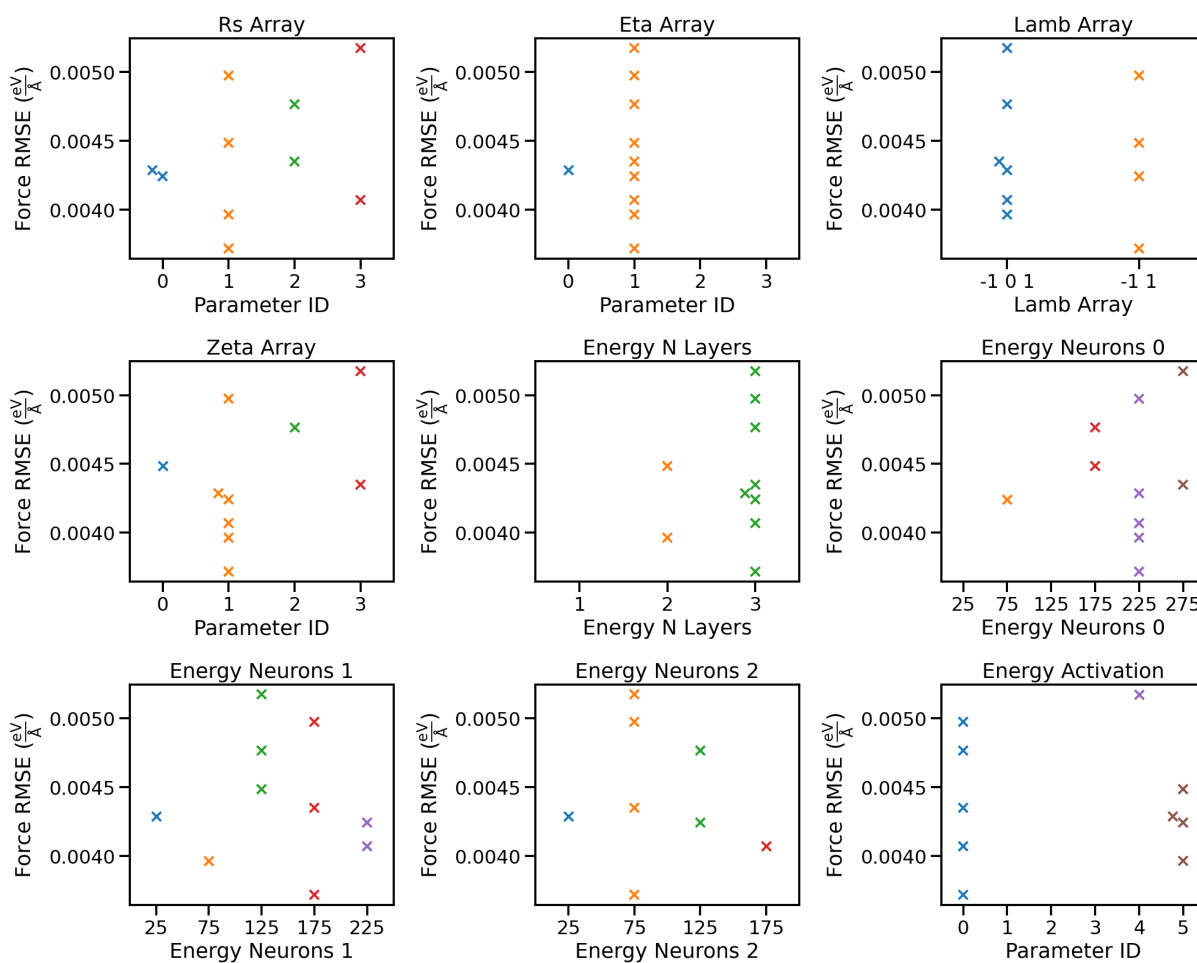


Figure D.33.: Hyperband search hyperparameter overview for 2G-HDNNP models from Top 10 trials

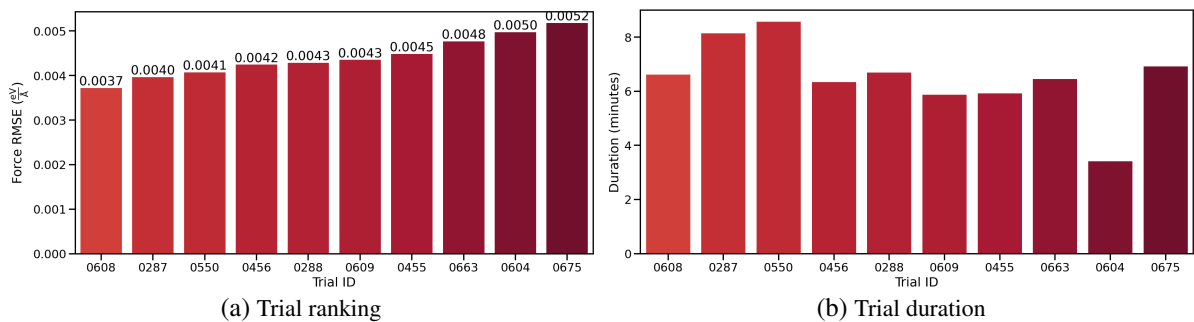


Figure D.34.: Ranking and duration of hyperband search trials for 2G-HDNNP models

Table D.16.: Top 3 hyperparameter configurations from the hyperband search for the 2G-HDNNP model on the thiol disulfide dataset, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
R_s array (Bohr)	{0, 3, 5, 7, 9}	{0, 3, 5, 7, 9}	{0, 4, 6, 8, 10, 12, 16}
η array (Bohr ⁻²)	{.03, .08, .16, .32, .6, .8, 1.0}	{.03, .08, .16, .32, .6, .8, 1.0}	{.03, .08, .16, .32, .6, .8, 1.0}
λ array	{-1, 1}	{-1, 0, 1}	{-1, 0, 1}
ζ array	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}
Hidden layers	225-175-75	225-75	225-225-175
Activation	ELU	Tanh	ELU

4th Generation HDNNP

Hyperband Search

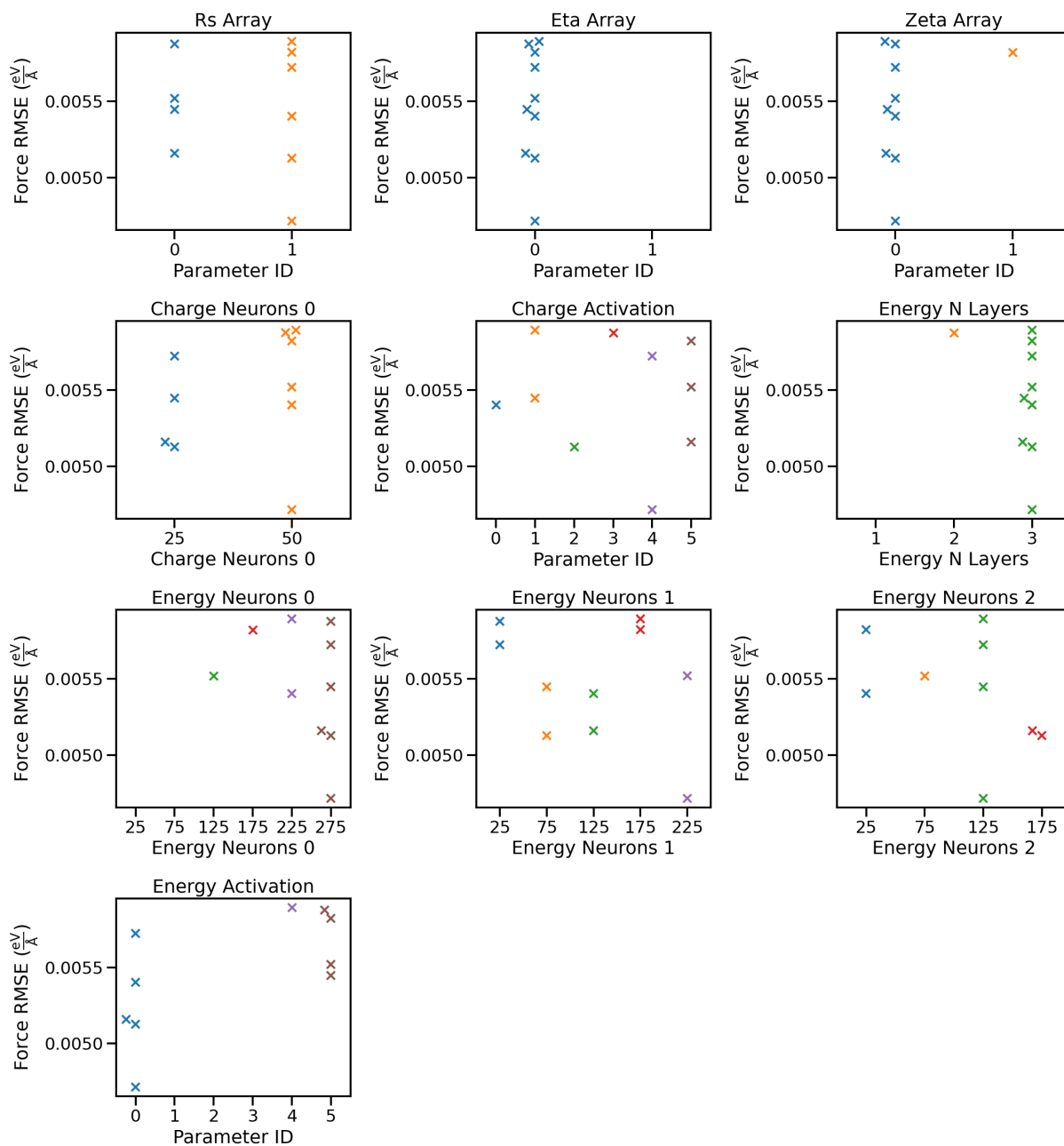


Figure D.35.: Hyperband search hyperparameter overview for 4G-HDNNP models from Top 10 trials

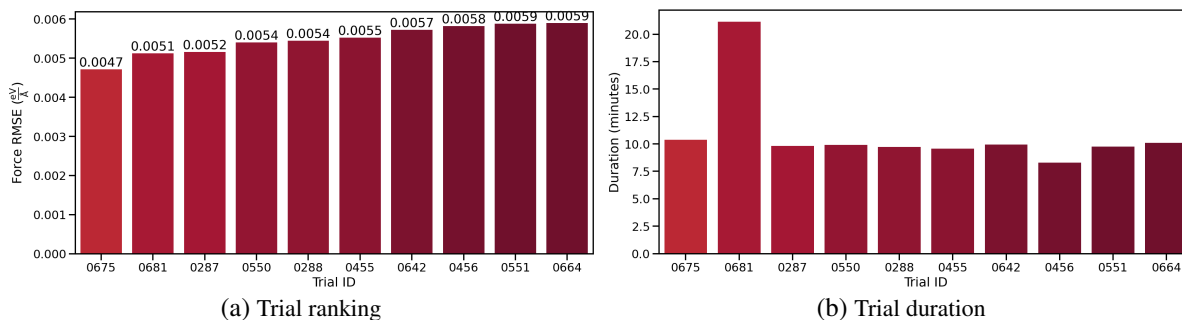


Figure D.36.: Ranking and duration of hyperband search trials for 4G-HDNNP models

Table D.17.: Top 3 hyperparameter configurations from the hyperband search for the 4G-HDNNP model on the thiol disulfide dataset, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
R_s array (Bohr)	{0, 4, 6, 8, 10, 12, 16}	{0, 4, 6, 8, 10, 12, 16}	{0, 4, 6, 8}
η array (Bohr ⁻²)	{.06, .16, .32, .6, .8, 1.0}	{.06, .16, .32, .6, .8, 1.0}	{.06, .16, .32, .6, .8, 1.0}
λ array	{-1, 1}	{-1, 1}	{-1, 1}
ζ array	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}	{1, 2, 4, 8, 16, 32}
Charge layers	50	25	25
Charge activation	Swish	ReLU	Tanh
Energy layers	275-225-125	275-75-175	275-125-175
Energy activation	ELU	ELU	ELU

SchNet

Hyperband Search

The same search space as in Table D.8 was used for the hyperband search of SchNet models for the thiol disulfide exchange reaction.

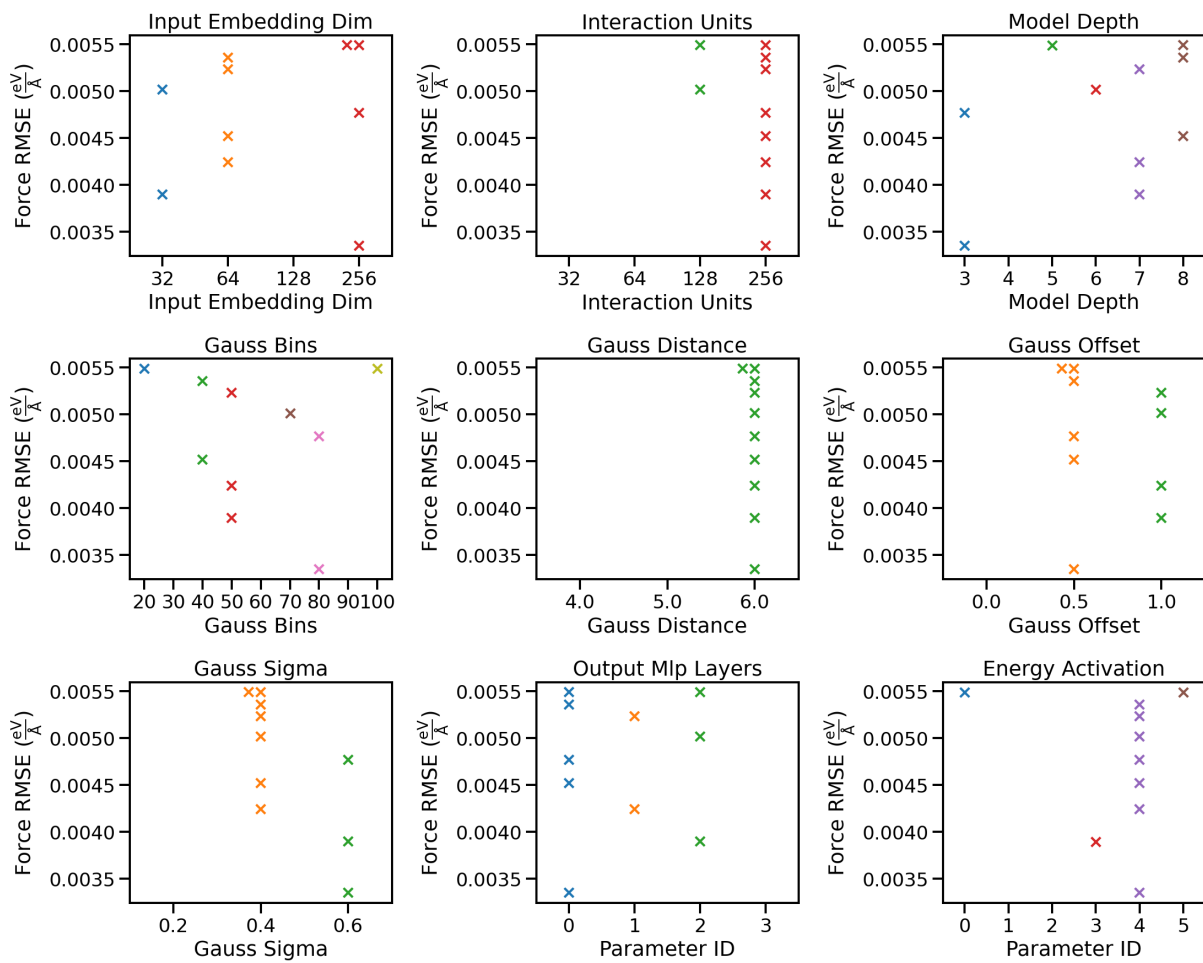


Figure D.37.: Hyperband search hyperparameter overview for SchNet models from Top 10 trials

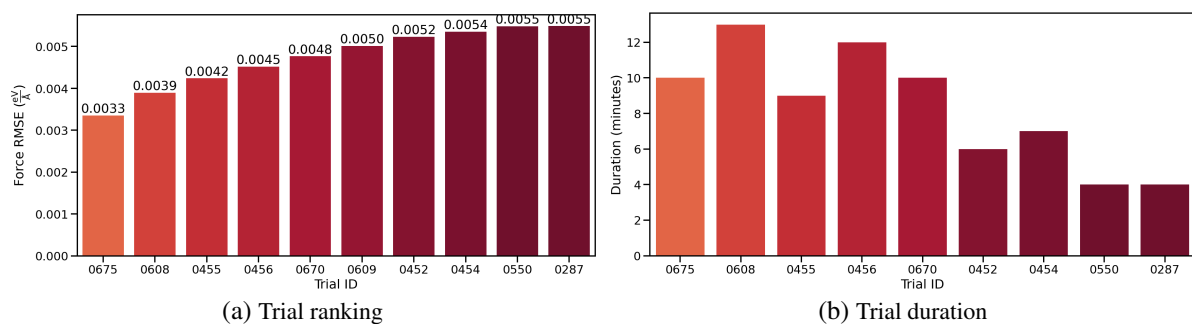


Figure D.38.: Ranking and duration of hyperband search trials for SchNet models

Table D.18.: Top 3 hyperparameter configurations from the hyperband search for the SchNet model on the thiol disulfide dataset, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
Input embedding units	256	32	64
Interaction units	256	256	256
Interaction blocks	3	7	7
Gaussian bins	80	50	50
Cutoff distance (Bohr)	6.0	6.0	6.0
Gaussian offset (Bohr)	0.5	1.0	1.0
Gaussian width (Bohr ⁻²)	0.6	0.6	0.4
Output MLP	128-128-64-1	256-128-1	128-64-1
Activation	Swish	Shifted Softplus	Swish

PaiNN

Hyperband Search

The same search space as in Table D.11 was used for the hyperband search of PaiNN models for the thiol disulfide exchange reaction.

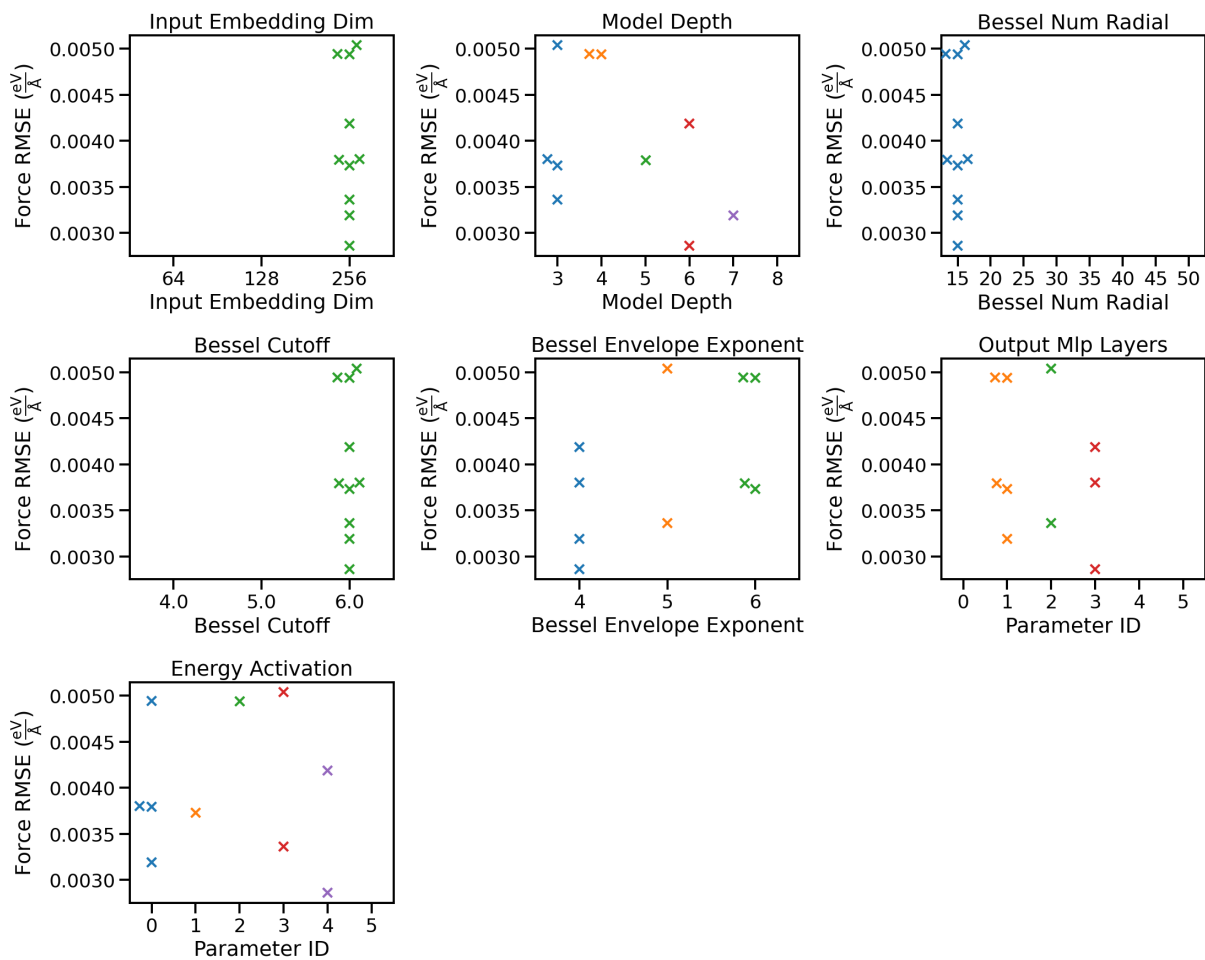


Figure D.39.: Hyperband search hyperparameter overview for PaiNN models from Top 10 trials

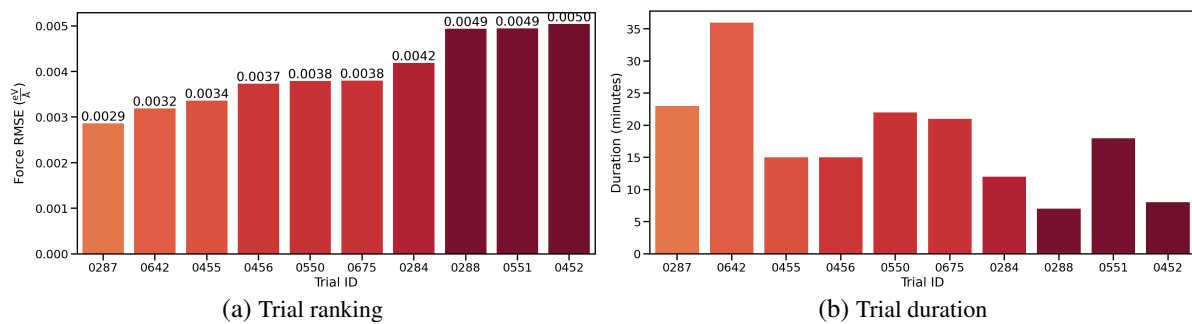


Figure D.40.: Ranking and duration of hyperband search trials for PaiNN models

Table D.19.: Top 3 hyperparameter configurations from the hyperband search for the PaiNN model on the thiol disulfide dataset, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
Input embedding units	256	256	256
Interaction blocks	6	7	3
Bessel functions	15	15	15
Cutoff distance (Bohr)	6.0	6.0	6.0
Envelope exponent	4	4	5
Output MLP	256-128-1	128-64-1	256-1
Activation	Swish	ELU	Shifted Softplus

Base MACE

Hyperband Search

The same search space as in Table D.14 was used for the hyperband search of MACE models for the thiol disulfide exchange reaction.

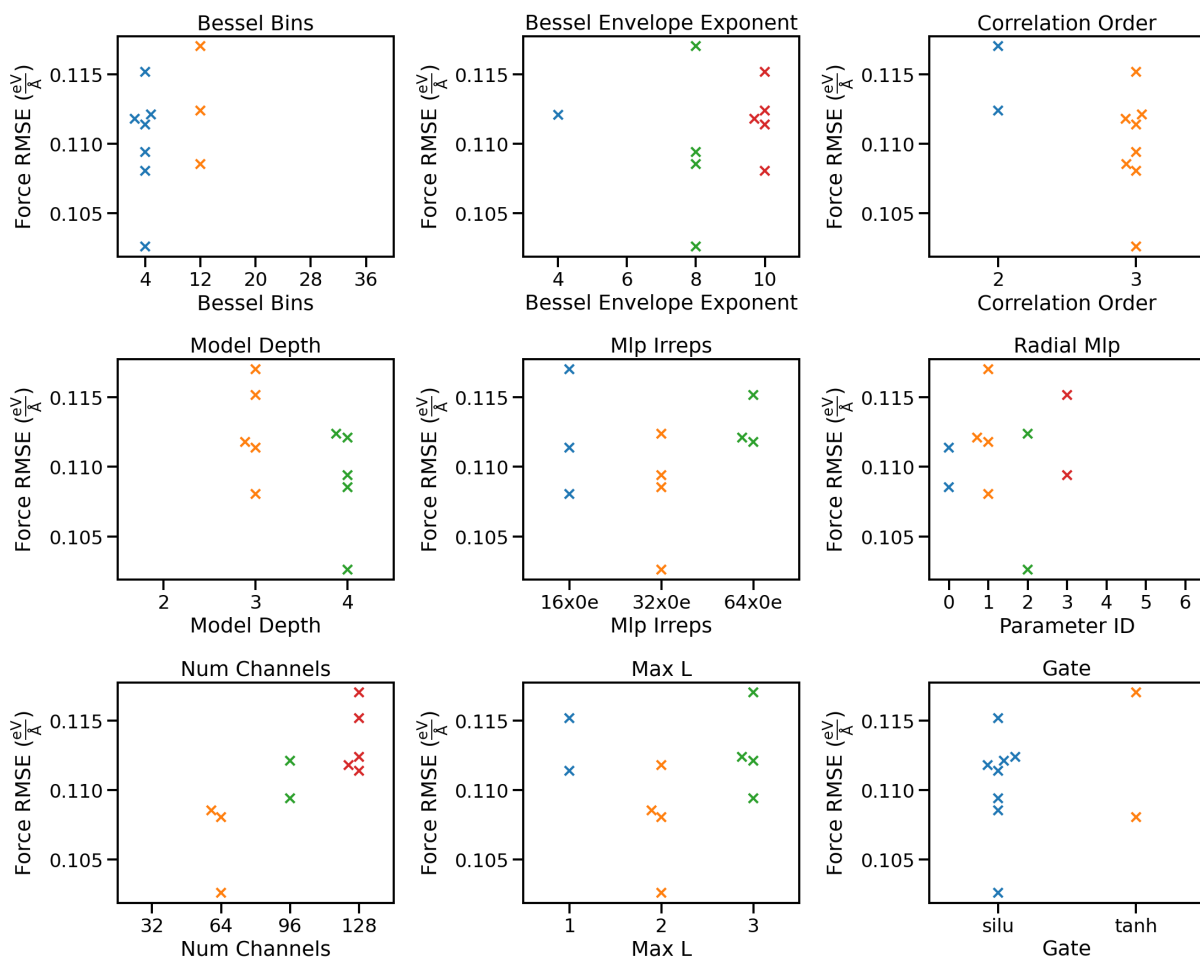


Figure D.41.: Hyperband search hyperparameter overview for Base MACE models from Top 10 trials

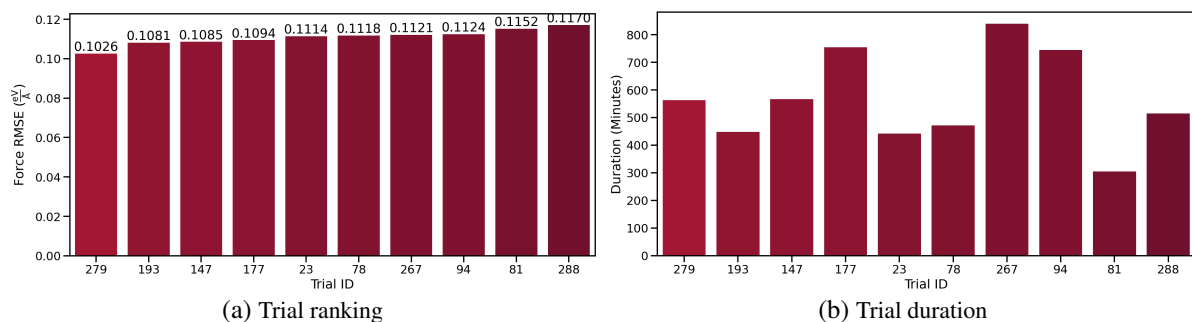


Figure D.42.: Ranking and duration of hyperband search trials for Base MACE models

Table D.20.: Top 3 hyperparameter configurations from the hyperband search for the Base MACE model on the thiol disulfide dataset, ranked by validation force RMSE.

Hyperparameter	Rank 1	Rank 2	Rank 3
MLP irreps	32x0e	16x0e	32x0e
Correlation order	3	3	3
Gate function	SiLU	Tanh	SiLU
$L_{\max}$	2	2	2
Number of channels	64	64	64
Envelope exponent	8	10	8
Interaction blocks	4	3	4
Bessel functions	4	4	12
Radial MLP	[128, 64, 32]	[128, 128]	[128, 128, 64]

D.5. Dataset Size Dependence — Thiol Disulfide Exchange Reaction

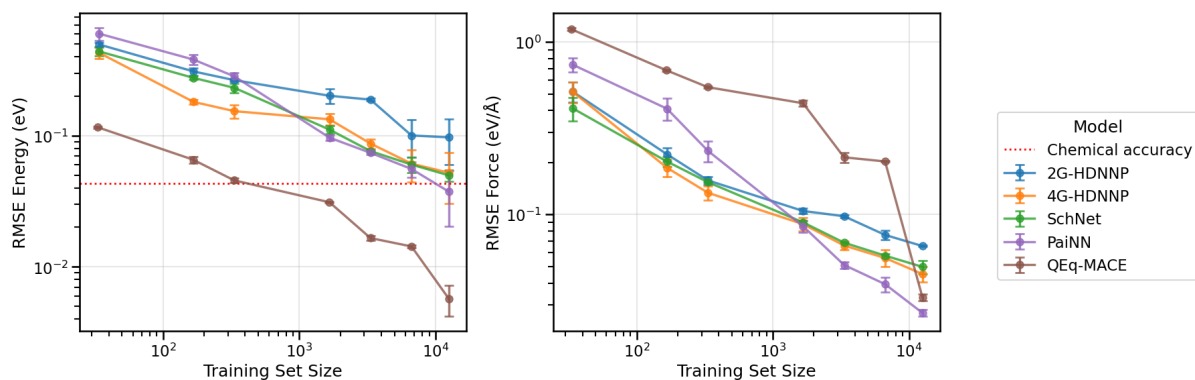


Figure D.43.: Dataset size dependence of different MLP architectures.

D.6. Δ -Learning — Alanine Dipeptide

Table D.21.: Exact hyperparameters used for Δ -training with 2G-HDNNP models for alanine dipeptide.

Hyperparameter	Vacuum	Environment
R_s array	{0.0, 1.0, 2.0, 5.0}	{0.0, 1.0, 2.0, 5.0}
η array	{0.0, 0.03, 0.08, 0.16, 0.3, 0.5}	{0.0, 0.03, 0.08, 0.16, 0.3, 0.5}
λ array	{-1, 1}	{-1, 1}
ζ array	{1, 2, 4, 8}	{1, 2, 4, 8}
Energy hidden layers	35-35	35-35
Energy activation	tanh-tanh	tanh-tanh
Radial cutoff (Bohr)	25	25
Angular cutoff (Bohr)	16	16
Epochs	250	250
Batch size	32	32
Initial / final lr	$5 \times 10^{-4} / 1 \times 10^{-6}$	$5 \times 10^{-4} / 1 \times 10^{-6}$
Early stopping	25	25
Force loss factor	200	200

Table D.22.: Exact hyperparameters used for Δ -training with 4G-HDNNP models for alanine dipeptide.

Hyperparameter	Vacuum	Environment
R_s array	{0.0, 1.0, 2.0, 5.0}	{0.0, 1.0, 2.0, 5.0}
η array	{0.0, 0.03, 0.08, 0.16, 0.3, 0.5}	{0.0, 0.03, 0.08, 0.16, 0.3, 0.5}
λ array	{-1, 1}	{-1, 1}
ζ array	{1, 2, 4, 8}	{1, 2, 4, 8}
Charge hidden layers	15	15
Charge activation	tanh	tanh
Energy hidden layers	35-35	35-35
Energy activation	tanh-tanh	tanh-tanh
Radial cutoff (Bohr)	25	25
Angular cutoff (Bohr)	16	16
Epochs	250	250
Batch size	32	32
Initial / final lr	$5 \times 10^{-4} / 1 \times 10^{-6}$	$5 \times 10^{-4} / 1 \times 10^{-6}$
Early stopping	25	25
Charge / energy / force loss factors	0 / 1 / 200	1 / 1 / 200
Max elements	30	30
Element mapping	{1, 6, 7, 8}	{1, 6, 7, 8}

Table D.23.: Exact hyperparameters used for Δ -training with Base MACE and QEq-MACE models for alanine dipeptide.

Hyperparameter	Base Vacuum	Base Env	QEq Vacuum	QEq Env
Model	ScaleShiftMACE	ScaleShiftMACE	maceQEqESP	maceQEqESP
Hidden irreps	16x0e+16x1o	16x0e+16x1o	16x0e+16x1o	16x0e+16x1o
Correlation order	3	3	3	3
$r_{\max}$ (Å)	5.0	5.0	5.0	5.0
Radial type	Bessel	Bessel	Bessel	Bessel
Num radial basis	8	8	8	8
Num cutoff basis	5	5	5	5
Gate	silu	silu	silu	silu
Loss	energy_forces	energy_forces	charges_energy_forces	charges_energy_forces
Epochs	100	100	100	300
Batch size	10	10	10	10
Loss weights	1 / 100 / 50	1 / 100 / 50	1 / 100 / 1	1 / 100 / 10
Optimizer	adam	adam	adam	adam
lr / swa_lr	$1 \times 10^{-2} / 1 \times 10^{-3}$	$1 \times 10^{-3} / 1 \times 10^{-4}$	$5 \times 10^{-4} / 5 \times 10^{-5}$	$5 \times 10^{-3} / 5 \times 10^{-4}$
Scheduler	ReduceLROnPlateau	ReduceLROnPlateau	ReduceLROnPlateau	ReduceLROnPlateau
lr factor	0.8	0.8	0.8	0.8
Sched. patience	3	3	3	3
Clip grad	10.0	10.0	10.0	5.0
Weight decay	1×10^{-7}	1×10^{-7}	1×10^{-6}	1×10^{-7}

D.7. Δ -Learning — Thiol Disulfide Exchange Reaction

For thiol disulfide Δ -training, the model setups are mostly identical to the alanine dipeptide case and also largely identical between vacuum and environment within each architecture. Only QEq-MACE uses different optimization settings between vacuum and environment (epochs 100 vs. 250, learning rates $1 \times 10^{-4} / 1 \times 10^{-5}$ vs. $1 \times 10^{-3} / 1 \times 10^{-4}$, and charge weight 1 vs. 10). These changes stem from the attempt to reduce the massive overfitting with the QEq-MACE architecture without proper charge targets.

Reference Comparison

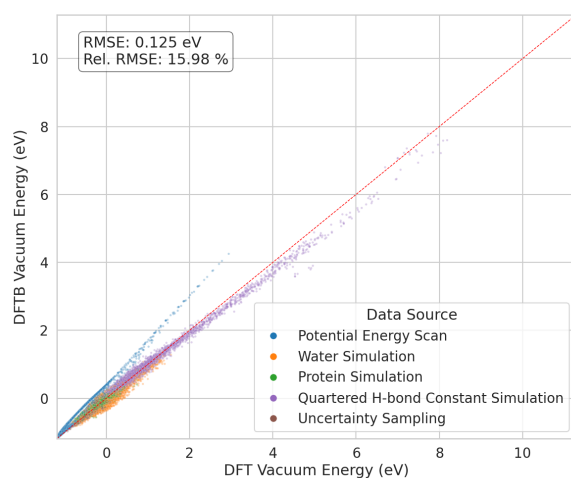
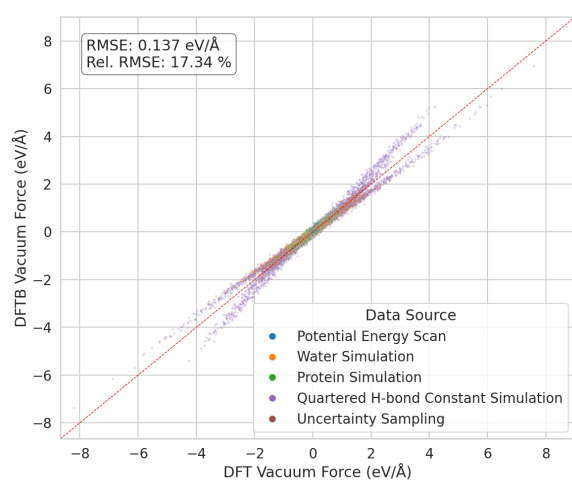
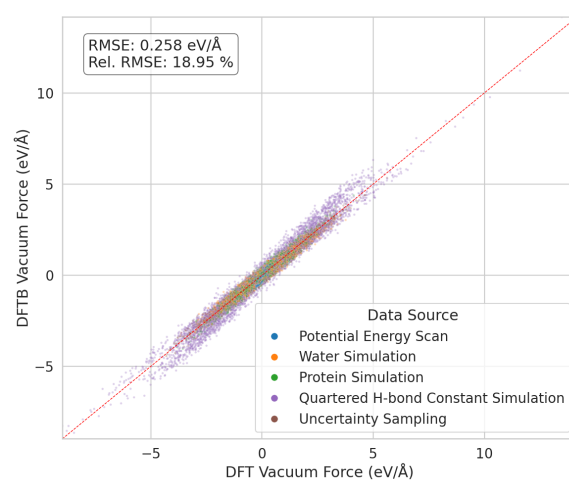


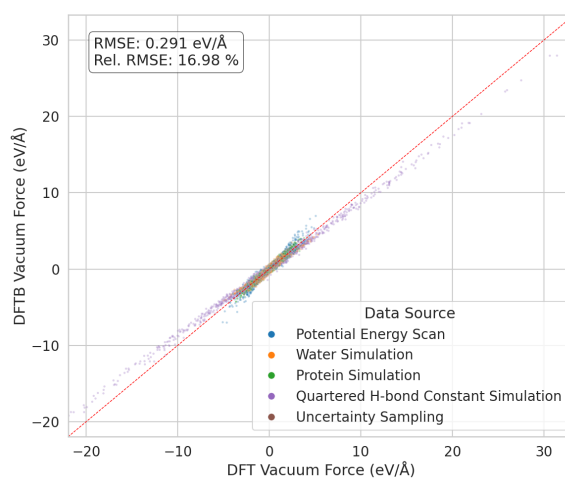
Figure D.44.: Comparison of B3LYP/aug-cc-pVTZ and DFTB3/3OB_{Mod S-S} energies rectified by their respective mean values for thiol disulfide exchange reaction in vacuum.



(a) Hydrogen

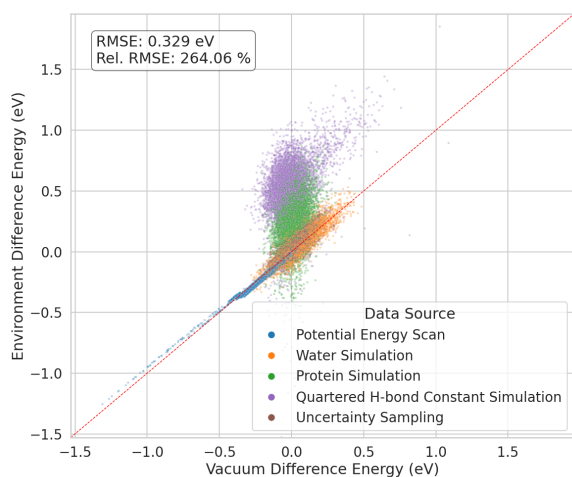


(b) Carbon

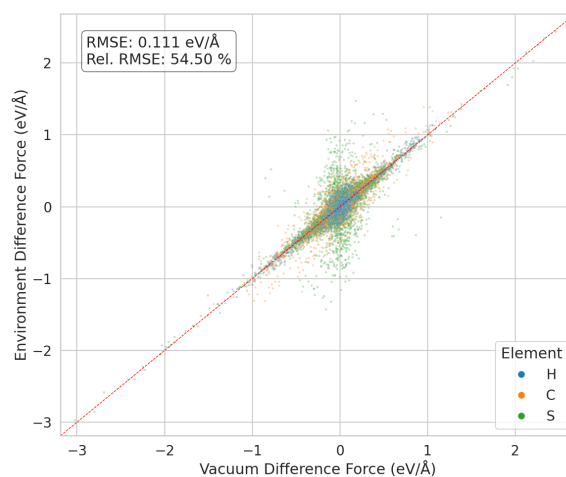


(c) Sulfur

Figure D.45.: Comparison of B3LYP/aug-cc-pVTZ and DFTB3/3OB_{Mod S-S} force components for thiol disulfide exchange reaction in vacuum for different elements.



(a) $(E_{\text{DFT, vacuum}} - E_{\text{DFTB, vacuum}})$ vs $(E_{\text{DFT, solvent}} - E_{\text{DFTB, solvent}})$



(b) $(F_{i,\text{DFT, vacuum}} - F_{i,\text{DFTB, vacuum}})$ vs $(F_{i,\text{DFT, solvent}} - F_{i,\text{DFTB, solvent}})$

Figure D.46.: Correlation between vacuum and solvent errors of energy and force components for the thiol disulfide exchange reaction with B3LYP/aug-cc-pVTZ and DFTB3/3OB_{Mod} S-S. The imperfect correlation indicates that learning the error in vacuum does not necessarily translate to learning the error in solvent.

Δ -Learning Performance

Table D.24.: Test set performance metrics for different MLP architectures using Δ -learning on thiol disulfide exchange reaction. Values represent mean performance across three independent training runs for the vacuum and environment datasets. 2G-HDNNP and Base MACE converge to the same error floor in solvent. QEq-MACE deteriorates substantially in both vacuum and solvent compared to direct training, especially for the energy prediction in solvent.

Δ Vacuum				
Method	Energy		Force	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2
2G-HDNNP	0.042	0.892	0.042	0.957
4G-HDNNP	0.028	0.952	0.031	0.977
Base MACE	0.023	0.968	0.023	0.987
QEq-MACE	0.063	0.763	0.047	0.946

Δ Environment				
Method	Energy		Force	
	RMSE (eV)	R^2	RMSE (eV/Å)	R^2
2G-HDNNP	0.199	0.535	0.091	0.843
4G-HDNNP	0.151	0.734	0.068	0.912
Base MACE	0.193	0.565	0.084	0.869
QEq-MACE	0.322	-0.229	0.104	0.783

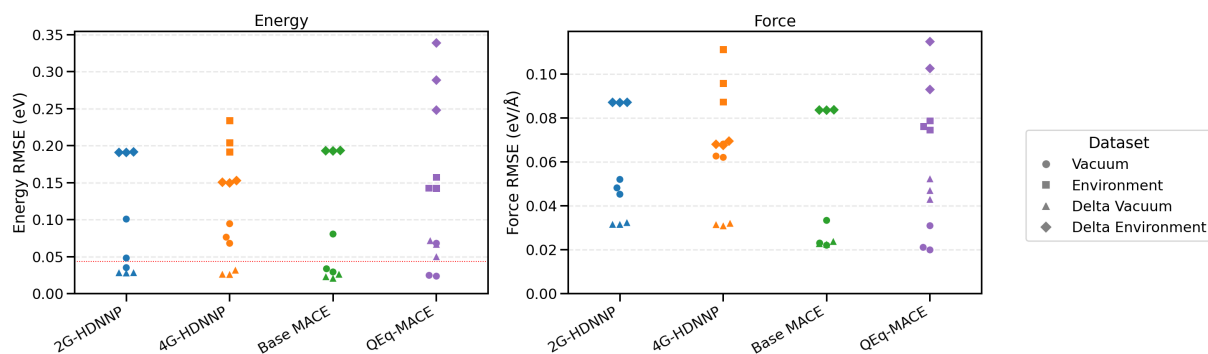


Figure D.47.: Comparison of test set RMSE for different MLP architectures using Δ -learning on thiol disulfide exchange reaction in vacuum and water. HDNNP models improve through Δ -learning both in vacuum and solvent, Base MACE only in solvent. QEq-MACE deteriorates substantially in both vacuum and solvent compared to direct training, especially for the energy prediction in solvent.

Use of Generative AI Tools

In accordance with the guidelines of the Karlsruhe Institute of Technology, this section documents the use of generative artificial intelligence (AI) tools during the preparation of this thesis.

Several generative AI systems were employed to support the research and writing process. These included Claude models (Sonnet/Opus), DeepL Write/Translator, Grammarly, and GitHub Copilot. The tools were primarily used for tasks such as software development support (code suggestions, generation, and debugging assistance), language refinement and stylistic improvement of text passages, and identifying relevant literature.

The use of these tools was strictly assistive. All generated outputs, including code and text, were critically reviewed, verified, and revised by the author before being incorporated into the thesis. Responsibility for the content, accuracy, interpretation of results, and compliance with scientific standards remains solely with the author.

No AI system was used to generate research results, fabricate sources, or replace independent scientific reasoning.