Alma Mater Studiorum · Università di Bologna

DOTTORATO DI RICERCA IN
DATA SCIENCE AND COMPUTATION
Ciclo XXXVI

Settore Concorsuale: 09/H1 – Sistemi di elaborazione delle informazioni

Settore Scientifico Disciplinare: IINF-05/A – Sistemi di elaborazione delle informazioni

Ab initio potentials for atomistic simulations via deep learning

Presentata da: Stefano Martire

Coordinatore Dottorato
Prof. Daniele Bonacorsi

Supervisore
Prof. Andrea Cavalli

Co-supervisore
Dr. Sergio Decherchi

Esame Finale Anno 2025

To Paola, Giovanna, Aldo, and Cesare.

Contents

Chapter 1

List of Acronyms

ACS	American Chemical Society
ACSF	Atom-Centered Symmetry Function
AEV	Atomic Environment Vector
BO	Born-Oppenheimer
CCSD	Coupled Cluster Single-Double-Triple
CELU	Continuously differentiable Exponential Linear Unit
CENT	Charge Equilibration Neural network Technique
DFT	Density Functional Theory
DFTB	Density Functional Tight Binding
DNA	Deoxyribonucleic Acid
DOI	Digital Object Identifier
FCHL	Faber-Christensen-Huang-Lilienfeld
FFNN	Feed-Forward Neural Network

GNN	Graph Neural Network
GPU	Graphics Processing Unit
GS	Ground State
HF	Hartree-Fock
HDNNP	High-Dimensional Neural Network Potential
KS	Kohn-Sham
LAMMPS	Large-scale Atomic/Molecular Massively Parallel Simulator
MBTR	Many-Body Tensor Representation
MD	Molecular Dynamics
MLP	Multi-Layer Perceptron
NNP	Neural Network Potential
PBE	Perdew-Burke-Ernzerhof
PES	Potential Energy Surface
RMSE	Root Mean Square Error
RNA	Ribonucleic Acid
SOAP	Smooth Overlap of Atomic Positions

Chapter 2

Introduction

The estimation of the potential energy of a given molecular configuration is a crucial task in computational chemistry, especially at the quantum level of theory. This involves mapping the coordinates and species of a molecule to its corresponding potential energy scalar value. Such estimations enable the calculation of forces through derivation, facilitating *e.g.* the determination of a molecular ensemble's ground state and making possible molecular dynamics simulations.

Computational techniques for this estimation are categorized based on their theoretical foundations, primarily into quantum methods and molecular mechanics ones. Density Functional Theory (DFT), an exemplar of quantum methods, approximates the solution of the many-body Schrödinger equation. While highly accurate, DFT is computationally intensive, with complexity scaling cubically with the number of *electrons* in the molecule. Conversely, molecular mechanics methods operate at a coarser level, offering rapid approximations of atomic forces with a complexity of $A \log A$, where A represents here the number of *atoms*. This efficiency makes molecular mechanics widely used in simulations, despite its inability to capture electronic phenomena like charge transfer, pH variations, and polarization due to its simplified model.

The recent advancements in parallel computing and high-performance platforms have enhanced the feasibility of large-scale computational quantum chemistry. Although DFT remains resource-intensive, the ability to conduct systematic computations has created new opportunities. The resulting wealth of simulated data now fuels machine learning algorithms capable of effectively approximating original quantum potentials at significantly reduced computational costs. In this context, the high-dimensional neural network introduced by Behler and Parrinello [16] stands as a pioneering work.

Following this initial contribution, neural network potentials have undergone significant enhancements, emerging as a viable alternative for quantum mechanical computations. Within the so called "feed-forward neural network" category, ANAKIN [121, 122, 38], developed by Smith and colleagues, remains a significant benchmark for machine learning potentials, as noted by other researchers [103, 78] too. ANAKIN's importance stems from its generic model trained on millions of molecules in off-equilibrium configurations, comprising H, C, N, O, F, S, and Cl atoms. As such, it serves also as the backbone for several, modern neural network potentials [12, 124], implying that advancements in this approach could benefit related methodologies.

However, the ANAKIN approach faces a possible limitation: the inefficient scaling of neural architecture size with the number of chemical species in the dataset. This inefficiency arises from: (i) the quadratic growth of input features relative to the number of species considered; (ii) the ANAKIN's use of a separate neural model for each species, leading to a possibly significant expansion of the overall network size when incorporating additional elements. Furthermore, the architecture precludes knowledge re-utilization when attempting to include new chemical elements in subsequent training procedures. While these are limitations from this perspective, actually these architectural choices also confer ANAKIN a wide representational ability.

This work introduces OBIWAN, a neural network potential that addresses the aforementioned challenges within the ANAKIN-style class of algorithms: (i) we present a novel, learnable layer capable of generating input vectors with lengths independent of the number of chemical species in the dataset; (ii) leveraging this new input representation, we deploy a single model for energy prediction, capable to interpret every input species. The final result is a neural network architecture with a topology completely independent of the number of chemical species seen during the training phase.

While other attempts have been made in this direction, to our knowledge, none has demonstrated performance comparable to ANAKIN. The SingleNN algorithm by Liu and colleagues [82] shared our objective but employed no new neural module, instead trading off model representation capability for transferability by reduce-summing in the chemical space. Crucially, it was trained and tested on only a few tens of thousands of molecules, insufficient for validating a general-purpose neural network potential. Profitt *et al.* developed a shared-weight, ANAKIN-like neural network potential [104], testing it on 400,000 (equilibrium) structures from QM9 [107]. However, by retaining the Behler and Parrinello molecular description, it did not address the chemical scaling problem that we tackle in this work.

Our closest competitor is the model published by Gastegger et al. [47], which we consider, along with ANAKIN, as our primary benchmark. However, this algorithm was also tested on significantly simpler datasets compared to ANAKIN, comprising just a few thousand molecular configurations, often in equilibrium states [111, 137, 84, 87, 64]. Consequently, a fair comparison between ANAKIN and this method is currently challenging.

The work of thesis is structured as follows: in Chapter 3 we illustrate the theoretical background, starting with the chemical and physical foundation and then passing through related, classical computational methods; in Chapter 4 we present feed-forward neural network potentials' state-of-the-art from which our framework was developed, reminding also possible alternatives; in Chapter 5 and 6 we finally detail and discuss our method and results together with all the open-source software codebase that we developed in order to support our experiments. Chapter 7 is the concluding one, containing also our perspectives on future works.

In accordance with the Terms and Rights in the Journal Publishing Agreement of ACS Publications pertaining the use in thesis of material that I have authored but for which ACS holds copyright [1], the reader will find in this work some of the figures originally published in [90].

Chapter 3

Theoretical background

Computer simulations are nowadays routinely employed to model biological systems at microscopic level, where molecules are represented as interacting entities. These representations can be highly detailed or, particularly for large systems, can employ a coarse-grained approach where a molecule is composed by atoms [65].

More specifically, given that systems of biological interest often comprise large sets of atoms, to describe their behavior one has to leverage statistical mechanics methods which come in two levels of accuracy: the quantum mechanical one, reflecting the fundamental nature of reality taking into consideration every electron, and the classical mechanics one, which approximates molecules as bonded atoms. The latter approach allows to model bigger biophysical systems with reasonable precision when chemical reactivity is not involved (e.g. bond breaking). In the following we introduce basic notions of statistical mechanics, atomistic interaction potentials and molecular dynamics.

3.1 Statistical mechanics

Statistical mechanics is a field that examines the large-scale properties of multi-particle systems by analyzing their microscopic characteristics [26]. In classical mechanics, a molecular system of N particles is described using two sets of variables:

- Positions: $\{\mathbf{r}_1,\ldots,\mathbf{r}_N\} \equiv \left\{\mathbf{r}^N\right\}$;
- Momenta: $\{\mathbf{p}_1,\ldots,\mathbf{p}_N\} \equiv \left\{\mathbf{p}^N\right\}$.

Here, $\mathbf{r}_i$ represents the three-dimensional Cartesian coordinate of the i-th particle. The system's degrees of freedom comprise these positions and momenta, which together form the phase space. A specific point in this phase space, represents a unique microstate of the system. The system's evolution over time is represented by a trajectory $\left\{\mathbf{r}^N(t), \mathbf{p}^N(t)\right\}$, which traces the system's path through phase space.

The total energy of the system is described by the Hamiltonian function (which becomes an operator in quantum mechanics). The Hamiltonian is the sum of kinetic K and potential U energies:

$$H(\mathbf{r}^N, \mathbf{p}^N) = K(\mathbf{p}^N) + U(\mathbf{r}^N) = \sum_i \frac{\mathbf{p}_i^2}{2M_i} + U(\mathbf{r}^N) \tag{3.1}$$

with M_i that is the mass of particle i.

In statistical mechanics, equilibrium macroscopic properties are constant over time. Each such property is calculated as an average $\langle O \rangle$ of a microscopic operator $O(\mathbf{r}^N, \mathbf{p}^N)$ across all possible states. This average is weighted by the probability $\rho(\mathbf{r}^N, \mathbf{p}^N)$ of the system being in each state:

$$\langle O \rangle = \int_{\{\mathbf{r}^N, \mathbf{p}^N\}} O(\mathbf{r}^N, \mathbf{p}^N) \rho(\mathbf{r}^N, \mathbf{p}^N) \, d\mathbf{r}^N d\mathbf{p}^N \tag{3.2}$$

where the integral is taken over all accessible states in the phase space.

Expressing the average $\langle O \rangle$ in this manner assumes that each point in phase space can be accessed with varying probabilities. This allows us to define the already cited time-independent probability density $\rho(\mathbf{r}^N, \mathbf{p}^N)$ for each microstate, whose integral is normalized by definition to 1.
In contrast, experimental measurements provide an average of the same operator $O(\mathbf{r}^N, \mathbf{p}^N)$ over both time and a macroscopic number of particles in the system, a quantity that can be expressed as:

$$\langle O \rangle = \lim_{\tau \to \infty} \frac{1}{\tau - \tau_0} \int_{\tau_0}^{\tau} O(\mathbf{r}^N(t), \mathbf{p}^N(t)) dt \tag{3.3}$$

where τ is the measured time and the integral represents the time average of the operator along a trajectory $\left\{ \mathbf{r}^N(t), \mathbf{p}^N(t) \right\}$.
The link between the time-based and ensemble-based approaches is formed by the so-called *ergodic hypothesis* [2]. This hypothesis suggests that, given an isolated system and an infinite amount of time for sampling, the system's trajectory will eventually pass through all possible microstates. If this hypothesis holds true, the averages calculated over time and those calculated across the ensemble become equivalent, regardless of the initial conditions of the infinitely long trajectory.

3.1.1 The canonical ensemble

If we assume that the system has time-constant number of particles N together with constant volume V and temperature T then the system is said to be in the "canonical ensemble". Its probability density is the Boltzmann distribution:

$$\rho_{NVT}(\mathbf{r}^N, \mathbf{p}^N) = \frac{1}{h^{3N} N!} \frac{e^{-\beta H(\mathbf{r}^N, \mathbf{p}^N)}}{Q_{NVT}} \tag{3.4}$$

where h is the Planck constant, β equal to $\frac{1}{k_B T}$ (k_B is the Boltzmann constant) and Q_{NVT} that is the normalization factor, *i.e.*:

$$Q_{NVT} = \frac{1}{h^{3N} N!} \int_{\{\mathbf{r}^N, \mathbf{p}^N\}} e^{-\beta H(\mathbf{r}^N, \mathbf{p}^N)} d\mathbf{r}^N d\mathbf{p}^N \quad . \tag{3.5}$$

Q_{NVT}, also known as the "partition function", is typically not calculable using Equation 3.5 in practical scenarios. However, its significance in statistical mechanics extends far beyond this limitation. The partition function serves as a cornerstone in understanding ensemble behavior. Theoretically, if one knows Q_{NVT} for a given ensemble, it becomes possible to derive all macroscopic properties of the system. A prime example of this is found in the canonical ensemble, where the Helmholtz free energy, $F(N,V,T)$, is directly related to the partition function [63]:

$$F(N,V,T) = \mathcal{U} - TS = -k_B T \ln Q_{NVT} \tag{3.6}$$

where S is the entropy of the system and $\mathcal{U}$ is the thermodynamic internal energy, representing the total energy contained within a system due to the microscopic kinetic and potential energies of its particles[1].

The connection between a system's observable characteristics and its partition function can be understood at the microscopic level. For example, we can express the internal energy using the following equation:

$$\mathcal{U} = \frac{1}{h^{3N} N! \, Q_{NVT}} \int_{\{\mathbf{r}^N, \mathbf{p}^N\}} H(\mathbf{r}^N, \mathbf{p}^N) e^{-\beta H(\mathbf{r}^N, \mathbf{p}^N)} d\mathbf{r}^N d\mathbf{p}^N \quad . \tag{3.7}$$

As a final note, it is worth reminding that, although in a quantum mechanics scenario the $\mathbf{r}$ and $\mathbf{p}$ operators do not commute, in classical mechanics

[1] Internal energy is primarily concerned with the energy associated with the internal state of the system, and it does not account for the kinetic or potential energy of the system as a whole in relation to external forces.

they do and for this reason they can be decoupled. Also, the integral over momentum of each particle can be computed independently as:

$$\frac{1}{h^3}\int e^{-\beta\frac{\mathbf{p}^2}{2m}}\,d\mathbf{p}^N = \frac{1}{\Lambda^3} \tag{3.8}$$

with Λ that is the de Broglie thermal wavelength of the particle. Thus, in classical mechanics, the canonical partition function becomes:

$$Q_{NVT} = \frac{1}{N!\Lambda^{3N}}\int e^{-\beta U(\mathbf{r}^N)}\,d\mathbf{r}^N = \frac{Z_{NVT}}{N!\Lambda^{3N}} \tag{3.9}$$

where Z_{NVT} is commonly called “configurational partition function”.

3.2 Potential energy surfaces

To discuss potentials we can start from the simple but instructive case of a diatomic molecule. In the classical mechanics case such system is composed of two atoms (two point particles) linked by a spring. As this model suggests, distorting the molecular geometry by stretching or compressing the spring increases system potential energy, which depends on atoms positions. On the other hand, quantum theory offers a more accurate representation of molecules. The uncertainty principle dictates that quantum particles, including atoms, cannot be confined to a single point. Instead, they continuously vibrate around their equilibrium position. Rather than having a specific position and momentum, atoms are described by wave functions. As a consequence, even as temperature approaches absolute zero, molecules retain both kinetic and potential energy.

Mathematically, molecular interactions are modeled using a potential energy surface (PES) [46]. This is a function that relates the potential energy of a system of N atoms to their $3N$ spatial coordinates:

$$U \equiv U(\mathbf{r}^N) \quad . \tag{3.10}$$

To describe how potential energy changes with a system's geometry, researchers use mathematical functions called force fields. As we will see in the next sections, these tools help map the complex landscape of PES across various molecular configurations.

3.2.1 Molecular mechanics potentials

A proper definition of the potential energy surface (PES) is fundamental for accurately simulating condensed matter at atomic or coarse-grained level. Most biological systems comprise organic molecules, characterized by a covalent bond backbone further interacting via relatively strong non-bonded Coulombic forces and weaker, yet pervasive, dispersion ones. This observation allows organic and biological systems to be conceptualized as assemblies of particles (typically atoms) and bonds, forming a network of springs with energy contributions from stretching, bending, and torsion. Additionally, particles carry Coulomb charges and interact through short-range pair potentials representing dispersion non-bonded forces. Short-range repulsion, arising from Pauli's exclusion principle, is also a universal feature of real systems and is incorporated into all atomistic and coarse-grained models.

These qualitative considerations form the basis of the *force field* model, which has proven to be the most successful and widely used approach for simulating biomolecules. In this approximation, the PES, call it U_{FF}, of the molecular system is expressed as a function of nuclear coordinates [81]:

$$U_{FF}(r) = \left(\sum_{pairs} U_{stretch}(r) + \sum_{trip.} U_{bend}(r) + \sum_{quadr.} U_{tors}(r) \right)_{bonded} + \\ + \sum \left(U_e(r) + U_{vdW}(r) \right)_{non-bonded} \quad . \tag{3.11}$$

The first part represents the above mentioned bonded interactions whereas the second the not bonded ones. Under common conditions, covalent bond distances remain nearly constant due to the significant energy required for both compression and stretching. This constancy is further reinforced by quantum effects, which effectively freeze vibrational modes with energies exceeding thermal energy. Consequently, for small deviations from the equilibrium bond distance $\bar{r}$, the stretching energy can be approximated using a Taylor expansion in terms of $r - \bar{r}$. In practice, the stretching term is often simplified to just the quadratic term of this Taylor expansion, providing a computationally efficient approximation that captures the essential behavior near equilibrium:

$$U_{stretch} = \frac{1}{2} k_{stretch}(r - \bar{r})^2 \tag{3.12}$$

where $k_{stretch}$ is the spring constant. It is worth highlighting here that this simplification of the stretching term has two, main, important implications: (i) the model's applicability is limited to conditions close to room temperature and atmospheric pressure; (ii) it cannot accurately represent phenomena involving bond breaking or formation.

The angle bending contribution to the system's potential energy is treated in a manner analogous to the stretching term:

$$U_{bend} = \frac{1}{2} k_{bend}(\theta - \bar{\theta})^2 \tag{3.13}$$

where $\bar{\theta}$ is the equilibrium bending angle. Bond bending energies are significantly higher than typical thermal energies. As a result, many molecular models simplify their approach by assuming fixed bond angles.

In contrast, torsional angles (also known as "dihedrals") are characterized by low internal rotation barriers. This allows for substantial variations in these angles during molecular dynamics. The torsional energy U_{tors} is a periodic function that repeats every 360° rotation of the dihedral angle. Its mathematical expression can take various forms depending on which specific atoms are considered in the calculation. Typically, the torsional energy is modeled using a truncated Fourier series:

$$U_{tors} = \sum_{dihedrals} \sum_{n} \frac{V_n}{2}[1 + \cos(n\phi - \gamma)] \tag{3.14}$$

where V_n is the torsional rotation barrier, ϕ the dihedral angle, and γ the phase angle. The complexity of the torsional potential and the desired level of accuracy determine the number of terms included in the Fourier sum. In practical applications, the sum typically includes terms up to $n = 4$.

As already mentioned, force fields also incorporate non-bonded interactions in calculating the system's potential energy. Starting from electrostatic interactions resulting from non-uniform charge distributions within molecules, the common approach to modeling them assumes that: (i) point charges are located at each atomic site; (ii) the sum of charges in neutral molecules equals zero. Given these assumptions, the electrostatic term is typically represented by a Coulomb potential:

$$U_e = \frac{1}{4\pi\epsilon_0} \frac{q_i q_j}{r_{ij}} \tag{3.15}$$

where q_i and q_j are the atomic charges, r_{ij} is the inter-nuclear distance between atoms i and j and ϵ_0 is the vacuum permittivity.

In actual systems, atomic charge values fluctuate based on the local bonding environment, which changes over time. Additionally, atoms in condensed matter exhibit polarization effects that vary with changing system configurations. These dynamic effects are not explicitly accounted for in "rigid-ion" force fields, which are the type used in the cases considered here.

The final component of the force field is the Lennard-Jones (LJ) potential [69], which models van der Waals interactions. This potential encompasses all non-electrostatic forces and consists of two main parts: (i) a short-range repulsion term, proportional to r_{ij}^{-12}; (ii) a medium-range attractive dispersion term, proportional to r_{ij}^{-6}. We have:

$$U_{vdW} = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{3.16}$$

with ϵ_{ij} that refers to the van der Waals well depth and σ_{ij} to the distance at which the potential is zero. In most cases, the full set of ϵ_{ij} and σ_{ij} is obtained from a more restricted set of homo-nuclear parameters through empirical equations.

Force fields vary in their mathematical formulation and parameter derivation methods [132], with structural parameters like bond lengths and angles determined using computational methods or experimental techniques. Among the various force fields available, AMBER [114] is widely used for simulating biological systems. Other popular force fields include CHARMM [22], OPLS [72], and GROMOS [18]. The utility of these force fields is significantly enhanced by the widespread availability of computer packages optimized for massively parallel architectures, designed to leverage their capabilities.

3.2.2 The Born-Oppenheimer approximation

In the previous section we discussed a first computationally efficient yet approximate description of the potential energy surface. We have to re-mind however that the fully accurate description of molecular interactions is the quantum mechanical one. In the following, we introduce an elegant description of such interactions, dubbed Born-Oppenheimer (BO) approximation, introduced by Max Born and J. Robert Oppenheimer in 1927 [21].

The BO approximation is a fundamental principle in quantum chemistry that allows for a significant simplification of molecular systems still maintaining their quantum description. Its key insight is that the Schrödinger equation, which describes the quantum state of a molecule, can be divided into two separate parts: (i) an electronic equation and (ii) a nuclear one. This separation is based on the vast difference in mass between electrons and nuclei which leads to a significant difference in their velocities, resulting in the much lighter electrons that are assumed to adjust instantaneously to any change in nuclear positions. By decoupling the electronic and nuclear motions, the BO approximation makes it possible to calculate molecular properties and energies more efficiently, paving the way for much of modern computational chemistry.

Given a system, the mathematical starting point is its Hamiltonian, from which one can derive its behaviour through the time-dependent Schrödinger equation

$$i\hbar \frac{\partial \Psi\left(\mathbf{r}^N(t)\middle|\mathbf{R}\right)}{\partial t} = H\Psi\left(\mathbf{r}^N(t)\middle|\mathbf{R}\right) \tag{3.17}$$

or alternatively by means of its time-independent version

$$H\left(\mathbf{r}^N\middle|\mathbf{R}\right)\Psi_\alpha\left(\mathbf{r}^N\middle|\mathbf{R}\right) = E_\alpha \Psi_\alpha\left(\mathbf{r}^N\middle|\mathbf{R}\right) \qquad \alpha = 0, 1, \ldots \tag{3.18}$$

where Ψ is/are the wave-function/s of the system while the notation $\left(\mathbf{r}^N\middle|\mathbf{R}\right)$ means that the function depends on the electrons' coordinates $\mathbf{r}^N$ given the nuclei's ones $\mathbf{R}$. This Hamiltonian can then be divided into nuclear and electronic terms

$$H = H_n + H_e \tag{3.19}$$

and the Hamiltonian of electrons in the "external" Coulomb potential is defined as:

$$H_e = T_e + U_{n,e} + U_{e,e} \quad , \tag{3.20}$$

with T_e that is the kinetic term while $U_{n,e}$ and $U_{e,e}$ are respectively the nucleum-electrons and electrons-electrons potential interactions. Due to the significant mass difference between nuclei and electrons (with the mass ratio M_n/M_e being at least 1800), we can reasonably assume that electrons move much more rapidly than nuclei. This leads to a scenario where nuclei essentially perceive electrons as a negative charge cloud. This cloud's attractive force between the positive nuclei and negative electrons effectively holds the nuclei in relatively fixed positions. Consequently, for any given arrangement of stationary nuclei, we focus on solving the electronic problem. This approach, in theory, yields an infinite set of eigenvalues and eigenvectors, with the associated wave functions ψ_i defined as:

$$H\psi_i\left(\mathbf{r}^N\middle|\mathbf{R}\right) = E_i\,\psi_i\left(\mathbf{r}^N\middle|\mathbf{R}\right) \qquad i = 0, 1, \ldots \tag{3.21}$$

The assumption that electrons evolve rapidly also suggests that they quickly settle into their instantaneous ground state, faster than the nuclei can move. As a result, we only need to consider the electronic ground state ($i = 0$).

When we apply this concept to the initial time-independent Schrödinger equation and project $\Psi_\alpha\left(\mathbf{r}^N\middle|\mathbf{R}\right)$ onto $\psi_0\left(\mathbf{r}^N\middle|\mathbf{R}\right)$, we derive a Schrödinger-like equation satisfied by the nuclear wave function χ_α [20, 101]:

$$\left[T + E_0(\mathbf{R})\right]\chi_\alpha(\mathbf{R}) = E_\alpha\,\chi_\alpha(\mathbf{R}) \tag{3.22}$$

with

$$\chi_\alpha(\mathbf{R}) = \int \Psi_\alpha\left(\mathbf{r}^N\middle|\mathbf{R}\right)\psi_0^*\left(\mathbf{r}^N\middle|\mathbf{R}\right) d\mathbf{r}^N \tag{3.23}$$

in which the asterisk represents the "complex conjugate" operator.

Equation 3.22 describes how the nuclei move on the potential energy surface $E_0(\mathbf{R})$. The Born-Oppenheimer approximation's key insight is this separation of electron and nuclear motion, and the identification of this potential energy surface that governs nuclear movement. When conditions involve low energy and temperature, the parameters $\mathbf{R}$ can be seen as the coordinates of classical particles. Under these conditions, we will write $U(\mathbf{R}) = E_0(\mathbf{R})$.

3.2.3 Ab initio potentials

Recent decades have seen significant progress in developing molecular mechanics force fields optimized for biological systems, balancing accuracy and computational efficiency. However, these empirical models have limitations. Key approximations include the use of fixed atomic charges, which inadequately represents charge polarizability, and the requirement for a fixed molecular topology, which precludes modeling chemical reactions or bond breaking and formation. This constraint necessitates pre-assigning specific protonation states to molecular systems. *Ab initio* potentials, deriving from quantum representations of atoms at the cost of a greater computational demand, try to provide this flexibility.

Biomolecules, like all condensed matter systems, can be conceptualized as assemblies of electrons and atomic nuclei. The just introduced Born-Oppenheimer approximation allows us to treat nuclei as classical particles, while electrons must be described as quantum particles moving within the Coulomb field generated by the nuclei.

Various methods exist to calculate the ground and excited states of N electrons in an external field, such as the so-called Hartree-Fock (HF) method and the Density Functional Theory (DFT) one [70, 127]. While HF includes exchange interactions exactly, it does not consider electron correlation effects, on the other hand DFT is capable of considering such information in an efficient way. In particular, DFT provides a theoretical framework for calculating the ground-state properties of many-electron systems. Instead of solving the full many-electron wave function, DFT focuses on the electron density $\rho_0(\mathbf{r})$, defined as

$$\rho_0(\mathbf{r}) = N \int d^3 r_2 \int d^3 r_3 \cdots \int d^3 r_N |\Psi(r_1, \ldots, r_N, \sigma_1, \ldots, \sigma_N)|^2 \quad , \tag{3.24}$$

where $r = (r_1, \ldots, r_N)$, Ψ is the many-electron wave function and the $\sigma_1, \ldots, \sigma_N$ are the spins.

The first Hohenberg-Kohn theorem proves that the ground-state energy of a system is uniquely determined by its ground-state electron density [77, 118]. The ground-state energy functional of the electrons is expressed as

$$E_0[\rho(\mathbf{r})] = T[\rho(\mathbf{r})] + E_{ee}[\rho(\mathbf{r})] + E_{Ne}[\rho(\mathbf{r})] \tag{3.25}$$

or equivalently

$$E_0[\rho(\mathbf{r})] = F_{\mathrm{HK}}[\rho(\mathbf{r})] + E_{Ne}[\rho(\mathbf{r})] \quad , \tag{3.26}$$

where

- $T[\rho(\mathbf{r})]$ is the exact kinetic energy functional of interacting electrons;
- $E_{ee}[\rho(\mathbf{r})]$ is the electron-electron interaction energy;
- $E_{Ne}[\rho(\mathbf{r})]$ is the interaction energy between electrons and the external potential due to the nuclei;
- $F_{\mathrm{HK}}[\rho(\mathbf{r})]$ is the Hohenberg-Kohn universal functional.

Since the exact form of $F_{\mathrm{HK}}[\rho(\mathbf{r})]$ is unknown, we must rely on approximate functionals. Various strategies exist for this purpose, ranging from the simplest local density approximation to more advanced generalized gradient approximations and, in recent years, machine learning based methods [39].

Minimizing $E_0[\rho(\mathbf{r})]$ with respect to $\rho(\mathbf{r})$ subject to the constraint that the total number of electrons is conserved gives the ground-state energy and density. This problem leads to the Kohn-Sham (KS) equations. The KS theory reformulates the problem through the introduction of a system of non-interacting electrons that reproduces the same ground-state density as the interacting system [77]. The total energy functional in this framework is

$$E[\rho] = T_s[\rho] + \int v_{\text{ext}}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + E_H[\rho] + E_{XC}[\rho] \quad , \tag{3.27}$$

where $T_s[\rho]$ is the kinetic energy of the non-interacting electrons, $E_H[\rho]$ is the classical Hartree energy defined as

$$E_H[\rho] = \frac{e^2}{2}\int d\mathbf{r}\int d\mathbf{r}'\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} \tag{3.28}$$

and $E_{XC}[\rho]$ is the exchange-correlation energy functional, which accounts for all remaining quantum mechanical effects, including many-body electron interactions.

The KS orbitals $\{\psi_i(\mathbf{r})\}$ are orthonormal and used to construct the density:

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} |\psi_i(\mathbf{r})|^2 \quad . \tag{3.29}$$

The KS equations, derived by applying the variational principle to $E[\rho]$, are

$$\left[-\frac{\hbar^2}{2m}\nabla^2 + v_{\text{eff}}(\mathbf{r})\right]\psi_i(\mathbf{r}) = \epsilon_i\psi_i(\mathbf{r}) \tag{3.30}$$

where the effective potential $v_{\text{eff}}(\mathbf{r})$ is given by

$$v_{\text{eff}}(\mathbf{r}) = v_{\text{ext}}(\mathbf{r}) + v_H(\mathbf{r}) + v_{XC}(\mathbf{r}) \tag{3.31}$$

and the ϵ_i are the energies associated with each single KS orbital.

Here $v_H(\mathbf{r})$ is equal to the Hartree potential while $v_{XC}(\mathbf{r})$ is the exchange-correlation potential. The choice of the exchange-correlation approximation defines different density functional variants. A widely used approach is the generalized gradient corrected approximation, with PBE (Perdew-Burke-Ernzerhof) [102] and B3LYP [73] being two prime examples.

While DFT primarily calculates energy and electron density for a given set of nuclear coordinates, its application to geometry optimization and molecular dynamics requires enhanced computational efficiency and stability [92]. The key to improved efficiency lies in recognizing that molecular dynamics and geometry optimization involve energy calculations for a series of closely related atomic configurations. Most energy evaluations can be performed by updating the previous calculation, which is faster than starting anew. This principle underpins modern DFT approaches to electronic structure and total energy computations.

From an application perspective, DFT serves as a method for computing the system's potential energy surface at the quantum level. The eventual underlying molecular dynamics framework (see next section) remains classical.

3.3 Molecular Dynamics

Molecular dynamics (MD) is a computational technique that excels in exploring localized regions of configuration space. It provides insights into a system's temporal evolution and enables the calculation of time-dependent properties.

In MD simulations, trajectories are generated through the step-wise numerical integration of classical Newtonian equations of motion. The system is represented by a collection of atomic coordinates $\{\mathbf{r}_1,\ldots,\mathbf{r}_N\} \equiv \left\{\mathbf{r}^N\right\}$ and their corresponding momenta $\{\mathbf{p}_1,\ldots,\mathbf{p}_N\} \equiv \left\{\mathbf{p}^N\right\}$. The classical equations of motion can be expressed using these variables, where $\mathbf{f}_i$ represents the force acting on the i-th atom, typically derived from the negative gradient of the potential energy function:

$$\mathbf{f}_i = -\frac{\partial U(\mathbf{r})}{\partial \mathbf{r}_i} \quad . \tag{3.32}$$

Calculating the classical trajectory for a system of N particles through analytical methods would require solving $3N$ interconnected, second-order, differential equations, making this approach impractical. To address this challenge, various methods based on time discretization of system evolution have been developed. Finite difference techniques enable the integration of motion equations in stages, separated by a time step δt.

The time step selection for MD simulations typically depends on the system's highest frequency motions. In biological systems, where C-H stretching represents the highest frequency oscillator (period of $10\,\text{fs} = 10^{-14}\,\text{s}$), a time step of 1 fs is commonly used. To increase the time step without compromising simulation accuracy, constraints can be applied to certain internal coordinates using methods like SHAKE [112].

Various algorithms, known as integrators, have been created to solve the equations of motion. These approximations use Taylor expansions for positions, velocities, and accelerations. A notable example is the Verlet integration scheme [133], which utilizes information from the previous time step t to compute new positions at time $t+\delta t$.

$$\mathbf{x}(t+\delta t) = \mathbf{x}(t) + \mathbf{v}(t)\delta t + \frac{1}{2}\mathbf{a}(t)\delta t^2 + o(\delta t^3) \tag{3.33}$$

$$\mathbf{x}(t-\delta t) = \mathbf{x}(t) - \mathbf{v}(t)\delta t + \frac{1}{2}\mathbf{a}(t)\delta t^2 - o(\delta t^3) \tag{3.34}$$

where $\mathbf{a}_i = \frac{\mathbf{f}_i}{m_i}$.

From previous equations we then obtain:

$$\mathbf{x}(t+\delta t) = 2\mathbf{x}(t) - \mathbf{x}(t-\delta t) + \mathbf{a}(t)\delta t^2 + o(\delta t^4) \tag{3.35}$$

with velocity at time t obtained via a two-sided finite difference:

$$\mathbf{v}(t) = \frac{\mathbf{x}(t+\delta t) - \mathbf{x}(t-\delta t)}{2\delta t} \quad . \tag{3.36}$$

For each particle in the system, the Verlet algorithm determines the position at time $t+\delta t$ using three key components: the current position $\mathbf{x}(t)$, the previous position $\mathbf{x}(t-\delta t)$ and the acceleration $\mathbf{a}(t)$, which is derived from the forces acting on the particle. A variant of this algorithm is the leapfrog method which we will employ later [52].

An alternative approach is the velocity Verlet scheme [126], a modification of the original Verlet algorithm. It is a time-reversible and symplectic algorithm, meaning it conserves energy over long time scales, making it well-suited for MD simulations. At each time step Δt, the Velocity Verlet equations for updating positions and velocities are as follows:

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{a}(t)\Delta t^2 \tag{3.37}$$

$$\mathbf{v}\left(t + \frac{\Delta t}{2}\right) = \mathbf{v}(t) + \frac{1}{2}\mathbf{a}(t)\Delta t \tag{3.38}$$

After this half-step for velocities, forces $\mathbf{f}(t+\Delta t)$ based on the new positions $\mathbf{r}(t+\Delta t)$ are computed, from which accelerations can be obtained. Lastly velocities are updated as per:

$$\mathbf{v}(t+\Delta t) = \mathbf{v}\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2}\mathbf{a}(t+\Delta t)\Delta t \tag{3.39}$$

This method provides both accurate energy conservation over time and efficient time integration, making it a popular choice in MD simulations.

Both Verlet and Velocity Verlet methods maintain long-term energy and momentum conservation when using sufficiently short time steps.

In MD simulations of large systems, most computational time is spent evaluating non-bonded interactions. In fact, while bonded interactions scale linearly with system size, achieving linear scaling for non-bonded interactions requires the use of neighbors lists for short-range repulsion and dispersion forces, while Coulomb interactions are calculated using the Ewald sum [80] for smaller systems and mesh-based versions [34] for larger ones. Moreover, periodic boundary conditions are typically employed to support Ewald sums and minimize finite size effects.

The natural ensemble for Hamiltonian dynamics is the microcanonical (NVE) ensemble, which fixes particle number N, volume V and total energy E. However, this ensemble has limited practical application as experiments occur at constant pressure and temperature. To align with experimental conditions, MD can be extended to isothermal-isobaric (NPT) [98] or canonical (NVT) [96] ensembles. These non-Hamiltonian dynamics couple the system to external pressure (barostat) or temperature (thermostat) baths. Thermostats modify equations of motion to achieve canonical ensemble probability density, requiring the definition of instantaneous temperature based on kinetic energy. Common thermostats include Nosé-Hoover [96, 60], Berendsen [17], Langevin [2] and Bussi-Parrinello [23]. Barostats control system pressure by adjusting volume [46].

Chapter 4

Deep learning for ab initio potentials

The significance of computational modeling in chemistry and materials science is growing rapidly, approaching parity with experimental methods. This advancement brings substantial challenges to theoretical approaches as researchers grapple with increasingly complex systems. These complexities manifest in larger system sizes, more diverse compositions, and a wider array of properties under investigation. As a result, scientists must carefully balance their choice of model systems, the properties they aim to study, and the level of accuracy they can feasibly achieve.

A crucial aspect of computational simulations is calculating potential energies and forces for specific atomic arrangements. While electronic structure methods are ideal for moderately sized systems, they quickly become computationally expensive for larger ones. Even with efficient techniques like density functional theory (DFT), which forms the basis of *ab initio* molecular dynamics (MD) [92], most research questions cannot be directly addressed using first-principle methods due to the aforesaid limitations.

The landscape of mathematical potentials has been dramatically transformed with the emergence of modern machine learning (ML) techniques, which significantly broadened the applicability of the former and expanded their reach in modeling complex systems. ML methods have now become commonplace across various disciplines including chemistry, physics, and materials science [25, 116, 75], with the 1995 groundbreaking work by Blank and colleagues [19] marking the beginning of ML's application in constructing atomistic potentials. Since then, numerous reviews have been published discussing ML potentials and their applications in computational simulations [14, 40, 95]. Also in recent years there has been a proliferation of ML potentials encompassing a wide range of approaches. Some examples are: Neural Network Potentials [88, 106, 16, 115, 105], Gaussian Approximation Potentials [8], Spectral Neighbor Analysis Potentials [36], Atomic Cluster Expansion [41], Kernel Ridge Regression Methods [113] and Support Vector Machines [134]. Each of these approaches leverages different ML techniques to model the complex relationships between atomic positions and potential energy, offering various trade-offs in terms of accuracy, efficiency, and applicability to different types of systems.

In this thesis we will refer to the so-called "high-dimensional" ML potentials, defined here to meet the following criteria [13]:

1. Scalability: the potential should be applicable to systems with a very large number of atoms, typically in the range of thousands;

2. Comprehensiveness: the potential should explicitly depend on all degrees of freedom within the system.

It's important to note that this definition is not exclusive, and other equally valid definitions exist in the field. This definition emphasizes the ability of these potentials to handle complex, large-scale systems while maintaining essential physical properties and accounting for all possible system variables, setting a high bar for what constitutes a truly high-dimensional ML potential in the context of atomistic simulations.

In the domain of condensed matter, mathematical potentials typically employ established machine learning methods like neural networks or kernel ridge regression, enabling simulations of systems with very large numbers of atoms. The use of ML methods for constructing these interatomic potentials offers several advantages:

1. Flexibility: their nonlinear functional form allows for highly accurate representation of reference data from electronic structure calculations;

2. Efficiency: they can be computed orders of magnitude faster than even relatively quick electronic structure methods like DFT;

3. Unbiased Approach: they don't require prior knowledge or assumptions about functional forms, allowing all types of atomic interactions to be described without bias and with consistent accuracy;

4. Universal Approximation: in the case of neural networks, it has been independently proven by several groups that they are universal approximators. This means they can, in principle, approximate any multidimensional real-valued function such as PESs with arbitrary precision [33, 62, 61].

These characteristics make ML potentials powerful tools for modeling complex atomic systems, bridging the gap between classical efficiency and quantum accuracy. [12]

4.1 Feed-forward neural networks

A pioneering effort in ML PES estimation was made by Sumpter and Noid in 1992 [125], which focused on macromolecules and training a neural network on vibrational spectra obtained from force fields. However, the advent of modern ML potentials, based on electronic structure calculations as defined before, has to be traced to the groundbreaking work by Blank and colleagues in 1995 [19]. In fact, this seminal paper introduced the first density functional theory DFT-based ML potential using H_2 adsorption on a Si(100) surface, providing a comprehensive analysis of the benefits and challenges of employing neural networks for PES representation and offering solutions that remain relevant today. The core of this initial ML potential was a feed-forward neural network (FFNN), which became the standard feature of almost all neural network potentials (NNPs) developed by various research groups in subsequent years.

Figure 4.1 illustrates the architecture of a feed-forward neural network, also known as Multi-layer Perceptron (MLP) [110], for a potential energy surface based on two input features x_1 and x_2 (for example some cartesian coordinates) which correspond to neurons (or nodes) in the input layer. The output layer's single node represents the system's potential energy, E, as a function of these features. Between the input and output layers there're can be one or more hidden layers (two in our case). The nodes in these hidden layers lack physical significance but provide the network's functional flexibility; increasing the number of hidden layers or the numbers of their neurons enhances the FFNN's capacity to fit complex functions.

As depicted by arrows in Figure 4.1, neurons in each layer connect to those in adjacent layers through weights, which serve as the model's fitting parameters. The notation w_{ij}^{kl} represents the weight connecting neuron i in layer k to neuron j in layer $l = k + 1$. Here, layers are numbered starting with 0 for the input layer while neurons in each layer are indexed starting

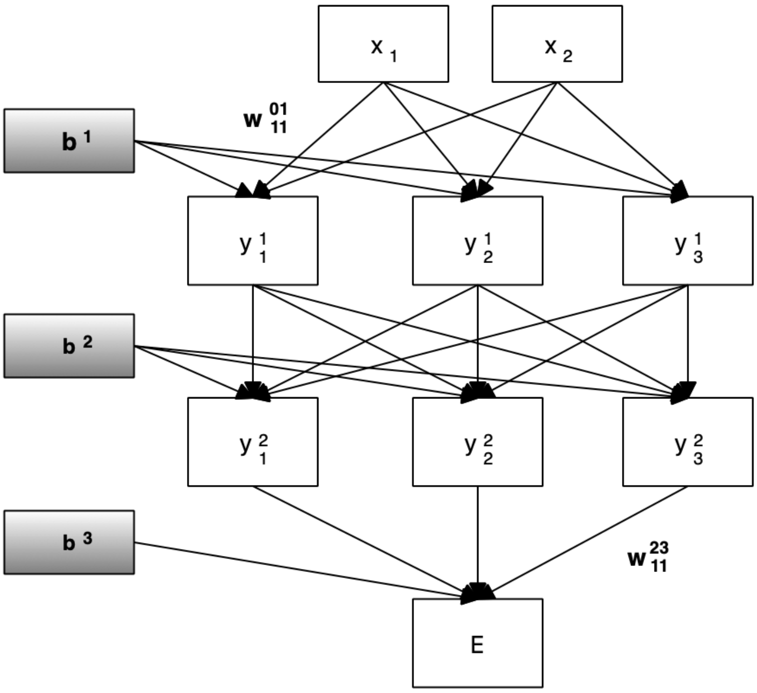


Figure 4.1: Computational graph of a small feed-forward neural network. A complete explanation of its analytic functional form (Equation 4.1) can be found in the main text.

from 1. Additionally, each node in the hidden layers and the output node connects to a bias node, providing an input of 1 scaled by a bias weight b_i^j, targeting neuron i in layer j (in Figure 4.1 those nodes are grouped in one vector per layer). The bias nodes work as adjustable offsets for each neuron, which is numerically advantageous when applying the nonlinear activation functions discussed below.

The FFNN's analytic form is determined by its architecture, *i.e.* by the number of hidden layers and relative nodes. For the simple network depicted in Figure 4.1, the analytic energy expression is given by:

$$E = f_1^3\left(b_1^3 + \sum_{l=1}^{3} w_{l1}^{23} f_l^2\left(b_l^2 + \sum_{k=1}^{3} w_{kl}^{12} f_k^1\left(b_k^1 + \sum_{j=1}^{2} w_{jk}^{01}\, x_j\right)\right)\right) \quad . \tag{4.1}$$

This nested function first computes a linear combination of the two input coordinates at each neuron k of the first hidden layer, using the connecting weights w_{jk}^{01} as coefficients. The result is then shifted by the respective bias weight b_k^1 and, subsequently, a nonlinear activation function f_k^1 is applied to each shifted sum, making the FFNN a nonlinear model capable of representing arbitrary functions. Various differentiable activation functions can be used for constructing PESs, provided they have a nonlinear region and saturate for extreme positive and negative arguments. Common examples include the hyperbolic tangent, sigmoid function, softplus function and Gaussians. The output node is an exception, typically using the linear function to avoid restricting the range of possible output energies.[2]

The activation function's application yields a value y_k^1 for each neuron in the first hidden layer. This process then repeats for subsequent layers, with values in the second layer calculated similarly, and so on until reaching the output node $E = y_1^3$ which in turn results in an output energy dependent on the connecting and bias weights determined through a training process utilising a known reference dataset as mentioned before. This one-way information flow gives rise to the term "feed-forward" neural network.

The just described vanilla architecture already offers a viable route for constructing potential energy surfaces [19]. Its numerous fitting parameters allow for a precise representation of potential energy based on atomic positions; it can describe all bonding types equally without requiring specific physical knowledge; it enables calculation of analytic derivatives, useful

[2] Also, while other activation functions like the rectified linear unit (ReLU) [138] are popular in neural networks for tasks such as classification, some of these functions can be problematic for representing continuous functions like PESs due to discontinuities in their derivatives.

for computing forces and optimizing weight parameters; energy and force calculations are significantly faster than even efficient electronic structure methods like DFT. However, these “first-generation” NNPs [12] had limitations too: increasing FFNN size for larger systems becomes computationally expensive and challenging to train; suitable input features incorporating translational, rotational, and permutational invariances are not straightforward to obtain (and the model itself does not benefit of these mathematical properties); a single FFNN cannot handle variable numbers of atoms, as its input vector size must be fixed during training. These limitations have been overcome through three key developments:

1. Recognizing that a significant portion of atomic interactions can be described through atoms’ interactions with their local chemical environments;
2. Developing a new type of descriptor with translational, rotational and permutational invariance;
3. Utilizing active learning to construct training sets in high-dimensional configuration spaces.

A crucial advancement was moving away from using a single neural-network forward-pass for the global potential energy surface (PES). Instead, the approach now exploits the locality of atomic interaction energies, which has proven surprisingly effective for many systems. This locality approximation, introduced by Behler and Parrinello in 2007 [16], has since become the foundation for many modern ML approaches. Under this framework, the potential energy is expressed as a sum of atomic energy contributions

$$E = \sum_{i=1}^{N} E_i \tag{4.2}$$

where N the number of atoms of the same species (this network model supports one chemical element only). The resulting energy is often referred to as "short-range" because every E_i is obtained as a function of the interaction between the central atom i and all its neighbours within a certain cutoff radius R_c, regardless of the physical nature of these interactions. While truncating atomic interactions beyond the cutoff is a significant approximation, cutoffs between 6 and 10 Å often prove sufficient for many systems, yielding reliable potential-energy surfaces with total energy errors of only a few meV/atom.

After defining interacting atoms via a cutoff radius, the next crucial step is converting structural information into suitable input for the neural network potential. This was achieved by introducing atom-centered symmetry functions (ACSFs) [9], a new type of descriptor that enabled the construction of machine learning PESs that precisely adhere to translational, rotational, and permutational invariance conditions for any system. By combining the locality approximation with ACSF descriptors, a separate, neural forward-pass can be used for each atom in the system to express atomic energy contributions.

The process for each atom i begins with its Cartesian coordinate position vector $\mathbf{R}_i$. These coordinates are transformed into a vector of symmetry function values $\mathbf{G}^i$ (see later for full details), which depends on the Cartesian coordinates of all neighboring atoms within the atomic cutoff sphere. This atomic symmetry function vector serves as input for an atomic forward-pass in the network.

Because of their inherent architecture these "second-generation" NNPs [12] are applicable to systems with any number of atoms. Adding or removing an atom simply involves including or deleting the corresponding neural pass in the scheme. This overcomes the limitation of most first-generation NNPs to systems with a fixed number of atoms, allowing for training on structures with varying atom counts and application to much larger systems than those used for the weights determination.

This scheme also achieves exact permutation invariance for two reasons: (i) the sum of atomic energies E_i in Equation 4.2 is independent of atom order; (ii) as will be further discussed, the ACSF input vectors are invariant to the order of the neighboring atoms.

After this discussion on the total energy expression, we now focus on another key step in constructing HDNNPs: selecting descriptors for the inputs. Today, numerous choices are available, each with its own strengths and weaknesses. They include: Smooth Overlap of Atomic Positions (SOAP) [7], Coulomb matrix [113], Spherical harmonics-based descriptors [68], Faber-Christensen-Huang-Lilienfeld (FCHL) descriptors [28, 27], Bag of bonds [55], Many-Body Tensor Representation (MBTR) [64] and Polynomials in moment tensor potentials [117] and the above mentioned ACSF. While a comprehensive is beyond the scope of this thesis, various comparisons can be found in the literature [141].

The primary challenge in developing suitable inputs for high-dimensional systems has been meeting the mandatory invariances of the potential energy surface. These invariances include:

1. Translational invariance;
2. Rotational invariance;
3. Permutational invariance, *i.e.* the one pertaining the order in which atoms are indexed.

This challenge arises because NNPs process numbers, and changes in input numbers alter output energies. This becomes problematic when numerical changes in input don't correspond to physically meaningful structural changes in the system. For example, many common coordinates, particularly Cartesian coordinates, often lack physical meaning in their absolute values: translating or rotating a rigid molecule in vacuum doesn't change relative atomic positions, so the energy should remain constant; however, Cartesian coordinate values would change, making them unsuitable as input for ML potentials.

A straightforward solution for translational and rotational invariance in descriptor-based methods would be using internal coordinates (interatomic distances, angles, dihedrals, ...), as successfully employed in classical force fields [128]. However, this solution only partially addresses the permutational invariance requirement, working well for simple force fields but not for many-body potentials like ML algorithms that couple all coordinates in a single functional expression. Moreover, neural networks require a fixed-size input vector, making the use of internal coordinates for variable numbers of atoms particularly challenging.

Again in 2007 from Behler and Parrinello, atom-centered symmetry functions (ACSFs) were introduced as a new class of descriptors to address these issues [16, 9]. ACSFs describe the positions of atoms surrounding a central atom within a defined cutoff radius R_c. This cutoff approach effectively reduces the complexity of atomic environments to the positions of nearby atoms, enabling the use of the energy expression in Equation 4.2.

The cutoff function $f_c(R_{ij})$, where R_{ij} represents the distance between the central atom i and a neighboring atom j, can take various functional forms [9, 120]. However, certain criteria must be met: the function must be differentiable and decay smoothly to zero in value, slope and preferably higher derivatives at the cutoff radius. This smooth decay is crucial to avoid discontinuities in energy and its gradients when atoms enter or leave the cutoff spheres *e.g.* during molecular dynamics simulations. A commonly used cutoff function is the monotonously decaying portion of the cosine function [16] shown in Figure 4.2:

$$f_C(R_{ij}) = \begin{cases} \frac{1}{2}\cos\left(\pi\frac{R_{ij}}{R_c}\right)+\frac{1}{2} & \text{if } R_{ij} \le R_c \\ 0 & \text{if } R_{ij} > R_c \end{cases} \tag{4.3}$$

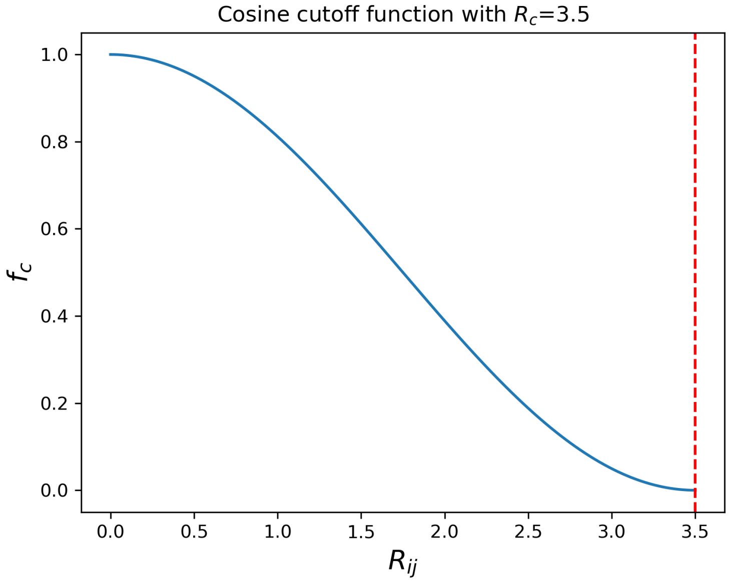


Figure 4.2: Monotonously decaying cosine cutoff function.

It has continuous values and slope at the cutoff radius. Also expressions based on hyperbolic tangents, exponentials or polynomials had been proposed [9, 120].

The next phase involves describing the positions of nearby atoms within the cutoff sphere. This is accomplished using two categories of symmetry functions: "radial" and "angular" ACSFs. They depend on the positions of all neighboring atoms within the cutoff sphere making them many-body functions, although they effectively consist of numerous two- and three-body terms.

Various functional forms are available for each type [9, 10]. These functions must meet several criteria: (i) they should decrease in value for distant neighbors near the cutoff radius, reflecting diminishing physical interaction and facilitating HDNNP representation; (ii) they need to differentiate different structures; (iii) the number of symmetry functions describing an atomic environment must remain constant regardless of changes in atomic coordination during simulations, as neural networks require fixed input dimensionality.

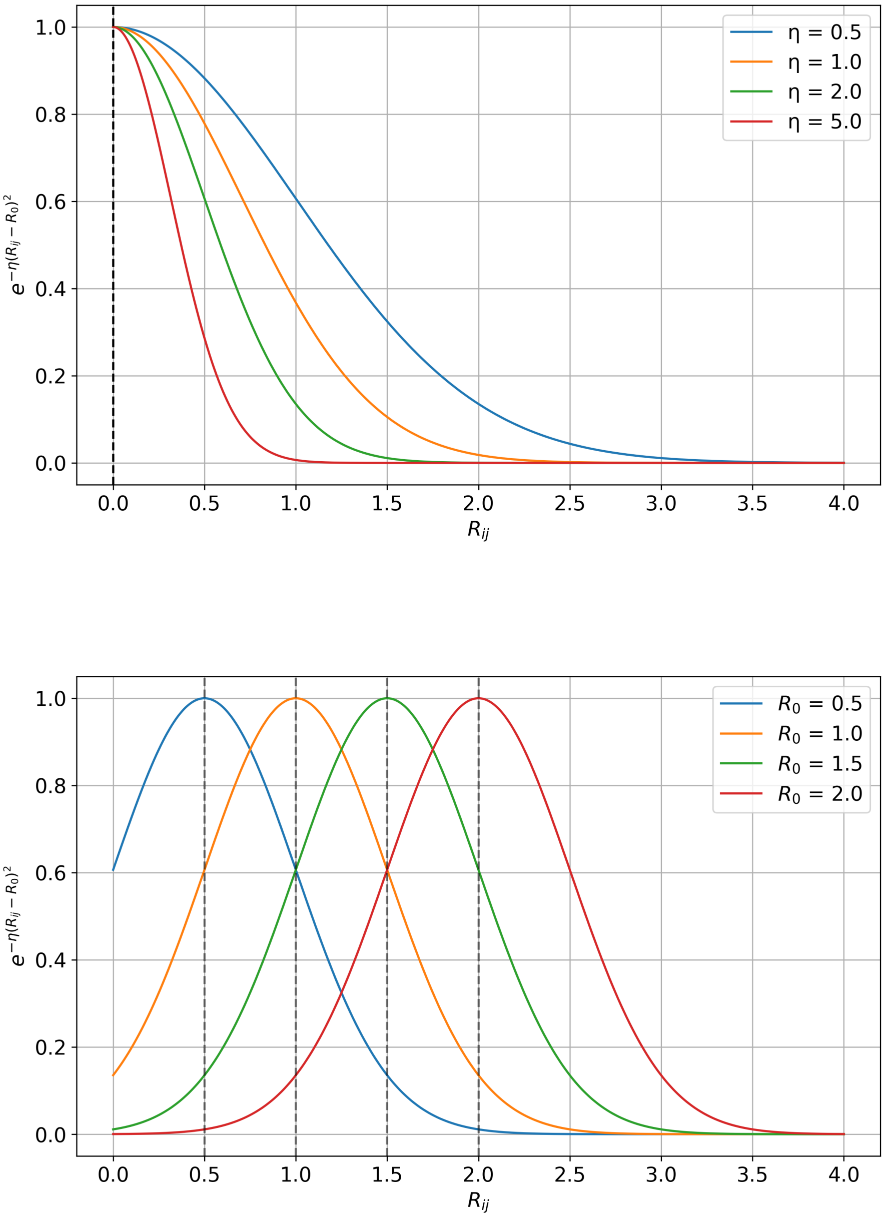
The most common radial function is a sum of Gaussian and cutoff function products for all atoms within a cutoff sphere:

$$G^i_{l(\eta,R_0)} = \sum_{j\neq i}^{\text{all atoms}} e^{-\eta(R_{ij}-R_0)^2} f_c(R_{ij}) \tag{4.4}$$

Using a Gaussian instead of the direct distance R_{ij} between the central atom i and neighbor j, combined with the cutoff function, ensures the required decay to zero in both value and slope at the cutoff radius. Summing the Gaussian functions of all neighboring atoms condenses the information into a single function value, independent of the number of atoms inside of the neighbours' sphere. The parameter η determines the effective spatial range of the radial function, while the maximum range is set by the cutoff radius. An optional shift parameter R_0 can be used to move the center of the Gaussians to a specific distance from the central atom. Figure 4.3 shows plots of the radial function from Equation 4.4 for both unshifted (a) and shifted (b) cases, with varying values of η and R_0 respectively. Using a set of these functions with different combinations of parameter values ensures good radial resolution.

All the real-values $G^i_{l(\eta,R_0)}$ functions obtained by varying the (η, R_0) hyperparameters are then concatenated to form the final $\mathbf{G}^i$ vector, which is the radial ACSF input of the neural model for the atom i. This meets the requirement for a fixed input vector size in atomic neural networks.

Radial functions alone are insufficient to differentiate between certain distinct atomic environments. For instance, they cannot distinguish between a tetrahedral coordination and a square planar one if all neighboring atoms are equidistant from the central atom. To address this limitation, angular



(a)

(b)

Figure 4.3: Behaviour of the Gaussian in Equation 4.4 varying (a) the inverse of the variance η and (b) the center R_0.

functions are introduced.

The most frequently used angular ACSF is expressed as:

$$H^i_{l(\zeta,\lambda,\eta,R_0)} = 2^{1-\zeta} \sum_{j<k\neq i}^{\text{all pairs}} (1+\lambda\cos\theta_{ijk})^{\zeta}\, e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} \times \\ \times f_c(R_{ij})\, f_c(R_{ik})\, f_c(R_{jk}) \tag{4.5}$$

In this equation, θ_{ijk} represents the angle formed between the vectors connecting the central atom i to its neighbors j and k. For simpler atomic environments, the Gaussian term typically uses $\eta = 1$. However, for more complex environments, multiple shells of angular functions can be created by employing different η values. Angular resolution is achieved by using a set of different ζ exponents, while the $2^{1-\zeta}$ serves as a normalization factor. The λ parameter, which can be either $+1$ or -1, is used to center the cosine term maxima at either $\theta_{ijk} = 0°$ or $\theta_{ijk} = 180°$ respectively.

In Equation 4.5, using all three pairwise cutoff functions for R_{ij}, R_{ik}, and R_{jk} ensures that only terms where all three distances are smaller than the cutoff radius contribute to the sum over all angles. This typically results in angular functions having a substantial number of terms less than one, leading to a smaller range of function values compared to radial functions. Consequently, it's common practice to rescale each symmetry function's value range to the interval $[0,1]$.

An alternative, less restrictive angular function can also be used:

$$H^i_{l(\zeta,\lambda,\eta,R_0)} = 2^{1-\zeta} \sum_{j<k\neq i}^{\text{all pairs}} (1+\lambda\cos\theta_{ijk})^{\zeta}\, e^{-\eta(R_{ij}^2+R_{ik}^2)} \times \\ \times f_c(R_{ij})\, f_c(R_{ik}) \tag{4.6}$$

This function includes a larger number of angle terms by removing the $R_{jk} \leq R_c$ restriction. This allows for a more comprehensive description of the angular environment, potentially capturing subtler structural features.

As before, all the real values $H^i_{l(\zeta,\lambda,\eta,R_0)}$ obtained by varying the corresponding hyper-parameters are then concatenated to form the final $\mathbf{H}^i$ vector, *i.e.* the angular ACSF input of the neural model for the atom i.

Equation 4.2's energy expression allows for the calculation of analytic derivatives, which are essential for determining atomic forces in applications like molecular dynamics simulations or geometry optimizations. The force component $F_{i\alpha}$ acting on atom i in direction α (where α can typically be x, y or z) as usual is the negative derivative of E with respect to the coordinate $R_{i\alpha}$:

$$F_{i\alpha} = -\frac{\partial E}{\partial R_{i\alpha}} = -\sum_{i=1}^{N} \frac{\partial E_i}{\partial R_{i\alpha}} \tag{4.7}$$

The chain rule has to be employed in order to take into account the transformation from Cartesian coordinates to atom-centered symmetry functions. As a result, we will have a first set of derivatives (atomic energy with respect to symmetry functions) determined by the atomic neural networks' architecture and a second set (symmetry functions with respect to Cartesian coordinates) defined by the functional forms of the ACSFs. This approach also allows for analytic calculation of other gradient-related properties, such as the stress tensor [9] and the Hessian.

The relationship between E and $F_{i\alpha}$ in Equation 4.7 is exact, ensuring consistency between energies and forces. This has interesting implications for the environment-dependence of forces [11], as illustrated in Figure 4.4: in fact, while atomic energies only depend on neighboring atom positions within the cutoff radius, forces depend on the atomic energies of all atoms within this radius; thus, since these atomic energies depend on their own local environments, forces can effectively be influenced by atom positions up to twice the cutoff radius around the central atom i.

It is worth repeating that in HDNNPs atomic forces are not obtained as independent additional output neurons. Thus, being them calculated

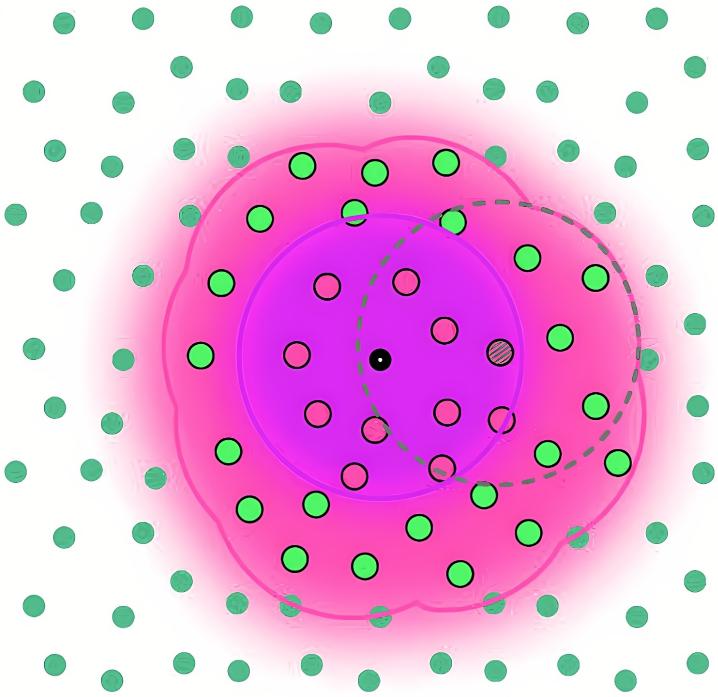


Figure 4.4: While atomic energies only depend on neighboring atom positions within the cutoff radius, forces depend on the atomic energies of all atoms within this radius; thus, since these atomic energies depend on their own local environments, forces can effectively be influenced by atom positions up to twice the cutoff radius around the central atom i.

as analytic derivatives of the energy, they depend on the same weight parameters of the atomic energy contributions. As will be discussed, this dependence allows for the use of force components, in addition to energies, during the network's training.

4.2 ANAKIN

Within this category of feed-forward neural networks, ANAKIN [121, 122, 38] stands out as a notable example. Developed by Smith et al., this method continues to serve as a significant benchmark for machine learning potentials in this class, as acknowledged by various researchers [103, 78]. ANAKIN's importance stems from its efficient, versatile design trained on an extensive dataset of millions of molecules in non-equilibrium states, encompassing H, C, N, O, F, S, and Cl atoms. Furthermore, it forms the foundation for numerous contemporary feed-forward networks [12, 124] and, consequently, any enhancements to this approach could potentially benefit related methodologies.

Figure 4.5 illustrates the computational graph of the ANAKIN architecture. The model is a significant enrichment of the previously discussed Beheler and Parrinello network model. One of the most relevant differences is that now the network can deal with many elements types hence is a universal approximator. To support this flexibility several network changes are necessary. Again the model calculates an "atomic energy" for each atom, which represents that same atom's contribution to the system's global energy based on its local environment. However this time one has a distinct network for each element type. These atomic energies are then summed to predict the total molecular energy. The architecture comprises three main components: (i) the Atomic Environment Vectors (AEVs) Computer, a non-learnable featurizer; (ii) the elemental MLPs block, which determines the atomic energies; (iii) the polynomial self-energies block, which applies

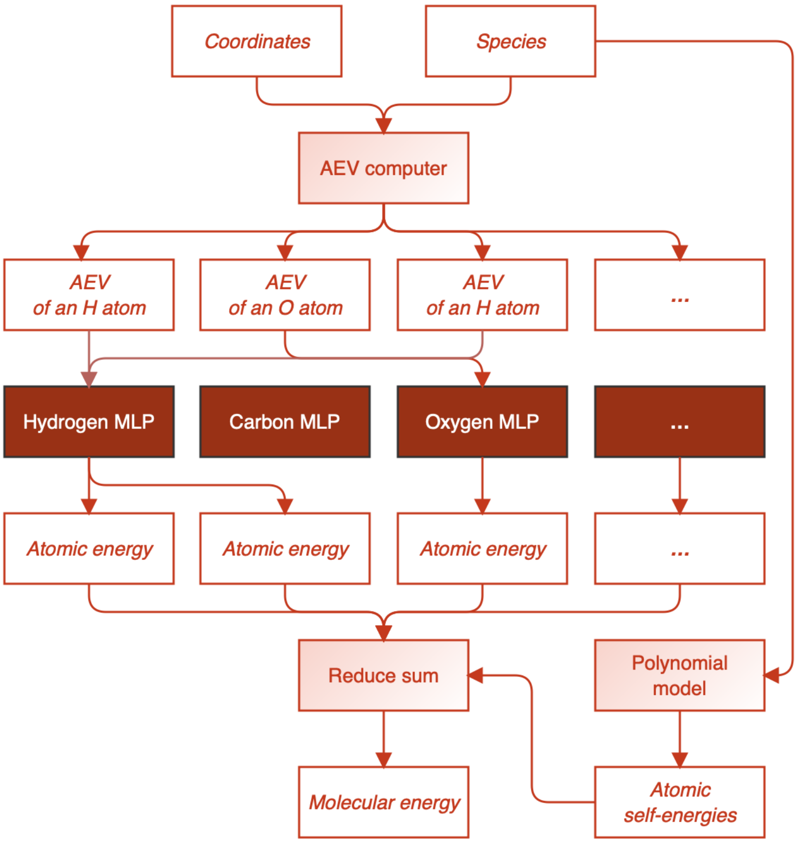


Figure 4.5: The ANAKIN computational graph.

a constant shift to each atomic energy based on the atom type, facilitating the learning process of the MLPs.

The AEV Computer takes as input a pair of sets containing the coordinates and the chemical identities of every atoms in the molecule. It outputs a set of AEVs, each describing the local environment of the relative atom. Figure 4.6 shows an example AEV for the i-th atom in a dataset containing only H, C, N, and O.

The AEV of the i-th atom consists of two parts: radial and angular. The radial part encodes 2-body information for all atom pairs including the central atom i employing the following, atom type enriched ACSF:

$$G^i_{l(\eta,R_0,\Phi)} = \sum_{j\neq i}^{\text{all }\Phi\text{ atoms}} \exp(-\eta(R_{ij}-R_0)^2)\, f_C(R_{ij}) \quad . \tag{4.8}$$

Here, Φ represents a chemical species, R_{ij} is the distance between atoms i and j and η and R_0 are the hyperparameters discussed in the previous section. Since the above radial function can't differentiate between different chemical elements in the neighborhood, a separate set of radial functions is created for each chemical species in the system represented by the Φ parameter. As a result, the number of radial functions for a given central atom increases linearly with the number of chemical species in the system. $f_C(R_{ij})$ is the monotonously decaying portion of the cosine function discussed in Equation 4.3. Albeit already discussed we report it also here for the sake of completeness:

$$f_C(R) = \begin{cases} \frac{1}{2}\cos\left(\frac{\pi R}{R_C}\right) + \frac{1}{2} & \text{if } R \leq R_C \\ 0 & \text{if } R > R_C \end{cases} \quad . \tag{4.9}$$

The angular part encodes 3-body information for all triplets including the

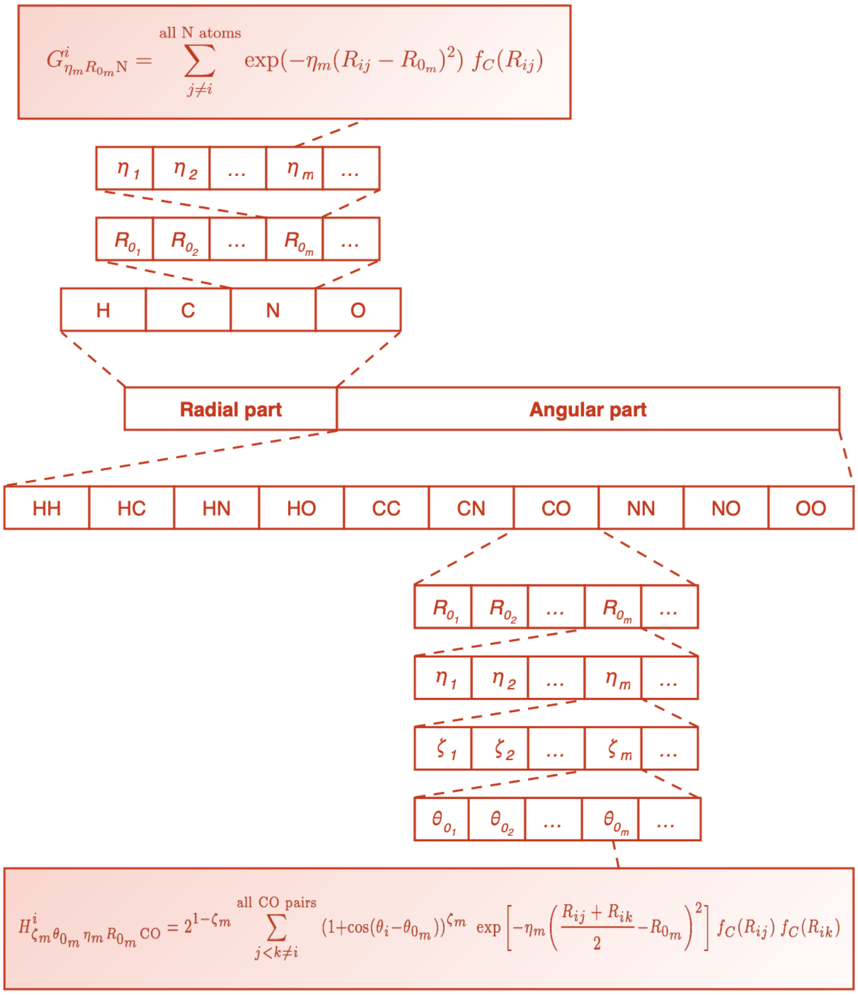

$G^i_{\eta_m R_{0_m} \mathrm{N}} = \sum_{j \neq i}^{\text{all N atoms}} \exp(-\eta_m (R_{ij} - R_{0_m})^2)\, f_C(R_{ij})$
η_1 η_2 ... η_m ...
R_{0_1} R_{0_2} ... R_{0_m} ...
H C N O
Radial part
Angular part
HH HC HN HO CC CN CO NN NO OO
R_{0_1} R_{0_2} ... R_{0_m} ...
η_1 η_2 ... η_m ...
ζ_1 ζ_2 ... ζ_m ...
θ_{0_1} θ_{0_2} ... θ_{0_m} ...
$H^i_{\zeta_m \theta_{0_m} \eta_m R_{0_m} \mathrm{CO}} = 2^{1-\zeta_m} \sum_{j<k\neq i}^{\text{all CO pairs}} (1+\cos(\theta_i - \theta_{0_m}))^{\zeta_m} \exp\left[-\eta_m \left(\frac{R_{ij}+R_{ik}}{2} - R_{0_m}\right)^2\right] f_C(R_{ij})\, f_C(R_{ik})$

Figure 4.6: Illustration of an ANAKIN atomic environment vector for a dataset containing only H, C, N and O.

Table 4.1: Architecture details of the ANAKIN Elemental MLPs.

	Layer 1	Layer 2	Layer 3	Layer 4
		Hydrogen		
Nodes	256	192	160	1
Activation	CELU	CELU	CELU	Linear
Regularization	L2 (5.0E-3)	L2 (1.0E-6)	L2 (1.0E-6)	None
		Carbon		
Nodes	224	192	160	1
Activation	CELU	CELU	CELU	Linear
Regularization	L2 (5.0E-3)	L2 (1.0E-6)	L2 (1.0E-6)	None
		Nitrogen/Oxygen		
Nodes	192	160	128	1
Activation	CELU	CELU	CELU	Linear
Regularization	L2 (5.0E-3)	L2 (1.0E-6)	L2 (1.0E-6)	None
		Sulfur/Fluorine/Chlorine		
Nodes	160	128	96	1
Activation	CELU	CELU	CELU	Linear
Regularization	L2 (5.0E-3)	L2 (1.0E-6)	L2 (1.0E-6)	None

central atom i. It uses a more complex ACSF:

$$\begin{aligned} H^i_{l(\zeta,\theta_0,\eta,R_0,\Phi,\Lambda)} = 2^{1-\zeta} \sum_{j<k\neq i}^{\text{all } \Phi\Lambda \text{ pairs}} & (1+\cos(\theta_i-\theta_0))^{\zeta} \times \\ & \times \exp\left[-\eta\left(\frac{R_{ij}+R_{ik}}{2}-R_0\right)^2\right] \times \\ & \times f_C(R_{ij}) f_C(R_{ik}) \end{aligned} \tag{4.10}$$

where Φ, Λ denote atom types. Analogously to the previous section, the final AEV is the concatenation of the radial and angular scalar values obtained with different combinations of the hyperparameters, as shown in Figure 4.6. This representation ensures rotational, translational, and axis-inversion invariance, as well as independence from atom ordering.

The Elemental MLPs block maps each AEV to its corresponding atomic energy. It consists of one multi-layer perceptron (MLP) for each chemical element in the dataset, with architectures detailed in Table 4.1. The atomic energy for each of the N atoms is computed by passing its AEV through the appropriate MLP based on the atom's type.

The final step involves computing atomic "self-energies", which can be interpreted as 1-body contributions of each atom interacting with itself. For this purpose ANAKIN employs a polynomial model fitted independently of the Elemental MLPs' training process. Despite its simplicity, this component is crucial, as approximately 99% of the energy in most molecular conformations comes from individual atomic self-energies [43]. Thus, incorporating this precomputed model facilitates a smoother training process for the neural network block.

The total molecular energy is obtained by summing the atomic energies from the MLPs and the polynomial self-energies. This scalar value preserves all the required invariances mentioned during our discussion.

4.3 Long-range and non-local interactions

Limiting the per-atom energy estimation to a cutoff sphere around it, as done often in neural network potentials, implicitly assumes that the energetic of the system is dominated by local interactions. However it is well known that in molecular systems long-range interactions such as electrostatics and dispersion forces are important factors. In fact, electrostatic interactions have been included and are crucial in classical atomistic potentials force fields, which typically use fixed partial charges for each atom types [31, 3, 93, 109]. Including long-range electrostatics in ML potentials too is relevant for various reasons, including the incorporation of physically meaningful energy terms via Coulomb's law thus improving the reliability of these models.

As a result, these efforts led the community to try to express partial charges and higher multipole moments using neural networks [35, 54], starting the so-called “third-generation” HDNNPs’ era, which use learnt, environment-dependent atomic charges to compute long-range electrostatic interactions explicitly. The first third-generation HDNNP was published in 2012, with applications to zinc oxide [5] and water dimers [94]. These potentials use a second set of atomic neural networks to express environment-dependent charges, which are then used to calculate long-range electrostatic energy without truncation.

The previous approach is insufficient when long-range charge transfer occurs. In such cases, an atom’s charge may depend on structural or electronic changes far beyond its local environment. Additionally, different ionization states can globally alter charge distribution, which methods based solely on local structural information cannot distinguish [37, 99, 76].

Charge equilibration methods [135], long used in advanced force fields like ReaxFF [131] and approximate electronic structure methods like DFTB [44], had offered a potential solution to the long-range charge transfer challenge. From them, an example of “fourth-generation” high-dimensional neural network potential dubbed 4G-HDNNP has been developed [76], combining the advantages of second-generation HDNNPs and the charge equilibration neural network technique (CENT) proposed by Goedecker and co-workers in 2015 [49, 45]. The total energy in 4G-HDNNPs is calculated as the sum of a short-range component and a long-range electrostatic one, using a method that differs from third-generation HDNNPs: the electrostatic energy is derived from charges obtained through a charge equilibration process, which relies on environment-dependent electronegativities expressed by atomic neural networks; these electronegativities are designed to reproduce reference Hirshfeld charges [57] from DFT calculations, rather than aiming to minimize electrostatic energy. As a result, these charges contain information about the global electronic structure, allowing both electrostatic and short-range energies to adapt consistently to charge density redistributions.

With a final note we underline that, as also remarked by other authors [12], despite these third- and fourth-generation HDNNPs models are the most advanced ones, often they are not the main choice for nowadays, practical routine. This is due primarily to the fact that contributions beyond the cutoff are often effectively screened in many systems, particularly condensed ones. The result is a potential energy error reduction of a few meV/atom that is not worth the increased computational cost for training and inference.

The above mentioned techniques are not the only ones usable to incorporate long range interactions. A further, indirect way to include these interactions is given by message passing techniques [50] implemented in systems such as AIMNet [139, 140, 4]. These methods involve iterative information transfer from one atom to another throughout the system, thus extending the effective range of representations and interactions along with the number of information passing steps included. From this point of view, when covering interactions with only a few nearby neighbor shells, they resemble second-generation high-dimensional neural network potentials (HDNNPs). At the other extreme, with an infinite number of information passing steps encompassing the entire system globally, they approach fourth-generation HDNNPs.

We will detail in the concluding parts of this thesis how OBIWAN [90], our newly proposed method, could benefit from message passing modules in cases where the interactions' landscape is too complex for a short-range only description, paving the way for further accuracy enhancements.

Chapter 5

Introducing OBIWAN

ANAKIN proved to be a largely successful and efficient network, yet it presents some possible limitations. Its primary constraint lies in the length of its Atomic Environment Vectors (AEVs), which serve as input to the neural component of the model. The AEV length grows quadratically with the number of species in the dataset due to the computation of angular components. This leads to significant computational demands even with fewer than a dozen elements, considering that the parameter count in the initial layers of the elemental MLPs is tied to the number of input features. Furthermore, incorporating new chemcal elements necessitates adding a new elemental MLP and retraining all the others from scratch, highlighting ANAKIN's inherent inability to leverage previously acquired chemical and physical knowledge.

It's worth noting that these limitations also constitute ANAKIN's strengths and the source of its expressiveness: the resulting representation of input molecules allows for precise learning of various potential energy surfaces. OBIWAN [90], our newly proposed architecture, was conceived as an optimal compromise between the aforementioned effectiveness, portability, and generality.

In the existing literature, the approach most comparable to OBIWAN is the one published by Gastegger *et al.* [47], which we refer to as the W-ANAKIN model. Aiming to eliminate the quadratic scaling of the original ANAKIN's input length, the authors reformulated the AEV computation procedure using a "weighted" ACSF expression. The equation for the radial part becomes

$$G^{i}_{l(\eta,R_0)} = \sum_{j\neq i}^{\text{all atoms}} z_j \exp(-\eta(R_{ij}-R_0)^2) f_C(R_{ij}) \tag{5.1}$$

while the angular part is given by

$$\begin{aligned} H^{i}_{l(\zeta,\lambda,\eta,R_0)} = 2^{1-\zeta} \sum_{j<k\neq i}^{\text{all pairs}} & z_j z_k \; (1+\lambda\cos(\theta_i))^{\zeta} \; \exp(-\eta(R_{ij}-R_0)^2) \times \\ & \times \exp(-\eta(R_{ik}-R_0)^2) \; \exp(-\eta(R_{jk}-R_0)^2) \times \\ & \times f_C(R_{ij}) \; f_C(R_{ik}) \; f_C(R_{jk}) \end{aligned} \tag{5.2}$$

where $z_j$and z_k represent atomic numbers. The key distinction from ANAKIN is that these formulas are no longer element-specific, with summation now performed over all atoms (or atom pairs) in the vicinity. Essentially, all ANAKIN AEV components associated with different species are condensed into a single scalar, attempting to preserve chemical information by weighting each addend j with the atomic number z_j for the radial part and each addend jk with the product of atomic numbers $z_j z_k$ for the angular part. Environmental sensing remains controlled solely by the symmetry functions' hyper-parameters, effectively creating an AEV whose length is independent of the number of species in the dataset. Each AEV undergoes a z-score transformation before being input into the Elemental MLPs, ensuring all descriptors have a mean of 0 and a standard deviation of 1.

As noted by Rostami and colleagues [111], the original W-ANAKIN paper [47] provides limited evidence of the approach's accuracy, given that the authors train and test the model on only a few tens of thousands of structures in their equilibrium states. Moreover, as will be demonstrated in Section 5.2, the fact that Equation 5.2 is nearly a direct application of the original Behler and Parrinello angular symmetry function [16] rather than the ANAKIN version leads to a decline in the algorithm's generalization capability. Thus, for fairness, our tests also considered a modified W-ANAKIN model, termed W-ANAKIN-OE, which reinstates the ANAKIN-optimized angular function. The W-ANAKIN-OE angular AEV expression is therefore:

$$\begin{aligned} H^{i,OE}_{l(\zeta,\theta_0,\eta,R_0)} = 2^{1-\zeta} \sum_{j<k\neq i}^{\text{all pairs}} & z_j z_k \, (1+\cos(\theta_i-\theta_0))^{\zeta} \times \\ & \times \, \exp\left[-\eta\left(\frac{R_{ij}+R_{ik}}{2}-R_0\right)^2\right] \times \\ & \times \, f_C(R_{ij}) \, f_C(R_{ik}) \end{aligned} \tag{5.3}$$

The radial symmetry function employed by W-ANAKIN-OE remains identical to that of W-ANAKIN.

5.1 Architecture

To address the limitations of the ANAKIN framework mentioned earlier, we replaced its deterministic symmetry functions with fully learnable components. We collectively named the resulting, enhanced AEVs as Deep Atomic Environment Vectors (DeepAEVs). This modification allowed us to train a single MLP for atomic energy inference, ultimately creating an architecture whose structure is independent of the chemical species present in the dataset. We named this approach OBIWAN [90].

Figure 5.1 illustrates the computational graph of our proposed model. In

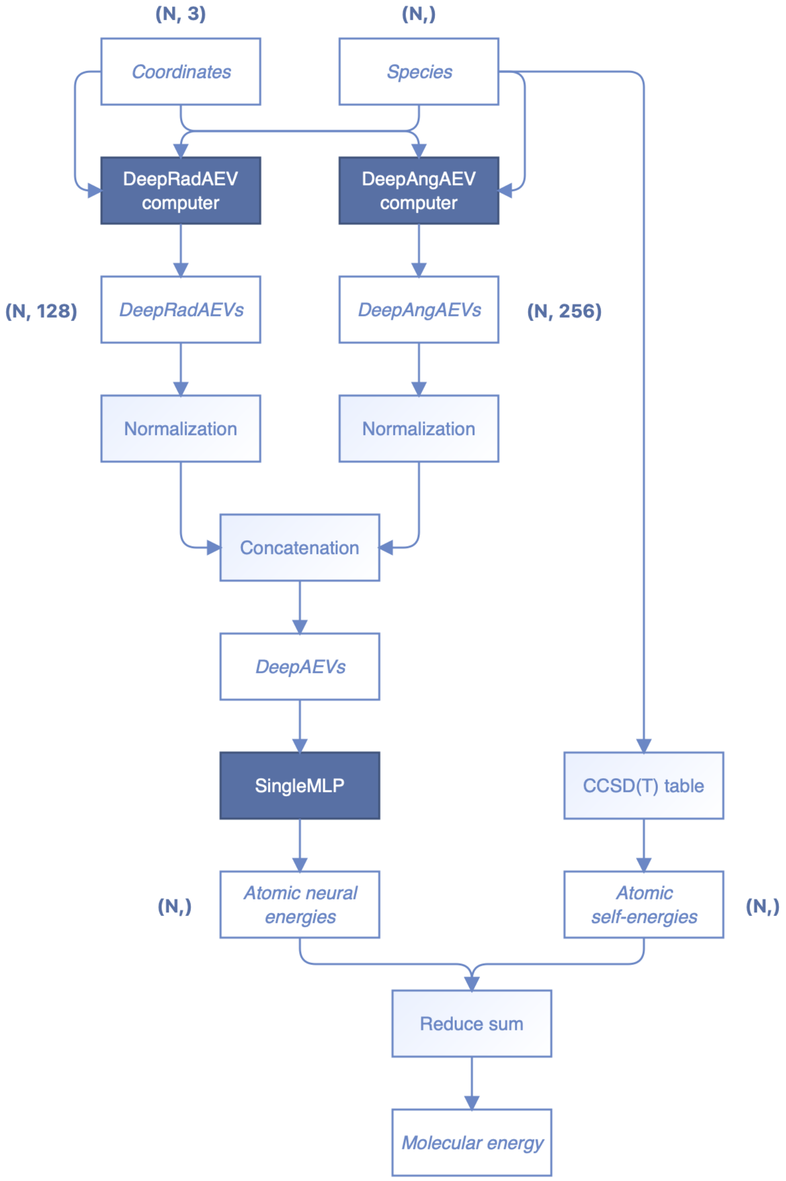


Figure 5.1: The OBIWAN computational graph.

the following, we will elaborate on the DeepAEVs, the single MLP for energies, and the self-energies strategy we adopted.

Deep Atomic Environment Vectors

For representing the input molecule, we maintain ANAKIN's efficient atomic neighbourhood scanning procedure and combine it with W-ANAKIN's global sum. For each atom i, its set of neighbours (or pairs of neighbours) is mapped to a new set of *vectors*; these vectors are then summed to obtain the AEV, which ultimately represents the featurized information of atom i and its neighbourhood. This process requires two key decisions: (i) selecting the geometric/physical observables used to construct the AEVs (*e.g.*, atomic distances and angles for ANAKIN); (ii) defining the mapping function applied to these observables to obtain the values for the sum-reduction (*e.g.*, the radial and angular ACSFs for ANAKIN).

For the radial component of our DeepAEVs, we use distances and atomic numbers as observables. We identified each neighbour atom j by its distance R_{ij} from the central atom i along with the atomic number z_j of the former. We also included the atomic number z_i of the central atom in each neighbour representation, providing every DeepAEV with knowledge of its center (unlike ANAKIN's center-agnostic AEVs). Thus, each neighbour j is represented by the set $\{R_{ij}, z_i, z_j\}$.

To transform these sets into an expressive feature space, we employ a dedicated MLP, denoted as $\overrightarrow{\mathcal{M}}_{\text{rad}}$. Since we're dealing with sets, we need to convert them into vectors before inputting them into $\overrightarrow{\mathcal{M}}_{\text{rad}}$. However, we wanted this conversion to be independent of any prescribed ordering to completely maintain the atoms' indices invariance required by neural network potentials: although one could order these three numbers by always placing the geometrical feature R_{ij} before the atomic numbers, deciding an order for z_i and z_j remains an issue, because one could always place z_i before z_j, but we aimed to represent the sub-system ij in a way that is independent of the choice of the central atom, resulting in a representation that is consistent across all AEVs to introduce an invariance that can aid $\overrightarrow{\mathcal{M}}_{\text{rad}}$ in its fitting process. Thus, instead of imposing an arbitrary order, we performed an explicit set-to-vector conversion. We achieved this by introducing a new layer, leveraging the bijective relationship between the zeros $x_1,\ldots,x_n$ (atomic numbers in our case) of an n-grade monic polynomial $P(x)$ and the ordered set of its coefficients $a_{n-1},\ldots,a_0$ (our desired vector).

Given the polynomial expression:

$$P(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 \tag{5.4}$$

all the $a_{n-1},\ldots,a_0$ can be analytically computed using the Viète-Girard formulas, of which some examples are:

$$\begin{gathered} a_{n-1} = -(x_1 + x_2 + \cdots + x_n) \\ a_{n-2} = (x_1x_2 + x_1x_3 + \cdots + x_1x_n) + (x_2x_3 + x_2x_4 + \cdots + x_2x_n) + \cdots + x_{n-1}x_n \\ \cdots \\ a_0 = (-1)^n(x_1x_2\cdots x_n) \end{gathered} \tag{5.5}$$

or more generally

$$a_{n-k} = (-1)^k \sum_{1\le i_1<i_2<\cdots<i_k\le n} x_{i_1}x_{i_2}\cdots x_{i_k} \quad . \tag{5.6}$$

For the radial part $\overrightarrow{G}^i_{\text{rad}}$ of our DeepAEV, we need these formulas up to the second order because we are considering a polynomial with only two zeros: z_i and z_j. Thus, denoting $\overrightarrow{\mathcal{V}}$ as the function that performs the Viète-Girard set-to-vector mapping and using $\big|$ to represent the concatenation operation, we have:

$$\overrightarrow{G}^i_{\text{rad}} = \sum_{j\neq i}^{\text{all atoms}} \overrightarrow{\mathcal{M}}_{\text{rad}}\Big(R_{ij} \,\Big|\, \overrightarrow{\mathcal{V}}(\{z_i, z_j\}) \Big) f_C(R_{ij}) \tag{5.7}$$

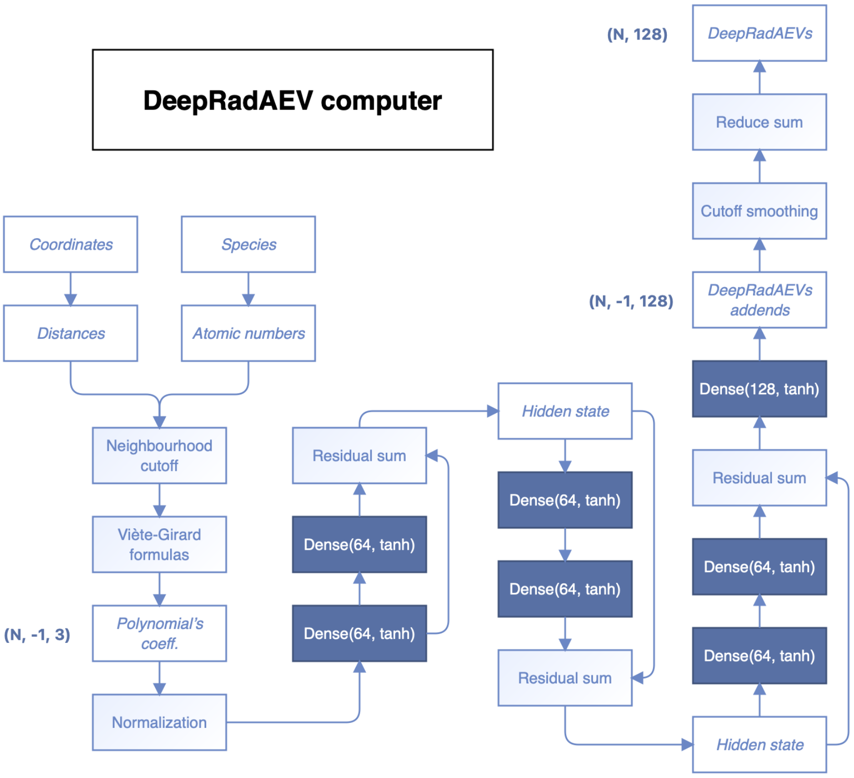


Figure 5.2: The radial part of our DeepAEV computing procedure.

with

$$\vec{\mathcal{V}}(\{z_i, z_j\}) \;=\; (z_i + z_j) \,\Big|\, (z_i * z_j) \tag{5.8}$$

Figure 5.2 illustrates the radial part of our DeepAEV computation procedure, including the architectural details of $\vec{\mathcal{M}}_{\text{rad}}$ with its main hyperparameters and residual connections [56]. For the cutoff smoothing operation, we retain the monotonic cosine function employed in ANAKIN.

The angular part $\vec{H}^i_{\text{ang}}$ of our DeepAEVs follows a similar strategy. To represent the unordered pairs of atoms jk in the neighbourhood of i, including information relative to the center, we consider the three triangle sides R_{ij}, R_{ik} and R_{jk} as geometrical features and z_i, z_j and z_k as chemical ones.

However, while $\vec{\mathcal{V}}$ can be directly used for the radial part, care must be taken here to preserve the association of each atom to its triangle vertex. To address this, we pair each atomic number z_i with its corresponding angle θ_i, interpreting the pair (z_i, θ_i) as the real and imaginary parts of a complex number, which can still serve as a root of a polynomial. Viète-Girard formulas apply to any polynomial with coefficients in a commutative ring, so this generalization poses no problem for our procedure. Naturally, a set of n complex numbers will be mapped to a complex-valued vector by our $\vec{\mathcal{V}}$ function, resulting in $2n$ real features for our mapping function.

Employing a new MLP $\overrightarrow{\mathcal{M}}_{\text{ang}}$ for the high-dimensional projection of the angular part $\vec{H}^i_{\text{ang}}$ of our DeepAEV, we have:

$$\vec{H}^i_{\text{ang}} = \sum_{j<k\neq i}^{\text{all pairs}} \overrightarrow{\mathcal{M}}_{\text{ang}}\Big(\vec{\mathcal{V}}(\{R_{ij}, R_{ik}, R_{jk}\}) \,\Big|\, \vec{\mathcal{V}}(\{z_i + I\theta_i, z_j + I\theta_j, z_k + I\theta_k\}) \Big) \times \\ \times\ f_C(R_{ij})\, f_C(R_{ik}) \tag{5.9}$$

where I is the imaginary unit. The $\vec{\mathcal{V}}$ evaluations are more complex here, so we've chosen not to report them; they can be found in the carefully commented GitHub repo cited in Chapter 6.

Figure 5.3 illustrates the angular part of our DeepAEV computation procedure, including the architectural details of $\overrightarrow{\mathcal{M}}_{\text{ang}}$ with its main hyperparameters and residual connections [56]. For the cutoff smoothing operation, we retain the monotonic cosine function employed in ANAKIN. Additionally, we also employed just a “double cutoff"” as in ANAKIN because we found it effectively leads to better performance.

Finally, each radial and angular part of the DeepAEV is independently divided by its norm before the usual concatenation that forms the actual environment vector. This final operation is crucial as it confines the vector coefficients (originally too large due to the summation over all neighbours)

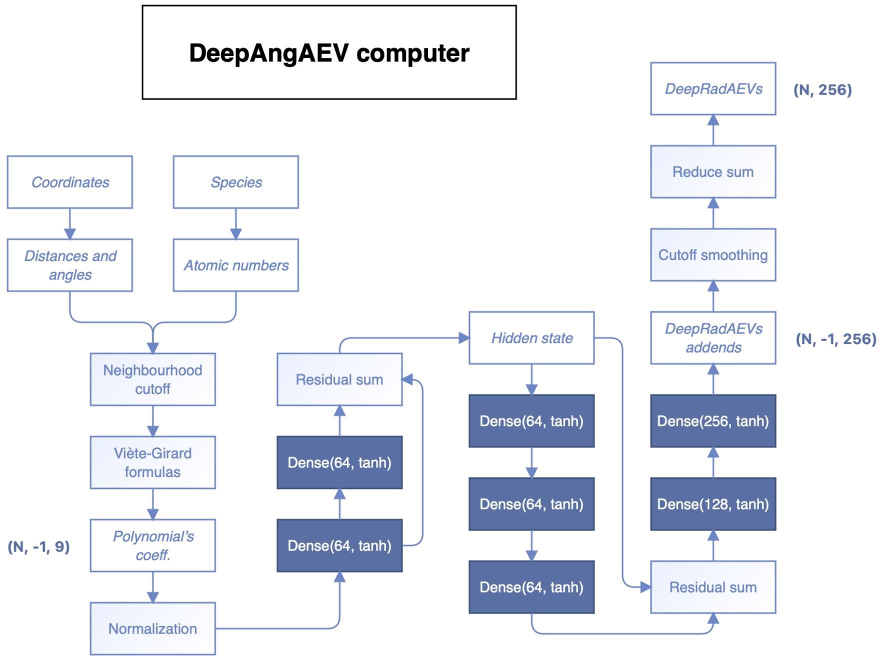


Figure 5.3: The angular part of our DeepAEV computing procedure.

to a numeric interval suitable for the subsequent MLP forward pass. It's worth noting that our implementation of the $\overrightarrow{\mathcal{V}}$ preprocessing function performs a similar rescaling operation, normalizing the resulting vector in a Group Normalization fashion [136] before passing it as input to $\overrightarrow{\mathcal{M}}_{\text{rad}}$ and $\overrightarrow{\mathcal{M}}_{\text{ang}}$.

All DeepAEV computations are a function of interatomic distances and angles only, thus maintaining all the geometrical invariance properties of ANAKIN necessary for a neural network potential. Moreover, thanks to our newly introduced Viète-Girard layer, the model remains invariant to the permutation of atom indices too.

Single MLP

At this stage of the pipeline, we have one AEV per atom, similar to ANAKIN. However, the key difference is that each vector now contains information about the chemical identity of the central atom i as well. This allows us to deploy just a single MLP for atomic energy inference, reducing ANAKIN's elemental forward passes to only one shared module.

As we will demonstrate in Section 5.2, thanks also to this single MLP OBIWAN results in an architecture that is completely independent of the chemical species present in the dataset. This allows the end user to leverage past chemical knowledge in new training routines involving unseen elements without the need to add blocks and restart training from scratch each time.

Coupled-cluster self-energies

As mentioned earlier, accurate energy estimation requires consideration of one-body contributions (“self-energies”) as well. This component is crucial in both training and inference as it constitutes the major part of the final molecular energy. While ANAKIN obtained these components from a data-dependent, learned polynomial, we opted for a more physically accurate approach. Following the procedure employed by Plé and colleagues [103], we evaluate the one-body contributions using CCSD(T) isolated atom energies. Specifically, we used the values tabulated in the work of Ranasinghe *et al.* [108]

Consequently, in our approach, adding a new element to the network simply involves retrieving (or computing) the corresponding CCSD(T) value once and for all. This differs significantly from the ANAKIN approach where adding a new element would not only require retraining the neural model from scratch, as previously mentioned, but would also necessitate retraining (and possibly changing the grade of) the polynomial for the self-energies.

5.2 Results

This section presents the OBIWAN [90] model’s performance across four distinct scenarios:

1. A comparative analysis of OBIWAN against ANAKIN and W-ANAKIN using the 5 million configurations dataset `ani2x` [38].
2. An evaluation of OBIWAN on the comprehensive test set `COMP6v2` [38, 122], following an extensive force-training process on the 10 million

configurations dataset resulting from the combination of ani1x [122, 123] and ani2x.

3. A validation of OBIWAN's capabilities in dihedral scanning and molecular dynamics.

4. An assessment of OBIWAN's ability to leverage previously acquired knowledge for efficient training on new elements, utilizing the recently released drugs and peptides dataset SPICE [43].

We begin by describing the aforementioned datasets, followed by the specifics of the employed architectures and training configurations. The final subsection presents our findings.

5.2.1 Datasets

The primary challenge in developing truly general and transferable neural network potentials is the scarcity of high-quality training data [43]. This limitation can lead to models that perform well within their specific chemical domain but struggle with broader, real-world scenarios involving numerous molecules and conformations. To fully realize the potential of machine learning in producing accurate and transferable forces and energies, training and validation datasets should adhere to the following key criteria [43]:

- Extensive chemical space coverage, encompassing a wide range of elements and bond types, resulting in diverse compounds;

- Broad conformational space representation, including off-equilibrium structures necessary for molecular dynamics simulations;

- Inclusion of both forces and energies as labels, providing atomic-level information [100] for models with atom-wise outputs;

- Utilization of the highest computationally feasible level of theory.

The datasets used to train the models in this thesis exemplify the aforesaid best practices and their main characteristics are listed below.

ani1x: An actively-learned dataset developed for early versions of ANAKIN [122, 123]. It contains:

- Molecules with H, C, N, and O atoms;
- 63,865 distinct configurations in 4,956,005 total 3D conformations;
- Labels for energy and forces;
- Computations using the wb97x DFT functional and 6-31g* basis set.

ani2x: An extension of ani1x using similar active learning techniques [38]. It features:

- Molecules with H, C, N, O, F, S, and Cl atoms;
- 13,405 distinct configurations in 4,695,707 total 3D conformations;
- Labels for energy and forces;
- Computations using the wb97x DFT functional and 6-31g* basis set.

COMP6v2: A comprehensive benchmark suite for assessing the transferability of general machine learning potentials [122, 38]. It comprises:

- Molecules with H, C, N, O, F, S, and Cl atoms;
- 8,247 distinct configurations in 157,728 total conformations;

- Labels for energy and forces;
- Computations using the wb97x DFT functional and 6-31g* basis set.

SPICE: A quantum chemistry dataset for simulating drug-like small molecules interacting with proteins [43]. The subset used in this study, "PubChem", includes:

- Structures with H, C, N, O, F, P, S, Cl, Br, and I atoms;
- 12,962 distinct configurations in 627,692 total conformations;
- Labels for energy and forces;
- Computations using the wB97M-D3(BJ) DFT functional and def2-TZVPPD basis set.

It's important to note that the ani1x and ani2x datasets, extensively used in our experiments, were developed through active learning cycles specifically optimized for the ANAKIN architecture [122, 123]. Consequently, OBIWAN's accuracy levels should be interpreted as both a validation of the effectiveness of this training technique and a lower bound of what OBIWAN could achieve with an active learning strategy tailored to it.

5.2.2 Models details

Our experiments incorporates three neural network potential architectures:

1. ANAKIN [121, 122, 38]: The starting model from which ours evolved;
2. W-ANAKIN [47]: To our knowledge, the most successful attempt in our same direction (including two enhanced versions developed by us);

3. OBIWAN: Our candidate model.

The architectural specifics for each are as follows:

ANAKIN: Re-implemented in TensorFlow [129], our ANAKIN model features:

- 112 radial and 896 angular AEV components, yielding a 1008-length concatenated AEV;
- Cutoff values: 5.1 Angstrom (radial) and 3.5 Angstrom (angular);
- Elemental MLPs: One feed-forward neural network per species, with layer and neuron counts as per Table 4.1;
- 1,713,223 trainable parameters.

AEV coefficients (η, ζ, etc.) and relevant self-energies are available in our GitHub repository [91].

W-ANAKIN: We developed three variants:

1. Original W-ANAKIN:
    - 26 radial and 6 angular AEV components;
    - Cutoff values aligned with ANAKIN for fair comparison;
    - Elemental MLP architectures matching ANAKIN (Table 4.1);
    - 401,479 trainable parameters.
2. W-ANAKIN-OE:
    - Incorporates symmetry functions from Roitberg et al. [121];
    - 422,983 trainable parameters.

3. W-ANAKIN-OE-FULL:
   - Input AEVs' length matching the OBIWAN DeepAEVs' length;
   - Enlarged Elemental MLPs;
   - 1,715,825 trainable parameters (for a fair comparison).

Detailed specifications for all variants are available in our GitHub repository [91].

OBIWAN: Our candidate model:

- DeepAEV computation as per Figures 5.2 and 5.3;
- 128 radial and 256 angular components, yielding 384-length DeepAEVs;
- $\overrightarrow{\mathcal{M}}_{\text{rad}}$ and $\overrightarrow{\mathcal{M}}_{\text{ang}}$: 29,376 and 58,624 neurons respectively (88,000 total learnable featurizer parameters);
- ANAKIN cutoff values for fair comparison;
- Single MLP: Dense layers [1024, 768, 512, 256, 1] with CELU hidden activations and linear output (1,706,753 parameters);
- CCSD(T) self-energies from Ranasinghe et al. [108];
- 1,794,753 total trainable parameters.

5.2.3 Training protocol

Our experiments utilized a consistent objective function across all models, mirroring the original ANAKIN approach. This function combines "normalized" energy and forces losses. For a given molecular configuration x_k, with atom i and coordinate j, let $\hat{E}(x_k)$ and $\hat{f}_{ij}(x_k) = -\nabla_{x_{k,i,j}}\hat{E}(x_k)$ represent predicted energies and forces, while $E(x_k)$ and $f_{ij}(x_k)$ denote QM reference values. The loss $\mathcal{L}$ for each sample is defined as:

$$\mathcal{L}(x_k) \equiv \frac{(\hat{E}(x_k) - E(x_k))^2}{\sqrt{N_k}} + w \frac{\sum_{i=1}^{N_k}\sum_{j=1}^{3}(\hat{f}_{ij}(x_k) - f_{ij}(x_k))^2}{N_k} \tag{5.10}$$

where N_k represents the atom count in the molecule and $w > 0$ balances energy and forces loss terms during training. The global loss is the mean of Equation 5.10 across all batch samples, with batch size dynamically scaling based on GPU count. Forces are estimated analytically via TensorFlow's automatic differentiation engine.

For weight updates, we employed the Adam optimizer [74] with default TensorFlow parameters, except for `epsilon` that we set equal to 1e-8. The learning rate's initial magnitude depended on the GPU count, halving every 20 consecutive epochs without training loss decrease. This schedule ceased at a learning rate of 1e-5, though training continued until reaching the desired plateau.

Additional training details include:

- OBIWAN Single MLP and ANAKIN/W-ANAKIN Elemental MLPs: Hidden layer kernels constrained to a maximum norm of 3.0;

- Hardware: Up to 4 V100 or A100 Nvidia GPUs, depending on cluster node availability;

- Training duration: Approximately 5-6 weeks on 4 A100s for OBIWAN on the 9.6M structure training set;

- Dataset split: 80% training, 20% validation.

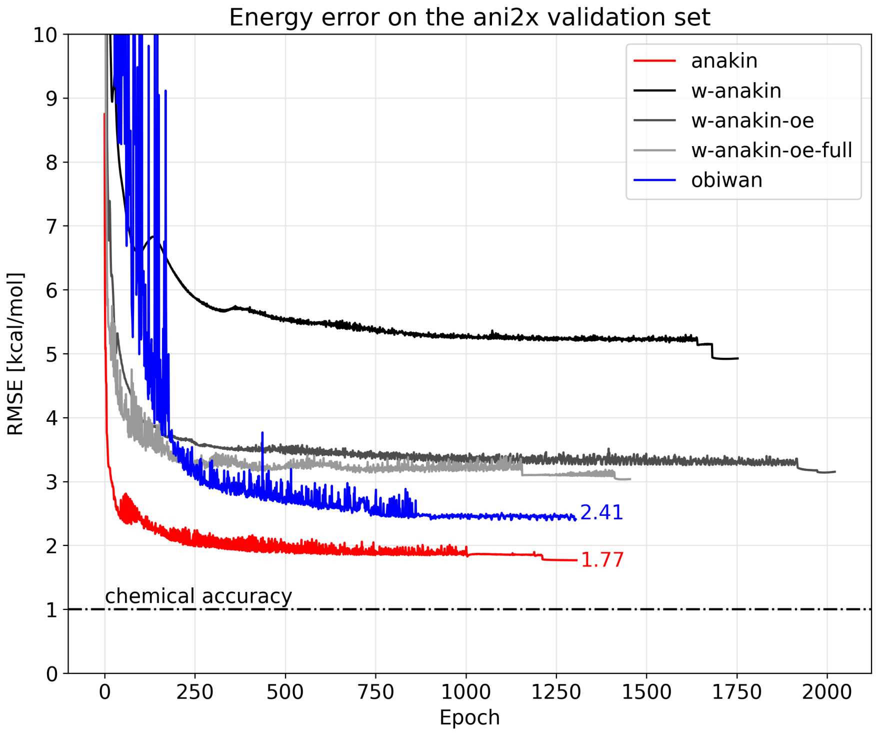


Figure 5.4: RMSE progression on the ani2x validation set for OBIWAN and competing models.

5.2.4 Tests

We conducted an initial experiment to evaluate OBIWAN's performance against its theoretical upper bound ANAKIN and its rival W-ANAKIN. As already said, W-ANAKIN represents the current best attempt that we found in literature to achieve species-independence remaining in the ANAKIN framework.

All the aforementioned models were trained on the $4,695,707$ energy values of the ani2x dataset, with the weight w of Equation 5.10 set equal to 0. Our comparison included not only the original W-ANAKIN [47], but also two enhanced versions we developed in order to match ANAKIN and

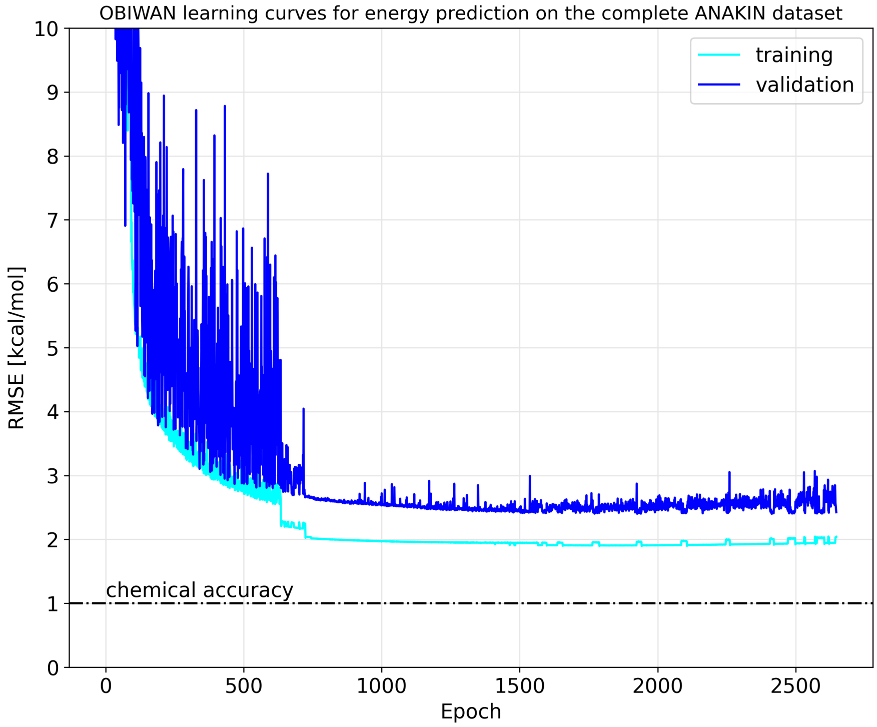
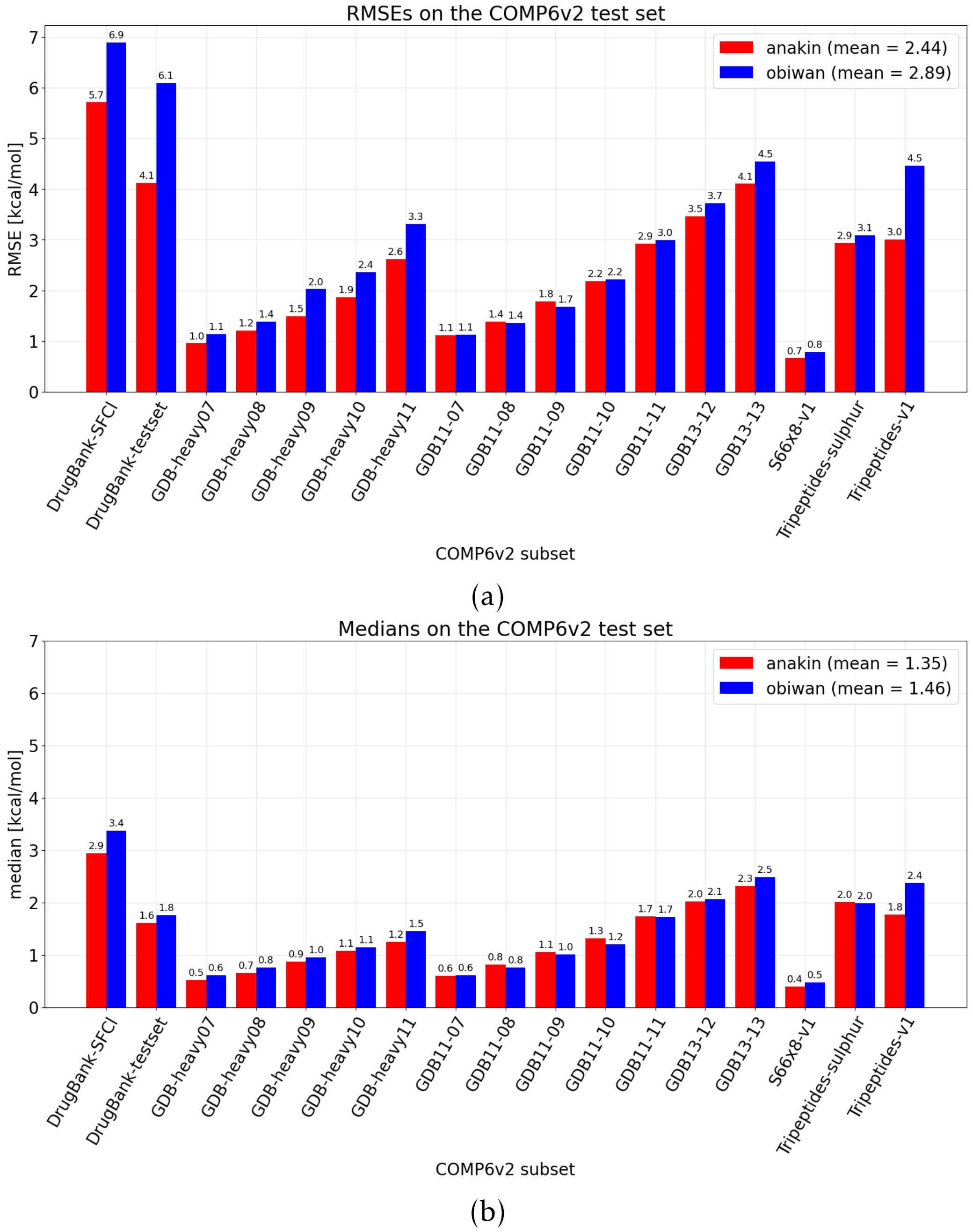


Figure 5.5: OBIWAN's training and validation errors on the combined ani1x and ani2x dataset.

OBIWAN's representational capabilities. Figure 5.4 displays the resulting RMSE curves for the validation set.

As expected, OBIWAN surpassed the original W-ANAKIN model, which was relatively small for this data-rich scenario. Notably, even equipping W-ANAKIN with OBIWAN's parameter count and input AEV length did not bridge the performance gap. Thus, OBIWAN's result establishes a new state-of-the-art benchmark for species-independent feed-forward neural networks in this task domain. It's also particularly noteworthy that OBIWAN's learned input representation (DeepAEV) is 2.6 times more compact than ANAKIN's original input, yet OBIWAN achieves comparable accuracy.



(a)

(b)

Figure 5.6: (a) ANAKIN and OBIWAN mean errors on the COMP6v2 benchmark suite; (b) ANAKIN and OBIWAN median errors on the COMP6v2 benchmark suite.

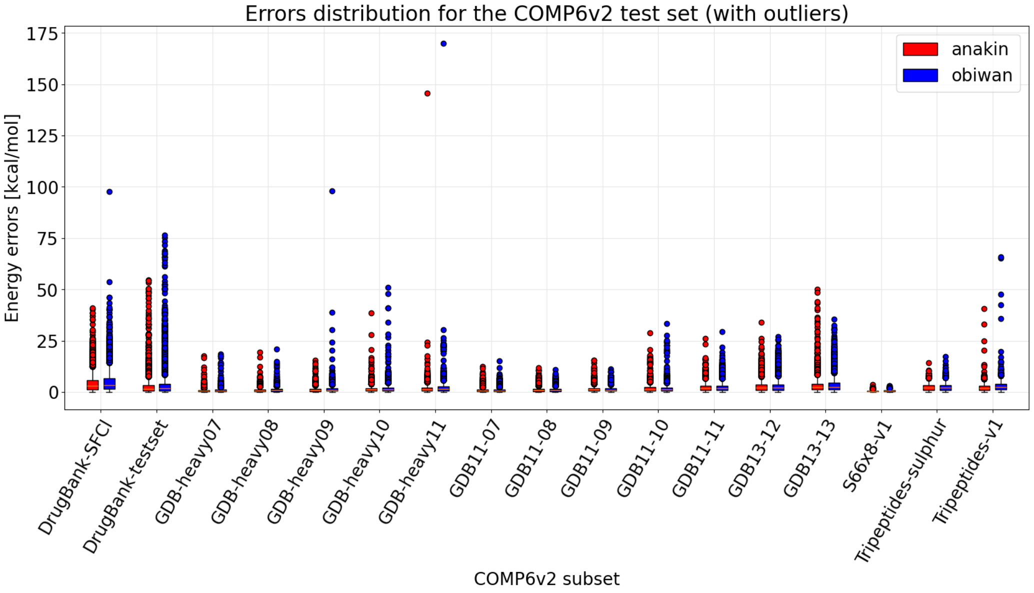


Figure 5.7: ANAKIN and OBIWAN error distributions on the COMP6v2 benchmark suite (including outliers).

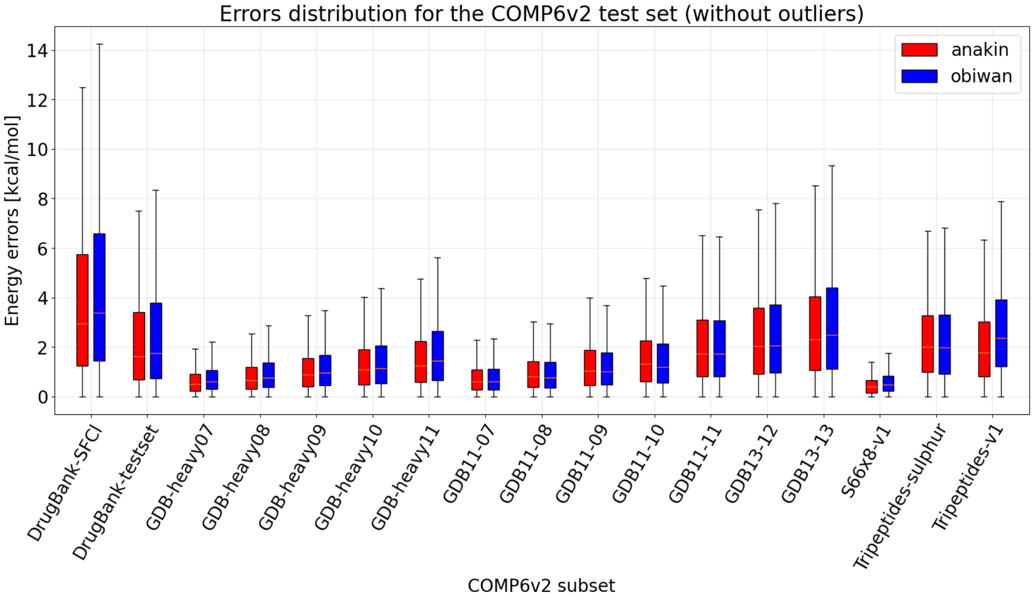


Figure 5.8: ANAKIN and OBIWAN error distributions on the COMP6v2 benchmark suite (excluding outliers).

To better compare the out-of-distribution performance of ANAKIN and OBIWAN we conducted a second experiment. We trained OBIWAN on the energies and forces of the $9,651,712$ structures resulting from the combination of the ani1x and ani2x datasets, with the weight w in Equation 5.10 set to 0.1 matching ANAKIN's original works [38, 48]. We then evaluated this OBIWAN model and the original ANAKIN one (trained on the same data) over the datasets from the COMP6v2 benchmark suite. Figure 5.5 illustrates OBIWAN's learning curves on the 9.6M sample dataset, while Figures 5.6 to 5.8 depict error statistics on the COMP6v2 suite. We excluded the ANI-MD subset from these experiments, as it comprises frames from molecular dynamics using ANAKIN as forces' predictor [122], potentially biasing the results and not accurately reflecting a general NNP extrapolation performance.

Figure 5.6 (a) shows that OBIWAN matches ANAKIN's performance in most energy prediction tasks where ANAKIN is deployable (*i.e.* when RMSE < 3kcal/mol). This similarity is particularly evident for the "not heavy" subsets containing only H, C, N, and O atoms. We hypothesize that this occurs because only the ani2x dataset includes molecules with F, S, and Cl atoms. Consequently, the final training set, combination of ani2x and ani1x, contains more examples of molecules with only H, C, N, and O elements, allowing OBIWAN's end-to-end nature to achieve greater accuracy for these elements. This trend is also visible when comparing DrugBank-SFCl to DrugBank-testset; the latter, lacking S, F, and Cl elements, shows slightly better performance.

For a more comprehensive analysis of these error statistics and to further validate our method, we also plotted the error distributions of OBIWAN and ANAKIN for each COMP6v2 test subset. Figure 5.7 reveals that both models exhibit critical outliers (*i.e.* points beyond 1.5 times the interquartile range from Q1 or Q3) in nearly identical positions. These few high-error samples compress the box plots to the chart's bottom edge, obscuring finer details. Therefore, Figure 5.8 presents the same box plots without

Table 5.1: Statistics of the relaxed 2D torsion profiles shown in Figure 5.9. RMSEs are in kcal/mol, measured relative to DFT.

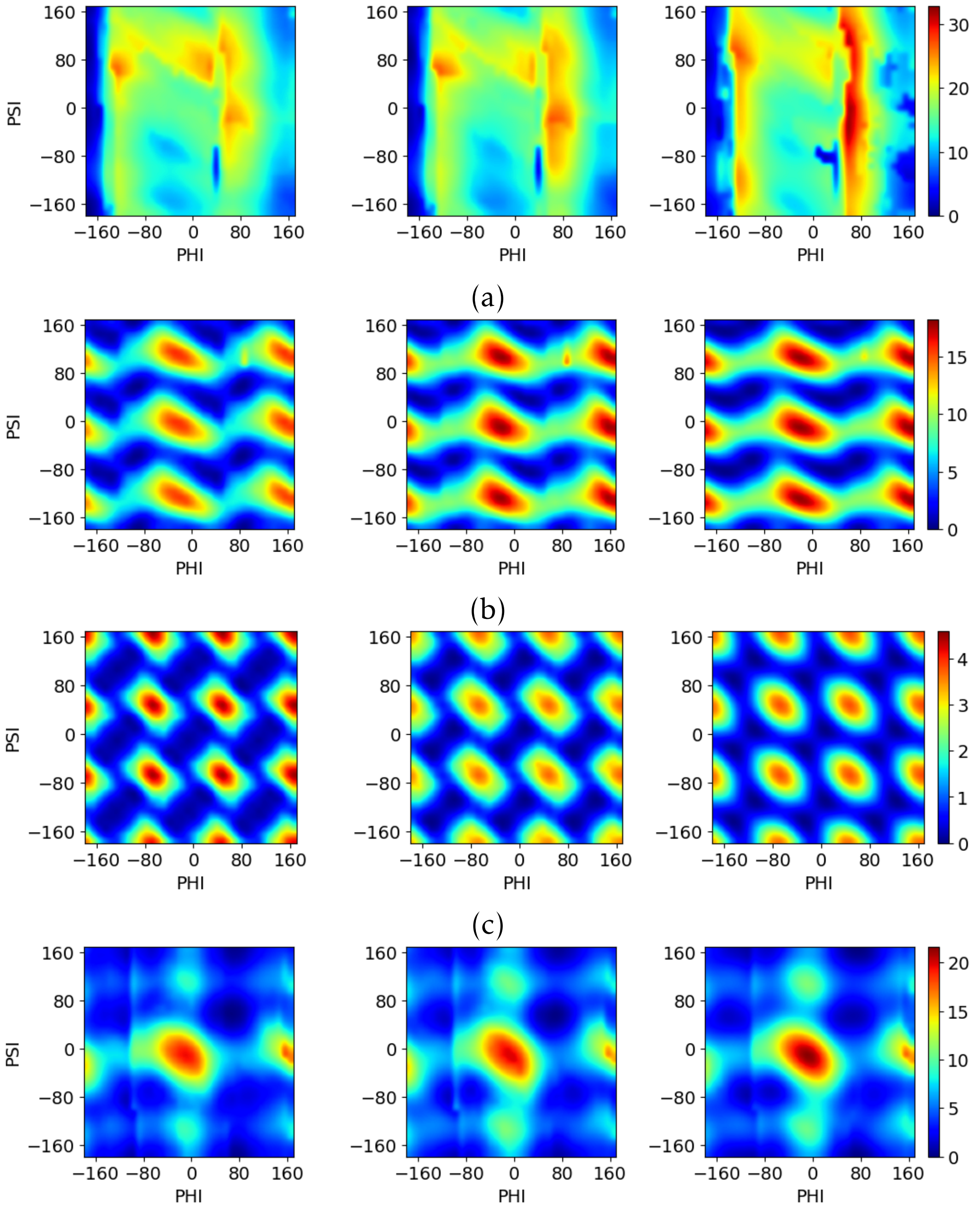
Molecule	num. atoms	RMSE OBI	RMSE ANI
Cysteine-Dipeptide	25	3.41	3.15
DDT	28	1.52	0.73
Hexafluoroacetone	10	0.29	0.22
Bendamustine	44	0.95	0.70

outliers, demonstrating similar error distributions across all sets for both models. Finally, Figure 5.6(b) displays the error *medians* of OBIWAN and ANAKIN on the COMP6v2 test subsets. This visualization shows that both models perform below the 3.0 kcal/mol deployability threshold when outliers are appropriately weighted, with most errors approaching chemical accuracy. This aligns with OBIWAN's satisfactory application results reported in subsequent paragraphs. It also suggests that both OBIWAN and ANAKIN might benefit from additional active learning rounds, or that some structures may have incorrect DFT reference energies.

Recognizing that low error metrics alone are insufficient to demonstrate a neural network potential's utility, we further validated OBIWAN empirically by employing it as a force field in two common computational chemistry tasks: 2D dihedral torsion scanning and molecular dynamics (MD).

We start deploying OBIWAN to estimate the potential energy surfaces of the same four small molecules (and dihedrals) examined in the ANI-2x study [38].

For each system in Table 5.1 and Figure 5.9, we computed the relaxed



(a)

(b)

(c)

(d)

Figure 5.9: Relaxed 2D torsion profiles for the four small molecules and dihedrals examined in the ANI-2x study [38]: (a) Cysteine-Dipeptide; (b) DDT; (c) Hexafluoroacetone; (d) Bendamustine. Columns represent OBIWAN, ANAKIN, and DFT results.

2D torsion profile using OBIWAN, ANI-2x, and DFT[3] to determine the relative potential energies for each method. We generated each surface by rotating two dihedrals in ten-degree increments and optimizing the resulting geometries with DFT while keeping those dihedrals fixed. Unlike the ANI-2x paper [38], we did not re-optimize the DFT geometry using either ANAKIN or OBIWAN for comparison purposes, an approach that naturally leads to higher ANAKIN errors compared to those reported in the original ANI-2x manuscript [38].

As shown in Table 5.1, OBIWAN achieves chemical accuracy for the same two bottom systems as ANAKIN, with a higher error only on the first molecule, similar to ANAKIN. Our error for DDT is twice that of ANAKIN, but still well below the 3.0 kcal/mol deployability threshold.

We further explored OBIWAN's applicability in MD simulations by generating a system trajectory for the GSK1107112A compound, the same molecule discussed in the ANAKIN-2 paper [38]. For this purpose, we developed a minimal TensorFlow MD engine from scratch (released with the OBIWAN code [91]) and conducted multiple simulations varying the timestep and the starting tautomeric configuration. For equation integration, we employed a leap-frog integrator with impulsive Langevin for temperature control [52]. Following standard procedures, we optimized the molecule's geometry, then equilibrated the system by gradually raising the temperature to 300 K and finally run the simulation for 1 ns. For the experiment shown in Figure 5.10, we used a 1 fs timestep, larger than the 0.4 fs employed by Devereux and colleagues in their corresponding work [38]. The energy values and mean force magnitudes for the last 25 ps are displayed. The simulation remained stable throughout its duration, even with the 1 fs time step. Our findings align with the ANAKIN-2 and DFT results reported in the cited work [38] in terms of energy range and fluctuation.

[3] data from ANI-2x paper [38]

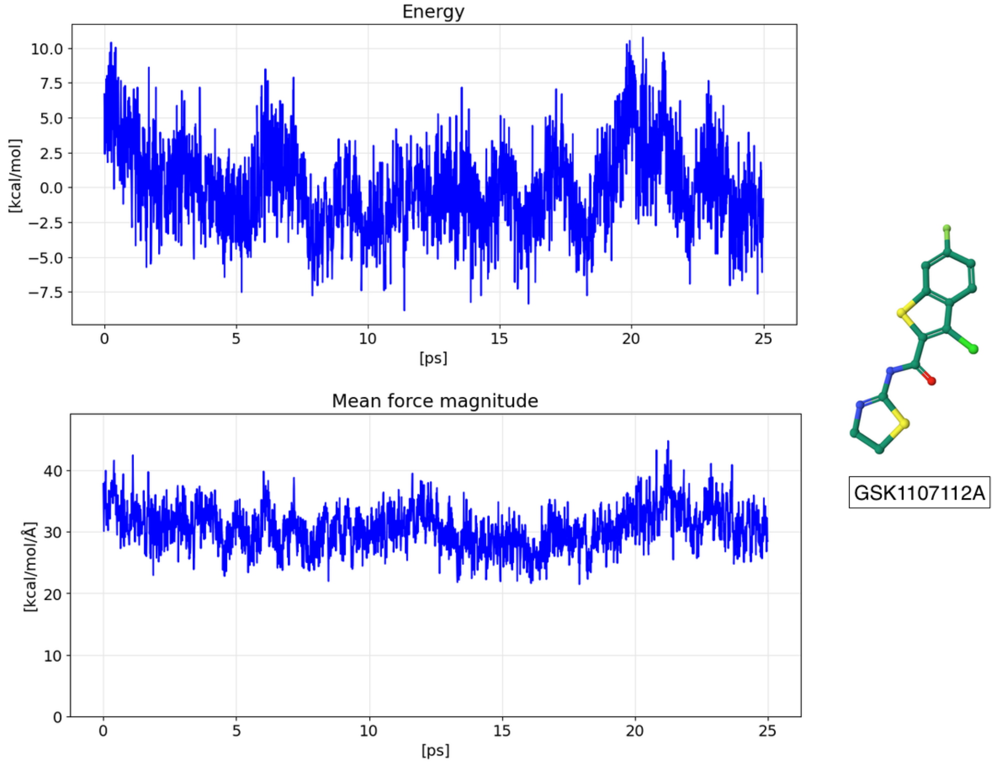


Figure 5.10: Energies (shifted to the mean) and mean forces' magnitudes for the final 25 ps of a molecular dynamics trajectory at 300 K using OBIWAN as accelerated force field. The drug ligand GSK1107112A was selected as it contains all atomic elements (H, C, N, O, S, F, and Cl) considered during training. Refer to the main text for a detailed description of this result.

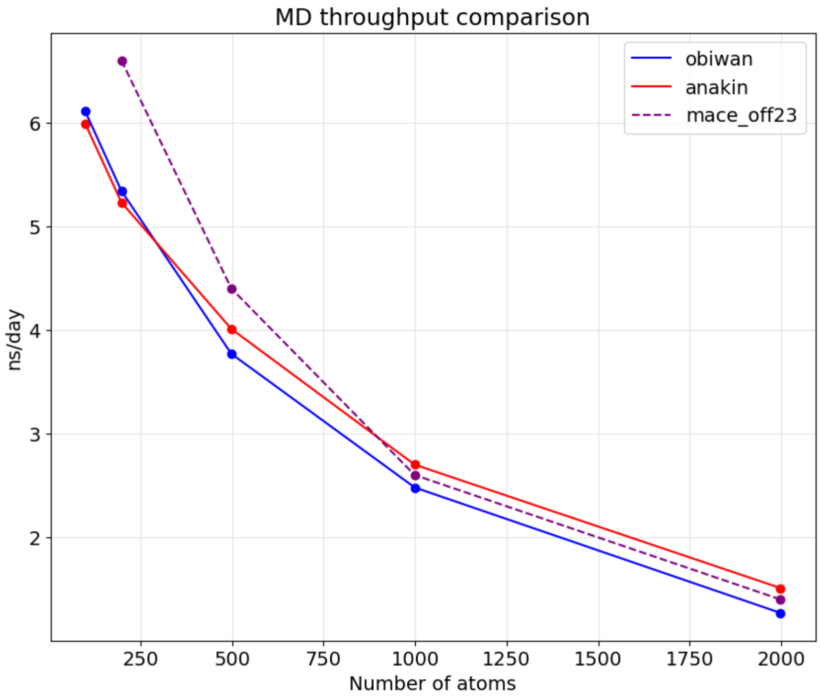


Figure 5.11: Throughput performance comparison of OBIWAN and ANAKIN. We conducted a series of MD runs on liquid water systems at 300 K and 1 g/cm^3, varying the number of molecules and using a 1 fs timestep. For this test, we used the TensorFlow version of ANAKIN that we implemented, as using the original PyTorch+CUDA code would have introduced unfair technological differences. All simulations were performed on a single A100 GPU. The MACE-OFF23 [78] performance curve is also included as reference.

Figure 5.11 presents a comparison of the inference performance between OBIWAN and ANAKIN on liquid water systems. Our modifications to the ANAKIN framework did not result in a degradation of its remarkable speed. It's also worth noting that we did not invest engineering effort in optimizing the MD engine code for performance (*e.g.* with domain decomposition or Verlet neighbor lists for linear scaling), leaving such production concerns for future framework developments. Additionally, a comparison with the recently released MACE-OFF architecture shows comparable performance despite not using LAMMPS [130], a much more optimized MD code.

OBIWAN's key feature is its ability to serve as a pre-trained model for diverse datasets, including those with previously unseen elements. To demonstrate this capability, we conducted the following experiment.

We trained OBIWAN on the $627,692$ energies ($\Rightarrow w = 0$) of `SPICE`, which includes molecules formed by H, C, N, O, F, P, S, Cl, Br, and I atoms. We performed two training runs: the first initializing OBIWAN from scratch; the second utilizing the OBIWAN instance pre-trained on the 9.6M molecules from the complete ANAKIN dataset. For the latter, we froze the first five layers of $\overrightarrow{\mathcal{M}}_{\text{rad}}$ and $\overrightarrow{\mathcal{M}}_{\text{ang}}$, and the first two layers of our Single MLP.

It's important to note that we employed the same OBIWAN architecture (pre-trained in one case) from the previous experiment, without altering any model or training procedure hyperparameters. As previously explained, our network *topology* disregards the chemical identity of encountered atoms, allowing users to incorporate new element types (three in this case: P, Br, and I) without modifications. Conversely, using ANAKIN in this manner is mathematically impossible due to its architecture's inherent dependence on the elements considered during the "pre-training". Moreover, to maintain a constant number of free parameters with ANAKIN for this new dataset, element-wise MLPs would need to be reduced as the AEV size increases: a 10 elements ANAKIN would result in a 1920-components AEVs, while OBIWAN's DeepAEVs remain constant at 384 components.

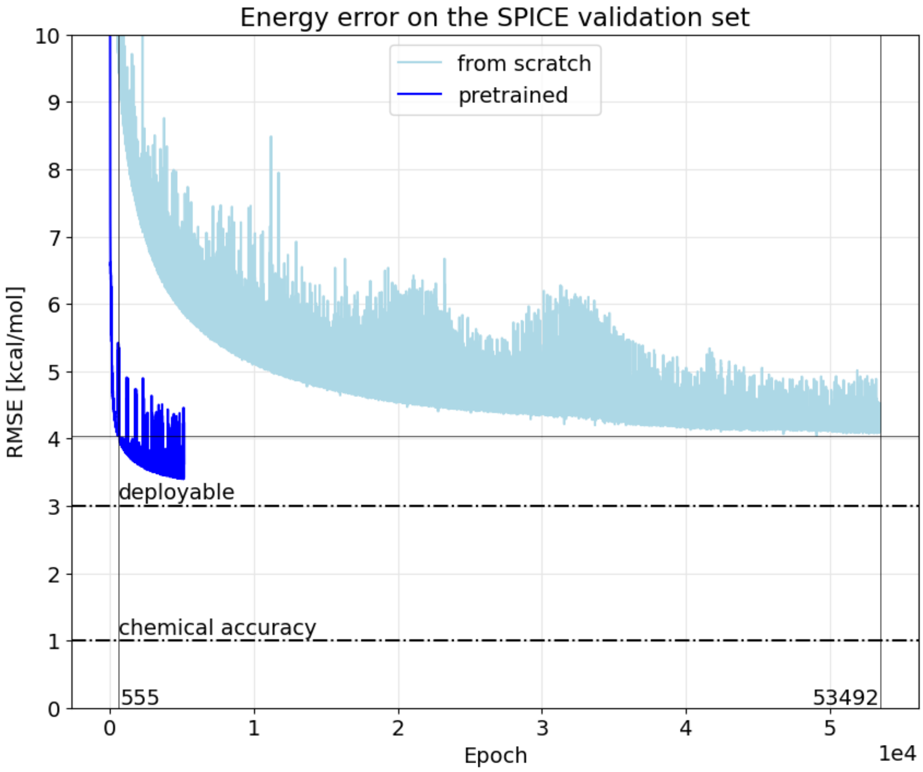


Figure 5.12: RMSE progression on the SPICE validation set for OBIWAN trained: (i) from scratch; (ii) after pre-training on the combined ani1x and ani2x datasets.

Given the extensive hyper-parameter search this would necessitate, we excluded ANAKIN from this final comparison.

Figure 5.12 displays the RMSE curves of the two version of OBIWAN over the validation set, demonstrating that starting from a pre-trained model enables a much faster convergence compared to starting from scratch. For instance, at epoch 5000, the pre-trained OBIWAN achieves an RMSE of approximately 3.4 kcal/mol, whereas the OBIWAN trained from scratch reaches about 6 kcal/mol with a significantly slower convergence. This indicates effective re-utilization of previously learned chemical and physical knowledge, making the model valuable even for a "small" dataset like the one used here, thus mitigating the substantial cost of DFT computations for training set construction. Consequently, this approach can save nearly two weeks of computing time on 4 A100 Nvidia GPUs for this training setup, resulting in notable energy savings and enhanced sustainability of the entire procedure.

To contextualize our results within the broader scope of neural network models beyond feed-forward networks, we compare our findings with the recent GNN-based MACE-OFF23 models on the SPICE dataset [78]. The MACE-OFF model with an inference speed comparable to OBIWAN achieves slightly better accuracy on this task: 2.5 kcal/mol vs 3.4 kcal/mol. This value was estimated from the paper's figures, assuming an average of 40 atoms per molecule and a reported error of 2.7 meV/atom on the PubChem SPICE subset [43]. However, it's worth noting that this result was achieved by: (i) training on a carefully augmented dataset; (ii) considering a "purified" subset of samples, recomputing DFT energies and forces for detected outliers that, according to the authors, contained "errors in the underlying electronic structure calculations" which skew the metrics with nonphysical high-error; (iii) conducting the training procedure in float64 precision, trading accuracy for memory footprint. Concluding, we want to repeat that in this experiment SPICE is not used as a benchmark to test OBIWAN's maximum achievable accuracy, but to showcase its effective transferability to unseen chemical species.

Chapter 6

Software

To run all the experiments we just presented, we developed an extensive, open-source codebase mostly available in our dedicated GitHub repository [91]. In the following sections we give a panoramic of all the software employed.

6.1 anakin-tf

We started our research by re-implementing the original ANAKIN framework [97] in TensorFlow [129].

The main effort here was the development of the AEVs' logic, which is essentially based on some combinatorial algebra techniques not common in classic neural networks scenarios. Figure 6.1 illustrates the maximum, component-wise, absolute error between our AEV and the PyTorch "ground truth" computed for more than $20,000$ atoms. As the reader can see, our TensorFlow results are equivalent to the original ones given the float32

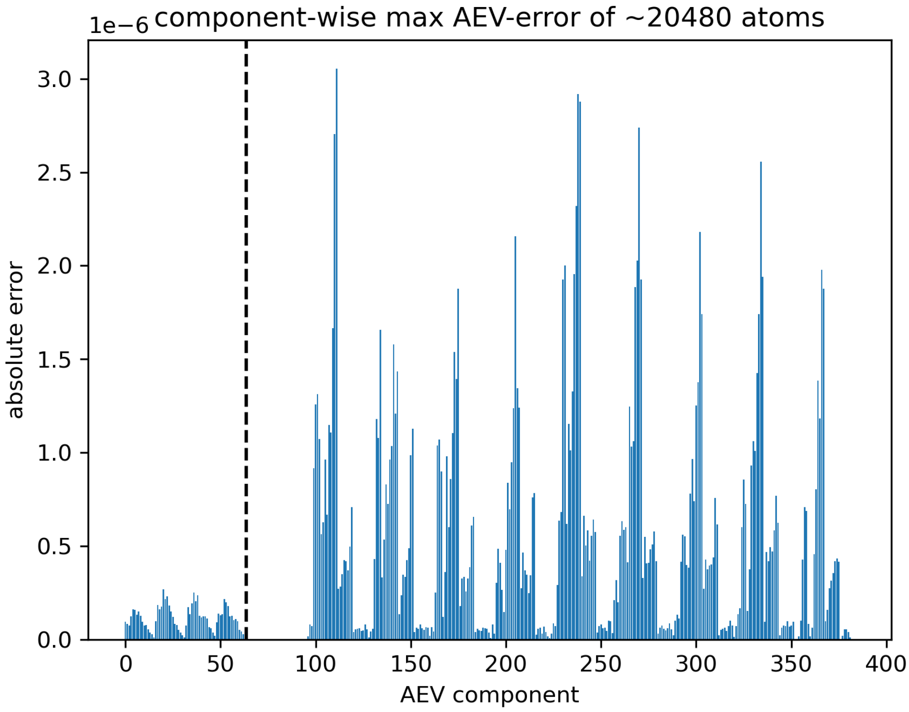


Figure 6.1: Numerical comparison between the original PyTorch AEVs and our TensorFlow implementation. The vertical dotted line separates the radial part from the angular part.

precision employed during the computation.

In the OBIWAN/architectures/anakin.py file of our repo one can find the ANAKIN-2x implementation we assembled following all the mathematical details given by the Authors [38]. In the same directory, also all the W-ANAKIN versions discussed in this work of thesis can be found.

6.2 obiwan-tf

The TensorFlow implementation of OBIWAN [91] is our main contribution. It includes all the models, data preprocessing, and training codes necessary to reproduce our results.

We ran our experiments in a Singularity [79] container. Its recipe can be found at OBIWAN/container.

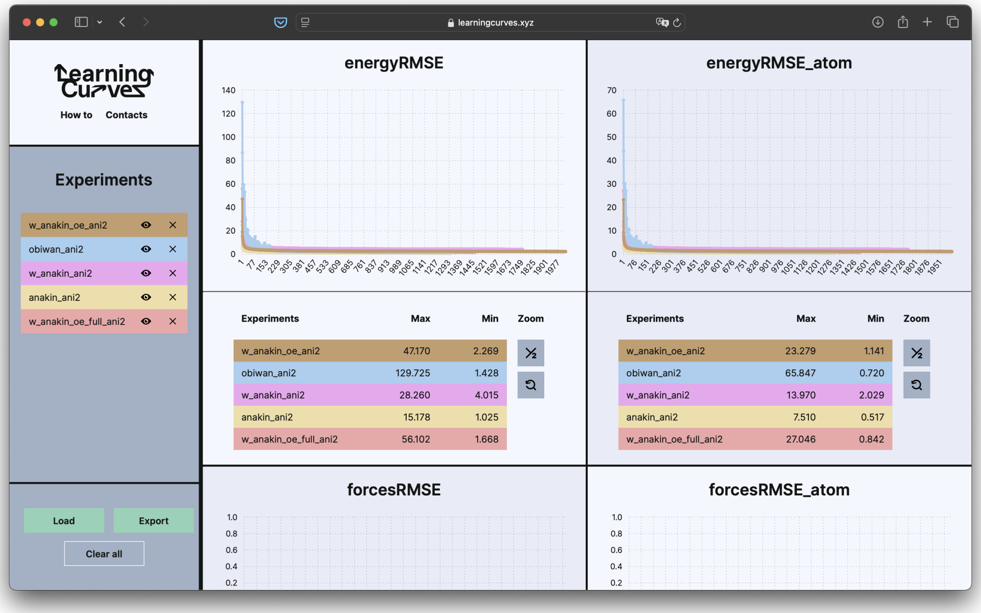
6.3 tensorflow-md

When we had to run molecular dynamics (MD) simulations in order to let OBIWAN produce the results shown in Figure 5.10, we didn't find any maintained or functioning MD engine interfaceable with TensorFlow models. All the major libraries recently shifted to the PyTorch paradigm [42]. For this reason we implemented tensorflow-md, released with our GitHub repo and reachable at OBIWAN/results/MD. It consists of a minimal MD engine completely written in TensorFlow. For equation integration, we employed a leap-frog integrator with impulsive Langevin for temperature control [52].

6.4 LearningCurves

During our research we often had to share the logged metrics of our experiments between the parts via internet. Unfortunately, we found that all commercial tools developed for this purpose exhibited characteristics not suitable for universities or institutions like us, primarily in terms of privacy: data had to be sent to their servers, stored in cloud, etc. On the other hand, all the available open-source solutions that we found had an uncomfortable user experience and, most importantly, they ran on PC only, which slowed our workflow.

We built LearningCurves to solve these issues. LearningCurves is a free, privacy-preserving, cross-platform web app we developed to share our machine learning metrics: no installation is required, running in the browser in pure javascript; it is accessible from every online device; the computation is managed by the frontend only, with zero data sent to the cloud. It is currently live at https://learningcurves.xyz/.



Experiments	Max	Min
w_anakin_oe_ani2	47.170	2.269
obiwan_ani2	129.725	1.428
w_anakin_ani2	28.260	4.015
anakin_ani2	15.178	1.025
w_anakin_oe_full_ani2	56.102	1.668

Experiments	Max	Min
w_anakin_oe_ani2	23.279	1.141
obiwan_ani2	65.847	0.720
w_anakin_ani2	13.970	2.029
anakin_ani2	7.510	0.517
w_anakin_oe_full_ani2	27.046	0.842

Figure 6.2: Screenshot of a typical LearningCurves dashboard.

Chapter 7

Conclusions

The development of quantum-accurate machine learning potentials has become a well-established research line. However, significant room for improvement remains, as a widely accepted foundation model is still lacking. A crucial aspect in developing such a model is the systematic incorporation of new atomic elements over time. In this thesis we introduced OBIWAN [90], a feed-forward neural network potentials that addresses this challenge with remarkable flexibility, overcoming limitations of previous models. As shown by our results, OBIWAN demonstrates transferability and leverages prior knowledge when learning from new data, even when encountering previously unseen chemical elements.

It is also worth reminding that computational efforts should align with green computing principles, minimizing resource usage where possible. As datasets grow, we anticipate increased focus on efficient computing, enabling scientific progress through sustainable methods. OBIWAN inherently embodies this vision, allowing users to achieve desired performance with much fewer training epochs by exploiting its generalized and re-usable architectural topology.

Our future research aims to achieve ANAKIN-level accuracy [121] through a series of network enhancements that we are planning. For the present work we care mostly about the architecture of the model, which is the part that was invalidating in previous methods. We didn't extensively optimise the hyper-parameter values of the architecture. Despite being a very computationally expensive job to carry on, we believe that it could improve significantly the accuracy and the generalization capabilities of OBIWAN, as it already did also in other deep learning scenarios [83].

With the same aim in mind, we think that an implementation of message passing techniques [50] is worth exploring too, trying to figure out in which real use cases the added computational effort could be a benefit. In this sense, the AIMNet series of potentials [139, 140, 4] is certainly the best reference, being it a graph neural network evolution of the original ANAKIN framework. In fact, in one of its published version AIMNet uses ANAKIN's symmetry functions as node embeddings, suggesting that our DeepAEVs formulation could be a possible enhancement of the pipeline. Also, our straightforward way to include novel elements without *any* modification of the weights matrices could be a very interesting feature for the end user. On the other hand, with AIMNet one should manually modify the tensors involved, updating the "implemented species" list [66] after every chemical upgrade.

We also plan to create customised version of OBIWAN tailored for RNA, DNA, and metals, perhaps employing targeted active learning rounds [122]. While classical molecular mechanics force fields exist for these structures, they lack the precision of those designed for proteins. In fact, developing advanced machine learning based force fields for nucleic acids could significantly advance studies of RNA-based vaccines, long noncoding RNAs, and other atomic-level investigations.

Concluding, we believe that OBIWAN can find significant application in several scenarios: single point energy estimation, minimization, dihedral scanning for molecular mechanics parameterizations and molecular dynamics. We ultimately conjecture that most of the molecular mechanics *stack* will be based on neural network based computations in the next years.

“Ringraziamenti”

These final acknowledgements paragraphs will be in italian because that is the language of all the interested people.

Comincio con il ringraziare doverosamente, ma anche molto sinceramente, le due persone che mi hanno attivamente permesso di realizzare questo grande obbiettivo: il professor Andrea Cavalli e il dottor Sergio Decherchi. Ricordo perfettamente le giornate passate a ricaricare la pagina web dell’università per vedere se erano usciti i primi risultati del concorso, poi le prove orali e poi ancora le telefonate, finali e quindi iniziali, in cui mi dicevate “Stefano vorremmo lavorare con te”. Ricordo i primi mesi, in cui non sapevo sostanzialmente nulla di Scienza con la S maiuscola, e gli ultimi, durante i quali tutte le parti del mosaico si sono ricollegate a formare il bellissimo disegno che ci eravamo promessi. Grazie davvero a entrambi, perché senza il vostro big bang questo universo non esisterebbe.

Procedendo logicamente e rimanendo ancora in ambito lavorativo (aggettivo quest’ultimo che ho la fortuna di fare davvero fatica a distinguere da “amicale”), devo senz’altro ringraziare i miei colleghi di rn-ai. Carlo, Stefano, Luca e Riccardo sono stati contemporaneamente fonti di ispirazione, aiuto concreto (*i.e.* supporto psicologico) e preziosi compagni di vita. La mia Compagnia dell’Anello contro il male della burocrazia italiana, senza la quale sarei tranquillamente perito dopo il primo PhD Day. Vi voglio bene perché siete stati tantissimo e farò tutto ciò che è nelle mie possibilità affinché continuiate a esserlo.

Io senza la vigorosa rete di amici che mi sorregge non sarei niente, l'ho sempre sostenuto e questo paragrafo è dedicato a loro. A tutte le persone che negli ultimi quattro anni hanno colorato i miei giorni e le mie notti, dai giochi da tavolo alle abbuffate di sushi, dalle feste in collina alle colazioni in piazza la domenica mattina. Un grazie particolare a Lorenzo e Giovanni, miei assistenti in questi anni, che metto nello stesso paragrafo degli amici perché tali si sono fin da subito rivelati.

Grazie infine alla mia famiglia, per tutto quello che fa da ventotto anni a questa parte. In particolarissimo modo, grazie a Paola, Giovanna e Aldo per essere la mia casa. E grazie Cesare, perché sei stato il mio migliore amico e lo sarai per sempre.

Per sempre vostro,

Stefano Martiro

Bibliography

[1] ACS Publication, Terms and Rights in the Journal Publishing Agreement. https://pubs.acs.org/page/copyright/journals/jpa_faqs.html

[2] Allen, M. P.; Tildesley, D. J., Computer Simulation of Liquids: Second Edition. *OUP Oxford* **2017**.

[3] Allinger, N. L.; Yuh, Y. H.; Lii, J.-H., Molecular Mechanics. The MM3 Force Field for Hydrocarbons. *J. Am. Chem. Soc.* **1989**, 111, 8551-8566.

[4] Anstine, D.; Zubatyuk, R.; Isayev, O., AIMNet2: A Neural Network Potential to Meet your Neutral, Charged, Organic, and Elemental-Organic Needs. *ChemRxiv* **2024**, doi:10.26434/chemrxiv-2023-296ch-v2.

[5] Artrith, N.; Morawietz, T.; Behler, J., High-Dimensional Neural-Network Potentials for Multicomponent Systems: Applications to Zinc Oxide. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2011**, 83, 153101.

[6] Ashcroft, N. W.; Mermin, N. D., Solid State Physics. *Harcourt College Publishers* **1976**.

[7] Bartók, A. P.; Kondor, R.; Csányi, G., On Representing Chemical Environments. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, 87, 184115.

[8] Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G., Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, 104, 136403

[9] Behler, J., Atom-Centered Symmetry Functions for Constructing High-Dimensional Neural Network Potentials. *J. Chem. Phys.* **2011**, 134, 074106.

[10] Behler, J., Constructing High-Dimensional Neural Network Potentials: A Tutorial Review. *Int. J. Quantum Chem.* **2015**, 115, 1032-1050.

[11] Behler, J., First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems. *Angew. Chem., Int. Ed.* **2017**, 56, 12828-12840.

[12] Behler, J., Four Generations of High-Dimensional Neural Network Potentials. *Chemical Reviews* **2021**, 121 (16), 10037-10072.

[13] Behler, J., Neural Network Potential-Energy Surfaces in Chemistry: a Tool for Large-Scale Simulations. *Phys. Chem. Chem. Phys.* **2011**, 13, 17930-17955.

[14] Behler, J., Perspective: Machine Learning Potentials for Atomistic Simulations. *J. Chem. Phys.* **2016**, 145, 170901.

[15] Behler, J.; Lorenz, S.; Reuter, K., Representing Molecule-Surface Interactions with Symmetry-Adapted Neural Networks. *J. Chem. Phys.* **2007**, 127, 014705.

[16] Behler, J.; Parrinello, M., Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, 98, 146401.

[17] Berendsen, H. J. C.; Postma, J. P. M.; Gunsteren, W. F. v.; DiNola, A.; Haak, J. R., Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, 81 (8), 3684-3690.

[18] Berendsen, H. J. C.; van der Spoel, D.; van Drunen, R., GROMACS: A message-passing parallel molecular dynamics implementation. *Comp. Phys. Comm.* **1995**, 91 (1), 43-56.

[19] Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J., Neural Network Models of Potential Energy Surfaces. *J. Chem. Phys.* **1995**, 103, 4129-4137.

[20] Born, M.; Huang, K., Dynamical theory of crystal lattices. *Clarendon press* **1954**.

[21] Born, M.; Oppenheimer, R., Zur Quantentheorie der Molekeln. *Ann. Phys.* **1927**, 389 (20), 457-484.

[22] Brooks, B. R.; Brooks, C. L.; MacKerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; Caflisch, A.; Caves, L.; Cui, Q.; Dinner, A. R.; Feig, M.; Fischer, S.; Gao, J.; Hodoscek, M.; Im, W.; Kuczera, K.; Lazaridis, T.; Ma, J.; Ovchinnikov, V.; Paci, E.; Pastor, R. W.; Post, C. B.; Pu, J. Z.; Schaefer, M.; Tidor, B.; Venable, R. M.; Woodcock, H. L.; Wu, X.; Yang, W.; York, D. M.; Karplus, M., CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, 30 (10), 1545-1614.

[23] Bussi, G.; Donadio, D.; Parrinello, M., Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, 126 (1), 014101.

[24] Car, R.; Parrinello, M., Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **1985**, 55 (22), 2471-2474.

[25] Carleo, G.; Cirac, I.; Cranmer, K.; Daudet, L.; Schuld, M.; Tishby, N.; Vogt-Maranto, L.; Zdeborova, L., Machine Learning and the Physical Sciences. *Rev. Mod. Phys.* **2019**, 91, 045002.

[26] Chandler, D., Introduction to Modern Statistical Mechanics. *Oxford University Press* **1987**.

[27] Christensen, A. S.; Bratholm, L. A.; Faber, F. A.; von Lilienfeld, O. A., FCHL Revisited: Faster and More Accurate Quantum Machine Learning. *J. Chem. Phys.* **2020**, 152, 044107.

[28] Christensen, A. S.; Faber, F. A.; von Lilienfeld, O. A., Operators in Quantum Machine Learning: Response Properties in Chemical Space. *J. Chem. Phys.* **2019**, 150, 064105.

[29] Cohen, R. E.; Mehl, M. J.; Papaconstantopoulos, D. A., Tight-Binding Total-Energy Method for Transition and Noble Metals. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1994**, 50, 14694-14697.

[30] Colombo, L., Tight-Binding Molecular Dynamics Simulations. *Comput. Mater. Sci.* **1998**, 12, 278-287.

[31] Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R., Jr; Merz, K.M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **1995**, 117, 5179-5197.

[32] Crespos, C.; Collins, M. A.; Pijper, E.; Kroes, G. J., Multi-Dimensional Potential Energy Surface Determination by Modified Shepard Interpolation for a Molecule-Surface Reaction: H2+Pt(111). *Chem. Phys. Lett.* **2003**, 376, 566-575.

[33] Cybenko, G., Approximation by Superpositions of a Sigmoidal Function. *Math. Control Sign. Systems* **1989**, 2, 303-314.

[34] Darden, T.; York, D.; Pedersen, L., Particle mesh Ewald: An Nlog(N) method for Ewald sums in large systems. *J. Chem. Phys.* **1993**, 98 (12), 10089-10092.

[35] Darley, M. G.; Handley, C. M.; Popelier, P. L. A., Beyond Point Charges: Dynamic Polarization from Neural Net Predicted Multipole Moments. *J. Chem. Theory Comput.* **2008**, 4, 1435-1448.

[36] Deng, Z.; Chen, C.; Li, X.-G.; Ong, S. P., An Electrostatic Spectral Neighbor Analysis Potential for Lithium Nitride. *npj Computational Materials* **2019**, 5, 75.

[37] Deringer, V. L.; Csányi, G., Machine-Learning Based Interatomic Potential for Amorphous Carbon. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, 95, 094203.

[38] Devereux, C.; Smith, J.; Davis, K.; Barros, K.; Zubatyuk, R.; Isayev, O.; Roitberg, A., Extending the Applicability of the ANI Deep Learning Molecular Potential to Sulfur and Halogens. *Journal of Chemical Theory and Computation* **2020**, DOI: 10.1021/acs.jctc.0c00121.

[39] Dick, S.; Fernandez-Serra M., Machine learning accurate exchange and correlation functionals of the electronic density. *Nat. Commun.* **2020**, Jul 14;11(1):3509, DOI: 10.1038/s41467-020-17265-7.

[40] Dral, P. O., Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, 11, 2336-2347.

[41] Drautz, R., Atomic Cluster Expansion for Accurate and Transferable Interatomic Potentials. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2019**, 99, 014104.

[42] Eastman, P., https://github.com/openmm.

[43] Eastman, P.; Behara, P.; Dotson, D.; Galvelis, R.; Herr, J.; Horton, J.; Mao, Y.; Chodera, J.; Pritchard, B.; Wang, Y.; *et al.*, SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. *Sci Data* **2023**, DOI: 10.1038/s41597-022-01882-6.

[44] Elstner, M.; Cui, Q.; Munih, P.; Kaxiras, E.; Frauenheim, T.;Karplus, M., Modeling Zinc in Biomolecules with the Self Consistent Charge-Density Functional Tight Binding (SCC-DFTB) Method: Applications to Structural and Energetic Analysis. *J. Comput. Chem.* **2003**, 24, 565-581.

[45] Faraji, S.; Ghasemi, S. A.; Rostami, S.; Rasoulkhani, R.; Schaefer, B.; Goedecker, S.; Amsler, M., High Accuracy and Transferability of a Neural Network Potential Through Charge Equilibration for Calcium Fluoride. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, 95, 104105.

[46] Frenkel, D.; Smit, B., Understanding Molecular Simulation. *Academic Press* **2001**, 638.

[47] Gastegger, M.; Schwiedrzik, L.; Bittermann, M.; Berzsenyi, F.; Marquetand, P., wACSF - Weighted atom-centered symmetry functions as descriptors in machine learning potentials. *J. Chem. Phys.* **2018**, DOI: 10.1063/1.5019667.

[48] Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J.; Roitberg, A., TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Network Potentials. *Journal of Chemical Information and Modeling* **2020**, 60 (7) 3408-3415, DOI: 10.1021/acs.jcim.0c00451.

[49] Ghasemi, S. A.; Hofstetter, A.; Saha, S.; Goedecker, S., Interatomic Potentials for Ionic Systems with Density Functional Accuracy Based on Charge Densities Obtained by a Neural Network. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2015**, 92, 045131.

[50] Gilmer, J.; Schoenholz, S.; Riley, P.; Vinyals, O.; Dahl, G., Neural message passing for Quantum chemistry. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, **2017**.

[51] Goedecker, S., Linear Scaling Electronic Structure Methods. *Rev. Mod. Phys.* **1999**, 71, 1085-1123.

[52] Goga, N.; Rzepiela, A.; de Vries, A.; Marrink, S.; Berendsen, H., Efficient Algorithms for Langevin and DPD Dynamics. *Journal of Chemical Theory and Computation* **2012**, 8 (10), 3637-3649, DOI: 10.1021/ct3000876.

[53] Goringe, C. M.; Bowler, D. R.; Hernandez, E., Tight-Binding Modelling of Materials. *Rep. Prog. Phys.* **1997**, 60, 1447-1512.

[54] Handley, C. M.; Popelier, P. L. A., Dynamically Polarizable Water Potential Based on Multipole Moments Trained by Machine Learning. *J. Chem. Theory Comput.* **2009**, 5, 1474-1489.

[55] Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A., Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *J. Phys. Chem. Lett.* **2015**, 6, 2326-2331.

[56] He, K.; Zhang, X.; Ren, S.; Sun, J., Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2016**, pp. 770-778, DOI: 10.1109/CVPR.2016.90.

[57] Hirshfeld, F. L., Bonded-Atom Fragments for Describing Molecular Charge Densities. *Theor. Chim. Acta* **1977**, 44, 129-138.

[58] Hobday, S.; Smith, R.; Belbruno, J., Applications of Neural Networks to Fitting Interatomic Potential Functions. *Modell. Simul. Mater. Sci. Eng.* **1999**, 7, 397-412

[59] Hobday, S.; Smith, R.; Belbruno, J., Application of Genetic Algorithms and Neural Networks to Interatomic Potentials. *Nucl. Instrum. Methods Phys. Res., Sect. B* **1999**, 153, 247-263.

[60] Hoover, W. G., Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A* **1985**, 31 (3), 1695-1697.

[61] Hornik, K., Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* **1991**, 4, 251-257.

[62] Hornik, K.; Stinchcombe, M.; White, H., Multilayer Feedforward Networks are Universal Approximators. *Neural Networks* **1989**, 2, 359-366.

[63] Huang, K., Statistical Mechanics. *Jonh Wiley and Sons* **1928**.

[64] Huo, H.; Rupp, M., Unified Representation of Molecules and Crystals for Machine Learning. *Mach. Learn.: Sci. Technol* **2022**, 3 045017, DOI: 10.1088/2632-2153/aca005.

[65] Ingólfsson, H. I.; Lopez, C. A.; Uusitalo, J. J.; de Jong, D. H.; Gopal, S. M.; Periole, X.; Marrink, S. J., The power of coarse graining in biomolecular simulations. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, 4 (3), 225-248.

[66] Isayev, O., https://github.com/isayevlab/AIMNet2/blob/main/aimnet2calc/aimnet2a

[67] Ischtwan, J.; Collins, M. A., Molecular Potential Energy Surfaces by Interpolation. *J. Chem. Phys.* **1994**, 100, 8080-8088.

[68] Jindal, S.; Chiriki, S.; Bulusu, S. S., Spherical Harmonics Based Descriptor for Neural Network Potentials: Structure and Dynamics of Au147 Nanocluster. *J. Chem. Phys.* **2017**, 146, 204301.

[69] Jones, J. E.; Sc., D., On the determination of molecular fields. —II. From the equation of state of a gas. *Proc. R. Soc. Lond. A* **1924**, 106 (738), 463-477.

[70] Jones, R. O., Density functional theory: Its origins, rise to prominence, and future. *Rev. Mod. Phys.* **2015**, 87, 897.

[71] Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R.W.; Klein, M. L., Comparison of Simple Potential Functions for Simulating Liquid Water. *J. Chem. Phys.* **1983**, 79, 926-935.

[72] Jorgensen, W. L.; Tirado-Rives, J., The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin. *J. Am. Chem. Soc.* **1988**, 110 (6), 1657-1666.

[73] Kim, K.; Jordan, K. D., Comparison of Density Functional and MP2 Calculations on the Water Monomer and Dimer. *The Journal of Physical Chemistry* **1994**, 98 (40), 10089-10094.

[74] Kingma, D.; Ba, J., Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* **2014**, DOI: 10.48550/arXiv.1412.6980.

[75] Kitchin, J. R., Machine Learning in Catalysis. *Nature Catalysis* **2018**, 1, 230-232.

[76] Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J., A Fourth-Generation High-Dimensional Neural Network Potential with Accurate Electrostatics Including Non-Local Charge Transfer. *Nat. Commun.* **2020**, 398.

[77] Kohn, W.; Sham, L. J., Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, 140, A1133.

[78] Kovács, D.; Moore, J.; Browning, N.; Batatia, I.; Horton, J.; Kapil, V.; Witt, W.; Magdău, I.; Cole, D.; Csányi, G., MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules. *arXiv:2312.15211v2 [physics.chem-ph],* DOI: 10.48550/arXiv.2312.15211.

[79] Kurtzer, G. M. *et al.*, Singularity 2.5.2 - Linux application and environment containers for science. *Zenodo* **2018**, https://doi.org/10.5281/zenodo.1308868.

[80] Leeuw, S. W. d.; Perram, J. W.; Smith, E. R., Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* **1980**, 373 (1752), 27-56.

[81] Lewars, E. G., Computational Chemistry. *Springer Netherlands* **2011**, 664.

[82] Liu, M.; Kitchin, J., SingleNN: Modified Behler–Parrinello Neural Network with Shared Weights for Atomistic Simulations with Transferability. *The Journal of Physical Chemistry* **2020** 124 (32), 17811-17818, DOI: 10.1021/acs.jpcc.0c04225.

[83] Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V., RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, https://doi.org/10.48550/arXiv.1907.11692.

[84] Zhang, L.; Han, J.; Wang, H.; Saidi, W.; Car, R.; Weinan, E., End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems. *In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*, DOI: 10.48550/arXiv.1805.09003.

[85] MacKerell, A. D., Jr; *et al.*, All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *J. Phys. Chem. B* **1998**, 102, 3586-3616.

[86] Makarov, D. E.; Metiu, H., Fitting Potential Energy Surfaces: A Search in the Function Space by Directed Genetic Programming. *J. Chem. Phys.* **1998**, 108, 590-598.

[87] Malshe, M.; Narulkar, R.; Raff, L. M.; Hagan, M.; Bukkapatnam, S.; Agrawal, P. M.; Komanduri, R., Development of Generalized Potential-Energy Surfaces Using Many-Body Expansions, Neural Networks, and Moiety Energy Approximations. *J. Chem. Phys.* **2009**, 130, 184102.

[88] Manzhos, S.; Carrington, T., Jr, Using Neural Networks to Represent Potential Surfaces as Sums of Products. *J. Chem. Phys.* **2006**, 125, 194105.

[89] Manzhos, S.; Carrington, T., Jr, Using Redundant Coordinates to Represent Potential Energy Surfaces with Lower-Dimensional Functions. *J. Chem. Phys.* **2007**, 127, 014103.

[90] Martire, S.; Decherchi, S.; Cavalli, A., OBIWAN: An Element-Wise Scalable Feed-Forward Neural Network Potential. *Journal of Chemical Theory and Computation* **2024**, DOI: 10.1021/acs.jctc.4c00342.

[91] Martire, S.; Decherchi, S., https://github.com/virtualmartire/OBIWAN.

[92] Marx, D.; Hutter, J., Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods. *Cambridge University Press: Cambridge* **2009**.

[93] Mayo, S. L.; Olafson, B. D.; Goddard, W. A., III DREIDING: A Generic Force Field for Molecular Simulations. *J. Phys. Chem.* **1990**, 94, 8897-8909.

[94] Morawietz, T.; Sharma, V.; Behler, J., A Neural Network Potential-Energy Surface for the Water Dimer Based on Environment-Dependent Atomic Energies and Charges. *J. Chem. Phys.* **2012**, 136, 064103.

[95] Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C., Machine Learning for Molecular Simulation. *Annu. Rev. Phys. Chem.* **2020**, 71, 361-390.

[96] Nosé, S., A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **1984**, 81 (1), 511-519.

[97] Open Consortium for AI in Quantum Chemistry, https://github.com/aiqm/torchani.git.

[98] Parrinello, M.; Rahman, A., Polymorphic transitions in single crystals: A new molecular dynamics method. *J. Appl. Phys.* **1981**, 52 (12), 7182-7190.

[99] Parsaeifard, B.; Finkler, J. A.; Goedecker, S., Detecting Non-Local Effects in the Electronic Structure of a Simple Covalent System with Machine Learning Methods. *arXiv:2008.11277v1* **2020**.

[100] Parsaeifard, B.; Sankar De, D.; Christensen, A.; Faber, F.; Kocer, E.; De, S.; Behler, J.; von Lilienfeld, O.; Goedecker, S., An assessment of the structural resolution of various fingerprints commonly used in machine learning. *Mach. Learn.: Sci. Technol.* **2021**, DOI 10.1088/2632-2153/abb212.

[101] Pauling, L.; Wilson, E. B., Introduction to quantum mechanics with applications to chemistry. *Courier Corporation* **2012**.

[102] Perdew, J. P.; Burke, K.; Ernzerhof, M., Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, 77 (18), 3865-3868.

[103] Plé, T.; Lagardère, L.; Piquemal, J., Force-field-enhanced neural network interactions: from local equivariant embedding to atom-in-molecule properties and long-range effects. *Chem. Sci.* **2023**, DOI: 10.1039/D3SC02581K.

[104] Profitt, T.; Pearson, J., A shared-weight neural network architecture for predicting molecular properties. *Phys. Chem. Chem. Phys.* **2019** ,21, 26175-26183, DOI: 10.1039/C9CP03103K.

[105] Pun, G. P. P.; Batra, R.; Ramprasad, R.; Mishin, Y., Physically Informed Artificial Neural Networks for Atomistic Modeling of Materials. *Nature Comm.* **2019**, 10, 2339.

[106] Raff, L. M.; Malshe, M.; Hagan, M.; Doughan, D. I.; Rockley,M. G.; Komanduri, R., Ab initio Potential-Energy Surfaces for Complex, Multichannel Systems Using Modified Novelty Sampling and Feedforward Neural Networks. *J. Chem. Phys.* **2005**, 122, 084104.

[107] Ramakrishnan, R.; Dral, P.; Rupp, M.; von Lilienfeld, O., Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, 1, 140022, DOI: https://doi.org/10.1038/sdata.2014.22.

[108] Ranasinghe, D.; Petersson, G., CCSD(T)/CBS atomic and molecular benchmarks for H through Ar. *J Chem Phys.* **2013**, DOI: 10.1063/1.4798707.

[109] Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A., III; Skiff, W. M., UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations. *J. Am. Chem. Soc.* **1992**, 114, 10024-10035.

[110] Rosenblatt, F., The Perceptron: A Probabilistic Model For Information Storage And Organization in the Brain. *Psychological Review* **1958**, 65, 386–408.

[111] Rostami, S.; Amsler, M.; Ghasemi, S., Optimized symmetry functions for machine-learning interatomic potentials of multicomponent systems. *J. Chem. Phys.* **2018**, 149 (12) 124106, DOI: 10.1063/1.5040005.

[112] Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C., Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *J. Comput. Phys.* **1977**, 23 (3), 327-341.

[113] Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A., Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, 108, 058301.

[114] Salomon-Ferrer, R.; Case, D. A.; Walker, R. C., An overview of the Amber biomolecular simulation package. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2013**, 3 (2), 198-210.

[115] Schütt, K.; Kindermans, P.-J.; Felix, H. E. S.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R., SchNet: A Continuous-Filter Convolutional Neural Network for Modeling Quantum Interactions. *Advances in Neural Information Processing Systems 30 (NIPS 2017)* **2017**, 991-1001.

[116] Segler, M. H. S.; Preuss, M.; Waller, M. P., Planning Chemical Syntheses with Deep Neural Networks and Symbolic AI. *Nature* **2018**, 555, 604-610.

[117] Shapeev, A. V., Moment Tensor Potentials: a Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **2016**, 14, 1153-1173.

[118] Sholl, D. S.; Steckel, J. A., Density Functional Theory: A Practical Introduction. *John Wiley and Sons, Inc.* **2009**, ISBN: 9780470373170.

[119] Sifain, A. E.; Lubbers, N.; Nebgen, B. T.; Smith, J. S.; Lokhov, A. Y.; Isayev, O.; Roitberg, A. E.; Barros, K.; Tretiak, S., Discovering a Transferable Charge Assignment Model Using Machine Learning. *J. Phys. Chem. Lett.* **2018**, 9, 4495-4501.

[120] Singraber, A.; Behler, J.; Dellago, C., Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials. *J. Chem. Theory Comput.* **2019**, 15, 1827-1840.

[121] Smith, J.; Isayev, O.; Roitberg, A., ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, DOI: 10.1039/C6SC05720A.

[122] Smith, J.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A., Less is more: Sampling chemical space with active learning. *J. Chem. Phys.* **2018**, DOI: 10.1063/1.5023802.

[123] Smith, J.; Zubatyuk, R.; Nebgen, B.; Lubbers, N.; Barros, K.; Roitberg, A.; Isayev, O.; Tretiak, S., The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Sci Data 7* **2020**, DOI: 10.1038/s41597-020-0473-z.

[124] Zhang, S.; Makoś, M.; Jadrich, R.; Kraka, E.; Barros, K.; Nebgen, B.; Tretiak, S.; Isayev, O.; Lubbers, N.; Messerly, R.; *et al.*, Exploring the frontiers of condensed-phase chemistry with a general reactive machine learning potential. *Nature Chemistry* **2024**, DOI: 10.1038/s41557-023-01427-3.

[125] Sumpter, B. G.; Noid, D. W., Potential Energy Surfaces for Macromolecules. A Neural Network Technique. *Chem. Phys. Lett.* **1992**, 192, 455-462.

[126] Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R., A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **1982**, 76 (1), 637-649.

[127] Szabo, A.; Ostlund, N. S., Modern quantum chemistry: introduction to advanced electronic structure theory. *Courier Corporation* **1996**.

[128] Tai No, K.; Ha Chang, B.; Yeon Kim, S.; Shik Jhon, M.; Scheraga, H. A., Description of the Potential Energy Surface of the Water Dimer with an Artificial Neural Network. *Chem. Phys. Lett.* **1997**, 271, 152-156.

[129] TensorFlow Developers, TensorFlow (v2.18.0-rc1). *Zenodo* **2024**, https://doi.org/10.5281/zenodo.13901223.

[130] Thompson, A.; Aktulga, H.; Berger, R.; Bolintineanu, D.; Brown, W.; Crozier, P.; in 't Veld, P.; Kohlmeyer, A.; Moore, S.; Nguyen *et al.*, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*, Volume 271, **2022**, DOI: https://doi.org/10.1016/j.cpc.2021.108171.

[131] van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A. III, ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, 105, 9396-9409.

[132] van Gunsteren, W. F.; Weiner, P. K.; Wilkinson, A. J., Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications. *Springer Netherlands* **1997**, 3, 618.

[133] Verlet, L., Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, 159 (1), 98-103.

[134] Vitek, A.; Stachon, M.; Krömer, P.; Snasel, V., Towards the Modeling of Atomic and Molecular Clusters Energy by Support Vector Regression. *IEEE 5th International Conference on Intelligent Networking and Collaborative Systems* **2013**, 121-126.

[135] Wilmer, C. E.; Kim, K. C.; Snurr, R. Q., An Extended Charge Equilibration Method. *J. Phys. Chem. Lett.* **2012**, 3, 2506-2511.

[136] Wu, Y.; Kaiming, H., Group normalization. *Proceedings of the European conference on computer vision (ECCV)* **2018**, DOI: 10.48550/1803.08494.

[137] Zhang, Y.; Hu, C.; Jiang, B., Embedded Atom Neural Network Potentials: Efficient and Accurate Machine Learning with a Physically Inspired Representation. *The Journal of Physical Chemistry Letters* **2019** 10 (17), 4962-4967, DOI: 10.1021/acs.jpclett.9b02037.

[138] Zeiler, M. D.; Ranzato, M.; Monga, R.; Mao, M.; Yang, K.; Le, Q. V.; Nguyen, P.; Senior, A.; Vanhoucke, V.; Dean, J.; Hinton, G. E., On Rectified Linear Units for Speech Processing. *IEEE International Conference on Acoustics, Speech and Signal Processing* **2013**, 3517-3521.

[139] Zubatyuk, R.; Smith, J. S.; Leszczynski, J.; Isayev, O., Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network. *Sci. Adv. 5*, **2019**.

[140] Zubatyuk, R.; Smith, J. S.; Nebgen, B. T.; Tretiak, S.; Isayev, O., Teaching a neural network to attach and detach electrons from molecules. *Nat. Commun. 12*, **2021**.

[141] Zuo, Y.; Chen, C.; Li, X.; Deng, Z.; Chen, Y.; Behler, J.; Csanyi, G.; Shapeev, A. V.; Thompson, A. P.; Wood, M. A.; Ong, S. P., A Performance and Cost Assessment of Machine Learning Interatomic Potentials. *J. Phys. Chem. A* **2020**, 124, 731-745.

:wq