universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Thermodynamic properties using neural network potentials in MD-simulations

verfasst von | submitted by

Sandra Leibetseder BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 875
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Bioinformatik
Betreut von | Supervisor:	Univ.-Prof. Mag. Dr. Stefan Boresch

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Univ.-Prof. Mag. Dr. Stefan Boresch for supervising my thesis. Thank you for your time, patience and explanations throughout the project.

A special thanks to Dipl.-Ing. Anna Katharina Picha, BSc for introducing me to the topic, for continuous support during my work at the MDY and providing a lot of knowledge.

Thanks also to my family and friends for the support along the way. Furthermore, I am deeply grateful to my parents, Ulrike and Wolfgang, for their unconditional love and support during my years of education. I could not have achieved that without you.
Finally, I want to thank my soulmate and partner, Vincent, for keeping me motivated during this challenging time. Your encouragement and emotional support meant more than words can express.

Thank you!

Abstract

Molecular dynamics (MD) simulations are an established tool in physics, chemistry and biology. The vast majority of these simulations use force fields to approximate the interactions between the atoms and molecules of the system of interest. Recent efforts have focused on replacing these force fields with neural network potentials (NNPs), which promise to offer quantum-mechanical accuracy at reasonable computational costs. The aim of this study is to evaluate the performance of existing pre-trained NNPs in MD simulations of water and two organic liquids: benzene and n-hexane. Structural, thermodynamic, and dynamic properties were obtained from simulations and compared critically to experimental data to identify the strength and weaknesses of two NNPs. Specifically, the NNPs ANI-2x and MACE-OFF23(S) were used. Additionally, all three liquids were also simulated using CHARMM General Force Fields (CGenFF). The results revealed several shortcomings of the tested NNPs: Using ANI-2x, water is overly structured, and its diffusion coefficient is three orders of magnitude too slow. While MACE-OFF23(S) produces reasonable results for water, it significantly overestimates the density (12-18%) for the examined systems. Overall, the force field simulations agree best with the experiments. These findings show that, even though they are no longer in the early stages of development, careful consideration must be given to the specific scope of application of NNPs. Since NNP performance varies depending on the liquid studied, NNPs are not yet fully transferable across different systems. Improving NNP performance when applied to bulk liquids is likely to require larger, more diverse training datasets and direct, careful validation by computing condensed phase properties.

Keywords: MD-Simulation, Neural network potentials (NNP), Thermodynamic properties, MACE-OFF23, ANI-2x

Kurzfassung

Molekulardynamik-Simulationen (MD) sind ein etabliertes Werkzeug in Physik, Chemie und Biologie. Die überwiegende Mehrheit dieser Simulationen verwendet Kraftfelder, um die Wechselwirkungen zwischen den Atomen und Molekülen des betreffenden Systems zu approximieren. Jüngste Bemühungen konzentrieren sich darauf, diese Kraftfelder durch neuronale Netzwerkpotenziale (NNPs) zu ersetzen, die eine quantenmechanische Genauigkeit bei angemessenen Rechenkosten versprechen. Das Ziel dieser Arbeit ist es, die Leistung bestehender vortrainierter NNPs in MD-Simulationen von Wasser und zwei organischen Flüssigkeiten – Benzol und n-Hexan – zu bewerten. Strukturelle, thermodynamische und dynamische Eigenschaften wurden aus Simulationen gewonnen und kritisch mit experimentellen Daten verglichen, um die Stärken und Schwächen der beiden NNPs zu identifizieren. Konkret wurden die NNPs ANI-2x und MACE-OFF23(S) verwendet. Zusätzlich wurden alle drei Flüssigkeiten auch mit dem CHARMM General Force Field (CGenFF) simuliert. Die Ergebnisse zeigten mehrere Mängel der getesteten NNPs: Bei Verwendung von ANI-2x ist Wasser übermäßig strukturiert und der Diffusionskoeffizient ist um drei Größenordnungen zu langsam. Während MACE-OFF23(S) für Wasser vernünftige Ergebnisse liefert, überschätzt es die Dichte (12-18%) für die untersuchten Systeme erheblich. Insgesamt stimmen die Kraftfeldsimulationen am besten mit den Experimenten überein. Diese Ergebnisse zeigen, dass obwohl NNPs sich nicht mehr in der frühen Entwicklungsphase befinden, der spezifische Anwendungsbereich von NNPs sorgfältig abgewogen werden muss. Da die Leistung von NNPs je nach untersuchter Flüssigkeit variiert, sind NNPs noch nicht vollständig auf verschiedene Systeme übertragbar. Die Verbesserung der NNP-Leistung bei der Anwendung auf Flüssigkeiten in großen Mengen erfordert wahrscheinlich größere, vielfältigere Trainingsdatensätze und eine direkte, sorgfältige Validierung durch die Berechnung der Eigenschaften in der kondensierten Phase.

Keywords: MD-Simulationen, Neuronale Netzpotentiale (NNP), Thermodynamische Eigenschaften, MACE-OFF23, ANI-2x

Contents

List of Abbreviations

Å	Ångström
AEV	Atomic environment vector
FF	Force field
fs	Femtosecond
GNN	Graph neural network
LNG	Langevin integrator
MD	Molecular dynamics
MLP	Machine learning potential
MSD	Mean-squared displacement
NH	Nose Hoover integrator
NN(P)	Neural network (potential)
NPT	Isothermal-Isobaric Ensemble
ns	Nanosecond
NVT	Canonical Ensemble
OFF	Organic force field
QM	Quantum mechanics
RDF	Radial distribution function

1 Introduction

Computer simulations are an essential tool used in all kinds of scientific fields, from natural sciences over computer sciences to engineering. For example, astronomers simulate the formation of galaxies [1], engineers analyze fluid flows [2], and geophysicists model the earth's internal dynamics [3]. For chemists and biologists, molecular dynamics (MD) simulation plays a central role in understanding the behavior of atoms and molecules over time [4].

Many experimental techniques (nuclear magnetic resonance (NMR) spectroscopy, infrared (IR) spectroscopy, UV/VIS spectroscopy, mass spectrometry, X-ray crystallography, or electron microscopy) provide valuable structural information about biomolecules. However, these experimental methods are limited in spatial and temporal resolution. They mostly report ensemble averaged behavior and cannot capture the real-time motion of individual atoms or molecules [5]. Still, the structural insights they offer are important to set up simulations.

Computational methods, like molecular dynamics simulations, can give insights into the movement and interaction of molecules. This makes it possible to study molecular vibrations, thermodynamic properties, and even conformational changes at atomic resolution. MD simulations allow the study of such systems at a fraction of the cost compared to laboratory experiments. It also gives the possibility to investigate processes that are difficult or impossible to observe in the laboratory. This makes it very interesting in the field of drug design and discovery [5].

Empirical force fields

The most accurate way to describe intra- and intermolecular interactions as needed in MD would be through quantum mechanics (QM). QM methods provide approximate solutions to the exact laws describing interactions within and between molecules (i.e., the Schrodinger equation), with the achievable accuracy depending on the amount of computation one is willing to afford. However, even with today's computing and storage capacities, this is computationally so costly that sufficiently accurate solutions of the Schroedinger equation are impossible for many applications of interest. In 1969, the first publication mentioning empirical force fields (FF) as we use them today was published [6]. A FF uses physically motivated, but compared to QM significantly simplified expressions to describe interatomic and intermolecular interactions. Even today, therefore, most MD simulations are done using empirical FFs. A classical FF consists of the following principal energy terms [7]:

$$U_{FF} = U_{bonds} + U_{angle} + U_{torsion} + U_{coulomb} + U_{VanDerWaals}$$

The FF accounts for the so-called bonded and non-bonded interactions. Bonded terms include bond stretching (U_{bonds}), angle bending (U_{angle}), and torsional rotations ($U_{torsion}$), which are divided into proper and improper torsions. Non-bonded terms include electrostatic interactions ($U_{coulomb}$), usually described by Coulomb's law, and van der Waals forces ($U_{VanDerWalls}$), often modeled by Lennard-Jones potential. One has to keep in mind that despite their utility FF do not perfectly reflect reality, they can only imitate the reality for the simulation. It is impossible to reproduce the full nature of intra- and intermolecular interactions using just FF terms.

Neural Network Potentials

Molecular dynamics simulations are a trade-off between accuracy and speed. Ideally, all interactions should be described by QM, but only FFs are fast enough for adequate simulation lengths and system sizes. In 2007, Behler and Parrinello published the groundbreaking work on a high-dimensional neural network representation of potential energy surfaces (PES) [8]. A decade later, in 2017, the first generally applicable implementation of the concept was launched. ANAKIN-ME (Accurate NeurAl networK engINe for Molecular Energies), also known as ANI-1 [9], is an extension of Behler and Parrinello's approach, applicable to any organic molecules consisting of elements C, H, N and O. NNPs, such as ANI-1, promise quantum mechanical accuracy at a reasonable cost.

Since 2020, the field of NNP research has increased massively [7, 10, 11]. However, a universally applicable model for arbitrary chemical systems, in terms of composition and size, is still lacking. Existing NNPs are typically trained on single molecules or very small systems. For example, the ANI model is trained on single molecules smaller than 8 atoms [9, 12].

Two more recent NNPs, MACE-OFF [13] and Nutmeg, both used primarily the SPICE [14] dataset. Nutmeg [14] used version 2 of SPICE, whereas MACE-OFF23 is based on a subset of version 1, with some custom additions. SPICE contains compounds with less than hundred atoms per molecule, or fewer than 150 atoms for water clusters. The subset consists of the ten most common elements: H, C, N, O, F, P, S, Cl, Br, I.

Nutmeg was trained with clusters of up to 30 water molecules, among others, but the simulation of a system of 346 waters was not successful [14].

The prerequisite for simulating complex heterogeneous systems and processes, such as protein dynamics or ligand binding, is to first validate the performance of NNPs in simpler, homogeneous environments. In this context, homogeneous systems refer to systems composed of identical molecules, such as bulk water or pure organic solvents, whereas heterogeneous systems involve different molecules, such as macromolecular complexes or biological membranes.

Given that NNP training datasets are currently limited to single molecules or small molecule clusters, the remaining question is how well these models can be transferred to condensed phase simulations, where many-body interactions dominate. In order to make the next big leap in NNP simulation towards complex heterogenic systems on big scales, it is, therefore, crucial to critically evaluate the performance of NNPs in large

homogeneous systems.

Objective of this thesis

The focus of this work is on the application of NNPs for MD simulations of pure liquids, specifically water, benzene, and n-hexane, which are compared to traditional force field simulations. The employed NNPs are ANI-2x and MACE-OFF23(S). The performance is evaluated by comparing thermodynamic properties such as heat of vaporization, isothermal compressibility, heat capacity, coefficient of thermal expansion, and density to experimental data. In addition, the structural properties are analyzed using the radial distribution function (RDF) and the mean square displacement (MSD) to determine the diffusion behavior during the simulation.

2 Theory

2.1 Molecular Dynamics Simulation

Molecular dynamics simulation helps to track the movement of individual atoms in a system over time by numerically solving Newton's equation of motion $F = m \cdot a$ for N particles. It shall be noted that all variables in this section are to be regarded as vectors, and the vector arrows have been omitted for better readability. As an initial condition, random velocities are assigned to given atom positions in the system. Then, the forces acting on each particle are computed and the equation of motion is integrated to propagate the coordinates to the next time step. Time steps are usually on a femtosecond scale to cover all relevant phenomena in the system and ensure the stability of the simulation [4]. The forces F acting on each atom are derived from the system's potential energy, which can be modeled by classical FF or NNP. To illustrate the principle of such numerical integration of Newton's equation of motion, the Verlet algorithm was chosen.

The algorithm uses a Taylor expansion series to obtain numerical integration. The first derivative of the position $r(t)$ is the velocity $v(t) = \frac{dr(t)}{dt}$. The second derivative of the position $r(t)$ with respect to t is the acceleration $a(t)$, which according to Newton's laws equals $\frac{F_i(t)}{m_i}$ for all particles $i = 1, \ldots, N$.
The Taylor series is expanded both forward and backward with respect to time t.

$$r(t + \delta t) = r(t) + v(t)\delta t + \frac{1}{2}a(t)(\delta t)^2 + \frac{1}{6}b(t)(\delta t)^3 + \frac{1}{24}c(t)(\delta t)^4$$
$$r(t - \delta t) = r(t) - v(t)\delta t + \frac{1}{2}a(t)(\delta t)^2 - \frac{1}{6}b(t)(\delta t)^3 + \frac{1}{24}c(t)(\delta t)^4$$

After adding these two equations, the odd-order terms cancel out and one obtains

$$r(t + \delta t) + r(t - \delta t) = 2r(t) + a(t)(\delta t)^2 + O((\delta t)^4)$$

with $O((\delta t)^4)$ being negligible provided δt is small enough. After rearranging the equation and substituting $a(t) = \frac{F_i(t)}{m_i}$, one obtains the numerical solution

$$r(t + \delta t) = 2r(t) - r(t - \delta t) + \frac{F_i(t)}{m_i}(\delta t)^2.$$

The Verlet algorithm does not calculate the velocities explicitly, but they can be estimated at any point in time by solving

$$v(t) = \frac{r(t + \delta t) - r(t - \delta t)}{2\delta t}.$$

2.2 Neural Network Potentials

This section introduces the architectures of ANI-2x [12] and MACE [15], two NNPs which use different approaches to represent the local atomic environment and predict the total energy of a system. While ANI-2x employs a vector-based approach, MACE uses a graph-based method involving message passing. It is important to differentiate between the MACE architecture itself and networks using the MACE architecture trained for specific applications. Two important examples are MACE-OFF [13] in organic chemistry, where OFF stands for organic force fields, and MACE-MP-0 in material science, where MP stands for 'Material Project' [15]. Although MACE-OFF23 and MACE-MP-0 use the same neural network architecture, they differ in the dataset used to train the NNP. In the following, models trained with the MACE architecture are referred to as MACE-OFF potentials.

2.2.1 ANI-2x

ANI-2x is a high-dimensional NNP based on atomic environment vectors (AEV) $\vec{G}_i^X$. For each atom in a system, one AEV is constructed which holds radial and angular descriptors about the environment around the corresponding atom. Each atom is sufficiently described in its chemical environment by the numerical AEV, which are created from molecular coordinates $\vec{q} = (q_1, q_2, ..., q_x)$.

An AEV has the general form of

$$\vec{G}_i^X = \{G_1, G_2, ..., G_M\}. \tag{2.1}$$

where X represents the chemical element, the subscript i labels the atom in the molecule, and G_M represent the number of features probing the chemical environment. In 2017, the ANI-1 model could process four elements, C, H, N and O; three years later, it was extended and now also includes the atoms S, F, and Cl. In fact, the ANI-2x potential comprises seven individual NNPs with identical architecture, one for each atom type for which it was trained.

Each individual AEV is processed by the element specific NN, which transforms the descriptor into the corresponding atomic energy contribution E_i. These individual energies for each atom E_i are not meaningful on their own, they can be seen as an incomplete result, and only become relevant after summing up all contributions to yield the total energy of the system E_T.

$$E_T = \sum_{\mathrm{i}}^{\text{all atoms}} E_i \tag{2.2}$$

The total energy E_T is a representative physical quantity that characterizes the state of the system. The forces acting on each atom can be obtained as the negative gradient of the total energy with respect to the position r of atom i.

$$F_i = -\nabla_{r_i} E_T$$

In NNPs, such as ANI-2x and MACE-OFF23, the gradients are computed by backpropagation (reverse-mode automatic differentiation) within the ML framework (PyTorch) [12]. During forward propagation, the computational graph that maps the atomic coordinates to the total energy is tracked. Automatic differentiation then uses this graph for backpropagating, obtaining the derivatives (forces) of the energy with respect to all atomic positions [16].

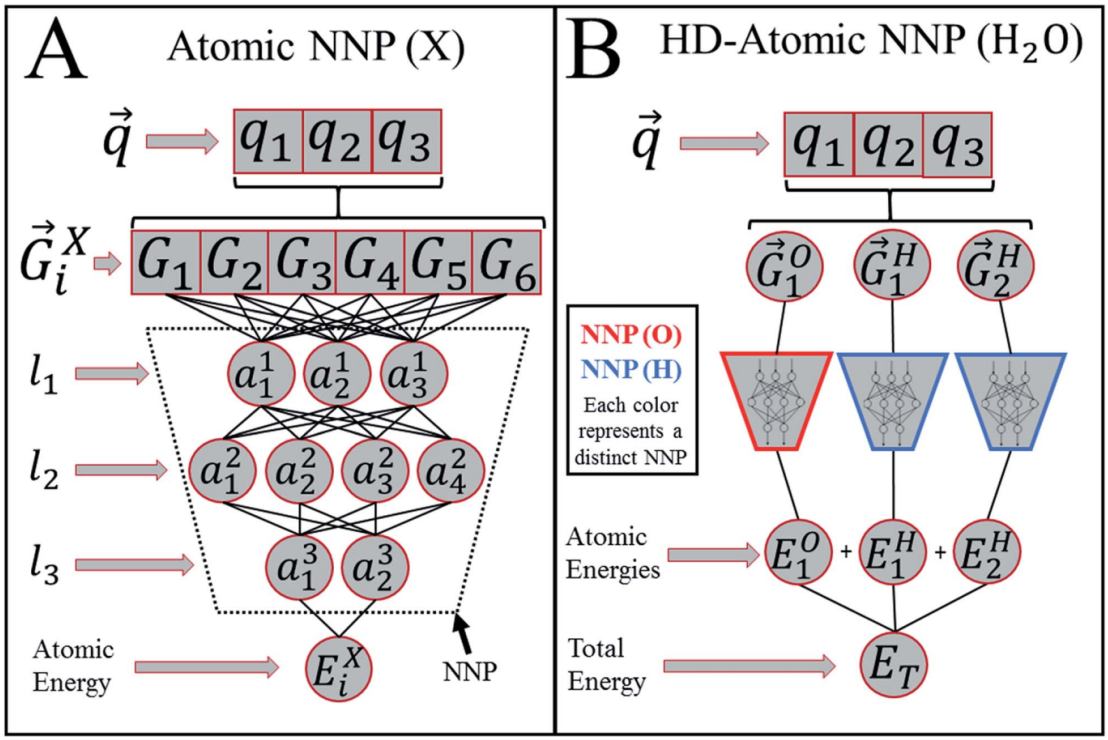


Figure 2.1: A schematic representation of the ANI-2x model, based on Behler and Parinello's work. (A) shows the general algorithmic structure of a high dimensional-NN (HD-NN). $\vec{q}$ denotes the molecular coordinates, G_i^X denotes the atomic environment vector (AEV), a_i^j represent the nodes in the three layers l_x, and E_i^X denotes the energy contribution. G_i^X is composed of $\vec{q}$ and environmental features G_m. The AEV is propagated through the atom specific NN and gives E_i^X as an output. (B) shows the application of HD-NN specific to water. Each atom type has a separate NN, each of which generating an energy contribution. Theses contributions sum up to give the total energy E_T. Figure from Smith et al [9].

The local environment

Smith et al. [9] have taken up Behler and Parinello's work [8] and further developed the symmetry function (SF) to single-atom AEV that solves the transferability problem

of the SF in complex chemical environments.

The transferability problem refers to the difficulty of applying a machine-learning potential (MLP) to structures beyond the set on which it was trained. Models usually perform better with structures similar to the training set. The reliability of these structures decreases significantly for systems outside the training database. [17]

The local environment of the atoms in the system is probed by two functions, the radial- and the angular-symmetry functions, with its predefined hyperparameters. These hyperparameters are ζ, θ_s, η, and R_s, all of which are described in more detail in the following. Although the model does not explicitly distinguish between bonded and non-bonded interactions, it implicitly learns this information from the distances and angles encoded in the AEV ad Eq. 2.1. To ensure computational efficiency, cut-off radii R_C are introduced, as it is unnecessary to compute all the system's atom-atom interactions. This cut-off function f_C guarantees that interactions are gradually reduced to zero as the interatomic distance R_{ij} approaches the cut-off distance R_C.

$$f_C(R_{ij}) = \begin{cases} 0.5 \times \cos\left(\dfrac{\pi R_{ij}}{R_C}\right) + 0.5 & \text{für } R_{ij} \leq R_C \\ 0.0 & \text{for } R_{ij} > R_C \end{cases}$$

The radial symmetry function

$$G^R_{m_a} = \sum_{i \neq j}^{\text{all atoms}} e^{-\eta(R_{ij}-R_s)^2} f_C(R_{ij})$$

encodes the distances between atom i and j with its set of hyperparameters $m_a = (\eta_a, R_{s_a})$. Here, η is constant and R_s is variable with 32 predefined parameters. R_s denotes the shifts of the center of peak. The hyperparameter η determines the width of the Gaussian distribution, while multiple R_s probe the area around the atom.

The angular symmetry function

$$G^{A_{mod}}_{m_b} = 2^{1-\zeta} \sum_{j,k \neq i}^{allatoms} (1 + \cos(\theta_{ijk} - \theta_s))^\zeta \times \exp\left[-\eta(\frac{R_{ij} + R_{ik}}{2})^2\right] f_C(R_{ij}) f_C(R_{ik})$$

encodes the angular environment between atoms i, j, and k, where θ_{ijk} denotes the angle between atoms i, j, and k. The set of hyperparameters $m_b = (\zeta_b, \theta_{s_b}, \eta_b, R_{s_b})$ includes radial and angular components.

η_b and R_s serve a similar role as in the radial SF, ζ and θ_s control the angular component. θ_s denotes the shifts in the angular environment and ζ denotes changes in the width of the peak.

η_s and ζ_s are constants, whereas R_s and θ_s vary over 8 radial and 8 angular parameters, respectively. In ANI-1, an AEV consisted of 768 input values, by extending the network by three atom types, an AEV for ANI-2x now holds 2352 input values. [9]

2.2.2 Graph neural networks and MACE

A graph neural network (GNN) is a type of neural network that is based on a graph structure and can formally be defined as $G = (V, E)$ where elements $x \epsilon V$ are called nodes and $e \epsilon E$ are called edges. Each edge can be seen as a connection between nodes, defined by a pair of nodes $\{x_1, x_2\}$. In a molecular system, atoms of a molecule can be represented as nodes which are connected with its neighboring nodes via edges. These edges are not restricted to chemical bonds or atom pairs, but formed between all atom pairs within a defined cut-off radius.

The advantage of using a GNN is the flow of information between nodes via edges, where the edges can also be used to encode pairwise distance. Message passing enables the network to iteratively update the representation of each node through its connecting neighbors. The key components of GNN are

- embedding vectors $h^{(0)}$
- messages m
- update function f_{update}

At startup, each atom is represented by a vector h_i^0 (*embedding vector*) that initializes node features, where i denotes the i^{th} atom of the respective molecule and 0 the recursive update state. The node features are iteratively refined through repeated information exchange with its neighboring nodes (see below). This vector h_i^0 is a learnable parameter, specific to each chemical element, that was determined during the NN's training phase.

Messages are encoded information flowing from one node to another. A message could be described as $m_{ji}^{(n)}$ where information is passed from node j to node i in iteration step n, this message holds information about the radial and angular environment of atom j.

Once all messages of a system are created, the embedding vector is updated by using an *update function*. Therefore all messages flowing to one node from its neighboring nodes within the cut-off radius are aggregated. An update function could, e.g., be formulated as

$$h_i^{n+1} = f_{update}(h_i^n, \sum_{j \epsilon N(i)} m_{ji}^{(n)})$$

with j being $\mathcal{N}(i) = \{j \mid r_{ij} \leq r_{\text{cut}}\}$. The nodes vector of atom i is updated using the information of this particular vector in the iteration n and all the messages m_{ij} that flow from neighboring atoms j to atom i. The update function could also be defined by a layer of a NN [18]

Figure 2.2 illustrates message passing. Assume we start in state n and collect all information within the cut-off radius for node i and the same for node j. Once nodes are updated, the system reaches state $n + 1$; here again all information within the cut-off radius is collected. After next update, node i implicitly has information about node e and its environment even if the two nodes are out of each others cut-off reach.
MACE updates its nodes three times, with the first update being the embedding. This

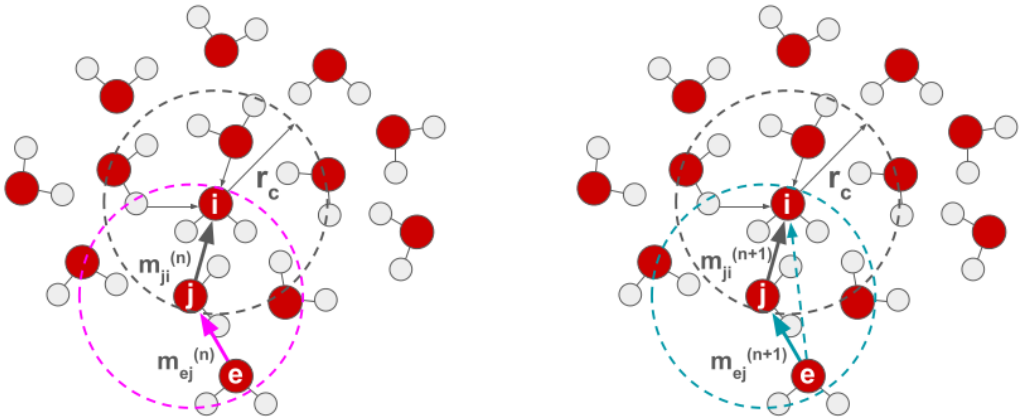


Figure 2.2: A schematic representation of message passing in a graph neural network (GNN). Left: At iteration n, atom i collects information from all atoms within the cut-off radius r_C. For example, message $m_{ji}^{(n)}$ for atom i. Right: At iteration step $n+1$, the process of collecting information is repeated. However, when creating messages for atom i, atom j has already incorporated information about atom e and passes this information on to atom i, implicitly. Figure from Picha et al. [19]

recursive update can double the effective cut-off radius, under the condition that a node lies at the boundary of the original cut-off.

The MACE architecture [20] can be trained on arbitrary datasets whereas MACE-OFF is a trained model using the MACE architecture. MACE-OFF is trained on the most common chemical elements H, C, N, O, F, P, S, Cl, Br, I. [13]

The size of the NN in MACE is predominantly dependent on the number of learnable chemical channels. These channels are treated independently and act as hyperparameters that encode the chemical environment. For example, the embedding vector of the MACE-OFF23(S) model is described by 96 channels, whereas the MACE-OFF23(M) model uses 128 channels. [20, 13] Table 2.3 summarizes the dataset used to train the MACE-OFF potential. It shows the chemical composition, system size and the number of structures used for training and testing. The columns from PubChem to Solvated Amino Acid have been transferred from the original SPICE dataset (version 1) without modifications. This subset is restricted to molecules with neutral formal charge. Besides solvated amino acids, which have larger molecules, this subset consists mainly of molecules of up to 50 atoms. Larger molecules containing up to 90 atoms were included from QMugs. To complete the newly created dataset, small water clusters, of up to 50 molecules, were added. This composition contains ten core elements H, C, N, O, F, P, S, Cl, Br, I. 95% of the samples were used to train the potential, the remaining 5% were used for testing. The dataset

was split, in a way that ensured conformers from the same structure were not present in both sets. [13]

	PubChem	DES370 K monomers	DES370 K dimers	dipeptides	solvated amino acids	water	QMugs	tripeptides
chemical elements	H, C, N, O, F, P, S, Cl, Br, I	H, C, N, O, F, P, S, Cl, Br, I	H, C, N, O, F, P, S, Cl, Br, I	H, C, N, O, S	H, C, N, O, S	H, O	H, C, N, O, F, P, S, Cl, Br, I	H, C, N, O
system size	3–50	3–22	4–34	26–60	79–96	3–150	51–90	30–69
# train	646821	16861	263065	19773	948	1597	2748	0
# test	33884	889	13896	1025	52	84	144	898

[a]The columns 'PubChem' to 'Solvated Amino Acids' correspond to the original SPICE dataset.

Figure 2.3: Composition of the MACE-OFF23 training data (Figure from Kovacs et al. [13])

The local environment The generated messages encode the local environment around each atom. The information for the messages are encoded using Bessel functions for the radial dependencies and spherical harmonics for the angular information. Bessel functions are defined as functions of radial distances

$$j_0^n(r_{ij}) = \sqrt{\frac{2}{r_{cut}}} \frac{\sin(\pi n \frac{r_{ij}}{r_{cut}})}{r_{ij}} f_{cut}(r_{ij})$$

where r_{ij} represents the distance between atoms i and j, r_{cut} the cut-off radius and n different wavenumbers of the Bessel function. [18]

Spherical harmonics are used to describe the spatial orientation of neighboring atoms relative to a central atom. Two angles define where each neighboring atoms lies on a virtual sphere around the center. The angular pattern is captured in a way that naturally respects rotational symmetry. Rotations in a atomic system are transformed in a consistent way, regardless of the orientation of the system. The combination of Bessel functions and spherical harmonics enables a representation in which we know exactly the orientation and the distance of each neighboring atom relative to the central atom. [20].

2.2.3 Training of NNP

Two essential parts are needed to train a NNP, the architecture and a training dataset [8]. The architecture of the network must be decided upon before training of a NN can begin. In other words specifying the number of layers, the sizes of embedding and bias vectors and weight matrices as well as activation functions [18]. The graph neural network builds the node vectors, those vectors are then propagated through a feed-forward neural network (FFNN) and has energies as output. A feed-forward neural network also called multilayer-perceptron, consists of multiple layers, an input layer, an output layer and one or more hidden layers [21].

Each layer is composed of interconnected perceptrons that pass information from the input to the output layer without forming cycles. Perceptrons are organised in layers. One perceptron can be described by

$$f(\vec{x}) = \sigma(A\vec{x} + \vec{b}) = \vec{x'}$$

where A denotes a matrix, $\vec{b}$ a bias vector, $\vec{x}$ the input features and σ the activation function (e.g., $\sigma(x) = \max(0, x)$). Each layer in a NN can have different activation functions, biases and weights. Initially weights are assigned random values. During training, the input features X are mapped to known output features Y using forward and backward propagation. As training progresses, the weights are adjusted to achieve better representation of the input features X. The weight adjustment is controlled by a loss function, which quantifies the difference between the predicted and the true outputs. NN's such as ANI-2x and MACE both incorporate energies and forces into their loss function. The ANI-2x loss function, for example, is expressed as follows

$$L = \frac{1}{N}\sum_{i=1}^{N}\left[(\hat{E}_i - E_i)^2 + \frac{l_0}{M_i}\sum_{j=1}^{M_i}(\hat{f_{ij}} - f_{ij})^2\right]$$

where N denotes the number of molecules, $\hat{E}_i$ and $\hat{f_{ij}}$ are the predicted energies and forces, E_i and f_{ij} are the quantum mechanical energies and forces, M denotes the number of atoms per molecule, and l_0 serves as a balancing factor between energies and forces during training. To ensure proper training on energies, the balancing factor l_0 is set to 0.1. Forces do not hold information about the molecules absolute energy; therefore, the contribution is weighted less [12, 13]. This prevents the network from overfitting of forces while neglecting accurate energy predictions. The objective is to find the best approximation for the unknown function that describes the relationship between input features X and output features Y.

A multi-layer perceptron can be written as

$$f = f_L \circ f_{L-1} \circ ... \circ f_1$$

which is equivalent to

$$f(x) = f_L(f_{L-1}(...(f_1(x))))$$

An example for a three layer NN can be expressed as

$$f(\vec{x}) = \sigma_3(A_3(\sigma_2(A_2(\sigma_1(A_1\vec{x} + \vec{b_1})) + \vec{b_2})) + \vec{b_3}) = \vec{y}.$$

A dataset is divided into a training set, which is used to optimize the weights, and a test set, which is used to evaluate the predicted outputs. In this context, both the neural networks of ANI-2x and MACE-OFF23 are trained on quantum mechanical energies as well as forces. The true outputs of the dataset are known, so the test set can be used to evaluate how well the network has learned this mapping. Quantities used for this evaluation include the standard deviation and the root mean squared error (RMSE).

2.3 Condensed Phase Properties

FF and NNP aim to reproduce real-world physical behavior in MD simulations, and should therefore calculate accurate physicochemical properties. To ensure the reliability of

such simulations, validation is crucial. This involves validating simulation results against experimentally measurable properties describing the relationships between energy, volume, temperature, and pressure in a system.

The following condensed phase properties characterize the behavior of liquids. These properties have been used over the years [22, 23] as standard criteria in the evaluation of FF development. Since they are routinely measured for industrial and chemical processes, a large amount of data is available. The development of FF or NNP benefits from the existing data.

In classical thermodynamics, the properties of interest, isothermal compressibility κ, heat capacity C_p, and the coefficient of thermal expansion α, are defined as follows:

$$\kappa = -\frac{1}{V}\frac{\partial V}{\partial P}$$

$$C_p = \frac{\partial H}{\partial T}$$

$$\alpha = \frac{1}{V}\frac{\partial V}{\partial T}$$

where V, P, T, H denote volume, pressure, temperature, enthalpy respectively, and ∂ denotes the partial derivative operator.

In the recent work of Picha et al "Condensed phase properties and transferable neural network potentials" statistical mechanical derivations on how to obtain these three properties from ensemble averages are provided [19].

In the following equations defining ΔH_{vap}, κ, C_p and α, R denotes the gas constant, T denotes the temperature, N denotes the number of particles in the respective system and k_B denotes the Boltzmann constant.

Heat of Vaporization

The Heat of vaporization describes the heat (energy) required to transfer a molecule from the liquid phase into the gas phase at constant temperature. ΔH_{vap} can be calculated as follows [23]:

$$\Delta H_{\text{vap}} = \langle E_{\text{gas}} \rangle - \langle E_{\text{liquid}} \rangle + RT \quad [\text{kJ/mol}] \tag{2.3}$$

where E_{gas} and E_{liquid} denote the mean energy of one molecule in either gas or liquid phase.

Isothermal Compressibility

The isothermal compressibility quantifies the change in volume in response to a change in pressure at a given temperature. It measures how compressible a fluid is under isothermal conditions.

$$\kappa = \frac{1}{\kappa_B T}\frac{\langle V^2 \rangle - \langle V \rangle^2}{\langle V \rangle} \quad [1/\text{bar}], \tag{2.4}$$

where $\langle V^2 \rangle$ denotes the mean squared volume, $\langle V \rangle^2$ denotes the squared mean volume, and $\langle V \rangle$ the mean volume

Heat Capacity

The heat capacity (specific heat capacity) is the amount of heat respectively energy needed to change the temperature of one gram of a material by one Kelvin (or degree Celsius) at constant pressure.

$$C_{\mathrm{p}} = \frac{1}{NRT^2}(\langle H^2\rangle - \langle H\rangle^2) \quad [\mathrm{cal/gram/K}], \tag{2.5}$$

where $\langle H^2\rangle$ denotes the mean squared enthalpy and $\langle H\rangle^2$ denotes the squared mean enthalpy.

Coefficient of thermal expansion

The coefficient of thermal expansion (also known as thermal expansion coefficient) is the amount the material or liquid expands with increasing temperature and vice versa contracts with decreasing temperature.

$$\alpha = \frac{\langle VH\rangle - \langle H\rangle\langle V\rangle}{RT^2\langle V\rangle} = \frac{Cov(VH)}{RT^2\langle V\rangle} \quad [1/\mathrm{K}], \tag{2.6}$$

where $\langle V\rangle$ denotes the mean volume and the expression $\langle VH\rangle - \langle H\rangle\langle V\rangle$ can be written as $Cov(VH)$ the covariance of the volume and the enthalpy.

2.4 Structural and dynamic properties

Diffusion

The self-diffusion coefficient (D) can be derived from the mean-squared displacement (MSD), which quantifies the drift of molecules through a system triggered by random collisions between them.

$$MSD(t) = \left\langle \frac{1}{N}\sum_{i=1}^{N} |r_i(0) - r_i(t)|^2 \right\rangle_t, \tag{2.7}$$

where r denotes the position of the particle i in Cartesian coordinates after time t.

The self-diffusion coefficient can be computed from the slope of the MSD.

$$D = \frac{1}{2d}\lim_{t\to\infty}\frac{d}{dt}MSD(t) \tag{2.8}$$

The pre-factor $\frac{1}{2d}$ depends on the dimensionality d of the system [24].

Radial Distribution Function

The radial distribution $g(r)$ function analyzes how atoms are

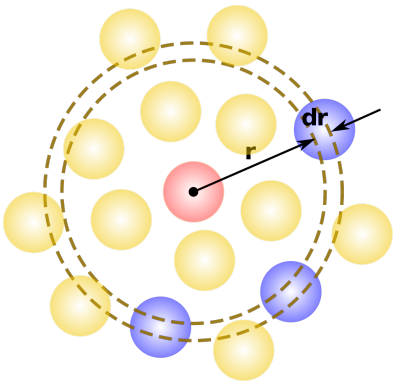


Figure 2.4: Schematic representation of RDF (Figure from Picha et al. [19])

spatially arranged relative to each other in a liquid. It is determined by measuring the distance between each pair of particles. Therefore, each atom serves as origin from which the distances to all other atoms between two concentric spheres with radius r and $r + dr$ are measured. The type of atom is also taken into account, e.g. RDF(CO) specifically accounts for the distances between atom type carbon and oxygen atoms. An average distribution of atoms is computed over all frames of a trajectory, i.e.,

$$g(r) = \frac{1}{\rho} \frac{1}{4\pi r^2 dr} \left\langle \frac{1}{N} \sum_{i=1}^{N} \sum_{j \neq i} \delta(r - r_{ij}) \right\rangle, \tag{2.9}$$

where ρ denotes the average density of the system, the volume of the difference between the spheres is approximated by $\frac{1}{4\pi r^2 dr}$, δ is the Dirac delta function given by [24]

$$\delta(x) = \begin{cases} 1, & x = 0, \\ 0, & \text{else.} \end{cases}$$

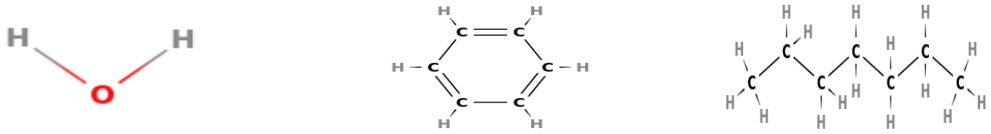
3 Methods

3.1 Simulation details

Three different molecular liquids, water, benzene and n-hexane, were chosen for the simulation and for comparison of the behavior of NNPs (see Figure 3.1c).

Water is a small, polar molecule with strong hydrogen bonds, which makes it challenging to simulate accurately. This is why evaluating the transferability of FF and NNPs for water is important. Benzene is a nonpolar molecule with an aromatic ring which shows π-π stacking interactions. In contrast to water and benzene, n-hexane is a nonpolar, aliphatic molecule. Both benzene and n-hexane are larger molecules than water and are often used as organic solvents, this makes their behavior interesting to study.

All three homogeneous systems were simulated at a temperature of 300 Kelvin.



(a) water (b) Benzene (c) n-hexane

Figure 3.1: Simulated species in homogeneous system. (a) water, (b) benzene, (c) n-hexane.

The boxes for the initial FF simulation of water, benzene and n-hexane were constructed using Charmm-GUI [25]. The number of molecules in each box determines its box length. The default cutoff for simulations with FF is 12 Å, therefore the optimal box length L should satisfy $L/2 > R_C$ [26]. If Boxes are smaller than twice the cutoff self-interactions will occur which violates the consistency of the minimum image convention. Boxes that are significantly larger than $2 * R_C$ would considerably increase computational costs. For these reasons, a box with a length of 25 Å was considered optimal. Additionally, to have comparable results, the systems should ideally consist of about the same amount of atoms. This led to different numbers molecule wise (see table 3.1).

The box sizes for the canonical ensemble (NVT) were calculated using Avogadro's constant to determine the number of molecules in units *mole*, which is required to calculate the mass m using equation $m = n * M$. The volume V is then calculated using the experimental density ρ of each system using $V = \frac{m}{\rho}$. The box length L can subsequently be derived from the volume $\sqrt[3]{V}$.

			NPT	NVT	Experiment
system	# molecules	# atoms	L box (Å)	L box (Å)	$\rho[g/mL]$
water	572	1716	25.72	25.798	0.997
benzene	130	1560	27.21	26.717	0.880
n-hexane	98	1960	27.73	27.768	0.655

Table 3.1: Simulation details of the three systems water, benzene and n-hexane. L is the length of the simulation boxes ($V = L^3$). The NPT box size is the instantaneous value after the first 10 ns of FF equilibration; the size of the NVT box is derived from the experimental density

3.2 OpenMM and Platform

The choice of ANI-2x and MACE-OFF(S) was motivated by the fact that these are fully trained and transferable models that are publicly available. In addition, efficient implementations in the OpenMM/ML software stack (version 8.1.2) were already available when this work was started. OpenMM also supports FF simulations, so all simulations could be carried out with one toolkit. MACE-OFF23 is provided in 3 different network sizes, small(S), medium(M) and large(L). In this thesis, only the MACE-OFF23-S model was used because computations with the medium network are extremely slow and would have required too much time. The large network is even slower and more memory-intensive and was, thus, unusable on the available hardware. The NN simulations were carried out on NVIDIA RTX 4090 GPUs using double precision floating point arithmetic (*float64*), while classical FF simulations were executed in OpenMM's mixed precision mode.

3.3 Integrator

Langevin Integrator Langevin dynamics is an extended integration method that controls temperature through stochastic and frictional forces. The Langevin integrator [27, 28] uses these two forces to mimic the effect of a heat bath. The random forces mimic random collisions and thereby regulate the temperature of the system. The frictional forces regulate the kinetic energy, the friction is proportional to the velocity of a particle, the faster a particle moves, the stronger the friction. These stochastic and dissipative impulses accelerate equilibration and thus improve convergence of thermodynamic averages. In order to study an isothermal-isobaric ensemble, it must be combined with a barostat to control the pressure. For this purpose the Monte Carlo barostat [29] was used in the NPT simulation. It adjusts the size of the periodic box to maintain constant pressure while keeping the same ratio of box lengths. However, this integrator is not suitable for

studying dynamic properties as it interferes with the dynamic behavior of the system.

Nose-Hoover integrator The Nose-Hoover integrator [30, 31] is a thermostat based on a variation of the verlet algorithm. It was used for simulations in the canonical ensemble (NVT). The temperature is connected to the kinetic energy of the system and is regulated by coupling it to a heat reservoir that exchanges energy with the system. Unlike the Langevin integrator, Nose-Hoover does not introduce stochastic forces but preserves the natural dynamical behavior of the system. Therefore, it is suitable for studying the dynamic properties of canonical ensembles.

3.4 Truncation of interactions

Particle-Mesh-Ewald (PME) summation is used to model the electrostatic interactions in simulation system with periodic boundary conditions [32]. PME was used for FF simulations, since neither ANI-2x nor MACE-OFF23 take long-range interactions into account. To avoid sharply truncated interactions in force field simulations, the Lennard-Jones forces are switched off between 10 and 12 Å, meaning the interactions are decreased to zero within this range.

The ANI-2X model uses a fixed predefined radial cutoff at 4.6 Å and an angular symmetry function cutoff at 3.1 Å. This decision is based on the distribution of atomic distances and the fact that the angular environment is less sampled in the ANI-1 data set [9].

The MACE-OFF23 model, on the other hand, has a fixed predefined cutoff at 4.5 Å. However, the effective range of the cutoff doubles as a result of message passing between atoms. [13].

3.5 Simulation setup

Figure 3.2 gives an overview of all simulation steps that were carried out. The wording *initial simulation* and *production* are used according to this figure. Each box represents a simulation and provides information about the simulation length and the model (FF or NN) used. The top left corner specifies the ensemble (NPT or NVT) and the integrator (LNG or NH) while the bottom left corner shows how many times the respective simulation was repeated.

The workflow starts with an initial FF equilibration of 10 ns for each system (water, benzene, n-hexane) using NPT ensemble and Langevin (LNG) integrator [27]. Based on this final configuration of this equilibration run, three further simulation branches were started. The first two branches used the NPT+LNG ensemble, while the third branch used the NVT ensemble with Nose-Hoover integrator (NH) [30, 31].

The first branch carried out one 1.1 ns initial simulation from the final configuration of the equilibration simulation using FF. Then five subsequent independent 1.1 ns production

simulations were added.

The second branch represents the simulations with both NNPs, ANI-2x and MACE-OFF23(S). For each NNP, one 1.1 ns simulation was carried out, also starting from the previous equilibration simulation. Subsequently, a further five independent 1.1 ns simulations were performed for the respective NNP.

In the third branch an additional 10 ns simulation with a the NVT ensemble was started. From its final configuration, 3.5 ns simulations were started with each studied model (FF, ANI-2x and MACE-OFF23(S)).

At the end of each simulation run, a restart file was written, which allowed the subsequent simulation to start exactly at the final configuration (box size and molecular coordinates) of the previous run, but with newly assigned random velocities to ensure statistical independence for the repeats of the production simulation. Simulation data, including simulation step, simulations time, potential energy, total energy, temperature, box volume, density and simulation speed were recorded every 100 steps. Trajectories of all atoms were output written to a file using the same interval, corresponding to one trajectory frame every 50 fs.

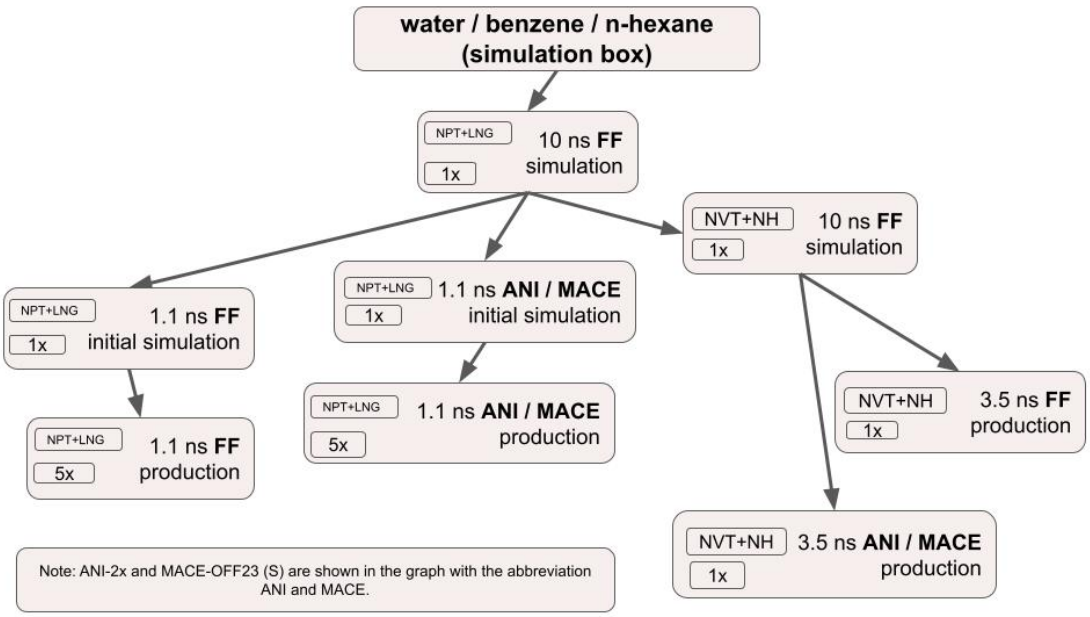


Figure 3.2: Workflow illustrating the dependency of the consecutive simulations, starting with a box from CHARMM-GUI. Langevin integrator is abbreviated to 'LNG' and Nose-Hoover integrator to NH. The thermodynamic properties and the self-diffusion coefficients were calculated from the production runs. The RDF's were calculated from the 1.1 ns simulation runs.

3.6 Data analysis

Additional simulations to the one described in the previous section 3.5 were performed in the gas phase, placing a single molecule in a box with no interactions to other molecules. The first 100 ps of every simulation were discarded as equilibrium of the simulation. The remaining 1 ns was used to calculate the condensed phase properties as described in equations 2.3, 2.4, 2.5, 2.6. The NVT ensemble simulation was used to track the self-diffusive coefficient (see table 10).

The simulation data were analyzed by calculating the statistical mean, standard deviation and relative error according to the following definitions:

Mean

$$\langle x \rangle = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{3.1}$$

Standard deviation

$$\sigma = \sqrt{\frac{1}{(N-1)} \sum_{i=1}^{N} (x_i - \langle x \rangle)^2} \tag{3.2}$$

Relative error

$$\epsilon_{rel} = \frac{\langle x \rangle - x_{ex}}{x_{ex}} \tag{3.3}$$

The variable N denotes the number of simulations runs, x_i denotes the value of a given property obtained from the i^{th} simulation run and x_{ex} denotes the corresponding experimental value. When calculating the properties, we always discarded the first 100 ps of the simulation as equilibration.

MD Analysis / NewAnalysis The trajectory files of the 3.5 ns NVT ensemble production (with box length set to the value corresponding to the experimental density) were analyzed using *MDAnalysis* [33, 34] and *NewAnalysis* [35], the latter being an in-house tool. To obtain the self-diffusive coefficient, the center of mass of each molecule in the trajectory was determined using *NewAnalysis*. Based on these positions, the MSD is calculated according to equation 2.7. Lastly, the self-diffusive coefficient is obtained from the MSD according to equation 2.8.
The *MDAnalysis* package was also used to compute the RDF according to equation 2.9. The RDFs were calculated from the 1.1 ns NPT simulation. This was done for all possible atom pairs (H-O, O-O, H-H, C-C, C-H) in each of the three systems water, benzene and n-hexane.

4 Results

This chapter presents the results of simulations of water, benzene, and n-hexane. To ensure clarity, all plots use the same color coding. FF is shown in red, ANI-2x in green, MACE-OFF23(S) in orange, the experimental data points are depicted in black, and the results of the initial simulation (Figure 3.2), if included, are depicted in grey.

Monitoring convergence Before exploring the results in greater detail, we studied the time that the thermodynamic properties needed to converge. Figures 4.1, 4.2, 4.3 show the convergence of all properties studied for water, benzene and n-hexane, respectively. Data were taken from the 1 ns initial simulation, and are plotted as a cumulative average over time (the first data point denotes the mean of the first 100 ps etc.). Each plot shows all five thermodynamic properties, from top to bottom: heat of vaporization $\triangle H_{vap}$, isothermal compressibility κ, heat capacity C_p, coefficient of thermal expansion α, and density ρ.

Water Figure 4.1 shows that $\triangle H_{vap}$ quickly converges within the first 200 ps for all three models. When using the ANI-2x model, all other quantities, i.e., κ, C_p, α, and ρ, converge slowly and, in fact, seem not to have reached convergence at the end of the 1 ns simulation time. κ, C_p, and ρ show a descending trend, while α shows an ascending pattern. These continuous drifts suggest that the ANI-2x model is unlikely to have converged. In contrast, the FF and MACE-OFF23(S) models display a much more rapid convergence for all five properties throughout the simulation. A closer observation of κ and α of the MACE-OFF23(S) model reveals a minimal decline at the beginning that appears to converge after 300 ps.

Benzene Figure 4.2 displays a very stable behavior for $\triangle H_{vap}$ and ρ for all three models. Within the first 400 ps, a slight ascending trend is noticeable, and no significant changes occur later. After 1 ns of simulation time, $\triangle H_{vap}$ and ρ clearly seem to have converged.

C_p shows a decreasing curve throughout the entire simulation period in all three models. Although the curves flattens out more and more towards the end, a downward trend is still noticeable. A similar behavior is seen for κ using the FF model. ANI-2x and MACE-OFF23(S) decrease only gradually after the initial drop and converge to similar values as FF.

α shows an interesting difference between models. While ANI-2x and MACE-OFF23(S) decrease at first, FF has an increasing trend, towards a similar value as ANI-2x, whereas MACE-OFF23(S) remains slightly higher than the other two.

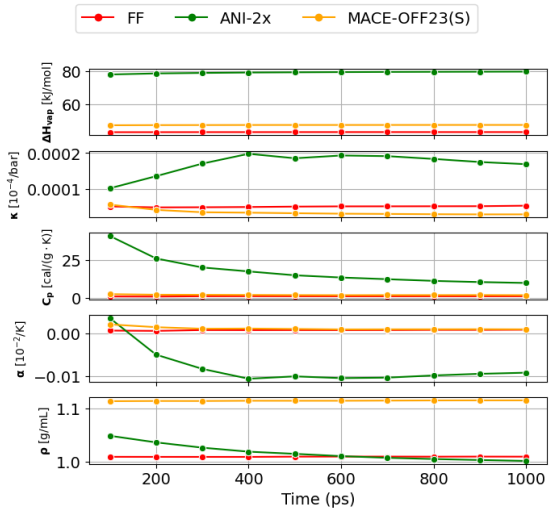


Figure 4.1: Monitoring the cumulative convergence of **water** properties studied during the 1 ns initial simulation. From top to bottom heat of vaporization $\triangle H_{vap}$, isothermal compressibility κ, heat capacity C_p, coefficient of thermal expansion α, and density ρ.

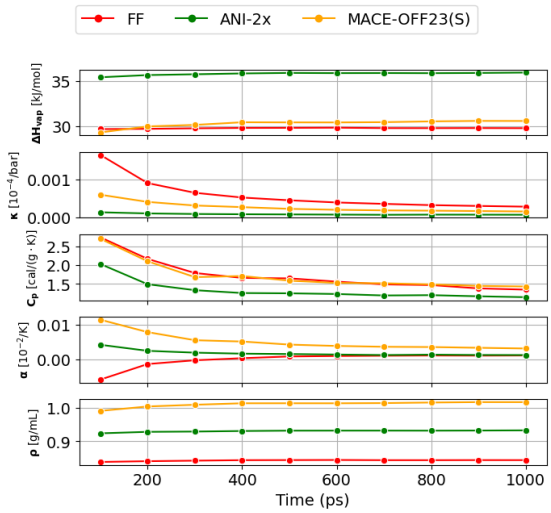


Figure 4.2: Monitoring the cumulative convergence of **benzene** properties studied during the 1 ns initial simulation. From top to bottom heat of vaporization $\triangle H_{vap}$, isothermal compressibility κ, heat capacity C_p, coefficient of thermal expansion α, and density ρ.

n-Hexane Figure 4.3 shows that for $\triangle H_{vap}$ and ρ the models ANI-2x and MACE-OFF23(S) take about half the simulation time to converge. κ and C_p have a decreasing curve for all three models throughout the entire monitoring period. The curve flattens out towards the end, but only FF and ANI-2x seem to have converged at the end of the simulation. MACE-OFF23(S) still has a noticeable decrease. For α, FF shows an increasing curve during the first 400 ps. In contrast, ANI-2x and MACE-OFF23(S) decrease during the same time, before all approach stable values. All three models seem converged after 1 ns of simulation.

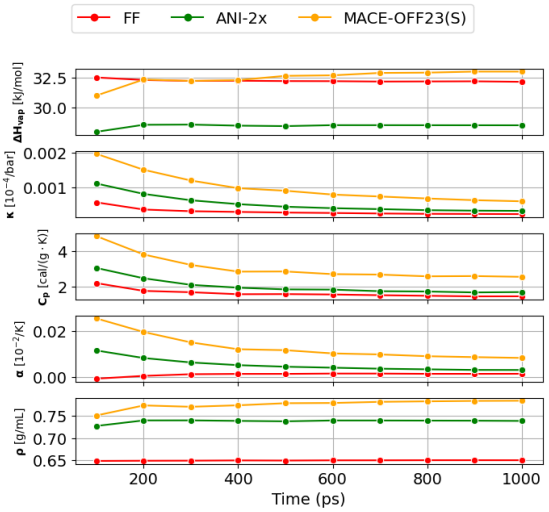


Figure 4.3: Monitoring the cumulative convergence of **n-hexane** properties studied during the 1 ns initial simulation. From top to bottom heat of vaporization $\triangle H_{vap}$, isothermal compressibility κ, heat capacity C_p, coefficient of thermal expansion α, and density ρ.

The accuracy of results with respect to the experimental values is discussed further in section 4.1 Seeing that not all models converge after 1 ns simulations time led to the decision to append an additional 1 ns simulation time in order to achieve an equilibrium. Five production runs were started using the final configuration of the initial simulation. These results are discussed in the following section.

4.1 Thermodynamic properties

Figures 4.4, 4.5, 4.6 show the results for the thermodynamic properties heat of vaporization $\triangle H_{vap}$, isothermal compressibility κ, heat capacity C_p, coefficient of thermal expansion α, and density ρ. The raw data used to create the figures are listed in tables 1, 3, 4, 6, 7, 9 in the appendix.

The symbols in the figures represent different data sources and levels of sampling. The crosses ($\times$) correspond to values obtained from the initial 1 ns simulation, dots represent

the mean values from five independent production runs with the standard deviation shown as whiskers. In cases like $\triangle H_{vap}$ and ρ the variability is minimal and, therefore, the whiskers are barely visible. Experimental values [36, 37] are shown as black dashed line for better comparison.

Water Examining the results for water, see figure 4.4, the largest deviation between the initial 1 ns simulation and the five production runs is observed for κ, C_p and α, using ANI-2x. In contrast, both FF and MACE-OFF show only minor differences between the single initial and the later five production runs for all properties. This suggests that these models produce more stable and converged estimates for water even with shorter sampling times.

All properties which showed lack of convergence, like κ, C_p and α, in figure 4.1, also have significant discrepancies between initial and production runs (figure 4.4). For properties such as $\triangle H_{vap}$ and ρ where the values converge well, no substantial differences are observed.

None of the three examined models exactly match the experimental data. However, FF is the closest with little to no standard deviation. MACE-OFF23(S) has a very small standard deviation across all properties as well, but it greatly overestimated the values of ρ and C_p and underestimates the value of κ with respect to the experiment. ANI-2x, on the other hand, has large standard deviations and poorly reproduces κ and C_p, respectively.

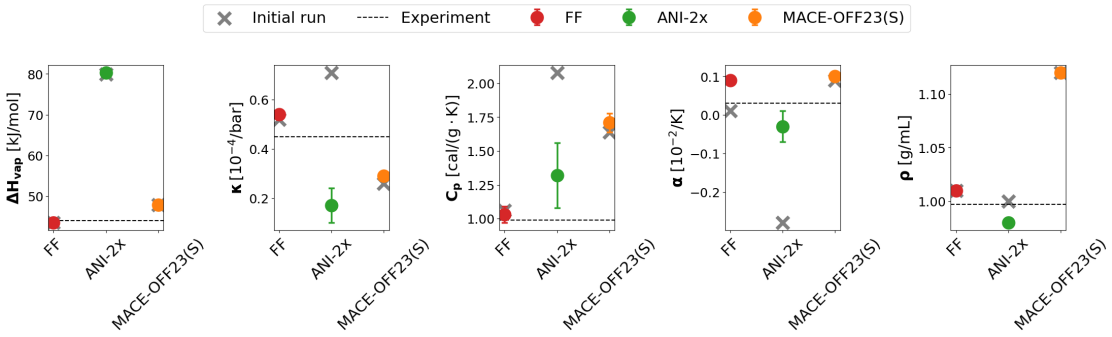


Figure 4.4: Averaged result with error bars for the condensed phase properties of **water**. FF is shown in red, ANI-2x in green, MACE-OFF23(S) in orange, the experimental data points are depicted in black and the results of the initial simulation if included is depicted with a grey $\times$. Each dot with whiskers denotes the mean and standard deviation of five independent 1 ns NPT simulations.

Benzene What stands out for the benzene system (see figure 4.5) is that the values from the initial simulation agree very well with those from the production runs for all properties, except α. For α, the initial values are consistently higher than the production values.

However, none of the models shows good agreement with the experimental data. All properties are either significantly over or underestimated. Only $\triangle H_{vap}$ predicted by ANI-2x is close to the experimental reference.

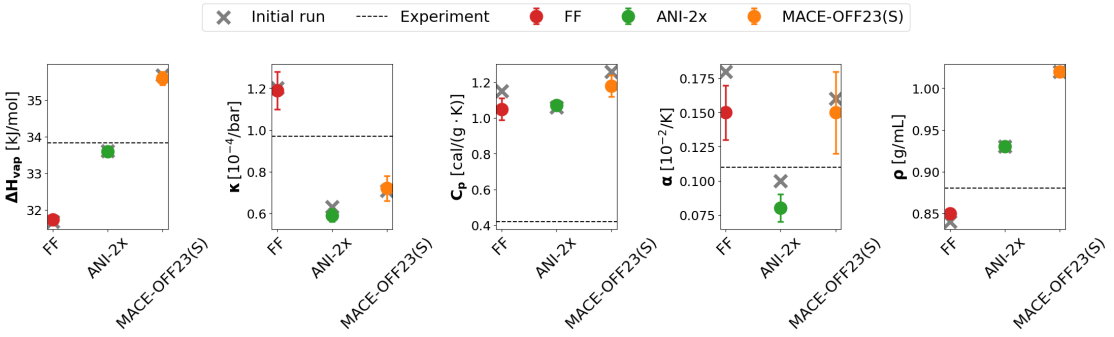


Figure 4.5: Averaged result with error bars for the condensed phase properties of **benzene**. FF is shown in red, ANI-2x in green, MACE-OFF23(S) in orange, the experimental data points are depicted in black and the results of the initial simulation if included is depicted with a grey $\times$. Each dot with whiskers denotes the mean and standard deviation of five independent 1 ns NPT simulations.

n-Hexane For n-hexane (see figure 4.6) the values from initial and production runs show good agreement across nearly all properties and systems. However, noticeable discrepancies are observed for C_p in FF, as well as for $\triangle H_{vap}$ and κ in MACE-OFF23(S). In contrast, ANI-2x demonstrates consistency between the initial and production simulations for all properties examined. Regarding the standard deviation, only MACE-OFF23(S)'s properties κ, C_p, α show large whiskers. FF and ANI-2x on the other hand, hardly show any.

FF has the best agreement with experimental data, $\triangle H_{vap}$, κ, α, and ρ are reproduced with good accuracy. ANI-2x only shows agreement for κ, α, while $\triangle H_{vap}$, C_p, and ρ strongly deviated from the experiment. MACE-OFF23(S) does not show agreement at all, with $\triangle H_{vap}$ being the closest to the experiment.

Besides these observations note, that the $\triangle H_{vap}$ has always little to no standard deviation in all three systems and also always agrees with the initial simulation. The reason is that $\triangle H_{vap}$ is the mean energy averaged over all molecules in the system.

4.2 Diffusion

Figure 4.7 presents the logarithm of the self-diffusion coefficients of water, benzene and n-hexane ($\log(D)$). The corresponding raw data is listed in table 10 in the appendix.

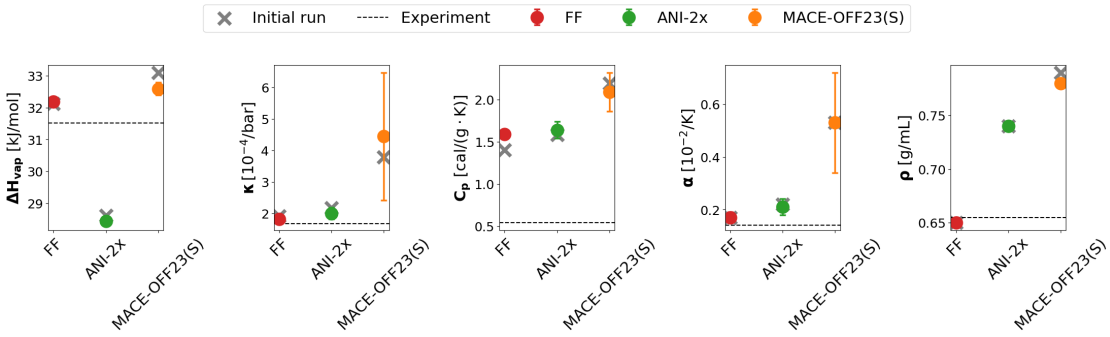


Figure 4.6: Averaged result with error bars for the condensed phase properties of **n-hexane**. FF is shown in red, ANI-2x in green, MACE-OFF23(S) in orange, the experimental data points are depicted in black and the results of the initial simulation if included is depicted with a grey ×. Each dot with whiskers denotes the mean and standard deviation of five independent 1 *ns* NPT simulations.

The classical FF has the best overall agreement with experimental data across water, benzene and n-hexane. Notably, MACE-OFF23(S) closely reproduces the experimental diffusion coefficient for water, outperforming the FF in this specific case. While the FF slightly overestimates the diffusion of water, this behavior is well documented and within the expected limits [38]. In contrast, ANI-2x significantly underestimates the self-diffusion coefficient for water, with a deviation of almost three orders of magnitude. ANI-2x and MACE-OFF23(S) follow the same trend when comparing the self-diffusion coefficient of benzene and n-hexane. Both underestimate the diffusion by one and two orders of magnitude, respectively. In comparison, FF performs very well for both benzene and n-hexane, showing excellent agreement with experimental data.

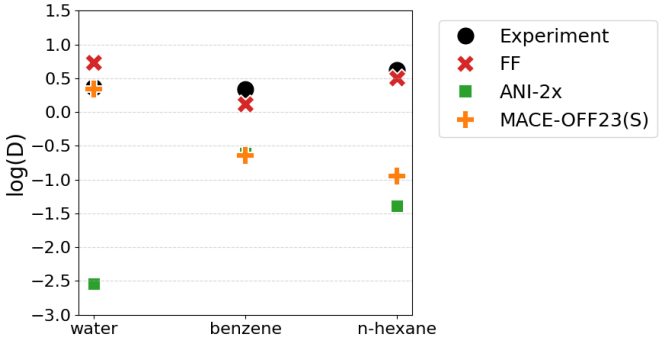


Figure 4.7: Logarithmic self-diffusion coefficient of all three systems, water, benzene and n-hexane, from the 3.5 *ns* NVT simulation run. FF is shown in red, ANI-2X in green, MACE-OFF23(S) in orange. The experimental value were obtained from [39] and are depicted in black.

4.3 Radial distribution function

Figures 4.8, 4.9 and 4.10 show the RDF (see Eq. 2.9) of each molecules' atom pair combinations. The color coding used here is the same as throughout chapter 4. Experimentally determined RDFs are available only for water[40]. Nevertheless, it seems interesting to compare the RDFS for the two organic liquids obtained by FF, ANI-2x, and MACE, respectively.

Water Figure 4.8 shows the atom pair RDFs of water (hydrogen-hydrogen, oxygen-hydrogen, oxygen-oxygen). The most striking feature is that RDFs obtained with ANI-2x are significantly more compact in structure than FF and MACE-OFF23(S). The RDFs of hydrogen-hydrogen and oxygen-oxygen clearly show that the maxima and minima of ANI-2x are much more pronounced than the experiment. In the oxygen-hydrogen RDF, this overpronunciation is only visible for the first maximum. FF and MACE-OFF23(S) are closer to the experiment [40] in all three figures. However, FF and MACE-OFF23(S) RDFs are closer to each other than the respective model is to the experiment, in particular in the hydrogen-hydrogen and oxygen-oxygen plots. In the oxygen-hydrogen plot FF aligns closely with the experiment, although MACE-OFF23(S) is also a good fit.

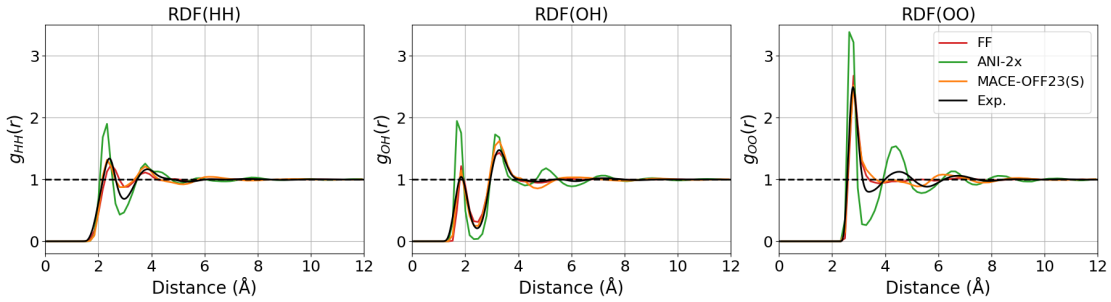


Figure 4.8: RDF of possible atom pairs in water are shown from left to right: RDF(hydrogen-hydrogen), RDF(oxygen-hydrogen), RDF(oxygen-oxygen); The RDFs were computed from the first of five 1 *ns* repeats of the NPT trajectory of the production simulation.

Benzene Figure 4.9 shows the atom pair RDFs of benzene (hydrogen-hydrogen, carbon-hydrogen, carbon-carbon). Due to the absence of experimental data, only differences between models can be characterized.

Neither the hydrogen-hydrogen nor the carbon-hydrogen plots have clearly visible maxima or minima for FF and ANI-2x. MACE-OFF23(S) on the other hand has clearly distinguishable first peaks in both plots, which are both below but near unity. This first maximum is nearly completely absent in FF and ANI-2x in both plots. Interestingly, in the hydrogen-hydrogen plot all three model align nearly perfectly till 12 Å, starting from the first minimum of MACE-OFF23(S). The carbon-carbon RDF is the most structured one of all plots. ANI-2x and MACE-OFF23(S) show a small maximum in the beginning

which gets more pronounced for the second and third maximum (same for the minima). FF does not have a clear first minimum and the second and third maximum are only separated through a small local minimum. According to the carbon-carbon RDFs, all three models start to align at the third minimum at about 7 Å.

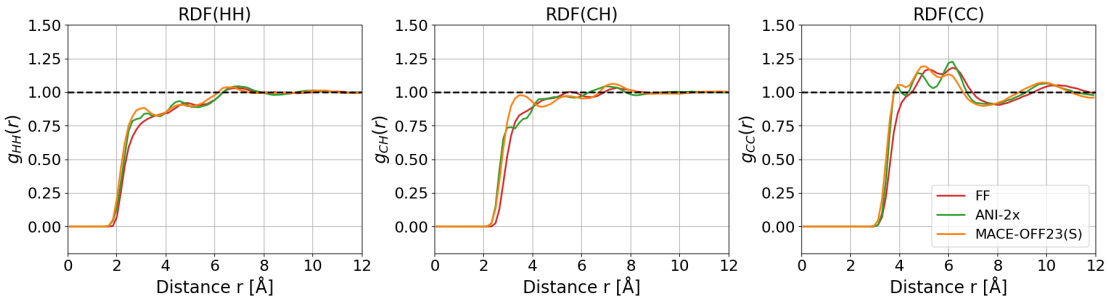


Figure 4.9: RDF of possible atom pairs in benzene are shown from left to right: RDF(hydrogen-hydrogen), RDF(carbon-hydrogen), RDF(carbon-carbon); The RDFs were computed from the first of five 1 ns repeats of the NPT trajectory of the production simulation.

n-Hexane Figure 4.10 shows the atom pair RDFs of n-hexane (hydrogen-hydrogen, carbon-hydrogen, carbon-carbon). As for benzene, we can only compare the RDFs between the models. For the hydrogen-hydrogen plot, ANI-2x and MACE-OFF23(S) show good accordance over the whole distance. Whereas these two have clear maxima, whereas the FF RDF has no visible maximum at the distance 3 Å. In the carbon-hydrogen plot, MACE-OFF23(S) has a peak at around 3 Å, whereas FF and ANI-2x only have shoulders. In the carbon-carbon plot, all three models have a noticeable first peak. Nevertheless, the distance of the maximum is at a clearly greater distance of around 5 Å. Furthermore, the carbon-carbon plot displays an oscillation around unity, indicating a more distinctive structure than can be observed in the other two discussed plots.

4.4 Performance

Table 4.1 lists the computational cost of the three systems water, benzene and n-hexane, simulated with the three models FF, ANI-2x and MACE-OFF23(S). The simulations for FF were carried out in OpenMM's mixed precision mode, for ANI-2x and MACE-OFF23(S) they were carried out on NVIDIA RTX 4090 GPUs using double precision floating point arithmetic (float64). The numbers were taken from the last 5 ps of simulations, as there was no further change in speed at this point. Comparing the costs of systems for each model, one sees that within the same model the costs are always in the same range. Nevertheless, the slight cost differences are due to the number of atoms in each box. Benzene has the fewest atoms and the lowest costs followed by water and then n-hexane, which has the most atoms and highest costs.

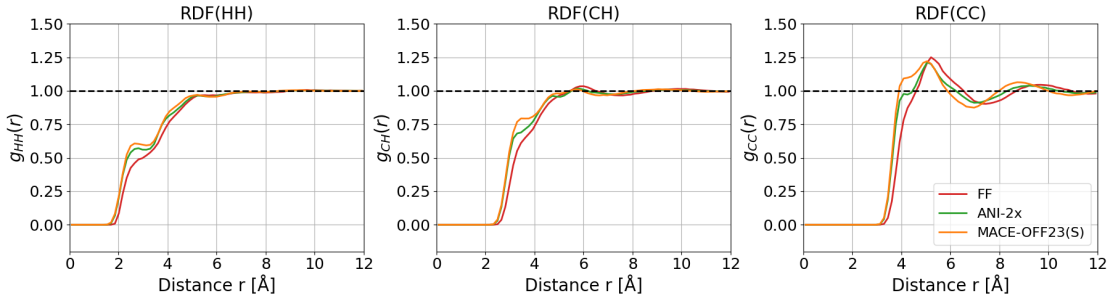


Figure 4.10: RDF of possible atom pairs in n-hexane are shown from left to right: RDF(hydrogen-hydrogen), RDF(carbon-hydrogen), RDF(carbon-carbon); The RDFs were computed from the first of five 1 *ns* repeats of the NPT trajectory of the production simulation.

	water	benzene	n-hexane
N molecules	572	130	98
N atoms	1716	1560	1960
NPT + L			
FF	38.7	38.9	32.6
ANI-2x	1.34	1.61	1.10
MACE-OFF23(S)	0.19	0.241	0.185
NVT + NH			
FF	23.8	24.2	19.9
ANI-2x	0.738	0.85	0.576
MACE-OFF23(S)	0.11	0.124	0.0955

Table 4.1: The computational costs of the simulations in [ns/day]. NPT simulations were carried out using a Langevin integrator, the NVT simulations with a Nose-Hoover thermostat. N denotes the number of molecules/atoms in each system. FF simulations were executed in OpenMM's mixed precision mode, ANI-2x and MACE-OFF23(S) on NVIDIA RTX 4090 GPUs using double precision floating point arithmetic (float64)

Much bigger differences are noticeable when comparing models with each other. Here it is only useful to compare ANI-2x with MACE-OFF23(S), because FF simulations were very fast and there would have been no need to run them on a GPU. One clearly sees that MACE-OFF23(S) has the higher costs and, therefore, took longer than ANI-2x. Using hexane as example in a 1.1 ns NPT ensemble simulation, ANI-2x ran 24 hours, while the simulation with MACE-OFF23(S) took 142 hours, i.e., almost 6 days. Comparing the simulations using Nose-Hoover with the ones using Langevin, one sees that Nose-Hoover takes twice the time compared to Langevin, indicating that the Nose-Hoover integrator is much less efficient. We note that the overhead of Nose-Hoover compared to Langevin dynamics is present also in the FF calculations, i.e., it has nothing to do with the use of an NNP. The longest simulation in this thesis was conducted with NVT+NH ensemble for n-hexane and ran for 880 hours, which is equivalent to approximately 36 days. Table 4.1 proves that the computational costs are dependent on the model and the integrator, but not so much on the molecule.

4.5 Discussion of Results

The results show that the investigated models do not only differ in the values of thermodynamic properties but also strongly in their convergence behavior. Some models (and systems) reach stable averages within the initial simulation, while others converge slowly or not at all in the respective time. This leads to significant discrepancies between initial and production simulations, as well as statistical inaccuracy. The differences in the required equilibration length can be directly linked to the self-diffusion coefficients.

In particular, according to figure 4.1, the simulation of water using the ANI-2x model does not show convergence. The properties obtained from the initial and production runs differ substantially and the five repetitions of the production run exhibit a large standard deviation (see figure 4.4). By contrast, the water simulations executed with FF and MACE-OFF23(S) show good convergence. The self-diffusive coefficient as depicted in figure 4.7 further supports these observations. For ANI-2x the value is approximately three orders of magnitude slower than the experiment and also the ones obtained from the FF and MACE-OFF23(S) simulations. Comparing the investigated NNPs, the MACE-OFF23(S) model produces more stable and converged results, whereas ANI-2x needs a longer equilibration time.

Similar patterns as for water can also be observed with n-hexane. Here, the simulations with MACE-OFF23(S) show large standard deviations for the properties κ, C_p and α (see fig. 4.6). In addition, one sees in figure 4.3 that these properties do not converge, after 1 ns they still show a descending trend. On the other hand, the simulations with FF and ANI-2x show good convergence and therefore also a very good agreement of the initial and production runs, with very little standard deviation. This is the case even though the self-diffusive coefficient is 1.5 orders of magnitude slower for MACE-OFF23(S) and about two orders of magnitude slower for ANI-2x compared to the experiment.

Benzene is the most challenging system in this study. Even though figure 4.2 does not show convergence for all three models for properties, such as C_p, figure 4.5 shows

good agreement between initial and production run as well as only a small standard deviation. Looking again at figure 4.2, α seems to have converged, but figure 4.5 shows larger standard deviation and the agreement between initial and production run is not good. The self-diffusive coefficient is one order of magnitude slower for both ANI-2x and MACE-OFF23(S) compared to FF and the experimental value.

Finally, the densities of the systems are of interest. The FF model shows the best agreement, with a deviation less than 3% compared to the experiment. Adequate behavior is observed for ANI-2x, with n-hexane being a notable exception, where the density is overestimated by about 13%. MACE-OFF23(S), on the other hand, predicts densities that are too high for all the examined systems. The densities for water, benzene and n-hexane are overestimated by 12%, 16% and 18%, respectively.

The observations between convergence behavior and consistency across simulation lengths highlights the importance of adequate equilibration time. The self-diffusive coefficient on the other hand can be a useful indicator of whether a system can reach equilibrium in a reasonable amount of time. Slow diffusion leads to longer simulation times, which can can be unacceptable for most applications. In case of ANI-2x water, the equilibration time would be three orders of magnitude longer.

5 Conclusion

The motivation behind this work was to assess whether NNPs can replace classical FF in terms of accuracy and computational efficiency. The examined NNPs ANI-2x and MACE-OFF23(S) promise quantum mechanical accuracy at reduced computational costs. Given that these NNPs were trained on small molecules and small clusters only, the research question to be addressed was whether these models are transferable to larger homogeneous systems. As anticipated, we identified several limitations. Our results show that transferability remains a major challenge as the models struggle to reproduce thermodynamic or structural properties across examined systems. The performance of the investigated NNPs in bulk systems does not yet satisfy the requirements for larger heterogeneous simulations.

It is important to emphasize that the training datasets of the ANI-2x and MACE-OFF23(S) models differ substantially. The ANI-2x model was trained using only extremely small water clusters, whereas MACE-OFF23(S) already includes clusters of up to 50 water molecules. These differences likely contribute to the poor performance of ANI-2x in water simulations. Due to these differences, it is impossible to conclude which NNP architecture is significantly better than the other. A fair comparison would require training of the two architectures using the same dataset. The slow convergence of ANI-2x and partially MACE-OFF23(S) limits their practical applicability. Additionally, MACE-OFF23(S) has limitations in predicting the density, as it consistently overestimates it by at least 12%. Ultimately, the high computational costs of both NNP models are also a significant obstacle to simulating larger heterogeneous systems like proteins in aqueous solutions.

Reaching the long-term goal of simulating large heterogeneous systems with NNPs at quantum mechanical accuracy and reasonable computational cost, therefore, seems still a long way down the road. Further progress is needed in several areas, most importantly in the complexity and variations of training datasets. Instead of only including small clusters of the same species, the datasets should be expanded with bulk properties, solvated complexes and conformations of molecules not being in the global (or a local) energetic minimum. Advances in computational hardware will make it easier to deal with larger datasets at reasonable costs. The major challenge that remains is still the creation of datasets that can accurately represent the structural and thermodynamic properties of condensed phase systems. A combination could hopefully improve the transferability of NNPs.

ANI-2x and MACE-OFF23 are so-called second generation NNPs [41] which only consider short-range interactions. Newer generations of NNPs will include long-range interaction such as electrostatics and dispersion which could improve the performance in condensed-phase systems. During FF development, the calculation of condensed-phase properties is a standard procedure. By contrast, currently, NNPs are predominantly

validated against energies and forces, which, however, is insufficient. The need to incorporate metrics based on the simulations, such as the condensed-phase properties, is only slowly being recognized [13, 19, 42]. Comparison to condensed-phase properties helps to assess whether simulations using NNPs can accurately reproduce physical behavior. Currently, it is not possible to incorporate it directly into the training process and it remains a separate, later validation step. While this still remains an open challenge, given the speed at which NNP development is progressing, it is possible that within a few years there will be NNPs that can correctly reproduce simulations with suitable thermodynamic properties.

Bibliography

[1] M. Vogelsberger, F. Marinacci, P. Torrey, and E. Puchwein, "Cosmological simulations of galaxy formation," *Nature Reviews Physics*, vol. 2, no. 1, pp. 42–66, 2020.

[2] T. Höhne, "Cfd simulation of a heat pipe using the homogeneous model," *International Journal of Thermofluids*, vol. 15, p. 100163, 2022.

[3] N. Schaeffer, D. Jault, H.-C. Nataf, and A. Fournier, "Turbulent geodynamo simulations: a leap towards earth's core," *Geophysical Journal International*, vol. 211, no. 1, p. 1–29, 2017.

[4] S. A. Hollingsworth and R. O. Dror, "Molecular dynamics simulation for all," *Neuron*, vol. 99, no. 6, pp. 1129–1143, 2018.

[5] R. O. Dror, R. M. Dirks, J. Grossman, H. Xu, and D. E. Shaw, "Biomolecular simulation: A computational microscope for molecular biology," *Annual Review of Biophysics*, vol. 41, no. Volume 41, 2012, pp. 429–452, 2012.

[6] S. Lifson and A. Warshel, "Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules," *The Journal of Chemical Physics*, vol. 49, no. 11, pp. 5116–5129, 1968.

[7] Y. Wang, K. Takaba, M. S. Chen, M. Wieder, Y. Xu, T. Zhu, J. Z. H. Zhang, A. Nagle, K. Yu, X. Wang, D. J. Cole, J. A. Rackers, K. Cho, J. G. Greener, P. Eastman, S. Martiniani, and M. E. Tuckerman, "On the design space between molecular mechanics and machine learning force fields," *Applied Physics Reviews*, vol. 12, p. 021304, 04 2025.

[8] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Phys. Rev. Lett.*, vol. 98, p. 146401, 2007.

[9] J. S. Smith, O. Isayev, and A. E. Roitberg, "Ani-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chemical Science*, vol. 8, no. 4, pp. 3192–3203, 2017.

[10] D. M. Anstine and O. Isayev, "Machine learning interatomic potentials and long-range physics," *The Journal of Physical Chemistry A*, vol. 127, no. 11, pp. 2417–2431, 2023. PMID: 36802360.

[11] R. Martin-Barrios, E. Navas-Conyedo, X. Zhang, Y. Chen, and J. Gulín-González, "An overview about neural networks potentials in molecular dynamics simulation," *International Journal of Quantum Chemistry*, vol. 124, no. 11, p. e27389, 2024.

[12] C. Devereux, J. S. Smith, K. K. Huddleston, K. Barros, R. Zubatyuk, O. Isayev, and A. E. Roitberg, "Extending the applicability of the ani deep learning molecular potential to sulfur and halogens," *Journal of Chemical Theory and Computation*, vol. 16, no. 7, pp. 4192–4202, 2020.

[13] D. P. Kovács, J. H. Moore, N. J. Browning, I. Batatia, J. T. Horton, Y. Pu, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole, and G. Csányi, "Mace-off: Short-range transferable machine learning force fields for organic molecules," *Journal of the American Chemical Society*, vol. 147, no. 21, pp. 17598–17611, 2025. PMID: 40387214.

[14] P. Eastman, B. P. Pritchard, J. D. Chodera, and T. E. Markland, "Nutmeg and spice: Models and data for biomolecular machine learning," *Journal of Chemical Theory and Computation*, vol. 20, no. 19, pp. 8583–8593, 2024. PMID: 39318326.

[15] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin, F. Berger, N. Bernstein, A. Bhowmik, F. Bigi, S. M. Blau, V. Cărare, M. Ceriotti, S. Chong, J. P. Darby, S. De, F. Della Pia, V. L. Deringer, R. Elijošius, Z. El-Machachi, E. Fako, F. Falcioni, A. C. Ferrari, J. L. A. Gardner, M. J. Gawkowski, A. Genreith-Schriever, J. George, R. E. A. Goodall, J. Grandel, C. P. Grey, P. Grigorev, S. Han, W. Handley, H. H. Heenen, K. Hermansson, C. H. Ho, S. Hofmann, C. Holm, J. Jaafar, K. S. Jakob, H. Jung, V. Kapil, A. D. Kaplan, N. Karimitari, J. R. Kermode, P. Kourtis, N. Kroupa, J. Kullgren, M. C. Kuner, D. Kuryla, G. Liepuoniute, C. Lin, J. T. Margraf, I.-B. Magdău, A. Michaelides, J. H. Moore, A. A. Naik, S. P. Niblett, S. W. Norwood, N. O'Neill, C. Ortner, K. A. Persson, K. Reuter, A. S. Rosen, L. A. M. Rosset, L. L. Schaaf, C. Schran, B. X. Shi, E. Sivonxay, T. K. Stenczel, C. Sutton, V. Svahn, T. D. Swinburne, J. Tilly, C. van der Oord, S. Vargas, E. Varga-Umbrich, T. Vegge, M. Vondrák, Y. Wang, W. C. Witt, T. Wolf, F. Zills, and G. Csányi, "A foundation model for atomistic materials chemistry," *The Journal of Chemical Physics*, vol. 163, p. 184110, 11 2025.

[16] J. S. Smith, N. Lubbers, A. P. Thompson, and K. Barros, "Simple and efficient algorithms for training machine learning potentials to force data," *arXiv preprint arXiv:2006.05475*, 2020.

[17] A. K. A. Kandy, K. Rossi, A. Raulin-Foissac, G. Laurens, and J. Lam, "Comparing transferability in neural network approaches and linear models for machine-learning interaction potentials," *Phys. Rev. B*, vol. 107, May 2023.

[18] J. Gasteiger, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," *arXiv preprint arXiv:2003.03123*, 2022.

[19] A. K. Picha, M. Wieder, and S. Boresch, "Transferable neural network potentials and condensed phase properties," *Journal of Chemical Information and Modeling*, vol. 65, no. 18, pp. 9483–9496, 2025. PMID: 40935126.

[20] D. P. Kovács, I. Batatia, E. S. Arany, and G. Csányi, "Evaluation of the mace force field architecture: From medicinal chemistry to materials science," *The Journal of Chemical Physics*, vol. 159, no. 4, 2023.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[22] W. L. Jorgensen, "Convergence of monte carlo simulations of liquid water in the npt ensemble," *Chemical Physics Letters*, vol. 92, no. 4, pp. 405–410, 1982.

[23] W. L. Jorgensen, J. D. Madura, and C. J. Swenson, "Optimized intermolecular potential functions for liquid hydrocarbons," *Journal of the American Chemical Society*, vol. 106, no. 22, pp. 6638–6646, 1984.

[24] M. E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation*. Oxford University Press, 2023.

[25] S. Jo, T. Kim, V. G. Iyer, and W. Im, "Charmm-gui: A web-based graphical user interface for charmm," *Journal of Computational Chemistry*, vol. 29, no. 11, pp. 1859–1865, 2008.

[26] M. Allen and D. Tildesley, *Computer simulation of liquids: Second edition*. 11 2017.

[27] D. S. Lemons and A. Gythiel, "Paul langevin's 1908 paper "on the theory of brownian motion" ["sur la théorie du mouvement brownien," c. r. acad. sci. (paris) 146, 530–533 (1908)]," *American Journal of Physics*, vol. 65, no. 11, pp. 1079–1081, 1997.

[28] A. Brünger, C. L. Brooks, and M. Karplus, "Stochastic boundary conditions for molecular dynamics simulations of st2 water," *Chemical Physics Letters*, vol. 105, no. 5, pp. 495–500, 1984.

[29] J. Åqvist, P. Wennerström, M. Nervall, S. Bjelic, and B. O. Brandsdal, "Molecular dynamics simulations of water and biomolecules with a monte carlo constant pressure algorithm," *Chemical Physics Letters*, vol. 384, no. 4, 2004.

[30] S. Nosé, "A unified formulation of the constant temperature molecular dynamics methods," *The Journal of Chemical Physics*, vol. 81, no. 1, pp. 511–519, 1984.

[31] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Phys. Rev. A*, vol. 31, pp. 1695–1697, Mar 1985.

[32] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, "A smooth particle mesh ewald method," *The Journal of Chemical Physics*, vol. 103, no. 19, pp. 8577–8593, 1995.

[33] N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "Mdanalysis: A toolkit for the analysis of molecular dynamics simulations," *Journal of Computational Chemistry*, vol. 32, no. 10, pp. 2319–2327, 2011.

[34] Richard J. Gowers, Max Linke, Jonathan Barnoud, Tyler J. E. Reddy, Manuel N. Melo, Sean L. Seyler, Jan Domański, David L. Dotson, Sébastien Buchoux, Ian M. Kenney, and Oliver Beckstein, "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations," in *Proceedings of the 15th Python in Science Conference* (Sebastian Benthall and Scott Rostrup, eds.), pp. 98 – 105, 2016.

[35] cbc-univie mdy, "New Analysis." `https://github.com/cbc-univie/mdy-newanalysis-package`, 2024. accessed: 2026-02-10.

[36] D. R. L. e., *CRC Handbook of Chemistry and Physics: A Ready-Reference of Chemical and Physical Data, 85th ed Edited by David R. Lide (National Institute of Standards and Technology). CRC Press LLC: Boca Raton, FL. 2004. 2712 pp. $139.99. ISBN 0-8493-0485-7.* Journal of the American Chemical Society, 2005.

[37] A. J. W. E. Chickos JS, *Enthalpies of Vaporization of Organic and Organometallic Compounds.* Journal of Physical and Chemical Reference Data, 2003.

[38] D. Braun, S. Boresch, and O. Steinhauser, "Transport and dielectric properties of water and the influence of coarse-graining: Comparing bmw, spc/e, and tip3p models," *The Journal of Chemical Physics*, vol. 140, no. 6, p. 064107, 2014.

[39] F. Zeng, R. Wan, Y. Xiao, S. Fan, C. Peng, and H. Liu, *Predicting the Self-Diffusion Coefficient of Liquids Based on Backpropagation Artificial Neural Network: A Quantitative Structure–Property Relationship Study.* Industrial and Engineering Chemistry Research, 2022.

[40] D. H. Brookes and T. Head-Gordon, "Family of oxygen–oxygen radial distribution functions for water," *The Journal of Physical Chemistry Letters*, vol. 6, no. 15, pp. 2938–2943, 2015. PMID: 26267185.

[41] J. Behler, "Four generations of high-dimensional neural network potentials," *Chemical Reviews*, vol. 121, no. 16, pp. 10037–10072, 2021. PMID: 33779150.

[42] X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, "Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations," *arXiv preprint arXiv:2210.07237*, 2023.

Appendix

Table 1: Condensed phase properties of **water** for 1 ns initial simulation run of FF, ANI-2x and MACE-OFF23(S).

Water	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	43.99	0.45	0.99	0.03	0.997
FF	43.49	0.52	1.06	0.10	1.01
ANI-2x	79.92	0.71	2.08	-0.28	1.00
MACE-OFF23(S)	47.79	0.26	1.64	0.09	1.12

Table 2: Relative error (eq. 3.3) of **water** for 1 ns initial simulation run of FF, ANI-2x and MACE-OFF23(S).

Water	ΔH_{vap}	κ	C_p	α	ρ
FF	-0.01	0.16	0.07	2.33	0.01
ANI-2x	0.82	0.58	1.10	-10.33	0.00
MACE-OFF23(S)	0.09	-0.42	0.66	2.0	0.12

Table 3: Condensed phase properties of **water** for five independent repeats of the 1 ns production runs of FF, ANI-2x and MACE-OFF23(S).

Water	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	43.99	0.45	0.99	0.03	0.997
FF 1	43.49	0.58	0.95	0.10	1.01
FF 2	43.48	0.53	1.08	0.09	1.01
FF 3	43.47	0.54	1.08	0.08	1.01
FF 4	43.48	0.55	1.07	0.10	1.01
FF 5	43.50	0.51	1.00	0.08	1.01
FF Mean	43.48	0.54	1.03	0.09	1.01
FF Std Dev	0.01	0.02	0.06	0.01	0.00
rel. error	-0.01	0.20	0.04	2.0	0.01
ANI-2x 1	80.29	0.10	1.22	0.01	0.98
ANI-2x 2	80.36	0.13	1.17	0.00	0.98
ANI-2x 3	80.22	0.27	1.19	-0.02	0.99
ANI-2x 4	80.30	0.19	1.74	-0.08	0.99
ANI-2x 5	80.27	0.16	1.29	-0.03	0.98
ANI-2x Mean	80.29	0.17	1.32	-0.03	0.98
ANI-2x Std Dev	0.05	0.07	0.24	0.04	0.00
rel. error	0.83	-0.62	0.33	-2.0	-0.02
MACE-OFF23(S) 1	47.80	0.30	1.79	0.12	1.12
MACE-OFF23(S) 2	47.80	0.28	1.60	0.09	1.12
MACE-OFF23(S) 3	47.79	0.30	1.69	0.11	1.12
MACE-OFF23(S) 4	47.79	0.28	1.72	0.10	1.12
MACE-OFF23(S) 5	47.83	0.29	1.76	0.10	1.12
MACE-OFF23(S) Mean	47.80	0.29	1.71	0.10	1.12
MACE-OFF23(S) Std Dev	0.02	0.01	0.07	0.01	0.00
rel. error	0.09	-0.36	0.73	2.33	0.12

Table 4: Condensed phase properties of **benzene** for 1 ns initial simulation run of FF, ANI-2x and MACE-OFF23(S).

Benzene	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	33.83	0.97	0.42	0.11	0.88
FF	31.68	1.20	1.15	0.18	0.84
ANI-2x	33.61	0.63	1.06	0.10	0.93
MACE-OFF23(S)	35.68	0.71	1.26	0.16	1.02

Table 5: Relative error (eq. 3.3 of **benzene** for 1 ns initial simulation run of FF, ANI-2x and MACE-OFF23(S).

Benzene	ΔH_{vap}	κ	C_p	α	ρ
FF	-0.06	0.24	1.74	0.64	-0.04
ANI-2x	-0.01	-0.35	1.52	-0.09	0.06
MACE-OFF23(S)	0.05	-0.27	2.00	0.45	0.16

Table 6: Condensed phase properties of **benzene** for five independent repeats of the 1 ns production runs of FF, ANI-2x and MACE-OFF23(S).

Benzene	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	33.83	0.97	0.42	0.11	0.88
FF 1	31.74	1.25	1.02	0.15	0.85
FF 2	31.71	1.18	1.03	0.14	0.85
FF 3	31.68	1.17	1.15	0.17	0.85
FF 4	31.70	1.05	1.01	0.13	0.85
FF 5	31.67	1.28	1.04	0.16	0.84
FF Mean	31.70	1.19	1.05	0.15	0.85
FF Std Dev	0.03	0.09	0.06	0.02	0.00
rel. error	-0.06	0.23	1.50	0.36	-0.03
ANI-2x 1	33.67	0.61	1.07	0.09	0.93
ANI-2x 2	33.63	0.58	1.05	0.07	0.93
ANI-2x 3	33.50	0.54	1.08	0.06	0.93
ANI-2x 4	33.60	0.63	1.05	0.08	0.93
ANI-2x 5	33.59	0.60	1.09	0.08	0.93
ANI-2x Mean	33.60	0.59	1.07	0.08	0.93
ANI-2x Std Dev	0.06	0.03	0.02	0.01	0.00
rel. error	-0.01	-0.40	1.55	-0.27	0.06
MACE-OFF23(S) 1	35.38	0.69	1.21	0.14	1.02
MACE-OFF23(S) 2	35.45	0.67	1.12	0.12	1.02
MACE-OFF23(S) 3	35.72	0.80	1.21	0.19	1.02
MACE-OFF23(S) 4	35.79	0.66	1.11	0.12	1.03
MACE-OFF23(S) 5	35.62	0.76	1.25	0.17	1.02
MACE-OFF23(S) Mean	35.60	0.72	1.18	0.15	1.02
MACE-OFF23(S) Std Dev	0.18	0.06	0.06	0.03	0.00
rel. error	0.05	-0.26	1.81	0.36	0.16

Table 7: Condensed phase properties of **n-hexane** for 1 ns initial simulation run of FF, ANI-2x and MACE-OFF23(S).

n-hexane	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	31.52	1.67	0.54	0.14	0.655
FF	32.12	1.91	1.40	0.17	0.65
ANI-2x	28.62	2.17	1.58	0.22	0.74
MACE-OFF23(S)	33.10	3.79	2.19	0.53	0.79

Table 8: Relative error (eq. 3.3 of **n-hexane** for 1 ns initial simulation run of FF , ANI-2x and MACE-OFF23(S).

n-hexane	ΔH_{vap}	κ	C_p	α	ρ
FF	0.02	0.14	1.59	0.21	-0.01
ANI-2x	-0.09	0.30	1.93	0.57	0.13
MACE-OFF23(S)	0.05	1.27	3.05	2.79	0.21

Table 9: Condensed phase properties of **n-hexane** for five independent repeats of the 1 ns production runs of FF, ANI-2x and MACE-OFF23(S).

n-hexane	$\Delta H_{vap}[kJ/mol]$	$\kappa[10^{-4}/bar]$	$C_p[cal/g/K]$	$\alpha[10^{-2}/K]$	$\rho[g/mL]$
Experiment	31.52	1.67	0.54	0.14	0.655
FF 1	32.21	1.90	1.61	0.18	0.65
FF 2	32.11	1.75	1.59	0.17	0.65
FF 3	32.29	1.77	1.55	0.15	0.65
FF 4	32.23	1.98	1.64	0.20	0.65
FF 5	32.06	1.70	1.56	0.17	0.65
FF Mean	32.18	1.82	1.59	0.17	0.65
FF Std Dev	0.09	0.12	0.04	0.02	0.00
rel. error	0.02	0.09	1.94	0.21	-0.01
ANI-2x 1	28.45	1.92	1.57	0.17	0.74
ANI-2x 2	28.45	1.98	1.68	0.23	0.74
ANI-2x 3	28.47	1.79	1.61	0.20	0.74
ANI-2x 4	28.35	2.30	1.80	0.25	0.74
ANI-2x 5	28.48	2.00	1.55	0.19	0.74
ANI-2x Mean	28.44	2.00	1.64	0.21	0.74
ANI-2x Std Dev	0.05	0.19	0.10	0.03	0.00
rel. error	-0.10	0.20	2.04	0.50	0.13
MACE-OFF23(S) 1	32.61	2.65	1.93	0.35	0.78
MACE-OFF23(S) 2	32.60	3.57	1.84	0.39	0.78
MACE-OFF23(S) 3	32.34	5.05	2.29	0.66	0.77
MACE-OFF23(S) 4	32.88	3.23	2.00	0.46	0.79
MACE-OFF23(S) 5	32.53	7.70	2.37	0.80	0.78
MACE-OFF23(S) Mean	32.59	4.44	2.09	0.53	0.78
MACE-OFF23(S) Std Dev	0.20	2.03	0.23	0.19	0.01
rel. error	0.03	1.66	2.87	2.79	0.19

Table 10: Self-diffusion coefficient from 3.5 ns NVT simulation and the experiment [39]

D [10^{-9} m^2/s]	Experiment	FF	ANI-2x	MACE-OFF23(S)
water	2.29	5.35	$2.88 * 10^{-3}$	2.19
benzene	2.15	1.30	$2.43 * 10^{-1}$	$2.26 * 10^{-1}$
n-hexane	4.14	3.15	$4.05 * 10^{-2}$	$1.14 * 10^{-1}$

List of Figures

List of Tables