Machine Learning in QM/MM Molecular Dynamics Simulations of Condensed-Phase Systems

Lennard Böselt, Moritz Thürlemann, and Sereina Riniker*

Laboratory of Physical Chemistry, ETH Zurich, Vladimir-Prelog-Weg 2, 8093 Zurich, Switzerland

Abstract

Quantum mechanics/molecular mechanics (QM/MM) molecular dynamics (MD) simulations have been developed to simulate molecular systems, where an explicit description of changes in the electronic structure is necessary. However, QM/MM MD simulations are computationally expensive compared to fully classical simulations as all valence electrons are treated explicitly and a self-consistent field (SCF) procedure is required. Recently, approaches have been proposed to replace the QM description with machine learned (ML) models. However, condensed-phase systems pose a challenge for these approaches due to long-range interactions. Here, we establish a workflow, which incorporates the MM environment as an element type in a high-dimensional neural network potential (HDNNP). The fitted HDNNP describes the potential-energy surface of the QM particles with an electrostatic embedding scheme. Thus, the MM particles feel a force from the polarized QM particles. To achieve chemical accuracy, we find that even simple systems require models with a substantial number of parameters, posing the risk of overfitting. To address this issue, we extend our approach to a $\Delta$-learning scheme, where the ML model learns the difference between a reference method (DFT) and a cheaper semi-empirical method (DFTB). We show that such a scheme reaches the accuracy of the DFT reference method, while requiring significantly less parameters. Furthermore, the $\Delta$-learning scheme is capable of correctly incorporating long-range interactions within a cutoff of 1.4 nm. It is validated by performing MD simulations of retinoic acid in water and the interaction between S-adenoslymethioniat with cytosine in water. The presented results indicate that $\Delta$-learning is a promising approach for (QM)ML/MM MD simulations of condensed-phase systems.

1 Introduction

Classical fixed-charge force fields (FF) are readily used to perform molecular dynamics (MD) simulations of condensed-phase systems. They consist of a relatively small number of parameters, such as bond-stretching, bond-angle bending, and torsional dihedral terms, partial charges and Lennard-Jones parameters. They are partially fitted against experimental values such as the density, heat of vaporisation, and solvation free energy. FF are the gold standard for simulations over long time scales of systems for which long-range interactions are essential. In classical simulations, averaged properties are computed by neglecting electron rearrangement. The parameterization of FF requires the availability of sufficient experimental data. On the other hand, quantum mechanics/molecular mechanics (QM/MM) MD simulations can provide a valuable alternative to classical FF simulations, when changes in the electronic structure are important or if reliable force-field parameters are not available.

*Corresponding author: email: sriniker@ethz.ch, ORCID: 0000-0003-1893-4031
In the QM/MM scheme, the QM zone simulated with density functional theory (DFT) or *ab initio* principles is placed into a classical environment (MM zone). This approach permits the simulation of the electronic structure of small systems in more realistic surroundings. Most crucial in QM/MM simulations is the description of the interaction between the QM and MM zones. This interaction can be based either on (i) mechanical constraints (i.e. “mechanical embedding” scheme), or (ii) electronic perturbations (i.e. “electrostatic embedding”). In mechanical embedding, the particles in the QM zone are assigned partial charges, which then interact with the MM zone on a classical level. Thus, the MM particles do not interact with the electron density of the QM solute but rather with point charges on a classical level. Mechanical embedding favours efficient implementations, however, a major drawback is that a set of classical parameters have to be determined for the QM zone, which is not trivial and sometimes not possible with sufficient accuracy.

In electrostatic embedding, the MM environment is incorporated into the Hamiltonian operator of the QM zone as electron operators. Thus, the electron density of the QM zone is perturbed by the MM environment, and the MM zone in turn interacts with the perturbed electronic structure of the QM solute. In other words, the MM particles feel a force from the MM-polarized QM solute. The advantages of electrostatic embedding are that no partial charges have to be assigned for the computation of the interactions between the MM and the QM zones, and that the description of the interaction is physically better motivated compared to mechanical embedding. A well known limitation is the neglect of polarization of the MM environment – unless polarizable FF are used. Furthermore, computation is more expensive compared to mechanical embedding, and it is not clear whether the partial charges of the MM zone are suitable for inclusion as one-electron Hamiltonians in the QM zone. Generally, an electrostatic embedding scheme is to be preferred as it has been shown to be more accurate, and it reflects thus the current standard protocol for QM/MM simulations.

In QM/MM, the description of the QM zone becomes the computational bottle-neck as it requires an expensive self-consistent field (SCF) procedure, and explicit treatment of all valence electrons. To partially circumvent these issues, semi empirical methods can be used to describe the QM zone. This extends the accessible time scales but also reduces the accuracy. An alternative is to employ machine-learned (ML) potentials.

In recent years, a lot of research effort has been invested into the development of ML models trained on the potential energy surface (PES) of QM systems. This enables ML-MD simulations at an accuracy level close to that of the electronic structure method chosen to generate the training set. The costs of the resulting ML-MD simulations are reduced drastically compared to a normal DFT or *ab initio* MD simulation as a SCF procedure is no longer necessary and the valence electrons do not have to be treated explicitly. A large amount of approaches have been reported in the literature to achieve this task. These approaches differ in:

- **Use case:** e.g. gas-phase or periodic box, small compounds or larger compounds, system specific or system unspecific
- **Descriptor used to encode the chemical structure:** e.g. Coulomb matrices, distance matrices, low-order polynomials, or so-called symmetry functions
- **Target output:** e.g. energies or gradients
- **Scope:** e.g. global or local
- **ML method:** e.g. neural networks or Kernel methods
- **Applying a restriction of the final form of the PES**
Other ML models do not target the PES but rather the electron density or the wave function. However, these will not be subject of the current study. All of the approaches listed above partially generate the correct parity behaviour of the system. In other words, swapping two elements of the same type and rotating or translating the system leaves the output unchanged or changes it equivariantly. The usage of the ML models heavily depends on the specific use case. In the following, selected approaches are discussed in more detail.

Permutational-invariant polynomial neural networks (PIP) are suitable for accurately describing reactions of small systems in the gas phase (less than 5 atoms). Here, low-order polynomials, which depend on the internuclear distance of the compounds, are used as a descriptor for rigorously introducing the correct parity behaviour for the system studied. The neural networks are trained on the total energy of the system and are global. Symmetric gradient domain machine learning (sGDML) targets systems up to 30 atoms in the gas phase, which do not undergo a change in topology. In this case, the compound is encoded as an inverse distance matrix. The latter is fed into a Kernel ridge regression model and trained in the gradient domain, i.e. the target output is the gradient of the system. The energy can be obtained by integrating the forces. The definition of the kernel used ensures that the integrated forces give rise to a continuously-differentiable PES. The models can be employed for spectroscopic studies.

Larger compounds and box simulations can be tackled with high-dimensional neural network potentials (HDNNP). The environment of each atom is encoded using symmetry functions. Simple examples of symmetry functions are Gaussians that depend on the interatomic distance, or trigonometric functions. They map the input to a higher dimension, which is subsequently summed up. The summation operator and the mapping to higher dimensions using symmetry functions introduces the correct parity behaviour while still allowing the HDNNP to learn the PES accurately. Using symmetry functions, Smith et al. showed that a universal force-field (ANI-1) can be constructed, which is capable of describing unseen compounds containing the same elements. The model was trained in a system unspecific manner on gas-phase data. The models can be employed in the drug-discovery process. For periodic-box simulations, HDNNP can be trained system-specific on previously sampled data points. The sampling of the data points can be done on a lower level of theory, as long as the trajectory samples all structures important for learning the PES. The data points are then re-computed on the desired level, and the HDNNP is trained to interpolate between the training data points. These models employ a cutoff for the symmetry functions and are thus local. They are trained on the energy and use optionally a gradient correction. HDNNP based on symmetry functions motivated other groups to apply a similar approach for different target outputs such as gradients, or with different ML models such as Kernel methods. The uniqueness of the descriptor has been discussed in theoretical work.

Another important contribution to the field is the concept of \(\Delta\)-learning. Here, the ML model is trained to reproduce the difference between two QM calculations, an expensive higher-level and a cheaper lower-level method. Different ML methods and descriptors can be used. Subsequently, the output on the higher level can be partially recovered by performing the calculation with the cheap method and applying the trained ML model. A good example of the \(\Delta\)-learning scheme was proposed by Shen et al. where a modified HDDNP was trained on the energy difference between a semi-empirical and an expensive QM method. It was used to reweight the free-energy profiles of reactions obtained with a QM/MM approach. More recently, the same authors introduced a model, which can be used to perform MD simulations. The approach requires as input for the ML model a reaction coordinate and the partial charges from the lower level method.

In our opinion and that of others, HDNNP have been proven to be the most suitable method for performing system-specific periodic-box simulations. However, their usage in the simulation of condensed-phase systems with biological relevance is still hampered because such systems typically
1. **Large number of element types**: HDNNP input descriptors scale exponentially with the number of element types.

2. **Many rotatable bonds and/or no symmetry, large system size**: The quality of HDNNP model depends on a densely sampled training set. The number of different possible system configuration scales exponentially with the number of rotatable bonds, which renders sufficient sampling for the training set increasingly challenging. In principle, if the ML model is able to learn the underlying physics, it could extrapolate to unseen system configurations. However, to the best of our knowledge, this was only partially achieved yet. The models, which achieve this task partially, employ a small cutoff for the multi-body (0.32 nm) and two-body terms (0.52 nm).

3. **Important long-range interactions**: HDNNP make use of a locality ansatz, which fails to describe long-range electrostatics. Long-range interactions can partially be introduced via a charge partition scheme and a charge summation scheme (e.g., Ewald summation). In QM/MM, this would correspond to a mechanical embedding scheme, but these are to the best of our knowledge not yet applicable to simulations of large condensed-phase systems.

4. **Large cutoff**: Classical force fields typically work with a cutoff of 1.0 - 1.4 nm for pairwise nonbonded interactions. Such large cutoffs are necessary to achieve the desired accuracy for bulk properties such as heat of vaporization or solvation free energy.

5. **Long time scales**: Although a prediction with the ML model is faster than a QM calculation, a small integration step (i.e., 0.5 fs) is still required for simulations. Furthermore, obtaining the gradients from a ML potential is usually significantly slower than calculating the gradients with a classical force field.

In addition, HDNNP are generally plagued by two issues: (i) Generating training sets can imply a significant time investment, and (ii) frequently used fully connected neural networks are prone to overfitting, especially if a high number of weights is necessary to converge to chemical accuracy.

A promising alternative for condensed-phase systems is therefore the combination of HDNNP and classical FF in a QM/MM-type approach, where the HDNNP is used for simulating the QM particles and to compute the interactions between the MM and the QM zone with electrostatic embedding. By using HDNNP for the QM zone, longer time scales will become accessible than with standard QM/MM simulations. Such a hybrid approach requires the training data set to be generated with a QM/MM scheme. In other words, the MM environment is incorporated as one-electron Hamiltonians in the QM reference calculation. With such an approach, the complexity of the generation of the training set is reduced drastically as not all atoms are treated on a QM level. Furthermore, it permits the use of a larger cutoff. Including long-range interactions directly in the ML model is therefore possible.

In this work, we assess the concept of (QM)ML/MM MD simulations for condensed-phase systems. To enable electrostatic embedding, we integrate the MM environment directly as an additional element type in the HDNNP. The symmetry functions including MM particles are weighted with their respective partial charge. Two-body and three-body symmetry functions are included for the description of the MM environment. The symmetry functions used follow the proposal of Smith et al. and are adapted for larger cutoffs. In the first part, we explore the effect of different parameters on the accuracy of (QM)ML/MM calculations of small molecules in water. The parameters include the application of gradient correction, the inclusion of three-body terms, and the cutoff distance for the descriptors. In the second part, we extend the methodology to a Δ-learning scheme with a semi-empirical method.
Semi-empirical methods are known to recover long-range interactions well\(^{11}\), an issue ML models with a locality ansatz usually struggle with. The approach is validated using different molecular systems. Finally, MD simulations are performed for retinoic acid in water (50 atoms in the QM zone and approximately 2500 classical partial charges), and the interaction of SAM with cytosin (63 atoms in the QM zone and approximately 3000 partial charges). The results are compared to standard QM/MM simulations.

2 Theory

2.1 Calculation of the Potential Energy of the System

2.1.1 QM Calculation

In non-relativistic Born-Oppenheimer MD simulations\(^ {15}\), the potential energy \( V \) of the system is a function of all Cartesian coordinates of the system,

\[
V_{\vec{r}}(\vec{R}) \equiv V,
\]

where \( \vec{R} \) are the Cartesian coordinates of all nuclei, and \( \vec{r} \) are the Cartesian coordinates of the electrons. The subscript \( \vec{r} \) indicates that the electronic configuration is treated under the Born-Oppenheimer approximation\(^ {16}\). According to Newton’s second law\(^ {17}\), the gradient on the nuclei of the system can be obtained by taking the derivative with respect to the coordinates \( \vec{R} \),

\[
- \frac{\partial V_{\vec{r}}(\vec{R}_i)}{\partial \vec{R}_i} = \vec{F}_i = m_i \frac{d\vec{v}_i}{dt}.
\]

In Eq. (2), \( \vec{F}_i \) is the force of particle \( i \), \( \vec{R}_i \) its coordinates, \( m_i \) its mass, \( \vec{v}_i \) its velocity, and \( t \) is the time. The system can then be propagated in time \( t \) using an integration algorithm such as the leap-frog algorithm\(^ {18}\). \( V \) can be computed by solving the Schrödinger equation\(^ {19}\),

\[
\hat{H} \psi_{\vec{r}}(\vec{R}) = V_{\vec{r}}(\vec{R}) \psi_{\vec{r}}(\vec{R}).
\]

In Eq. (3), \( \hat{H} \) is the full Hamiltonian operator, and \( \psi \) is the time-independent Born-Oppenheimer approximated wave function of the system. Solving for \( V \) is computationally unfeasible for large systems. Hence, approximated solutions to Eq. (3) were developed, which can be grouped into four classes:\(^ {20}\) (1) \textit{Ab initio} methods use the correct Hamiltonian and an approximated wave function \( \psi \). (2) Density functional theory\(^ {21}\) (DFT) computes the electron density \( \rho_{\vec{r}}(\vec{R}) \), which is used to calculate the energy of the molecular system, i.e. \( V_{\vec{r}}[\rho_{\vec{r}}(\vec{R})] \). Modern DFT approaches achieve an accuracy comparable to \textit{ab initio} methods (or above), while still being able to simulate larger systems. However, DFT requires a treatment of all valence electrons and an SCF procedure, which makes it unfeasible to perform long MD simulations of large systems. (3) Semi-empirical methods use an approximated Hamiltonian \( \hat{H} \) and correct for the error made by introducing a set of fitted parameters. Most prominent example is the PM7 method\(^ {22}\). Another attractive alternative is density functional tight binding (DFTB)\(^ {23}\). In DFTB, the electron density \( \rho \) is expanded in a series around a reference density \( \rho_0 \) and truncated. The error made is corrected with fitted parameters, similarly to other semi-empirical methods. (4) The last class includes empirical models with effective parameters such as classical force fields\(^ {18,24,25}\).
2.1.2 Classical Force Fields

In classical fixed-charge MD simulations, the potential energy $V^N$ of the system is calculated with a force field,\(^1\) which is the sum of bonded and nonbonded interactions terms,

$$V^N(\vec{R}) = V^{\text{bond}} + V^{\text{angle}} + V^{\text{dihedral}} + V^{\text{el}} + V^{\text{vdW}}, \quad (4)$$

where $V^{\text{bond}}$ is the contribution of all covalent bonds, $V^{\text{angle}}$ that of all covalent angles, and $V^{\text{dihedral}}$ that of all covalent dihedrals. The nonbonded terms consist of electrostatic ($V^{\text{el}}$) and van der Waals ($V^{\text{vdW}}$) interactions. The parameters of these interactions terms are typically fitted to reproduce experimental and/or QM reference data. Eq. (4) is inexpensive to solve, and depends only on the number of nuclei (atoms). However, fixed-charge force fields treat electronic effects only in an averaged field, and can thus not be used to simulate chemical reactions. The simplicity of Eq. (4) allows to use relatively large cutoffs, and methods are available to include long-range interactions beyond a given cutoff.

2.1.3 QM/MM Scheme

QM/MM is a hybrid approach that combines a QM subsystem with an MM environment,\(^5\) thus striking a balance between cost and accuracy. The challenge is to describe the interactions between the parts appropriately. Here, the question is how to obtain the total energy $V^{\text{total}}$ of the system, which is now a combination of the QM subsystem $V^{\text{QM}}$ and the MM surrounding $V^{N,\text{MM}}$. A first approach to compute the total energy of the system is a subtractive scheme,\(^7\)

$$V^{\text{total}} = V^{\text{QM}}(\vec{r}) + V^{\text{N, total}}(\vec{R}) - V^{\text{N, QM}}(\vec{R}), \quad (5)$$

where $V^{\text{QM}}(\vec{r})$ is the energy of the QM subsystem, $V^{\text{N, total}}(\vec{R})$ the energy of the complete system calculated with the classical force field, and $V^{\text{N, QM}}(\vec{R})$ the energy of the QM zone calculated with the force field. The more prominent alternative is the additive scheme,\(^7\) which is also used in this work, i.e.

$$V^{\text{total}} = V^{\text{QM}} + V^{\text{N, MM}} + V^{\text{QM-MM}}. \quad (6)$$

In this case, the force field is used to calculate the energy of the MM region, while the interaction between the QM and MM parts ($V^{\text{QM-MM}}$) is added to the energy expression. The latter term can be computed either via a mechanical embedding or an electrostatic embedding scheme.\(^9\) In mechanical embedding, the particles in the QM subsystem are assigned partial charges, which then interact with the MM surrounding. However, as already discussed in the Introduction, this scheme is inferior to electrostatic embedding, where the Hamiltonian operator $\hat{H}$ is extended as (in atomic units)\(^13\)

$$\hat{H}^{\text{QM/MM}} = \hat{H} - \sum_i \sum_j N_{\text{MM}} q_j \frac{1}{|\vec{r}_i - \vec{R}_j|} + \sum_i \sum_j N_{\text{QM}} q_i N_{\text{MM}} \frac{Q_j}{|\vec{R}_i - \vec{R}_j|}. \quad (7)$$

$N_{\text{el}}$ is here the number of the QM electrons in the system, $N_{\text{MM}}$ the number of partial charges, $N_{\text{QM}}$ the number of QM atoms, $q_j$ the partial charge of MM atom $j$, $\vec{r}_i$ the coordinates of electron $i$, $\vec{R}_j$ the coordinates of MM atom $j$, and $\vec{R}_i$ the coordinates of QM atom $i$. The QM subsystem is thus directly influenced by the MM partial charges, while the latter feel a force of the perturbed QM zone. The energy expression of the complete system can then be written as,

$$(\hat{H}^{\text{QM/MM}} + V^{N,\text{MM}})\psi_{\vec{r}} = (V^{\text{QM}} + V^{\text{QM/MM}} + V^{N,\text{MM}})\psi_{\vec{r}} = V^{\text{total}}\psi_{\vec{r}}. \quad (8)$$
Feed-forward neural networks (FNN) consist of a number of sequentially chained (non)-linear functions. The concepts described above are independent of the ML method used and the transform of words, an additional loss term. As a third option, information about the gradients can be included in the training process. In other words, an additional loss term \( L_P(V_{\tilde{V}}(\tilde{R}), \tilde{V}_{\tilde{r}}(\tilde{R})) = \arg\min_P(g_c(V_{\tilde{r}}(\tilde{R}), \tilde{V}_{\tilde{r}}(\tilde{R}))) \) becomes minimal, where \( g_c \) is an arbitrary convex function. Examples are mean squared errors and mean absolute errors. \( \tilde{V}_{\tilde{r}}(\tilde{R}) \) can subsequently be used to propagate an MD trajectory.

Different target outputs can be chosen. Instead of learning the complete potential \( \tilde{V}_{\tilde{r}}(\tilde{R}) \), it is possible to learn only the difference between two levels of theory (e.g. semi-empirical and DFT). The function \( f \) thus becomes a mapping of \( f_P : h(\tilde{R}) \rightarrow \Delta \tilde{V}_{\tilde{r}}(\tilde{R}). \) Alternatively, it is possible to directly learn the gradient \( f_P : h(\tilde{R}) \rightarrow \frac{\partial \tilde{V}_{\tilde{r}}}{\partial \tilde{R}}. \) An obstacle when learning gradients is that \( \int_{-\infty}^{\infty} d\tilde{R} \frac{\partial \tilde{V}_{\tilde{r}}}{\partial \tilde{R}} = V \) must be ensured. This can for example be achieved with the sGDML method. As a third option, information about the gradients can be included in the training process. In other words, an additional loss term \( L_P(V, \tilde{V}) = \arg\min_P(g_c(V, \tilde{V}) + g_c(\frac{\partial V}{\partial \tilde{R}}, \frac{\partial \tilde{V}}{\partial \tilde{R}})) \) is added to account for the gradients. The concepts described above are independent of the ML method used and the transformation \( h \) is applied beforehand. It is essential that \( h \) does not remove information relevant for the target output.

2.2 Machine Learning of Potential-Energy Surfaces

Supervised ML methods can be used to learn a function \( f_P \), which describes the relation \( f_P : h(\tilde{R}) \rightarrow \tilde{V}_{\tilde{r}}(\tilde{R}). \) The tilde denotes that the quantity is estimated. \( P \) is a set of parameters introduced by the ML model. The idea of the ML approaches is to sample configurations \( \tilde{R} \) on a PES to obtain the potential energy \( \tilde{V}_{\tilde{r}}(\tilde{R}) \), and then fit the parameters \( P \) in the function \( f_P \) to obtain an accurate description of \( \tilde{V}_{\tilde{r}}(\tilde{R}) \). The configurations \( \tilde{R} \) can be transformed using a function \( h \) before being fed into the ML model. An example would be a distance transformation (distance matrix). The parameters \( P \) of the ML model are fitted such that the loss \( L_P(V_{\tilde{r}}(\tilde{R}), \tilde{V}_{\tilde{r}}(\tilde{R})) = \arg\min_P(g_c(V_{\tilde{r}}(\tilde{R}), \tilde{V}_{\tilde{r}}(\tilde{R}))) \) becomes minimal, where \( g_c \) is an arbitrary convex function. Examples are mean squared errors and mean absolute errors. \( \tilde{V}_{\tilde{r}}(\tilde{R}) \) can subsequently be used to propagate an MD trajectory.

Different target outputs can be chosen. Instead of learning the complete potential \( \tilde{V}_{\tilde{r}}(\tilde{R}) \), it is possible to learn only the difference between two levels of theory (e.g. semi-empirical and DFT). The function \( f \) thus becomes a mapping of \( f_P : h(\tilde{R}) \rightarrow \Delta \tilde{V}_{\tilde{r}}(\tilde{R}). \) Alternatively, it is possible to directly learn the gradient \( f_P : h(\tilde{R}) \rightarrow \frac{\partial \tilde{V}_{\tilde{r}}}{\partial \tilde{R}}. \) An obstacle when learning gradients is that \( \int_{-\infty}^{\infty} d\tilde{R} \frac{\partial \tilde{V}_{\tilde{r}}}{\partial \tilde{R}} = V \) must be ensured. This can for example be achieved with the sGDML method. As a third option, information about the gradients can be included in the training process. In other words, an additional loss term \( L_P(V, \tilde{V}) = \arg\min_P(g_c(V, \tilde{V}) + g_c(\frac{\partial V}{\partial \tilde{R}}, \frac{\partial \tilde{V}}{\partial \tilde{R}})) \) is added to account for the gradients. The concepts described above are independent of the ML method used and the transformation \( h \) is applied beforehand. It is essential that \( h \) does not remove information relevant for the target output.

2.2.1 High-Dimensional Neural Network Potentials (HDNNP)

Feed-forward neural networks (FNN) consist of a number of sequentially chained (non)-linear functions \( \sigma^l \), weight matrices \( W^l \) and biases \( b^l \), where \( l \) is the number of layers. A FNN can be defined recursively as

\[
x^{l+1} = \sigma^l(W^l x^l + b^l),
\]

where \( x \) is the input vector. The weight matrices and biases correspond to the parameters \( P \). In addition, there are user-defined hyperparameters such as the number of layers \( l \), the number of neurons per layer, and the type of functions \( \sigma^l \).

FNN are the method of choice in cases, where a memory intensive data set is required and complicated transformations are necessary in order to learn the underlying relation between the input \( z \) and the output \( \tilde{V} \). Moreover, they provide more flexibility compared to Kernel ridge regression and Gaussian processes. It has been observed in Kaggle contests that two hidden layers usually perform best if no special activation functions are used. In addition, FNN with a lower number of parameters tend to generalize better and require smaller training sets.
The FNN architecture does not contain any information about the parity of the system. In chemistry, this is an issue because the rotation and translation of the system or swapping of atoms of the same type need to leave the total energy of the system unchanged. It is possible to augment the training data set in order to force the FNN to learn the correct symmetry. However, this increases the training set by multiple orders of magnitude. Behler et al.\textsuperscript{18} identified this issue and developed a FNN architecture and an input-coordinate transformation $h$, which guarantee the correct parity behaviour, the so-called high-dimensional neural network potentials (HDNNP). Instead of using only one FNN, each element type in the system is described by its own FNN. Each FNN is called an NNP because it describes the atomic energy of the respective atom type. The total energy of the system $V$ is thus separated, i.e., $V = \sum_i^{N_H} V_i^H + \sum_i^{N_C} V_i^C + \ldots$, where $N_H$ is the number of hydrogen atoms, $N_C$ the number of carbon atoms, and so on. Note that this separation ansatz is an approximation, it has no quantum-mechanical foundation. It guarantees that the total energy of a system is invariant with respect to swapping the positions of two atoms of the same type. However, rotation and translation still change the total energy. To address this issue, Behler et al. used the concept of symmetry functions.\textsuperscript{18} The Cartesian coordinates are transformed using a set of symmetry functions, which are invariant with respect to rotation and translation,\textsuperscript{18,29}

$$\mathcal{S}^{t,R}_i(r) = \sum_j e^{-\eta (r_{ij} - \mu)^2}$$

and

$$\mathcal{S}^{t,A}_i = 2^{\zeta - 1} \sum_{j \neq k \neq i} (1 - \cos(\theta_{ijk} - \theta_S))^\zeta \cdot e^{-\frac{(r_{ij} + r_{ik} - \mu_A)^2}{2}} \eta_A,$$

where $r_{ij}$ is the distance between atoms $i$ and $j$, and $\theta_{ijk}$ is the angle spanned by atoms $i$, $j$ and $k$. The additional parameters $\eta$, $\mu$, $\mu_A$, $\zeta$, $\theta_S$ and $\eta_A$ need to be set manually. The superscript $t$ (for type) indicates that the summation only runs over a specific element type. Symmetry functions describe the chemical environment of each atom with radial (Eq. (11)) and angular features (Eq. (12)). The summation operator enforces correct parity behaviour.\textsuperscript{18,29}

Applying symmetry functions to periodic boundary conditions requires a cutoff. For this, a cutoff function $f_C$ can be added, i.e.

$$\mathcal{S}^{t,R}_i(r) = \sum_j e^{-\eta (r_{ij} - \mu)^2} \cdot f_C(r_{ij}; R^{QM-QM})$$

and

$$\mathcal{S}^{t,A}_i = 2^{\zeta - 1} \sum_{j \neq k \neq i} N (1 - \cos(\theta_{ijk} - \theta_S))^\zeta \cdot e^{-\frac{(r_{ij} + r_{ik} - \mu_A)^2}{2}} \eta_A \cdot f_C(r_{ij}; R_A) \cdot f_C(r_{ik}; R_A),$$

with the cutoff function $f_C$ defined as,

$$f_C(r_{ij}; R) = \begin{cases} 0.5 \cdot (1 + \cos(\frac{r_{ij} \pi}{R})) & \text{for } r_{ij} \leq R \\ 0 & \text{else} \end{cases}$$

To mimic long-range interactions, a second ML model can be trained on QM-derived partial charges and multipole moments to be used in an Ewald summation scheme. For covalently bonded or metallic systems, the cutoff $R^{QM-MM}$ can be chosen relatively small because the contributions of the long-range interactions are minor. However, as discussed in the Introduction, much larger cutoffs are required for condensed-phase systems.

Even though HDNNP are a powerful technique, they are limited in the following ways: First, the number of angle terms of the symmetry functions scale exponentially with the number of element
types as all possible combinations must be considered. For example, a system consisting of hydrogen and carbon atoms involves five unique angle combinations. Introducing an additional element type (e.g., oxygen) adds five more unique combinations. This issue can be resolved partially by introducing weighted symmetry functions, i.e., instead of introducing a new element type, the symmetry functions are weighted with respect to their atomic number and/or charge. Other issues with HDNNP are that the full correct permuting behaviour is only partially recovered, and that the descriptors used are non-unique. However, the latter issue is in our opinion more of theoretical nature, and should not limit the usage of HDNNPs in practical applications. Lastly, one has to keep in mind that HDNNPs do not resolve the inherent general limitations of ML approaches. For example, the requirement for large training sets, danger of overfitting, and limited extrapolation capabilities apply also to HDNNPs.

2.3 (QM)ML/MM MD Simulations Using HDNNP

In order to combine HDNNP and QM/MM MD simulations with electrostatic embedding, it is necessary to incorporate the MM environment into the descriptor of the HDNNP, in order to find a representation \( f_P \) for \( V_{QM} + V_{QM-MM} \) in Eq. (6). \( V_{N,MM} \) is obtained using the classical force field, which means that a significant amount of complexity is removed from the ML model. The MM environment can be encoded in the HDNNP by introducing the MM particles as an additional element type, which is weighted with the respective partial charge. For the QM particles, we obtain

\[
S_{i,R}^{t}(r) = \sum_{j}^{N(t)_{QM}} e^{-\eta(r_{ij} - \mu)^2} \cdot f_C(r_{ij}; R_{sym,r}) \tag{16}
\]

and

\[
S_{i}^{t,A} = 2^{\kappa - 1} \sum_{j \neq k \neq i}^{N(t)_{QM}} (1 - \cos(\theta_{ijk} - \theta_S))^{\zeta} \cdot e^{-(\frac{r_{ij} + r_{ik} - r_A}{2})^2 \cdot \eta_A} \cdot f_C(r_{ij}; R_A) \cdot f_C(r_{ik}; R_A) \tag{17}
\]

MM particles are introduced as a new element type by

\[
S_{i,R}^{t}(r) = \sum_{j}^{N_{MM}} e^{-\eta(r_{ij} - \mu)^2} \cdot f_C(r_{ij}; R_{sym,r}) \cdot Z(j), \tag{18}
\]

\[
S_{i}^{t,A} = 2^{\kappa - 1} \sum_{j \neq i}^{N(t)_{QM}} \sum_{k}^{N_{MM}} (1 - \cos(\theta_{ijk} - \theta_S))^{\zeta} \cdot e^{-(\frac{r_{ij} + r_{ik} - r_A}{2})^2 \cdot \eta_A} \cdot f_C(r_{ij}; R_A) \cdot f_C(r_{ik}; R_A) \cdot Z(k) \tag{19}
\]

and

\[
S_{i}^{t,A} = 2^{\kappa - 1} \sum_{j \neq k}^{N_{MM}} (1 - \cos(\theta_{ijk} - \theta_S))^{\zeta} \cdot e^{-(\frac{r_{ij} + r_{ik} - r_A}{2})^2 \cdot \eta_A} \cdot f_C(r_{ij}; R_A) \cdot f_C(r_{ik}; R_A) \cdot Z(j) \cdot Z(k) \tag{20}
\]

In Eqs. (18-20), the summation operator iterates over all MM particles \( N_{MM} \) and QM particles \( N(t)_{QM} \) of type \( t \). The function \( Z(t) \) returns the partial charge of the MM particle. For SPC/E water, we have \( Z(O) = -0.8476 \) and \( Z(H) = 0.4238 \).

This approach requires that the data set is generated within the same QM/MM approach. No partitioning of the QM zone or MM zone is performed. The model \( f_P \) is then trained on the generated data points. The learned potential \( V_P^{QM} + V_P^{QM/MM} \) in combination with the classical force field can subsequently be used to propagate the total system in time.
3 Methods

3.1 Systems

Two types of systems were investigated: (i) single molecules in water, including benzene, uracil and retionic acid, and (ii) transition states of chemical reactions in water, namely the $S_N2$ reaction of $\text{CH}_3\text{Cl}$ with $\text{Cl}^-$ and the reaction of $\text{S-adenosylmethionate}$ (SAM) with cytosine. The systems were chosen to represent different system sizes, use cases, and difficulties. The systems were either treated in a static or dynamic manner. Static refers to a retrospective evaluation of sampled configurations, whereas dynamic indicates that an actual MD simulation was performed using the gradients provided by the ML model.

3.2 General Computational Details

All MD simulations were performed using the GROMOS software package and interfaced to DFTB+/19.2 and ORCA/4.2.0. The basis set and functionals used in this study are the gradient corrected Becke-Perdew functional (BP86), Head-Gordon’s range-separated functional $\omega$B97X-D3, and Grimme’s double hybrid functional B2-PLYP. The basis sets used in this study were Ahlrichs and Weigend’s def2-SVP and def2-TZVP.

All QM calculations used the resolution of identity, Weigend’s auxiliary basis, and Grimme’s dispersion correction with Becke-Johnson damping (except for $\omega$B97X-D3). DFT calculations, which require the computation of a Hartree-Fock reference wave function, were further accelerated using the so-called COSX approximation. The ORCA computations used TightSCF convergence criteria and the integration Grid5. The grid was changed to Grid6 in the final iteration. For the systems retinoic acid and SAM/cytosine, the integration Grid7 was selected to avoid oscillating energies during the computation. Otherwise, standard parameters were used. DFTB computations were performed with Grimme’s D3 dispersion correction with Becke-Johnson damping.

All point charges within the cutoff radius $r$ were included in an electrostatic embedding scheme in the QM computation. Selected solvent atoms beyond the cutoff were included to avoid bond-breaking and creation of artificial charges. The convergence criteria was set to $10^{-6}$ eV for the systems benzene, uracil and $\text{CH}_3\text{Cl}/\text{Cl}^-$, and $10^{-5}$ eV for the others. Structures, which did not converge in the maximum number of steps, were discarded. MD simulations were performed with a convergence criteria of $10^{-5}$ eV and a Broyden mixer with a mixing parameter of $\alpha = 0.3$ to ensure numerical stability.

Newton’s equations of motion were integrated with a time step of 0.5 fs if not noted otherwise. The temperature was kept constant in the MD simulations with the Nosé-Hover chain thermostat and two baths (one coupled to the internal motion and the rotation of the solute, and the other one to the solvent). Berendsen’s barostat was used for constant pressure simulations. Long-range electrostatic interactions beyond the cutoff of 1.4 nm were included using a reaction-field method. Note that the reaction field acts only on the MM particles. The water model used in this study was SPC/E. All bonds between MM particles were constrained using the SHAKE algorithm and a relative tolerance of $10^{-4}$. For technical reasons, a topology file of the QM solute is required as input in GROMOS. These parameters were not used throughout the QM/MM simulation. The topology files were obtained from the ATB server.

The neural networks were implemented using Tensorflow/keras. We exported the computational graphs trained in Python to the C++ GROMOS code.
3.3 Initial Structures

Initial structures of the individual solutes were generated using the ATB server. Initial structures of the transition states of chemical reactions were generated by placing the reactants manually close to the transition state. During sampling, a biasing potential was applied of the form,
\[ V_{\text{bias}} = \frac{1}{2} k (r_{ij} + d \cdot r_{kl} - r_0)^2, \]  
(21)
where \( k \) is the force constant, and \( r_{ij}, r_{kl} \) are the distances between the atoms \( i \) and \( j \), and \( k \) and \( l \), respectively. \( r_0 \) is the ideal distance, and \( d \in \{-1, +1\} \) determines, whether the distances are subtracted or added. For both systems, we set \( d = -1 \) and \( r_0 = 0.00 \). For the \( \text{CH}_3\text{Cl}/\text{Cl}^- \) system, the indexing is shown in the following,
\[ \text{Cl}_j\text{C}_{i,k}\text{H}_3^+ + \text{Cl}^-_l \rightarrow \text{ClCH}_3 + \text{Cl}^- \]  
(22)

The structure was minimized using a force constant for the biasing potential \( k = 60'000 \text{kJ mol}^{-1} \text{nm}^{-2} \).

For the SAM/cytosine system, atom \( i \) was the carbon atom participating in the reaction, atom \( j \) the sulfur atom, atom \( k \) was the same carbon atom as atom \( i \), and atom \( l \) corresponds to the carbon atom in \( \alpha \)-position to the amine group.

For all systems, the solute(s) was solvated in a periodic box of SPC/E water using the GROMOS++ package of programs with a minimum solute-wall distance of 1.4 nm and a minimum solute-solvent distance of 0.3 nm. The size of the simulation boxes were between 3.3 nm and 5.0 nm. All systems were initially relaxed at 0 K using a gradient descent algorithm. A configuration was considered to be converged when the predicted change in energy was less than 0.1 kJ mol\(^{-1}\) averaged over all particles. The thermostat temperature was set to 300 K or 500 K with a coupling constant of 0.1 ps.

3.4 Sampling of Configurations

3.4.1 Small Systems

For the three relatively small systems (solute(s) with low flexibility), a set of configurations was sampled, which was split afterwards into a training/validation sets for the ML model. For benzene in water, a cutoff for the partial charges in the QM Hamiltonian of \( r = 0.6 \text{ nm} \) was used. For uracil in water as well as the \( \text{CH}_3\text{Cl}/\text{Cl}^- \) transition state, the cutoff for the partial charges was set to \( r = 1.4 \text{ nm} \).

Benzene and uracil were simulated with DFTB at a temperature \( T = 500 \text{ K} \) and a pressure \( p = 1 \text{ bar} \) for 10'000 steps. The obtained trajectory was sorted by the QM/MM energy. The lowest representative of each bin was recomputed on a higher level of theory. Every second recomputed point was placed in the validation set, every other point in the training set, resulting in a training/validation split of 50%. The \( \text{CH}_3\text{Cl}/\text{Cl}^- \) system was simulated at \( T = 300 \text{ K}, p = 1 \text{ bar}, k = 20'000 \text{ kJ mol}^{-1} \text{nm}^{-2} \) for 100'000 steps. Every tenth step was recomputed on a higher level of theory. Every second recomputed data point was placed in the validation set, every other point in the training set, resulting in a training/validation split of 50%.

Note that commonly used sampling strategies such as normal-mode sampling are difficult to perform for these systems as they do not account for the solvent environment. In future work, we will explore improved sampling approaches.

3.4.2 Large Systems

To assess the performance of the ML approaches in a real setting, MD production runs were performed for two larger systems: (i) retinoic acid in water (50 QM atoms, 2500 partial charges contributing to
the QM/MM Hamiltonian), and (ii) the SAM/cytosine transition state in water (63 QM atoms, 3000 partial charges contributing to the QM/MM Hamiltonian).

The training set for retinoic acid in water was generated by running DFTB MD simulations at 300 K and a pressure set at 1 bar for 100’000 steps. Every 10th step was recomputed on a higher level of theory (10’000 data points in total). The data set was randomly shuffled, and 50% of the data points were placed in the training set, and the other 50% in the validation set. For the SAM/cytosine system, we performed DFTB MD simulations at 500 K and a pressure set at 1 bar for 105 steps with an integration step of 1.0 fs. The data set was randomly shuffled, and 90% of the data points were placed in the training set.

3.5 MD Simulations Using the Fitted ML Models

For the large systems, (QM)ML/MM MD simulations were performed using the fitted ML models. The time step was set for both system to 0.5 fs, the temperature was kept at $T = 300$ K, the pressure was set to 1 bar.

3.6 Neural Networks

We implemented the HDNNP in Tensorflow/keras with a float64 precision. Each HDNNP used at most two hidden layer and thus followed the recommendation in Ref. 59. The activation function mila was used with a coefficient $\beta = -0.25$, because it is continuously-differentiable and is known to outperform commonly used activation functions such as tanh. The ML models were trained using the Adamax optimizer with varying learning rates. The number of neurons per hidden layer ranged for each NNP from 20 to 160 neurons. As a loss function, we used

$$L = \frac{1}{N} \sum_i (V_i - \tilde{V}_i)^2$$  \hspace{1cm} (23)

and

$$L' = L + \frac{\omega_0}{3N_{QM}} \cdot \sum_i \sum_{\alpha} (-F_{i\alpha} + \tilde{F}_{i\alpha})^2 + \frac{\omega_1}{3N_{MM}} \cdot \sum_i \sum_{\alpha} (-F_{i\alpha} + \tilde{F}_{i\alpha})^2,$$  \hspace{1cm} (24)

where $N_{MM}$ is the number of MM particles in the QM/MM Hamiltonian, $N_{QM}$ the number of QM particles, and $\omega_0$ and $\omega_1$ are weight parameters for the gradient contribution. If not noted otherwise, $\omega_0 = 1$ and $\omega_1 = 1$. Note that $L$ is a non-gradient corrected loss function and $L'$ is a gradient corrected loss function. All models were trained on a NVIDIA K1400 with 3GB VRAM using single batch training. Static models were initially trained for 500 steps with a learning rate of $3.5 \cdot 10^{-4}$. The training was continued for 500 steps if the mean absolute error on the energies of the training set was above 10 kJ mol$^{-1}$. Below 10 kJ mol$^{-1}$, training was continued until the change in the mean absolute error of the energies on the training set was lower than 0.5 kJ mol$^{-1}$ over 100 steps. The learning rate was changed to $3.5 \cdot 10^{-5}$, and training was continued. The training was stopped when the change in the mean absolute error of the energies on the training set was lower than 0.5 kJ mol$^{-1}$ over 50 steps.

In two cases, this training procedure led to an unreasonable error on the training set (M4 and M5 trained on 10% of the training set, see Results and Discussion). For these cases, we manually restarted the training procedure for additional 2000 steps with a learning rate of $3.5 \cdot 10^{-4}$. The dynamic models (for retinoic acid and SAM/cytosine) were trained for 50 steps, each with a learning rate of $3.5 \cdot 10^{-4}$.\"
3.7 Symmetry Functions

The symmetry functions were extracted from Refs. 29,36. In contrast to Smith et al., we did not apply any prefactors. In addition, we allowed for a longer cutoff and extended the spacing of the radial functions. Table 1 lists the symmetry functions, which encode environment. The cutoffs for the radial symmetry functions were varied in this study and are provided in the Results and Discussion section for all models.

| Parameter | Type    | From   | To     | Step size |
|-----------|---------|--------|--------|-----------|
| $\mu$     | Radial  | 0.09   | 1.40   | 0.026875  |
| $\eta$    | Radial  | 160    | 160    | 0         |
| $\mu_A$   | Angular | 0.09   | 0.285  | 0.065     |
| $R_A$     | Angular | 0.35   | 0.35   | 0         |
| $\eta_A$  | Angular | 80     | 80     | 0         |
| $\zeta$   | Angular | 32     | 32     | 0         |
| $\theta_S$| Angular | 0.19634954 | 2.9452431 | 0.19634954 |

Table 1: Symmetry functions used in this work, taken from Ref. 36 and adapted to units nm and nm$^{-1}$. The cutoffs for the radial symmetry functions for the different models are provided in the Results and Discussion section.

For the MM particles, we used either the same symmetry functions as for the QM subsystems (two-body terms and three-body terms), or omitted the three-body terms (only two-body terms). All symmetry functions that include MM particles were weighted with the corresponding partial charge. Note that symmetry functions could be varied to optimize the results. However, Smith et al.\cite{29,36} have shown that their set of symmetry functions covers important local interactions and torsional space in organic compounds, while still being computationally efficient and less prone to overfitting.

In the following, we will refer to three cutoffs: $R_c$ is the cutoff chosen in the QM/MM calculation to generate the training/validation sets. All partial charges within the cutoff were included in the Hamiltonian $\hat{H}^{\text{QM/MM}}$. $R_{\text{sym,QM}}$ is the cutoff used for the radial symmetry functions encoding the QM environment, whereas $R_{\text{sym,MM}}$ is the cutoff used for the radial symmetry functions encoding the MM environment.

4 Results and Discussion

4.1 Assessing the Effect of Different Parameters on the Prediction Accuracy

The loss function of neural networks has multiple local minima. The training procedure usually finds a configuration of weight parameters close to a local minimum, but not necessarily close to the global minimum. To assess the impact of gradients correction, multi-body terms, and the cutoff size, we chose test systems with a conformationally rigid solute in water where the conformational space can be sampled completely. Three different solutes were investigated, which differ in the polarity: (i) benzene (apolar), uracil (polar), and the transition state of CH$_3$Cl/Cl$^-$ (charged). The sampled configurations were split 50/50 in training and validation sets (see Methods).

4.1.1 Gradient Correction and Multi-body Terms

The first test system consists of benzene in water. Benzene is an apolar solute. Thus, long range interactions do not contribute to the PES of the QM/MM region. This means that the training/validation
set could be generated with a cutoff of $R_c = 0.6$ nm. This test system is suitable to study the effect of including gradient correction and multi-body terms (Eqs. (18-20)) in the description of the PES for the MM part. Note that the descriptor always contains multi-body information for the QM particles (Eq. (14)). Table 2 summarizes the settings of four ML models used for the comparison. All models use the parameters proposed by Ref. [29] however, we account for the interactions of the QM subsystem with the MM particles by introducing them as a new element type (see Theory section). Model 1 (M1) is the baseline model, including only two-body information, no gradient correction, and a small cutoff $R_{QM-MM}^{sym,r} = 0.52$ nm. M2 includes both three-body terms for the MM point charges and gradient information, while M3 only includes gradient information. M4 additionally weights the gradients of the MM particles in the loss function, and uses a larger cutoff for the QM-QM symmetry functions ($R_{QM-QM}^{sym,r} = 1.0$ nm).

| Settings ID | Three-body terms for MM? | Gradient correction | Cutoff |
|-------------|--------------------------|---------------------|--------|
|             |                          | $\omega_0$ | $\omega_1$ | $R_{QM-MM}^{sym,r}$ [nm] | $R_{QM-QM}^{sym,r}$ [nm] | $R_c$ [nm] |
| M1          | No                       | 0   | 0       | 0.52  | 0.52  | 0.60  |
| M2          | Yes                      | 1   | 1       | 0.52  | 0.52  | 0.60  |
| M3          | No                       | 1   | 1       | 0.52  | 0.52  | 0.60  |
| M4          | No                       | 1   | 1-200   | 1.00  | 1.00  | 0.60  |

Table 2: Settings for the ML models used to test the influence of including gradient correction and/or three-body terms for the MM point charges. $\omega_0$ and $\omega_1$ are weight parameters for the gradient contribution (Eq. (24)). $R_c$ is the cutoff radius for the MM partial charges in the QM/MM calculation to generate the training/validation, $R_{QM-MM}^{sym,r}$ is the cutoff for the QM-QM symmetry functions, and $R_{QM-QM}^{sym,r}$ the cutoff for the QM-MM symmetry functions. The test system is benzene in water.

Figure 1: Performance of the settings M1-M3 on the validation set of benzene in water as a function of the number of neurons in each hidden layer. The reference data points were computed with B2-PLYP/def2-TZVP. The performance is reported as the mean absolute error (MAE). (Left): MAE of the predicted energies. (Middle): MAE of the predicted QM gradients. (Right): MAE of the predicted MM gradients. The dashed purple line shows the accuracy achieved with the DFTB Hamiltonian on the same data set for comparison. The black dashed line indicates chemical accuracy (i.e. 4.18 kJ mol$^{-1}$ for the energies, and 41.8 kJ mol$^{-1}$ nm$^{-1}$ for the gradients).

We first analyzed the model performance as a function of the number of neurons in the hidden layers for settings M1-M3. As can be seen in Figure 1, the accuracy of the predicted energies (left panel) and QM gradients (middle panel) generally improved with increasing number of neurons per layer. Inclusion of three-body terms for the QM - MM interactions (M2 versus M1) does not improve the accuracy. These terms are expensive to compute and increase the amount of memory drastically.
We will thus not include them in the subsequent models. Introducing a gradient correction (M3 versus M1), on the other hand, improved the prediction accuracy of the QM forces by a factor of three to five (middle panel of Figure 1). This is largely to be expected because information about the gradients is given to the model (the amount of data is multiplied by a factor of three per data point). Nevertheless, also the accuracy of the predicted energies of the validation set is improved, which indicates that the model trained on the energies alone (M1) likely overfitted the PES. The accuracy of the predicted forces of the MM particles improved also slightly when including gradient correction (right panel).

All trained HDNNP models outperform the DFTB Hamiltonian (purple lines) on the predicted energies and the QM gradients (note that DFTB was parametrized on a different level of theory than used in this study). Interestingly, the gradients on the MM particles are predicted more accurately with DFTB than with the ML models. One can ask whether this difference (about 3 kJ mol\(^{-1}\) nm\(^{-1}\)) in accuracy is significant because the absolute deviation of the gradients on the MM particles is low compared to the absolute deviation of the gradients on the QM particles. However, we note that the majority of the particles in the system is affected by this, and therefore the higher deviation is potentially problematic.

This issue can be resolved relatively simply by weighting the contributions of the MM gradients in the loss function with a constant factor as done with settings M4 (Figure 2). In this case, the cutoff of the QM-QM symmetry functions was increased to 1.0 nm to ensure that the descriptor of each atom “sees” all solvent atoms and MM atoms. The M4 variant with \(\omega_1 = 1\) serves thereby as baseline (i.e. increased cutoff but no weighting of the MM gradient contribution). For \(\omega_1 < 200\), the prediction accuracy of the MM gradients is not improved. At \(\omega_1 = 200\), however, the HDNNP finds a different minimum, yielding a better description of the energies, QM gradients, and MM gradients simultaneously. These results indicate that a correct description by the model depends heavily on the regularization in the training procedure. With the improvements, the accuracy of the predicted MM gradients from the ML model outperforms that of the DFTB Hamiltonian.

Figure 2: Performance of the settings M4 on the validation set of benzene in water as a function of the weight \(\omega_1\). The reference data points were computed with B2-PLYP/def2-TZVP. All models used 160 neurons per layer. Performance is reported as the mean absolute error (MAE). (Left): MAE of the predicted energies. (Middle): MAE of the predicted QM gradients. (Right): MAE of the predicted MM gradients. The dashed purple line shows the accuracy achieved with the DFTB Hamiltonian on the same data set for comparison. The black dashed line indicates chemical accuracy (i.e. 4.18 kJ mol\(^{-1}\) for the energies, and 41.8 kJ mol\(^{-1}\) nm\(^{-1}\) for the gradients).

4.1.2 \(\Delta\)-Learning

As ML models use a locality ansatz, it is difficult for them to describe long-range interactions. Semi-empirical methods, on the other hand, are known to be able to account for long-range interactions.
Thus, a $\Delta$-learning scheme based on a semi-empirical method might be a valuable approach. In such a scheme, the ML model introduces a correction to the semi-empirical method to recover the accuracy of the higher-level method. Although this decreases naturally the computational efficiency of the production run, since a additional computation is required at each time step $t$, we will demonstrate in the following that the number of weights in the ML model can be reduced significantly. As a result, the amount of training data points needed is also reduced, the training procedure is accelerated, and the extrapolation capabilities of the model are possibly increased.

The $\Delta$-learning model M5 uses the same settings as M3, i.e. only two-body terms for the QM-MM interactions, gradient correction ($\omega_0=1$, $\omega_1=1$), and a cutoff for the symmetry functions of 0.52 nm. In Figure 3, the performance of M5 is compared to that of the models M3 and M4 (with $\omega_1=200$). Using $\Delta$-learning improved the accuracy of all three properties, i.e. energies, QM gradients, and MM gradients significantly already at a much smaller number of neurons per hidden layer. This indicates that only a reduced number of training data points is necessary to achieve the accuracy of the reference method (see discussion below). It also suggests that the extrapolation capabilities of M5 might be better than those of the previous models. This makes the model particularly interesting for use cases, where the ensemble of conformations/configurations of the system in the training set is not be complete. The $\Delta$-learning scheme will be used in the rest of this study.

Figure 3: Comparison of the performance of M3 and M5 on the validation set of benzene in water as a function of the number of neurons in the hidden layer. The reference data points were computed with B2-PLYP/def2-TZVP. The performance of M4 with $\omega_1=200$ and 160 neurons per layer is shown as red dashed line. The performance is reported as the mean absolute error (MAE). (Left): MAE of the predicted energies. (Middle): MAE of the predicted QM gradients. (Right): MAE of the predicted MM gradients. The dashed purple line shows the accuracy achieved with the DFTB Hamiltonian on the same data set for comparison. The black dashed line indicates chemical accuracy (i.e. 4.18 kJ mol$^{-1}$ for the energies, and 41.8 kJ mol$^{-1}$ nm$^{-1}$ for the gradients).

To the best of our knowledge, the lowest level of theory available for such a $\Delta$-learning scheme is DFTB. Other semi-empirical methods do not incorporate the partial charges in an SCF manner in the Hamiltonian. Rather, they describe the interaction based on parameters. This means that the gradients on the MM particles usually do not correlate with those computed by the reference method (DFT).

4.1.3 Size of the Training Set

A major issue for learning of complex systems is that not all relevant configurations can be enumerated. Thus, ML approaches that can be trained on a small data set are especially valuable. As shown above, $\Delta$-learning requires less weight parameters to achieve chemical accuracy on the validation set.
of benzene in water, which indicates that less training points are needed compared to the ML models trained directly on the full energies and gradients. To quantify this observation, the M4 ($\omega_1 = 200$, 160 neurons per layer) and M5 ($\omega_1 = 1$, 20 neurons per layer) models were trained on only 10% of the original training set. This was done by selecting every tenth data point of the training set, which corresponds to 211 data points in total. The validation set remained the same. The results are presented in Table 3. While the MAE on the energies of the validation set is 6.0 kJmol$^{-1}$ for M5, the error increases to 9.5 kJmol$^{-1}$ for M4. Furthermore, we observe a much larger drop in accuracy on the MM gradients for M4 than for M5 when trained with only 10% of the data. The $\Delta$-learning model is thus more robust towards smaller training set sizes. However, we note that the description of the MM particles becomes slightly worse compared to DFTB.

| Model | Energy MAE [kJmol$^{-1}$] | Energy RMSE [kJmol$^{-1}$] | QM gradients MAE [kJmol$^{-1}$ nm$^{-1}$] | QM gradients RMSE [kJmol$^{-1}$ nm$^{-1}$] | MM gradients MAE [kJmol$^{-1}$ nm$^{-1}$] | MM gradients RMSE [kJmol$^{-1}$ nm$^{-1}$] |
|-------|---------------------------|-----------------------------|------------------------------------------|------------------------------------------|------------------------------------------|------------------------------------------|
| M4 (100%) | 2.61 (2.36) | 3.40 (3.04) | 41.40 (38.90) | 57.80 (52.70) | 5.75 (5.52) | 10.29 (9.84) |
| M4 (10%) | 9.48 (8.95) | 11.86 (11.72) | 88.70 (79.57) | 124.43 (102.99) | 16.04 (20.21) | 27.11 (36.22) |
| M5 (100%) | 2.80 (2.67) | 3.58 (3.41) | 26.68 (26.36) | 35.66 (34.69) | 5.25 (5.22) | 9.77 (9.83) |
| M5 (10%) | 6.00 (5.30) | 7.26 (6.27) | 38.88 (38.09) | 50.93 (49.72) | 7.59 (8.64) | 16.40 (14.44) |
| DFTB | 13.54 (13.83) | 14.75 (15.10) | 138.80 (138.70) | 189.40 (190.60) | 7.34 (7.31) | 13.70 (13.70) |

Table 3: Comparison of the performance of M4 and M5 on the validation set of benzene in water when trained on 10% or 100% of the training set. Performance is reported as the mean absolute error (MAE) and root-mean-square-error (RMSE). The value for the training set is given in brackets. The reference data points were computed with B2-PLYP/def2-TZVP. The performance of DFTB is given as a reference. After the standard training protocol described in the method section, M4 (10%) and M5 (10%) were trained for 2000 more steps with a learning rate of $3.5 \cdot 10^{-4}$.

### 4.1.4 Cutoff Size

Fitting a ML model for the benzene in water system did not require the inclusion of long-range interactions. To assess the effect of the cutoff size and thus the improved accounting for long-range interactions, the test system of uracil in water is used in the following. The cutoff $R_c$ is increased up to 1.40 nm, which corresponds to 2000-3000 MM partial charges in the cutoff sphere. A cutoff of 1.4 nm is used in the GROMOS force field, of which the SPC/E water model is part of. Table 4 lists the settings M6-M8. All models use gradient correction and the $\Delta$-learning scheme with DFTB. M6 serves as a “worst-case” model, where the MM particles are simulated solely with the semi-empirical method and the ML model introduces just a local correction to the QM subsystem.

| Settings ID | $R_{QM-MM}^{sym,r}$ [nm] | $R_{QM-QM}^{sym,r}$ [nm] | $R_c$ [nm] | $\omega_1$ |
|-------------|---------------------------|---------------------------|-----------|-----------|
| M6          | 0.52                      | 0.00                      | 1.40      | 1         |
| M7          | 0.52                      | 0.52                      | 1.40      | 1         |
| M8          | 1.40                      | 1.40                      | 1.40      | 1         |

Table 4: Settings for the $\Delta$-learning ML models used to test the influence of the cutoff size. All models employ only two-body terms for the QM - MM interactions and gradient correction. The models were trained with 20 neurons per hidden layer. $R_c$ is the cutoff radius for the MM partial charges in the QM/MM calculation, $R_{QM-MM}^{sym,r}$ is the cutoff for the QM-QM symmetry functions, and $R_{QM-MM}^{sym,r}$ the cutoff for the QM-MM symmetry functions. The test system is uracil in water.

The results with the models M6-M8 and the DFTB baseline on the validation set of uracil in water
are summarized in Table 5. Model M6 with no information about the MM environment (cutoff QM-MM set to zero) performs worst but still better than the DFTB baseline. Including the MM particles in the QM calculation (M7 versus M6) improved the accuracy on all three properties. Increasing the cutoff from 0.52 nm to 1.40 nm (M8 versus M7) reduced the error on the energies and QM gradients further, however, the error on the MM gradients increased indicating overfitting. This behaviour may be explained with the insensitivity of the loss function towards the comparatively small gradients of the MM particles (Eq. 24). The ML model can thus use these input features to achieve a better performance measured by the loss function. This is achieved by improving the accuracy of the total energies and the QM gradients. However, the trade off is that the prediction accuracy of the MM gradients becomes worse. We already demonstrated for the benzene test system that increasing the weights of the MM gradients in the loss function can resolve this issue partially. Here, we found that using a lower cutoff for the symmetry functions ($R_{QM-MM}^{sym} = 0.52$ nm, M7) regularizes the model similarly.

| Model | Energy | QM gradients | MM gradients |
|-------|--------|---------------|--------------|
|       | MAE    | RMSE          | MAE          | RMSE         | MAE          | RMSE         |
|       | [kJ mol$^{-1}$] | [kJ mol$^{-1}$] | [kJ mol$^{-1}$ nm$^{-1}$] | [kJ mol$^{-1}$ nm$^{-1}$] | [kJ mol$^{-1}$ nm$^{-1}$] | [kJ mol$^{-1}$ nm$^{-1}$] |
| M6    | 4.03 (3.94) | 5.25 (5.14) | 54.20 (54.20) | 74.00 (74.10) | 1.35 (1.35) | 5.27 (5.30) |
| M7    | 2.06 (2.03) | 2.53 (2.49) | 37.90 (37.50) | 50.86 (50.21) | 1.19 (1.20) | 3.81 (3.80) |
| M8    | 1.20 (1.14) | 1.45 (1.36) | 28.53 (26.85) | 37.48 (35.18) | 1.76 (1.76) | 4.27 (4.26) |
| DFTB  | 6.55 (6.56) | 8.29 (8.27) | 311.70 (312.80) | 429.10 (429.80) | 1.35 (1.35) | 5.27 (5.27) |

Table 5: Performance of the models M6-M8 and the DFTB baseline on the validation set of uracil in water. Performance is reported as the mean absolute error (MAE) and root-mean-square-error (RMSE). The value for the training set is given in brackets. The training set was computed with BP86/def2-TZVP.

The results for the benzene and uracil test systems show that constructing an ML model, which handles long-range interactions correctly, is challenging, and that the ML model requires regularization to avoid overfitting. The model can be regularized by increasing the weights of the MM gradients or by decreasing the cutoff in combination with $\Delta$-learning. However, so far the test systems were not charged. For charged systems, incorporation of long-range electrostatic interactions will be even more important.

To assess the transferability of the conclusions from the previous test systems, we analyzed the performance of the ML models on a third test system, the transition state of the $S_N2$ reaction between $CH_3Cl$ and $Cl^-$. For this system, larger changes in the electronic structure within the configurational ensemble are expected. Table 6 lists the settings investigated for the $CH_3Cl/Cl^-$ test system.
Settings ID | \( {R}^{\text{QM-QM}}_{\text{sym},r} \) [nm] | \( {R}^{\text{QM-MM}}_{\text{sym},r} \) [nm] | \( R_c \) [nm] | Gradient correction
--- | --- | --- | --- | ---
M9 | 0.52 | 0.00 | 1.40 | 1
M10 | 0.52 | 0.52 | 1.40 | 1
M11 | 1.40 | 1.40 | 1.40 | 1
M12 | 1.40 | 1.40 | 1.40 | 200

Table 6: Settings for the \( \Delta \)-learning ML models used to test the influence of the cutoff size. All models employ only two-body terms for the QM - MM interactions and gradient correction. The models were trained with 20 neurons per hidden layer. The test system is the transition state of \( \text{CH}_3\text{Cl}/\text{Cl}^- \) in water. \( R_c \) is the cutoff radius for the MM partial charges in the QM/MM calculation, \( {R}^{\text{QM-QM}}_{\text{sym},r} \) is the cutoff for the QM-QM symmetry functions, and \( {R}^{\text{QM-MM}}_{\text{sym},r} \) the cutoff for the QM-MM symmetry functions.

The results with the models M9-M12 and the DFTB baseline on the validation set of the transition state of \( \text{CH}_3\text{Cl}/\text{Cl}^- \) in water are summarized in Table 7. For model M9-M11 we see in general the same behaviour as in the previous test system with uracil in water. Not including the MM partial charges in the ML model as in model M9 results again in the worst performance, far from chemical accuracy. Including them with a cutoff of 0.52 nm (M10) reduces the error on the energies and QM gradients significantly, and increasing the cutoff to 1.4 nm (M11) improved the performance further. The effect of the increased cutoff is more dramatic for the \( \text{CH}_3\text{Cl}/\text{Cl}^- \) test system compared to uracil in water due to the stronger long-range electrostatic interactions in the charged system. Furthermore, DFTB describes the training and validation set much worse than it was the case for the uracil test system, i.e. the correction by the ML model is larger. However, as observed above, the prediction accuracy of the gradients on the MM particles decreased with increasing cutoff for the QM - MM interactions. Thus, we tested model M12, where the weights of the MM partial charges in the loss function were increased (\( \omega_1 = 200 \)). This improved the performance on the MM gradients, but the accuracy of the DFTB baseline was not reached. This indicates that stronger regularization of the ML model during the training procedure is necessary.

| Model | Energy MAE [kJ mol\(^{-1}\)] | Energy RMSE [kJ mol\(^{-1}\)] | QM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | QM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] |
|---|---|---|---|---|---|---|
| M9 | 10.06 (9.53) | 12.51 (11.84) | 129.01 (127.05) | 196.76 (167.7) | 2.10 (2.10) | 16.40 (16.40) |
| M10 | 2.35 (2.20) | 3.05 (2.86) | 41.70 (40.50) | 56.10 (54.40) | 3.26 (3.30) | 23.00 (22.80) |
| M11 | 0.86 (0.79) | 1.14 (1.04) | 24.80 (22.90) | 32.90 (29.90) | 4.70 (4.67) | 23.40 (23.10) |
| M12 | 1.21 (1.10) | 1.60 (1.44) | 28.20 (26.60) | 37.40 (35.20) | 3.91 (3.89) | 22.92 (22.65) |
| DFTB | 17.91 (17.84) | 222.10 (221.00) | 202.60 (201.70) | 302.10 (300.70) | 2.10 (2.10) | 16.40 (16.40) |

Table 7: Performance of the models M9-M12 and the DFTB baseline on the validation set of the transition state of \( \text{CH}_3\text{Cl}/\text{Cl}^- \) in water. Performance is reported as the mean absolute error (MAE) and root-mean-square-error (RMSE). The value for the training set is given in brackets. The training set was computed with Head-Gordon’s range separated functional \( \omega B97X-D3 \).

From the findings discussed above, we can conclude that the choice of the cutoff size depends on the polarity of the solute (i.e. how important long-range electrostatic interactions are). In general, a larger cutoff is recommended for polar solutes in polar solvents, however, it increases also the risk of overfitting. This can be accounted for by increasing the weights of the MM partial charges in the loss function.
4.2 Application of ML Models in MD Simulations

So far we have assessed the performance of the ML models in terms of MAE (and RMSE) on a validation set. However, these results can be to some degree misleading because rare outliers get averaged. Such rare outliers might be less tolerable in the context of an actual MD simulations, where the results of the next step depend directly on the results of the previous step. MD simulations require thus the inclusion of relatively high-energy configurations in the training set, such that repulsion and attraction can be learned. In order to judge the usefulness of HDNNP models in practice, it is important to test their performance in actual MD simulations. For this purpose, we have chosen two test systems with larger conformational flexibility: (i) retinoic acid (a vitamin A derivative) in water, and (ii) the transition state of the reaction of SAM with cytosine in water (Figure 4). Sampling of the training set was performed in two different ways. For the SAM/cytosine system, sampling was performed at elevated temperature (500K) with a time step of 1fs. For the retinoic acid system, this strategy resulted in an unstable trajectory. Thus, sampling was performed at 300K with a time step of 0.5fs (see Methods for more details).

The ML models investigated employ the $\Delta$-learning scheme with DFTB as baseline, and use 20 neurons per hidden layer, only two-body terms for the QM - MM interactions, gradient correction (with $\omega_0 = 1$ and $\omega_1 = 200$), and a cutoff of 1.4nm.

![Figure 4: Two larger test systems used for MD simulations. (Left): retinoic acid in water (50 QM atoms and about 2500 MM partial charges with a cutoff of 1.4nm). (Right): transition state of the chemical reaction between SAM and cytosine in water (63 QM atoms and about 3000 MM partial charges with a cutoff of 1.4nm).](image)

4.2.1 Retinoic Acid in Water

First, we assessed the performance of the ML model on a validation set, as done in the previous sections. Table 8 summarizes the results in terms of MAE and RMSE, Figure 5 shows them graphically. Using $\Delta$-learning model, the performance is improved compared to the DFTB baseline for all three properties, including the MM gradients.

| Model | Energy MAE [kJ mol\(^{-1}\)] | Energy RMSE [kJ mol\(^{-1}\)] | QM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | QM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| ML    | 4.07 (4.10)     | 5.16 (5.20)     | 41.25 (42.22)   | 59.64 (106.28)  | 1.30 (1.30)     | 3.54 (3.55)     |
| DFTB  | 17.91 (17.84)   | 222.10 (221.00)| 202.60 (201.70)| 302.10 (300.70)| 2.10 (2.10)     | 16.40 (16.40)   |

Table 8: Performance of the $\Delta$-learning model and the DFTB baseline on the validation set of retinoic acid in water. Performance is reported as the mean absolute error (MAE) and root-mean-square-error (RMSE). The value for the training set is given in brackets. The data points were computed on the BP86/def2-TZVP level of theory.
Next, we tested the trained ML model by performing an (QM)ML/MM MD simulation for 5000 steps (top panel in Figure 6). In order to compare the ML-corrected energies with those of the DFT reference and the DFTB baseline simulation, single point calculations were performed for each configuration in the ML + DFTB trajectory. As shown in Figure 6, the energies of ML + DFTB agree better with the DFT reference than the DFTB baseline. Although the latter performs relatively well, one should keep in mind that the deviations from the reference method accumulate along an MD trajectory. Thus, even relatively small deviations might result in largely different trajectories when starting from the same coordinates. To assess the stability of the ML model, we extended the simulation to 200’000 steps (bottom panel in Figure 6).
Figure 6: MD simulation of retinoic acid in water (integration size 0.5 fs). (Top): Energy trajectory from 5000 consecutive steps performed by the ML + DFTB model (green). In each step, the energies were also computed with DFTB (blue) and the DFT reference (BP86/def2-TZVP, black) for comparison. The energy deviation from the reference over the complete trajectory with respect to the minimum energy was: MAE = 9.18 kJ mol$^{-1}$ and RMSE = 11.28 kJ mol$^{-1}$ for the ML + DFTB model, and MAE = 20.78 kJ mol$^{-1}$ and RMSE = 22.77 kJ mol$^{-1}$ for DFTB. (Bottom): Energy trajectory from 200'000 consecutive steps performed by the ML + DFTB model (green).

4.2.2 SAM/Cytosine Transition State in Water

In biochemistry, S-adenosylmethionate (SAM) is a co-factor for the transfer of a methyl group by enzymes (methyltransferases). Here, we investigated the transition state of the chemical reaction between SAM and cytosine. Again, we first assessed the performance on a validation set as done above (Table 9 and Figure 7). The $\Delta$-learning model clearly outperforms the DFTB baseline. However, the difference between the prediction accuracy of the energies in the training set and the validation set indicates that the ML model overfitted the energies. Nevertheless, the performance on the gradients (both QM and MM) looks reasonable. Note that the training set contains a much larger number of high-energy conformations than in the retinoic acid system due to the sampling strategy used ($T = 500\, \text{K}$).
Figure 7: Comparison between the reference DFT method and the DFTB Hamiltonian (blue) and the ML + DFTB model (green) on the validation set of the transition state of SAM/cytosine in water. (Left): Energies. (Right): QM gradients.

| Model | Energy MAE [kJ mol\(^{-1}\)] | Energy RMSE [kJ mol\(^{-1}\)] | QM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | QM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients MAE [kJ mol\(^{-1}\) nm\(^{-1}\)] | MM gradients RMSE [kJ mol\(^{-1}\) nm\(^{-1}\)] |
|-------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| ML    | 15.19 (4.55)    | 19.06 (5.76)    | 85.13 (75.44)   | 116.51 (100.98) | 2.03 (2.15)     | 5.25 (5.26)     |
| DFTB  | 32.00 (26.64)   | 32.50 (39.17)   | 313.50 (314.20) | 418.70 (420.20) | 2.43 (2.42)     | 8.24 (8.24)     |

Table 9: Performance of the ∆-learning model and the DFTB baseline on the validation set of the transition state of SAM and cytosine in water. Performance is reported as the mean absolute error (MAE) and root-mean-square-error (RMSE). The value for the training set is given in brackets. The training set was computed with BP86/def2-SVP.

Next, we performed a (QM)ML/MM MD simulation for 3000 steps using the fitted model (top panel in Figure 8). As for the test system with retinoic acid in water, the energies with ML + DFTB agreed well with the DFT reference, and outperformed the DFTB baseline as indicated by the MAE and RMSE over the trajectory. To assess the stability of the model, we extended the simulation to 50’000 steps (bottom panel in Figure 8). For both test systems, it was possible to carry out a stable (QM)ML/MM MD simulation for the selected number of steps (up to 200’000 steps). Of course, the possibility cannot be excluded at this point that a configuration, which is not well represented by the ML model, might be encountered in even longer simulations.
4.3 General Discussion

The $\Delta$-learning scheme requires an explicit computation with the lower-level method at each time step. In this work, we have used DFTB as the lower-level method. For the SAM/cytosine system with 63 QM atoms and up to 3000 MM particles in the cutoff sphere, one evaluation with DFTB required less than a second while the corresponding DFT computation took 10 to 30 minutes more than three magnitudes longer on the same hardware. Using a better basis set, such as def2-TZVP, increased the computing time to 60 minutes. Thus, the additional time needed for the DFTB calculation at each step of the $\Delta$-learning scheme is less expensive than e.g. the pair-list creation in the MD engine.

An obvious limitation of the $\Delta$-learning approach are configurations, for which the DFTB PES is substantially different from the reference DFT PES. In such cases, the DFTB computation will converge very slowly (or not at all), limiting the usage. However, we did not encounter such an issue in MD simulations.

5 Summary and Conclusion

In this work, we investigated the use of HDNNP in (QM)ML/MM MD simulations of condensed-phase systems with DFT (or ab initio) accuracy for the QM subsystem. Standard QM/MM MD simulations...
at this level of accuracy are very expensive and only applicable to small systems. Using semi-empirical methods to describe the QM parts is much faster but also less accurate. In the HDNNP, the MM partial charges up to a certain cutoff are integrated as additional element type.

We assessed the influence of different parameters on the prediction accuracy of energies, gradients of the QM particles, and gradients of the MM particles. First, we used a simple test system of an apolar, rigid solute in water. Three-body terms to describe the QM - MM interactions could be neglected, however, the inclusion of a gradient correction in the training procedure of the ML model was crucial to obtain sufficient accuracy on the gradients. In general, we found that classical HDNNP require a large number of weight parameters to fit the PES of even small condensed-phase systems (e.g. benzene in water) accurately. Furthermore, strong regularization on the MM gradients was necessary during the training procedure to predict these gradients with decent accuracy. This is possibly due to the locality ansatz, which makes it difficult for the HDNNP to describe long-range interactions well. As an alternative, we employed a ∆-learning scheme, where the lower-level method is DFTB. Semi-empirical methods are known to handle the long-range interactions well. Using the ∆-learning model allowed us to decrease the number of weight parameters while retaining a high accuracy. Smaller models usually require less training data points and display improved extrapolation capabilities. This may be a key factor for large biomolecular systems, where it is not feasible to enumerate all conformations/configurations for the training set.

For solutes with increased polarity (e.g. uracil, and the transition state of CH$_3$Cl/Cl$^-$ in water), the long-range electrostatic interactions become more important and thus the size of the cutoff $R_c$. Increasing the cutoff, which was necessary to better describe long-range interactions, improved the prediction accuracy of the energy and QM gradients. However, at the same time the performance on the MM gradients was reduced. A possible explanation is that the loss function in the training procedure is not sensitive to the relatively small gradients on the MM particles (compared to the QM gradients), and thus the ML model can use these input features to achieve a better performance on the energies and the QM gradients at the expense of the MM gradients (i.e. overfitting). This issue can be resolved to some degree by increasing the weights of the MM particles in the loss function.

The final ∆-learning model was tested further by performing actual (QM)ML/MM MD simulations of relatively large systems, i.e. retinoic acid in water and the transition state of the chemical reaction between SAM and cytosine. Stable trajectories were obtained with an accuracy close to the DFT reference. We envision that the results and findings presented in this study will enable the use of (QM)ML/MM MD simulations in practical applications.

Acknowledgements

The authors thank Patrick Bleiziffer and Annick Renevey for helpful discussions. S.R. gratefully acknowledges financial support by the Swiss National Science Foundation (Grant Number 200021-178762) and by ETH Zurich (ETH-34 17-2).

References

[1] S. Riniker. Fixed-Charge Atomistic Force Fields for Molecular Dynamics Simulations in the Condensed Phase: An Overview. J. Chem. Inf. Model, 58, 565–578, 2018.

[2] P. S. Nerenberg, T. Head-Gordon. New Developments in Force Fields for Biomolecular Simulations. Curr. Opin. Struct. Biol., 49, 129–138, 2018.

[3] N. Schmid, A. P. Eichenberger, A. Choutko, S. Riniker, M. Wigner, A. E. Mark, W. F. van
Gunsteren. Definition and Testing of the GROMOS Force-field Versions 54A7 and 54B7. *Eur. Biophys. J.*, 40(843), 2011.

[4] J. W. Ponder, D. A. Case. Force Fields for Protein Simulations. *Adv. Prot. Chem.*, 66, 27–85, 2003.

[5] A. Warshel, M. Levitt. Theoretical Studies of Enzymatic Reactions: Dielectric, Electrostatic and Steric Stabilization of the Carbonium Ion in the Reaction of Lysozyme. *J. Mol. Biol.*, 103, 227–249, 1976.

[6] A. J. Mulholland, P. D. Lyne, M. Karplus. Ab Initio QM/MM Study of the Citrate Synthase Mechanism. A Low-Barrier Hydrogen Bond Is not Involved. *J. Am. Chem. Soc.*, 122, 534–535, 2000.

[7] M. Senn, W. Thiel. QM/MM Methods for Biomolecular Systems. *Angew. Chem. Int. Ed.*, 48, 1198–1229, 2009.

[8] G. Groenhof. Introduction to QM/MM simulations. *J. Mol. Biol.*, 924, 43–66, 2013.

[9] H. Lin, D. G. Truhlar. QM/MM: What Have We Learned, Where Are We, and Where Do We Go From Here? *Theor. Chem. Acc.*, 117, 185–199, 2007.

[10] D. Loco, O. L. Lagardère, S. Caprasecca, F. L. Orcid, B. M. Orcid, J.-P. Piquemal. Hybrid QM/MM Molecular Dynamics with AMOeba Polarizable Embedding. *J. Chem. Theory Comput.*, 13, 4025–4033, 2017.

[11] C. Panosetti, A. Engelmann, L. Nemec, K. Reuter, J. T. Margraf. Learning to Use the Force: Fitting Repulsive Potentials in Density-Functional Tight-Binding with Gaussian Process Regression. *J. Chem. Theory Comput.*, 16, 2181–2191, 2020.

[12] E. Brunk, U. Röthlisberger. Mixed Quantum Mechanical/Molecular Mechanical Molecular Dynamics Simulations of Biological Systems in Ground and Electronically Excited States. *Chem. Rev.*, 115, 6217–6263, 2015.

[13] L. W. Chung, W. M. C. Sammera, R. Ramozzi, A. J. Page, M. Hatanaka, G. P. Petrova, T. V. Harris, X. Li, Z. Ke, F. Liu, et al. The ONIOM Method and Its Applications. *Chem. Rev.*, 115, 5678–5796, 2015.

[14] F. Neese. Software Update: The ORCA Program System, Version 4.0. *Comput. Mol. Sci.*, 8, e1327, 2018.

[15] S. G. Balasubramani, G. Chen, S. Coriani, M. Diedenhofen, M. Frank, Y. Franzke, F. Furche, R. Grotjahn, M. Harding, C. Hättig, et al. TURBOMOLE: Modular Program Suite for Ab Initio Quantum-chemical and Condensed-matter Simulations. *J. Chem. Phys.*, 152, 185–199, 2020.

[16] A. S. Christensen, T. Kubar, Q. Cui, M. Elstner. Semiempirical Quantum Mechanical Methods for Noncovalent Interactions for Chemical and Biochemical Applications. *Chem. Rev.*, 116, 5301–5337, 2015.

[17] J. J. P. Stewart. Optimization of Parameters for Semiempirical Methods VI: More Modifications to the NDDO Approximations and Re-optimization of Parameters. *J. Molec. Modeling*, 19, 1–32, 2013.
[18] J. Behler. Atom-centered Symmetry Functions for Constructing High-dimensional Neural Network Potentials. *J. Chem. Phys.*, 134, 074106, 2011.

[19] B. Jiang, H. Guo. Permutation Invariant Polynomial Neural Network Approach to Fitting Potential Energy Surfaces. *J. Chem. Phys.*, 139, 054112, 2013.

[20] J. Li, B. Jiang, H. Guo. Permutation Invariant Polynomial Neural Network Approach to Fitting Potential Energy Surfaces. II. Four-atom Systems. *J. Chem. Phys.*, 139, 204103, 2013.

[21] J. Behler. Representing Potential Energy Surfaces by High-dimensional Neural Network Potentials. *J. Phys.: Condens. Matter*, 26, 183001, 2014.

[22] J. Behler. Constructing High-dimensional Neural Network Potentials: A Tutorial Review. *Quant. Chem.*, 115, 1032–1050, 2015.

[23] V. Botu, R. Ramprasad. Adaptive Machine Learning Framework to Accelerate Ab Initio Molecular Dynamics. *Quant. Chem.*, 115, 1074–1083, 2015.

[24] R. Ramakrishnan, P. O. Dral, M. Rupp, O. A. von Lilienfeld. Big Data Meets Quantum Chemistry Approximations: The $\Delta$-Machine Learning Approach. *J. Chem. Theory Comput.*, 11, 2087–2096, 2015.

[25] M. Gastegger, J. Behler, P. Marquetand. Machine Learning Molecular dynamics for the simulation of infrared spectra. *Chem. Sci.*, 8, 6924–6935, 2017.

[26] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, P. Marquetanda. wACSF – Weighted Atom-centered Symmetry Functions as Descriptors in Machine Learning Potentials. *J. Chem. Phys.*, 148, 421709, 2018.

[27] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K.-R. Müller. Machine Learning of Accurate Energy-conserving Molecular Force Fields. *Science Adv.*, 3, e1603015, 2017.

[28] S. Chmiela, H. E. Sauceda, K. R. Mueller, A. Tkatchenko. Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields. *Nature Communications*, 9, 3887, 2018.

[29] J. S. Smith, O. Isayev, A. E. Roitberg. ANI-1: An Extensible Neural Network Potential with DFT Accuracy at Force Field Computational Cost. *Chem. Sci.*, 8, 3192–3203, 2017.

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

[31] G. Montavon, K. Hansen, S. Fazli, M. Rupp, F. Biegler, A. Ziehe, A. Tkatchenko, A. V. Lilienfeld, K.-R. Müller. In F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25*, 440–448. Curran Associates, Inc., 2012.

[32] C. Brunken, M. Reiher. Self-Parametrizing System-Focused Atomistic Models. *J. Chem. Theory Comput.*, 16, 1646–1665, 2020.

[33] A. Fabrizio, A. Grisafi, B. Meyer, M. Ceriotti, C. Corminboeuf. Electron Density Learning of Non-covalent Systems. *Chem. Sci.*, 110, 9424–9432, 2019.

[34] K. T. Schütt, P.-J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, K.-R. Müller. SchNet: A Continuous-filter Convolutional Neural Network for Modeling Quantum Interactions, 2017.
[35] A. Sannai, Y. Takai, M. Cordonnier. Universal Approximations of Permutation Invariant/Equivariant Functions by Deep Neural Networks, 2020. URL https://openreview.net/forum?id=HkeZQJBKDB.

[36] J. Stevenson, L. D. Jacobson, Y. Zhao, C. Wu, J. Maple, K. Leswing, E. Harder, R. Abel. Schrodinger-ANI: An Eight-Element Neural Network Interaction Potential with Greatly Expanded Coverage of Druglike Chemical Space, 2019.

[37] S. N. Pozdnyakov, M. J. Willatt, A. P. Bart, C. Ortner, G. Csan, M. Ceriotti. On the Completeness of Atomic Structure Representations.

[38] L. Shen, J. Wu, W. Yang. Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks. J. Chem. Theory Comput., 35, 479–506, 2016.

[39] L. Shen, W. Yang. Molecular Dynamics Simulations with Quantum Mechanics/Molecular Mechanics and Adaptive Neural Networks. J. Chem. Theory Comput., 14, 3379–3390, 2018.

[40] Y. Zhang, A. Khorshidi, G. Kastlunger, A. A. Peterson. The Potential for Machine Learning in Hybrid QM/MM Calculations. J. Chem. Phys., 148, 241740, 2018.

[41] A. Grisafi, M. Ceriotti. Incorporating Long-range Physics in Atomic-scale Machine Learning. J. Chem. Phys., 151, 204105, 2019.

[42] H. Junming, Y. Shao, J. Kato. Do Better Quality Embedding Potentials Accelerate the Convergence of QM/MM Models? The Case of Solvated Acid Clusters. Molecules, 26, 2466, 2018.

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

[44] A. Gordon, E. Eban, O. Nachum, B. Chen, H. Wu, T.-J. Yang, E. Choi. MorphNet: Fast and Simple Resource-Constrained Structure Learning of Deep Networks.

[45] M. Reiher, A. Wolf. Relativistic Quantum Chemistry. Wiley-VCH, 2014.

[46] M. Born, R. Oppenheimer. Zur Quantentheorie der Molekeln. Annalen der Physik, 389, 457–484, 1927.

[47] I. Newton. Axioms or Laws of Motion. Mathematical Principles of Natural Philosophy. 1, 1729.

[48] S. K. Gray. Symplectic Integrators for Large Scale Molecular Dynamics Simulations: A Comparison of Several Explicit Methods. J. Chem. Phys., 101, 4062, 1994.

[49] E. Schrödinger. Quantisierung als Eigenwertproblem. Annalen der Physik, 385, 437–490, 1926.

[50] I. N. Levine. Quantum Chemistry. Pearson, 2013.

[51] W. Kohn, L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. Phys. Rev., 140, 1133–1138, 1965.

[52] B. Hourahine, B. Aradi, V. Blum, F. Bonača, A. Bučkeri, C. Chamacho, C. Cevallos, M. Y. Deshaye, T. Dumițrica, A. Domínguez, et al. DFTB+, a Software Package for Efficient Approximate Density Functional Theory Based Atomistic Simulations Featured. J. Chem. Phys., 152, 124101, 2020.
Toward a Practical Method for Adaptive QM/MM Simulations. *J. Chem. Theory Comput.*, 5, 2212–2221, 2009.

A Charge-scaling Method to Treat Solvent in QM/MM Simulations. *Theo. Chem. Acc.*, 109, 118–124, 2003.

An Efficient Real Space Multigrid QM/MM Electrostatic Coupling. *J. Chem. Theory Comput.*, 1, 1176–1184, 2005.

Towards Machine Learning. MIT Press, 2014.

A Review of Applications. *Expert Systems with Applications*, 36, 2–17, 2009.

Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences.

Self-Normalizing Neural Networks.

Deep Sets, 2017.

A Critical Comparison of Neural Network Potentials for Molecular Reaction Dynamics with Exact Permutation Symmetry. *Phys. Chem. Chem. Phys.*, 21, 9672–9682, 2019.

Architecture, implementation and parallelization of the GROMOS software for biomolecular simulation. *Comp. Phys. Comm.*, 183, 890–903, 2012.

Interfacing the GROMOS (Bio)molecular Simulation Software to Quantum-Chemical Program Packages. *J. Comput. Chem.*, 33, 2108–2177, 2012.

Long-range Corrected Hybrid Density Functionals with Damped Atom–Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.*, 44, 501–508, 2001.

Semiempirical Hybrid Density Functional with Perturbative Second-order Correlation. *J. Chem. Phys.*, 124, 034108, 2006.

Balanced Basis Sets of Split Valence, Triple Zeta Valence and Quadruple Zeta Valence Quality for H to Rn: Design and Assessment of Accuracy. *Phys. Chem. Chem. Phys.*, 7, 3297–3305, 2005.

Use of Approximate Integrals in Ab Initio Theory. An Application in MP2 Energy Calculations. *Chem. Phys. Lett.*, 208, 3590363, 1993.

Accurate Coulomb-fitting Basis Sets for H to Rn. *Phys. Chem. Chem. Phys.*, 9, 1057–1065, 2006.

A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.*, 132, 154104, 2010.

Effect of the Damping Function in Dispersion Corrected Density Functional Theory. *J. Comput. Chem.*, 32, 1456–1465, 2011.
[71] D. R. Hartree. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part II. Some Results and Discussion. *Mathematical Proceedings of the Cambridge Philosophical Society*, 24, 111–132, 1928.

[72] S. Kossmann, F. Neese. Comparison of Two Efficient Approximate Hartee-Fock Approaches. *Chem. Phys. Lett.*., 481, 240–243, 2009.

[73] C. G. Broyden. A Class of Methods for Solving Nonlinear Simultaneous Equations. *Math. Computation*, 19, 577–593, 1965.

[74] S. Nose. A Unified Formulation of the Constant Temperature Molecular Dynamics Methods. *J. Chem. Phys.*, 81, 511, 1984.

[75] W. G. Hoover. Canonical Dynamics: Equilibrium Phase-space Distributions. *Phys. Rev. A (General Physics)*, 31, 1695–1697, 1985.

[76] H. J. C. Berendsen, J. R. Grigera, T. P. Straatsma. The Missing Term in Effective Pair Potentials. *J. Phys. Chem.*, 91, 6269–6271, 1987.

[77] H. C. Berendsen, W. F. van Gunsteren. Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perspectives in Chemistry. *Angew. Chem. Int. Ed.*, 29, 992–1023, 1990.

[78] H. J. C. Berendsen, J. R. Grigera, T. P. Straatsma. The missing term in effective pair potentials. *J. Phys. Chem.*, 91, 6269–6271, 1987.

[79] V. Kräuter, W. van Gunsteren, P. Hünenberger. A fast SHAKE algorithm to solve distance constraint equations for small molecules in molecular dynamics simulations. *J Chem Comp*, 22, 501–508, 2001. doi:10.1002/1096-987X.

[80] A. Malde, L. Zuo, M. Breeze, M. Stroet, D. Poger, P. Nair, C. Oostenbrink, A. Mark. An Automated Force Field Topology Builder (ATB) and Repository: Version 1.0. *J. Chem. Theory Comput.*, 7, 4026–4037, 2011.

[81] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[82] A. P. Eichenberger, J. R. Allison, J. Dolenc, D. P. Geerke, B. A. C. Horta, K. Meier, C. Oostenbrink, N. Schmid, D. Steiner, D. Wang, et al.. The GROMOS++ Software for the Analysis of Biomolecular Simulation Trajectories. *J. Chem. Theory Comput.*, 7, 3379–3390, 2011.

[83] https://github.com/digantamisra98/Mila

[84] D. P. Kingma, J. L. Ba. Adam: A Method for Stochastic Optimization, 2014.