JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

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

Lennard Böselt, Moritz Thürlemann, and Sereina Riniker*

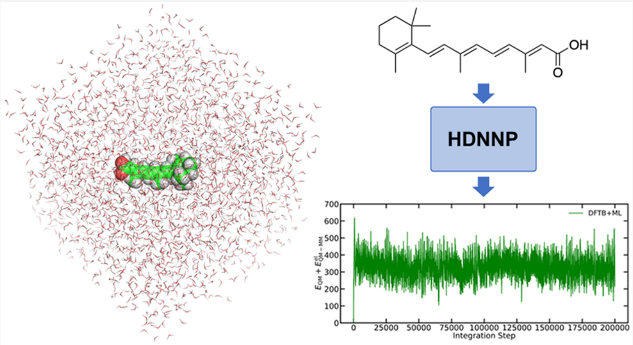
Cite This: *J. Chem. Theory Comput.* 2021, 17, 2641−2658

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

Downloaded via 73.68.53.135 on October 26, 2023 at 02:08:48 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

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 strong gradient regularization, a large number of data points, and a substantial number of parameters. To address this issue, we extend our approach to a Δ-learning scheme, where the ML model learns the difference between a reference method (density functional theory (DFT)) and a cheaper semiempirical method (density functional tight binding (DFTB)). We show that such a scheme reaches the accuracy of the DFT reference method while requiring significantly less parameters. Furthermore, the Δ-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 and cytosine in water. The presented results indicate that Δ-learning is a promising approach for (QM)ML/MM MD simulations of condensed-phase systems.



1. INTRODUCTION

Classical fixed-charge force fields (FFs) are readily used to perform molecular dynamics (MD) simulations of condensed-phase systems.[1,2] 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 vaporization, and solvation free energy.[3,4] FFs 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)[5−8] 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.

In the QM/MM scheme, the QM zone simulated with the 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 on either (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 favors efficient implementations; however, a major drawback is that a set of classical parameters has to be determined for the QM zone, which is not trivial and sometimes not possible with sufficient accuracy.[9] In electro-

Received: October 22, 2020
Published: April 5, 2021

ACS Publications

© 2021 The Authors. Published by American Chemical Society

https://doi.org/10.1021/acs.jctc.0c01112

static embedding, the MM environment is incorporated into the Hamiltonian operator of the QM zone as one-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 interactions between the MM and 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 FFs are used.[10] 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 Hamiltonans in the QM zone.[9,11] 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.[7,9,11−13]

In QM/MM, the description of the QM zone becomes the computational bottleneck as it requires an expensive self-consistent field (SCF) procedure and explicit treatment of all valence electrons.[14,15] To partially circumvent these issues, semiempirical methods[12,16] can be used to describe the QM zone.[17] This extends the accessible time scales but also reduces the accuracy.[12] An alternative is to employ machine-learned (ML) potentials.

In recent years, much research effort has been invested into the development of ML models trained on the potential-energy surface (PES) of QM systems.[18−30] 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 an SCF procedure is no longer necessary and the valence electrons do not have to be treated explicitly. A large number of approaches have been reported in the literature to achieve this task. These approaches differ in:

- use case: e.g., gas-phase[19,20] or periodic box,[18] small compounds[20] or larger compounds,[29] system-specific[27,28] or system-unspecific;[29]
- descriptor used to encode the chemical structure: e.g., Coulomb matrices,[31] distance matrices,[27,28] low-order polynomials,[19,20] or so-called symmetry functions;[18,23,29]
- target output: e.g., energies[18,19] or gradients;[23,27,28]
- scope: e.g., global[27,28] or local;[18,21−23,29]
- ML method: e.g., neural networks[18,19] or Kernel methods;[27,28,32]
- applying a restriction of the final form of the PES;[32] and
- convolution of the chemical graph: using a set of hard-coded functions,[18,19,27] or using other techniques such as graph (convolutional) architectures.[33]

Other ML models do not target the PES but rather the electron density[34] or the wave function itself.[35] However, these will not be the subject of the current study. All of the approaches listed above partially generate the correct parity behavior of the system. In other words, swapping two elements of the same type and rotating or translating the system leave the output unchanged or change 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 (PIP)[19] neural networks are suitable for accurately describing reactions of small systems in the gas phase (less than five 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 behavior 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)[27,28] 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.[28]

Larger compounds and box simulations can be tackled with high-dimensional neural network potentials (HDNNPs).[18,21,22] 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 introduce the correct parity behavior[36] while still allowing the HDNNP to learn the PES accurately. Using symmetry functions, Smith et al.[29] 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.[37] For periodic-box simulations, HDNNP can be trained system-specific on previously sampled data points.[18,21,22] 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 recomputed 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.[22] 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.[23] The uniqueness of the descriptor has been discussed in a theoretical work.[38]

Another important contribution to the field is the concept of Δ-learning.[24] 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 Δ-learning scheme was proposed by Shen et al.,[39] where a modified HDNNP was trained on the energy difference between a semiempirical 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 that can be used to perform MD simulations.[40] 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,[30] HDNNP has been proven to be currently 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 involve the following.[11,26,30,32,41]

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 the HDNNP model depends on a densely sampled training set. The number of different possible system configurations 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 many-body (0.35 nm) and two-body (0.52 nm) terms.[29,37]
3. Important long-range interactions: HDNNP makes use of a locality ansatz,[42] 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).[22] In QM/MM, this would correspond to a mechanical embedding scheme.[43] ML approaches have been proposed to specifically target this issue,[38,42] 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.[1−3] Such large cutoffs are necessary to achieve the desired accuracy for bulk properties such as heat of vaporization or solvation free energy.[43]
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 an ML potential is usually significantly slower than calculating the gradients with a classical force field.[37]

In addition, HDNNP and other ML models 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,[44] especially if a high number of weights is necessary to converge to chemical accuracy.[45]

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. 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.[29,37] 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[24] with a semiempirical method. Semiempirical 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 *S*-adenosylmethionate (SAM) with cytosin (63 atoms in the QM zone and approximately 3500 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. Choice of the Model. In nonrelativistic Born−Oppenheimer approximated,[46] time-independent wavefunction mechanics,[47] the electronic energy including nuclear repulsion E_{QM} can be computed by solving the Eigen function[48]

$$\hat{H}_{\mathrm{QM}}\psi_{\vec{R}}(\vec{r}) = E_{\mathrm{QM}}(\vec{R})\psi_{\vec{R}}(\vec{r}) \tag{1}$$

In eq 1, $\psi_{\vec{R}}(\vec{r})$ is the electronic wave function, $\vec{r}$ are the coordinates of the electrons, and $\vec{R}$ are the coordinates of the nuclei. The subscript $\vec{R}$ denotes that the wave function depends on the coordinates of the nuclei as a parameter. We will use the notation $\psi_{\vec{R}}(\vec{r}) \equiv \psi(\vec{r})$ in the following. $\hat{H}_{\mathrm{QM}}$ is the electronic Hamiltonian operator including the nuclear core−core repulsion and reads in atomic units

$$\hat{H}_{\mathrm{QM}} = -\frac{1}{2}\sum_{i}^{N_{\mathrm{el}}}\nabla_i^2 + \sum_{i<j}^{N_{\mathrm{el}}}\frac{1}{|\vec{r}_i - \vec{r}_j|} - \sum_{i}^{N_{\mathrm{el}}}\sum_{j}^{N_{\mathrm{QM}}}\frac{Z_j}{|\vec{r}_i - \vec{R}_j|} + \sum_{i<j}^{N_{\mathrm{QM}}}\frac{Z_iZ_j}{|\vec{R}_i - \vec{R}_j|} \tag{2}$$

where Z_j is the charge of the nuclei j. The indices run over the total number of electrons N_{el} and the total number of nuclei N_{QM}. The quantum Born−Oppenheimer approximated potential energy E_{QM} is obtained by computing the expectation value

$$E_{\mathrm{QM}}(\vec{R}) = \frac{\langle\psi(\vec{r})|\hat{H}_{\mathrm{QM}}\psi(\vec{r})\rangle}{\langle\psi(\vec{r})|\psi(\vec{r})\rangle} \tag{3}$$

According to Newton's second law,[49] the gradients on the nuclei of the system can be computed by taking the derivative with respect to the coordinates $\vec{R}$[47]

$$-\frac{\partial E_{\mathrm{QM}}(\vec{R})}{\partial \vec{R}_i} = \vec{F}_i = m_i \frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} \tag{4}$$

In eq 4, $\vec{F}_i$ is the force of particle i, m_i is its mass, $\vec{v}_i$ is its velocity, and t is the time. The system can then be propagated in time using an integration algorithm such as the leap-frog algorithm.[50] Propagating the system using the quantum Born–Oppenheimer approximated PES is computationally unfeasible for large systems. Hence, approximated solutions to eqs 1 and 2 were developed, which can be grouped into four classes.[51] (1) Ab initio methods use the correct Hamiltonian and an approximated wave function $\psi(\vec{r})$. (2) DFT[52] computes the electron density $\rho(\vec{r})$, which is used to calculate the energy of the molecular system, i.e., $E[\rho(\vec{r})]$. Modern DFT approaches achieve an accuracy comparable to that of 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) Semiempirical methods use an approximated Hamiltonian $\hat{H}$ and correct for the error made by introducing a set of fitted parameters. The most prominent example is the PM7 method.[17] Another attractive alternative is density functional tight binding (DFTB),[53] which can be considered a semiempirical variant of DFT. In DFTB, the electron density ρ is expanded in a series around a reference density ρ_0 and truncated. The error made is corrected with fitted parameters, similar to other semiempirical methods. (4) The last class includes empirical models with effective parameters such as classical force fields.[1,2,16]

2.1.2. Classical Force Fields. In classical fixed-charge MD simulations, the potential energy of the system is calculated with a force field,[1,2,16] which is the sum of bonded and nonbonded interactions terms

$$E_{\mathrm{MM}}(\vec{R}) = E^{\mathrm{bond}}(\vec{R}) + E^{\mathrm{angle}}(\vec{R}) + E^{\mathrm{dihedral}}(\vec{R}) + E^{\mathrm{el}}(\vec{R}) + E^{\mathrm{vdW}}(\vec{R}) \tag{5}$$

where E^{bond} is the contribution of all covalent bonds, E^{angle} is that of all covalent angles, and E^{dihedral} is that of all covalent dihedrals. The nonbonded terms consist of electrostatic (E^{el}) and van der Waals (E^{vdW}) interactions. In many force fields, the van der Waals interactions are only calculated up to a cutoff and neglected afterward (straight truncation).[1] To indicate that only short-range van der Waals interactions are included, we will use the notation $E^{\mathrm{vdW,SR}}$ in the following. The parameters of these interaction terms are typically fitted to reproduce experimental and/or QM reference data. Equation 5 is inexpensive to solve and depends only on the number of nuclei (atoms). The inclusion of experimental data in the fitting procedure allows one to incorporate long-range effects in an implicit manner. Thus, they are capable of reproducing bulk properties, such as the solvation free energy, accurately (see, e.g., refs 54–56). However, fixed-charge force fields treat electronic effects only in an averaged manner and can thus not be used to simulate chemical reactions. The simplicity of eq 5 allows one 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,[5−8] 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 energy $E_{\mathrm{QM/MM}}$ of the system, which is now a combination of the QM subsystem E_{QM} and the MM surrounding E_{MM}. A first approach to compute the total energy of the system is a subtractive scheme[7]

$$E_{\mathrm{QM/MM}}(\vec{R}) = E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E_{\mathrm{MM}}(\vec{R}) - E_{\mathrm{MM}}(\vec{R}_{\mathrm{QM}}) \tag{6}$$

where $E_{\mathrm{QM}}(\vec{R})$ is the energy of the QM subsystem, $E_{\mathrm{MM}}(\vec{R})$ is the energy of the complete system calculated with the classical force field, and $E_{\mathrm{MM}}(\vec{R}_{\mathrm{QM}})$ is the energy of the QM zone calculated with the force field. Note the distinction between $\vec{R}$, which refers to all nuclei in the system, and $\vec{R}_{\mathrm{QM}}$ and $\vec{R}_{\mathrm{MM}}$, which refer to the nuclei treated quantum-mechanically (QM zone) and the atoms treated classically (MM zone), respectively. The more prominent alternative is the additive scheme,[7] which is also used in this work, i.e.,

$$E_{\mathrm{QM/MM}}(\vec{R}) = E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E^{\mathrm{el}}_{\mathrm{QM-MM}}(\vec{R}) + E^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}}(\vec{R}) + E_{\mathrm{MM}}(\vec{R}_{\mathrm{MM}}) \tag{7}$$

Here, the superscript "el" stands for the electrostatic interactions and the superscript "vdW,SR" stands for the short-range van der Waals interactions. 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, $E^{\mathrm{el}}_{\mathrm{QM-MM}}$ and $E^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}}$, is added to the energy expression. The former 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. In electrostatic embedding, two Hamiltonians are introduced for dealing with the interactions between the MM environment and the QM particles. In atomic units, $\hat{H}^{\mathrm{el}}_{\mathrm{QM-MM}}$ reads[7]

$$\hat{H}^{\mathrm{el}}_{\mathrm{QM-MM}} = -\sum_{i}^{N_{\mathrm{MM}}} \sum_{j}^{N_{\mathrm{el}}} \frac{q_i}{|\vec{R}_{\mathrm{MM},i} - \vec{r}_j|} + \sum_{i}^{N_{\mathrm{QM}}} \sum_{j}^{N_{\mathrm{MM}}} \frac{Z_i q_j}{|\vec{R}_{\mathrm{QM},i} - \vec{R}_{\mathrm{MM},j}|} \tag{8}$$

where N_{MM} is the number of partial charges and q_j is the partial charge of MM atom j. The QM subsystem is thus directly influenced by the MM partial charges, while the latter particles feel a force of the perturbed QM zone. $\hat{H}^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}}$ deals with the short-range van der Waals interaction and is treated classically. It reads as follows

$$\hat{H}^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}} = E^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}}(\vec{R}) = \sum_{i}^{N_{\mathrm{QM}}} \sum_{j}^{N_{\mathrm{MM}}} 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{|\vec{R}_i - \vec{R}_j|}\right)^{12} - \left(\frac{\sigma_{ij}}{|\vec{R}_i - \vec{R}_j|}\right)^{6}\right) \tag{9}$$

where ϵ_{ij} and σ_{ij} are fitted parameters. Thus, in electrostatic embedding, eq 7 becomes

https://doi.org/10.1021/acs.jctc.0c01112

$$E_{\mathrm{QM/MM}}(\vec{R}) = \frac{\langle \psi(\vec{r})|(\hat{H}_{\mathrm{QM}} + \hat{H}^{\mathrm{el}}_{\mathrm{QM-MM}})\psi(\vec{r})\rangle}{\langle \psi(\vec{r})|\psi(\vec{r})\rangle} + E^{\mathrm{vdW,SR}}_{\mathrm{QM-MM}}(\vec{R}) + E_{\mathrm{MM}}(\vec{R}_{\mathrm{MM}}) \quad (10)$$

Typically, not all MM partial charges are included in the summation in eq 8 but only those within a cutoff radius R_c.[7] This leads to a PES, which is noncontinuous at the cutoff. This issue can be (partially) resolved by adaptive resolution schemes.[57] It has been found that the cutoff radius R_c needs to be chosen relatively large (i.e., 1.4 nm) to converge to accurate results.[11]

2.2. Machine Learning of Potential-Energy Surfaces.

Supervised ML methods can be used to learn a function f_P,[58−60] which describes the relation f_P: $h(\vec{R}) \rightarrow \tilde{E}(\vec{R})$. 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 $\vec{R}$ on a PES and then fit the parameters P in the function f_P to obtain an accurate description of $\tilde{E}(\vec{R})$. The configurations $\vec{R}$ 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 $L_P(E(\vec{R}), \tilde{E}(\vec{R})) = \mathrm{argmin}_P(g_c(E(\vec{R}), \tilde{E}(\vec{R})))$ becomes minimal, where g_c is an arbitrary convex function. Examples are mean-squared errors and mean absolute errors (MAEs). $\tilde{E}(\vec{R})$ can subsequently be used to propagate an MD trajectory.[22]

Different target outputs can be chosen. Instead of learning the complete potential $\tilde{E}(\vec{R})$, it is possible to learn only the difference between two levels of theory (e.g., semiempirical and DFT)[12]

$$\Delta E(\vec{R}) = E^{\mathrm{expensive}}(\vec{R}) - E^{\mathrm{cheap}}(\vec{R}) \quad (11)$$

The function f thus becomes a mapping of f_P: $h(\vec{R}) \rightarrow \Delta\tilde{E}(\vec{R})$. Alternatively, it is possible to directly learn the gradients.[23] An obstacle when learning gradients is that $-\int_{-\infty}^{\infty} \mathrm{d}\vec{R}\frac{\partial E(\vec{R})}{\partial \vec{R}} = E$ must be ensured. This can, for example, be achieved with the sGDML method.[27,28] As a third option, information about the gradients can be included in the training. For this, an additional loss term $L_P(E, \tilde{E}) = \mathrm{argmin}_P\left(g_c(E, \tilde{E}) + g_c\left(\frac{\partial E}{\partial \vec{R}}, \frac{\partial \tilde{E}}{\partial \vec{R}}\right)\right)$ is added to account for the gradients. The concepts described above are independent of the ML method used and the transformation h applied. It is essential that h does not remove information relevant for the target output.

2.2.1. *High-Dimensional Neural Network Potentials (HDNNPs).*

Feed-forward neural networks (FNNs) consist of a number of sequentially chained (non)-linear functions σ^l, weight matrices W^l, and biases b^l, where l is the number of layers. An FNN can be defined recursively as[58]

$$x^{l+1} = \sigma^l(W^l x^l + b^l) \quad (12)$$

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 σ^l.

FNNs are the method of choice in cases where a memory-intensive data set is required and complicated transformations are necessary to learn the underlying relation between input and output. Moreover, they provide more flexibility compared to Kernel ridge regression and Gaussian processes. It has been observed in Kaggle contests[61] that two hidden layers usually perform best if no special activation functions are used. In addition, FNNs with a lower number of parameters tend to generalize better and require smaller training sets.[45]

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 energy of the system unchanged. It is possible to augment the training data set to force the FNN to learn the correct symmetry. However, this increases the training set by multiple orders of magnitude. Behler et al.[18] identified this issue and developed an FNN architecture and an input-coordinate transformation h, which guarantees the correct parity behavior, the so-called high-dimensional neural network potentials (HDNNPs). 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 element type. The total energy of the system E is thus separated, i.e., $E = \sum_i^{N_\mathrm{H}} E_i^\mathrm{H} + \sum_i^{N_\mathrm{C}} E_i^\mathrm{C} + \cdots$, where N_H is the number of hydrogen atoms, N_C is the number of carbon atoms, and so on. Note that this separation ansatz is an approximation, and 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.[18] The Cartesian coordinates are transformed using a set of symmetry functions, which are invariant with respect to rotation and translation[18,29]

$$S_i^{t,\mathrm{Rad}}(\vec{R}) = \sum_j \mathrm{e}^{-\eta_{\mathrm{Rad}}\cdot(R_{ij}-\mu_{\mathrm{Rad}})^2} \quad (13)$$

and

$$S_i^{t,\mathrm{Ang}}(\vec{R}) = 2^{\zeta-1}\sum_{j\neq k\neq i}(1-\cos(\theta_{ijk}-\theta_\mathrm{S}))^{\zeta}\, \mathrm{e}^{-(((R_{ij}-R_{ik})/2)-\mu_{\mathrm{Ang}})^2\cdot\eta_{\mathrm{Ang}}} \quad (14)$$

where $R_{ij} \equiv |\vec{R}_i - \vec{R}_j|$ and θ_{ijk} is the angle spanned by atoms i, j, and k. The additional parameters η_{Rad}, μ_{Rad}, μ_{Ang}, ζ, θ_S, and η_{Ang} need to be set manually. The superscript t (for type) indicates that the summation only runs over a specific element type. "Rad" stands for radial and "Ang" for angular. Symmetry functions describe the chemical environment of each atom with radial (eq 13) and angular (eq 14) features. The summation operator enforces correct parity behavior.[62]

Applying symmetry functions to periodic boundary conditions requires a cutoff. For this, a cutoff function f_C can be added with the cutoff parameters $R_{\mathrm{sym,Rad}}$ (for radial symmetry functions) and $R_{\mathrm{sym,Ang}}$ (for angular symmetry functions), i.e.,

$$S_i^{t,\mathrm{Rad}}(\vec{R}) = \sum_j \mathrm{e}^{-\eta_{\mathrm{Rad}}\cdot(R_{ij}-\mu_{\mathrm{Rad}})^2}\cdot f_\mathrm{C}(R_{ij};\, R_{\mathrm{sym,Rad}}) \quad (15)$$

and

$$S_i^{t,\mathrm{Ang}}(\vec{R}) = 2^{\zeta-1} \sum_{j\neq k\neq i}^{N} (1-\cos(\theta_{ijk}-\theta_S))^{\zeta} \\ \mathrm{e}^{-(((R_{ij}-R_{ik)/2)}-\mu_{\mathrm{Ang}})^2\cdot\eta_{\mathrm{Ang}}}\cdot f_C(R_{ij};\ R_{\mathrm{sym,Ang}})\cdot \\ f_C(R_{ik};\ R_{\mathrm{sym,Ang}}) \tag{16}$$

with the cutoff function f_C defined as

$$f_C(R_{ij};\ R_{\mathrm{sym}}) = \begin{cases} 0.5\cdot\left(1+\cos\left(\dfrac{R_{ij}\cdot\pi}{R_{\mathrm{sym}}}\right)\right), & \text{for } R_{ij} \le R_{\mathrm{sym}} \\ 0, & \text{else} \end{cases} \tag{17}$$

Note that the symmetry functions S take over the role of the function h introduced in the previous section. 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 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.

HDNNPs present a powerful technique as these models can learn the complete Born−Oppenheimer-approximated PES (given an appropriate training set), i.e.

$$\underbrace{E_{\mathrm{QM}}(\vec{R})}_{\text{learned by HDNNP}} = \frac{\langle\psi(\vec{r})|\hat{H}_{\mathrm{QM}}\psi(\vec{r})\rangle}{\langle\psi(\vec{r})|\psi(\vec{r})\rangle} \tag{18}$$

Applying Δ-learning and using the definition of ΔE in eq 11, we can write

$$E_{\mathrm{QM}}^{\mathrm{cheap}}(\vec{R}) + \underbrace{\Delta E(\vec{R})}_{\text{learned by HDNNP}} = \frac{\langle\psi(\vec{r})|\hat{H}_{\mathrm{QM}}\psi(\vec{r})\rangle}{\langle\psi(\vec{r})|\psi(\vec{r})\rangle} \tag{19}$$

However, HDNNPs 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. Instead of introducing a new element type, the symmetry functions are weighted with respect to their atomic number and/or charge.[26] Other issues with HDNNPs are that the full correct permuting behavior is only partially recovered[63] and that the descriptors used are nonunique.[42] However, the latter issue is, in our opinion, more of a 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 the original work, HDNNPs are used to learn the complete Born−Oppenheimer-approximated PES. In this work, we propose to combine the HDNNP with classical force fields in a QM/MM-type approach to reduce the complexity of learning. Hence, instead of eq 18, our target is

$$\underbrace{E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E_{\mathrm{QM-MM}}^{\mathrm{el}}(\vec{R})}_{\text{learned by HDNNP}} + E_{\mathrm{MM}}(\vec{R}_{\mathrm{MM}}) + E_{\mathrm{QM-MM}}^{\mathrm{vdW,SR}}(\vec{R}) \\ = E(\vec{R}) \tag{20}$$

or in the case that Δ-learning is applied it is

$$E_{\mathrm{QM,cheap}}(\vec{R}_{\mathrm{QM}}) + E_{\mathrm{QM-MM,cheap}}^{\mathrm{el}}(\vec{R}) \\ + \underbrace{\Delta E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + \Delta E_{\mathrm{QM-MM}}^{\mathrm{el}}(\vec{R})}_{\text{learned by HDNNP}} + E_{\mathrm{MM}}(\vec{R}_{\mathrm{MM}}) \\ + E_{\mathrm{QM-MM}}^{\mathrm{vdW,SR}}(\vec{R}) \\ = E(\vec{R}) \tag{21}$$

$E_{\mathrm{MM}}(\vec{R}_{\mathrm{MM}})$ and $E_{\mathrm{QM-MM}}^{\mathrm{vdW,SR}}(\vec{R})$ are provided by the classical force field. Thus, it is expected that the complexity of the model can be reduced drastically. Furthermore, long-range interactions are now included via point charges, allowing one to use a larger cutoff in the construction of the HDNNP training set. For this approach, it is necessary to incorporate the MM environment into the descriptor of 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

$$S_i^{t,\mathrm{Rad}}(\vec{R}_{\mathrm{QM}}) = \sum_{j}^{N(t)_{\mathrm{QM}}} \mathrm{e}^{-\eta_{\mathrm{Rad}}\cdot(R_{ij}-\mu_{\mathrm{Rad}})^2}\cdot f_C(R_{ij};\ R_{\mathrm{sym,Rad}}^{\mathrm{QM-QM}})) \tag{22}$$

and

$$S_i^{t,\mathrm{Ang}}(\vec{R}_{\mathrm{QM}}) = 2^{\zeta-1} \sum_{j\neq k\neq i}^{N(t)_{\mathrm{QM}}} (1-\cos(\theta_{ijk}-\theta_S))^{\zeta} \\ \mathrm{e}^{-(((R_{ij}-R_{ik})/2)-\mu_{\mathrm{Ang}})^2\cdot\eta_{\mathrm{Ang}}}\cdot f_C(R_{ij};\ R_{\mathrm{sym,Ang}})\cdot \\ f_C(R_{ik};\ R_{\mathrm{sym,Ang}}) \tag{23}$$

MM particles are introduced as a new element type by

$$S_i^{t,\mathrm{Rad}}(\vec{R}) = \sum_{j}^{N_{\mathrm{MM}}} \mathrm{e}^{-\eta_{\mathrm{Rad}}\cdot(R_{ij}-\mu_{\mathrm{Rad}})^2}\cdot f_C(r_{ij};\ R_{\mathrm{sym,Rad}}^{\mathrm{QM-MM}})\cdot Z(j) \tag{24}$$

$$S_i^{t,\mathrm{Ang}}(\vec{R}) = 2^{\zeta-1} \sum_{j\neq i}^{N(t)_{\mathrm{QM}}} \sum_{k}^{N_{\mathrm{MM}}} (1-\cos(\theta_{ijk}-\theta_S))^{\zeta} \\ \mathrm{e}^{-(((R_{ij}-R_{ik})/2)-\mu_{\mathrm{Ang}})^2\cdot\eta_{\mathrm{Ang}}}\cdot f_C(R_{ij};\ R_{\mathrm{sym,Ang}})\cdot \\ f_C(R_{ik};\ R_{\mathrm{sym,Ang}})\cdot Z(k) \tag{25}$$

and

$$S_i^{t,\mathrm{Ang}}(\vec{R}) = 2^{\zeta-1} \sum_{j\neq k}^{N_{\mathrm{MM}}} (1-\cos(\theta_{ijk}-\theta_S))^{\zeta} \\ \mathrm{e}^{-((R_{ij}-R_{ik})/2)-\mu_{\mathrm{Ang}})^2\cdot\eta_{\mathrm{Ang}}}\cdot f_C(R_{ij};\ R_{\mathrm{sym,Ang}})\cdot \\ f_C(R_{ik};\ R_{\mathrm{sym,Ang}})\cdot Z(j)\cdot Z(k) \tag{26}$$

In eqs 24−26, the summation operator iterates over all MM particles N_{MM} and QM particles $N(\mathrm{t})_{\mathrm{QM}}$ of type t. The function $Z(i)$ returns the partial charge of the MM particle. For SPC/E water, we have $Z(\mathrm{O}) = -0.8476$ and $Z(\mathrm{H}) = 0.4238$. $R_{\mathrm{sym,Rad}}^{\mathrm{QM-QM}}$

and $R_{\text{sym,Rad}}^{\text{QM-MM}}$ are cutoff parameters for the radial symmetry functions, the first one influencing the QM–QM interaction and the second one targeting the inclusion of the MM particles.

This approach requires that the data set be 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 $\tilde{E}_{\text{QM-MM}}^{\text{el}}(\vec{R}) + E_{\text{QM}}(\vec{R}_{\text{QM}})$ 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) chemical reactions in water (constrained close to the transition state), namely, the second-order nucleophilic substitution (S_N2) reaction of CH_3Cl with Cl^- and the reaction of *S*-adenosylmethionate (SAM) with cytosine. The systems were chosen to represent different system sizes, use cases, and difficulties. The systems were treated in either 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[64,65] interfaced to DFTB+/19.2[53] and ORCA/4.2.0.[14] The functionals used in this study are the gradient-corrected Becke–Perdew functional (BP86), Head-Gordon's range-separated functional ωB97X-D3,[66] and Grimme's double-hybrid functional B2-PLYP.[67] The basis set used is Ahlrichs and Weigend's def2-TZVP.[68]

All QM calculations used the resolution of identity,[69] Weigend's auxiliary basis[70] functions, and Grimme's dispersion correction with Becke–Johnson damping[71,72] (except for ωB97X-D3). DFT calculations, which require the computation of a Hartree–Fock[73] reference wave function, were further accelerated using the so-called COSX[74] approximation. The ORCA computations used TightSCF convergence criteria and the integration Grid5. The grid was changed to Grid6 in the final iteration. Otherwise, standard parameters were used. DFTB computations were performed with Grimme's D3 dispersion correction with Becke–Johnson damping.[71,72]

All point charges within the cutoff radius R_c 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 were set to 10^{-8} eV for all systems. Structures that did not converge in the maximum number of steps were discarded. MD simulations were performed with a convergence criterion of 10^{-6} eV and a Broyden mixer[75] 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. The temperature was kept constant in the MD simulations with the Nosé–Hoover chain[76,77] thermostat with a coupling constant of 0.1 ps and two baths (one coupled to the internal motion and the rotation of the solute, and the other one to the solvent). A weak-coupling[78] barostat was used for constant-pressure simulations with a coupling constant of 0.5 ps and an isothermal compressibility of 4.575×10^{-4} (kJ mol^{-1} $\text{nm}^{-3})^{-1}$. Long-range electrostatic interactions beyond the cutoff of 1.4 nm were included using a reaction-field method.[79] Note that the reaction field acts only on the MM particles. The water model used in this study was SPC/E.[80] All bonds between MM particles were constrained using the SHAKE algorithm[81] and a relative tolerance of 10^{-4}. The motion of the center of mass was removed every 1000 steps. A topology file of the QM solute is required as input in GROMOS. The Lennard-Jones parameters are needed to evaluate $E_{\text{QM-MM}}^{\text{vdW,SR}}$ since this interaction between the QM zone and MM zone is treated on the classical level. All other parameters are discarded for the QM solute. The topology files were obtained from the ATB[82] server.

The neural networks were implemented using Tensorflow/keras.[83] 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[82] server. Initial structures for the chemical reactions were generated by placing the reactants manually close to the transition state. During sampling, a biasing potential was applied of the form

$$V^{\text{bias}} = \frac{1}{2}k(R_{ij} + d\cdot R_{kl} - R_0)^2 \tag{27}$$

where k is the force constant and R_{ij} and 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 performed $d = -1$ and $R_0 = 0.00$. For the CH_3Cl/Cl^- system, the indexing is shown as follows

$$Cl_jC_{i,k}H_3 + Cl_l^- \rightarrow ClCH_3 + Cl^- \tag{28}$$

The structure was minimized using a force constant for the biasing potential $k = 2000$ kJ mol^{-1} nm^{-2}.

For the SAM/cytosine system, atom i is the carbon atom participating in the reaction, atom j is the sulfur atom, atom k is the same carbon atom as atom i, and atom l corresponds to the carbon atom in the α-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,[84] with a minimum solute–wall distance of 1.6 nm and a minimum solute–solvent distance of 0.25 nm. The sizes of the simulation boxes were between 3.3 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 298 K or 400 K with a coupling constant of 0.1 ps.

3.4. Sampling of Configurations. All systems were simulated for 20 000 steps at $T = 400$ K, $p = 1$ bar. The first 10 000 steps were discarded as equilibration. The remaining 10 000 frames were split into 70/20/10 training/validation/test sets. Thus, the last 1000 steps were considered as the external test set. The performed splitting mimics the envisioned use case and gives a validation and a test set, which do not resemble the training set as much as if a random partitioning would have been performed.

For benzene in water, a cutoff for the partial charges in the QM/MM Hamiltonian of $R_c = 0.6$ nm was used. For all other systems, the cutoff for the partial charges was set to $R_c = 1.4$ nm. For CH_3Cl/Cl^{-1} and SAM, a biasing potential with $k = 2000$ kJ mol^{-1} nm^{-1} was employed. Note that for the training of the HDNNP, the biasing potential is removed. Commonly used sampling strategies such as normal-mode sampling are difficult to employ for these systems as these techniques do not account for the solvent environment.

https://doi.org/10.1021/acs.jctc.0c01112

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 systems to 0.5 fs, the temperature was set to T = 298 K, and the pressure was set to 1 bar. For the SAM/cytosine system, the biasing force constant was set to k = 2000 kJ mol^{-1} nm^{-2}.

3.6. Neural Networks. We implemented the HDNNP in Tensorflow/keras with a float64 precision. Each HDNNP used at most two hidden layers and thus followed the recommendation in ref 61. The activation function mila[85] was used with a coefficient β of −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 Adam optimizer.[86] The number of neurons per hidden layer ranged for each NNP from 10 to 80 neurons. As a loss function, we used

$$L = \frac{1}{N} \cdot \sum_{i}^{N} (E_i - \tilde{E}_i)^2 \qquad (29)$$

and

$$L' = L + \frac{\omega_0}{3N_{\mathrm{QM}}} \cdot \sum_{i}^{N_{\mathrm{QM}}} \sum_{\alpha}^{3} (-F_{i\alpha} + \tilde{F}_{i\alpha})^2 + \frac{\omega_1}{3N_{\mathrm{MM}}} \cdot \sum_{i}^{N_{\mathrm{MM}}} \sum_{\alpha}^{3} (-F_{i\alpha} + \tilde{F}_{i\alpha})^2 \qquad (30)$$

where N_{MM} is the number of MM particles, N_{QM} is the number of QM particles, and ω_0 and ω_1 are weight parameters for the gradient contribution. Unless noted otherwise, ω_0 = 1 and ω_1 = 1. Note that L is a nongradient corrected loss function and L' is a gradient-corrected loss function. All models were initially trained for 50 steps with a learning rate of 3.5 × 10^{-4}. The SAM/cytosine system was trained for 20 epochs. We monitored the validation loss during the training process. After training, the model with the lowest loss on the validation set was recovered.

3.7. Symmetry Functions. The symmetry functions were extracted from refs 29 and 37. 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. Table1 lists the parameters for the symmetry functions in eqs 13−26, which encode the environment. The cutoffs for the radial symmetry functions were varied in this study and are provided in Section 4 for all models.

Table 1. Parameters of the Symmetry Functions Used in This Work;, See Equations 13−26[a]

parameter	type	from	to	step size
μ_{Rad}	radial	0.09	1.40	0.026875
η_{Rad}	radial	160	160	0
μ_{Ang}	angular	0.09	0.285	0.065
$R_{\mathrm{sym,Ang}}$	angular	0.35	0.35	0
η_{Ang}	angular	80	80	0
ζ	angular	32	32	0
θ_S	angular	0.19634954	2.9452431	0.19634954

[a] The symmetry functions were taken from ref 37 and adapted to units nm and nm^{-1}. The cutoffs for the radial symmetry functions for the different models are provided in Section 4.

For the MM particles, we either used 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.[29,37] 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 data sets. The summation in eq 8 goes over all partial charges within R_c. Similarly, the pairwise van der Waals interactions in eq 9 are calculated within R_c. $R^{\mathrm{QM-QM}}_{\mathrm{sym,Rad}}$ is the cutoff used for the radial symmetry functions encoding the QM environment (see eq 22), whereas $R^{\mathrm{QM-MM}}_{\mathrm{sym,Rad}}$ is the cutoff used for the radial symmetry functions encoding the MM environment (see eq 24). $R^{\mathrm{QM-QM}}_{\mathrm{sym,Rad}}$ and $R^{\mathrm{QM-MM}}_{\mathrm{sym,Rad}}$ correspond to the cutoffs used in the HDNNP.

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 gradient correction, many-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: benzene (apolar), uracil (polar), and the transition state of $\mathrm{CH_3Cl/Cl^-}$ (charged). The trajectories were split into 70/20/10 training/validation/test sets (see Section 3). As discussed in Section 2, the ML models learn the quantity $E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E^{\mathrm{el}}_{\mathrm{QM-MM}}(\vec{R})$. The performances of the ML models are thus assessed based on the error on this quantity (energy) and on the respective derivatives (forces), i.e., the gradients on the QM particles

$$\frac{\partial(E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E^{\mathrm{el}}_{\mathrm{QM-MM}}(\vec{R}))}{\partial \vec{R}_{\mathrm{QM}}} = -\vec{F}_{\mathrm{QM}} \qquad (31)$$

and the gradients on the MM particles resulting from the QM zone

$$\frac{\partial(E_{\mathrm{QM}}(\vec{R}_{\mathrm{QM}}) + E^{\mathrm{el}}_{\mathrm{QM-MM}}(\vec{R}))}{\partial \vec{R}_{\mathrm{MM}}} = -\vec{F}_{\mathrm{MM}} \qquad (32)$$

4.1.1. Gradient Correction and Many-Body Terms. The first test system consists of benzene in water. Benzene is an apolar solute. Thus, long-range interactions do not contribute significantly to the potential energy compared to polar solutes. Point charges further away than 0.6 nm from the solute atom were not included in the evaluation of eq 8. The test system is suitable to study the effect of including gradient correction and many-body terms (eqs 24−26) in the description of the PES for the MM part. Note that the descriptor always contains many-body information for the QM particles (eq 16). 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

Table 2. Settings for the ML Models Used to Test the Influence of Including Gradient Correction or Three-Body Terms for the MM Point Charges[a]

		gradient correction		cutoff		
settings ID	three-body terms for MM?	ω_0	ω_1	$R^{\mathrm{QM-QM}}_{\mathrm{sym,Rad}}$ (nm)	$R^{\mathrm{QM-MM}}_{\mathrm{sym,Rad}}$ (nm)	R_c (nm)
M1	no	0	0	1.0	1.0	0.6
M2	yes	0	0	1.0	1.0	0.6
M3	no	1	1	1.0	1.0	0.6
M4	no	1	1−200	1.0	1.0	0.6

[a] ω_0 and ω_1 are weight parameters for the gradient contribution (eq 30). R_c is the cutoff radius for the MM partial charges in the QM/MM calculation to generate the data set, $R^{\mathrm{QM-QM}}_{\mathrm{sym,Rad}}$ is the cutoff for the QM−QM symmetry functions, and $R^{\mathrm{QM-MM}}_{\mathrm{sym,Rad}}$ is the cutoff for the QM−MM symmetry functions. The test system is benzene in water.

subsystem with the MM particles by introducing them as a new element type (see Section 2). Model 1 (M1) is the baseline model, including only two-body information and no gradient correction. M2 includes three-body terms for the MM point charges. M3 only includes gradient information. M4 additionally weights the gradients of the MM particles in the loss function. Note that the cutoffs $R^{\mathrm{QM-QM}}_{\mathrm{sym,Rad}}$ and $R^{\mathrm{QM-MM}}_{\mathrm{sym,Rad}}$ for M1−M4 are larger than R_c. This means that the models "know" about all particles contributing to the potential energy.

We first analyzed the model performance as a function of the number of neurons in the hidden layers for M1 and M2 (Figure 1). The accuracy on the energies of the validation/test set is far away from chemical accuracy for both models, and also, the performance on the QM gradients (mid-panel) and the MM gradients (right panel) is poor. Including three-body terms in the descriptor for the MM particles (M2) does not improve the accuracy. Interestingly, the performance decreases even further with an increasing number of neurons in the hidden layers. This is in line with the observation in ML that large numbers of weight parameters should be avoided when no regularization techniques are applied.

M3 includes gradient information of the QM and MM particles in the training procedure and thus regularizes the weights of the models. The accuracy of the energies of the validation set is now clearly better than the chemical accuracy limit, and also the performance of the gradients is substantially improved. We observe that the performance is stabilized, meaning that a larger number of neurons per layer slightly improves the performance instead of worsening it.

The influence of regularizing the models via gradients can be further highlighted by varying the parameter ω_1. In eq 30, ω_1 scales the contribution of the error on the MM gradients. The gradients on the MM particles are much smaller than the QM gradients since the quantity only incorporates the electrostatic interaction with the QM zone and not the interaction of the MM particles with themselves. The latter interactions as well as the short-range van der Waals interactions between the QM and MM particles are modulated with the classical force field. Thus, the contribution of the MM particles to the loss function is substantially smaller than the contribution of the energy and forces of the QM particles if $\omega_1 = 1$ is used. The performance of the training set measured by the loss function is thus substantially increased at the cost of a worse description of the interaction with the MM particles. This behavior is clearly nonphysical and leads to an HDNNP, which performs worse on the validation (and test) set. Scaling the MM gradients via ω_1 solves this issue. Figure 2 shows the performance of M4 on the energies and gradients as a function of the parameter ω_1. By increasing ω_1, the prediction accuracy can be further improved.

4.1.2. Δ-Learning. As ML models use a locality ansatz, it is difficult for them to describe long-range interactions. Semiempirical methods, on the other hand, are known to be able to account for long-range interactions. Thus, a Δ-learning[24] scheme based on a semiempirical method might be a valuable approach. In such a scheme, the ML model introduces a correction to the semiempirical method to recover the accuracy of the higher-level method. Although this decreases naturally the computational efficiency of the production run, since an additional computation is required at each time step t, we will demonstrate in the following that the performance is increased substantially and that the number of weight parameters can be reduced. As a result, the number of training data points needed is reduced, the training procedure is accelerated, and the extrapolation capabilities of the model are possibly increased.

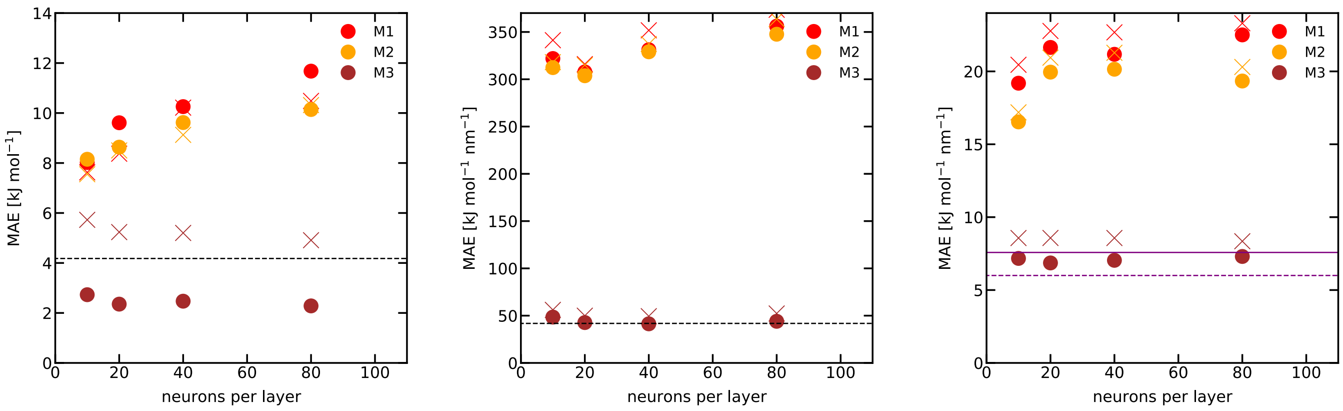


Figure 1. Performance of the settings M1−M3 for benzene in water as a function of the number of neurons in each hidden layer. The reference data points were computed with B2-PLYP/def2-TZVP. The performance is reported for the validation set (filled circles) and the test set (crosses) as the mean absolute error (MAE). 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). (Left) MAE of the predicted energies. (Middle) MAE of the predicted QM gradients. (Right) MAE of the predicted MM gradients. For comparison, the purple lines denote the performance of DFTB on the validation set (dashed line) and the test set (solid line).

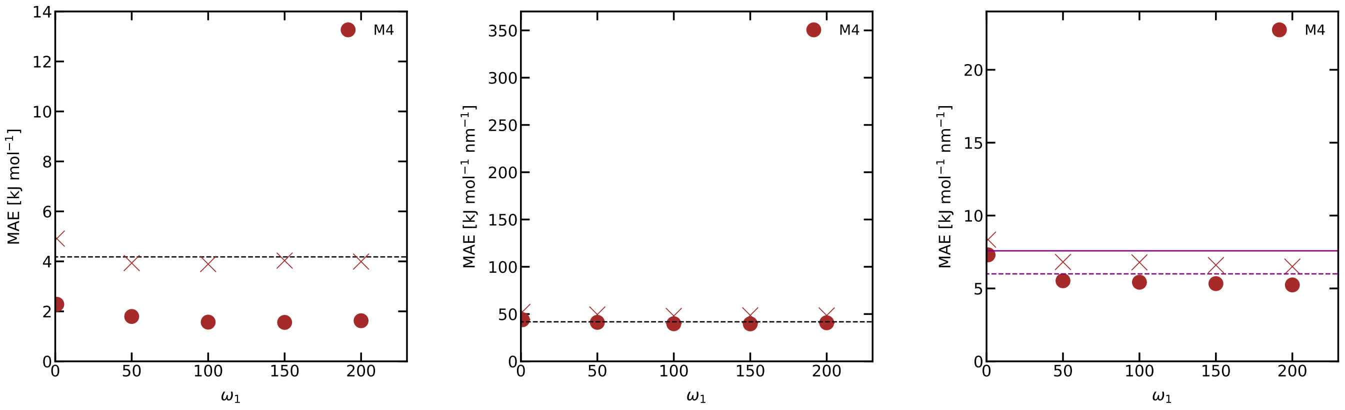


Figure 2. Performance of the setting M4 for benzene in water as a function of the weight ω_1 (see eq 30). The reference data points were computed with B2-PLYP/def2-TZVP. All models used 80 neurons per layer. The performance is reported for the validation set (filled circles) and the test set (crosses) as the mean absolute error (MAE). 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). (Left) MAE of the predicted energies. (Middle) MAE of the predicted QM gradients. (Right) MAE of the predicted MM gradients. For comparison, the purple lines denote the performance of DFTB on the validation set (dashed line) and the test set (solid line).

Table 3 summarizes the settings. The Δ-learning model M5 uses the same settings as M1, i.e., only two-body terms for the QM−MM interactions, no gradient correction (ω_0 = 0, ω_1 = 0), and a cutoff for the symmetry functions of 1.0 nm. M6 uses the same settings as M3, i.e., gradient information is included in the training procedure (ω_0 = 1, ω_1 = 1). M7 uses stronger gradient regularization (ω_0 = 1, ω_1 = 200) and is thus comparable to M4. In Figure 3, the Δ-learning models are compared to M3.

Table 3. Settings for the ML Models Used to Test the Influence of Including Gradient Correction together with Δ-Learning[a]

		gradient correction		cutoff		
settings ID	Δ-learning	ω_0	ω_1	$R_{\text{sym,Rad}}^{\text{QM−QM}}$ (nm)	$R_{\text{sym,Rad}}^{\text{QM−MM}}$ (nm)	R_c (nm)
M5	yes	0	0	1.0	1.0	0.6
M6	yes	1	1	1.0	1.0	0.6
M7	yes	1	200	1.0	1.0	0.6

[a] ω_0 and ω_1 are weight parameters for the gradient contribution (eq 30). R_c is the cutoff radius for the MM partial charges in the QM/MM calculation to generate the data set, $R_{\text{sym,Rad}}^{\text{QM−QM}}$ is the cutoff for the QM−QM symmetry functions, and $R_{\text{sym,Rad}}^{\text{QM−MM}}$ is the cutoff for the QM−MM symmetry functions. The test system is benzene in water.

The results show that using gradient correction is crucial for performance as M3 (gradient correction but no Δ-learning) clearly outperforms M5 (Δ-learning but no gradient correction). At the same time, the prediction accuracy can be substantially increased using Δ-learning in combination with gradient correction (M6). However, while the MAE with M6 is below 4.18 kJ mol^{-1} for the validation set, it is above it for the test set. By increasing the regularization (M7, ω_1 = 200), the performance can be further improved for the test set. M7 is thus the overall best performing model. In general, it is important to carefully monitor the performance of neural networks as a function of their parameters.

Using Δ-learning and gradient correction improved the accuracy of all three properties, i.e., energies, QM gradients, and MM gradients, already at a small number of neurons per hidden layer. This indicates that only a reduced number of

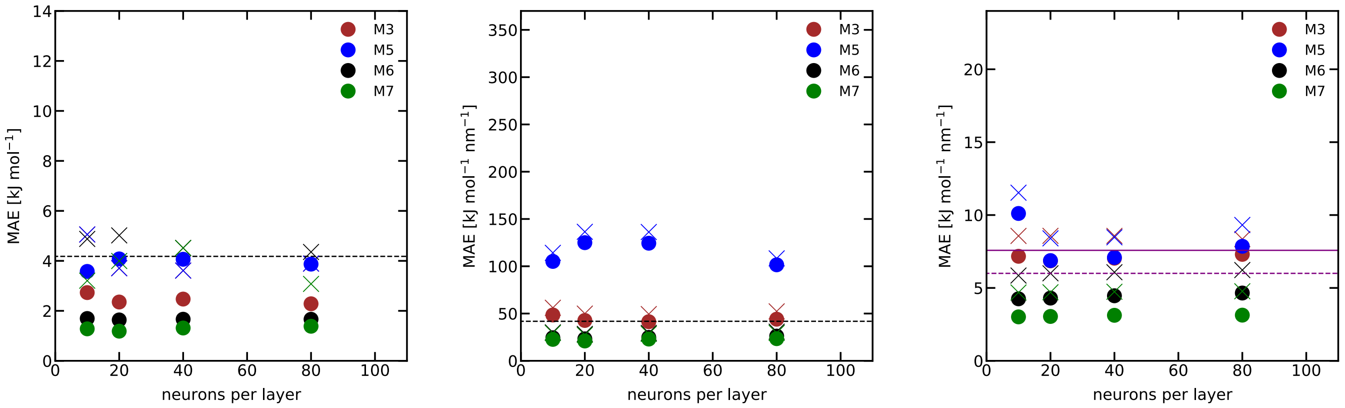


Figure 3. Performance of the settings M3 (no Δ-learning) and M5−M7 (Δ-learning) for benzene in water as a function of the number of neurons in each hidden layer. The reference data points were computed with B2-PLYP/def2-TZVP. The performance is reported for the validation set (filled circles) and the test set (crosses) as the mean absolute error (MAE). 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). (Left) MAE of the predicted energies. (Middle) MAE of the predicted QM gradients. (Right) MAE of the predicted MM gradients. For comparison, the purple lines denote the performance of DFTB on the validation set (dashed line) and the test set (solid line).

Table 4. Comparison of the Performance of M4 and M7 on the Data Set of Benzene in Water When Trained on 10% or 70% of the Data Set[a]

	energy		QM gradients		MM gradients	
model	MAE (kJ mol^{-1})	RMSE (kJ mol^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})
M4 (70%)	2.5, 1.8, 4.0	3.0, 2.2, 6.1	31.9, 34.0, 48.5	41.8, 45.9, 70.7	4.2, 5.2, 6.5	6.8, 8.5, 11.2
M4 (10%)	6.2, 3.9, 5.0	6.6, 4.8, 6.7	64.6, 89.7, 96.2	86.6, 122.0, 135.0	5.4, 7.1, 8.4	8.3, 12.3, 14.3
M7 (70%)	1.8, 1.6, 3.2	2.2, 2.1, 4.6	23.7, 22.8, 29.4	31.1, 30.0, 44.2	3.5, 3.0, 4.7	6.4, 5.4, 8.9
M7 (10%)	1.1, 1.6, 3.2	1.4, 2.0, 4.1	25.4, 31.1, 39.3	33.1, 41.7, 59.9	3.5, 3.5, 5.1	6.1, 6.2, 9.5
DFTB	–, 6.0, 5.0	–, 7.5, 6.6	–, 143.2, 154.2	–, 190.2, 209.3	–, 6.0, 7.6	–, 10.5, 14.1

[a] Performance is reported as the mean absolute error (MAE) and root-mean-square error (RMSE). The first number indicates the performance on the training set, the second number that on the validation set, and the third number that on the test set. The reference data points were computed with B2-PLYP/def2-TZVP. For comparison, the performance of DFTB on the validation and test set is given. After the standard training protocol described in Section 3, M4 (10%) and M7 (10%) were trained for 50 more steps with a learning rate of 3.5 × 10^{-4}.

training data points is necessary to achieve the accuracy of the reference method (see Section 4.1.3). It also suggests that the extrapolation capabilities of M6/M7 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 complete. To the best of our knowledge, the lowest level of theory available for such a Δ-learning scheme is DFTB. Other semiempirical 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 in learning of complex systems is that not all relevant configurations can be enumerated. Thus, ML approaches that can be trained on smaller data sets are especially valuable. As shown above, Δ-learning requires less weight parameters to achieve chemical accuracy on the validation and test sets 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 (ω_1 = 200, 80 neurons per layer) and M7 (ω_1 = 200, 10 neurons per layer) models were trained on only 10% of the data set. This was done by changing the training/validation/test splitting from 70/20/10 to 10/20/10 (i.e., discarding the last 60% of the training set). More precisely, the ML models with a reduced data set are fitted on the first 1000 data points of the training set, while the validation and test sets remain the same. The results are presented in Table 4. It is striking that for M4 the prediction accuracy on the energies and gradients of the same validation and test sets drops by up to 50% when trained only on 10% of the data set. For M7, on the other hand, the error on the energies remains relatively constant when only trained on 10% of the data set. The performance of the gradients worsens as well but to a smaller extent.

4.1.4. Cutoff Size. Fitting an ML model for the benzene-in-water system did not require the inclusion of long-range interactions as the partial charges beyond 0.6 nm were truncated. To assess the effect of 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 1.4 nm, which corresponds to 1500−2000 MM partial charges in the cutoff sphere. A cutoff of 1.4 nm is used in the GROMOS force field,[3] of which the SPC/E water model is part of. Table 5 lists settings M8−M13. All models use gradient correction, and M11−M13 follow the Δ-learning scheme with DFTB. M11 serves as the "worst-case" model for Δ-learning, where the MM particles are simulated solely with the semiempirical method, and the ML model introduces a local correction to the QM subsystem.

Table 5. Settings for the ML Models Used to Test the Influence of the Cutoff Size[a]

	cutoff			
settings ID	$R_{\mathrm{sym,Rad}}^{\mathrm{QM-QM}}$ (nm)	$R_{\mathrm{sym,Rad}}^{\mathrm{QM-MM}}$ (nm)	R_c (nm)	Δ-learning
M8	0.52	0.00	1.40	no
M9	0.52	0.52	1.40	no
M10	1.40	1.40	1.40	no
M11	0.52	0.00	1.40	yes
M12	0.52	0.52	1.40	yes
M13	1.40	1.40	1.40	yes

[a] All models employ only two-body terms for the QM−MM interactions and gradient correction. The models were trained with 10 neurons per hidden layer. R_c is the cutoff radius for the MM partial charges in the QM/MM calculation, $R_{\mathrm{sym,Rad}}^{\mathrm{QM-QM}}$ is the cutoff for the QM−QM symmetry functions, and $R_{\mathrm{sym,Rad}}^{\mathrm{QM-MM}}$ is the cutoff for the QM−MM symmetry functions. The test system is uracil in water.

The results with models M8−M13 and the DFTB baseline for uracil in water are shown in Figure 4 and summarized in Table 6. We will first discuss the models without Δ-learning, M8−M10. For these, long-range interactions need to be learned completely by the ML model. M8 has no information about the surrounding environment, and thus the performance on the energies and the QM gradients is poor. Increasing the cutoff of the symmetry function for the description of the MM environment to 0.52 nm (M9) increases the performance substantially. The MAE on the energies on the validation set is reduced by nearly a factor 4, and the description of the gradients also becomes better. Interestingly, increasing the cutoff further worsens the results. The reason is likely that the training data set is relatively sparse (7000 data points). Increasing the cutoff in the symmetry functions also increases the number of weight parameters, which need to be fitted. This complicates the training procedure, making it harder to find meaningful minima. The situation is different for the Δ-learning approach (M11−M13). Here, the semiempirical method can be used to incorporate important long-range interactions, while the ML model serves as a local correction. A striking example is provided by M11. M11 like M8 has no information about the surrounding partial charges. Nevertheless, the model almost reaches chemical accuracy on the validation set. Compared to M8, the gradients on the QM particles are improved by a factor 3 and the gradients on the partial charges by a factor 7. Including information about the

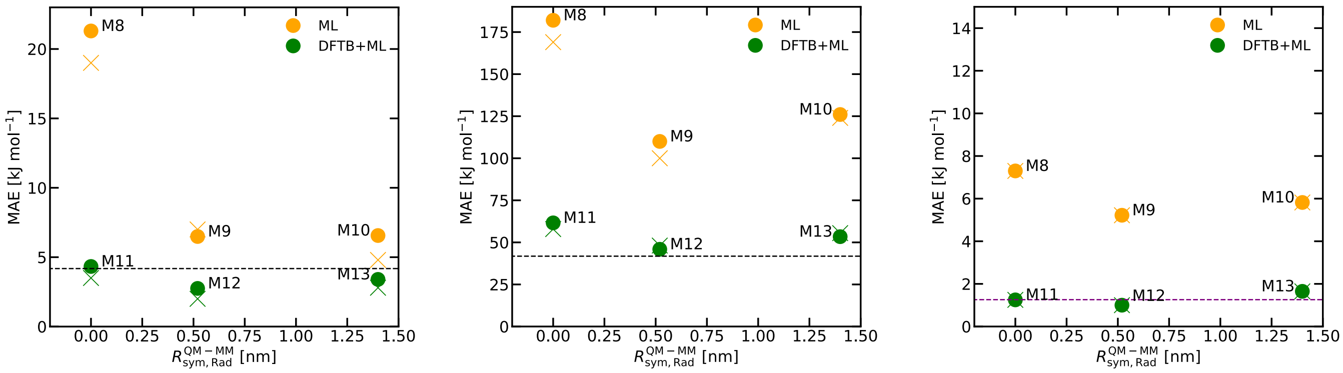


Figure 4. Performance of settings M8−M10 (no Δ-learning, yellow) and M11−M13 (Δ-learning, green) on the validation (filled circles) and test (crosses) sets of uracil in water as a function of the cutoff $R^{QM-MM}_{sym,Rad}$. The reference data points were computed with ωB97X-D3/def2-TZVP. The performance is reported as the mean absolute error (MAE). 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). (Left) MAE of the predicted energies. (Middle) MAE of the predicted QM gradients. (Right) MAE of the predicted MM gradients. For comparison, the purple lines denote the performance of DFTB on the validation set (dashed line) and the test set (solid line).

Table 6. Performance of Models M8−M13 for Uracil in Water[a]

	energy		QM gradients		MM gradients	
model	MAE (kJ mol^{-1})	RMSE (kJ mol^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})
M8	16.2, 21.3, 19.0	20.0, 26.3, 23.5	128.0, 182.0, 169.0	170.0, 278.0, 242.0	7.8, 7.3, 8.1	23.3, 23.5, 24.6
M9	5.2, 6.5, 7.0	6.2, 8.5, 8.6	69.9, 110.0, 100.0	91.6, 176.0, 139.0	5.6, 5.2, 5.89	10.7, 10.1, 11.1
M10	2.9, 6.6, 4.8	3.7, 8.5, 6.0	63.8, 126.0, 124.0	81.7, 191.0, 181.0	6.2, 5.8, 6.5	11.6, 10.9, 12.0
M11	4.6, 4.3, 3.5	5.8, 5.5, 4.2	47.1, 61.6, 57.9	60.6, 81.8, 76.5	1.3, 1.3, 1.3	5.2, 5.5, 5.0
M12	1.2, 2.8, 2.0	1.5, 3.6, 2.7	34.3, 45.9, 48.1	44.6, 62.0, 64.4	1.1, 1.0, 1.0	3.2, 3.2, 3.1
M13	0.9, 3.4, 2.8	1.2, 4.2, 3.5	29.8, 53.4, 55.5	38.4, 72.2, 75.4	1.7, 1.7, 1.7	3.7, 3.9, 3.8
DFTB	–, 12.4, 12.7	–, 16.0, 15.2	–, 160.0, 151.5	–, 700.1, 716.9	–, 1.3, 1.3	–, 5.5, 5.0

[a] The performance is reported as the mean absolute error (MAE) and root-mean-square error (RMSE). The first value indicates the performance on the training set (70%), the second number on the validation set (20%), and the third number on the test set (10%). The reference data points were computed with ωB97X-D3/def2-TZVP. For comparison, the performance of DFTB on the validation and test sets is given.

Table 7. Performance of Models M11−M13 for the (Close to) Transition State of CH_3Cl/Cl^- in Water[a]

	energy		QM gradients		MM gradients	
model	MAE (kJ mol^{-1})	RMSE (kJ mol^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})
M11	4.6, 3.7, 3.5	6.3, 4.6, 5.0	59.9, 56.5, 45.5	94.1, 83.9, 86.4	1.2, 1.1, 1.1	6.5, 5.2, 5.2
M12	1.8, 1.7, 2.2	2.3, 2.1, 2.8	24.9, 29.5, 30.7	33.6, 40.8, 44.2	1.0, 1.0, 1.0	3.9, 3.3, 3.0
M13	1.2, 2.3, 4.5	1.5, 2.9, 4.9	25.8, 40.0, 43.7	33.6, 54.4, 67.7	1.5, 1.4, 1.4	4.2, 3.8, 3.6
DFTB	–, 6.6, 6.8	–, 8.4, 9.4	–, 191.6, 203.9	–, 236.6, 276.9	–, 1.1, 1.1	–, 5.2, 5.2

[a] The performance is reported as the mean absolute error (MAE) and root-mean-square error (RMSE). The first value indicates the performance on the training set (70%), the second number on the validation set (20%), and the last number on the test set (10%). The reference data points were computed with Head-Gordon's range-separated functional ωB97X-D3. For comparison, the performance of DFTB on the validation and test sets is given.

partial charges in the surrounding up to 0.52 nm improves the results further. Again, going from a cutoff of 0.52 to 1.4 nm worsens the results, likely due to the same reason. However, given a sufficiently large training set (larger than in the current cases), a $R_{sym,Rad}$ > 0.52 nm may be desirable.

Finally, we compare DFTB to the ML models. All ML models outperform DFTB in predicting the QM gradients. However, DFTB outperforms the ML models without Δ-learning on the MM gradients. This highlights again the importance of applying Δ-learning for systems, where long-range interactions are important and only sparse data sets are available. Further, it is interesting to observe that DFTB outperforms M8, which has no information about the surrounding partial charges, on the QM energies. This finding suggests that well-fitted semiempirical methods may be preferred over ML models if the latter do not account appropriately for the surrounding partial charges.

To assess the transferability of the conclusions from the previous test systems, we analyzed the performance of settings M11−M13 on a third test system, the (close to) 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. Furthermore, as it is a charged system, it is interesting to see how well the long-range interactions are described by the Δ-learning models. The results for M11−M13 for the CH_3Cl/Cl^- test system are summarized in Table 7. In general, we see the same behavior as for uracil in water, i.e., a cutoff of 0.52 nm performs the best.

4.2. Application of ML Models in MD Simulations. So far, we have assessed the performance of the ML models in

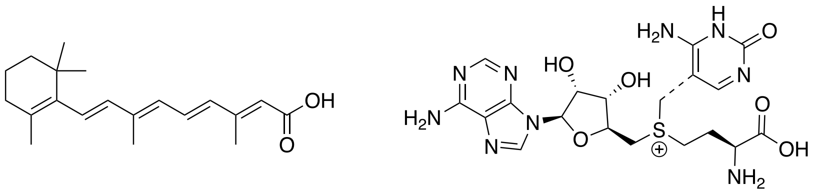


Figure 5. Two larger test systems used for MD simulations. (Left) Retinoic acid in water (50 QM atoms and about 2500 MM partial charges with a cutoff of 1.4 nm). (Right) Transition state of the chemical reaction between SAM and cytosine in water (63 QM atoms and about 3500 MM partial charges with a cutoff of 1.4 nm).

Table 8. Performance of the Δ-Learning Model on the Training and Validation Sets of Retinoic Acid in Water[a]

	energy		QM gradients		MM gradients	
model	MAE (kJ mol^{-1})	RMSE (kJ mol^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})
ML	3.9, 4.4	4.4, 5.0	43.8, 44.8	59.7, 61.8	1.1, 1.1	3.3, 3.1
DFTB	–, 9.9	–, 12.6	–, 216.3	–, 286.7	–, 1.3	–, 4.5

[a] The performance is reported as the mean absolute error (MAE) and root-mean-square error (RMSE). The first value corresponds to the training set, and the second value to the performance on the validation set. The reference data points were computed with BP86/def2-TZVP. For comparison, the performance of DFTB on the validation set is given.

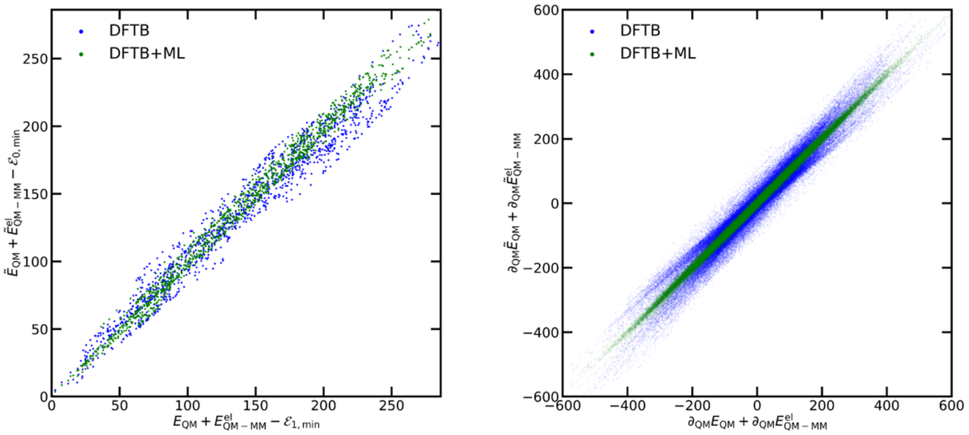


Figure 6. Comparison between the reference DFT method and the DFTB Hamiltonian (blue) or the DFTB + ML model (green) on the validation set of retinoic acid in water. The reference data points were computed with BP86/def2-TZVP. (Left) Energies, units in kJ mol^{-1}. (Right) QM gradients, units in kJ mol^{-1} nm^{-1}. We define $\frac{\partial}{\partial \vec{R}_{QM}} \equiv \partial_{QM}$. The minimum energy for the reference and the ML prediction is subtracted, as indicated with $-\epsilon_{0,min}$ and $-\epsilon_{1,min}$.

terms of MAE (and RMSE) on a validation and a test 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 simulation, where the results of the next step depend directly on the results of the previous step.

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 (close to) transition state of the reaction of SAM with cytosine in water (Figure 5). Sampling to generate the training and validation sets was performed at an elevated temperature (400 K) with a time step of 0.5 fs for both systems. Instead of an external test set, MD simulations were carried out at $T = 298$ K. Our main objective in this section is to show that given a sparse training set the proposed Δ-learning scheme is able to produce stable trajectories, which describe the PES sufficiently well even for unseen data points. We therefore use as above only 10 000 data points for training, even though the complexity of the systems is much higher compared to the previously studied ones. For real case studies, e.g., the computation of free-energy profiles, a more sophisticated sampling procedure for the training set is recommended.

The ML models investigated employ the settings M12, i.e., the Δ-learning scheme with DFTB as the baseline, 10 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 $R_{sym,Rad}$ of 0.52 nm. The QM/MM reference calculations were computed with a cutoff R_c of 1.40 nm.

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, and Figure 6 shows them graphically. Using the Δ-learning model, the performance is improved compared to the DFTB baseline for all three properties, including the MM gradients.

Next, we tested the trained ML model by performing a (QM)ML/MM MD simulation for 5000 steps (top panel in Figure 7). To compare the ML-corrected energies with those of the DFT reference and the DFTB baseline simulation, single

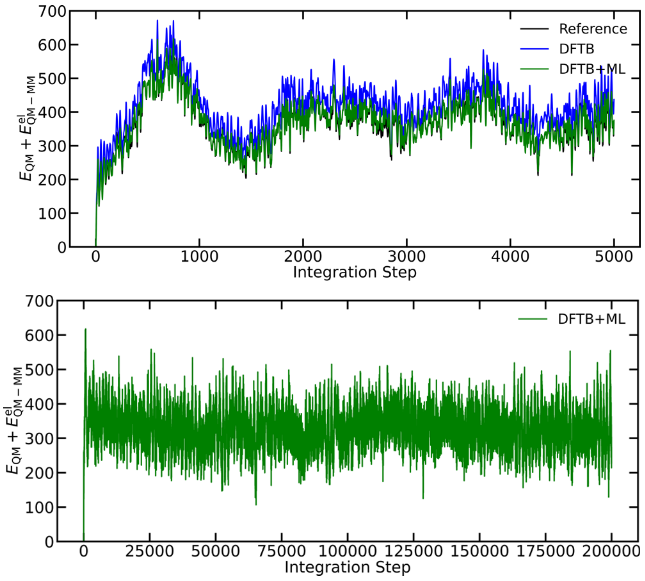


Figure 7. MD simulation of retinoic acid in water (integration step 0.5 fs). (Top) Energy (E_{QM} + E^{el}_{QM-MM}) trajectory from 5000 consecutive steps performed by the DFTB + ML model (green). In each step, the energies were also computed with DFTB (blue) and the DFT reference (BP86/def2-TZVP, black) for comparison. The trajectories are referenced to their zero energy. The energy deviation from the reference over the complete trajectory with respect to the minimum energy is MAE = 5.8 kJ mol^{-1} and RMSE = 7.6 kJ mol^{-1} for the DFTB + ML model and MAE = 52.8 kJ mol^{-1} and RMSE = 54.3 kJ mol^{-1} for DFTB. (Bottom) Energy trajectory from 200 000 consecutive steps performed by the DFTB + ML model (green).

point calculations were performed for each configuration in the DFTB + ML trajectory. As shown in Figure 7, the energies of DFTB + ML 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. We note that the MAE on the trajectory is 5.8 kJ mol^{-1} larger than the MAE values of 3.9 and 4.4 kJ mol^{-1} for the training and validation sets, respectively. The reason for this is that the starting point of the trajectory is the relaxed structure, a configuration never seen by the HDNNP. To achieve a higher accuracy in practical MD simulations for complex molecules such as retinoic acid, more training data points will be needed for the HDNNP. To assess the stability of the ML model, we extended the simulation to 200 000 steps (bottom panel in Figure 7).

4.2.2. SAM/Cytosine Transition State in Water. In biochemistry, *S*-adenosylmethionate (SAM) is a cofactor 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 8). The Δ-learning model clearly outperforms the DFTB baseline.

Next, we performed a (QM)ML/MM MD simulation for 2000 steps using the fitted model (top panel in Figure 9). As for the test system with retinoic acid in water, the energies with DFTB + ML 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 9). 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 Δ-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 3500 MM particles in the cutoff sphere, one evaluation with DFTB required less than a second (single core), while the corresponding DFT computation takes 60−80 min (on 4 cores, CPU time), which is more than 3 orders of magnitude longer. Thus, the additional time needed for the DFTB calculation at each step of the Δ-learning scheme is less expensive than, e.g., the pairlist creation in the MD engine. An obvious limitation of the Δ-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 the presented MD simulations.

5. SUMMARY AND CONCLUSIONS

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 semiempirical methods to describe the QM part 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

Table 9. Performance of the Δ-Learning Model on the Training and Validation Sets of the (Close to) Transition State of SAM and Cytosine in Water[a]

	energy		QM gradients		MM gradients	
model	MAE (kJ mol^{-1})	RMSE (kJ mol^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})	MAE (kJ mol^{-1} nm^{-1})	RMSE (kJ mol^{-1} nm^{-1})
ML	6.3, 8.5	6.9, 9.7	74.8, 74.6	101.0, 100.0	2.3, 2.3	6.7, 6.7
DFTB	–, 17.1	–, 21.9	–, 263.7	–, 377.2	–, 2.3	–, 7.6

[a]The performance is reported as the mean absolute error (MAE) and root-mean-square error (RMSE). The first value corresponds to the training set and the second value, to the performance on the validation set. The reference data points were computed with BP86/def2-TZVP. For comparison, the performance of DFTB on the validation set is given.

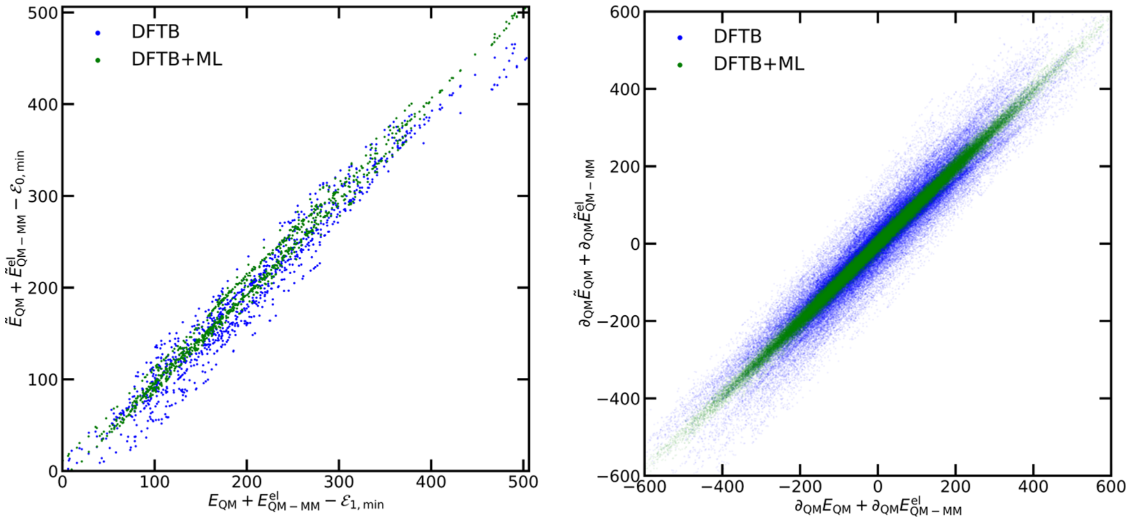


Figure 8. Comparison between the reference DFT method and the DFTB Hamiltonian (blue) or the DFTB + ML model (green) on the validation set of the (close to) transition state of SAM/cytosine in water. (Left) Energies, units in kJ mol^{-1}. (Right) QM gradients, units in kJ mol^{-1} nm^{-1}. We define $\frac{\partial}{\partial \vec{R}_{QM}} \equiv \partial_{QM}$. The minimum energy for the reference and the ML prediction is subtracted, as indicated with $-\epsilon_{0,\text{min}}$ and $-\epsilon_{1,\text{min}}$.

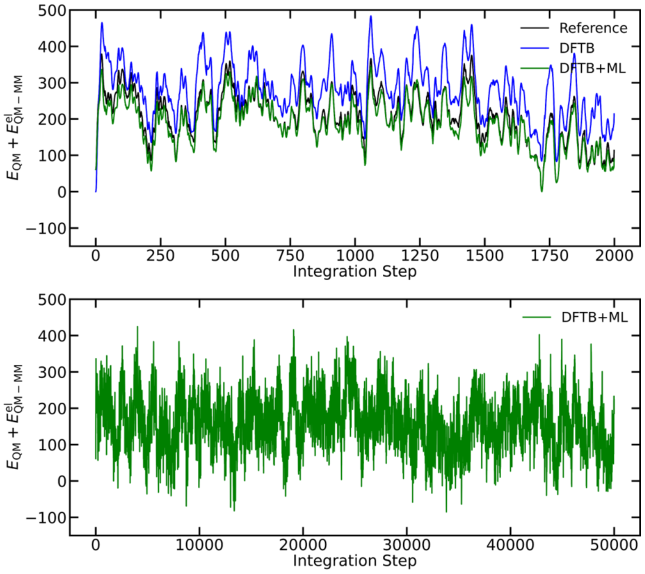


Figure 9. MD simulation of the SAM/cytosine transition state in water (integration step 0.5 fs). (Top) Energy (E_{QM} + E^{el}_{QM-MM}) trajectory from 2000 consecutive steps performed by the DFTB + ML model (green). In each step, the energies were also computed with DFTB (blue) and the DFT reference (BP86/def2-TZVP, black) for comparison. The energy deviation from the reference over the complete trajectory with respect to the minimum energy is MAE = 18.1 kJ mol^{-1} and RMSE = 22.2 kJ mol^{-1} for the DFTB + ML model and MAE = 74.9 kJ mol^{-1} and RMSE = 79.4 kJ mol^{-1} for DFTB. (Bottom) Energy trajectory from 50 000 consecutive steps performed by the DFTB + ML model (green).

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 and energies. In general, we found that strong regularization was necessary during the training procedure to predict these gradients with decent accuracy. This may no longer apply when larger training sets are used. As an alternative, we employed a Δ-learning scheme, where the lower-level method is DFTB. Semiempirical methods are known to handle long-range interactions well. For uracil in water, we showed that Δ-learning allows one to decrease the cutoff of the symmetry functions substantially while still outperforming classical HDNNP. The removal of symmetry functions reduces the number of weight parameters and the complexity of the training procedure. 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.

The final Δ-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. The performance can be further boosted by a more sophisticated sampling strategy for the training and validation set. We envision that the results and findings presented in this study will enable the use of (QM)ML/MM MD simulations in practical applications.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.0c01112.

Description of the implementation of Tensorflow/keras in a C++ environment, and correlation plots for models M1−M13 (energies, QM gradients for all models, MM gradients only for the models for the benzene test system) (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Sereina Riniker − *Laboratory of Physical Chemistry, ETH Zurich, 8093 Zurich, Switzerland;* orcid.org/0000-0003-1893-4031; Email: sriniker@ethz.ch

Authors

Lennard Böselt − *Laboratory of Physical Chemistry, ETH Zurich, 8093 Zurich, Switzerland*

Moritz Thürlemann – *Laboratory of Physical Chemistry, ETH Zurich, 8093 Zurich, Switzerland*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jctc.0c01112

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

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). M.T. was supported by the NCCR MARVEL, funded by the Swiss National Science Foundation.

■ REFERENCES

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

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

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

(4) Ponder, J. W.; Case, D. A. Force Fields for Protein Simulations. *Adv. Protein Chem.* **2003**, *66*, 27–85.

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

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

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

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

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

(10) Loco, D.; Lagardére, O. L.; Caprasecca, S.; Orcid, F. L.; Orcid, B. M.; Piquemal, J.-P. Hybrid QM/MM Molecular Dynamics with AMOEBA Polarizable Embedding. *J. Chem. Theory Comput.* **2017**, *13*, 4025–4033.

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

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

(13) Chung, L. W.; Sammera, W. M. C.; Ramozzi, R.; Page, A. J.; Hatanaka, M.; Petrova, G. P.; Harris, T. V.; Li, X.; Ke, Z.; Liu, F.; Li, H.-B.; Ding, L.; Morokuma, K. The ONIOM Method and Its Applications. *Chem. Rev.* **2015**, *115*, 5678–5796.

(14) Neese, F. Software Update: The ORCA Program System, Version 4.0. *WIREs Comput. Mol. Sci.* **2018**, *8*, No. e1327.

(15) Balasubramani, S. G.; Chen, G.; Coriani, S.; Diedenhofen, M.; Frank, M.; Franzke, Y.; Furche, F.; Grotjahn, R.; Harding, M.; Hättig, C.; Hellweg, A. TURBOMOLE: Modular Program Suite for Ab Initio Quantum-chemical and Condensed-matter Simulations. *J. Chem. Phys.* **2020**, *152*, No. 184107.

(16) Christensen, A. S.; Kubar, T.; Cui, Q.; Elstner, M. Semiempirical Quantum Mechanical Methods for Noncovalent Interactions for Chemical and Biochemical Applications. *Chem. Rev.* **2016**, *116*, 5301–5337.

(17) Stewart, J. J. P. Optimization of Parameters for Semiempirical Methods VI: More Modifications to the NDDO Approximations and Re-optimization of Parameters. *J. Mol. Model.* **2013**, *19*, 1–32.

(18) Behler, J. Atom-centered Symmetry Functions for Constructing High-dimensional Neural Network Potentials. *J. Chem. Phys.* **2011**, *134*, No. 074106.

(19) Jiang, B.; Guoa, H. Permutation Invariant Polynomial Neural Network Approach to Fitting Potential Energy Surfaces. *J. Chem. Phys.* **2013**, *139*, No. 054112.

(20) Li, J.; Jiang, B.; Guoa, H. Permutation Invariant Polynomial Neural Network Approach to Fitting Potential Energy Surfaces. II. Four-atom Systems. *J. Chem. Phys.* **2013**, *139*, No. 204103.

(21) Behler, J. Representing Potential Energy Surfaces by High-dimensional Neural Network Potentials. *J. Phys.: Condens. Matter* **2014**, *26*, No. 183001.

(22) Behler, J. Constructing High-dimensional Neural Network Potentials: A Tutorial Review. *Int. J. Quantum Chem.* **2015**, *115*, 1032–1050.

(23) Botu, V.; Ramprasad, R. Adaptive Machine Learning Framework to Accelerate Ab Initio Molecular Dynamics. *Int. J. Quantum Chem.* **2015**, *115*, 1074–1083.

(24) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.

(25) Gastegger, M.; Behler, J.; Marquetand, P. Machine Learning Molecular Dynamics for the Simulation of Infrared Spectra. *Chem. Sci.* **2017**, *8*, 6924–6935.

(26) Gastegger, M.; Schwiedrzik, L.; Bittermann, M.; Berzsenyi, F.; Marquetanda, P. wACSF-Weighted Atom-centered Symmetry Functions as Descriptors in Machine Learning Potentials. *J. Chem. Phys.* **2018**, *148*, No. 241709.

(27) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine Learning of Accurate Energy-conserving Molecular Force Fields. *Sci. Adv.* **2017**, *3*, No. e1603015.

(28) Chmiela, S.; Sauceda, H. E.; Mueller, K. R.; Tkatchenko, A. Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields. *Nat. Commun.* **2018**, *9*, No. 3887.

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

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

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

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

(33) Schütt, K. T.; Kindermans, P.-J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. In *SchNet: A Continuous-filter Convolutional Neural Network for Modeling Quantum Interactions*, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

(34) Fabrizio, A.; Grisafi, A.; Meyer, B.; Ceriotti, M.; Corminboeuf, C. Electron Density Learning of Non-covalent Systems. *Chem. Sci.* **2019**, *10*, 9424–9432.

(35) Qiao, Z.; Welborn, M.; Anandkuar, A.; Manby, F. R.; Miller, T. F. OrbNet: Deep Learning for Quantum Chemistry Using Symmetry-Adapted Atomic-Orbital. *J. Chem. Phys.* **2020**, *153*, No. 124111.

(36) Sannai, A.; Takai, Y.; Cordonnier, M. Universal Approximations of Permutation Invariant/Equivariant Functions by Deep Neural Networks. 2020, https://openreview.net/forum?id=HkeZQJBKDB.

(37) Stevenson, J.; Jacobson, L. D.; Zhao, Y.; Wu, C.; Maple, J.; Leswing, K.; Harder, E.; Abel, R. Schrödinger-ANI: An Eight-Element Neural Network Interaction Potential with Greatly Expanded Coverage of Druglike Chemical Space. **2019**, arXiv:1912.05079. arXiv.org e-Print archive. https://arxiv.org/abs/1912.05079.

(38) Pozdnyakov, S. N.; Willatt, M. J.; Bart, A. P.; Ortner, C.; Cśan, G.; Ceriotti, M. On the Completeness of Atomic Structure Representations. 2020, arXiv:2001.11696. arXiv.org e-Print archive. https://arxiv.org/abs/2001.11696.

(39) Shen, L.; Wu, J.; Yang, W. Multiscale Quantum Mechanics/Molecular Mechanics Simulations with Neural Networks. *J. Chem. Theory Comput.* **2016**, *12*, 4934−4946.

(40) Shen, L.; Yang, W. Molecular Dynamics Simulations with Quantum Mechanics/Molecular Mechanics and Adaptive Neural Networks. *J. Chem. Theory Comput.* **2018**, *14*, 1442−1455.

(41) Zhang, Y.; Khorshidi, A.; Kastlunger, G.; Peterson, A. A. The Potential for Machine Learning in Hybrid QM/MM Calculations. *J. Chem. Phys.* **2018**, *148*, No. 241740.

(42) Grisafi, A.; Ceriotti, M. Incorporating Long-range Physics in Atomic-scale Machine Learning. *J. Chem. Phys.* **2019**, *151*, No. 204105.

(43) Junming, H.; Shao, Y.; Kato, J. Do Better Quality Embedding Potentials Accelerate the Convergence of QM/MM Models? The Case of Solvated Acid Clusters. *Molecules* **2018**, *26*, No. 2466.

(44) Behler, J. First Principles Neural Network Potentials for Reactive Simulations of Large Molecular and Condensed Systems. *Angew. Chem., Int. Ed.* **2017**, *56*, 12828−12840.

(45) Gordon, A.; Eban, E.; Nachum, O.; Chen, B.; Wu, H.; Yang, T.-J.; Choi, E. MorphNet: Fast and Simple Resource-Constrained Structure Learning of DeepNetworks. 2017, arXiv:1711.06798. arXiv.org e-Print archive. https://arxiv.org/abs/1711.06798.

(46) Born, M.; Oppenheimer, R. Zur Quantentheorie der Molekeln. *Ann. Phys.* **1927**, *389*, 457−484.

(47) Reiher, M.; Wolf, A. *Relativistic Quantum Chemistry*; Wiley-VCH, 2014.

(48) Schrödinger, E. Quantisierung als Eigenwertproblem. *Ann. Phys.* **1926**, *385*, 437−490.

(49) Newton, I. Axioms or Laws of Motion. *Mathematical Principles of Natural Philosophy*; A. Motte, 1729; Vol. *1*.

(50) Gray, S. K.; et al. Symplectic Integrators for Large Scale Molecular Dynamics Simulations: A Comparison of Several Explicit Methods. *J. Chem. Phys.* **1994**, *101*, 4062.

(51) Levine, I. N. *Quantum Chemistry*; Pearson, 2013.

(52) Kohn, W.; Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **1965**, *140*, A1133−A1138.

(53) Hourahine, B.; Aradi, B.; Blum, V.; Bonafe, F.; Buccheri, A.; Chamacho, C.; Cevallos, C.; Deshaye, M. Y.; Dumitrica, T.; Dominguez, A.; Ehlert, S.; Elstner, M.; van der Heide, T.; Herman, J.; Irle, S.; Kranz, J. J.; Köhler, C.; Kowalczyk, T.; Kubar, T.; Lee, I. S.; Lutsker, V.; Maurer, R. J.; Min, S. K.; Mitchell, I.; Negre, C.; Niehaus, T. A.; Niklasson, A. M. N.; Page, A. J.; Pecchia, A.; Penazzi, G.; Persson, M. P.; Rezac, J.; Sanchez, C. G.; Sternberg, M.; Stöhr, M.; Stuckenberg, F.; Tkatcheno, A.; Yu, V. W.; Freuenheim, T. DFTB+, a Software Package for Efficient Approximate Density Functional Theory Based Atomistic Simulations Featured. *J. Chem. Phys.* **2020**, *152*, No. 124101.

(54) Mobley, D. L.; Guthrie, J. P. FreeSolv: A Database of Experimental and Calculated Hydration Free Energies, with Input Files. *J. Comput.-Aided Mol. Des.* **2014**, *28*, 711−720.

(55) Sweere, A. J. M.; Farrije, J. G. E. M. Accuracy Test of the OPLS-AA Force Field for Calculating Free Energies of Mixing and Comparison with PAC-MAC. *J. Chem. Theory Comput.* **2017**, *13*, 1911−1923.

(56) Kashefolgheta, S.; Oliveira, M. P.; Rieder, S. R.; Horta, B. A. C.; Acree, W. E., Jr.; Hünenberger, P. H. Evaluating Classical Force Fields against Experimental Cross-Solvation Free Energies. *J. Chem. Theory Comput.* **2020**, *16*, 7556−7580.

(57) Bulo, R. E.; Ensing, B.; Sikkema, J.; Visscher, L. Toward a Practical Method for Adaptive QM/MM Simulations. *J. Chem. Theory Comput.* **2009**, *5*, 2212−2221.

(58) Alpaydin, E. *Introduction to Machine Learning*; MIT Press, 2014.

(59) Paliwal, M.; Kumar, U. A. Neural Networks and Statistical tTechniques: A Review of Applications. *Expert Syst. Appl.* **2009**, *36*, 2−17.

(60) Kanagawa, M.; Hennig, P.; Sejdinovic, D.; Sriperumbudur, B. K. Gaussian Processes and Kernel Methods: A Review on Connections and Equivalences. 2018, arXiv:1807.02582. arXiv.org e-Print archive. https://arxiv.org/abs/1807.02582.

(61) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. 2017, arXiv:1706.02515. arXiv.org e-Print archive. https://arxiv.org/abs/1706.02515.

(62) Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R.; Smola, A. Deep Sets. 2017, arXiv:1703.06114. arXiv.org e-Print archive. https://arxiv.org/abs/1703.06114.

(63) Li, J.; Song, K.; Behler, J. A Critical Comparison of Neural Network Potentials for Molecular Reaction Dynamics with Exact Permutation Symmetry. *Phys. Chem. Chem. Phys.* **2019**, *21*, 9672−9682.

(64) Schmid, N.; Christ, C. D.; Christen, M.; Eichenberger, A. P.; van Gunsteren, W. F. Architecture, Implementation and Parallelization of the GROMOS Software for Biomolecular Simulation. *Comput. Phys. Commun.* **2012**, *183*, 890−903.

(65) Meier, K.; Schmid, N.; van Gunsteren, W. F. Interfacing the GROMOS (Bio)molecular Simulation Software to Quantum-Chemical Program Packages. *J. Comput. Chem.* **2012**, *33*, 2108−2177.

(66) Chai, J.-D.; Head-Gordon, M. M. Long-range Corrected Hybrid Density Functionals with Damped Atom-Atom Dispersion Corrections. *Phys. Chem. Chem. Phys.* **2008**, *44*, 6615.

(67) Grimme, S. Semiempirical Hybrid Density Functional with Perturbative Second-order Correlation. *J. Chem. Phys.* **2006**, *124*, No. 034108.

(68) Weigend, F.; Ahlrichs, R. 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.* **2005**, *7*, 3297−3305.

(69) Feyereisen, M.; Fitzgerld, G.; Komornicki, A. Use of Approximate Integrals in Ab Initio Theory. An Application in MP2 Energy Calculations. *Chem. Phys. Lett.* **1993**, *208*, 359−363.

(70) Weigend, F. Accurate Coulomb-fitting Basis Sets for H to Rn. *Phys. Chem. Chem. Phys.* **2006**, *8*, 1057−1065.

(71) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* **2010**, *132*, No. 154104.

(72) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the Damping Function in Dispersion Corrected Density Functional Theory. *J. Comput. Chem.* **2011**, *32*, 1456−1465.

(73) Hartree, D. R. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part II. Some Results and Discussion. *Math. Proc. Cambridge Philos. Soc.* **1928**, *24*, 111−132.

(74) Kossmann, S.; Neese, F. Comparison of Two Efficient Approximate Hartee-Fock Approaches. *Chem. Phys. Lett.* **2009**, *481*, 240−243.

(75) Broyden, C. G. A Class of Methods for Solving Nonlinear Simultaneous Equations. *Math. Comput.* **1965**, *19*, 577−593.

(76) Nosé, S. A Unified Formulation of the Constant Temperature Molecular Dynamics Methods. *J. Chem. Phys.* **1984**, *81*, 511.

(77) Hoover, W. G. Canonical Dynamics: Equilibrium Phase-space Distributions. *Phys. Rev. A* **1985**, *31*, 1695−1697.

(78) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; DiNola, A.; Haak, J. R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, *81*, 3684−3690.

(79) Tironi, I. G.; et al. A Generalized Reaction Field Method for Molecular Dynamics Simulations. *J. Chem. Phys.* **1995**, *102*, 5451−5459.

(80) Berendsen, H. J. C.; Grigera, J. R.; Straatsma, T. P. The Missing Term in Effective Pair Potentials. *J. Phys. Chem. A* **1987**, *91*, 6269−6271.

(81) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. C. Numerical Integration of the Cartesian Equations of Motion of a System with Constraints: Molecular Dynamics of n-Alkanes. *J. Comput. Phys.* **1977**, *23*, 327−341.

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

(83) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015, https://www.tensorflow.org/. Software available from tensorflow.org.

(84) Eichenberger, A. P.; Allison, J. R.; Dolenc, J.; Geerke, D. P.; Horta, B. A. C.; Meier, K.; Oostenbrink, C.; Schmid, N.; Steiner, D.; Wang, D.; van Gunsteren, W. F. The GROMOS++ Software for the Analysis of Biomolecular Simulation Trajectories. *J. Chem. Theory Comput.* **2011**, *7*, 3379−3390.

(85) https://github.com/digantamisra98/Mila.

(86) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. 2014, arXiv:1412.6980. arXiv.org e-Print archive. https://arxiv.org/abs/1412.6980.