

RESEARCH ARTICLE | MAY 22 2023

## Implicit solvent approach based on generalized Born and transferable graph neural networks for molecular dynamics simulations

Special Collection: [Machine Learning Hits Molecular Simulations](#)

Paul Katzberger  ; Sereina Riniker  



*J. Chem. Phys.* 158, 204101 (2023)

<https://doi.org/10.1063/5.0147027>



View  
Online



Export  
Citation

### Articles You May Be Interested In

Assessment of proton-coupled conformational dynamics of SARS and MERS coronavirus papain-like proteases: Implication for designing broad-spectrum antiviral inhibitors

*J. Chem. Phys.* (September 2020)

A generalized Kirkwood implicit solvent for the polarizable AMOEBA protein model

*J. Chem. Phys.* (August 2023)

# Implicit solvent approach based on generalized Born and transferable graph neural networks for molecular dynamics simulations

Cite as: J. Chem. Phys. 158, 204101 (2023); doi: 10.1063/5.0147027

Submitted: 18 February 2023 • Accepted: 3 May 2023 •

Published Online: 22 May 2023



View Online



Export Citation



CrossMark

Paul Katzberger and Sereina Riniker<sup>a)</sup>

## AFFILIATIONS

Department of Chemistry and Applied Biosciences, ETH Zürich, Vladimir-Prelog-Weg 2, 8093 Zürich, Switzerland

Note: This paper is part of the JCP Special Topic on Machine Learning Hits Molecular Simulations.

<sup>a)</sup>Author to whom correspondence should be addressed: [sriniker@ethz.ch](mailto:sriniker@ethz.ch)

## ABSTRACT

Molecular dynamics simulations enable the study of the motion of small and large (bio)molecules and the estimation of their conformational ensembles. The description of the environment (solvent) has, therefore, a large impact. Implicit solvent representations are efficient but, in many cases, not accurate enough (especially for polar solvents, such as water). More accurate but also computationally more expensive is the explicit treatment of the solvent molecules. Recently, machine learning has been proposed to bridge the gap and simulate, in an implicit manner, explicit solvation effects. However, the current approaches rely on prior knowledge of the entire conformational space, limiting their application in practice. Here, we introduce a graph neural network based implicit solvent that is capable of describing explicit solvent effects for peptides with different compositions than those contained in the training set.

© 2023 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0147027>

09 November 2024 18:37:23

## I. INTRODUCTION

Molecular dynamics (MD) simulations employ Newton's equation of motion to study the dynamics of (bio)molecular systems.<sup>1</sup> In recent years, MD has not only become a pivotal tool for the investigation of biomolecular processes, such as membrane permeation of drug molecules or protein folding,<sup>2</sup> but also accelerated and supported drug discovery.<sup>3</sup> The conformational ensembles of molecules are strongly influenced by the surrounding medium. While intramolecular hydrogen bonds (H-bonds) are favored in vacuum and apolar solvents (e.g., chloroform), they are generally disfavored in polar solvents (e.g., water).<sup>4</sup> Such effects of the (local) environment can be incorporated by explicitly simulating solvent molecules. This approach provides good accuracy since it includes both short-range and long-range interactions. Both contributions are needed in order to describe an accurate conformational ensemble.<sup>5</sup> However, explicit-solvent simulations come at the cost of substantially increasing the number of degrees of freedom (DOFs) in the system, which results in substantially higher computational costs as well as slower effective sampling.<sup>6</sup> In addition,

the potential energy of a single solute conformation is no longer an instantaneous property in explicit-solvent simulations as an infinite number of solvent configurations (i.e., arrangement of the solvent molecules around the solute) exist for a single solute conformation. Thus, the calculation of the solvation free energy of a single conformer requires integrating out the contributions of individual solvent configurations.<sup>5</sup>

To simultaneously calculate the solvation free energy without the need to evaluate multiple solvent configurations and to reduce the number of DOFs in the system, implicit-solvent methods have been developed.<sup>7</sup> These approaches aim at modeling the solute–solvent interactions in an implicit manner by predicting the mean solvation energy and forces for a given solute conformation.<sup>6</sup> This free energy contribution of the solvent is often described by a separation into a polar and a non-polar contribution. While the polar contribution includes screening of charge–charge interactions as well as stabilization of polar groups, the non-polar contribution consists of the free energy required to create a cavity within the solvent to fit the solute as well as the dispersion effects between the solute and solvent.<sup>8</sup>

As the net effect of the non-polar part of the solvation free energy is generally much smaller than that of the polar part in polar solvents (e.g., water), it is often omitted in simulations.<sup>9</sup> This choice is also related to the fundamental limitations of the computationally efficient surface area (SA) models (see below), which cannot accurately describe the two counteracting non-polar contributions. For instance, pairwise non-polar interactions are known to be over-stabilized by current SA models.<sup>10</sup> Even though neglecting the non-polar part comes with theoretical drawbacks and its correct representation has been argued to be essential for the improvement of implicit-solvent models, approaches using a functional form designed to “only” account for the polar contributions with manually adjusted parameters have become widely used. Nguyen *et al.*<sup>11</sup> demonstrated that these models can reproduce a high percentage of the native conformational minima of a large set of proteins, while Rubenstein *et al.*<sup>12</sup> showed that these implicit solvents were able to match or exceed the performance of Rosetta scoring functions.

The most common approach for calculating the polar contribution of the solvation free energy replaces the solvent by a continuum electrostatic. Examples are Poisson–Boltzmann (PB),<sup>13</sup> generalized Born (GB),<sup>14</sup> or fast analytical continuum treatments of solvation (FACTS).<sup>15</sup> Note that the GB and FACTS models are approximations of the PB model. However, the current implicit-solvent models do not accurately reproduce the short-range effects of solvent molecules and, thus, often do not reproduce the secondary structures of peptides correctly.<sup>16</sup> Only very recently, the use of machine learning (ML) approaches has started to be explored for implicit-solvent models. Chen *et al.*<sup>17</sup> used graph neural networks (GNNs) to reproduce the potential energies and forces of two small peptides. This method is, however, not yet practical because the full conformational ensemble needs to be generated first via explicit-solvent simulations, before the GNN model can be trained to reproduce it. The GB models remain, therefore, the most commonly used implicit solvent approach to date.

The GB equation was first introduced by Still *et al.*<sup>14</sup> and defines the polar part of the solvation free energy  $\Delta G^P$  in terms of the atomic partial charges  $q_i$  and  $q_j$ , the distance  $r_{ij}$ , and the effective Born radii  $R_i$  and  $R_j$  of two atoms  $i$  and  $j$ ,

$$\Delta G^P = -\frac{1}{2} \left( \frac{1}{\epsilon_{in}} - \frac{1}{\epsilon_{out}} \right) \sum_{i,j} \frac{q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j} \exp\left(\frac{-r_{ij}^2}{4R_i R_j}\right)}. \quad (1)$$

The Born radii are further calculated using the Coulomb integral  $I_i$  and the intrinsic radius  $\rho_i$ ,

$$R_i = (\rho_i^{-1} - I_i)^{-1}. \quad (2)$$

The Coulomb integral can be derived from the Coulomb field approximation (CFA) and the intrinsic radius,

$$I_i = \frac{1}{4\pi} \int_{\Omega, r > \rho_i} \frac{1}{r^4} d^3 r. \quad (3)$$

Typically, the integral is solved analytically by using a pairwise de-screening approximation.<sup>18</sup> While the functional form of different GB models is the same, the manner in which this integral is calculated distinguishes them: GB-HCT,<sup>18</sup> GB-OBC,<sup>19</sup> or GB-Neck.<sup>20</sup>

In addition, ML has also been proposed to directly approximate reference Born radii calculated by PB calculations.<sup>21,22</sup>

Similarly, different models have been proposed for the non-polar part  $\Delta G^{NP}$  of the solvation free energy. A common choice is to approximate it by a surface area (SA) approach such that  $\Delta G^{NP}$  is proportional to the solvent-accessible surface area with

$$\Delta G^{NP} = \sum_i^N \sigma_i \cdot A_i, \quad (4)$$

where  $\sigma_i$  is the atomic effective surface tension coefficient and  $A_i$  is the atomistic solvent-accessible surface area of atom  $i$ .<sup>10</sup> One common approximation made is that  $\sigma_i$  is the same for all atoms, such that Eq. (4) simplifies to

$$\Delta G^{NP} = \sigma \cdot A, \quad (5)$$

where  $\sigma$  is the solvation parameter and  $A$  is the total surface area of the solute. A popular approach to estimate the surface area has been proposed by Schaefer *et al.*,<sup>23</sup> where the Born radii calculated for the GB models are also used for this task, making it a computationally attractive choice. In a slightly adapted form, this approach has been implemented in the OpenMM simulation software.<sup>24</sup> The expression is given in Eq. (6),

$$A = \sum_i^N 4\pi(r_i + r_p)^2 \cdot \left(\frac{r_i}{R_i}\right)^6, \quad (6)$$

where  $N$  is the number of atoms,  $r_i$  is the radius of atom  $i$ ,  $r_p$  is the radius of the probe, and  $R_i$  is the Born radius of atom  $i$ . This equation comes with the elegant feature that, in the limit of no screening ( $R_i = r_i$ ), the expression for each atom returns the surface area of the free ion in a given solvent and, in the limit of full screening ( $R_i = \infty$ ), Eq. (6) returns zero for an atom unreachable by the solvent.

The calculation of the effective Born radii in standard GB models can be thought of as a one-pass message passing algorithm, where information is sent from each node to all other nodes within a cut-off. GNNs are, therefore, a natural choice to further develop GB based models. When multiple passes are used, GNN can aggregate information, intrinsically encode the geometric environment, and thereby go beyond the pairwise de-screening approximation. In general, it is mainly the local environment, dominated by short-range interactions, that is expected to benefit from this description, as the GB models can describe long-range interactions well. Therefore, the robustness and quality of a GNN based implicit solvent could be enhanced by using a  $\Delta$ -learning<sup>25</sup> approach rather than predicting the solvation forces directly with the GNN. This means that rather than replacing the GB model entirely, an ML correction could be added to a base model (i.e., GB-Neck). In related fields, such as ML for QM/MM simulations of condensed-phase systems, a similar approach has been demonstrated to lead to stable simulations.<sup>26</sup> In addition, the  $\Delta$ -learning scheme could allow smaller cutoffs for the GNN as the long-range interactions are already well described by the GB model, reducing the computational cost.

Here, we use this idea to develop an ML-based implicit solvent approach, which can be employed to simulate molecules without the need for a complete conformational ensemble from explicit-solvent simulations for training. The training set for our ML approach consists of subsets of conformers extracted from explicit-solvent

simulations (here, with the TIP5P water model<sup>27</sup>), for which the mean solvation forces are calculated by keeping the solute position constrained and averaging over a multitude of solvent configurations. Note that this procedure to generate the solvation forces differs from the one proposed by Chen *et al.*,<sup>17</sup> as here the averaging is performed over multiple solvent configurations. The resulting mean solvation forces are subsequently used to train the GNN. In this context, it is important to note that the extracted forces contain the total solute–solvent interactions captured by the explicit solvent model and no separation into polar and non-polar force contributions is made. A GNN—*independent* of the chosen functional form—will, therefore, be trained to reproduce the total forces.

To assess the performance of our approach, test systems were chosen that (a) are interesting for the application of implicit solvents, (b) are challenging for current implicit solvents, (c) have fast kinetics, and (d) can be analyzed directly without dimensionality reduction or Markov state modeling. The aforementioned criteria are met by a class of small peptides, which are able to form a salt bridge. The importance of salt bridges for the stability of proteins<sup>28</sup> makes them an interesting target for implicit-solvent models. At the same time, previous studies by Nguyen *et al.*<sup>9</sup> have shown that GB-based implicit solvents could not accurately describe them. Although the parameters of the GB-Neck2 model<sup>9</sup> have been manually adjusted such that the height of the energy barrier matches a TIP3P solvent<sup>29</sup> simulation, this implicit solvent failed to reproduce other key characteristics of the system, such as the position of the free-energy minimum of the salt bridge or effects attributed to a single explicit water molecule between the opened salt bridge.

In this study, we investigate similar small peptides featuring a salt bridge between lysine (K, LYS) and glutamic acid (E, GLU) connected via two alanine (A, ALA) and one variable residue forming the peptides KAXAE, with X being valine (V, VAL), leucine (L, LEU), isoleucine (I, ILE), phenylalanine (F, PHE), serine (S, SER), threonine (T, THR), tyrosine (Y, TYR), and proline (P, PRO). In addition, we also test the approach on the same peptide KAAE as in Ref. 9, together with the longer variants KAAAE and KAAAAE. The performance is compared to the state-of-the-art implicit solvent GB-Neck2 as well as explicit-solvent simulations with TIP3P and TIP5P. To explore the generalizability and transferability characteristics of the GNN, the model is challenged to simulate peptides outside of the training set. Finally, to test the choice of functional form for the GNN, a simple SA based term to estimate the non-polar contribution to the solvation free energy is added to the GNN and its effect is evaluated.

## II. METHODS

### A. Molecular dynamics simulations

Starting coordinates and topologies were generated using the AmberTools21<sup>30</sup> software package. The amino acids and capping groups were parametrized using the AMBER force field ff99SB-ILDN.<sup>31</sup> All simulations were performed using OpenMM (version 7.7.0 for TIP3P, TIP5P, GB-Neck2, SA GB-Neck2, and GNN, or 8.0.0 for the GB-Neck2.mb3 and SA GB-Neck2.mb3 simulations).<sup>24</sup> The implicit solvents, including the SA term [Eq. (6)], were simulated by setting the SA variable in OpenMM to “ACE.” The default  $\sigma$  value of  $2.259 \text{ kJ mol}^{-1} \text{ nm}^{-2}$  was used. For the explicit-solvent simulations, the peptides were solvated in a box of water

(TIP3P or TIP5P) with a padding of 1 nm. For all systems, an energy minimization using the L-BFGS algorithm<sup>32</sup> was performed with a tolerance of  $10 \text{ kJ mol}^{-1} \text{ nm}^{-1}$ . All bonds involving hydrogens were constrained using the SETTLE<sup>33</sup> and constant constraint matrix approximation (CCMA)<sup>34</sup> algorithms for water and peptide bonds, respectively. For all simulations, Langevin dynamics were used with the LFMiddle discretization scheme.<sup>35</sup> For the explicit-solvent simulations, a cutoff of 1 nm for the long-range electrostatic interactions was used together with the particle mesh Ewald (PME) correction.<sup>36</sup> The simulation temperature was set to 300 K, and a time step of 2 fs was applied. The simulations with explicit solvent, or GB-Neck2, were carried out for 1  $\mu\text{s}$  and then split into five 200 ns blocks. Simulations with the GNN to describe the solvent were performed using the OpenMM-Torch package (version 0.6, url: <https://github.com/openmm/openmm-torch>) by introducing the GNN as an additional force to the vacuum force field. These simulations were carried out for 30 ns in triplicate. Note that explicit-solvent simulations with TIP3P were only performed for peptides KAPAE, KASAE, KATAE, and KAYAE.

### B. Generation of the training set

From the explicit-solvent simulations with TIP5P, a conformer was extracted every 200 ps for the peptides with X being apolar (i.e., KAVAE, KALAE, KAIAE, KAFAE, and KAPAE) and every 100 ps for the peptides with X being polar (i.e., KASAE, KATAE, and KAYAE). To calculate the mean solvation forces for each conformer, the solute atoms were positionally constrained and an explicit-solvent simulation was performed for 200 ps. The solvent–solute forces were evaluated every 0.2 ps by calculating the forces for each solute atom within the explicit simulation and subtracting the forces of the same solute atom in vacuum.

### C. Graph neural networks

Two GNN architectures were explored, sharing a three-pass neural network as the core architecture. Both GNNs were applied in the subsequent simulations via a  $\Delta$ -learning scheme.<sup>25</sup> In the first case (abbreviated as GNN+), the energies of the base model (i.e., GB-Neck2) and the GNN were summed up following a traditional  $\Delta$ -learning approach (the forces were then obtained via standard derivation). In the second case (abbreviated as GNN\*), the functional form of the base model (GB-Neck2) was adjusted such that the Born radii are scaled by the GNN within a predefined range according to Eq. (7),

$$R'_i = R_i * (p + S(\phi(R, q, r_a, r, R_{\text{cutoff}}))) \cdot (1 - p) \cdot 2, \quad (7)$$

where  $R_i$  is the Born radius calculated based on the Neck integral,  $p$  is the scaling parameter to adjust the strength of the scaling of the Born radii (i.e., 1 = no scaling applied; 0 = maximum scaling applied),  $S$  is the sigmoid function, and  $\phi$  is the function approximated by the GNN based on the Born radii  $R$ , the charges  $q$ , the atomic radii  $r_a$  of all atoms, the distances  $r$  between all atoms, and a cutoff radius  $R_{\text{cutoff}}$ . The SA GNN\* model was implemented in the same way as the GNN\* model with the additional SA term defined in Eq. (6).

The GNN\* and GNN+ networks share key architectural elements as both employ three passes through interaction networks that only differ in the shape of the input and output, followed by SiLU activations.<sup>37</sup> One interaction network pass is characterized by

a concatenation of the node features of the sending and receiving nodes together with the distance encoded by a radial Bessel<sup>38</sup> function of length 20, followed by a multi-layer perceptron (MLP) with two layers and SiLU activation functions. As an aggregation method, the sum of all messages was taken and the node-wise computation employed again a two-layer MLP with SiLU activation functions. All hidden layers have a size of 128. As an embedding for the atoms, the GNN+ model used the partial charges and atomic radii from the force field and GBNeck2, respectively, while the GNN\* model additionally incorporates the calculated Born radius from the Neck integral. A schematic representation of the GNN architectures is shown in Fig. 1.

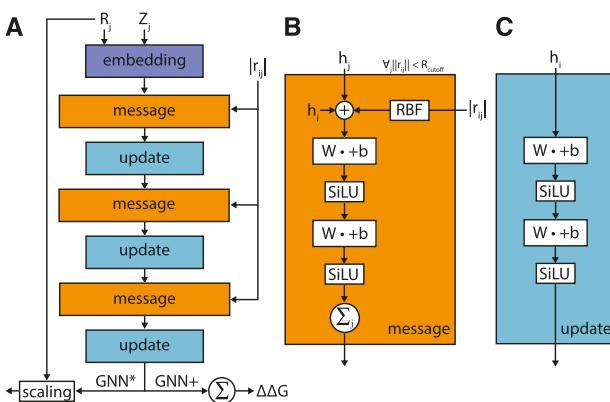
The GNNs were trained on randomly selected 80% of the conformations, using the Adam optimizer<sup>39</sup> for 100 epochs with a batch size of 32 and an exponentially decaying learning rate starting at 0.001 and decaying by two orders of magnitude to 0.00001. The mean squared error (MSE) was chosen as the loss function, and the samples were randomly shuffled after each epoch.

#### D. Training-test splits

To assess the generalizability and transferability of the GNN, we composed different training and test splits (Table I). Splits 1–4 include all KAXAE peptides for training but one. In split 5, all KAXAE peptides with X = polar are used for training, while the ones with X = apolar are simulated prospectively. Split 6 is the other way around (i.e., training on X = apolar; testing on X = polar). Finally, in split 7, all peptides with five residues (except X = A) were used for training, while prospective simulations were performed for KAAAE, KAAAE, and KAAAAE.

#### E. Data analysis

We used MDTraj (version 1.9.7)<sup>40</sup> for the analysis of the trajectories. To estimate the statistical uncertainty of the different

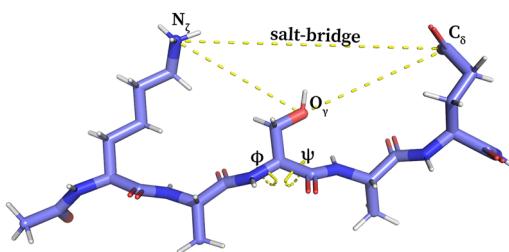


**FIG. 1.** Schematic representation of the GNN architectures. Node-wise computations are shown in blue, while message operations are shown in orange. (a) Schematic representation of the computation through the GNN. (b) Message computations start by concatenating the atom features of the sending and receiving nodes with the distance encoded by a Bessel function (RBF), followed by a two-layer MLP with SiLU activation functions. (c) Node-wise computations of aggregation by summation followed by a two-layer MLP with SiLU activation functions.

**TABLE I.** Training and test splits of peptides KAXAE with X being V, L, I, F, P, S, T, Y, "A, and AA (note that KAXAE with X = " corresponds to KAAAE). The peptides used for training are marked with , while the peptides simulated prospectively (testing) are marked with .

Split	V	L	I	F	P	S	T	Y	"	A	AA
1											
2											
3											
4											
5											
6											
7											

approaches, the explicit-solvent simulations were divided into five 200 ns blocks and analyzed separately. In addition, the training and simulation of the GNN were repeated three times with different random seeds. All free-energy profiles were calculated with a Jacobian correction factor of  $4\pi r^2$ .<sup>41</sup> A key feature that the ML-based implicit solvent should reproduce is the correct representation of the salt bridge between LYS N $\zeta$  and GLU C $\delta$ . We have identified three main characteristics concerning the salt bridge for all test systems (Fig. 2): (i) a double well in the free-energy minimum at 0.33 and 0.37 nm with different weights, corresponding to two different closed salt-bridge geometries; (ii) the height of the energy barrier for the opening of the salt bridge at 0.43 nm; and (iii) a dip in the free-energy profile at 0.55 nm, corresponding to conformations with one water molecule between LYS N $\zeta$  and GLU C $\delta$  in a hydrogen-bond network. In addition to the salt bridge, the backbone dihedral angles  $\phi$  and  $\psi$  of the central amino acid are monitored. For polar central residues, the distance between the oxygen of the hydroxy group of the polar side chains and the LYS N $\zeta$  and GLU C $\delta$  is monitored (Fig. 2).



**FIG. 2.** Structural characteristics of KASAE as an example (light blue) with the highlighted salt bridge, the distances SER O<sub>y</sub>–LYS N<sub>z</sub> and SER O<sub>y</sub>–GLU C $\delta$ , and the backbone torsional angles  $\phi$  and  $\psi$  of the central residue.

### III. RESULTS AND DISCUSSION

#### A. Implicit-solvent models with and without non-polar term

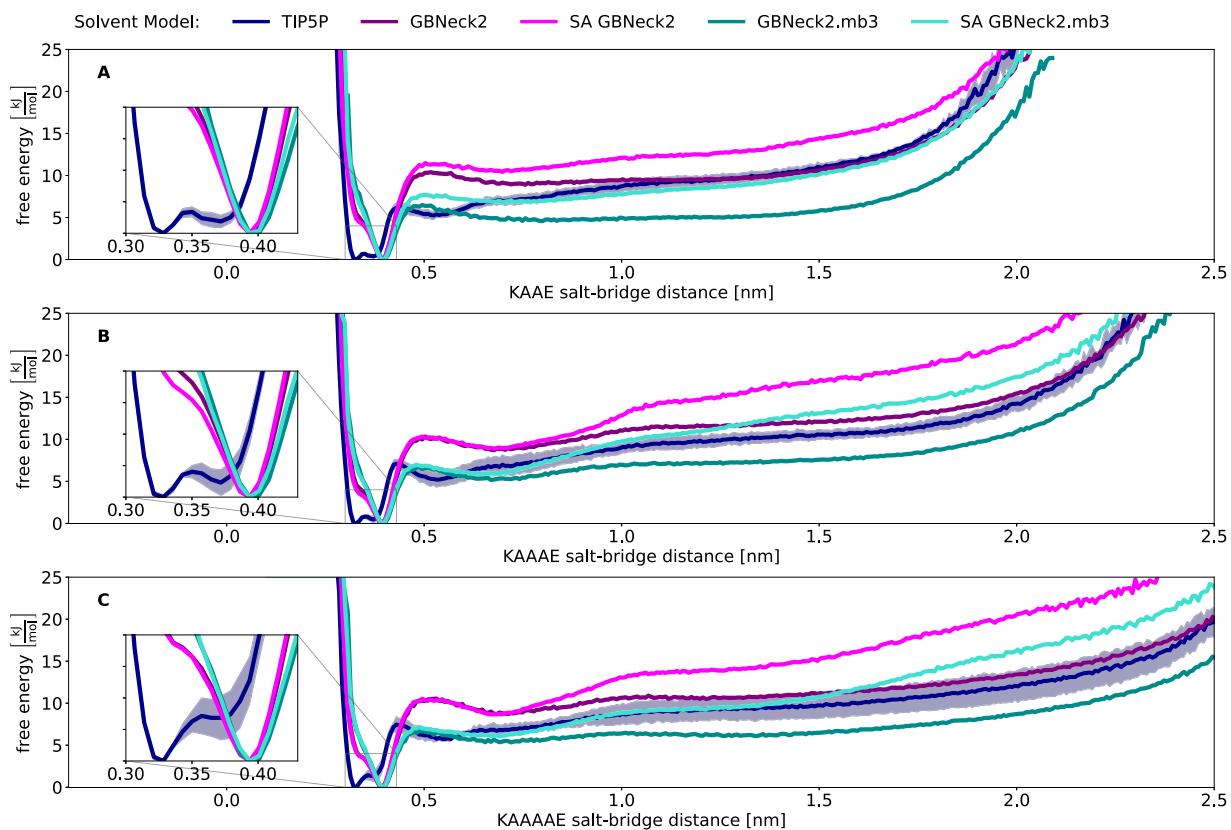
Implicit-solvent models, such as GB-Neck2, are often used, in practice, without a non-polar term.<sup>9,11,12,16</sup> To better understand the effect of the non-polar contribution in implicit-solvent models via a SA term [Eq. (6)], we compared four different models for reproducing the free-energy profile of the salt-bridge in the test peptides: (1) GB-Neck2, (2) GB-Neck2 with a SA term (SA GB-Neck2), (3) GB-Neck2 with the mbondi3 radii set from Ref. 9 (GB-Neck2.mb3), and (4) GB-Neck2 with the mbondi3 radii set plus the SA term (SA GB-Neck2.mb3). We chose the three peptides KAAE, KAAAE, and KAAAAE for this analysis to assess the long-range behavior of the implicit-solvent models. The results are shown in Fig. 3.

While none of the tested implicit-solvent models can correctly reproduce the double well in the free-energy minimum of the salt bridge (0.33–0.37 nm), the barrier height for the opening of the salt bridge (at 0.43 nm) is dominated by the radii set used, with GB-Neck2.mb3 being closer to the TIP5P reference due to manual adjustment of the radii (see Ref. 9). Only at longer distances (>0.60 nm) is an effect of the SA term visible, i.e., more extended

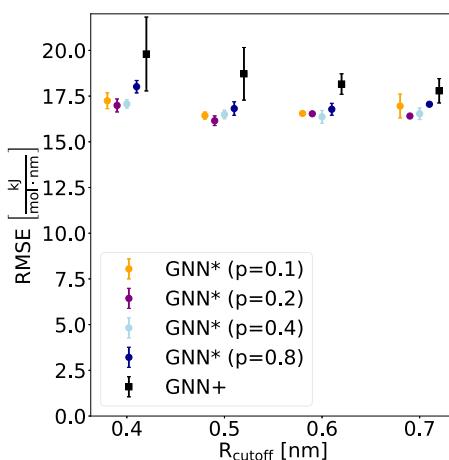
conformations are less favorable when the SA term is included. This observation is in line with the findings by Chen and Brooks<sup>10</sup> that pairwise non-polar interactions are over-stabilized by current SA models and, hence, lead to more compact structures. Since the GB-Neck2 model without an SA term and the original radii set without manual adjustments showed the best performance for long-range interactions (purple line in Fig. 3), we selected this model as our baseline and for the development of an ML-based implicit solvent.

#### B. Comparison of CNN architectures

We explored two GNN architectures with  $\Delta$ -learning schemes for the ML-based implicit-solvent model: GNN+ and GNN\*. To compare the architectures and evaluate their hyperparameters, we calculated the solvation forces of simulation snapshots in a retrospective manner. We considered three peptides (KAAE, KALAE, and KASAE) for training (using a random 20% subset as a validation set during the GNN training process) and two peptides (KATAE and KAAAE) as external test sets. The peptides KAIAE, KAPAE, and KAYAE were not included in this benchmark in order to not bias the choice of model architecture for the subsequent simulation studies. The investigated hyperparameters were the cutoff radius  $R_{\text{cutoff}}$  for



**FIG. 3.** Comparison of the implicit-solvent models GB-Neck2 (purple), SA GB-Neck2 (magenta), GB-Neck2.mb3 (dark cyan), and SA GB-Neck2.mb3 (turquoise), as well as the explicit solvent model TIP5P (navy), to simulate the free-energy profile of the salt bridge for KAAE (a), KAAAE (b), and KAAAAE (c). MD simulations were performed for 1  $\mu$ s and split into five blocks of 200 ns. The standard deviation over the blocks of the TIP5P solvent model is indicated by the shaded area.



**FIG. 4.** Comparison of the root-mean-square error (RMSE) of the forces predicted by the GNN\* and GNN+ models with  $\Delta$ -learning for the external test set. The GNN\* models with the scaling parameter  $p$  of 0.1, 0.2, 0.4, or 0.8 are shown as colored dots (orange, purple, light blue, and navy, respectively), while the GNN+ is shown as black squares. The standard deviation over the replicates is denoted by error bars.

which the fully connected graph is constructed, as well as the scaling parameter  $p$  of the GNN\*, which regulates how much the Born radii of the GB-Neck2 model are allowed to change. The results of the benchmarking are summarized in Fig. 4.

Interestingly, the GNN\* model was found to perform significantly better than the GNN+ model in predicting the forces of the external test set over the entire range of tested cutoff radii, reaching an RMSE of  $16.2(3)$   $\text{kJ mol}^{-1} \text{nm}^{-1}$ . In addition, smaller deviations between the different random seeds were observed with the GNN\* (indicated by lower standard deviations), which is a desirable feature. The effect of the scaling parameter on the GNN\* models is more subtle. Values of 0.1, 0.2, and 0.4 gave essentially the same

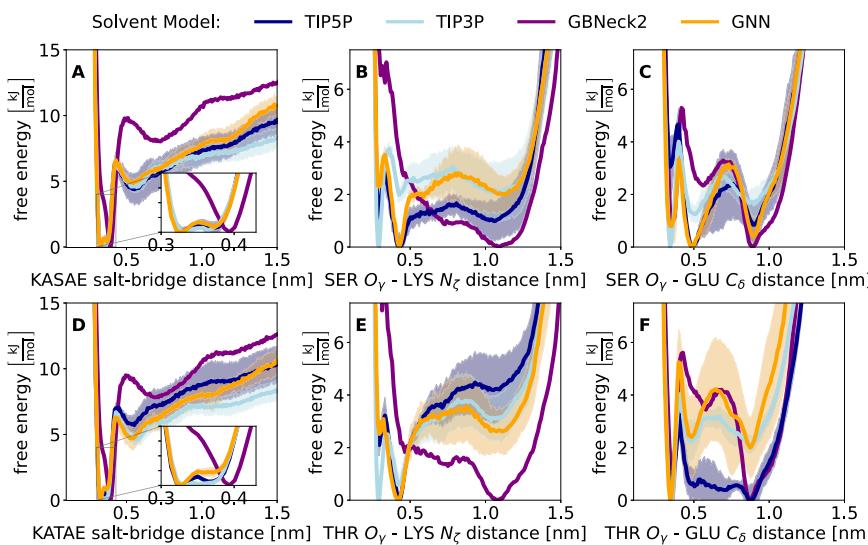
results, while the error increased slightly for a scaling parameter of 0.8. The impact of the cutoff radius,  $R_{\text{cutoff}}$ , is also small, although  $R_{\text{cutoff}} = 0.4$  nm is likely too short. As longer radii (i.e., 0.7 nm) did not improve the performance significantly but increased the computational cost, we decided to focus in the following on the GNN\* architecture with cutoff radii of 0.5 and 0.6 nm together with scaling parameters of 0.1 and 0.2.

### C. Prospective molecular dynamics simulations

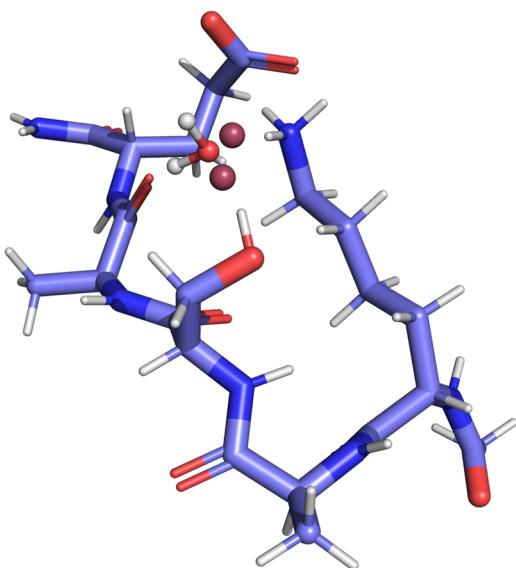
To investigate the ability of the ML-based implicit solvent to simulate novel peptides, we composed different training and test splits (Table I). First, we assessed the simulation performance of the GNN\* models with radii = 0.5 or 0.6 nm and scaling parameters = 0.1 or 0.2 on the training/test splits 1, 2, and 3 (i.e., training on all peptides except KASAE, KATAE, or KAYAE, respectively, and prospective simulation of the left-out peptide). The GNN\* model with a radius of 0.6 nm and a scaling parameter of 0.1 yielded the most stable simulation results, as indicated by the smallest deviations between the different random seeds for the closed salt-bridge conformations (see Figs. S1–S4 in the supplementary material). The observation that models with a similar performance on a retrospective test set show more different behaviors in prospective simulations is in line with the findings by Fu *et al.*<sup>42</sup> and Stocker *et al.*<sup>43</sup> Based on these results, the following analyses were performed using only the GNN\* model with a radius of 0.6 nm and a scaling parameter of 0.1.

### 1. KASAE, KATAE, or KAYAE as test peptide

The training/test splits 1–3 (Table I) are particularly interesting as they challenge the model the most. Among the three, the simulations of KASAE (split 1) and KATAE (split 2) are expected to be easier for the model as there is a similar residue in the training set (THR vs SER). Free-energy profiles of the salt bridge, the  $O_y$ –LYS  $N_\zeta$  distance, and the  $O_y$ –GLU  $C_\delta$  distance are shown in Fig. 5.

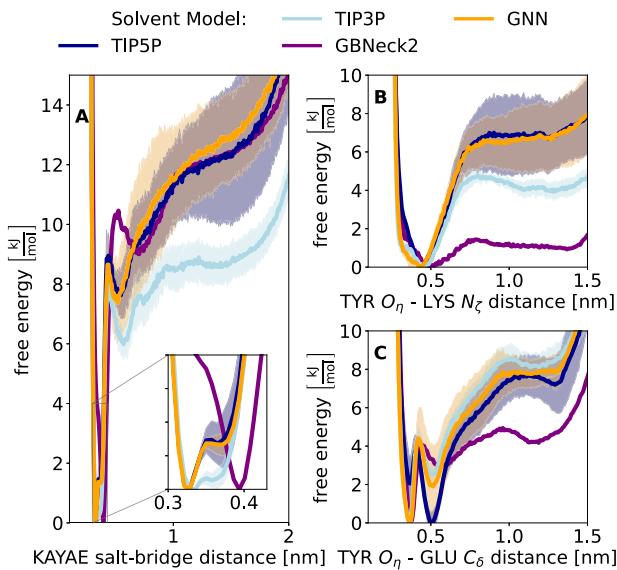


**FIG. 5.** Comparison of the GNN\* implicit-solvent model (orange, 3  $\times$  30 ns) with the explicit TIP5P (navy blue, 5  $\times$  200 ns) and TIP3P (light blue, 5  $\times$  200 ns) and the GB-Neck2 implicit solvent (purple, 5  $\times$  200 ns). The results for KASAE (split 1) are shown in the top row, and the results for KATAE (split 2) are shown in the bottom row. [(a) and (d)] Free-energy profile of the salt bridge. [(b) and (e)] Distance  $O_y$ –LYS  $N_\zeta$ . [(c) and (f)] Distance  $O_y$ –GLU  $C_\delta$ . The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model (not shown for GB-Neck2 for clarity).



**FIG. 6.** Example conformation of KASAE (light blue) with a SER O<sub>γ</sub>–LYS N<sub>ζ</sub> distance of 0.43 nm. The TIP5P water molecule interacting with the SER O<sub>γ</sub>, the LYS N<sub>ζ</sub>, and a carbonyl of the backbone is shown with its off-site charges displayed in dark red.

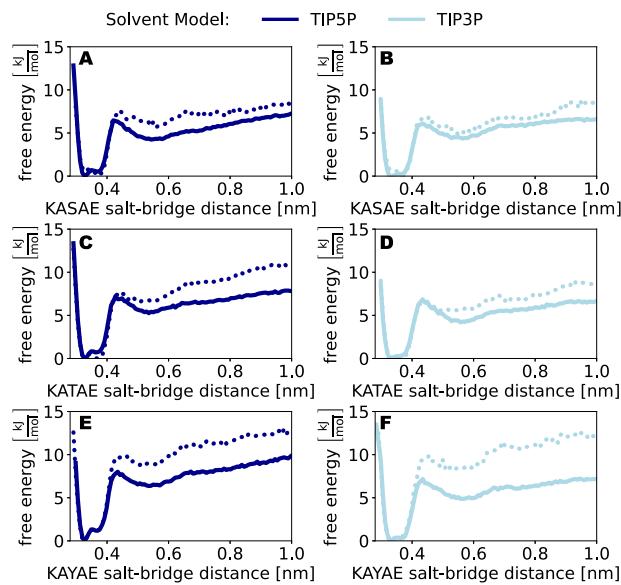
The GNN\* implicit solvent was able to correctly reproduce all desired properties of the salt bridge in TIP5P explicit water, featuring the correct double well in the free-energy profile at short distances, the correct energy barrier of opening for the salt bridge,



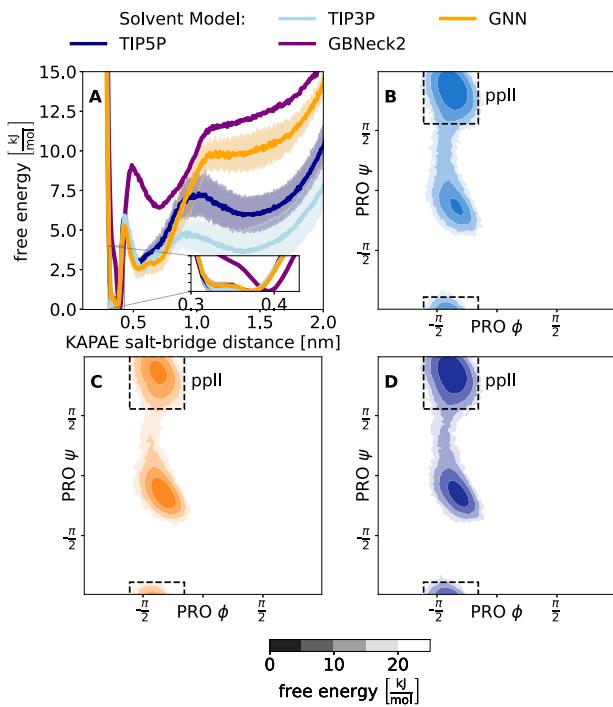
**FIG. 7.** Comparison of the GNN\* implicit-solvent model (orange, 3 × 30 ns) with the explicit TIP5P (navy blue, 5 × 200 ns) and TIP3P (light blue, 5 × 200 ns) and the GB-Neck2 implicit solvent (purple, 5 × 200 ns). The results for KAYAE (split 3) are shown. (a) Free-energy profile of the salt bridge. (b) Distance TYR O<sub>γ</sub>–LYS N<sub>ζ</sub>. (c) Distance TYR O<sub>γ</sub>–GLU C<sub>δ</sub>. The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model (not shown for GB-Neck2 for clarity).

and a local minimum at 0.55 nm [Figs. 5(a) and 5(d)]. As discussed earlier, the GB-Neck2 model fails to describe these short-range features. Larger deviations can be observed for the SER/THR O<sub>γ</sub>–LYS N<sub>ζ</sub> distance [Figs. 5(b) and 5(e)]. Again, the GB-Neck2 model is not able to capture the key characteristics of the TIP5P free-energy surface, whereas the GNN\* reproduces the minima correctly and shows good agreement with the local minima of the direct hydrogen bond between O<sub>γ</sub> and N<sub>ζ</sub> at 0.29 nm. Interestingly, the TIP3P model shows quite different characteristics compared to TIP5P. The minima of the direct hydrogen bond are lower than the second minima at 0.43 nm. The latter minima contain conformations, where an explicit water molecule forms a hydrogen bond network between the salt-bridge partners. Analyzing similar conformations in the TIP5P solvent simulation, we could identify examples that benefit from the specific geometry of the TIP5P water representation (Fig. 6). We hypothesize that the “triangle” between the SER O<sub>γ</sub>, the LYS N<sub>ζ</sub>, and a carbonyl of the peptide backbone allows for a trident “binding” site for the explicit TIP5P water molecules, thus stabilizing these conformations, which is not possible with the simplified TIP3P model.

A more challenging case is KAYAE (split 3) as none of the amino acids in the training set is very similar to TYR. Thus, the model needs to learn about the solvation of the TYR side chain from different amino acids, i.e., the model has to generalize from PHE and SER/THR to a combination of the two. If the model achieves a good accuracy for split 3, it demonstrates its transferability to peptides with increasing structural differences in the training set. The free-energy profile of the KAYAE salt bridge and the



**FIG. 8.** Free-energy profile of the salt-bridge distance for KASAE (top), KATAE (middle), and KAYAE (bottom) in the explicit-solvent simulations with TIP5P {left [(a), (c), and (e)] navy blue, 5 × 200 ns} and TIP3P {right [(b), (d), and (f)] light blue, 5 × 200 ns}. The dotted line corresponds to conformations with the distance between O of the hydroxy group and LYS N<sub>ζ</sub> < 0.60 nm, and the solid line corresponds to conformations with the distance between O of the hydroxy group and LYS N<sub>ζ</sub> > 0.60 nm.



**FIG. 9.** Comparison of the GNN\* implicit-solvent model (orange,  $3 \times 30$  ns) with explicit TIP5P (navy blue,  $5 \times 200$  ns) and TIP3P (light blue,  $5 \times 200$  ns) and the GB-Neck2 implicit solvent (purple,  $5 \times 200$  ns). The results for KAPAE (split 4) are shown. (a) Free-energy profile of the salt bridge. The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model (not shown for the GB-Neck2 implicit solvent for clarity). (b) Ramachandran plot of PRO with GB-Neck2. (c) Ramachandran plot of PRO with GNN\*. (d) Ramachandran plot of PRO with TIP5P. The polyproline II state is highlighted in the Ramachandran plots by a dashed black line.

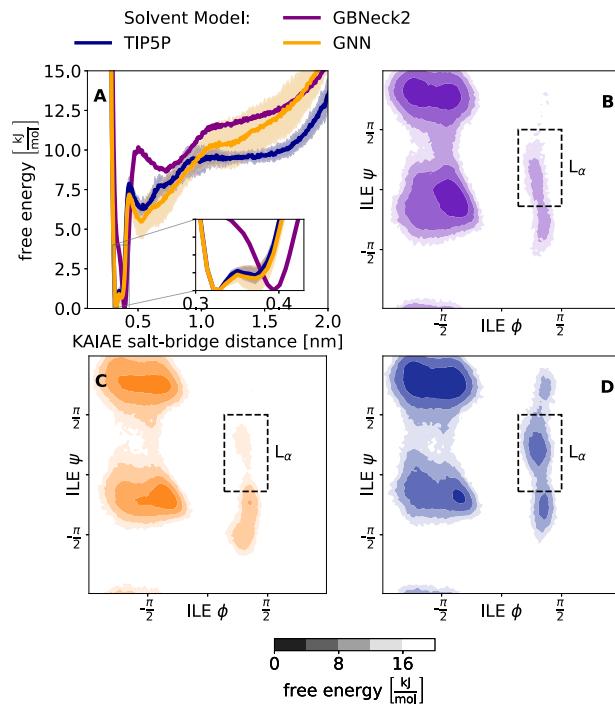
distances TYR  $O_{\eta}$ -LYS  $N_{\zeta}$  and TYR  $O_{\eta}$ -GLU  $C_{\delta}$  are shown in Fig. 7. Again, the GNN\* model reproduces the salt-bridge free-energy profile of TIP5P very well [Fig. 7(a)]. Interestingly, the TIP3P solvent shows in this case a different behavior than TIP5P at short distances. The double well of the salt bridge is significantly different, and the barrier height (at 0.43 nm) is lower for TIP3P. For the distance TYR  $O_{\eta}$ -LYS  $N_{\zeta}$ , the TIP5P and GNN\* show the same behavior, while the energy barrier with TIP3P is  $\sim 3$  kJ mol $^{-1}$  lower [Fig. 7(b)]. For the distance TYR  $O_{\eta}$ -GLU  $C_{\delta}$ , the minimum with TIP3P and GNN\* is the direct H-bond distance at 0.37 nm, while the minimum with TIP5P is at a distance of 0.51 nm, where one explicit water interacts in a H-bond network between the salt-bridge partners [Fig. 7(c)].

While the free-energy profiles of the salt bridge were similar for TIP5P and TIP3P for KASAE and KATAE, they differed for KAYAE. Therefore, we investigated this observation further. The difference in barrier height for opening the salt bridge [Fig. 7(a)] could be due to different preferences for the distance TYR  $O_{\eta}$ -LYS  $N_{\zeta}$ . In Fig. 8, the free-energy profile of the salt bridge for KASAE, KATAE, and KAYAE was calculated once for conformations with the distance between O of the hydroxy group and LYS  $N_{\zeta} < 0.6$  nm and once for those with this distance  $> 0.6$  nm. Intriguingly, the energy barrier

to open the salt bridge of KAYAE is higher when TYR  $O_{\eta}$  interacts with the LYS  $N_{\zeta}$  (e.g., either via a direct hydrogen bond or mediated by a water molecule) in both TIP5P and TIP3P water. For KASAE and KATAE, this is not the case. From these findings, it follows that because TIP5P and GNN\* favor conformations of KAYAE with a distance TYR  $O_{\eta}$ -LYS  $N_{\zeta} < 0.6$  nm more than TIP3P [Fig. 7(b)], the energy barrier of the salt bridge is higher than in TIP3P.

## 2. Proline as special case: KAPAE

An interesting case is PRO as the central residue (split 4 in Table I). Proline has a different Ramachandran plot than the other amino acids and is, therefore, possibly challenging to describe correctly by the GNN\* approach. As shown in Fig. 9(a), the free-energy profile of the salt bridge is reproduced well, although deviations from TIP5P occur at long distances. Interestingly, at these long distances, TIP3P and TIP5P also disagree with each other. While TIP3P yields lower free energies for the open state compared to TIP5P, the GNN\* predicts even higher free energies. As PRO is in the middle of the peptide, the sampling of its backbone dihedral angles will influence the long salt-bridge distances. The Ramachandran plots for PRO are shown in Figs. 9(b)-9(d) for TIP3P, GNN\*, and TIP5P. The main difference is in the population of the polyproline II state, which is



**FIG. 10.** Comparison of the GNN\* implicit-solvent model (orange,  $3 \times 30$  ns) with explicit TIP5P (navy blue,  $5 \times 200$  ns) and TIP3P (light blue,  $5 \times 200$  ns) and the GB-Neck2 implicit solvent (purple,  $5 \times 200$  ns). The results for KAIAE (split 5) are shown. (a) Free-energy profile of the salt bridge. The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model (not shown for the GB-Neck2 implicit solvent for clarity). (b) Ramachandran plot of ILE with GB-Neck2. (c) Ramachandran plot of ILE with GNN\*. (d) Ramachandran plot of ILE with TIP5P. The L $\alpha$  state is highlighted in the Ramachandran plots by a dashed black line.

overestimated with TIP3P and underestimated with GNN\* compared to TIP5P (see also Fig. S5 in the supplementary material). It is, therefore, important to include PRO in the training set to ensure proper sampling of its backbone conformations.

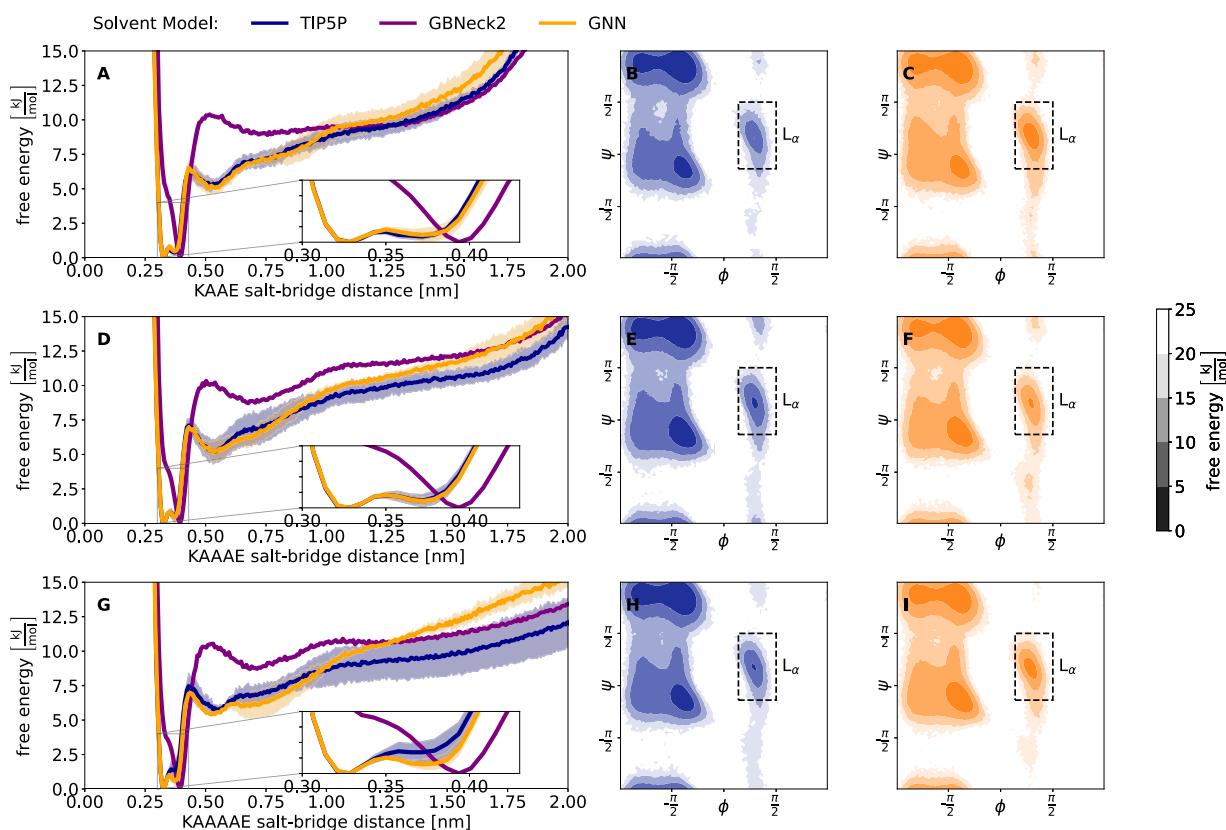
### 3. Apolar vs polar residues

The results above demonstrated that the GNN\* model is able to reproduce the key characteristics of explicit-water simulations of peptides different from the training set. Next, we investigated how different training set compositions influenced the simulation performance of the model. We, therefore, created two training sets (splits 5 and 6 in Table I) by dividing all peptides into two excluding subsets: (1) the central residue has a polar side chain (i.e., KASAE, KATAE, and KAYAE) and (2) the central residue has an apolar side chain (i.e., KAAAE, KAIAE, KALAE, KAFAE, and KAPAE). In split 5, training was carried out with the first group and prospective simulations were performed for the second group. As shown in Fig. 10 for KAIAE, the generalization from the peptides with a polar central residue to those with an apolar one is good. For all tested peptides, the GNN\* model is able to reproduce the free-energy profile of the salt bridge, the double well minima, the height of the energy barrier

for opening, and the first dip of the reference TIP5P simulation. The corresponding results for KAAAE, KALAE, KAFAE, and KAPAE are shown in Figs. S6–S9 in the supplementary material.

To probe the conformational sampling of the central residue in more detail, we compared its Ramachandran plot for the different solvent descriptions. With the exception of the  $L_\alpha$  state, the Ramachandran plots with GNN\* and TIP5P agree well. Note that the transition into the  $L_\alpha$  state is a rare event. In the TIP5P reference simulations of KAIAE, this state is sampled in only one of the five 200 ns blocks (see Figs. S10–S14 in the supplementary material). The differences in the population of the  $L_\alpha$  state between GNN\* and TIP5P may, therefore, stem from finite sampling effects.

The inverse, i.e., training on peptides with an apolar central residue and testing on peptides with a polar central residue (split 6), is more challenging. The GNN\* model was still superior to the GB-Neck2 implicit solvent in reproducing key characteristics of the TIP5P reference simulations; however, deviations were observed for the interactions of the polar central residue with the salt-bridge partners (i.e., the distance between  $O$  of the hydroxy group and LYS  $N_\zeta$ /GLU  $C_\delta$ ) (see Figs. S15–S17 in the supplementary material). These results indicate that the extent to which the GNN\* model



**FIG. 11.** Comparison of the GNN\* implicit-solvent model (orange, 3 × 30 ns) with explicit TIP5P (navy blue, 5 × 200 ns) and the GB-Neck2 implicit solvent (purple, 5 × 200 ns). The results for KAAE (top), KAAAE (middle), and KAAAAE (bottom) are shown (split 7). [(a), (d), and (g)] Free-energy profile of the salt bridge. The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model (not shown for the GB-Neck2 implicit solvent for clarity). [(b), (e), and (h)] Combined Ramachandran plot of all ALA residues with TIP5P. [(c), (f), and (i)] Combined Ramachandran plot of all ALA residues with GNN\*. The  $L_\alpha$  state is highlighted by a dashed black line.

can generalize from the training set is limited to similar functional groups. For instance, if no hydroxy group is present in the training set, the ability of the model to represent its interactions is limited. On the other hand, the TYR case demonstrates that the model is able to generalize from a hydroxy group in SER/THR to one in a different local environment. Taken together, these findings suggest that it is important for a generally applicable GNN\* implicit-solvent approach to include all functional groups in the training set, but that the model does not have to have seen the complete molecule for good performance.

#### 4. Varying the length of the peptide

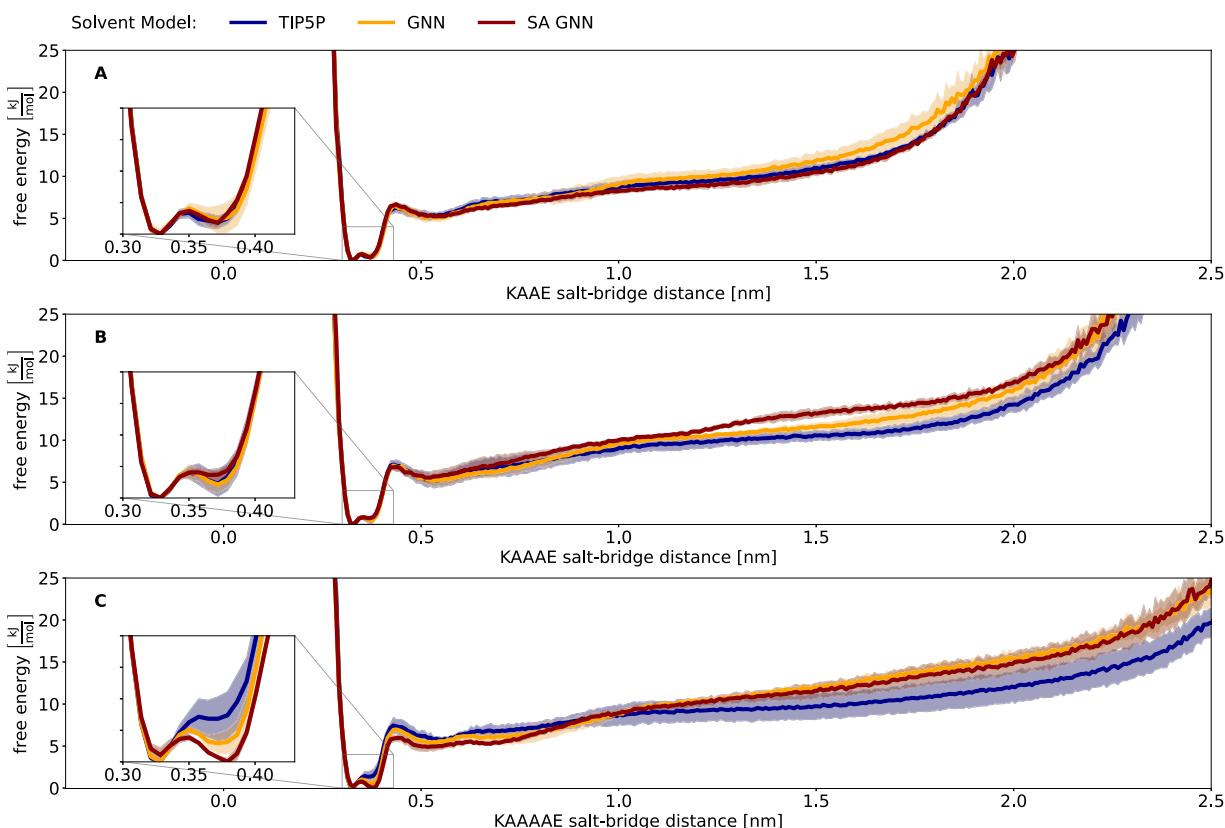
Finally, we investigated whether the GNN\* model is able to generalize to larger or smaller peptides by removing the middle amino acid or instead inserting an extra ALA residue, i.e., peptides KAAE and KAAAAE (split 7 in Table I). In addition, we included KAAAE (the same size) in the test set for comparison. The resulting free-energy profiles and Ramachandran plots for KAAE, KAAAE, and KAAAAE are shown in Fig. 11. For all three peptide lengths, the GNN\* is able to reproduce the free-energy profile of the salt bridge of the TIP5P reference simulation for short distances (i.e., <1 nm), including the double well, the height of the energy

barrier, and the first dip. For the KAAE and KAAAE cases, the long-range behavior also matches the TIP5P simulation to a high degree. Only for KAAAAE, a deviation between GNN\* and TIP5P is observed at longer distances (i.e., >1 nm), which could highlight a potential weak point of the GNN\*. While generalization to shorter peptides works well, longer peptides require either inclusion in the training set or the introduction of a long-range correction in order to describe the elongated conformations accurately. As discussed earlier, differences in the population of the  $L_\alpha$  state may likely be due to finite sampling effects.

#### D. Effect of the functional form of the GNN\* model

To investigate whether the deviation between the GNN\* model and the TIP5P reference simulations at long distances shown in Fig. 11 could be due to the neglected SA term in the GNN\* architecture, we have trained a new GNN\* model with the SA GB-Neck2 functional form (SA GNN\*) and tested it in prospective MD simulations for split 7 (Fig. 12).

For all three peptides, the simulations of both GNN architectures produce almost identical results. While the SA term of the SA GNN\* architecture does not seem to improve the model, it also does not disfavor more extended conformations (distances >0.60 nm) as



**FIG. 12.** Comparison of the implicit-solvent models GNN\* (orange, 3 × 30 ns) and SA GNN\* (dark red, 3 × 30 ns) with explicit TIP5P (navy blue, 5 × 200 ns). The results for KAAE (a), KAAAE (b), and KAAAAE (c) are shown (split 7). The shaded area indicates the standard deviation over the blocks or replicates of the corresponding solvent model.

shown for the SA GB-Neck2 models (Fig. 3). As both GNNs are able to learn the total solvent force (polar plus non-polar) from the training data, it suggests that the GB-Neck2 functional form by itself can capture parts of the non-polar effects with careful parametrization. However, it may also be the case that Eqs. (5) and (6) are simply not expressive enough and that a different (more accurate) functional form for the non-polar contribution leads to a better performance. This possibility will be investigated in future work.

### 1. Timings

One major advantage of standard implicit-solvent models is that they are much faster to compute than explicit solvent molecules. When employing GNNs for this task, the computational costs are currently still too high. Using a desktop PC with an Intel(R) Xeon(R) W-1270P central processing unit (CPU) with a clock rate of 3.80 GHz and an NVIDIA(R) Quadro(R) P2200 graphics processing unit (GPU),  $\sim$ 46 ns d $^{-1}$  of the peptide KASAE could be obtained with our proof-of-concept implementation of the GNN implicit solvent, whereas  $\sim$ 200 ns d $^{-1}$  was reached with explicit TIP5P simulations. Similar observations were made in Ref. 44 for classical force-field terms. The slower speed of GNNs represents a major challenge for their application to replace explicit-solvent simulations. However, this is primarily a technical issue and not a fundamental limitation. While the TIP5P explicit simulation is highly optimized, our GNN implementation is not yet. Currently, the GNN is evaluated on the GPU, while the classical forces are evaluated on the CPU, leading to a high communication cost and low utilization of the GPU. Recently, two approaches have been reported to dramatically increase the speed of NN potentials. The first option is the optimization of the operations of the GNN to better suit the application of MD simulations.<sup>45</sup> The second option involves batching of multiple simulations that run on one GPU in parallel.<sup>17</sup> Both approaches have been shown to bring the speed of NN potentials on a par with their classical counterparts. In this work, we focused on providing a conceptual proof-of-concept that developing an ML-based transferable implicit solvent is possible. Improving the computational performance of the implementation is part of the future work to develop a practically usable GNN implicit solvent.

## IV. CONCLUSION

In this work, we have developed a GNN-based implicit solvent that can be trained on a set of peptides and used to prospectively simulate peptides with different compositions. The GNN\* model is based on the functional form of the GB-Neck2 implicit solvent with a  $\Delta$ -learning scheme. To validate our approach, we have chosen a traditionally hard problem for implicit-solvent models where the local effects of explicit water molecules play a key role: the free-energy profile of a salt bridge. Here, the salt bridge is formed by peptides with the composition KAXAE, where X can be varied. We could demonstrate that the GNN\* implicit solvent was able to reproduce the key characteristics of the reference explicit-solvent simulations with TIP5P, matching or surpassing the accuracy of explicit-solvent simulations with the simpler TIP3P water model. With different training/test splits, we assessed the ability of the GNN\* model to generalize to unseen amino acids and varying peptide lengths. In addition, the usage of an explicit SA term in the functional form of the GNN\* architecture was investigated but found to not improve

the performance for the given test set. Overall, we found that the model has a high transferability as long as all functional groups are represented in the training set. For instance, if an aliphatic hydroxy group (SER or THR) is in the training set, it is sufficient for the model to correctly describe the aromatic hydroxy group of TYR. These findings are encouraging as they suggest that the training set for a globally applicable ML-based implicit-solvent model may not need to be extremely large but “only” contain all necessary functional groups. The results of this work present an important step toward the development of such a model capable of replacing explicit-solvent simulations to an increasing degree. Future work will explore not only the extension of the approach to different systems but also the treatment of non-polar solute–solvent interactions.

## SUPPLEMENTARY MATERIAL

See the supplementary material for additional Figs. S1–S17 referenced in the text.

## ACKNOWLEDGMENTS

The authors acknowledge the financial support from ETH Zurich (Research Grant No. ETH-50 21-1). The authors thank Moritz Thürlemann for helpful discussions.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

**Paul Katzberger:** Conceptualization (equal); Data curation (lead); Formal analysis (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (supporting). **Sereina Riniker:** Conceptualization (equal); Funding acquisition (lead); Project administration (lead); Supervision (lead); Writing – original draft (supporting); Writing – review & editing (lead).

## DATA AVAILABILITY

The data that support the findings of this study are openly available at <https://github.com/rinikerlab/GNNImplicitSolvent> and are openly available in ETH Research Collection at <https://doi.org/10.3929/ethz-b-000599309> and also available from the corresponding author upon reasonable request.

## REFERENCES

1. T. Hansson, C. Oostenbrink, and W. F. van Gunsteren, “Molecular dynamics simulations,” *Curr. Opin. Struct. Biol.* **12**, 190–196 (2002).
2. M. Gruebele, “Protein folding: The free energy surface,” *Curr. Opin. Struct. Biol.* **12**, 161–168 (2002).
3. J. Mortier, C. Rakers, M. Bermudez, M. S. Murgueitio, S. Riniker, and G. Wolber, “The impact of molecular dynamics on drug design: Applications for the characterization of ligand-macromolecule complexes,” *Drug Discov. Today* **20**, 686–702 (2015).

- <sup>4</sup>A. Cumberworth, J. M. Bui, and J. Gsponer, "Free energies of solvation in the context of protein folding: Implications for implicit and explicit solvent models," *J. Comput. Chem.* **37**, 629–640 (2016).
- <sup>5</sup>J. Zhang, H. Zhang, T. Wu, Q. Wang, and D. van der Spoel, "Comparison of implicit and explicit solvent models for the calculation of solvation free energy in organic solvents," *J. Chem. Theory Comput.* **13**, 1034–1043 (2017).
- <sup>6</sup>B. Roux and T. Simonson, "Implicit solvent models," *Biophys. Chem.* **78**, 1–20 (1999).
- <sup>7</sup>C. J. Cramer and D. G. Truhlar, "Implicit solvation models: Equilibria, structure, spectra, and dynamics," *Chem. Rev.* **99**, 2161–2200 (1999).
- <sup>8</sup>A. V. Onufriev and D. A. Case, "Generalized Born implicit solvent models for biomolecules," *Annu. Rev. Biophys.* **48**, 275–296 (2019).
- <sup>9</sup>H. Nguyen, D. R. Roe, and C. Simmerling, "Improved generalized Born solvent model parameters for protein simulations," *J. Chem. Theory Comput.* **9**, 2020–2034 (2013).
- <sup>10</sup>J. Chen and C. L. Brooks III, "Implicit modeling of nonpolar solvation for simulating protein folding and conformational transitions," *Phys. Chem. Chem. Phys.* **10**, 471–481 (2008).
- <sup>11</sup>H. Nguyen, J. Maier, H. Huang, V. Perrone, and C. Simmerling, "Folding simulations for proteins with diverse topologies are accessible in days with a physics-based force field and implicit solvent," *J. Am. Chem. Soc.* **136**, 13959–13962 (2014).
- <sup>12</sup>A. B. Rubenstein, K. Blacklock, H. Nguyen, D. A. Case, and S. D. Khare, "Systematic comparison of Amber and Rosetta energy functions for protein structure evaluation," *J. Chem. Theory Comput.* **14**, 6015–6025 (2018).
- <sup>13</sup>N. A. Baker, "Improving implicit solvent simulations: A Poisson-centric view," *Curr. Opin. Struct. Biol.* **15**, 137–143 (2005).
- <sup>14</sup>W. C. Still, A. Tempczyk, R. C. Hawley, and T. Hendrickson, "Semianalytical treatment of solvation for molecular mechanics and dynamics," *J. Am. Chem. Soc.* **112**, 6127–6129 (1990).
- <sup>15</sup>U. Haberthür and A. Caflisch, "FACTS: Fast analytical continuum treatment of solvation," *J. Comput. Chem.* **29**, 701–715 (2007).
- <sup>16</sup>E. J. M. Lang, E. G. Baker, D. N. Woolfson, and A. J. Mulholland, "Generalized Born implicit solvent models do not reproduce secondary structures of de novo designed Glu/Lys peptides," *J. Chem. Theory Comput.* **18**, 4070–4076 (2022).
- <sup>17</sup>Y. Chen, A. Krämer, N. E. Charron, B. E. Husic, C. Clementi, and F. Noé, "Machine learning implicit solvation for molecular dynamics," *J. Chem. Phys.* **155**, 084101 (2021).
- <sup>18</sup>G. D. Hawkins, C. J. Cramer, and D. G. Truhlar, "Pairwise solute descreening of solute charges from a dielectric medium," *Chem. Phys. Lett.* **246**, 122–129 (1995).
- <sup>19</sup>A. Onufriev, D. Bashford, and D. A. Case, "Exploring protein native states and large-scale conformational changes with a modified generalized Born model," *Proteins: Struct., Funct., Bioinf.* **55**, 383–394 (2004).
- <sup>20</sup>J. Mongan, C. Simmerling, J. A. McCammon, D. A. Case, and A. Onufriev, "Generalized Born model with a simple, robust molecular volume correction," *J. Chem. Theory Comput.* **3**, 156–169 (2007).
- <sup>21</sup>D. Horvath, G. Marcou, and A. Varnek, "Big data' fast chemoinformatics model to predict generalized Born radius and solvent accessibility as a function of geometry," *J. Chem. Inf. Model.* **60**, 2951–2965 (2020).
- <sup>22</sup>S. S. M. Mahmoud, G. Esposito, G. Serra, and F. Fogolari, "Generalized Born radii computation using linear models and neural networks," *Bioinformatics* **36**, 1757–1764 (2020).
- <sup>23</sup>M. Schaefer, C. Bartels, and M. Karplus, "Solution conformations and thermodynamics of structured peptides: Molecular dynamics simulation with an implicit solvation model," *J. Mol. Biol.* **284**, 835–848 (1998).
- <sup>24</sup>P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," *PLoS Comput. Biol.* **13**, e1005659 (2017).
- <sup>25</sup>R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Big data meets quantum chemistry approximations: The  $\Delta$ -machine learning approach," *J. Chem. Theory Comput.* **11**, 2087–2096 (2015).
- <sup>26</sup>L. Bösel, M. Thürlemann, and S. Riniker, "Machine learning in QM/MM molecular dynamics simulations of condensed-phase systems," *J. Chem. Theory Comput.* **17**, 2641–2658 (2021).
- <sup>27</sup>Y. Khalak, B. Baumeier, and M. Karttunen, "Improved general-purpose five-point model for water: TIP5P/2018," *J. Chem. Phys.* **149**, 224507 (2018).
- <sup>28</sup>H. Meuzelaar, J. Vreede, and S. Woutersen, "Influence of Glu/Arg, Asp/Arg, and Glu/Lys salt bridges on  $\alpha$ -helical stability and folding kinetics," *Biophys. J.* **110**, 2328–2341 (2016).
- <sup>29</sup>W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, "Comparison of simple potential functions for simulating liquid water," *J. Chem. Phys.* **79**, 926–935 (1983).
- <sup>30</sup>D. A. Case, K. Belfon, I. Y. Ben-Shalom, S. R. Brozell, D. S. Cerutti, T. E. Cheatham III, V. W. D. Cruzeiro, T. A. Darden, R. E. Duke, G. Giambasu, M. K. Gilson, H. Gohlke, A. W. Goetz, R. Harris, S. Izadi, S. A. Izmailov, K. Kasavajhala, A. Kovalenko, R. Krasny, T. Kurtzman, T. S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, V. Man, K. M. Merz, Y. Miao, O. Mikhailovskii, G. Monard, H. Nguyen, A. Onufriev, F. Pan, S. Pantano, R. Qi, D. R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C. L. Simmerling, N. R. Skrynnikov, J. Smith, J. Swails, R. C. Walker, J. Wang, L. Wilson, R. M. Wolf, X. Wu, Y. Xiong, Y. Xue, D. M. York, and P. A. Kollman, *AMBER 2021* (University of California, San Francisco, 2021).
- <sup>31</sup>K. Lindorff-Larsen, S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw, "Improved side-chain torsion potentials for the Amber ff99SB protein force field," *Proteins: Struct., Funct., Bioinf.* **78**, 1950–1958 (2010).
- <sup>32</sup>D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Program.* **45**, 503–528 (1989).
- <sup>33</sup>S. Miyamoto and P. A. Kollman, "SETTLE: An analytical version of the SHAKE and RATTLE algorithm for rigid water models," *J. Comput. Chem.* **13**, 952–962 (1992).
- <sup>34</sup>P. Eastman and V. S. Pande, "Constant constraint matrix approximation: A robust, parallelizable constraint method for molecular simulations," *J. Chem. Theory Comput.* **6**, 434–437 (2010).
- <sup>35</sup>Z. Zhang, X. Liu, K. Yan, M. E. Tuckerman, and J. Liu, "Unified efficient thermostat scheme for the canonical ensemble with holonomic or isokinetic constraints via molecular dynamics," *J. Phys. Chem. A* **123**, 6056–6079 (2019).
- <sup>36</sup>T. Darden, D. York, and L. Pedersen, "Particle mesh Ewald: An  $N \cdot \log(N)$  method for Ewald sums in large systems," *J. Chem. Phys.* **98**, 10089–10092 (1993).
- <sup>37</sup>S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.* **107**, 3–11 (2018).
- <sup>38</sup>J. Gasteiger, C. Yeshwanth, and S. Günnemann, "Directional message passing on molecular graphs via synthetic coordinates," in *35th Conference on Neural Information Processing Systems* (NeurIPS, 2021), Vol. 34, pp. 15421–15433.
- <sup>39</sup>D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations* (ICLR, San Diego, 2015).
- <sup>40</sup>R. T. McGibbon, K. A. Beauchamp, M. P. Harrigan, C. Klein, J. M. Swails, C. X. Hernández, C. R. Schwantes, L.-P. Wang, T. J. Lane, and V. S. Pande, "MDTraj: A modern open library for the analysis of molecular dynamics trajectories," *Biophys. J.* **109**, 1528–1532 (2015).
- <sup>41</sup>D. R. Herschbach, H. S. Johnston, and D. Rapp, "Molecular partition functions in terms of local properties," *J. Chem. Phys.* **31**, 1652–1661 (1959).
- <sup>42</sup>X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, "Forces are not enough: Benchmark and critical evaluation for machine learning fields with molecular simulations," *arXiv:2210.07237* (2022).
- <sup>43</sup>S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, and J. T. Margraf, "How robust are modern graph neural network potentials in long and hot molecular dynamics simulations?," *Mach. Learn.: Sci. Technol.* **3**, 045010 (2022).
- <sup>44</sup>S. Doerr, M. Majewski, A. Pérez, A. Krämer, C. Clementi, F. Noe, T. Giorgino, and G. De Fabritiis, "TorchMD: A deep learning framework for molecular simulations," *J. Chem. Theory Comput.* **17**, 2355–2363 (2021).
- <sup>45</sup>R. Galvelis, A. Varela-Rial, S. Doerr, R. Fino, P. Eastman, T. E. Markland, J. D. Chodera, and G. De Fabritiis, "NNP/MM: Fast molecular dynamics simulations with machine learning potentials and molecular mechanics," *arXiv:2201.08110* (2022).