MBX: A Many-Body Energy and Force Calculator for Data-Driven Many-Body Simulations

Marc Riera,[1, a)] Christopher Knight,[2, b)] Ethan F. Bull-Vulpe,[1] Xuanyu Zhu,[1] Henry Agnew,[1] Daniel G. A. Smith,[3] Andrew C. Simmonett,[4] and Francesco Paesani[1, 5, 6, 7, c)]

[1)]*Department of Chemistry and Biochemistry, University of California San Diego, La Jolla, California 92093, USA*

[2)]*Argonne National Laboratory, Computational Science Division, Lemont, IL 60439, United States*

[3)]*Molecular Sciences Software Institute, Blacksburg, Virginia 24060, USA*

[4)]*Laboratory of Computational Biology, National Heart, Lung and Blood Institute, National Institutes of Health, Bethesda, Maryland 20892, USA*

[5)]*Materials Science and Engineering, University of California San Diego, La Jolla, California 92093, USA*

[6)]*Halicioğlu Data Science Institute, University of California San Diego, La Jolla, California 92093, United States*

[7)]*San Diego Supercomputer Center, University of California San Diego, La Jolla, California 92093, USA*

[a)]Electronic mail: mrierari@ucsd.edu

[b)]Electronic mail: knightc@anl.gov

[c)]Electronic mail: fpaesani@ucsd.edu

Abstract

MBX is a C++ library that implements many-body potential energy functions (PEFs) within the "many-body energy" (MB-nrg) formalism. MB-nrg PEFs integrate an underlying polarizable model with explicit machine-learned representations of many-body interactions to achieve chemical accuracy from the gas to the condensed phases. MBX can be employed either as a stand-alone package or as an energy/force engine that can be integrated with generic software for molecular dynamics and Monte Carlo simulations. MBX is parallelized internally using OpenMP, and can utilize MPI when available in interfaced molecular simulation software. MBX enables classical and quantum molecular simulations with MB-nrg PEFs, as well as hybrid simulations that combine conventional force fields and MB-nrg PEFs, for diverse systems ranging from small gas-phase clusters to aqueous solutions and molecular fluids to biomolecular systems and metal-organic frameworks.

I. INTRODUCTION

Molecular dynamics (MD) and Monte Carlo (MC) simulations[1,2] have been widely used for understanding and characterizing structural, thermodynamic, and dynamical properties of molecular systems, from small gas-phase clusters to extended materials and biomolecular systems.[3–8] The potential energy function (PEF) used to represent the multidimensional potential energy surface associated with the molecular system being studied directly determines the level of realism as well as the predictive power of any MD and MC simulation.

In the early days of molecular simulations, due to limited computational resources, the only viable options for PEFs were empirically parameterized force fields (FFs) that use relatively simple expressions to describe intramolecular distortions and pairwise-additive functions to describe intermolecular interactions.[9,10] Although more advanced (nonpolarizable and polarizable) FFs developed over the past five decades[11–15] remain the most commonly used PEFs in MD and MC simulations, machine-learning (ML) models trained on electronic structure data have become increasingly popular, promising higher accuracy than conventional FFs.[16–19] Some examples of ML PEFs include neural network potentials (NNPs),[20–29] equivariant graph neural network potentials,[30] Gaussian approximation potentials (GAPs),[31] moment tensor potentials (MTPs),[32] and spectral neighbor analysis potentials (SNAPs),[33] as well as PEFs based on the atomic cluster expansion,[34] kernel ridge regression methods,[35] gradient-domain machine learning (GDML),[36]

and support vector machines (SVM).[37] Permutationally invariant polynomials (PIPs) have also been used, either as standalone fitting functions[38–62] or in combination with neural networks (PIP-NNs).[63–66] Many ML PEFs are, however, limited in their transferability - those designed to mimic gas phase properties perform well under those conditions but may not be as accurate when applied to condensed-phase systems,[67–69] and models that are trained to reproduce condensed-phase properties may not perform as well in the gas phase or at interfaces.[70,71]

Ten years ago, Babin, Medders, and Paesani introduced MB-pol, a data-driven many-body PEF for water rigorously derived from "first principles".[72–74] MB-pol combines physics-based many-body models with data-driven machine-learned representations of individual many-body interactions that are expressed in terms of multidimensional PIPs. These machine-learned PIPs were shown to account for limitations in classical representations of molecular interactions that arise when overlapping electron densities lead to quantum-mechanical effects that do not have a classical counterpart, such as exchange-repulsion, charge transfer, and charge penetration.[75–77] The PIPs of MB-pol were trained on large datasets of many-body energies calculated at the coupled cluster level of theory, including single, double, and perturbative triple excitations, i.e., CCSD(T), the current "gold standard" for chemical accuracy.[78] By construction, MB-pol is fully transferable across all phases,[79,80] accurately reproducing the properties of small gas-phase clusters,[81–92] liquid water,[93–99] the air/water interface,[100–104] and ice.[105–110] Remarkably, MB-pol was shown to be the first and, currently, only water PEF able to correctly predict the phase diagram of water.[111] More recently, an updated version of MB-pol, MB-pol(2023), which was trained on larger training sets of many-body interactions, was shown to achieve even higher accuracy for simulations of water in both gas and liquid phases.[112]

Building on the accuracy and predictive power of MB-pol, many-body PEFs for various molecular systems were developed, including halide[113–119] and alkali-metal[120–124] ions in water, molecular fluids,[125–128] small molecules in water,[129,130] and generic covalently-bonded molecules in the gas phase.[131] These many-body PEFs were developed within the many-body energy (MB-nrg) theoretical/computational framework,[113,120] which effectively generalizes the MB-pol framework to arbitrary molecules. Briefly, the MB-nrg PEF of a system is built upon a baseline physics-based model describing permanent electrostatics, London dispersion forces, and many-body polarization, which is supplemented by explicit machine-learned n-body PIPs. As in MB-pol, the MB-nrg PIPs effectively represent quantum-mechanical many-body interactions arising from the overlap of the electron densities of individual monomers.[113,120]

Here, we introduce MBX (Many-Body eXpansion),[132] a modular C++ library that can either be used as a standalone software for calculating MB-nrg energies and forces for the molecular system of interest or interfaced with external MD and MC engines to perform classical and quantum simulations of the molecular system of interest across different thermodynamic states and phases, in both periodic and non-periodic conditions, using the corresponding MB-nrg PEFs. Importantly, MBX is interfaced with MB-Fit,[133] a Python software infrastructure that provides an integrated suite of codes for the automated development of MB-nrg PEFs for generic molecules, from training set generation to PEF fitting and implementation.[134]

II. THEORY: MB-NRG POTENTIAL ENERGY FUNCTIONS

The energy of a system containing N (atomic or molecular) monomers (hereafter referred to as 1-mers) can be rigorously expressed as a sum of n-body energy contributions ($1 \leq n \leq N$) according to the many-body expansion (MBE) of the energy:[135]

$$E_N(1,\ldots,N) = \sum_{i=1}^{N} \varepsilon^{1\mathrm{B}}(i) + \sum_{i<j}^{N} \varepsilon^{2\mathrm{B}}(i,j) + \sum_{i<j<k}^{N} \varepsilon^{3\mathrm{B}}(i,j,k) + \ldots + \varepsilon^{\mathrm{NB}}(1,\ldots,N) \qquad (1)$$

where each 1-body energy, $\varepsilon^{1\mathrm{B}}(i)$, is the energy of the isolated ith 1-mer, $E_1(i)$. For $n \geq 2$, the n-body energies, $\varepsilon^{n\mathrm{B}}$ are defined recursively according to the following expression:

$$\varepsilon^{n\mathrm{B}}(1,\ldots,n) = E_n(1,\ldots,n) - \sum_{i=1}^{n} \varepsilon^{1\mathrm{B}}(i) - \sum_{i<j}^{n} \varepsilon^{2\mathrm{B}}(i,j) - \sum_{i<j<k}^{n} \varepsilon^{3\mathrm{B}}(i,j,k) - \ldots - \sum_{i<j<k<\ldots}^{n} \varepsilon^{(n-1)\mathrm{B}}(i,j,k,\ldots) \qquad (2)$$

It should be noted that within the MB-nrg theoretical/computational framework the reference zero for the energy scale (where $E_N = 0$) corresponds to the molecular configuration in which all N 1-mers are separated by infinite distances and each 1-mer is in its minimum-energy geometry. As a consequence, $\varepsilon^{1\mathrm{B}}(i)$ corresponds to the distortion energy of the ith 1-mer relative to its minimum-energy geometry. Since the MBE converges quickly for molecular systems with localized electron densities, i.e., molecular systems with large electronic band gaps,[136–139] the MBE provides a rigorous and efficient theoretical/computational framework for the development of many-body PEFs where each n-body term of Eq. 1 is fitted to reproduce the corresponding n-body reference energies calculated from "first principles".

As in MB-pol,[72–74] the MB-nrg PEFs integrate physics-based many-body terms, representing contributions to molecular interactions that can be accurately represented by classical expressions

(e.g., permanent electrostatics and polarization), with explicit machine-learned representations of individual n-body terms in the MBE, which effectively recover quantum-mechanical interactions arising from the overlap of 1-mer's electron densities (e.g., exchange-repulsion, charge transfer, and charge penetration) that cannot be represented by classical expressions.[140] Specifically, the MB-nrg theoretical/computational framework approximates the MBE defined in Eq. 1 as:

$$E_N = V^{1\mathrm{B}} + V^{2\mathrm{B}} + V^{3\mathrm{B}} + \ldots + V^{n\mathrm{B}} + V_{\mathrm{elec}} \tag{3}$$

where $n \leq N$ and N is the total number of 1-mers in the system.

Each of the $V^{n\mathrm{B}}$ terms of an MB-nrg PEF includes an n-body machine-learned term ($V^{n\mathrm{B}}_{\mathrm{ML}}$) for each n-mer. Each $V^{n\mathrm{B}}_{\mathrm{ML}}$ is expressed as a product of a switching function and a PIP (i.e., $V^{n\mathrm{B}}_{\mathrm{ML}} = s^{n\mathrm{B}} V^{n\mathrm{B}}_{\mathrm{PIP}}$). The switching function ($s^{n\mathrm{B}}$) ensures that the contribution from the associated $V^{n\mathrm{B}}_{\mathrm{ML}}$ term goes to zero as any subset of the 1-mers in an n-mer is separated from the rest.

Following the original MB-pol PEF,[72,73] a given n-body PIP takes the following form:

$$V^{n\mathrm{B}}_{\mathrm{PIP}}(\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_n | \nu(\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_n)) = \sum_{l=1}^{L} c_l \cdot \eta_l(\xi_1, \xi_2, \ldots, \xi_\lambda) \tag{4}$$

Here, $\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_n$ are n 1-mers which compose an n-mer of type $\nu(\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_n)$, L is the number of linear parameters, c_l are the linear parameters, η_l are the symmetrized monomials built from the variables, $\xi_{1-\lambda}$, each of which is an exponential of an interatomic distance with one of the following forms:

$$\xi^{\exp}(R_{mn}) = e^{-k_{\tau(mn)} R_{mn}} \tag{5a}$$

$$\xi^{\exp 0}(R_{mn}) = e^{-k_{\tau(mn)} (R_{mn} - d_{0,\tau(mn)})} \tag{5b}$$

$$\xi^{\mathrm{coul}}(R_{mn}) = e^{-k_{\tau(mn)} R_{mn}} / R_{mn} \tag{5c}$$

$$\xi^{\mathrm{coul0}}(R_{mn}) = e^{-k_{\tau(mn)} (R_{mn} - d_{0,\tau(mn)})} / R_{mn} \tag{5d}$$

where m, n are the indices for the physical atoms or fictitious sites defined by the n-mer's geometry, and R_{mn} is the distance between two atoms/sites. $\tau(mn)$ maps the pair of atoms/sites into distinct classes, such that all atom/site pairs within the same class share the same nonlinear fitting parameters $k_{\tau(mn)}$ and $d_{0,\tau(mn)}$. There is one unique set of monomials (η_l), linear fitting parameters (c_l), and non-linear fitting parameters [$k_{\tau(mn)}$, $d_{0,\tau(mn)}$] for each unique n-mer type [$\nu(\mathrm{M}_1, \mathrm{M}_2, \ldots, \mathrm{M}_n)$].

In Eq. 3, $V^{1\mathrm{B}}$ is the total 1-body energy given by

$$V^{1\mathrm{B}} = \sum_{i=1}^{N} V^{1\mathrm{B}}_{\mathrm{ML}}(\mathrm{M}_i | \nu(\mathrm{M}_i)) \;\; (+V^{1\mathrm{B}}_{\mathrm{disp}}) \tag{6}$$

Because a switching function is not used for the 1-body term, $V_{\mathrm{ML}}^{\mathrm{1B}}\left(\mathrm{M}_i|\nu(\mathrm{M}_i)\right)$ is simply a machine-learned PIP representing the 1-body energy of the ith 1-mer with functional form as in Eq. 4:

$$V_{\mathrm{ML}}^{\mathrm{1B}}\left(\mathrm{M}_i|\nu(\mathrm{M}_i)\right) = V_{\mathrm{PIP}}^{\mathrm{1B}}\left(\mathrm{M}_i|\nu(\mathrm{M}_i)\right) \tag{7}$$

$V_{\mathrm{disp}}^{\mathrm{1B}}$ represents the 1-body dispersion energy, as a sum of interatomic pairwise contributions:

$$V_{\mathrm{disp}}^{\mathrm{1B}} = \sum_{i=1}^{N}\left[\sum_{\substack{k,l\in\mathrm{M}_i\\ l\neq k}} -\Delta_{kl} f(\mathrm{b}_{kl}R_{kl})\frac{C_{6,kl}}{R_{kl}^6}\right] \tag{8}$$

where R_{kl} is the distance between atoms k and l located on 1-mer M_i, $\mathrm{C}_{6,kl}$ is the corresponding dispersion coefficient, and $\Delta_{kl} = 0$ if the atom pair is excluded or 1 otherwise. $f(\mathrm{b}_{kl}R_{kl})$ is the Tang-Toennies damping function,[141]

$$f(\mathrm{b}_{kl},R_{kl}) = 1 - \exp(-\mathrm{b}_{kl}R_{kl})\sum_{n=0}^{6}\frac{(\mathrm{b}_{kl}R_{kl})^n}{n!} \tag{9}$$

where b_{kl} is a fitting parameter. By convention, all atom pairs that participate in a bond, angle, or dihendral angle are excluded ($\Delta_{kl} = 0$). Thus, for most 1-mers, all atom pairs are excluded and $V_{\mathrm{disp}}^{\mathrm{1B}} = 0$. However, for large 1-mers, this may not be the case.

The explicit 2-body term of an MB-nrg PEF, V^{2B} in Eq. 3, is expressed as

$$V^{\mathrm{2B}} = \sum_{\substack{i=1\\ j>i}}^{N} V_{\mathrm{ML}}^{\mathrm{2B}}\left(\mathrm{M}_i,\mathrm{M}_j|\nu(\mathrm{M}_i,\mathrm{M}_j)\right) + V_{\mathrm{disp}}^{\mathrm{2B}} \tag{10}$$

Here, $V_{\mathrm{ML}}^{\mathrm{2B}}\left(\mathrm{M}_i,\mathrm{M}_j|\nu(\mathrm{M}_i,\mathrm{M}_j)\right)$ is a 2-body machine-learned term representing the 2-body energy of the 2-mer composed by the ith and jth 1-mers, constructed as a product of a switching function and a PIP with functional form as in Eq. 4:

$$V_{\mathrm{ML}}^{\mathrm{2B}}\left(\mathrm{M}_i,\mathrm{M}_j|\nu(i,j)\right) = s^{\mathrm{2B}}\left(\mathrm{M}_i,\mathrm{M}_j|\nu(\mathrm{M}_i,\mathrm{M}_j)\right) V_{\mathrm{PIP}}^{\mathrm{2B}}\left(\mathrm{M}_i,\mathrm{M}_j|\nu(\mathrm{M}_i,\mathrm{M}_j)\right) \tag{11}$$

$V_{\mathrm{disp}}^{\mathrm{2B}}$ in Eq. 10 is the total 2-body dispersion energy calculated as a sum of pairwise additive contributions associated with each pair of atoms located on the two 1-mers in a 2-mer:[140]

$$V_{\mathrm{disp}}^{\mathrm{2B}} = \sum_{\substack{i=1\\ j>i}}^{N}\left[\sum_{k\in\mathrm{M}_i}\sum_{l\in\mathrm{M}_j} -f(\mathrm{b}_{kl}R_{kl})\frac{C_{6,kl}}{R_{kl}^6}\right] \tag{12}$$

where R_{kl} is the distance between atoms k and l located on 1-mers M_i and M_j, respectively, $\mathrm{C}_{6,kl}$ is the corresponding dispersion coefficient, and $f(\mathrm{b}_{kl}R_{kl})$ is the Tang-Toennies damping function

(Eq. 9). In both Eq. 8 and Eq. 12, the dispersion coefficients are calculated using the Exchange Dipole Moment (XDM) model.[142–144]

All other explicit many-body terms ($V^{n\mathrm{B}}$) in Eq. 3 take the following form:

$$V^{n\mathrm{B}} = \sum_{\substack{i=1 \\ j>i \\ \cdots \\ l>k}}^{N} V_{\mathrm{ML}}^{n\mathrm{B}}\left(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l|\nu(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l)\right) \tag{13}$$

where each $V_{\mathrm{ML}}^{n\mathrm{B}}\left(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l|\nu(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l)\right)$ is built as the product of a switching function and a PIP with functional form as in Eq. 4:

$$\begin{aligned} &V_{\mathrm{ML}}^{n\mathrm{B}}\left(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l|\nu(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l)\right) = \\ &s^{n\mathrm{B}}\left(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l|\nu(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l)\right) V_{\mathrm{PIP}}^{n\mathrm{B}}\left(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l|\nu(\mathrm{M}_i,\mathrm{M}_j,\ldots,\mathrm{M}_l)\right) \end{aligned} \tag{14}$$

Explicit n-body terms may be retained up to an arbitrary n-body level. Generally, it is sufficient to truncate these terms at the $n = 3$ or $n = 4$ level, depending on the system being studied. Specific details about the switching functions ($s^{2\mathrm{B}}$, $s^{3\mathrm{B}}$, and $s^{4\mathrm{B}}$), including functional forms used by the MB-nrg PEFs available in MBX, are discussed in the Supplementary Information.

Finally, the electrostatics term, V_{elec}, in Eq. 3 is based on a modified version of the Thole model[145] introduced in Ref. 146 and further refined for the MB-pol PEF.[72,73] V_{elec} represents permanent electrostatics by a sum of Coulomb interactions between smeared partial charges located on each 1-mer as well as induced electrostatics (up to dipoles) by an implicit many-body polarization term. Within the MB-nrg theoretical/computational framework, the partial charges, which can have fixed or geometry-dependent values, are obtained by fitting the multipole moments calculated from "first principles" for each isolated 1-mer and can be placed on both physical atoms and fictitious sites.

In MBX, V_{elec} is represented by four terms describing charge-charge interactions (V_{qq}), charge-dipole interactions ($V_{\mathrm{q}\mu}$), dipole-dipole interactions ($V_{\mu\mu}$), and the polarization energy (V_{pol}), re-

spectively. Each of these terms is defined as follows:

$$V_{\mathrm{qq}} = \sum_{i}^{N} \sum_{j>i} q_i \widehat{T}_{ij} q_j \tag{15a}$$

$$V_{\mathrm{q}\mu} = \sum_{i}^{N} \sum_{j>i} \left(\mu_i^{\alpha} \widehat{T}_{ij}^{\alpha} q_j - q_i \widehat{T}_{ij}^{\alpha} \mu_j^{\alpha} \right) \tag{15b}$$

$$V_{\mu\mu} = -\sum_{i}^{N} \sum_{j>i} \mu_i^{\alpha} \widehat{T}_{ij}^{\alpha\beta} \mu_j^{\beta} \tag{15c}$$

$$V_{\mathrm{pol}} = \frac{1}{2} \sum_{i=1}^{N} \boldsymbol{\mu}_i \widehat{\alpha}_i^{-1} \boldsymbol{\mu}_i \tag{15d}$$

where the Einstein notation is used for repeated Greek letters (e.g., μ_i^{α} is a condensed form of $\sum_{\alpha=x,y,z} \mu_i^{\alpha}$) In Eqs. 15a-d, N is the total number of electrostatic sites in the system, q_i is the charge of site i, $\boldsymbol{\mu}_i$ is the dipole moment of site i, $\widehat{\alpha}_i$ is the polarizability of site i ($\widehat{\alpha}_i$ becomes a scalar if it is isotropic), and $\widehat{T}_{ij}$, $\widehat{T}_{ij}^{\alpha}$, and $\widehat{T}_{ij}^{\alpha\beta}$ are the electrostatic tensors defined as follows:

$$\widehat{T}_{ij} = S_0(R_{ij}) \frac{1}{R_{ij}} \tag{16a}$$

$$\widehat{T}_{ij}^{\alpha} = \nabla_{\alpha} \widehat{T}_{ij} = -S_1(R_{ij}) \frac{R_{ij}^{\alpha}}{R_{ij}^3} \tag{16b}$$

$$\widehat{T}_{ij}^{\alpha\beta} = \nabla_{\alpha} \widehat{T}_{ij}^{\beta} = S_2(R_{ij}) \frac{3R_{ij}^{\alpha} R_{ij}^{\beta}}{R_{ij}^5} - S_1(R_{ij}) \frac{\delta_{\alpha\beta}}{R_{ij}^3} \tag{16c}$$

$$\widehat{T}_{ij}^{\alpha\beta\gamma} = \nabla_{\alpha} \widehat{T}_{ij}^{\beta\gamma} = -S_3(R_{ij}) \frac{15}{R_{ij}^7} R_{ij}^{\alpha} R_{ij}^{\beta} r_{\gamma} + S_2(R_{ij}) \frac{3}{R_{ij}^5} \left(R_{ij}^{\alpha} \delta_{\beta\gamma} + R_{ij}^{\beta} \delta_{\alpha\gamma} + R_{ij}^{\gamma} \delta_{\alpha\beta} \right) \tag{16d}$$

Here, α,β,γ define any of the Cartesian directions (x, y, or z), R_{ij} is the distance between atoms i and j, and δ is the Kronecker delta. The functions $S_i(r)$ are the screening functions designed to smear the charges over space, which can be recursively derived from Eq. 18a as

$$S_k(r) = S_{k-1} - \frac{r}{2k-1} \frac{\partial}{\partial r} S_{k-1}(r) \tag{17}$$

As in MB-pol,[72,73] the screening functions for the MB-nrg PEFs are given by

$$S_0(r) = 1 - e^{-a\left(\frac{r}{A}\right)^4} + \frac{a^{1/4} r}{A} \Gamma\left(\frac{3}{4}, a\left(\frac{r}{A}\right)^4\right) \tag{18a}$$

$$S_1(r) = 1 - e^{-a\left(\frac{r}{A}\right)^4} \tag{18b}$$

$$S_2(r) = S_1(r) - \frac{4a}{3} \left(\frac{r}{A}\right)^4 e^{-a\left(\frac{r}{A}\right)^4} \tag{18c}$$

$$S_3(r) = S_2(r) - \frac{4a}{15} \left(\frac{r}{A}\right)^4 e^{-a\left(\frac{r}{A}\right)^4} \left(4a\left(\frac{r}{A}\right)^4 - 1\right) \tag{18d}$$

Here, a is the Thole damping, which can be different for charge-charge, charge-dipole, and dipole-dipole interactions, $A = \left(\alpha_i \alpha_j\right)^{1/6}$, with i and j being the two sites involved, $r = R_{ij}$, and α is the polarizability factor that is usually set to be the same as the polarizability. The interested reader is referred to Ref. 147 for specific details about the derivation of Eqs. 14-17.

III. SOFTWARE STRUCTURE

The C++ source code of MBX is organized into four modules, each of which handles specific functions: `building block` is responsible for maintaining the state of the system; `potential` evaluates the various components of the MB-nrg PEFs; `I/O` manages inputs, outputs, and interfaces with MD drivers; and `utilities` contains functions to execute miscellaneous support tasks. The `potential` module is further divided into sub-modules to calculate each of the energy contributions described in Eq. 3: n-body PIPs, 2-body dispersion, permanent electrostatics, and many-body polarization. The general workflow for an energy calculation step performed by MBX is shown in Fig. III.

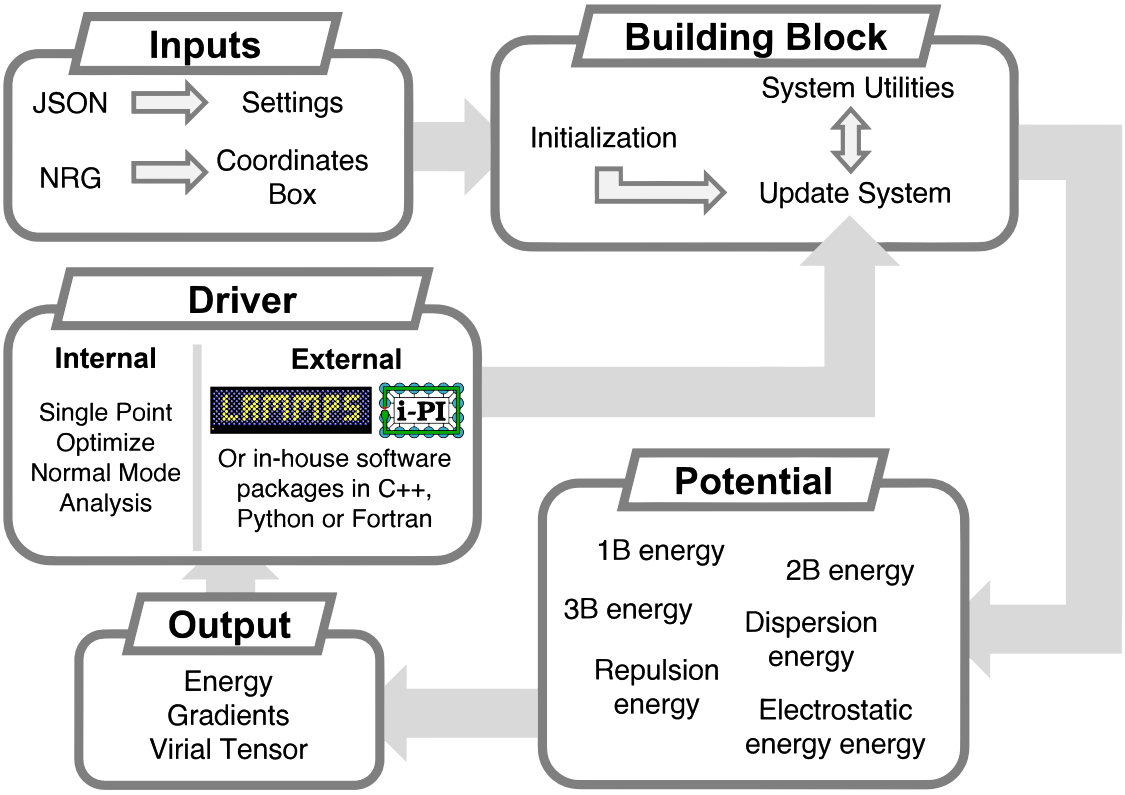

FIG. 1. General workflow for an energy and force calculation step in MBX.

A. Input

Since all PEF parameters are directly compiled into MBX, the user only needs to provide minimal information that is passed to MBX through two files: the NRG file, which contains information about all 1-mers in the system and their initial coordinates, and the JSON file, which specifies details about the calculation to be performed, such as enabling or disabling n-body terms for certain n-mers, and assorted settings, such as the algorithm for the calculation of many-body polarization and convergence threshold for the induced dipole moments. More information about the format and contents of the NRG and JSON files are discussed in the Supplementary Material.

B. Building block

The `building block` module contains the `System` class, which stores all the information about the 1-mers in the system. MBX provides a function that reads the NRG file, and creates and initializes a `System` instance. The initialized `System` can then be configured using the JSON parameters that control the energy calculation. Initializing a `System` object requires multiple memory allocation calls and is therefore not instantaneous, but it is performed only once per system of interest.

While MD and MC software implementations for force fields typically treat atoms as the smallest unit, MBX considers 1-mers, each consisting of a few atoms, as the smallest unit of the system since the n-body PIPs, which form the backbone of MB-nrg PEFs, are evaluated on these 1-mers. For this reason, the `System` class stores data on a per-1-mer basis.

For each 1-mer type defined in a MB-nrg PEF, the parameters defining all relevant atomic quantities (e.g., charges, polarizabilities, and dispersion coefficients) are automatically compiled into MBX. Because the n-body PIPs of a given MB-nrg PEF are fitted over the underlying representation of electrostatics and dispersion, the parameters entering the expressions for $V_{\text{disp}}^{\text{2B}}$ (Eq. 10) and V_{elec} (Eqs. 14a-d) are intertwined with each MB-nrg PEF. As a consequence, if the user wishes to adopt a different set of electrostatic or dispersion parameters, all n-body PIPs will need to be refitted using MB-Fit.[133]

The `System` class oversees the calculations of each contribution to the total energy by delegating to the appropriate functions within the `potential` module (see below). Each function returns the energy and associated gradients of a particular energy contribution with respect to the coordi-

nates of the atoms. Once the `System` object is initialized, it is not possible in the current version of MBX to add new 1-mers or change the type of existing ones without rebuilding the `System` instance. The atomic coordinates can be updated at any time as long as they are in the same order as the initial set of coordinates. Similarly, any parameters specifying the type of calculation to be performed (e.g., algorithm for many-body polarization, convergence threshold for the induced dipoles, and box size and shape for calculations in periodic boundary conditions) can be changed at any time.

MBX initializes a `System` object through the following steps:

1. Create a new `System` object with default parameters corresponding to those used for a gas-phase calculation.

2. Add 1-mers to the `System` using the `AddMonomer` member function. The coordinates, atom labels, and 1-mer type for each 1-mer are stored in the `System` object.

3. After all 1-mers have been added, initialize the `System`. This involves storing the properties for each 1-mer and reordering the 1-mers for optimization of parallelization. The reordering process groups 1-mers of the same type together and orders the types by increasing number of 1-mers. For example, the input for a system of 250 CO_2 molecules (i.e., 1-mers of type CO_2) and 300 H_2O molecules (i.e., 1-mers of type H_2O) can be provided in any order, but MBX will reorder it such that the CO_2 molecules come before the H_2O molecules.

4. Set the physical properties of the atoms, such as charges, polarizabilities, and dispersion coefficients, using helper functions.

5. Parse the JSON file containing information about box size and shape for calculations in periodic boundary conditions, cutoffs, and type of MB-nrg PEFs, as well as other options that control and determine the type of calculation and energy calls to be performed. If a JSON file is not found or not present, the defaults are used.

C. Machine-learned 1-body term: $V_{\mathrm{ML}}^{\mathrm{1B}}$

Since different MD and MC engines have different conventions regarding storage of atom coordinates, MBX first translates the atoms in the 1-mer to obey the minimum-image convention

before evaluating the 1-body PIPs. This is only necessary when performing a calculation in periodic boundary conditions. MBX selects the first atom of each 1-mer as the reference atom to identify the minimum images of the other atoms in the 1-mer. This reference atom is then placed in the principal box and the closest images of other atoms in the 1-mer are selected through an algorithm that operates in fractional coordinates.

D. Machine-learned n-body terms ($n > 1$): $V_{\mathrm{ML}}^{n\mathrm{B}}$

MBX supports n-body PIPs with arbitrary values for n, which can be generated with MB-Fit,[133,134] and currently already provides functions to evaluate 2-body, 3-body, and 4-body PIPs. Adding n-body PIPs with larger values of n is trivial and does not require any significant refactoring of the source code. In order to efficiently evaluate all $V_{\mathrm{ML}}^{n\mathrm{B}}$ terms (Eq. 4), MBX first identifies all n-mers for which it is possible that the associated n-body switching function ($s^{n\mathrm{B}}$) is non-zero. An n-mer is accepted and passed to the polynomial evaluation if and only if some 1-mer within the n-mer is within a predefined n-body cutoff ($r_{\mathrm{cutoff}}^{n\mathrm{B}}$) of all other 1-mers in the n-mer. In other words, there must be a "central" 1-mer, and all other 1-mers must be within the n-body cutoff of the "central" 1-mer. This idea is formalized in the following criterion:

> CENTER-NEIGHBOR CRITERION: *Using the first atom of each* 1*-mer to define the position of the* 1*-mer, the center-neighbor criterion for a given n-mer is satisfied if and only if there exists at least one* 1*-mer ("center") such that the distances between the "center"* 1*-mer and all other n-1* 1*-mers ("neighbors") in the n-mer are smaller than the n-body cutoff* $r_{\mathrm{cutoff}}^{n\mathrm{B}}$.

The value for each $r_{\mathrm{cutoff}}^{n\mathrm{B}}$ used by the CENTER-NEIGHBOR CRITERION is specified by the user in the JSON file. As a consequence, MBX only needs to collect n-mers for which the CENTER-NEIGHBOR CRITERION is satisfied and pass this information to the PIP evaluator. The rules for setting appropriate values for $r_{\mathrm{cutoff}}^{n\mathrm{B}}$ are discussed in the Supplementary Information.

MBX uses a K-D Tree to search for n-mers that satisfy the CENTER-NEIGHBOR CRITERION, after which the evaluation of the n-body PIPs with $n > 1$ is effectively the same as for the 1-body PIPs. Using a K-D Tree allows MBX to quickly identify relevant n-mers and avoid the need for a double or triple loop over all 1-mers, which would be extremely slow. The Nanoflann library[148] is used to implement the K-D Tree and perform the radial search. It should be noted that, although not negligible, the CPU time required to create the tree and perform the search is still a small fraction of

the CPU time required to calculate the n-body PIP contributions. The K-D Tree implementation in MBX is as follows: first, a tree is built using the first atom of each 1-mer as a point in the tree. Once the tree containing all the 1-mers is completed, MBX loops over all the 1-mers, which will be the candidate "center" 1-mer in the current loop, and performs a radial search of all other 1-mers that are within $r^{n\mathrm{B}}_{\mathrm{cutoff}}$, which will be the candidate "neighbor" 1-mers. Then, n-mers are constructed from the "center" 1-mer and each combination of n-1 "neighbors". By construction, each of the constructed n-mers necessarily satisfies the CENTER-NEIGHBOR CRITERION. However, it is possible that the same n-mer can be selected several times, with different 1-mers acting as the "center". To avoid double counting, the candidate n-mer is considered valid only if the 1-mer index of the "center" is the smallest among all valid "center" 1-mers.

Although K-D Trees were not originally designed for use in periodic boundary conditions, MBX has implemented a patch that allows for their use in such cases by replicating the box in space. This implies that instead of building a tree for a single copy of the system as done in the gas phase, MBX builds a tree for 27 copies: the original one and the twenty six adjacent boxes. Only images within the main simulation box are eligible "center" 1-mers. Future versions of MBX will implement more advanced solutions to address the potential memory cost of this process when the target number of 1-mers is large. After obtaining the lists of n-mers, MBX sends batches of multiple n-mers of the same type to the PIP functions, which then transform the coordinates into PIP variables and calculate the corresponding PIP values.

E. Physics-based terms

MBX defines two distinct classes that are dedicated to calculating the following non-bonded interactions: dispersion (`Dispersion` class) and permanent and induced electrostatics (`Electrostatics` class). As in conventional force fields and discussed in Section II, MBX excludes these non-bonded interactions for atom pairs that are part of a bond, angle, or dihedral angle. However, MBX does not scale these interactions as common force fields do - for a particular atom pair, they are either entirely enabled or entirely disabled (hence Δ_{kl} in Eq. 8). Generally, all atom pairs within a 1-mer are excluded, but in the event that a 1-mer contains non-excluded pairs both classes calculate the contributions from 1-mer dispersion and electrostatics (i.e., dispersion energy as in Eq. 8 and 1-body contributions to V_{elec} in Eq. 3, respectively) in a first step, ignoring any pair in the excluded pairs list. Then, the intermolecular contributions are calculated in a double-loop over

the 1-mer types. For each pair of 1-mer types, the contributions to the dispersion and electrostatics energies are calculated. Before evaluation, the coordinates and associated properties (e.g., atomic charges and polarizabilities, dispersion coefficients, etc.) are reordered to maximize speedup from vectorization through single-instruction multiple-data (SIMD) operations.

1. Dispersion

As shown in Eq. 12, the dispersion energy of a MB-nrg PEF is calculated in real space as a pairwise-additive potential using pair-defined dispersion coefficients ($C_{6,kl}$) that are calculated using the XDM model.[142–144] If the molecular system of interest is in periodic boundary conditions, the long-range contribution to the dispersion energy is calculated in reciprocal space using the particle mesh Ewald (PME) algorithm as implemented in the helPME library.[149,150] PME uses atom-defined C_6 which are then combined using the usual geometric mean combination rule to obtain pair coefficients (i.e., $C_{6,kl} = \sqrt{C_{6,kk}C_{6,ll}}$). A discontinuity in the energy and its gradients can occur if the $C_{6,kl}$ pair coefficients used to calculate the dispersion energy in real space are abruptly changed to the values used by the PME algorithm at the cutoff distance. To avoid this discontinuity, MBX applies a switching function of the same form as that used for the 2-body PIP switching function (see Supplementary Information), enabling a smooth transition from the $C_{6,kl}$ used in real space to the $C_{6,kl}$ used in the PME calculation.

2. Electrostatics

The electrostatics calculation involves several steps, including the computation of the permanent electric field, the calculation of the long-range electric field using the PME algorithm as implemented in the helPME library,[149,150] and the determination of the induced dipoles using one of three algorithms: iterative, conjugate gradient, or always stable predictor-corrector.[151] The permanent contribution to the electrostatic energy is straightforward to calculate and relatively fast. However, the bottleneck of the electrostatics calculation is to obtain the induced dipoles on each site. While the analytical solution of the induced dipole moments is possible, it is not efficient for large systems,[147] and it has not been implemented in MBX. A detailed description of the possible methods to solve for the induced dipole moments can be found in Ref. 147.

```cpp
// Needed to read the NRG file
#include "io_tools/read_nrg.h"
// Needed to use the system class
#include "bblock/system.h"
#include <vector>
#include <string>
int main() {
    // Declare systems vector
    std::vector<bblock::System> systems;
    // Read systems from NRG file
    std::string input = "input.nrg";
    tools::ReadNrg(&input[0], systems);
    // Set up from json file
    std::string json_file = "input.json";
    systems[0].SetUpFromJson(&json_file[0]);
    // Compute energy
    double e = systems[0].Energy(true);
    // Retrieve gradients
    std::vector<double> grads = systems[0].GetRealGrads();
    return 0;
}
```

FIG. 2. Example of a C++ main function to use the MBX library with a NRG and a JSON file.

F. Output

Once all energy and gradient contributions have been calculated, they are summed and stored in the `System` object, ready to be retrieved by the user or a MD/MC driver. After this step is completed, external modifications to the coordinates of the system such as progression to the next MD/MC step can be performed. The new coordinates are set in the same `System` instance, which can then be used to perform another energy/force calculation.

While energies and forces are the most commonly retrieved information by MD and MC drivers, MBX provides interfaces to retrieve any of the system's properties, including, but not limited to, charges, permanent and induced dipole moments, and the virial tensor.

IV. DRIVERS

MBX has three built-in drivers to perform single point calculations, geometry optimizations, and normal-mode analyses, all written in C++. A simple example on how to use MBX to read an NRG file and set up the system with a JSON file is shown in Fig. 2.

Besides the internal drivers discussed above, the current version of MBX also provides an ef-

ficient interface to popular software packages LAMMPS[152] and i-PI[153] for both classical MD and quantum path-integral molecular dynamics (PIMD) simulations.[2] MBX acts as a client that returns MB-nrg energies and forces, while the actual MD steps are controlled by the MD engine. In the case of i-PI, the communication between MBX and i-PI can be established in two ways: Internet and Unix domain sockets. For LAMMPS, MBX is connected through the combination of specific `FIX` and `PAIR_STYLE` commands in the LAMMPS input. The MBX/LAMMPS and MBX/i-PI interfaces have already been used to study the water vapor/liquid equilibrium,[104] CH_4/H_2O[126,128] and CO_2/H_2O[125,127] mixtures, and ions in solution.[118,119,122] In the current version of MBX, all of the computationally expensive functions are parallelized using OpenMP to maximize use of large many-core compute nodes. This design readily enables other "driver" codes, serial or parallel, to couple with MBX and perform advanced calculations, such as MD and PIMD simulations using LAMMPS or i-PI.

The pure driver-only nature of i-PI makes the interface with MBX very simple. A single driver code that communicates with the i-PI socket is enough to allow both packages to communicate. The driver code receives the coordinates and the simulation cell from i-PI through a socket, sets them into MBX, and performs the energy calculation for those coordinates. Gradients and energies are then retrieved from MBX and sent through the socket to i-PI that performs the time evolution for each time step, updating both atom coordinates and simulation cell, which are then sent back to the driver.

In the case of LAMMPS, MBX is tightly coupled to enable large-scale parallel simulations with minimal overhead. LAMMPS is parallelized using a spatial domain decomposition algorithm whereby the simulation is partitioned into sub-domains and individual MPI ranks are responsible for computing all tasks within the sub-domain to which they have been assigned. In MBX, minimal changes were necessary to enable the calculation of the real-space interactions within each LAMMPS sub-domain containing local and ghost particles. Local particles are contained within the sub-domain owned by an MPI rank and ghost particles are replicated from neighboring sub-domains owned by other MPI ranks. For performance reasons, the iterative electrostatic solver in MBX was enabled with MPI and does not need to interact with LAMMPS during intermediate steps. In current CPU-only data-driven many-body simulations with MBX+LAMMPS, the performance bottleneck functions include evaluation of the n-body PIP terms, and calculation of the long-range portion of the electrostatic and dispersion interactions that include evaluation of distributed 3D Fast Fourier Transforms (FFTs). The electrostatic solver involves an iterative

TABLE I. Effective Lennard-Jones parameters for MB-pol water.

Atom	σ (Å)	ε (kcal/mol)
O	3.26393	0.26948
H	2.68354	3.7×10^{-10}

calculation of induced dipole moments requiring repeated communication with neighboring MPI ranks and evaluation of multiple 3D FFTs. These terms of the MB-nrg PEF along with all the others can be evaluated independently of one another and in arbitrary order.

The LAMMPS interface also enables hybrid FF/MB-nrg simulations where some interactions are described by conventional force fields (e.g., AMBER,[154] CHARMM,[155] and OPLS[156]) and other interactions are described by MB-nrg PEFs. In these hybrid simulations, the electrostatic energy is exclusively computed by MBX, while the remaining non-bonded interactions between FF and MB-nrg molecules are represented by Lennard-Jones potentials that can be derived using standard Lorentz-Berthelot mixing rules. In the case of FF molecules solvated in MB-pol water, the recommended effective Lennard-Jones parameters for MB-pol are listed in Table I.

Importantly, given its modularity and portability, MBX can be used in combination with any software package (e.g., in-house software developed within a research group) that supplies atom coordinates and expects energies and forces. MBX modules and sub-modules can be included by other C++ codes and `System` objects can be instantiated and used like any other C++ class. MBX also provides wrapper interfaces in C, FORTRAN and Python. The `System` class by itself is too big to be automatically adapted to other languages. However, for each one of the main `System` member function, there is a wrapper that enables calls from other programming languages. While not all of the member functions are wrapped, implementing a wrapper to retrieve a property that is currently not available is a simple and straightforward process.

V. PARALLELIZATION

In order to perform calculations on large systems, it is necessary to parallelize the evaluation of the various contributions to the total potential energy and forces. MBX exploits two sources of parallelization. Internally, MBX parallelizes the calculation of the various PEF contributions using OpenMP. Externally, MBX can exploit MPI parallelization schemes implementing domain decomposition which may be available in the interfaced molecular simulation software. For ex-

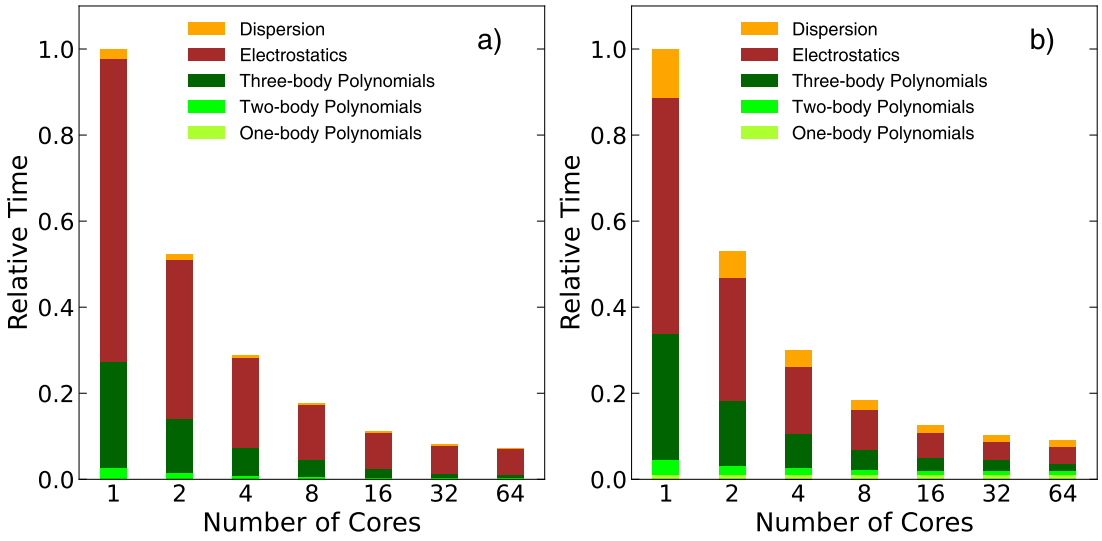

FIG. 3. Relative time to calculate all energies and gradients for a cubic box of 2048 water molecules in MBX in periodic boundary conditions. Calculations were each performed 100 times, and the average was taken. The relative times are presented as a function of the number of OpenMP threads used with MBX as a standalone code (a) and with LAMMPS using a single MPI rank (b), being the reference time the average time taken when using 1 OMP thread. All the calculations were performed on a compute node with two sockets each with 64 2.6GHz AMD 7H12 Rome processors.

ample, since LAMMPS is able to partition the simulation box into sub-domains overseen by individual MPI ranks, the MBX/LAMMPS interface allows each LAMMPS MPI rank to use one or more MBX OpenMP threads. This implies that both sources of parallelization (OpenMP in MBX and MPI in LAMMPS or other software) can be used together.

As a showcase of the OpenMP parallelization, Fig. 3 reports the mean runtime of an energy calculation for a box of 2048 water molecules as a function of the number of cores. The timings observed suggest that the OpenMP parallelization is efficient up to about 16 threads, after which MBX is not currently able to take full advantage of further parallelization through OpenMP. Also shown in Fig. 3 is the runtime when the calculations are performed within LAMMPS using a single MPI rank (and the indicated number of OpenMP threads). As expected, the scaling for both MBX as a standalone code and when interfaced with LAMMPS using a single MPI rank is essentially identical, since the OpenMP parallelization is internal to MBX. It should be noted here that, as is generally the case, the electrostatics represents the most expensive energy contribution to calculate. Since the i-PI interface utilizes no additional source of parallelization, the relative

times profile of MBX in i-PI is essentially identical to that obtained when MBX is interfaced with LAMMPS in Fig. 3.

When the simulations are driven by LAMMPS, MBX can also take advantage of parallelization over MPI ranks. Fig. 4 shows the relative times associated with the MBX energy and gradient calculations when interfaced with LAMMPS, utilizing several different combinations of MPI ranks and OpenMP threads. Comparing columns $[1,2]$ and $[1,4]$ with columns $[2,1]$ and $[4,1]$, it is clear that the OpenMP parallelization is more effective when the total number of available threads is small. However, as n_{OMP} gets larger and approaches the parallelization limit observed in Fig. 3, the use of MPI ranks is more effective in achieving the best performance. The optimal combination of OpenMP threads and MPI ranks depends on various factors, including the system's size and topology (i.e., cluster, bulk, or interface). It should be noted that the evaluation of all individual contributions to the energy scales relatively well with both MPI and OpenMP paralellization, with

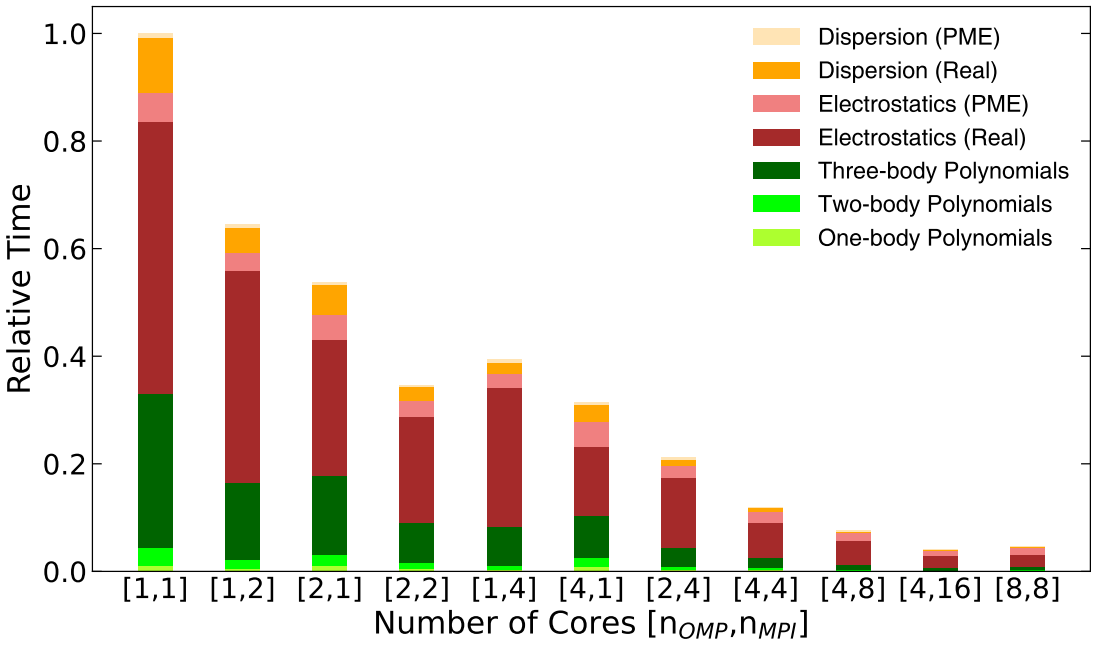

FIG. 4. Relative time to calculate all energies and gradients for a cubic box of 2048 water molecules in periodic boundary conditions using MBX interfaced with LAMMPS. Calculations were each performed 100 times, and the average was taken. The relative times are presented as a function of the number of OpenMP threads (n_{OMP}) per MPI rank and the number of MPI ranks (n_{MPI}), being the time corresponding to 1 OMP thread and 1 MPI rank the reference. Calculations were performed on a compute node with two sockets each with 64 2.6GHz AMD 7H12 Rome processors.

the exception of the PME part of the electrostatics, which will be the focus of further optimizations in the subsequent releases of MBX. The actual timings associated with the MBX energy and gradient calculations shown in Figs. 3 and 4 are reported in the Supplementary Material.

All timings reported in Figs. 3 and 4 were obtained for simulations of 2048 water molecules in a periodic cubic box carried out on a compute node with two sockets each with 64 2.6GHz AMD 7H12 Rome processors using a convergence threshold (ε) for the atomic induced dipole moments of 10^{-16}, which corresponds to each component of the induced dipole moment of each atom being converged up to the 8th decimal digit. The convergence criterion is met when the squared difference between successive iterations (k and k+1) of each induced dipole moment component (α) for each atom i, $\mu_{\mathrm{ind}_{i,\alpha}}$, is smaller than the tolerance ε:

$$\left(\mu_{\mathrm{ind}_{i,\alpha}}^{(k+1)} - \mu_{\mathrm{ind}_{i,\alpha}}^{(k)}\right)^2 < \varepsilon, \; \forall \, i, \alpha \tag{19}$$

A threshold $\varepsilon = 10^{-16}$ corresponds to a conservative and safe convergence criterion for all systems that we have simulated with our MB-nrg PEFs to date. However, it is worth noting that larger values up to $\varepsilon = 10^{-8}$ are sufficient for systems with weaker responses to electric fields (e.g., neat H_2O, CO_2, CH_4 solutions). A systematic analysis of the energy conservation and associated energy fluctuations for simulations of 2048 water molecules in a periodic cubic box carried out in the microcanonical (NVE = constant number of molecules, volume, and energy) ensemble as a function of the convergence tolerance is reported in the Supplementary Material.

VI. CONCLUSIONS

Over the last decade, data-driven many-body MB-nrg PEFs have been shown to accurately predict the properties of various molecular systems from the gas to the condensed phase. By integrating an underlying many-body polarizable model with explicit machine-learned representations of individual n-body interactions, MB-nrg PEFs achieve chemical accuracy in the representation of molecular interactions at both short and long range, and at all n-body orders.

In this work, we introduced MBX, a C++ modular library that enables MB-nrg energy and forces calculations. MBX is divided into modules responsible for particular tasks. The `potential` module is divided into sub-modules, each handling one specific energy contribution: n-body PIPs, dispersion energy, and electrostatics. Other modules are responsible for input/output, interfacing with drivers (e.g., software for MD and MC simulations), and constructing the `System` class that

stores the state of the molecular system.

While MBX can be used as a standalone software, it also provides interfaces to common MD packages such as i-PI and LAMMPS along with interfaces written in Fortran and Python that can be seamlessly used in combination with third-party software (e.g., in-house software developed by a research group). Both interfaces have already been used to study various molecular systems, including liquid water, CO_2/H_2O mixtures, CH_4/H_2O mixtures, hydrated alkali-metal ion clusters, and ionic solutions.

MBX includes an internal OpenMP parallelization that is more efficient when the number of threads is small. When interfaced with external software that provides its own MPI parallelization (e.g., LAMMPS), MBX enables efficient MB-nrg energy and force calculations that take advantage of both OpenMP and MPI parallelizations. Future versions of MBX will include improved parallelization schemes as well as the implementation of the extended MB-nrg framework introduced in Ref. 131 for covalently-bonded molecules, with the goal of enabling fast MB-nrg energy/force calculations which, in turn, will enable chemically accurate large-scale computer simulations of generic molecular systems.

VII. SUPPLEMENTARY MATERIAL

Description of the MBX input file formats and functional form of the switching functions for the MB-nrg PEFs.

VIII. ACKNOWLEDGEMENT

The authors thank Henry Agnew for his help with the computational performance analysis. Different aspects of this work were supported by the National Science Foundation through grants nos. CHE-1954895 and CHE-2102309 (overall software development and implementation), and the Air Force Office of Scientific Research through grant no. FA9550-16-1-0327 (PIP optimization). M.R. was partially supported by a Software Fellowship from the Molecular Sciences Software Institute (MolSSI), which was initially funded by the National Science Foundation through grant ACI-1547580. D.S. was supported by the Molecular Sciences Software Institute (MolSSI), which is funded by the National Science Foundation through grant no. CHE-2136142. A.S. was supported by the intramural research program of the National Heart, Lung, and Blood Institute. C.K.

was supported by the Office of Science, U.S. Department of Energy, under contract DE-AC02-06CH11357. This research used resources of the Extreme Science and Engineering Discovery Environment (XSEDE), which was supported by the National Science Foundation through grant no. ACI-1548562), the Department of Defense High Performance Computing Modernization Program (HPCMP), and the Triton Shared Computing Cluster (TSCC) at the San Diego Supercomputer Center.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY STATEMENT

Any data generated and analyzed in this study are available from the authors upon request. MBX can be downloaded from https://github.com/paesanilab/MBX.

REFERENCES

[1]D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, Vol. 1 (Elsevier, 2001).

[2]M. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010).

[3]W. F. van Gunsteren and H. J. Berendsen, "Computer simulation of molecular dynamics: Methodology, applications, and perspectives in chemistry," Angew. Chem. Int. Ed. **29**, 992–1023 (1990).

[4]K. Binder, *Monte Carlo and Molecular Dynamics Simulations in Polymer Science* (Oxford University Press, 1995).

[5]A. Warshel, "Computer simulations of enzyme catalysis: Methods, progress, and insights," Annu. Rev. Biophys. Biomol. Struct. **32**, 425–443 (2003).

[6]M. Karplus and J. Kuriyan, "Molecular dynamics and protein function," Proc. Natl. Acad. Sci. U.S.A. **102**, 6679–6685 (2005).

[7]J. D. Durrant and J. A. McCammon, "Molecular dynamics simulations and drug discovery," BMC Biol. **9**, 71 (2011).

[8]K. Ohno, K. Esfarjani, and Y. Kawazoe, *Computational Materials Science: From Ab Initio to Monte Carlo Methods* (Springer, 2018).

[9]S. Lifson and A. Warshel, "Consistent force field for calculations of conformations, vibrational spectra, and enthalpies of cycloalkane and n-alkane molecules," J. Chem. Phys. **49**, 5116–5129 (1968).

[10]A. Warshel and S. Lifson, "Consistent force field calculations. II. Crystal structures, sublimation energies, molecular and lattice vibrations, molecular conformations, and enthalpies of alkanes," J. Chem. Phys. **53**, 582–594 (1970).

[11]A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, and W. M. Skiff, "UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations," J. Am. Chem. Soc. **114**, 10024–10035 (1992).

[12]T. A. Halgren and W. Damm, "Polarizable force fields," Curr. Opin. Struct. Biol. **11**, 236–242 (2001).

[13]A. D. MacKerell Jr, "Empirical force fields for biological macromolecules: Overview and issues," J. Comput. Chem. **25**, 1584–1604 (2004).

[14]P. S. Nerenberg and T. Head-Gordon, "New developments in force fields for biomolecular simulations," Curr. Opin. Struct. Biol. **49**, 129–138 (2018).

[15]J. A. Harrison, J. D. Schall, S. Maskey, P. T. Mikulski, M. T. Knippenberg, and B. H. Morrow, "Review of force fields and intermolecular potentials used in atomistic computational materials research," Appl. Phys. Rev **5**, 031104 (2018).

[16]J. Behler, "Perspective: Machine learning potentials for atomistic simulations," J. Chem. Phys. **145**, 170901 (2016).

[17]V. L. Deringer, M. A. Caro, and G. Csányi, "Machine learning interatomic potentials as emerging tools for materials science," Adv. Mater. **31**, 1902765 (2019).

[18]F. Noé, A. Tkatchenko, K.-R. Müller, and C. Clementi, "Machine learning for molecular simulation," Annu. Rev. Phys Chem. **71**, 361–390 (2020).

[19]T. Mueller, A. Hernandez, and C. Wang, "Machine learning for interatomic potential models," J. Chem. Phys. **152**, 050902 (2020).

[20]T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, "Neural network models of potential energy surfaces," J. Chem. Phys. **103**, 4129–4137 (1995).

[21]H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson, "Representation of intermolecular potential functions by neural networks," J. Phys. Chem. A **102**, 4596–4605 (1998).

[22]S. Lorenz, A. Groß, and M. Scheffler, "Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks," Chem. Phys. Lett. **395**, 210–215 (2004).

[23]S. Manzhos and T. Carrington Jr, "Using neural networks to represent potential surfaces as sums of products," J. Chem. Phys. **125**, 194105 (2006).

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

[25]S. A. Ghasemi, A. Hofstetter, S. Saha, and S. Goedecker, "Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network," Phys. Rev. B **92**, 045131 (2015).

[26]J. S. Smith, O. Isayev, and A. E. Roitberg, "ANI-1: An extensible neural network potential with dft accuracy at force field computational cost," Chem. Sci. **8**, 3192–3203 (2017).

[27]K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "Schnet – A deep learning architecture for molecules and materials," J. Chem. Phys. **148**, 241722 (2018).

[28]L. Zhang, J. Han, H. Wang, R. Car, and E. Weinan, "Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics," Phys. Rev. Lett. **120**, 143001 (2018).

[29]O. T. Unke and M. Meuwly, "Physnet: A neural network for predicting energies, forces, dipole moments, and partial charges," J. Chem. Theory Comput. **15**, 3678–3693 (2019).

[30]S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," Nat. Commun. **13**, 2453 (2022).

[31]A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**, 136403 (2010).

[32]A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," Multiscale Model. Simul. **14**, 1153–1173 (2016).

[33]A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials," J. Comput. Phys. **285**, 316–330 (2015).

[34]R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," Phys. Rev. B **99**, 014104 (2019).

[35]M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).

[36]S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, "Machine learning of accurate energy-conserving molecular force fields," Sci. Adv. **3**, e1603015 (2017).

[37]A. Vitek, M. Stachon, P. Krömer, and V. Snáel, "Towards the modeling of atomic and molecular clusters energy by support vector regression," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems* (IEEE, 2013) pp. 121–126.

[38]B. J. Braams and J. M. Bowman, "Permutationally invariant potential energy surfaces in high dimensionality," Int. Rev. Phys. Chem. **28**, 577–606 (2009).

[39]Z. Xie and J. M. Bowman, "Permutationally invariant polynomial basis for molecular energy surface fitting via monomial symmetrization," J. Chem. Theory Comput. **6**, 26–34 (2010).

[40]Y. Wang, X. Huang, B. C. Shepler, B. J. Braams, and J. M. Bowman, "Flexible, ab initio potential, and dipole moment surfaces for water. I. Tests and applications for clusters up to the 22-mer," J. Chem. Phys. **134**, 094509 (2011).

[41]Y. Wang and J. M. Bowman, "Ab initio potential and dipole moment surfaces for water. II. Local-monomer calculations of the infrared spectra of water clusters," J. Chem. Phys. **134**, 154510 (2011).

[42]P. Barragán, R. Prosmiti, Y. Wang, and J. M. Bowman, "Full-dimensional (15-dimensional) ab initio analytical potential energy surface for the H_7^+ cluster," J. Chem. Phys. **136**, 224302 (2012).

[43]J. S. Mancini and J. M. Bowman, "Communication: A new ab initio potential energy surface for HCl–H_2O, diffusion Monte Carlo calculations of d_0 and a delocalized zero-point wavefunction," J. Chem. Phys. **138**, 121102 (2013).

[44]E. Kamarchik, D. Toffoli, O. Christiansen, and J. M. Bowman, "Ab initio potential energy and dipole moment surfaces of the $F^-(H_2O)$ complex," Spectrochim. Acta A Mol. Biomol. Spectrosc. **119**, 59–62 (2014).

[45]R. Conte, P. L. Houston, and J. M. Bowman, "Communication: A benchmark-quality, full-dimensional ab initio potential energy surface for Ar-HOCO," J. Chem. Phys. **140**, 151101 (2014).

[46]J. S. Mancini and J. M. Bowman, "A new many-body potential energy surface for HCl clusters and its application to anharmonic spectroscopy and vibration–vibration energy transfer in the

HCl trimer," J. Phys. Chem. A **118**, 7367–7374 (2014).

[47]C. Qu, R. Conte, P. L. Houston, and J. M. Bowman, ""Plug and play" full-dimensional ab initio potential energy and dipole moment surfaces and anharmonic vibrational analysis for CH_4–H_2O," Phys. Chem. Chem. Phys. **17**, 8172–8181 (2015).

[48]R. Conte, C. Qu, and J. M. Bowman, "Permutationally invariant fitting of many-body, non-covalent interactions with application to three-body methane–water–water," J. Chem. Theory Comput. **11**, 1631–1638 (2015).

[49]Z. Homayoon, R. Conte, C. Qu, and J. M. Bowman, "Full-dimensional, high-level ab initio potential energy surfaces for $H_2(H_2O)$ and $H_2(H_2O)_2$ with application to hydrogen clathrate hydrates," J. Chem. Phys. **143**, 084302 (2015).

[50]C. Qu and J. M. Bowman, "An ab initio potential energy surface for the formic acid dimer: Zero-point energy, selected anharmonic fundamental energies, and ground-state tunneling splitting calculated in relaxed 1–4-mode subspaces," Phys. Chem. Chem. Phys. **18**, 24835–24840 (2016).

[51]Y. Wang, J. M. Bowman, and E. Kamarchik, "Five ab initio potential energy and dipole moment surfaces for hydrated NaCl and NaF. I. Two-body interactions," J. Chem. Phys. **144**, 114311 (2016).

[52]Q. Yu and J. M. Bowman, "Ab initio potential for $H_3O^+ \rightarrow H^+ + H_2O$: A step to a many-body representation of the hydrated proton?" J. Chem. Theory Comput. **12**, 5284–5292 (2016).

[53]Q. Wang and J. M. Bowman, "Two-component, ab initio potential energy surface for CO_2–H_2O, extension to the hydrate clathrate, $CO_2@(H_2O)_{20}$, and vscf/vci vibrational analyses of both," J. Chem. Phys. **147**, 161714 (2017).

[54]C. Qu, Q. Yu, and J. M. Bowman, "Permutationally invariant potential energy surfaces," Annu. Rev. Phys. Chem. **69**, 151–175 (2018).

[55]C. Qu and J. M. Bowman, "IR spectra of $(HCOOH)_2$ and $(DCOOH)_2$: Experiment, VSCF/VCI, and ab initio molecular dynamics calculations using full-dimensional potential and dipole moment surfaces," J. Phys. Chem. Lett. **9**, 2604–2610 (2018).

[56]C. Qu and J. M. Bowman, "High-dimensional fitting of sparse datasets of CCSD(T) electronic energies and MP2 dipole moments, illustrated for the formic acid dimer and its complex IR spectrum," J. Chem. Phys. **148**, 241713 (2018).

[57]C. Qu and J. M. Bowman, "Assessing the importance of the H_2–H_2O–H_2O three-body interaction on the vibrational frequency shift of H_2 in the sII clathrate hydrate and comparison with

experiment," J. Phys. Chem. A **123**, 329–335 (2018).

[58]A. Nandi, C. Qu, and J. M. Bowman, "Full and fragmented permutationally invariant polynomial potential energy surfaces for trans and cis N-methyl acetamide and isomerization saddle points," J. Chem. Phys. **151**, 084306 (2019).

[59]C. Qu and J. M. Bowman, "A fragmented, permutationally invariant polynomial approach for potential energy surfaces of large molecules: Application to N-methyl acetamide," J. Chem. Phys. **150**, 141101 (2019).

[60]A. Nandi, C. Qu, and J. M. Bowman, "Using gradients in permutationally invariant polynomial potential fitting: A demonstration for CH_4 using as few as 100 configurations," J. Chem. Theory Comput. **15**, 2826–2835 (2019).

[61]A. Nandi, C. Qu, P. L. Houston, R. Conte, Q. Yu, and J. M. Bowman, "A CCSD(T)-based 4-body potential for water," J. Phys. Chem. Lett. **12**, 10318–10324 (2021).

[62]A. Nandi, C. Qu, P. L. Houston, R. Conte, and J. M. Bowman, "δ-machine learning for potential energy surfaces: A PIP approach to bring a DFT-based PES to CCSD(T) level of theory," J. Chem. Phys. **154**, 051102 (2021).

[63]B. Jiang and H. Guo, "Permutation invariant polynomial neural network approach to fitting potential energy surfaces," J. Chem. Phys. **139**, 054112 (2013).

[64]J. Li, B. Jiang, and H. Guo, "Permutation invariant polynomial neural network approach to fitting potential energy surfaces. II. Four-atom systems," J. Chem. Phys. **139**, 204103 (2013).

[65]B. Jiang and H. Guo, "Permutation invariant polynomial neural network approach to fitting potential energy surfaces. III. Molecule-surface interactions," J. Chem. Phys. **141**, 034109 (2014).

[66]C. Xie, X. Zhu, D. R. Yarkony, and H. Guo, "Permutation invariant polynomial neural network approach to fitting potential energy surfaces. IV. Coupled diabatic potential energy matrices," J. Chem. Phys. **149**, 144107 (2018).

[67]T. Morawietz and J. Behler, "A density-functional theory-based neural network potential for water clusters including van der Waals corrections," J. Phys. Chem. A **117**, 7356–7366 (2013).

[68]C. Schran, J. Behler, and D. Marx, "Automated fitting of neural network potentials at coupled cluster accuracy: Protonated water clusters as testing ground," J. Chem. Theory Comput. **16**, 88–99 (2019).

[69]D. Rosenberger, J. S. Smith, and A. E. Garcia, "Modeling of peptides with classical and novel machine learning force fields: A comparison," J. Phys. Chem. B **125**, 3598–3612 (2021).

[70]S. Yue, M. C. Muniz, M. F. Calegari Andrade, L. Zhang, R. Car, and A. Z. Panagiotopoulos, "When do short-range atomistic machine-learning models fall short?" J. Chem. Phys. **154**, 034111 (2021).

[71]Y. Zhai, A. Caruso, S. L. Bore, Z. Luo, and F. Paesani, "A "short blanket" dilemma for a state-of-the-art neural network potential for water: Reproducing experimental properties or the physics of the underlying many-body interactions?" J. Chem. Phys. **158**, 084111 (2023).

[72]V. Babin, C. Leforestier, and F. Paesani, "Development of a "first principles" water potential with flexible monomers: Dimer potential energy surface, VRT spectrum, and second virial coefficient," J. Chem. Theory Comput. **9**, 5395–5403 (2013).

[73]V. Babin, G. R. Medders, and F. Paesani, "Development of a "first principles" water potential with flexible monomers. II: Trimer potential energy surface, third virial coefficient, and small clusters," J. Chem. Theory Comput. **10**, 1599–1607 (2014).

[74]G. R. Medders, V. Babin, and F. Paesani, "Development of a "first-principles" water potential with flexible monomers. III. Liquid phase properties," J. Chem. Theory Comput. **10**, 2906–2910 (2014).

[75]B. B. Bizzarro, C. K. Egan, and F. Paesani, "Nature of halide–water interactions: Insights from many-body representations and density functional theory," J. Chem. Theory Comput. **15**, 2983–2995 (2019).

[76]C. K. Egan, B. B. Bizzarro, M. Riera, and F. Paesani, "Nature of alkali ion–water interactions: Insights from many-body representations and density functional theory. II," J. Chem. Theory Comput. **16**, 3055–3072 (2020).

[77]F. Paesani, "Water: Many-body potential from first principles (from the gas to the liquid phase)," in *Handbook of Materials Modeling: Methods: Theory and Modeling*, edited by W. Andreoni and S. Yip (Springer, 2020) pp. 635–660.

[78]J. Rezac and P. Hobza, "Benchmark calculations of interaction energies in noncovalent complexes and their applications," Chem. Rev. **116**, 5038–5071 (2016).

[79]S. K. Reddy, S. C. Straight, P. Bajaj, C. Huy Pham, M. Riera, D. R. Moberg, M. A. Morales, C. Knight, A. W. Götz, and F. Paesani, "On the accuracy of the MB-pol many-body potential for water: Interaction energies, vibrational frequencies, and classical thermodynamic and dynamical properties from clusters to liquid water and ice," J. Chem. Phys. **145**, 194504 (2016).

[80]F. Paesani, "Getting the right answers for the right reasons: Toward predictive molecular simulations of water with many-body potential energy functions," Acc. Chem. Res. **49**, 1844–1851

(2016).

[81]J. O. Richardson, C. Pérez, S. Lobsiger, A. A. Reid, B. Temelso, G. C. Shields, Z. Kisiel, D. J. Wales, B. H. Pate, and S. C. Althorpe, "Concerted hydrogen-bond breaking by quantum tunneling in the water hexamer prism," Science **351**, 1310–1313 (2016).

[82]W. T. Cole, J. D. Farrell, D. J. Wales, and R. J. Saykally, "Structure and torsional dynamics of the water octamer from thz laser spectroscopy near 215 μm," Science **352**, 1194–1197 (2016).

[83]J. D. Mallory and V. A. Mandelshtam, "Diffusion Monte Carlo studies of MB-pol $(H_2O)_{2-6}$ and $(D_2O)_{2-6}$ clusters: Structures and binding energies," J. Chem. Phys. **145**, 064308 (2016).

[84]P. E. Videla, P. J. Rossky, and D. Laria, "Communication: Isotopic effects on tunneling motions in the water trimer," J. Chem. Phys. **144**, 061101 (2016).

[85]S. E. Brown, A. W. Götz, X. Cheng, R. P. Steele, V. A. Mandelshtam, and F. Paesani, "Monitoring water clusters "melt" through vibrational spectroscopy," J. Am. Chem. Soc. **139**, 7082–7088 (2017).

[86]C. L. Vaillant and M. T. Cvitaš, "Rotation-tunneling spectrum of the water dimer from instanton theory," Phys. Chem. Chem. Phys. **20**, 26809–26813 (2018).

[87]C. Vaillant, D. Wales, and S. Althorpe, "Tunneling splittings from path-integral molecular dynamics using a Langevin thermostat," J. Chem. Phys. **148**, 234102 (2018).

[88]M. Schmidt and P.-N. Roy, "Path integral molecular dynamic simulation of flexible molecular systems in their ground state: Application to the water dimer," J. Chem. Phys. **148**, 124116 (2018).

[89]K. P. Bishop and P.-N. Roy, "Quantum mechanical free energy profiles with post-quantization restraints: Binding free energy of the water dimer over a broad range of temperatures," J. Chem. Phys. **148**, 102303 (2018).

[90]P. E. Videla, P. J. Rossky, and D. Laria, "Isotopic equilibria in aqueous clusters at low temperatures: Insights from the MB-pol many-body potential," J. Chem. Phys. **148**, 084303 (2018).

[91]N. R. Samala and N. Agmon, "Temperature dependence of intramolecular vibrational bands in small water clusters," J. Phys. Chem. B **123**, 9428–9442 (2019).

[92]M. T. Cvitaš and J. O. Richardson, "Quantum tunnelling pathways of the water pentamer," Phys. Chem. Chem. Phys. **22**, 1035–1044 (2020).

[93]G. R. Medders and F. Paesani, "Infrared and Raman spectroscopy of liquid water through "first-principles" many-body molecular dynamics," J. Chem. Theory Comput. **11**, 1145–1154 (2015).

[94]S. C. Straight and F. Paesani, "Exploring electrostatic effects on the hydrogen bond network of liquid water through many-body molecular dynamics," J. Phys. Chem. B **120**, 8539–8546 (2016).

[95]S. K. Reddy, D. R. Moberg, S. C. Straight, and F. Paesani, "Temperature-dependent vibrational spectra and structure of liquid water from classical and quantum simulations with the MB-pol potential energy function," J. Chem. Phys. **147**, 244504 (2017).

[96]K. M. Hunter, F. A. Shakib, and F. Paesani, "Disentangling coupling effects in the infrared spectra of liquid water," J. Phys. Chem. B **122**, 10754–10761 (2018).

[97]Z. Sun, L. Zheng, M. Chen, M. L. Klein, F. Paesani, and X. Wu, "Electron-hole theory of the effect of quantum nuclei on the X-ray absorption spectra of liquid water," Phys. Rev. Lett. **121**, 137401 (2018).

[98]A. P. Gaiduk, T. A. Pham, M. Govoni, F. Paesani, and G. Galli, "Electron affinity of liquid water," Nat. Commun. **9**, 247 (2018).

[99]V. Cruzeiro, A. Wildman, X. Li, and F. Paesani, "Relationship between hydrogen-bonding motifs and the $1b_1$ splitting in the X-ray emission spectrum of liquid water," J. Phys. Chem. Lett. **12**, 3996–4002 (2021).

[100]G. R. Medders and F. Paesani, "Dissecting the molecular structure of the air/water interface from quantum simulations of the sum-frequency generation spectrum," J. Am. Chem. Soc. **138**, 3912–3919 (2016).

[101]D. R. Moberg, S. C. Straight, and F. Paesani, "Temperature dependence of the air/water interface revealed by polarization sensitive sum-frequency generation spectroscopy," J. Phys. Chem. B **122**, 4356–4365 (2018).

[102]S. Sun, F. Tang, S. Imoto, D. R. Moberg, T. Ohto, F. Paesani, M. Bonn, E. H. Backus, and Y. Nagata, "Orientational distribution of free OH groups of interfacial water is exponential," Phys. Rev. Lett. **121**, 246101 (2018).

[103]S. Sengupta, D. R. Moberg, F. Paesani, and E. Tyrode, "Neat water–vapor interface: Proton continuum and the nonresonant background," J. Phys. Chem. Lett. **9**, 6744–6749 (2018).

[104]M. C. Muniz, T. E. Gartner III, M. Riera, C. Knight, S. Yue, F. Paesani, and A. Z. Panagiotopoulos, "Vapor-liquid equilibrium of water with the MB-pol many-body potential," J. Chem. Phys. **154**, 211103 (2021).

[105]C. H. Pham, S. K. Reddy, K. Chen, C. Knight, and F. Paesani, "Many-body interactions in ice," J. Chem. Theory Comput. **13**, 1778–1784 (2017).

[106]D. R. Moberg, S. C. Straight, C. Knight, and F. Paesani, "Molecular origin of the vibrational structure of ice I_h," J. Phys. Chem. Lett. **8**, 2579–2583 (2017).

[107]D. R. Moberg, P. J. Sharp, and F. Paesani, "Molecular-level interpretation of vibrational spectra of ordered ice phases," J. Phys. Chem. B **122**, 10572–10581 (2018).

[108]D. R. Moberg, D. Becker, C. W. Dierking, F. Zurheide, B. Bandow, U. Buck, A. Hudait, V. Molinero, F. Paesani, and T. Zeuch, "The end of ice I," Proc. Natl. Acad. Sci. U.S.A. **116**, 24413–24419 (2019).

[109]L. del Rosso, M. Celli, D. Colognesi, S. Rudic, N. J. English, and L. Ulivi, "Density of phonon states in cubic ice ic," J. Phys. Chem. C **125**, 23533–23538 (2021).

[110]S. Rasti, E. Ö. Jónsson, H. Jónsson, and J. Meyer, "New insights into the volume isotope effect of ice ih from polarizable many-body potentials," J. Phys. Chem. Lett. **13**, 11831–11836 (2022).

[111]S. L. Bore and F. Paesani, "Realistic phase diagram of water from "first principles" data-driven quantum simulations," Nat. Commun. **14**, 3349 (2023).

[112]X. Zhu, M. Riera, E. F. Bull-Vulpe, and F. Paesani, "MB-pol(2023): Sub-chemical accuracy for water simulations from the gas to the liquid phase," J. Chem. Theory Comput. **19**, 3551–3566 (2023).

[113]P. Bajaj, A. W. Götz, and F. Paesani, "Toward chemical accuracy in the description of ion–water interactions through many-body representations. I. Halide–water dimer potential energy surfaces," J. Chem. Theory Comput. **12**, 2698–2705 (2016).

[114]P. Bajaj, X.-G. Wang, T. Carrington Jr, and F. Paesani, "Vibrational spectra of halide–water dimers: Insights on ion hydration from full-dimensional quantum calculations on many-body potential energy surfaces," J. Chem. Phys. **148**, 102321 (2018).

[115]P. Bajaj, J. O. Richardson, and F. Paesani, "Ion-mediated hydrogen-bond rearrangement through tunnelling in the iodide–dihydrate complex," Nat. Chem. **11**, 367 (2019).

[116]P. Bajaj, D. Zhuang, and F. Paesani, "Specific ion effects on hydrogen-bond rearrangements in the halide–dihydrate complexes," J. Phys. Chem. Lett. **10**, 2823–2828 (2019).

[117]P. Bajaj, M. Riera, J. K. Lin, Y. E. Mendoza Montijo, J. Gazca, and F. Paesani, "Halide ion microhydration: Structure, energetics, and spectroscopy of small halide–water clusters," J. Phys. Chem. A **123**, 2843–2852 (2019).

[118]A. Caruso and F. Paesani, "Data-driven many-body models enable a quantitative description of chloride hydration from clusters to bulk," J. Chem. Phys. **155**, 064502 (2021).

[119]A. Caruso, X. Zhu, J. L. Fulton, and F. Paesani, "Accurate modeling of bromide and iodide hydration with data-driven many-body potentials," J. Phys. Chem. B **126**, 8266–8278 (2022).

[120]M. Riera, N. Mardirossian, P. Bajaj, A. W. Götz, and F. Paesani, "Toward chemical accuracy in the description of ion–water interactions through many-body representations. Alkali-water dimer potential energy surfaces," J. Chem. Phys. **147**, 161715 (2017).

[121]M. Riera, S. E. Brown, and F. Paesani, "Isomeric equilibria, nuclear quantum effects, and vibrational spectra of $M^+(H_2o)_{n=1-3}$ clusters, with M = Li, Na, K, Rb, and Cs, through many-body representations," J. Phys. Chem. A **122**, 5811–5821 (2018).

[122]D. Zhuang, M. Riera, G. K. Schenter, J. L. Fulton, and F. Paesani, "Many-body effects determine the local hydration structure of Cs^+ in solution," J. Phys. Chem. Lett. **10**, 406–412 (2019).

[123]M. Riera, J. J. Talbot, R. P. Steele, and F. Paesani, "Infrared signatures of isomer selectivity and symmetry breaking in the $Cs^+(H_2O)_3$ complex using many-body potential energy functions," J. Chem. Phys **153**, 044306 (2020).

[124]D. Zhuang, M. Riera, R. Zhou, A. Deary, and F. Paesani, "Hydration structure of Na^+ and K^+ ions in solution predicted by data-driven many-body potentials," J. Phys. Chem. B **126**, 9349–9360 (2022).

[125]M. Riera, E. P. Yeh, and F. Paesani, "Data-driven many-body models for molecular fluids: CO_2/H_2O mixtures as a case study," J. Chem. Theory Comput. **16**, 2246–2257 (2020).

[126]M. Riera, A. Hirales, R. Ghosh, and F. Paesani, "Data-driven many-body models with chemical accuracy for CH_4/H_2O mixtures," J. Chem. Phys. B **124**, 11207–11221 (2020).

[127]S. Yue, M. Riera, R. Ghosh, A. Z. Panagiotopoulos, and F. Paesani, "Transferability of data-driven, many-body models for CO_2 simulations in the vapor and liquid phases," J. Chem. Phys. **156**, 104503 (2022).

[128]V. N. Robinson, R. Ghosh, C. K. Egan, M. Riera, C. Knight, F. Paesani, and A. Hassanali, "The behavior of methane–water mixtures under elevated pressures from simulations using many-body potentials," J. Chem. Phys. **156**, 194504 (2022).

[129]V. W. D. Cruzeiro, E. Lambros, M. Riera, R. Roy, F. Paesani, and A. W. Gotz, "Highly accurate many-body potentials for simulations of N_2O_5 in water: Benchmarks, development, and validation," J. Chem. Theory Comput. **17**, 3931–3945 (2021).

[130]R. Zhou, M. Riera, and F. Paesani, "Towards data-driven many-body simulations of biomolecules in solution: N-methyl acetamide as a proxy for the protein backbone," J. Chem

Theory Comput. **19**, 4308–4321 (2023).

[131]E. F. Bull-Vulpe, M. Riera, S. L. Bore, and F. Paesani, "Data-driven many-body potential energy functions for generic molecules: Linear alkanes as a proof-of-concept application," J. Chem. Theory Comput. **XXX**, in press, https://doi.org/10.1021/acs.jctc.2c00645 (2023).

[132]"MBX: An energy and force calculator for data-driven many-body potential energy functions," `http://paesanigroup.ucsd.edu/software/mbx.html` (2019).

[133]E. F. Bull-Vulpe, M. Riera, A. W. Götz, and F. Paesani, "MB-Fit: Software infrastructure for data-driven many-body potential energy functions," J. Chem. Phys. **155**, 124801 (2021).

[134]"MB-Fit: Software infrastructure for data-driven many-body potential energy functions," `https://github.com/paesanilab/MB-Fit` (2021).

[135]R. K. Nesbet, "Atomic Bethe-Goldstone equations," in *Advances in Chemical Physics* (John Wiley & Sons, Ltd, 1969) pp. 1–34.

[136]D. Hankins, J. W. Moskowitz, and F. H. Stillinger, "Water molecule interactions," J. Chem. Phys. **53**, 4544–4554 (1970).

[137]H. Stoll, "Correlation energy of diamond," Phys. Rev. B **46**, 6700 (1992).

[138]H. Stoll, "On the correlation energy of graphite," J. Chem. Phys. **97**, 8449–8454 (1992).

[139]H. Stoll, "The correlation energy of crystalline silicon," Chem. Phys. Lett. **191**, 548–552 (1992).

[140]A. J. Stone, *The Theory of Intermolecular Forces* (Oxford University Press, Oxford, 2013).

[141]K. T. Tang and J. P. Toennies, "An improved simple model for the van der waals potential based on universal damping functions for the dispersion coefficients," J. Chem. Phys. **80**, 3726–3741 (1984).

[142]A. D. Becke and E. R. Johnson, "Exchange-hole dipole moment and the dispersion interaction," J. Chem. Phys. **122**, 154104 (2005).

[143]E. R. Johnson and A. D. Becke, "A post-Hartree–Fock model of intermolecular interactions," J. Chem. Phys. **123**, 024101 (2005).

[144]E. R. Johnson and A. D. Becke, "A post-Hartree-Fock model of intermolecular interactions: Inclusion of higher-order corrections," J. Chem. Phys. **124**, 174104 (2006).

[145]B. T. Thole, "Molecular polarizabilities calculated with a modified dipole interaction," Chem. Phys. **59**, 341–350 (1981).

[146]C. J. Burnham, D. J. Anick, P. K. Mankoo, and G. F. Reiter, "The vibrational proton potential in bulk liquid water and ice," J. Chem. Phys. **128**, 154519 (2008).

[147]J. Sala, E. Guardia, and M. Masia, "The polarizable point dipoles method with electrostatic damping: Implementation on a model system," J. Chem. Phys. **133**, 234101 (2010).

[148]J. L. Blanco and P. K. Rai, "nanoflann: A C++ header-only fork of FLANN, a library for nearest neighbor (NN) with KD-trees," `https://github.com/jlblancoc/nanoflann` (2014).

[149]A. C. Simmonett and B. R. Brooks, "Analytical hessians for ewald and particle mesh ewald electrostatics," J. Chem. Phys. **154**, 104101 (2021).

[150]A. C. Simmonett and B. R. Brooks, "A compression strategy for particle mesh ewald theory," J. Chem. Phys. **154**, 054112 (2021).

[151]J. Kolafa, "Time-reversible always stable predictor–corrector method for molecular dynamics of polarizable molecules," J. Comput. Chem. **25**, 335–342 (2004).

[152]A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," Comp. Phys. Comm. **271**, 108171 (2022).

[153]V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bienvenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. Van Speybroeck, and M. Ceriotti, "i-PI 2.0: A universal force engine for advanced molecular simulations," Comput. Phys. Commun. **236**, 214–223 (2019).

[154]W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," J. Am. Chem. Soc. **117**, 5179–5197 (1995).

[155]B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, "CHARMM: A program for macromolecular energy, minimization, and dynamics calculations," J. Comput. Chem. **4**, 187–217 (1983).

[156]W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids," J. Am. Chem. Soc. **118**, 11225–11236 (1996).

GOVERNMENT LICENSE