The Journal of Chemical Physics

RESEARCH ARTICLE | NOVEMBER 07 2023

libMBD: A general-purpose package for scalable quantum many-body dispersion calculations

Special Collection: Modular and Interoperable Software for Chemical Physics

Jan Hermann ✉ ; Martin Stöhr ; Szabolcs Góger ; Shayantan Chaudhuri ; Bálint Aradi ; Reinhard J. Maurer ; Alexandre Tkatchenko ✉

Check for updates

J. Chem. Phys. 159, 174802 (2023)
https://doi.org/10.1063/5.0170972

View Online

Export Citation

CrossMark

AIP Publishing

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

libMBD: A general-purpose package for scalable quantum many-body dispersion calculations

Cite as: J. Chem. Phys. 159, 174802 (2023); doi: 10.1063/5.0170972
Submitted: 5 August 2023 • Accepted: 17 October 2023 •
Published Online: 7 November 2023

View Online Export Citation CrossMark

Jan Hermann,[1,a),b)] Martin Stöhr,[2] Szabolcs Góger,[2] Shayantan Chaudhuri,[3,c)] Bálint Aradi,[4] Reinhard J. Maurer,[3,5] and Alexandre Tkatchenko[2,a)]

AFFILIATIONS

[1] Department of Mathematics and Computer Science, FU Berlin, 14195 Berlin, Germany
[2] Department of Physics and Materials Science, University of Luxembourg, L-1511 Luxembourg City, Luxembourg
[3] Department of Chemistry, University of Warwick, Coventry CV4 7AL, United Kingdom
[4] Bremen Center for Computational Materials Science, University of Bremen, 28359 Bremen, Germany
[5] Department of Physics, University of Warwick, Coventry CV4 7AL, United Kingdom

Note: This paper is part of the JCP Special Topic on Modular and Interoperable Software for Chemical Physics.
[a)] **Authors to whom correspondence should be addressed:** science@jan.hermann.name and alexandre.tkatchenko@uni.lu
[b)] **Current address:** Microsoft Research AI4Science, Karl-Liebknecht-Str. 32, 10178 Berlin, Germany.
[c)] **Current address:** University of Nottingham, School of Chemistry, Nottingham, NG7 2RD, United Kingdom.

ABSTRACT

Many-body dispersion (MBD) is a powerful framework to treat van der Waals (vdW) dispersion interactions in density-functional theory and related atomistic modeling methods. Several independent implementations of MBD with varying degree of functionality exist across a number of electronic structure codes, which both limits the current users of those codes and complicates dissemination of new variants of MBD. Here, we develop and document libMBD, a library implementation of MBD that is functionally complete, efficient, easy to integrate with any electronic structure code, and already integrated in FHI-aims, DFTB+, VASP, Q-Chem, CASTEP, and Quantum ESPRESSO. libMBD is written in modern Fortran with bindings to C and Python, uses MPI/ScaLAPACK for parallelization, and implements MBD for both finite and periodic systems, with analytical gradients with respect to all input parameters. The computational cost has asymptotic cubic scaling with system size, and evaluation of gradients only changes the prefactor of the scaling law, with libMBD exhibiting strong scaling up to 256 processor cores. Other MBD properties beyond energy and gradients can be calculated with libMBD, such as the charge-density polarization, first-order Coulomb correction, the dielectric function, or the order-by-order expansion of the energy in the dipole interaction. Calculations on supramolecular complexes with MBD-corrected electronic structure methods and a meta-review of previous applications of MBD demonstrate the broad applicability of the libMBD package to treat vdW interactions.

I. INTRODUCTION

Van der Waals (vdW) dispersion interactions stem from electron correlation induced by the long-range part of the Coulomb force. Since they are attractive under normal circumstances, their importance grows with the characteristic length scale in an atomistic system, making them an often decisive force in large molecular complexes, molecular crystals, nanostructured materials, surface phenomena, and soft matter in general. At the same time, the workhorse electronic structure method for simulating molecules and materials—Kohn–Sham density-functional theory (KS-DFT)—usually neglects vdW interactions in the popular semilocal and hybrid approximations. As a result, a large number of approaches to mitigate this issue have been developed, and some of them are now used routinely when performing DFT calculations.[1–3]

One such popular approach is the many-body dispersion (MBD) method, which is based on a model Hamiltonian of charged harmonic oscillators that capture the long-range electrodynamic response of an atomic system and yield its vdW energy.[4] Its essential offering is that the coarse-graining of the full electronic structure to oscillators makes it efficient, while full many-body treatment of the electronic fluctuations provides consistent description of beyond-pairwise vdW interactions.[5] Both these characteristics make the

MBD framework especially apt for large-scale modeling of molecular systems, for which the demand is bound to only increase in the future.[6] Given the existence of MBD integrations with even more efficient substitutes of DFT based on density-functional tight binding (DFTB)[7] and machine learning,[8–11] as well as the fact that MBD is an approach that is still evolving and being improved and extended (see below), the situation calls for an efficient, flexible, and reusable implementation of MBD that can be easily deployed in any given electronic structure code. Here, we develop and document libMBD, a library package that answers this call. This puts MBD alongside other popular approaches to treat vdW dispersion interactions in DFT such as DFT-D4[12] or the vdW-DF family,[3] which are already represented by standalone programs[13] or libraries.[14]

libMBD[15] is written in modern Fortran with additional bindings available for C and Python. It uses MPI[16]/ScaLAPACK[17] parallelization, and implements MBD for both finite and periodic systems, with analytical gradients of the energy with respect to all input parameters, which enables both structure optimization as well as self-consistent vdW energy with respect to the electron density.[18] libMBD is designed as a framework for any method based on the MBD Hamiltonian, and has a built-in support for the original MBD formulation,[4] the range-separated self-consistently screened variant (MBD@rsSCS),[19] the metal-surface parametrizations (MBD$^{\text{surf}}$),[20,21] as well as the recent hybrid of MBD with nonlocal functionals (MBD-NL).[22] For convenience, an efficient implementation of the predecessor of MBD, the pairwise TS model,[23] is included as well. Finally, libMBD offers not only the vdW energy and its gradients, but can calculate also other quantities derived from the MBD Hamiltonian, namely the eigenstate charge fluctuations and induced charge-density polarization,[24] first-order Coulomb corrections,[25] the dielectric function,[26] and the order-by-order expansion of the energy in the dipole interaction.[27]

The popularity of MBD as a method is revealed through its independent implementations in numerous electronic structure codes, namely FHI-aims,[28,29] VASP,[30,31] Q-Chem,[32,33] Quantum ESPRESSO,[34,35] CASTEP,[36,37] and DFT-D4.[12,13] The original standalone reference implementation of MBD is available for download via Internet Archive.[38] While some of these implementations introduced methodological innovations—such as the one in VASP, which introduced a reciprocal-space formulation,[39] and the one in Quantum ESPRESSO, which introduced analytical gradients[40]—all of them lack both in computational efficiency and in capability compared to libMBD. The single exception is the linear-scaling stochastic evaluation of MBD,[41] which is more efficient for very large systems, but for the moment lacks analytical gradients. Furthermore, none of these MBD implementations have been designed as a library and their reuse across different electronic structure codes is problematic in theory, and nonexistent in practice. In contrast, libMBD requires only the quantities that directly depend on the electron density to be evaluated in the host program (since this implementation is usually code-specific), and has been already integrated into several such codes, including FHI-aims, VASP, Q-Chem, Quantum ESPRESSO, DFTB+,[42] and ASE.[43,44]

The flexibility of MBD as a general approach can be seen in the ever-growing list of methods and models that are based on it, and for this reason libMBD has been designed from the start as a framework that is easy to extend and modify. Besides the methods listed above that are already incorporated in libMBD, there are the fractionally ionic variant (MBD/FI)[45] and its successor "universal" MBD (uMBD),[46] the Wannier-function parameterization (vdW-WF2),[47] beyond-dipole extension,[48] anisotropic extension,[49] extension to excited states,[50] and the MBD variant of DFT-D4.[12] If implemented in or with libMBD, these methods and models or any future ones would become instantly available across all the electronic structure codes to which libMBD has already been or will be integrated, and could take advantage of parallelization, analytical gradients, and both finite and periodic formulations. Next to computational efficiency, this is the biggest offering of libMBD.

The rest of the paper is structured as follows: Sec. II reviews all the MBD methods implemented in libMBD in terms of equations, as well as the physics of integrating MBD with DFT and its substitutes. Section III documents the structure of the code and the public interface served to the electronic structure codes, its scaling properties, numerical convergence, functionality, and existing interfaces. Section IV describes a number of example MBD calculations and a meta-review of important past applications of MBD. The paper is concluded in Sec. V.

II. THEORY

This section reviews all mathematical equations that define most of the functionality of libMBD. It is structured such that equations in Secs. II A–II G are implemented in libMBD, while equations in Sec. II H must be implemented in the codes with which libMBD is integrated, since their implementation is inherently code-specific.

A. Many-body dispersion

Any MBD method maps an atomic system to a model Hamiltonian of charged harmonic (Drude) oscillators located at $\mathbf{R}_i$, with static polarizabilities $\alpha_{0,i}$ and frequencies ω_i. In most variants, the oscillators represent atoms, but both finer fragments, such as Wannier functions,[47] and coarser fragments, such as whole molecules,[51] are also possible. As a result of the coarse-graining, only the long-range part of the interaction between oscillators is expected to represent faithfully the original electronic system, justifying a dipole approximation. The MBD Hamiltonian for a finite system of N oscillators then reads

$$\begin{aligned}&H_{\text{MBD}}(\{\mathbf{R}_i,\alpha_{0,i},\omega_i,\boldsymbol{\kappa}_i\})\\&\quad:=\sum_i -\frac{1}{2}\nabla^2_{\xi_i}+\sum_i\frac{1}{2}\omega_i^2\xi_i^2+\frac{1}{2}\sum_{i\neq j}\omega_i\omega_j\sqrt{\alpha_{0,i}\alpha_{0,j}}\boldsymbol{\xi}_i\\&\qquad\cdot\mathbf{T}^{\text{lr}}(\boldsymbol{\kappa}_i,\boldsymbol{\kappa}_j,\mathbf{R}_j-\mathbf{R}_i)\boldsymbol{\xi}_j.\end{aligned}\tag{1}$$

Here, $\boldsymbol{\xi}_i=\sqrt{m_i}(\mathbf{r}_i-\mathbf{R}_i)$ are displacements of the oscillating charges weighted by the oscillator masses m_i, and $\mathbf{T}^{\text{lr}}$ is a damped dipole interaction tensor, parametrized through some single-oscillator properties $\boldsymbol{\kappa}_i$. The mass is in principle the third parameter (after static polarizability and frequency) required to fully specify a charged quantum oscillator, but it has no effect on the energy under the dipole approximation. Other sets of three independent oscillator parameters can be used as well, for example the C_6 dispersion coefficient or charge q, but these are always connected through simple relations, such as

$$4C_6 = 3\omega\alpha_0^2, \qquad q^2 = m\alpha_0\omega^2. \tag{2}$$

The damped dipole tensor must reduce to the "bare" dipole tensor $\mathbf{T}$ for large distances,

$$\lim_{r\to\infty}\mathbf{T}^{\mathrm{lr}}(\boldsymbol{\kappa},\boldsymbol{\kappa}',\mathbf{r}) = \mathbf{T}(\mathbf{r}), \tag{3}$$

$$T_{ab}(\mathbf{r}) = \frac{\partial^2}{\partial r_a \partial r_b}\frac{1}{|\mathbf{r}|} = \frac{-3r_a r_b + \delta_{ab}r^2}{r^5}. \tag{4}$$

Within this general MBD framework, a particular MBD model is then specified through the level of coarse-graining (typically atoms), the parametrization of the oscillators, and the choice of $\mathbf{T}^{\mathrm{lr}}$, which necessarily depends on the effective range (degree of non-locality) of the electronic structure method to which MBD is coupled.

A major consequence of the dipole approximation is that the MBD Hamiltonian can be exactly diagonalized, after which it describes a system of uncoupled collective oscillations $\tilde{\xi}_k = \sum_{ia} C_{ia,k}\xi_{ia}$. Here, we use a shorthand notation in which the particle and Cartesian indexes are joined, for example $(\boldsymbol{\xi}_i)_a = \xi_{ia} = (\boldsymbol{\xi})_{ia}$ and $(\mathbf{C}_{ij})_{ab} = C_{ia,jb} = (\mathbf{C})_{ia,jb}$. The MBD energy is then obtained as the change in the zero-point energy of the oscillations (fluctuations) induced by the dipole interaction,

$$\mathbf{Q}_{ij} = \delta_{ij}\omega_i^2\mathbf{I} + \omega_i\omega_j\sqrt{\alpha_{0,i}\alpha_{0,j}}\mathbf{T}_{ij}^{\mathrm{lr}} \tag{5}$$

$$\mathbf{Q} \equiv \mathbf{C}\boldsymbol{\Lambda}C^\dagger, \qquad \boldsymbol{\Lambda} \equiv \operatorname{diag}(\tilde{\omega}_1^2,\dots,\tilde{\omega}_{3N}^2) \tag{6}$$

$$E_{\mathrm{MBD}} = \sum_{k=1}^{3N}\frac{\tilde{\omega}_k}{2} - 3\sum_i\frac{\omega_i}{2} \tag{7}$$

with $\mathbf{T}_{ij}^{\mathrm{lr}} \equiv \mathbf{T}^{\mathrm{lr}}(\boldsymbol{\kappa}_i,\boldsymbol{\kappa}_j,\mathbf{R}_j - \mathbf{R}_i)$. Note that the diagonalization leads to complex-valued energy if some of the eigenvalues of $\mathbf{Q}$ are negative. This can result from a failure of the dipole approximation, which has two possible underlying causes in practice. First, the local polarizability model might be flawed and overestimating the local response, which was commonly occurring with the original parametrization based on Hirshfeld volumes for ionic compounds. This issue can be resolved by using better polarizability models (Sec. II H). Second, the strong assumption about the locality of the charge fluctuations and the resulting aptness of the dipole approximation is inappropriate in metallic systems, in which the delocalized charge fluctuations of the conducting electrons cannot be adequately described by a local oscillator model. Here, only an effective treatment applicable to hybrid organic–inorganic interfaces is possible,[21,22] although a recent extension of MBD suggests that a more general solution might be viable.[52]

B. Periodic boundary conditions

In a periodic system, the collective oscillations have an associated wave vector $\mathbf{q}$ from the first Brillouin zone (FBZ) as a good quantum number, and the energy per periodic unit cell can be calculated as an integral over the FBZ, evaluated in practice as a sum over a $\mathbf{q}$-point mesh,

$$E_{\mathrm{MBD}} = \int_{\mathrm{FBZ}}\mathrm{d}\mathbf{q}\,E_{\mathrm{MBD}}(\mathbf{q}). \tag{8}$$

$E_{\mathrm{MBD}}(\mathbf{q})$ is obtained from Eqs. (5)–(7) by making $\tilde{\boldsymbol{\omega}}$, $\mathbf{T}^{\mathrm{lr}}$, and $\mathbf{C}$ formally dependent on $\mathbf{q}$ and considering

$$\mathbf{T}_{ij}(\mathbf{q}) = \sum_{\mathbf{n}}{}'\,\mathbf{T}(\mathbf{R}_{\mathbf{n}ij})\mathrm{e}^{-\mathrm{i}\mathbf{q}\cdot\mathbf{R}_{\mathbf{n}ij}}, \qquad \mathbf{R}_{\mathbf{n}ij} = \mathbf{R}_j + \mathbf{R}_{\mathbf{n}} - \mathbf{R}_i. \tag{9}$$

The infinite sum of the dipole tensor over all periodic copies of the unit cell, indexed by $\mathbf{n}$, and skipping the $\mathbf{n} = \mathbf{0}$, $i = j$ terms, is evaluated efficiently with Ewald summation,[53]

$$\begin{aligned}\mathbf{T}_{ij}(\mathbf{q}) \approx{}& \sum_{\mathbf{n}}^{0<R_{\mathbf{n}ij}<R_\mathrm{c}}\mathbf{T}^{\mathrm{erfc}}(\mathbf{R}_{\mathbf{n}ij},\gamma)\mathrm{e}^{-\mathrm{i}\mathbf{q}\cdot\mathbf{R}_{\mathbf{n}ij}}\\ &+\frac{4\pi}{\Omega}\sum_{\mathbf{m}}^{0<k_{\mathbf{m}}<k_\mathrm{c}}\hat{\mathbf{k}}_{\mathbf{m}}\otimes\hat{\mathbf{k}}_{\mathbf{m}}\,\mathrm{e}^{-\frac{k_{\mathbf{m}}^2}{4\gamma^2}-\mathrm{i}\mathbf{G}_{\mathbf{m}}\cdot\mathbf{R}_{ij}}\\ &-\delta_{ij}\frac{4\gamma^3}{3\sqrt{\pi}}\mathbf{I} + \delta(\mathbf{q})\frac{4\pi}{3\Omega}\mathbf{I}, \qquad \mathbf{k}_{\mathbf{m}} = \mathbf{G}_{\mathbf{m}} + \mathbf{q}.\end{aligned} \tag{10}$$

Here, $\mathbf{m}$ indexes the cells of the reciprocal lattice, $\mathbf{G}_{\mathbf{m}}$ is a reciprocal-lattice vector, R_c and k_c are the real-space and reciprocal-space cutoffs, respectively, γ is the Ewald range-separation parameter (see Sec. III C), Ω is the unit-cell volume, and $\mathbf{T}^{\mathrm{erfc}}$ is the short-range part of the bare dipole tensor,

$$\begin{aligned}T_{ab}^{\mathrm{erfc}}(\mathbf{r},\gamma) &= \frac{-3r_a r_b C(r,\gamma) + \delta_{ab}r^2 B(r,\gamma)}{r^5}\\ B(R,\gamma) &= \operatorname{erfc}(\gamma R) + \frac{2\gamma R}{\sqrt{\pi}}\mathrm{e}^{-(\gamma R)^2}\\ C(R,\gamma) &= 3\operatorname{erfc}(\gamma R) + \frac{2\gamma R}{\sqrt{\pi}}(3 + 2(\gamma R)^2)\mathrm{e}^{-(\gamma R)^2}.\end{aligned} \tag{11}$$

The last term in Eq. (10) is the so-called surface term,[54,55] which does not contribute to the energy (in practice the $\mathbf{q}$-point mesh avoids the Γ-point) or most other physical observables, and is stated here only for completeness.

C. Self-consistent screening

When put in an external electric field $\mathbf{E}^{\mathrm{ext}}$, a polarizable system partially (or fully) screens the field through induced polarization that is self-consistent with the total field. For charged harmonic oscillators interacting via some dipole interaction tensor $\mathbf{T}^{\mathrm{scs}}$, this reads

$$\mathbf{p}_i = -\alpha_i\mathbf{E}_i = -\alpha_i\left(\sum_j\mathbf{T}_{ij}^{\mathrm{scs}}\mathbf{p}_j + \mathbf{E}_i^{\mathrm{ext}}\right) \tag{12}$$

or equivalently in the shorthand notation

$$\mathbf{p} = -\boldsymbol{\alpha}(\mathbf{T}^{\mathrm{scs}}\mathbf{p} + \mathbf{E}^{\mathrm{ext}}) = -(\boldsymbol{\alpha}^{-1} + \mathbf{T}^{\mathrm{scs}})^{-1}\mathbf{E}^{\mathrm{ext}} \tag{13}$$

$$\bar{\boldsymbol{\alpha}} \equiv (\boldsymbol{\alpha}^{-1} + \mathbf{T}^{\mathrm{scs}})^{-1}. \tag{14}$$

Here, $\bar{\boldsymbol{\alpha}}_{ij}$ is a nonlocal anisotropic polarizability that describes the dipole on the i-th oscillator induced by an external field on the j-th oscillator.

In the MBD@rsSCS method, self-consistent screening is used to obtain a refined set of oscillator parameters before they are used in the MBD Hamiltonian. To do this, one can assume a homogeneous external electric field and contract $\bar{\boldsymbol{\alpha}}$ to effective local polarizability, and average out the angular dependence to obtain isotropic polarizability,

$$\bar{\alpha}_i = \frac{1}{3}\operatorname{Tr}\left[\sum\nolimits_j \bar{\boldsymbol{\alpha}}_{ij}\right]. \tag{15}$$

Effective screened oscillator frequencies $\bar{\omega}_i$ are obtained through C_6 coefficients

$$\bar{\omega} = \frac{4\bar{C}_6}{3\bar{\alpha}_0^2}, \qquad \bar{C}_6 = \frac{3}{\pi}\int_0^\infty \mathrm{d}u\,\bar{\alpha}(\mathrm{i}u)^2 \tag{16}$$

by screening the dynamic oscillator polarizability evaluated at imaginary frequency,

$$\alpha(\mathrm{i}u) = \frac{\alpha_0}{1 + u^2/\omega^2}, \qquad \alpha(0) \equiv \alpha_0. \tag{17}$$

In practice, the isotropic local polarizability in Eq. (17) is evaluated on a quadrature grid of imaginary frequencies, screened through Eq. (14) at each grid point, contracted via Eq. (15), and the integral in Eq. (16) is calculated numerically (see Sec. III C) to yield the C_6 coefficients and oscillator frequencies.

D. Analytical gradients

Analytical formulas for the gradients of the MBD energy with respect to all Hamiltonian input parameters can be obtained straightforwardly by application of the derivative chain rule. The two steps related to matrix diagonalization in Eq. (6) and matrix inversion and contraction in Eqs. (14) and (15), which are nontrivial, and for which naive application of the chain rule could result in inefficient evaluation of the gradients are explicitly stated below. When implemented, these formulas can be straightforwardly parallelized and result in a computational cost for the gradients that scales with the same power of system size as the energy evaluation.

The gradient of the MBD energy with respect to any Hamiltonian oscillator parameter X_i is

$$\frac{\partial E_{\mathrm{MBD}}}{\partial X_i} = \operatorname{Re}\frac{1}{2}\sum_{ajb}\left(\mathbf{C}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{C}^\dagger\right)_{ia,jb}\frac{\partial Q_{jb,ia}}{\partial X_i} - \frac{3}{2}\frac{\partial\omega_i}{\partial X_i}. \tag{18}$$

The gradient of the contracted SCS polarizability with respect to any oscillator parameter X_i is

$$\begin{aligned}\frac{\partial\bar{\alpha}_i}{\partial X_j} &= -\frac{1}{3}\sum_{ab}\left(B_{ia,jb}\sum\nolimits_k\bar{\alpha}_{ka,jb} + \bar{\alpha}_{jb,ia}\sum\nolimits_k B_{jb,ka}\right)\\ \mathbf{B} &= \bar{\boldsymbol{\alpha}}\mathbf{A}, \qquad A_{ia,jb} = \delta_{ij}\delta_{ab}\frac{\partial(\alpha_i^{-1})}{\partial X_i} + \frac{\partial T^{\mathrm{scs}}_{ia,jb}}{\partial X_i}.\end{aligned} \tag{19}$$

E. MBD properties

The normalized ground-state wave function of the molecular MBD Hamiltonian can be expressed as

$$\Psi(\boldsymbol{\xi}) = \left(\prod_{k=1}^{3N}\left(\frac{\tilde{\omega}_k}{\pi}\right)^{\frac{1}{4}}\right)\exp\left(-\frac{1}{2}\boldsymbol{\xi}\cdot\mathbf{C}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{C}^\dagger\boldsymbol{\xi}\right). \tag{20}$$

While only two parameters per oscillator, $\alpha_{0,i}$ and ω_i, are needed to get the MBD energy, a third parameter, such as m_i or q_i, is needed to evaluate the wave function in real space, in order to convert from $\boldsymbol{\xi}_i$ to $\mathbf{r}_i$. The wave function of the non-interacting oscillators is simply

$$\Psi_0(\boldsymbol{\xi}) = \left(\prod_{i=1}^{N}\left(\frac{\omega_i}{\pi}\right)^{\frac{3}{4}}\right)\exp\left(-\sum_i\frac{1}{2}\omega_i\xi_i^2\right). \tag{21}$$

The MBD wave function has been used to calculate two properties of interest—the polarization of the electron density due to vdW interactions,[24]

$$\begin{aligned}n_{\mathrm{pol}}(\mathbf{r}) &= \langle\Psi|\hat{n}|\Psi\rangle - \langle\Psi_0|\hat{n}|\Psi_0\rangle\\ \hat{n} &= \sum_i q_i\delta(\mathbf{r} - \mathbf{r}_i)\end{aligned} \tag{22}$$

and the first-order correction to the MBD energy from the Coulomb interaction,[25]

$$\begin{aligned}E^{(1)}_{\mathrm{Coul}} &= \langle\Psi|\hat{V}_{\mathrm{Coul}} - \hat{V}_{\mathrm{dip}}|\Psi\rangle - \langle\Psi_0|\hat{V}_{\mathrm{Coul}}|\Psi_0\rangle\\ \hat{V}_{\mathrm{Coul}} &= \frac{1}{2}\sum_{i\neq j}q_iq_j\left(\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} - \frac{2}{|\mathbf{r}_i - \mathbf{R}_j|} + \frac{1}{|\mathbf{R}_i - \mathbf{R}_j|}\right)\\ \hat{V}_{\mathrm{dip}} &= \frac{1}{2}\sum_{i\neq j}q_iq_j(\mathbf{r}_i - \mathbf{R}_i)\cdot\mathbf{T}_{ij}(\mathbf{r}_j - \mathbf{R}_j).\end{aligned} \tag{23}$$

All these expectation values can be evaluated analytically, and the derivations and final results can be found in Ref. 56

Instead of diagonalizing the MBD Hamiltonian, the MBD energy can be equivalently obtained as an integral over the imaginary frequency, which also allows for a many-body expansion of the energy in the orders of $\mathbf{T}^{\mathrm{lr}}$,[27]

$$E_{\mathrm{MBD}} = \frac{1}{2\pi}\int_0^\infty \mathrm{d}u\operatorname{Tr}\left[\ln\left(\mathbf{I} + \boldsymbol{\alpha}(\mathrm{i}u)\mathbf{T}^{\mathrm{lr}}\right)\right] \tag{24}$$

$$= -\frac{1}{2\pi}\int_0^\infty \mathrm{d}u\sum_{n=2}^\infty\frac{(-1)^n}{n}\operatorname{Tr}\left[\left(\boldsymbol{\alpha}(\mathrm{i}u)\mathbf{T}^{\mathrm{lr}}\right)^n\right]. \tag{25}$$

In practice, $\boldsymbol{\alpha}\mathbf{T}^{\mathrm{lr}}$ can be replaced with the Hermitian $\boldsymbol{\alpha}^{1/2}\mathbf{T}^{\mathrm{lr}}\boldsymbol{\alpha}^{1/2}$ since both are equivalent under the trace operator.

This integral formulation has two applications. First, one can analyze the contribution of the second-order (pairwise) and higher many-body orders to the vdW energy. Second, the eigenvalues x_k of $\boldsymbol{\alpha}(\mathrm{i}u)\mathbf{T}^{\mathrm{lr}}$ can be heuristically renormalized to obtain a real-valued MBD energy even when the exact diagonalization gives a complex-valued energy due to the dipole collapse.[45] In particular, one replaces

$$\begin{aligned}\ln(1 + x_k) &:= \ln(1 + \tilde{x}_k) - \tilde{x}_k\\ \tilde{x}_k &= \begin{cases} x_k & x_k \geq 0\\ -\operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x_k^4\right)^{1/4} & x_k < 0.\end{cases}\end{aligned} \tag{26}$$

Another property of interest is the macroscopic dielectric constant $\boldsymbol{\epsilon}_{\mathrm{M}}$ of a periodic system of charged oscillators, which can be calculated via SCS from $\bar{\boldsymbol{\alpha}}$ as

$$\boldsymbol{\epsilon}_{\mathrm{M}} = \lim_{\mathbf{q}\to 0}\left(1 + \frac{4\pi}{\Omega}\left(\sum_{ij}\bar{\boldsymbol{\alpha}}_{ij}(\mathbf{q})\right)\right)^{-1}. \tag{27}$$

F. Pairwise dispersion

The second (lowest) order of the many-body expansion in Eq. (25) corresponds to the familiar pairwise expression for the vdW energy,

$$\begin{gathered} E^{(2)}_{\mathrm{MBD}} = -\frac{1}{2}\sum_{i\neq j} C_{6,ij}\frac{f(\boldsymbol{\kappa}_i,\boldsymbol{\kappa}_j,R_{ij})}{R_{ij}^6} \\ C_{6,ij} = \frac{3}{2}\frac{\alpha_{0,i}\alpha_{0,j}}{\omega_i^{-1}+\omega_j^{-1}}, \qquad f(\boldsymbol{\kappa}_i,\boldsymbol{\kappa}_j,R_{ij}) = \frac{R_{ij}^6}{6}\operatorname{Tr}\left[\left(\mathbf{T}^{\mathrm{lr}}_{ij}\right)^2\right]. \end{gathered} \tag{28}$$

Here, f is a damping function ($f \to 1$ when $R_{ij}\to\infty$) and is typically specified directly rather than through $\mathbf{T}^{\mathrm{lr}}$. This *ansatz* has been used in numerous vdW models, among them the predecessor of MBD, the TS method.[23]

Under periodic boundary conditions, the corresponding lattice sum of Eq. (28) converges absolutely, unlike the dipole tensor, but the rate of convergence can be slow enough to present a computational bottleneck. An alternative is to use the Ewald technique,[57]

$$\begin{aligned} \sum_{\mathbf{n}}{}' \frac{1}{R_{\mathbf{n}ij}^6} &\approx \sum_{\mathbf{n}}^{0<R_{\mathbf{n}ij}<R_{\mathrm{c}}} \gamma^6\phi_r(\gamma R_{\mathbf{n}ij}) \\ &\quad + \frac{1}{\Omega}\sum_{\mathbf{m}}^{0<G_{\mathbf{m}}<k_{\mathrm{c}}} \gamma^3\phi_k\left(\frac{G_{\mathbf{m}}}{\gamma}\right)\cos\left(-\mathrm{i}\mathbf{G}_{\mathbf{m}}\cdot\mathbf{R}_{ij}\right) + \delta_{ij}\frac{\gamma^6}{6} \\ \phi_r(r) &= \left(\frac{1}{r^6}+\frac{1}{r^4}+\frac{1}{2r^2}\right)\mathrm{e}^{-r^2} \\ \phi_k(k) &= \frac{\pi^{\frac{3}{2}}}{12}\left((4-2k^2)\mathrm{e}^{-\frac{k^2}{4}} + k^3\sqrt{\pi}\,\mathrm{erfc}\left(\frac{1}{2}k\right)\right) \end{aligned} \tag{29}$$

with the same notation as in Eq. (10).

G. MBD methods

The general MBD framework discussed up to this point is concretized into an MBD method by specifying at least two aspects. First, how are the MBD oscillators parametrized (α_0, ω). Second, how is $\mathbf{T}^{\mathrm{lr}}$ chosen to smoothly integrate the long-range correlation from MBD with a given method that accounts for the short-range correlation, typically KS-DFT or its substitute. While independent to a certain degree, these two choices are ultimately related in practice.

1. TS

In the pairwise TS method,[23] each atom is mapped to a single oscillator parametrized through scaling of reference free-atom vdW parameters based on atomic volumes,

$$\begin{gathered} \alpha_{0,i} := \alpha^{\mathrm{ref}}_{0,i}\gamma_i, \qquad \gamma_i = \frac{V_i}{V_i^{\mathrm{ref}}} \\ \omega_i := \frac{4C^{\mathrm{ref}}_{6,ii}\gamma_i^2}{3\left(\alpha^{\mathrm{ref}}_{0,i}\gamma_i\right)^2} = \frac{4C^{\mathrm{ref}}_{6,ii}}{3\left(\alpha^{\mathrm{ref}}_{0,i}\right)^2} = \omega_i^{\mathrm{ref}} \\ R^{\mathrm{vdW}}_i := R^{\mathrm{vdW,ref}}_i\gamma_i^{\frac{1}{3}}. \end{gathered} \tag{30}$$

Here, V_i is some measure of a volume of an atom in a molecule or material and V_i^{ref} is the same measure for a corresponding free atom. Possible choices for the volume measure are discussed in Sec. II H.

The logistic sigmoid function parametrized with volume-rescaled vdW radii R^{vdW} is used as the damping function in Eq. (28),

$$\begin{gathered} f(R_i^{\mathrm{vdW}},R_j^{\mathrm{vdW}},R_{ij}) := \left(1+\mathrm{e}^{-a(\eta-1)}\right)^{-1} \\ \eta = \frac{R_{ij}}{s_R\left(R_i^{\mathrm{vdW}}+R_j^{\mathrm{vdW}}\right)}. \end{gathered} \tag{31}$$

The damping parameter s_R adjusts the onset of the vdW interaction and is optimized separately for each individual short-range correlation model, and $a = 20$.

2. MBD@rsSCS

The MBD@rsSCS method[19] is the many-body extension of the TS method, in which the oscillators interact through the dipole tensor derived from the full Coulomb interaction of two oscillator (Gaussian) charge densities with widths σ_i^2,

$$\begin{aligned} T^{\mathrm{GG}}_{ab}(\mathbf{r},\sigma) &= \frac{\partial^2}{\partial r_a\partial r_b}\frac{\mathrm{erf}(\zeta)}{r} \\ &= \left(\mathrm{erf}(\zeta)-\Theta(\zeta)\right)T_{ab}(\mathbf{r}) + 2\zeta^2\Theta(\zeta)\frac{r_ar_b}{r^5} \\ \Theta(\zeta) &= \frac{2\zeta}{\sqrt{\pi}}\mathrm{e}^{-\zeta^2}, \qquad \zeta = \frac{r}{\sqrt{\sigma_i^2+\sigma_j^2}}. \end{aligned} \tag{32}$$

The oscillator width is related to its polarizability through certain self-consistence requirements,[58]

$$\sigma_i = \left(\frac{1}{3}\sqrt{\frac{2}{\pi}}\alpha_i\right)^{\frac{1}{3}}. \tag{33}$$

To integrate this full-range interaction with KS-DFT while avoiding double-counting of the short-range correlation, $\mathbf{T}^{\mathrm{GG}}$ is range-separated, and the short-range part is used in SCS to refine the TS oscillator parameters from Eq. (30), while the long-range part is used in the MBD Hamiltonian,

$$\mathbf{T}^{\mathrm{scs}} := \left(1-g(R_i^{\mathrm{vdW}},R_j^{\mathrm{vdW}},R_{ij})\right)\mathbf{T}^{\mathrm{GG}}(\alpha_i,\alpha_j,\mathbf{R}_{ij}) \tag{34}$$

$$\begin{aligned} \mathbf{T}^{\mathrm{lr}} &:= g(\bar{R}_i^{\mathrm{vdW}},\bar{R}_j^{\mathrm{vdW}},R_{ij})\mathbf{T}^{\mathrm{GG}}(\alpha_i,\alpha_j,\mathbf{R}_{ij}) \\ &\approx g(\bar{R}_i^{\mathrm{vdW}},\bar{R}_j^{\mathrm{vdW}},R_{ij})\mathbf{T}(\mathbf{R}_{ij}) \end{aligned} \tag{35}$$

$$\bar{R}_{\mathrm{vdW}} = R_{\mathrm{vdW}}\left(\frac{\bar{\alpha}_0}{\alpha_0}\right)^{\frac{1}{3}}. \tag{36}$$

Here, g is a damping function of the same form as f in Eq. (31), but with $a = 6$ and s_R usually denoted as β.

3. MBD-NL

The MBD-NL method[22] removes the need for SCS, and the MBD Hamiltonian is parametrized directly with

$$\alpha_{0,i} := \alpha_{0,i}^{\mathrm{ref}} \frac{\alpha_{0,i}^{\mathrm{df}}}{\alpha_{0,i}^{\mathrm{df,ref}}}, \qquad \omega_i := \omega_i^{\mathrm{ref}} \frac{\omega_i^{\mathrm{df}}}{\omega_i^{\mathrm{df,ref}}}. \tag{37}$$

Here, a polarizability functional of the density is used to obtain vdW parameters (α_0^{df}, ω^{df}) for an atom in a molecule or material and the corresponding free atom (see Sec. II H). MBD-NL uses the same parametrization of $\mathbf{T}^{\mathrm{lr}}$ in Eq. (35) as MBD@rsSCS, but with vdW radii in Eq. (31) replaced with

$$R_i^{\mathrm{vdW}} := \frac{5}{2}(\alpha_{0,i}^{\mathrm{ref}})^{\frac{1}{7}} \left(\frac{\alpha_{0,i}^{\mathrm{df}}}{\alpha_{0,i}^{\mathrm{df,ref}}} \right)^{\frac{1}{3}}. \tag{38}$$

H. Integration with short-range models

MBD models the long-range part of electron correlation and must be always coupled with a model that captures the rest of the electronic energy, including the short-range correlation,

$$E = E_{\mathrm{sr}} + E_{\mathrm{MBD}}. \tag{39}$$

Typically, this short-range model is some semilocal or hybrid version of KS-DFT, but it can also be its computationally more efficient substitute such as DFTB or a machine-learned interatomic potential. In any case, the short-range model and MBD must be integrated in two ways. First, the damping parameter in an MBD method (Sec. II G) must be adjusted to avoid double counting of correlation in the intermediate range. This is typically done by fitting on the S66x8 dataset of vdW dimers of small organic molecules.[59] Second, the short-range model is used to parametrize the MBD Hamiltonian, that is, to obtain the vdW parameters of each oscillator. How this is done depends on a particular model and the various options are described below. In the context of libMBD, the following equations must be necessarily evaluated in the code that implements the short-range model and couples to libMBD.

1. Hirshfeld volumes

This parametrization uses Eq. (30), in which the atomic volumes are calculated from the Hirshfeld-partitioned electron density obtained from a DFT calculation,

$$\begin{gathered} V_i := V[n_i], \qquad V_i^{\mathrm{ref}} := V[n_i^{\mathrm{ref}}] \\ V[n] = \int \mathrm{d}\mathbf{r}\, n(\mathbf{r}) r^3, \qquad n_i(\mathbf{r}) = n(\mathbf{R}_i + \mathbf{r}) w_i^{\mathrm{H}}(\mathbf{r}) \\ w_i^{\mathrm{H}}(\mathbf{r}) = \frac{n_i^{\mathrm{ref}}(\mathbf{r})}{\sum_j n_j^{\mathrm{ref}}(\mathbf{r} + \mathbf{R}_i - \mathbf{R}_j)}. \end{gathered} \tag{40}$$

Here n_i^{ref} are electron densities of free atoms located at $\mathbf{r} = 0$. A straightforward extension is to use not only neutral free atoms, but also free ions[60] for the density partitioning, which improves description of ionic compounds. This was further extended to use not only ion densities, but also ionic reference vdW parameters by using a piecewise linear dependence of atomic polarizabilities on the charge.[45] This results in atomic $\alpha_i(\mathrm{i}u)$ that cannot be expressed through a single oscillator with Eq. (17), requiring either direct use of Eq. (24) or calculation of effective oscillator frequencies via Eq. (16).

2. MBD-NL

Electron density from a KS-DFT calculation is also used in MBD-NL to parametrize the MBD Hamiltonian. Here, a renormalized VV10 polarizability functional of the density[61] is coarse-grained via Hirshfeld partitioning,

$$\alpha_i^{\mathrm{df}}(\mathrm{i}u) := \int \mathrm{d}\mathbf{r} \frac{g(I,\chi) n(\mathbf{r}) w_i^{\mathrm{H}}(\mathbf{r})}{\frac{4\pi}{3} n(\mathbf{r}) + C \frac{|\nabla n(\mathbf{r})|^4}{n(\mathbf{r})^4} + u^2} \tag{41}$$

and the resulting atomic dynamic polarizabilities are reduced to effective oscillator frequencies (ω_i^{df}) via Eq. (16). The free-atom reference values ($\alpha_{0,i}^{\mathrm{df,ref}}$, $\omega_i^{\mathrm{df,ref}}$) are obtained by evaluating the functional on free-atom densities. The renormalized VV10 functional uses a semilocal cutoff function g,[22] [Eq. (7)] which is a function of the local ionization potential I and the iso-orbital indicator χ, which in turn are functions of the electron density, its gradient, and the kinetic energy density, quantities readily available in any code that implements meta-GGA functionals.

3. Charge population analysis

In the absence of a real-space representation of the electron density, the above-mentioned methods are not applicable. Many electronic structure methods exist where the electron density is not at all or only rarely evaluated explicitly. This includes approximate methods such as DFTB, as well as other density-matrix-based approaches. Stohr *et al.*[7] found that a correlation between effective atomic polarizability of an atom in a molecule can be established based not only on the effective volume of an atom, but also on measures derived from the density matrix represented in a local atomic orbital basis set, $|\psi_a\rangle = \sum_\nu c_\nu^a |\phi_\nu\rangle$, where single-electron states $|\psi_a\rangle$ are expanded in atomic orbital basis functions $|\phi_\nu\rangle$. When scaling free-atom vdW parameters [Eq. (30)] with the degree of hybridization between atoms as measured by the sum over the on-site component of the Mulliken charges (or equivalently the atom-projected trace over the density matrix) and the atomic charge Z_i,

$$\gamma_i := \frac{h_i}{Z_i} = \frac{\sum_a f_a \sum_{\nu \in i} |c_\nu^a|^2}{Z_i} \tag{42}$$

static atomic polarizabilities can be predicted that are similar to the ones obtained by rescaling with Hirshfeld volumes [Eq. (40)].

4. Machine-learned interatomic potentials

Several works have shown that the effective atoms-in-molecules volume scaling ratios as a function of the atomic positions can be accurately represented by machine-learning regression.[8–10] Once a continuous representation of the scaling ratio is constructed, an MBD description of the long-range correlation energy can be straightforwardly coupled to short-range force fields or atomistic machine-learned potentials. In cases where these potentials are directly trained on semilocal approximations to DFT, the choice of the range-separation parameter remains identical to the optimal choice for the underlying density-functional approximation.

III. IMPLEMENTATION

A. Code structure and API

libMBD implements all equations in Secs. II A–II G in Fortran 2008. To enable straightforward and rapid development of new variants and extensions of MBD, libMBD has been designed from the start to maintain a close correspondence between code and physics (Fig. 1). Equations are documented in source code with LATEX right next to the individual procedures that implement them and rendered in auto-generated documentation (https://libmbd.github.io). Gradients of the energy are computed in a forward mode, where each individual function that computes a value from its arguments also optionally returns the partial derivatives of the output value with respect to some (or all) of its input arguments. Parallelization is available either by trivial distribution of the **q**-point mesh [Eq. (8)] or by distribution of atom-indexed matrices (**T**, **Q**, $\tilde{\boldsymbol{\alpha}}$) via BLACS.[62] Linear-algebra operations are handled via LAPACK,[63] ScaLAPACK,[17] or ELPA[64,65] interfaced through ELSI.[66,67] libMBD can be used natively from Fortran, as a plain binary library via C API made by the iso_c_binding Fortran module, or from Python via a C extension that internally binds to the C API. The Fortran API (Listing 1) is based on two derived types, mbd_input_t and mbd_calc_t, where the former is a plain data structure of all input parameters used to initialize the latter, which is an opaque handle that provides access to computation and results. The Python API (Listing 2) is based on the MBDGeom class that represents a given geometry and provides access to computation via its methods.

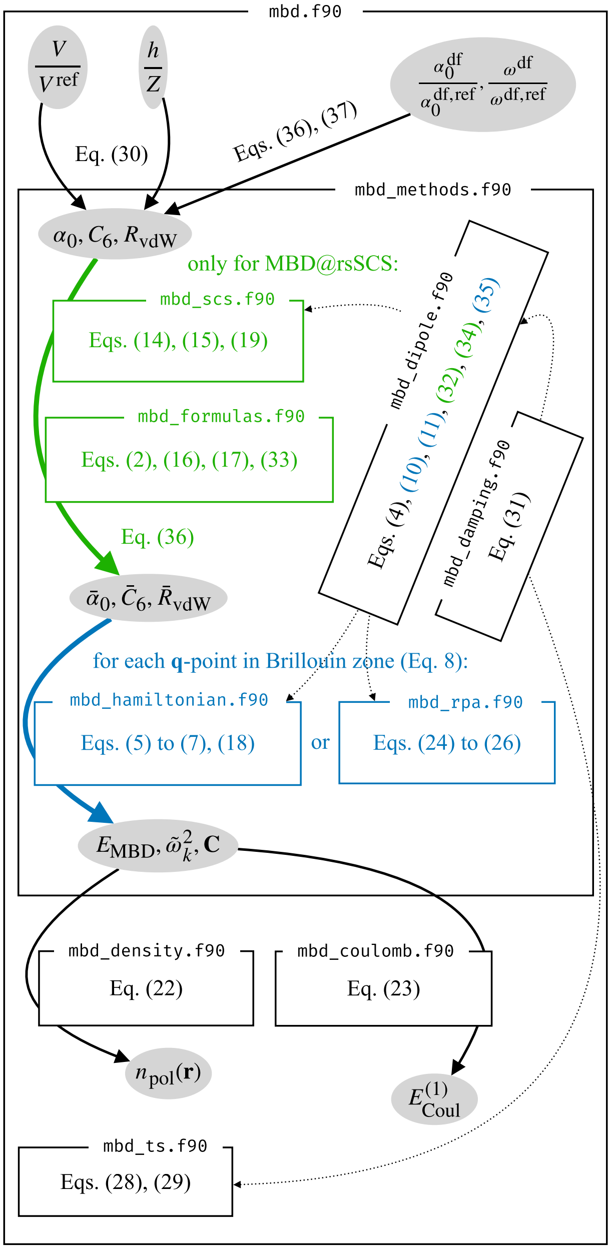
mbd.f90
$\frac{V}{V^{\text{ref}}}$
$\frac{h}{Z}$
Eq. (30)
Eqs. (36), (37)
$\frac{\alpha_0^{\text{df}}}{\alpha_0^{\text{df,ref}}}, \frac{\omega^{\text{df}}}{\omega^{\text{df,ref}}}$
mbd_methods.f90
$\alpha_0, C_6, R_{\text{vdW}}$
only for MBD@rsSCS:
mbd_scs.f90
Eqs. (14), (15), (19)
mbd_formulas.f90
Eqs. (2), (16), (17), (33)
Eq. (36)
$\bar{\alpha}_0, \bar{C}_6, \bar{R}_{\text{vdW}}$
mbd_dipole.f90
Eqs. (4), (10), (11), (32), (34), (35)
mbd_damping.f90
Eq. (31)
for each **q**-point in Brillouin zone (Eq. 8):
mbd_hamiltonian.f90
Eqs. (5) to (7), (18)
or
mbd_rpa.f90
Eqs. (24) to (26)
$E_{\text{MBD}}, \tilde{\omega}_k^2, \mathbf{C}$
mbd_density.f90
Eq. (22)
mbd_coulomb.f90
Eq. (23)
$n_{\text{pol}}(\mathbf{r})$
$E^{(1)}_{\text{Coul}}$
mbd_ts.f90
Eqs. (28), (29)

FIG. 1. Mapping code to physics. libMBD was designed for ease of development, which is achieved through tight correspondence between code and physics. An overview of an MBD@rsSCS calculation is shown on the background of a map of MBD source files with the individual implemented equations.

Listing 1. libMBD's Fortran API minimal example.

```
use mbd, only: mbd_input_t, mbd_calc_t
use iso_fortran_env, only: real64

type(mbd_input_t) :: inp
type(mbd_calc_t) :: calc
real(real64) :: energy, gradients(3, 2)
integer :: code
character(200) :: origin, msg

inp%atom_types = [’Ar’, ’Ar’]
inp%coords = reshape([0d0, 0d0, 0d0, 0d0, 0d0, 7.5d0], [3, 2])
inp%method = ’mbd-rsscs’
inp%xc = ’pbe’
call calc%init(inp)
call calc%get_exception(code, origin, msg)
if (code > 0) then
    print *, msg
    stop 1
end if
call calc%update_vdw_params_from_ratios([0.98d0, 0.98d0])
call calc%evaluate_vdw_method(energy)
call calc%get_gradients(gradients)
call calc%destroy()
```

Listing 2. libMBD's Python API minimal example.

```
from pymbd.fortran import MBDGeom

geom = MBDGeom([(0, 0, 0), (0, 0, 7.5)])
energy = geom.mbd_energy_species(
    [’Ar’, ’Ar’], [0.98, 0.98], beta=0.83
)
```

80% of the code base of libMBD is covered by tests, which include both unit tests of analytical gradients of individual procedures as well as regression tests of the total energies and gradients. libMBD is built with CMake.[68]

B. Scalability

For libMBD to be practical, it needs to keep up with the codes that implement the corresponding short-range models (DFT and its substitutes) in terms of scalability to large systems, and scalability to a large number of processors. libMBD uses only process parallelization through the MPI and BLACS, and no thread parallelization.

Tested on various supercells of crystalline urethane (unit cell with 26 atoms), with or without periodic boundary conditions, the MBD method and its implementation in libMBD scale with the number of atoms, N, as follows (Fig. 2). For small N, the computational cost is dominated by the construction of the various matrices and as such grows quadratically with N. At around a few hundred atoms (finite system) or a few thousand atoms (periodic system), the matrix operations that scale cubically with N (multiplication, diagonalization, inversion) start to dominate. The MBD analytical gradients as implemented in Ref. 40 add one extra factor of N to the scaling (making it quartic) due to explicit evaluation of the gradient of $\tilde{\boldsymbol{\alpha}}$ [see Eqs. (44) and (45) therein]. libMBD avoids this by calculating only the gradient of the fused evaluation of $\tilde{\boldsymbol{\alpha}}$ and its contraction to atoms [Eq. (19)], and as a result calculation of the gradients changes only the prefactor in the scaling behavior.

Tested on a $5 \times 5 \times 5$ supercell of urethane (3250 atoms), libMBD exhibits good strong scaling with the number of processor cores n (Fig. 3). The construction of the various matrices ($\mathbf{T}$, $\mathbf{Q}$, $\tilde{\boldsymbol{\alpha}}$) is trivially parallelizable and has perfect strong scaling up to $n = 4096$. The evaluation of the MBD gradients requires only very little inter-process communication and shows similarly good behavior. The distributed matrix inversion (SCS) and diagonalization (MBD) as provided by ScaLAPACK and ELPA exhibits strong scaling up to $n = 256$ on our particular setup, at which point the scaling plateaus. The evaluation of SCS gradients requires only matrix multiplications and contractions, but its current simple implementation in libMBD requires an unnecessary amount of inter-process communication, making the strong scaling plateau already at $n = 64$ for finite systems. Note that this slight inefficiency still allows the calculation of the MBD energy and gradients in ~100 s at $n = 256$, and furthermore it affects only the MBD@rsSCS method, not the newer MBD-NL method which does not use SCS. Nevertheless if better strong scaling is required in the future, the inter-process communication in the evaluation of the SCS gradients can be massively optimized.

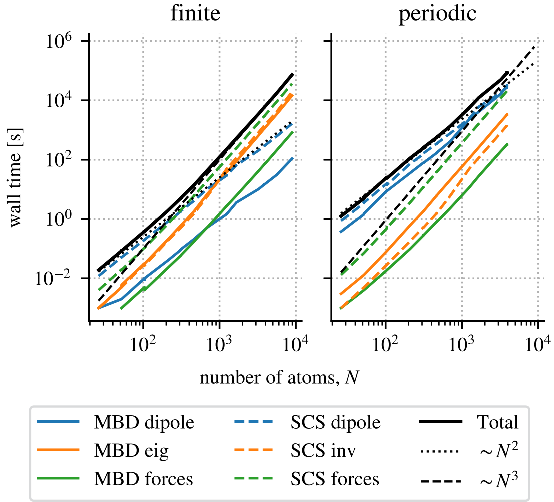

FIG. 2. System-size scaling. The scaling of an MBD calculation with system size transitions from quadratic for small systems where the matrix construction dominates to cubic for large systems where matrix operations (multiplication, diagonalization, inversion) dominate.

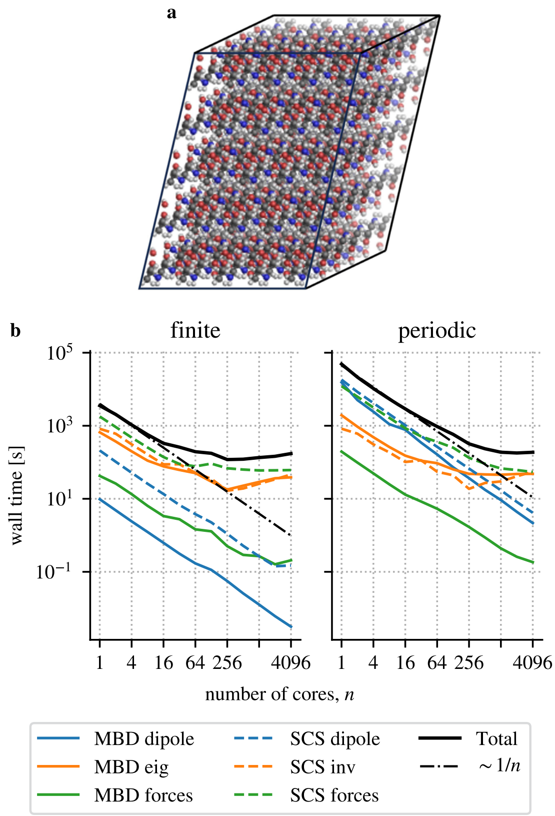

FIG. 3. Strong scaling. (a) 5 × 5 × 5 urethane supercell used for benchmarking. (b) libMBD parallelizes well to hundreds of CPU cores, and is mostly limited only by the parallelization of the matrix operations in external libraries. The current simple implementation of the SCS forces uses an unnecessary amount of inter-process communication, resulting in a relatively early plateau for finite systems. This will be addressed in future work.

C. Convergence and default parameters

There are a few critical parameters that determine the numerical accuracy of the calculated MBD energy with respect to the exact

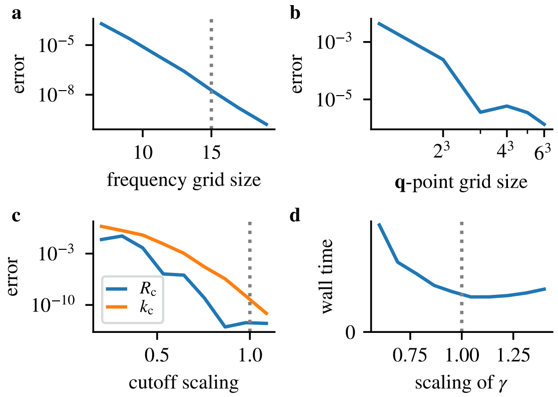

FIG. 4. Numerical convergence. Where possible, libMBD offers sane defaults for parameters that ensure numerical convergence. All errors are relative. Dotted vertical line denotes the default value of a parameter. (a) The MBD energy converges exponentially with respect to the frequency grid [Eq. (16)]. (b) Polynomial convergence with respect to the size of the **q**-point grid [Eq. (8)]. (c) Exponential convergence with respect to the real- and reciprocal-space cutoffs, R_c, k_c, in the Ewald summation [Eq. (10)]. (d) Optimal range separation in Ewald summation to minimize computational cost.

energy as defined in Secs. II A–II G. The SCS is evaluated and integrated on an imaginary-frequency grid [Eq. (16)], for which libMBD uses a Gauss–Legendre quadrature transformed from $[-1, 1]$ to $[0, \infty]$ with $x \to 0.6 \cdot (1 + x)/(1 - x)$. The MBD energy converges exponentially with the grid size, and using the default 15 grid points, it is converged to relative error of 10^{-8} [Fig. 4(a)]. The frequency grid size is the only parameter for calculations on finite systems. For periodic systems, there are additional parameters related to converging the infinite lattice sums involved.

The MBD energy per unit cell of a periodic system is calculated by integration over the FBZ [Eq. (8)]. libMBD uses a uniform quadrature grid in the reciprocal space, offset from the origin since the MBD energy has a removable singularity at the Γ-point (see Sec. II B). The convergence of the MBD energy with the **q**-point grid size is polynomial and system-dependent [see Fig. 4(b) for example convergence on urethane], and as such the grid needs to be specified by the user.

The Ewald summation of the dipole interactions between the origin cell and the infinite crystal [Eq. (10)] is controlled by three interconnected parameters—the range-separation parameter γ, and the real- and reciprocal-space cutoffs R_c, k_c. libMBD uses the following default prescription for the parameters,

$$R_c := \frac{6}{\gamma}, \quad k_c := 10\gamma, \quad \gamma := \frac{2.5}{\sqrt[3]{\Omega}}, \tag{43}$$

where Ω is the unit-cell volume. The MBD energy converges exponentially with the cutoffs and the default values ensure convergence to relative error of 10^{-10} [Fig. 4(c)]. The range-separation parameter does not affect accuracy, but through its determination of the cutoffs affects computational cost. The default value in libMBD ensures an optimal balance between the costs of the real- and reciprocal-space parts [Fig. 4(d)].

TABLE I. Available features in libMBD.

	Python bindings	Periodic	Parallel	Gradients[a]
MBD energy	✓	✓	✓	✓
SCS	✓	✓	✓	✓
TS energy	✓	✓	✓	✓
MBD energy via $\int du$	✓	✓	✓	
MBD eigenstates	✓	✓	✓	
MBD pol. density	✓			
MBD Coulomb corr.	✓			

[a] Includes forces, stress tensor, and energy gradients with respect to the vdW parameters (for self-consistence).

D. Functionality

libMBD implements all equations in Secs. II A–II G, with all physical quantities accessible via the Python bindings (Table I). Namely, this includes the calculation of MBD and TS energies, evaluation of SCS, evaluation of the MBD energy via the imaginary-frequency integration [Eq. (24)], access to MBD eigenstates, and evaluation of the MBD polarization density and Coulomb correction. All energy evaluations are implemented for both finite and periodic systems, parallelized, and with available analytical gradients (except for the imaginary-frequency integration). Evaluation of MBD properties such as the Coulomb correction or density polarization is currently implemented only for finite systems and is not parallelized.

E. Existing interfaces

libMBD was originally developed alongside FHI-aims[28,29] and was tightly integrated into it. Due to specific build requirements of FHI-aims, libMBD is currently distributed with FHI-aims in a lightly pre-processed form. The current API of libMBD was then based on the preexisting framework for interfacing external libraries in DFTB+,[42,69] which was the first third-party program in which libMBD was integrated. As a result, the DFTB+ integration is very lightweight and can serve as a model for interfaces with other programs. libMBD has been since integrated into Quantum ESPRESSO,[34,35] VASP,[31] and Q-Chem[32,33] without having to change anything in the existing source code, demonstrating its universality.

While libMBD is typically integrated into a larger electronic structure program as part of a DFT or similar calculation, this is not the only use case. Especially when developing new methods or when investigating unexpected behavior of existing methods, it is practical to be able to evaluate MBD separately from DFT, and to be able to do so as flexibly as possible. For this reason, libMBD was from the start developed together with a Python interface called pyMBD. Beyond using pyMBD directly (Listing 2), it can also serve as a bridge between libMBD and other Python programs. One such existing example is the interface to ASE.[9,43]

IV. APPLICATIONS

To demonstrate the functionality of libMBD, here we report on novel calculations done on supramolecular complexes and review selected previous applications of MBD.

A. Interaction energies

We apply the MBD scheme via the libMBD interface within the FHI-aims[29] software package to benchmark the interaction energies of structures in the L7 dataset[70] against existing data in the literature. We compare the interaction energies of the L7 complexes as calculated with MBD against higher-level theories, in particular diffusion Monte Carlo (DMC)[71,72] data from Refs. 73 and 74 and local natural orbital coupled cluster with single, double, and perturbative triple excitations [LNO-CCSD(T)][75,76] data from Ref. 73. Both DMC and LNO-CCSD(T) have been shown to accurately predict intermolecular interactions of small organic molecules. We also compare MBD-calculated interaction energies against DFT-D3[77] data from Ref. 78 and DFT-D4 data from Refs. 12 and 79. Interaction energies were calculated as the difference between the total energy of the structure and the sum of the monomer energies.

DFT calculations were performed with the FHI-aims software package, using the PBE[80] and HSE06[81,82] density-functional approximations. The standard screening parameter of 0.11 $/a_0^{-1}$ was used for HSE06, and the numeric atomic orbitals were represented using a "tight" basis set.[29,83] The total energy, sum of eigenvalues, and charge density criteria were set to 1×10^{-6} eV, 1×10^{-2} eV, and 1×10^{-5} eV/a_0^{-3}, respectively. Input and output files for all DFT+MBD calculations are available as a dataset in the NOMAD electronic structure data repository.[84]

The L7 dataset[70] was chosen as it contains non-covalent complexes considerably larger than other datasets such as S22[85] or S66.[86,87] The L7 structures [Figs. 5(a)–5(g)] comprise intermolecular complexes between 48 and 112 atoms in size and are mostly dispersion-dominated, which makes them not only a good representation of biochemical structures but also provides a suitable test case to assess the accuracy of MBD for medium to large complexes involving π–π stacking, electrostatic interactions, and hydrogen bonding.[73] The L7 dataset contains a parallel-displaced π–π-stacked coronene dimer (C2C2PD), a π–π-stacked coronene· · ·adenine dimer (C3A), a π–π-stacked circumcoronene· · ·Watson–Crick hydrogen-bonded guanine-cytosine dimer (C3GC), an octodecane dimer in stacked parallel conformation (CBH), a π–π-stacked Watson–Crick hydrogen-bonded guanine-cytosine dimer (GCGC), a stacked guanine trimer (GGG), and a phenylalanine residues trimer in mixed hydrogen-bonded-stacked conformation (PHE). In addition, a larger system of a C_{60} buckyball inside a [6]-cycloparaphenyleneacetylene ring (C_{60}@[6]CPPA), symmetrized to the D_{3d} point group and comprising 132 atoms, was considered [Fig. 5(h)]. The C_{60}@[6]CPPA complex has a large polarizability[88] which gives rise to considerable dispersion interactions, and the confinement between the ring and the buckyball can result in non-trivial long-range repulsive interactions.[25,89]

DFT+MBD performs fairly well for the supramolecular complexes with respect to the higher-level LNO-CCSD(T) and DMC methods (Table II). For the majority of L7 complexes (C3A, C3GC, GCGC, GGG, and PHE), DFT+MBD is within chemical accuracy (±1 kcal mol^{-1}) of LNO-CCSD(T) and/or DMC. For complexes where the difference between dispersion-corrected DFT and higher-level interaction energies is greater than 1 kcal mol^{-1}, both PBE+MBD and HSE06+MBD generally outperform PBE+D3 with respect to the higher-level methods, as can be seen for both C2C2PD and CBH.

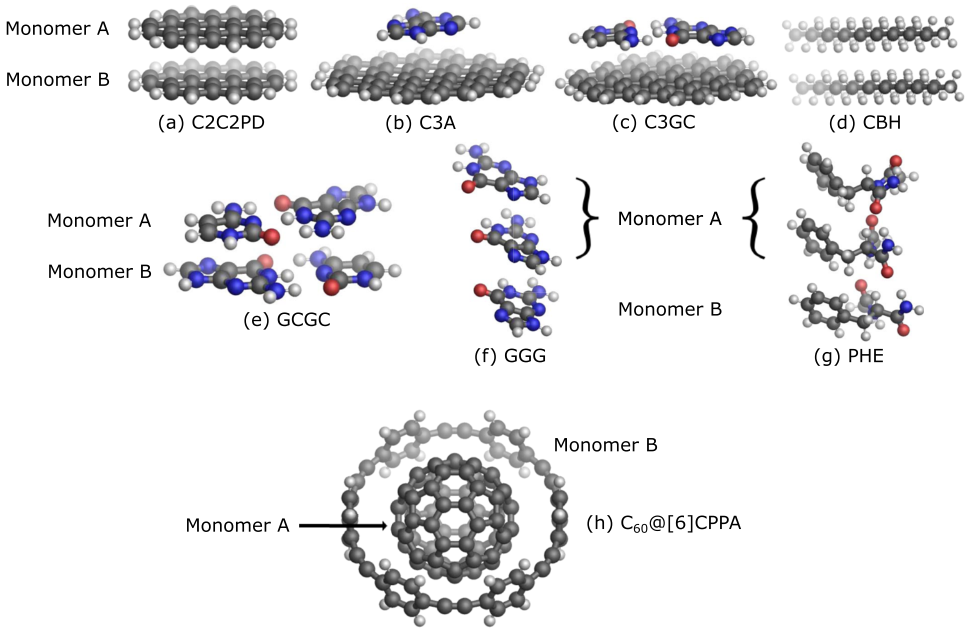

FIG. 5. Supramolecular complexes. All complexes were split into two interacting fragments, denoted as "Monomer A" and "Monomer B". H, C, N and O atoms are shown in white, gray, blue, and red, respectively. (a)–(g) L7 dataset. (h) C_{60}@[6]CPPA complex.

TABLE II. Interaction energies (kcal mol^{-1}) of supramolecular complexes with different methods.

Complex	LNO-CCSD(T)[a,b]	FN-DMC[a,c]	DMC[d]	PBE+D3[e]	PBE0+D4[a]	PBE+MBD	PBE0+MBD[a]	HSE06+MBD
C2C2PD	−20.6(6)	−18.1(8)	−17.5(7)	−15.78	−21.15	−16.57	−17.04	−17.34
C3A	−16.5(8)	−15.0(1.0)	−16.6(9)	−13.58	−16.46	−13.43	−13.86	−14.05
C3GC	−28.7(1.0)	−24.2(1.3)	−25.1(9)	−22.87	−27.60	−22.80	−23.48	−23.88
CBH	−11.0(2)	−11.4(8)	−10.9(8)	−14.16	−11.68	−13.95	−12.70	−12.95
GCGC	−13.6(4)	−12.4(8)	−10.6(6)	−11.75	−15.54	−11.74	−11.61	−11.78
GGG	−2.1(2)	−1.5(6)	−2.0(4)	−1.39	−2.10	−1.66	−1.27	−1.31
PHE	−25.4(2)	−26.5(1.3)	−24.9(0.6)	−25.00	−26.96	−26.56	−25.83	−27.65
C_{60}@[6]CPPA	−41.7(1.7)	−31.1(1.4)	⋯	⋯	−39.47	−29.59	−32.14	−33.89

[a] Data from Ref. 73.
[b] The indicated errors are extrapolated from the convergence of basis sets and local approximations.
[c] The indicated errors account for the stochastic uncertainty of the estimation and identify a 95% confidence interval (i.e. two standard deviations).
[d] Data from Ref. 74.
[e] Data from Ref. 78.

LNO-CCSD(T) and FN-DMC are in close agreement for five of the L7 structures (Table II). However, their interaction energies differ by 1.1 kcal mol^{-1} for C2C2PD, 2.2 kcal mol^{-1} for C3GC, and 7.6 kcal mol^{-1} for the C_{60}@[6]CPPA complex. Interestingly, PBE0+D4 is in close agreement with LNO-CCSD(T) across all complexes, including the three aforementioned structures, with a mean absolute deviation of 1.1 kcal mol^{-1}. In contrast, all the DFT+MBD approaches (PBE+MBD, PBE0+MBD, and HSE06+MBD) are much closer to FN-DMC, especially for the C_{60}@[6]CPPA complex. The differences between LNO-CCSD(T) and DFT+MBD, and between FN-DMC and DFT+D for the C2C2PD, C3GC and C_{60}@[6]CPPA complexes makes it difficult to evaluate a reliable reference interaction energy for these three complexes. This disparity should motivate the continued development of dispersion-correction methods in order to better characterize the interactions within dispersion-dominated complexes.

B. MBD properties

The MBD method is based on an effective Hamiltonian and as such provides access to more than just energies (Sec. II E). The coupled zero-point oscillations of the electron density and the density polarization have been used to reinterpret π–π stacking.[24,90] Both can be readily calculated with libMBD using the pyMBD interface.

The C3A dimer from the L7 dataset exhibits typical behavior of these quantities in a π–π complex. The low-energy zero-point MBD oscillations display long-range order spanning the whole complex, and corresponding to simple electrostatic patterns such as a dipole moment across a whole molecule interacting with an opposite dipole moment on the other molecule [Fig. 6(a)]. The vdW interaction in general polarizes the electron density out-of-plane in π–π complexes. The C3A dimer demonstrates that while the coupled oscillations are completely delocalized even in the case of differently sized interaction partners, the density polarization is mostly localized in the overlap region between the two molecules [Fig. 6(b)].

C. Previous applications of MBD

The MBD scheme has been used to identify chemical and thermodynamic trends across a broad range of materials and complexes, and has often succeeded even when other vdW-inclusive methods have failed.[91,92]

On many occasions, MBD has been able to accurately predict the thermodynamic stability of polymorphs often exhibited by molecular crystals, which are typically governed by non-covalent interactions.[91,92] The ability to correctly identify the energetic rankings of such vdW-bound systems has applications in numerous fields such as pharmacy and organic electronics.[91,93] One well-studied case

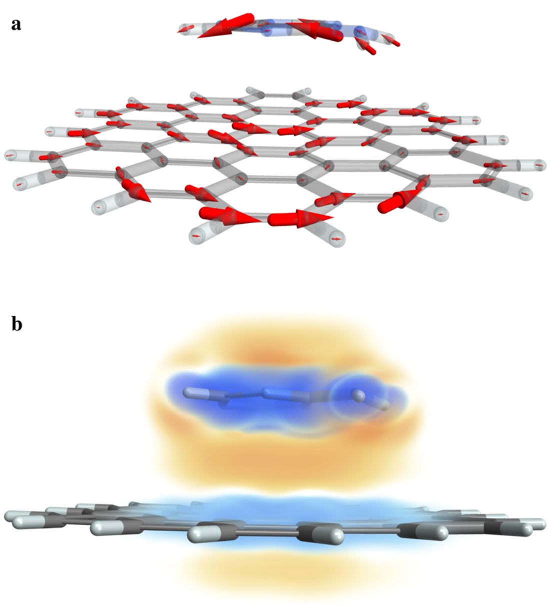

FIG. 6. MBD wavefunction. libMBD gives access not only to energies, but also to other expectation values of the MBD wavefunction. Here calculations on the π–π-stacked circumcoronene⋅⋅⋅adenine dimer (C3A) of the L7 dataset. (a) One of the low-energy zero-point oscillations of the MBD Hamiltonian [Eq. (6)], displayed as instantaneous dipole displacements of the electronic clouds around each atom. (b) Polarization of the electron density due to the vdW interaction [Eq. (22)], with density accumulation and depletion colored as orange and blue, respectively.

is oxalic acid,[91] for which many vdW-inclusive methods do not correctly predict the relative stability of its two polymorphs, whereas PBE+MBD agrees with experimental results.[94] Another example is coumarin, where the inclusion of beyond-pairwise interactions significantly improves the predicted energy ranking of crystalline polymorphs compared to the pairwise TS method.[95] Furthermore, the explicit accounting for many-body interactions elucidates the relative prevalence of a particular aspirin polymorph ("form 1") over the other in nature, whereas most other electronic structure methods (both with and without vdW corrections) predict both polymorphs to be energetically degenerate.[96]

MBD has also been shown to accurately calculate interaction energies for supramolecular host–guest complexes.[91] In particular, DMC calculations show that the binding energies of the C_{70}-fullerene to [10]- and [11]-cycloparaphenylene are degenerate (within DMC uncertainty).[24] Only the explicit inclusion of many-body interactions alongside DFT correctly predicts this degeneracy, whereas pairwise or two- and three-body vdW models show a clear preference for the 10-membered ring.[24,97]

Other applications that the MBD approach has been used for include layered materials[39] and molecular switches.[98] In the former case, the heat of adsorption of toluene on graphene, as calculated using PBE+MBD, was found to be in good agreement with experiment.[39] Furthermore, the balance between Pauli repulsion, chemical binding, electrostatic interactions, and MBD forces was observed to be critical for achieving molecular switches between the two bistable configurations of gold hexamers on single-walled carbon nanotubes, while the many-body effects were controlled by the former two interactions.[98]

Many-body dispersion effects have also been shown to be important for the description of hybrid organic–inorganic interfaces where MBD clearly outperforms additive pairwise dispersion schemes. For example, MBD was shown to improve the adsorption energetics and geometries of adsorbates such as xenon, graphene, 3,4,9,10-perylene-tetracarboxylic dianhydride (PTCDA), and 7,7,8,8-tetracyanoquinodimethane (TCNQ) on silver surfaces, as they reduce the overbinding typically found in pairwise additive dispersion-correction schemes.[99,100] Recently, PBE+MBD-NL provided an accurate prediction of the energetic ranking and the adsorption height of different adsorption phases of TCNQ on Ag(100) when compared to x-ray standing wave measurements.[101] MBD also permits changes in the anisotropic polarizability tensor, the description of adsorbate vibrations and adsorbate-surface interaction screening to be captured.[99] Furthermore, MBD interactions were shown to be a key factor behind the stabilization of a PTCDA molecule being brought to a standing configuration on Ag(111) surfaces using a scanning probe microscopy tip.[102] At a temperature of 5 K, the MBD-calculated lifetime of the standing configuration was calculated to be 1.7×10^5 s, which was consistent with experimental data; however, the vdW$^{\text{surf}}$-calculated lifetime was evaluated to be only 0.1 s.[102] In addition, the MBD-calculated energy barrier heights were around 170% larger than the vdW$^{\text{surf}}$-calculated barrier heights.[102] This is because the screening of long-range vdW interactions via non-additive many-body interactions reduces the molecule-surface attraction, which ultimately stabilizes the molecule in the upright geometry due to higher-order contributions to the vdW energy contained in MBD that counteract the pairwise contributions within vdW$^{\text{surf}}$.

V. CONCLUSIONS

libMBD is a mature, efficient, yet flexible library that offers all the functionality one might need to model vdW interactions in molecules and materials with MBD. It can be used to enhance any electronic structure code that can interface with Fortran, C, or Python external libraries with a powerful implementation of the MBD and TS methods, provided the said code can calculate the Hirshfeld volumes or evaluate the VV polarizability functional (for MBD-NL). With the recent advent of machine learning models for Hirshfeld volumes and atomic polarizabilities in general, it is even possible to calculate MBD energies with libMBD without running a single electronic structure calculation. libMBD is hosted and developed on GitHub under the Mozilla Public License 2.0, is distributed with the ESL Bundle,[103,104] and can be easily installed from conda-forge[105] and PyPI.[106] We hope that libMBD either standalone or embedded in a growing number of third-party programs will equip researchers with a new computational tool to study molecules and materials.

ACKNOWLEDGMENTS

We thank R. A. DiStasio Jr., A. Ambrosetti, and T. Markovich for countless discussions about MBD, V. Gobre for writing the original FHI-aims implementation of MBD, and F. Knoop for correcting mistakes in the equations in the early version of the manuscript. Funding is acknowledged from the German Ministry for Education and Research (Berlin Institute for the Foundations of Learning and Data, BIFOLD), the EPSRC Centre for Doctoral Training in Diamond Science and Technology (Grant No. EP/L015315/1), the Research Development Fund of the University of Warwick, the UKRI Future Leaders Fellowship programme (Grant No. MR/S016023/1), the European Research Council (ERC-AdG Grant No. FITMOL), and the Luxembourg National Research Fund (FNR-CORE Grant BroadApp No. C20/MS/14769845). We are grateful for computing resources from the Scientific Computing Research Technology Platform of the University of Warwick, the EPSRC-funded High-End Computing Materials Chemistry Consortium (Grant No. EP/R029431/1) for access to the ARCHER2 UK National Supercomputing Service (https://www.archer2.ac.uk), the EPSRC-funded UKCP Consortium (Grant No. EP/P022561/1), and the partially EPSRC-funded UK Materials and Molecular Modelling Hub (Grant Nos. EP/P020194 and EP/T022213) for access to Young.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Jan Hermann: Conceptualization (equal); Software (equal); Writing – original draft (equal). **Martin Stöhr**: Software (supporting); Writing – original draft (supporting). **Szabolcs Góger**: Software (supporting). **Shayantan Chaudhuri**: Data curation (supporting); Visualization (supporting); Writing – original draft

(supporting). **Bálint Aradi**: Conceptualization (supporting); Methodology (supporting). **Reinhard J. Maurer**: Conceptualization (equal); Funding acquisition (equal); Software (supporting); Supervision (equal); Writing – review & editing (equal). **Alexandre Tkatchenko**: Conceptualization (equal); Funding acquisition (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in NOMAD, at http://doi.org/10.17172/NOMAD/2023.06.16-2.

REFERENCES

[1] J. Hermann, R. A. DiStasio, Jr., and A. Tkatchenko, Chem. Rev. **117**, 4714 (2017).
[2] S. Grimme, A. Hansen, J. G. Brandenburg, and C. Bannwarth, Chem. Rev. **116**, 5105 (2016).
[3] K. Berland, V. R. Cooper, K. Lee, E. Schröder, T. Thonhauser, P. Hyldgaard, and B. I. Lundqvist, Rep. Prog. Phys. **78**, 066501 (2015).
[4] A. Tkatchenko, R. A. DiStasio, Jr., R. Car, and M. Scheffler, Phys. Rev. Lett. **108**, 236402 (2012).
[5] A. Ambrosetti, N. Ferri, R. A. DiStasio, Jr., and A. Tkatchenko, Science **351**, 1171 (2016).
[6] A. V. Akimov and O. V. Prezhdo, Chem. Rev. **115**, 5797 (2015).
[7] M. Stöhr, G. S. Michelitsch, J. C. Tully, K. Reuter, and R. J. Maurer, J. Chem. Phys. **144**, 151101 (2016).
[8] T. Bereau, R. A. DiStasio, A. Tkatchenko, and O. A. von Lilienfeld, J. Chem. Phys. **148**, 241706 (2018).
[9] J. Westermayr, S. Chaudhuri, A. Jeindl, O. T. Hofmann, and R. J. Maurer, Digital Discovery **1**, 463 (2022); arXiv:2202.13009.
[10] P. P. Poier, T. Jaffrelot Inizan, O. Adjoua, L. Lagardère, and J.-P. Piquemal, J. Phys. Chem. Lett. **13**, 4381 (2022); arXiv:2203.15739.
[11] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, Chem. Rev. **121**, 10142 (2021).
[12] E. Caldeweyher, S. Ehlert, A. Hansen, H. Neugebauer, S. Spicher, C. Bannwarth, and S. Grimme, J. Chem. Phys. **150**, 154122 (2019).
[13] See https://github.com/dftd4/dftd4 for more information about DFT-D4.
[14] A. H. Larsen, M. Kuisma, J. Löfgren, Y. Pouillon, P. Erhart, and P. Hyldgaard, Modell. Simul. Mater. Sci. Eng. **25**, 065004 (2017).
[15] See https://github.com/libmbd/libmbd for more information about libMBD.
[16] See https://www.mpi-forum.org for more information about MPI.
[17] See https://netlib.org/scalapack/ for more information about ScaLAPACK.
[18] N. Ferri, R. A. DiStasio, Jr., A. Ambrosetti, R. Car, and A. Tkatchenko, Phys. Rev. Lett. **114**, 176802 (2015).
[19] A. Ambrosetti, A. M. Reilly, R. A. DiStasio, Jr., and A. Tkatchenko, J. Chem. Phys. **140**, 18A508 (2014).
[20] V. G. Ruiz, W. Liu, E. Zojer, M. Scheffler, and A. Tkatchenko, Phys. Rev. Lett. **108**, 146103 (2012).
[21] V. G. Ruiz, W. Liu, and A. Tkatchenko, Phys. Rev. B **93**, 035118 (2016).
[22] J. Hermann and A. Tkatchenko, Phys. Rev. Lett. **124**, 146401 (2020).
[23] A. Tkatchenko and M. Scheffler, Phys. Rev. Lett. **102**, 073005 (2009).
[24] J. Hermann, D. Alfè, and A. Tkatchenko, Nat. Commun. **8**, 14052 (2017).
[25] M. Stöhr, M. Sadhukhan, Y. S. Al-Hamdani, J. Hermann, and A. Tkatchenko, Nat. Commun. **12**, 137 (2021).
[26] T.-T. Cui, J.-C. Li, W. Gao, J. Hermann, A. Tkatchenko, and Q. Jiang, J. Phys. Chem. Lett. **11**, 1521 (2020).
[27] A. Tkatchenko, A. Ambrosetti, and R. A. DiStasio, Jr., J. Chem. Phys. **138**, 074106 (2013).
[28] See https://fhi-aims.org for more information about FHI-aims.
[29] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Comput. Phys. Commun. **180**, 2175 (2009).
[30] See https://www.vasp.at for more information about VASP.
[31] G. Kresse and J. Furthmüller, Phys. Rev. B **54**, 11169 (1996).
[32] See https://www.q-chem.com for more information about Q-Chem.
[33] E. Epifanovsky, A. T. B. Gilbert, and X. Feng, J. Chem. Phys. **155**, 084801 (2021).
[34] See https://www.quantum-espresso.org for more information about Quantum ESPRESSO.
[35] P. Giannozzi, O. Baseggio, P. Bonfà, D. Brunato, R. Car, I. Carnimeo, C. Cavazzoni, S. de Gironcoli, P. Delugas, F. Ferrari Ruffino, A. Ferretti, N. Marzari, I. Timrov, A. Urru, and S. Baroni, J. Chem. Phys. **152**, 154105 (2020).
[36] See http://www.castep.org/ for more information about CASTEP.
[37] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. I. J. Probert, K. Refson, and M. C. Payne, Z. Kristallogr. - Cryst. Mater. **220**, 567 (2005).
[38] See https://web.archive.org/web/20220307232553/http://www.fhi-berlin.mpg.de/~tkatchen/MBD for more information about MBD.
[39] T. Bučko, S. Lebègue, T. Gould, and J. G. Ángyán, J. Phys.: Condens.Matter **28**, 045201 (2016).
[40] M. A. Blood-Forsythe, T. Markovich, R. A. DiStasio, Jr., R. Car, and A. Aspuru-Guzik, Chem. Sci. **7**, 1712 (2016).
[41] P. P. Poier, L. Lagardère, and J.-P. Piquemal, J. Chem. Theory Comput. **18**, 1633 (2022).
[42] B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshaye, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim, J. Chem. Phys. **152**, 124101 (2020).
[43] See https://wiki.fysik.dtu.dk/ase/ for more information about ASE.
[44] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, J. Phys.: Condens.Matter **29**, 273002 (2017).
[45] T. Gould, S. Lebègue, J. G. Ángyán, and T. Bučko, J. Chem. Theory Comput. **12**, 5920 (2016).
[46] M. Kim, W. J. Kim, T. Gould, E. K. Lee, S. Lebègue, and H. Kim, J. Am. Chem. Soc. **142**, 2346 (2020).
[47] P. L. Silvestrelli, J. Chem. Phys. **139**, 054106 (2013).
[48] D. Massa, A. Ambrosetti, and P. L. Silvestrelli, Electron. Struct. **3**, 044002 (2021).
[49] P. Bandyopadhyay, Priya, and M. Sadhukhan, Phys. Chem. Chem. Phys. **24**, 8508 (2022).
[50] A. Ambrosetti, P. Umari, P. L. Silvestrelli, J. Elliott, and A. Tkatchenko, Nat. Commun. **13**, 813 (2022).
[51] A. P. Jones, J. Crain, V. P. Sokhan, T. W. Whitfield, and G. J. Martyna, Phys. Rev. B **87**, 144103 (2013).
[52] J. F. Dobson and A. Ambrosetti, J. Chem. Theory Comput. **19**, 6434 (2023).
[53] S. W. de Leeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. London, Ser. A **373**, 57 (1980).
[54] S. W. de Leeuw, J. W. Perram, and E. R. Smith, Proc. R. Soc. London, Ser. A **373**, 27 (1980).
[55] V. Ballenegger, J. Chem. Phys. **140**, 161102 (2014).
[56] J. Hermann, "Towards unified density-functional model of van der Waals interactions," Ph.D. thesis, Humboldt University, Berlin, 2018; https://edoc.hu-berlin.de/handle/18452/19417.
[57] N. Karasawa and W. A. Goddard, J. Phys. Chem. **93**, 7320 (1989).
[58] A. Mayer, Phys. Rev. B **75**, 045407 (2007).
[59] B. Brauer, M. K. Kesharwani, S. Kozuch, and J. M. L. Martin, Phys. Chem. Chem. Phys. **18**, 20905 (2016).
[60] T. Bučko, S. Lebègue, J. G. Ángyán, and J. Hafner, J. Chem. Phys. **141**, 034114 (2014).
[61] O. A. Vydrov and T. Van Voorhis, J. Chem. Phys. **133**, 244103 (2010).
[62] See https://netlib.org/blacs/ for more information about BLACS.

[63] See https://netlib.org/lapack/ for more information about LAPACK.

[64] See https://elpa.rzg.mpg.de for more information about ELPA.

[65] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, J. Phys.: Condens. Matter **26**, 213201 (2014).

[66] See http://www.elsi-interchange.org/ for more information about ELSI.

[67] V. W.-z. Yu, C. Campos, W. Dawson, A. García, V. Havu, B. Hourahine, W. P. Huhn, M. Jacquelin, W. Jia, M. Keçeli, R. Laasner, Y. Li, L. Lin, J. Lu, J. Moussa, J. E. Roman, Á. Vázquez-Mayagoitia, C. Yang, and V. Blum, Comput. Phys. Commun. **256**, 107459 (2020).

[68] See https://cmake.org for more information about CMake.

[69] See https://dftbplus.org for more information about DFTB+.

[70] R. Sedlak, T. Janowski, M. Pitoňák, J. Řezáč, P. Pulay, and P. Hobza, J. Chem. Theory Comput. **9**, 3364 (2013).

[71] P. J. Reynolds, D. M. Ceperley, B. J. Alder, and W. A. Lester, J. Chem. Phys. **77**, 5593 (1982).

[72] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal, Rev. Mod. Phys. **73**, 33 (2001).

[73] Y. S. Al-Hamdani, P. R. Nagy, A. Zen, D. Barton, M. Kállay, J. G. Brandenburg, and A. Tkatchenko, Nat. Commun. **12**, 3927 (2021).

[74] A. Benali, H. Shin, and O. Heinonen, J. Chem. Phys. **153**, 194113 (2020).

[75] P. R. Nagy, G. Samu, and M. Kállay, J. Chem. Theory Comput. **14**, 4193 (2018).

[76] P. R. Nagy and M. Kállay, J. Chem. Theory Comput. **15**, 5275 (2019).

[77] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, J. Chem. Phys. **132**, 154104 (2010).

[78] Y. Xu, R. Friedman, W. Wu, and P. Su, J. Chem. Phys. **154**, 194106 (2021).

[79] E. Caldeweyher, C. Bannwarth, and S. Grimme, J. Chem. Phys. **147**, 034112 (2017).

[80] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[81] J. Heyd, G. E. Scuseria, and M. Ernzerhof, J. Chem. Phys. **118**, 8207 (2003).

[82] A. V. Krukau, O. A. Vydrov, A. F. Izmaylov, and G. E. Scuseria, J. Chem. Phys. **125**, 224106 (2006).

[83] V. Havu, V. Blum, P. Havu, and M. Scheffler, J. Comput. Phys. **228**, 8367 (2009).

[84] S. Chaudhuri, J. Hermann, and R. J. Maurer, DFT+MBD for L7, http://doi.org/10.17172/NOMAD/2023.06.16-2, 2023.

[85] P. Jurečka, J. Šponer, J. Černý, and P. Hobza, Phys. Chem. Chem. Phys. **8**, 1985 (2006).

[86] J. Řezáč, K. E. Riley, and P. Hobza, J. Chem. Theory Comput. **7**, 2427 (2011).

[87] J. Řezáč, K. E. Riley, and P. Hobza, J. Chem. Theory Comput. **10**, 1359 (2014).

[88] R. Antoine, Ph. Dugourd, D. Rayane, E. Benichou, M. Broyer, F. Chandezon, and C. Guet, J. Chem. Phys. **110**, 9771 (1999).

[89] M. Sadhukhan and A. Tkatchenko, Phys. Rev. Lett. **118**, 210402 (2017).

[90] K. Carter-Fenk and J. M. Herbert, Phys. Chem. Chem. Phys. **22**, 24870 (2020).

[91] M. Stöhr, T. Van Voorhis, and A. Tkatchenko, Chem. Soc. Rev. **48**, 4118 (2019).

[92] P. Xu, M. Alkan, and M. S. Gordon, Chem. Rev. **120**, 12343 (2020).

[93] D.-K. Bučar, R. W. Lancaster, and J. Bernstein, Angew. Chem., Int. Ed. **54**, 6972 (2015).

[94] A. M. Reilly and A. Tkatchenko, Chem. Sci. **6**, 3289 (2015).

[95] A. G. Shtukenberg, Q. Zhu, D. J. Carter, L. Vogt, J. Hoja, E. Schneider, H. Song, B. Pokroy, I. Polishchuk, A. Tkatchenko, A. R. Oganov, A. L. Rohl, M. E. Tuckerman, and B. Kahr, Chem. Sci. **8**, 4926 (2017).

[96] A. M. Reilly and A. Tkatchenko, Phys. Rev. Lett. **113**, 055701 (2014).

[97] J. Antony, R. Sure, and S. Grimme, Chem. Commun. **51**, 1764 (2015).

[98] Y. Chen, W. Gao, and Q. Jiang, J. Phys. Chem. C **123**, 9217 (2019).

[99] R. J. Maurer, V. G. Ruiz, and A. Tkatchenko, J. Chem. Phys. **143**, 102808 (2015).

[100] P. J. Blowey, B. Sohail, L. A. Rochford, T. Lafosse, D. A. Duncan, P. T. P. Ryan, D. A. Warr, T.-L. Lee, G. Costantini, R. J. Maurer, and D. P. Woodruff, ACS Nano **14**, 7475 (2020).

[101] B. Sohail, P. J. Blowey, L. A. Rochford, P. T. P. Ryan, D. A. Duncan, T.-L. Lee, P. Starrs, G. Costantini, D. P. Woodruff, and R. J. Maurer, J. Phys. Chem. C **127**, 2716 (2023).

[102] M. Knol, H. H. Arefi, D. Corken, J. Gardner, F. S. Tautz, R. J. Maurer, and C. Wagner, Sci. Adv. **7**, eabj9751 (2021).

[103] See https://esl.cecam.org/bundle/ for more information about ESL Bundle.

[104] M. J. T. Oliveira, N. Papior, Y. Pouillon, V. Blum, E. Artacho, D. Caliste, F. Corsetti, S. de Gironcoli, A. M. Elena, A. García, V. M. García-Suárez, L. Genovese, W. P. Huhn, G. Huhs, S. Kokott, E. Küçükbenli, A. H. Larsen, A. Lazzaro, I. V. Lebedeva, Y. Li, D. López-Durán, P. López-Tarifa, M. Lüders, M. A. L. Marques, J. Minar, S. Mohr, A. A. Mostofi, A. O'Cais, M. C. Payne, T. Ruh, D. G. A. Smith, J. M. Soler, D. A. Strubbe, N. Tancogne-Dejean, D. Tildesley, M. Torrent, and V. W.-z. Yu, J. Chem. Phys. **153**, 024117 (2020).

[105] See https://conda-forge.org for more information about Conda-forge.

[106] See https://pypi.org for more information about PyPI.