CrystaLattE: Automated computation of lattice energies of organic crystals exploiting the many-body expansion to achieve dual-level parallelism

Cite as: J. Chem. Phys. 151, 144103 (2019); doi: 10.1063/1.5120520
Submitted: 18 July 2019 • Accepted: 22 September 2019 •
Published Online: 8 October 2019

Carlos H. Borca,[1] Brandon W. Bakr,[1] Lori A. Burns,[1] and C. David Sherrill[1,2,a)]

AFFILIATIONS

[1] School of Chemistry and Biochemistry, Georgia Institute of Technology, Atlanta, Georgia 30332, USA
[2] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, USA

[a)] Electronic mail: sherrill@gatech.edu

ABSTRACT

We present an algorithm to compute the lattice energies of molecular crystals based on the many-body cluster expansion. The required computations on dimers, trimers, etc., within the crystal are independent of each other, leading to a naturally parallel approach. The algorithm exploits the long-range three-dimensional periodic order of crystals to automatically detect and avoid redundant or unnecessary computations. For this purpose, Coulomb-matrix descriptors from machine learning applications are found to be efficient in determining whether two N-mers are identical. The algorithm is implemented as an open-source Python program, CrystaLattE, that uses some of the features of the Quantum Chemistry Common Driver and Databases library. CrystaLattE is initially interfaced with the quantum chemistry package Psi4. With CrystaLattE, we have applied the fast, dispersion-corrected Hartree–Fock method HF-3c to the lattice energy of crystalline benzene. Including all 73 symmetry-unique dimers and 7130 symmetry-unique trimers that can be formed from molecules within a 15 Å cutoff from a central reference monomer, HF-3c plus an Axilrod-Teller-Muto estimate of three-body dispersion exhibits an error of only −1.0 kJ mol^{-1} vs the estimated 0 K experimental lattice energy of −55.3 ± 2.2 kJ mol^{-1}. The convergence of the HF-3c two- and three-body contributions to the lattice energy as a function of intermonomer distance is examined.

I. INTRODUCTION

The many-body expansion (MBE) is a widely applied fragmentation approach for intermolecular interactions.[1–23] It is based on the idea that the full interaction of an N-particle aggregate can be decomposed as an expansion of two-, three-, four-, . . ., N-body interactions. As a byproduct of using the MBE, the decomposition of the energy allows for meaningful physical insights about the forces that bind aggregates together.[24] Additionally, the nonadditive many-body energy contribution per molecule generally decreases as the number of molecules being considered increases. Therefore, the accuracy of the many-body expansion is typically incremental: the more terms counted in before truncation, the more accurate it becomes.

Organic crystals are good candidates to exploit the MBE approach. These solids are held together by noncovalent interactions that can be accurately described by fragmenting the system into smaller many-body aggregates (for small molecules, it is easier to use individual molecules as the bodies in the MBE, and sum over monomers, dimers, trimers, etc., of molecules). Additionally, in contrast to disordered systems like liquids, the three-dimensional order of crystalline structures introduces redundancies that can be exploited to reduce the number of computations required to obtain the ground-state properties of such systems.

There have been several successful efforts in computing the lattice energies of organic crystals using quantum chemical methods and the MBE in the last decade.[18,19,25–29] Highly accurate estimation of the crystal lattice energy of benzene, for example, has been achieved by our research group and others.[30–34] However, the lack of generally available software to automate the process to compute lattice energies has limited the application of this approach. Also, if not automated, the setup process to execute these calculations is tedious and error-prone.[35]

In this article, we present an algorithm and associated software, CrystaLattE, to automate the calculation of lattice energies of molecular crystals via the MBE. CrystaLattE was designed to return a highly accurate lattice energy, given a crystalline structure; thus, it requires only a Crystallographic Information File (CIF) as input.[36,37] The motivation stems from the current challenges in Crystal Structure Prediction (CSP). With the recent advances in efficient sampling of structures, detecting probable crystal polymorphs is no longer the bottleneck of successful CSP.[38,39] There are several algorithms that perform well in this task.[39–43] However, distinguishing which of these polymorphs is the most stable remains the most demanding task.[38,44–46] This is because in many cases, two or more of these low-lying polymorphs may have lattice energies that differ by less than 1 kcal mol^{-1}.[28,47,48] Therefore, coupling highly accurate wave-function methods with the MBE could potentially have a significant impact in overcoming this challenge.

Of course, the use of high-accuracy methods leads to a large computational cost. Hence, we developed CrystaLattE to be highly scalable and to allow for dual-level parallel computation. Our strategy to compute lattice energies also opens the possibility of using multiscale techniques that may help further increase efficiency.

In Sec. II, we present a brief explanation of the MBE approach and how we exploit it in the context of crystal lattice energy computations. We then describe the algorithm for automated computation of lattice energies implemented in CrystaLattE. Finally, the approach is applied to compute the crystal lattice energy of benzene using Grimme's triply corrected Hartree-Fock method (HF-3c).[49] We demonstrate that HF-3c, one of the simplest and cheapest wave-function methods currently available, turns out to be reasonably accurate for computing the crystal lattice energy of benzene using the MBE. We also discuss the convergence of two- and three-body interactions in this system with respect to cutoff distances at the HF-3c level.

II. THEORY

In the many-body expansion, the total energy of a molecular cluster is given as

$$E = \sum_{I}^{N} E_I + \sum_{I<J}^{N} \Delta E_{IJ} + \sum_{I<J<K}^{N} \Delta E_{IJK} + \cdots. \tag{1}$$

Here, N is the number of monomers in the cluster, E_I is the energy of monomer I, ΔE_{IJ} is the interaction energy of the dimer formed by monomers I and J,

$$\Delta E_{IJ} = E_{IJ} - E_I - E_J, \tag{2}$$

and ΔE_{IJK} is the nonadditive three-body energy of the trimer formed by monomers I, J, and K,

$$\Delta E_{IJK} = E_{IJK} - \Delta E_{IJ} - \Delta E_{IK} - \Delta E_{JK} - E_I - E_J - E_K. \tag{3}$$

Equation (1) is exact if it is not truncated (i.e., if one includes terms up to the nonadditive N-body contribution, for a cluster of N molecules). Truncating at some lower order (typically at the level of two-, three-, or four-body interactions) gives an efficient way to estimate the total energy at a substantially reduced computational cost.

The lattice energy of a crystal is defined as the energy required to construct the lattice starting from a state where all its molecules are infinitely separated. In our initial applications, we are examining fairly rigid molecules, and so, we neglect any monomer deformation terms. Hence, the lattice energy is the sum of the N-body interaction energies, ΔE_{IJ}, ΔE_{IJK}, etc., up through the chosen truncation level. However, if we model the crystal as an infinitely extended solid, then there are an infinite number of dimers, trimers, etc. So, we compute the lattice energy *per monomer* (or per mole of monomers) to obtain a finite result. For this purpose, we pick a *reference* monomer that will appear in all N-mers of the MBE. Then, we select all dimers, trimers, etc., that satisfy various filtering criteria discussed below.

For each N-mer retained, we compute its N-body interaction energy. This energy is divided by N, the number of monomers in the N-mer, to obtain the contribution *per monomer*. We only need to compute each symmetry-unique N-mer once and multiply it by the number of symmetry-equivalent N-mers that contain the same reference monomer. The contribution, $\mathscr{C}_{N\text{-mer}}$, of each N-mer to the crystal lattice energy is thus computed as

$$\mathscr{C}_{N\text{-mer}} = \mathscr{R}_{N\text{-mer}} \times \frac{\Delta E_{I\ldots N}}{N}, \tag{4}$$

where N is the number of monomers and $\mathscr{R}_{N\text{-mer}}$ is the number of replicas of each unique N-mer. For dimers, $\Delta E_{I\ldots N}$ is the dimer interaction energy, as defined in Eq. (2); for trimers and higher-order N-mers, $\Delta E_{I\ldots N}$ corresponds to the nonadditive portion of the many-body energy, e.g., as defined for a trimer in Eq. (3).

In our initial applications, we are primarily interested in nonpolar or weakly polar molecules. For strongly polar molecules, convergence may be aided by using electrostatic embedding techniques.[12,18,19] This feature has not yet been implemented, but it should be straightforward to do so.

III. ALGORITHM AND SOFTWARE IMPLEMENTATION

CrystaLattE was designed to take a CIF and return its lattice energy, using the MBE. The approach described in Eq. (1) offers several advantages. First, the strength of the interactions decays as the inverse of the powers of the separations between monomers: energies rapidly decay with the distance. Therefore, we have implemented distance cutoffs in CrystaLattE to make calculations more efficient. As our initially targeted applications involve lattice energy computations at fixed geometries, we have not yet employed smoothing functions for the distance cutoffs; this would be required to avoid discontinuities in the lattice energy as a function of the lattice parameters as monomers move inside or outside the cutoff distance. Of course, discontinuities will decrease in magnitude for larger cutoff values.

The second advantage of Eq. (1) is that the strength of the interactions of high-order N-mers will decay more rapidly with the distance than the low-order N-mers. For example, trimer interactions are expected to decay orders of magnitude faster than the dimer interactions. Thus, the contribution of high-order N-mers practically vanishes, and usually, the expansion can be truncated at the level of trimers or tetramers.[35]

Third, crystals are long-range, 3D-ordered structures that exhibit symmetrical properties. For that reason, the geometries of the many-body interactions occurring in the crystal are often repeated. Then, the number of interactions to be computed can be significantly reduced by detecting such redundancies. CrystaLattE includes a sophisticated filtering procedure designed to avoid unnecessary computations.

Fourth, the lattice energy is decomposed into inherently independent interactions. On the one hand, this characteristic allows for the application of multiscale modeling techniques: various methods can be applied to compute interactions at different ranges or for distinct N-mer orders. On the other hand, computations of independent interactions can be distributed, enabling pleasant parallelism: our code runs in a dual-parallel fashion, by distributing multithreaded jobs on multiple nodes.

A. Details of the implementation

CrystaLattE is distributed as free and open-source software, under the GNU Lesser General Public License version 3 (LGPL v3). We have made the code available through GitHub at https://github.com/carlosborca/CrystaLattE.

Originally, this project was thought of as an add-on to the computational chemistry package Psi4, which already contains some tools required to execute automated many-body computations.[50] Our code also relies on several tools contained in the Quantum Chemistry Common Driver and Databases (QCDB) set of modules and scripts.[51] Therefore, our code is written in Python 3, which simplified this integration. In addition, we also employ PyCIFRW, a module that provides support for reading and writing CIFs using Python.[52] PyCIFRW is now maintained and developed at the Australian Nuclear Science and Technology Organisation (ANSTO) by Dr. James R. Hester. Psi4, QCDB, and PyCIFRW are the software dependencies of CrystaLattE.

Currently, CrystaLattE uses the Psi4 Application Programming Interface (API) mode as its back-end for fully automated execution on a terminal. In that case, the user needs only to provide a CIF. However, CrystaLattE can also be executed in semiautomatic fashion. In this case, the user provides the CIF and the code returns all the input files that must be run. The user has the freedom to choose how to run those input files. After all outputs are generated, an analysis script extracts the quantities required, applies the MBE, and returns the lattice energy.

The semiautomatic mode provides flexibility and control on how to distribute calculations among the available resources. In addition, when running in semiautomatic mode, besides the ability to generate Psithon inputs for Psi4, CrystaLattE can also output LibEFP[53,54] and GAMESS[55,56] inputs to compute the many-body interactions with the Effective Fragment Potential (EFP) method.[57,58]

In Subsections III B–III F, we describe the general workflow of CrystaLattE and provide further detail on the most important parts of the code. In addition, a diagram summarizing the execution routine can be found in Fig. 1.

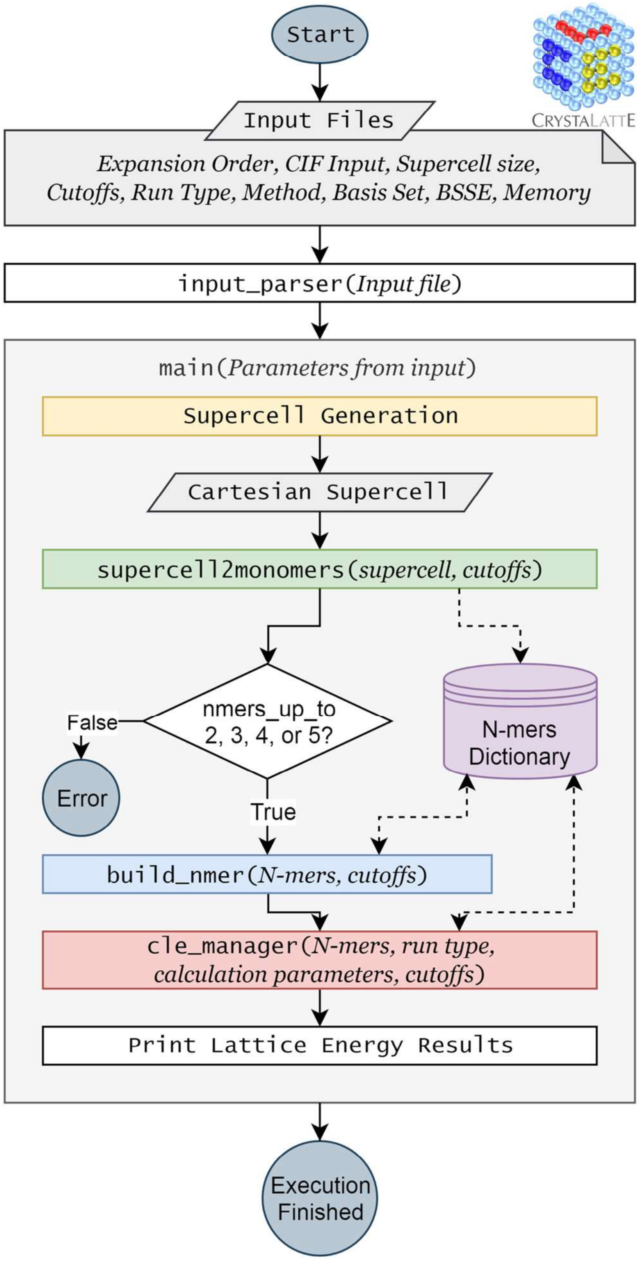
Start
CrystaLattE
Input Files
Expansion Order, CIF Input, Supercell size,
Cutoffs, Run Type, Method, Basis Set, BSSE, Memory
input_parser(Input file)
main(Parameters from input)
Supercell Generation
Cartesian Supercell
supercell2monomers(supercell, cutoffs)
nmers_up_to 2, 3, 4, or 5?
False
Error
True
N-mers Dictionary
build_nmer(N-mers, cutoffs)
cle_manager(N-mers, run type, calculation parameters, cutoffs)
Print Lattice Energy Results
Execution Finished

FIG. 1. General usage routine of CrystaLattE. Provided with a CIF file, the code automatically computes the lattice energy of a molecular crystal employing the MBE approach. The most important functions and data structures are colored.

B. Start-up routine and supercell preparation

The execution starts by reading a configuration input. This file contains information about the execution mode, computational resources, MBE truncation order, cutoffs, methods setup, and CIF details.

A CIF contains the information necessary to reconstruct a Cartesian representation of the unit cell of its corresponding crystal. Employing the `CifFile()` function from the PyCIFRW module, CrystaLattE extracts the positions of the atoms, the element labels, and the allowed symmetry operations of the unit cell.

The code then proceeds to generate a *supercell*, by replicating, a given number of times, the unit cell in the three crystalline-coordinate dimensions *a*, *b*, and *c*. Once the unit cell has been replicated in every direction, the code transforms the atomic-position coordinates from the crystalline coordinate system to the Cartesian coordinate system.

The unit cell must be replicated enough times such that a sphere of radius equivalent to the *monomer cutoff* (explained in Subsection III C) can be fully embedded in the resulting supercell. In other words, the shortest Cartesian dimension of the generated supercell must be at least twice the length of the monomer cutoff. Therefore, we replicate the unit cell an odd number of times in each dimension to guarantee that the original reference unit cell is kept closest to the origin. This is helpful in enumerating monomers as discussed in Subsection III C.

Then, the center of this supercell, calculated as the average of each of its Cartesian dimensions, is translated to the origin. Once the process is completed, the generated supercell is written to the execution directory as an XYZ file.

C. Identification of monomers

The next step is to detect all the monomers in the supercell. An implementation of the Breadth-First Search (BFS) algorithm[59] for molecular bonding, packaged in QCDB, is employed for that purpose.[51] The BFS is an algorithm for searching tree or graph data structures, like bond mapping in molecules. By traversing through nearest neighbors, it will detect which atoms are connected by a bond using criteria based on typical van der Waals radii of elements.

Once all the monomers in the supercell have been identified, CrystaLattE enumerates the monomers starting from the one with the shortest position vector from the origin. That is why the unit cell is replicated an odd number of times to guarantee that the indexing process starts from the center of the central unit cell and not from one of its edges. That procedure ensures that the indices are assigned in increasing order of proximity to the center of the supercell. At the same time, the algorithm discards any monomers for which their atom closest to the origin has a position vector longer than the monomer cutoff.

As shown in Fig. 2, the result is a quasispherical aggregate of monomers that have at least one atom whose distance to the origin is within the monomer cutoff. The code creates a dictionary for each monomer, containing information about its geometry and center of mass (COM).

D. Construction and filtering of *N*-mers

In most cases, truncation of the MBE at the level of trimers should provide acceptable results. However, CrystaLattE can compute the MBE up to the order of pentamers. After all monomers within the monomer cutoff distance are extracted from the supercell, the next task is the construction of all *N*-mers, up to the level requested by the user.

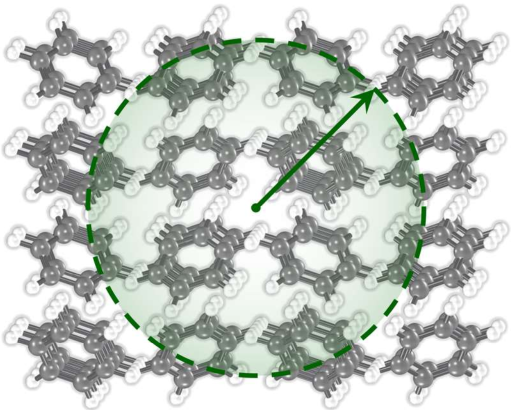

FIG. 2. Illustration of the monomer cutoff. From the center of the supercell, any monomer that includes at least one atom inside a sphere of radius equivalent to the user-specified monomer cutoff will be stored; otherwise, the monomer will be discarded.

An *N*-mer builder function has been implemented for that purpose. Taking the first monomer, the one closest to the origin, as the reference, the function combines it with all other monomers to form all the dimers that contain the reference monomer. If the user has requested the expansion to include higher *N*-mers, CrystaLattE follows the same procedure: taking the first monomer as a reference, it forms all *N*-mers that arise from permutationally distinct combinations with all the other monomers.

Given that crystals are repetitions of a unit cell, many of the *N*-mer structures generated may be symmetry-equivalent replicas of each other. In addition, as briefly discussed in Sec. II, the contribution of an *N*-mer to the lattice energy decays rapidly when the separation of its monomers increases, especially for high-order *N*-mers. In such cases, calculations of many long-range separated *N*-mers may turn out to be unnecessary. Therefore, we have devised a series of filtering schemes to keep the number of required calculations to the minimum.

The importance of the implementation of these filters will be discussed later in Sec. V, but they are essential to keep the computational cost of the MBE approach within practical reach. There are five filters coded in the *N*-mer builder function. The first checks that the distance between the closest neighbors of each pair of monomers in the *N*-mer is shorter than a given *N-mer cutoff* (see Fig. 3). The user can specify different cutoffs for different *N*-mer orders. The second verifies that the distance between the COMs of each pair of monomers in the *N*-mer is shorter than a given *COM cutoff* (see Fig. 3). This is a global cutoff affecting *N*-mers of all orders. The next three filters are designed to face the problem of symmetry-uniqueness.

Once the cutoffs have been satisfied, if the algorithm finds two structures that are identical, only the one generated first will remain. The other one will be counted in as a replica of the first and discarded. A *replica multiplier* will be later used to account for all discarded replicas when reconstructing the MBE energy.

The third filter works comparing the nuclear repulsion energy (NRE) between two *N*-mers to determine if they are different. Calculating the NRE of a chemical system is a simple and fast operation.

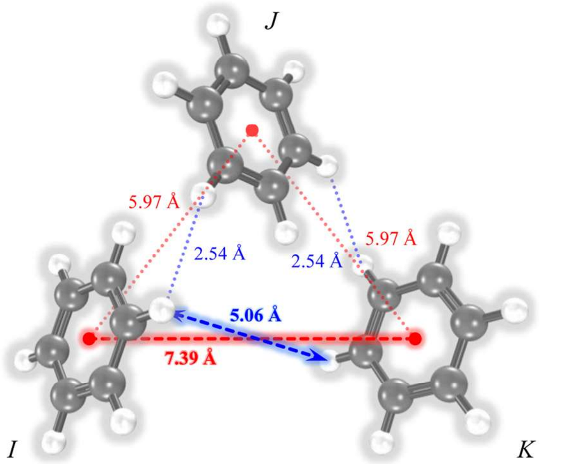

FIG. 3. Different types of cutoffs for a benzene trimer. The red dashed line shows the COM cutoff. Among the three vectors between each pair of COMs, the longest shall not be longer than a user-specified distance. The blue dashed line shows the trimer cutoff. Taking the shortest interatomic separation between each pair of monomers, the longest of these three vectors shall not be longer than a user-specified distance.

The NRE filter can unequivocally determine if two dimer structures are identical or not. However, oddly enough, we have found that for trimers and higher-order N-mers, there are cases in which different structures can have exactly equivalent NREs (we discuss one of these cases and its implications in Appendix A). If the NRE of a new N-mer is different from that of each of the previously stored N-mers, the N-mer is kept. However, if the NRE matches that of a previously generated N-mer, then the algorithm needs to proceed to the next filters: the chemical-space descriptor filter or ultimately the *B787 Dreamaligner* filter. These two filters are complex enough to merit a deeper explanation.

1. The chemical-space filter

CrystaLattE employs a machine-learning descriptor that defines a molecule's point in *chemical space* to determine uniqueness among N-mers.[60] The code computes the descriptor by finding and sorting the eigenvalues of a slightly modified Coulomb matrix, usually denoted as **M**.

The off-diagonal elements are determined by the Coulombic interaction between the charge Z_i of nucleus i and the charge Z_j of nucleus j, separated by a distance R_{ij},

$$M_{ij} = \frac{Z_i Z_j}{R_{ij}}. \tag{5}$$

As explained by Rupp *et al.*, the expression for diagonal elements was determined by a polynomial fit of nuclear charge to atomic energies,[60]

$$M_{ii} = \frac{1}{2} Z_i^{2.4}. \tag{6}$$

This descriptor encodes the identities of the atoms, through its dependence upon nuclear charges, and it is invariant to translation and rotation, due to its dependence on interatomic distances.

Once the point in chemical space is defined for a given N-mer, its equivalency to another N-mer can be determined by computing the distance in chemical space from such other N-mer. The distance is computed as the *norm between the sorted arrays of eigenvalues*.

Using this metric, CrystaLattE can identify geometrically equivalent N-mers rapidly. Two geometrically equivalent systems will have the same ordered eigenvalues, independent of the ordering of the atoms in **M** (Appendix B provides further details).

2. The B787 Dreamaligner filter

Although the chemical space eigenvalues descriptor turned out to be extremely reliable, it was implemented after we initially developed and tested a function to align N-mers, which we call the *B787 Dreamaligner*.

The B787 function[51,61] takes two molecules of identical atom types and identical, superimposable, mirror-superimposable, or nonsuperimposable geometries and computes the best shift, rotation, and atom map (and optional mirror-image boolean) for aligning one (C) to the other (R) that minimizes the root-mean-square deviation (RMSD) of the atomic coordinates. The optimal rotation, **U**, is obtained by minimizing the norm of the residual $\|\mathbf{R} - \mathbf{UC}\|$ using the unit quaternion formulation of the Kabsch algorithm.[62,63] Since a physically meaningful Kabsch result requires correct atom mapping, the smallest Kabsch RMSD is selected after iterating over all reasonable atom mappings, a number that increases factorially with the system size if generated by simple permutation among atoms of a type. To heavily filter atom mappings, the Kuhn-Munkres (Hungarian) algorithm[64,65] is first used to find a best mapping of the two systems' atoms by minimizing a *cost matrix* formed by a Coulomb matrix-based measure of how well an atom of system C replaces a like-type atom of system R by similarity of distance to other like-type atoms. The resulting optimal mapping is invariant to exchange of symmetry equivalent atoms due to the cost matrix's formulation as a holistic descriptor of each atom to the whole system. Since Kabsch operates back in Cartesian space where atom ordering does matter (a square with vertices numbered around the perimeter cannot be rotated onto a congruent square with vertices numbered across the diagonal), all spatial-symmetry-equivalent mappings need to be generated and evaluated. The reduced cost matrix (maximal zeros) emerging from the Kuhn-Munkres is used to seed the Uno algorithm[66] for enumerating the complete set of optimal solutions (and sorted nonoptimal) given one optimal solution. To provide leeway for nonexact alignments, a user-adjustable parameter controls the quantity of nonoptimal mappings to pass to Kabsch. The Kuhn-Munkres and Uno algorithms are applied to each atom type separately, and the resulting mapping candidates combined permutatively and returned as a generator to the Kabsch RMSD minimization loop. Several options are available to the user to tune whether systems C and R are expected to be nonsuperimposable, mirror-superimposable, or already like-ordered so that the stages can be skipped if unneeded. While the Kabsch algorithm was implemented directly from the literature, the Kuhn-Munkres and Uno algorithms were lightly adapted and integrated from independent implementations in, respectively, the SciPy library[67] and a software repository on GitHub.[66,68]

For practical applications, CrystaLattE employs the chemical-space descriptor detailed above to detect replicas. This is because the Coulomb-matrix algorithm is faster in our implementation, especially for large molecules. However, CrystaLattE also provides the aligner as an alternative option.

E. Prioritizing calculations

Once symmetry-unique N-mers have been found, redundancies accounted for, and far-separated structures rejected, the next step is determining the order in which the calculations should run.

We have devised a criterion to prioritize the execution of those N-mers from which we expect a greater contribution to the lattice energy, inspired by the R^{-9} dependence of the Axilrod-Teller-Muto (ATM) potential.[69,70] In the ATM potential, the strength of the dispersion interaction of three bodies A, B, and C depends inversely upon the separations between them: $1/(R_{AB}^3 R_{AC}^3 R_{BC}^3)$.

If the bodies are small molecules, the distances R_{AB}, R_{AC}, and R_{BC} are defined as the separation between COMs of three monomers A, B, and C. For large anisotropic molecules, however, this approximation may break down.[23,71] Therefore, instead of using COM separations by default, we use closest-contact separations in CrystaLattE.

For a given N-mer, its priority criterion, $\mathscr{P}$, is then calculated using the closest-contact separations among all pairs of monomers that compose it,

$$\mathscr{P} = \prod_{I<J} \frac{1}{R_{ij}^3}. \tag{7}$$

In this case, i and j denote the closest atoms in two different monomers I and J, and the product runs over all dimers in the N-mer. Alternatively, if the user chooses, R_{ij} may be replaced by R_{IJ}, the separation among COMs of the monomers I and J, to resemble the ATM potential. $\mathscr{P}$ is computed and stored when the N-mer is built. Thus, the execution order is established when the population of the N-mers dictionary takes place, with N-mers computed in order of descending values of $\mathscr{P}$.

F. Computation manager

The final piece of the code is the computation manager, which controls the execution of calculations and the extraction of required data from their corresponding outputs. Each N-mer computation is totally independent of the others, so they can be executed in a pleasantly parallel fashion. When running in semiautomatic mode, once a calculation has been executed, the order used to construct the partial lattice energy is controllable by the user. In addition, each individual N-mer computation may be run on multiple cores, leading to a dual-level parallelism.

The CrystaLattE manager has several run types. The default execution mode is `psi4api`, which is fully automatic, and runs Psi4 in the back-end through its API, utilizing the maximum number of threads available. When running in `psi4api` mode, the manager will access the N-mers dictionary and start executing the calculations for dimers first, then trimers, and so on, according to their priority order.

When each calculation is completed, the manager retrieves that N-mer's contribution to the many-body energy from the Psi4 API and stores it in the dictionary of the corresponding N-mer. Then, it multiplies that quantity times the corresponding number of replicas and divides over the number of monomers in the N-mer to obtain the contribution of that structure to the crystal lattice energy, as in Eq. (4). After the arithmetic is done, the code prints the results on a summary table, which is also written to disk in a comma-separated values (CSV) file.

By contrast, the `psithon` execution mode is designed to be semiautomatic: it requires no user intervention to generate the N-mer structures, but instead of running them on the back-end, it writes a standard Psi4 input file for each unique N-mer structure so that the user may execute them manually. These files are written to disk on the working directory where the CrystaLattE input is located.

Each of the Psi4 inputs contains comments with relevant data about the structure, such as the N-mer name, the number of replicas, the priority quantities, closest-contact separations, and COM separations. These pieces of information are useful once all outputs have been gathered. At this point, the user executes CrystaLattE in `psithonyzer` mode to reconstruct the lattice energy from the contributions of each structure.

Besides these regular execution modes, the manager can also generate inputs to be run on GAMESS or in LibEFP, including the generation of MAKEFP protocol inputs to generate effective fragment potentials (EFPs), if these are not available. An implementation to generate NWChem[72] input files is in progress. Eventually, we expect to support many quantum chemistry programs via the QCDB project.

IV. COMPUTATIONAL DETAILS

Calculations performed with Kruse and Grimme's geometric counterpoise (gCP) scheme in conjunction with dispersion-corrected HF on minimal basis sets can yield accurate binding energies at a low computational cost.[73] Consequently, we decided to test the HF-3c method[49] with its natural MINIX basis set and assess its performance at computing the lattice energy of crystalline benzene with the MBE.

A. CrystaLattE setup

An experimentally determined structure of crystalline benzene at 138 K was obtained in CIF format from the Cambridge Structural Database (CSD).[74] The structure corresponds to CSD code `BENZEN01`, and it was the same that was employed in previous computational studies of crystalline benzene.[30,33,34,75]

The unit cell described by the CIF was replicated seven times in each crystalline-coordinate dimension to generate the supercell. A monomer cutoff distance of 15 Å was employed to obtain the quasi-spherical supercell. The MBE was truncated at the level of tetramers for input generation, but calculations were run only on dimers and trimers. With the intention of considering all possible two-, three-, and four-body interactions in this system consistent with the monomer cutoff, the COM, dimer, trimer, and tetramer cutoffs were set to an arbitrary large distance so that the monomer cutoff is the only distance-based filtering procedure to discard structures. CrystaLattE was executed in the semiautomated *Psithon* mode.

B. Psi4 setup

All calculations were executed using Psi4 1.3a1. Although CrystaLattE supports the Valiron-Mayer (or hierarchical) function counterpoise (VMFC)[76] and the Boys and Bernardi counterpoise (CP) correction,[77,78] for basis set superposition error treatment, they were not used in these calculations. Instead, the HF-3c method contains its own internal corrections for basis set superposition errors. The self-consistent field (SCF) procedure was carried out using the density-fitting approximation. The convergence criterion for the SCF energy was set at 10^{-8} a.u. to improve the numerical accuracy of the results. The convergence criterion for SCF density, defined as the maximum absolute value of the orbital gradient, was left at 10^{-6} a.u.

V. RESULTS AND DISCUSSION

In this section, we present the results focusing first on software-related details and then on the data obtained for the benzene crystal using the HF-3c method.

A. Execution details

All tests described below were executed on a computer with an Intel i7-3930K processor running at 3.20 GHz, with 6 physical cores, and a total of 12 threads. The system also included 64 GB of total RAM memory and a local RAID-0 array of scratch disks consisting of three 7200 RPM hard drives.

The first step of the algorithm was the generation of the quasi-spherical aggregate from the supercell structure. Immediately after, the BFS algorithm was executed to identify all the monomers present in this quasisphere. The BFS algorithm required 236 s and found 166 monomers in the aggregate.

The process continued by generating all possible dimers that contain the reference monomer anchored in the central unit cell. Then, the filtering process eliminated any replicas and stored only the symmetry-unique dimers. 92 dimers were found to be replicas of another dimer and 73 unique dimers were identified and stored. The execution of this stage took 1 s. Next, all possible trimers that contained the reference monomer were generated and analyzed by the filtering scheme. 6400 trimers were found to be replicas of another trimer and were discarded. 7130 symmetry-unique trimers were found and stored. The trimers' generation and filtering process took 203 s. We could refer to this as the *input preparation time*, and we expect this to be longer in `psithon` mode than in `psi4api` mode, considering that the `psithon` mode writes an input file to disk for each *N*-mer.

Although we limited the execution of calculations at the level of trimers, we wanted to explore how the process of generation and filtering at the level of tetramers would work. Therefore, tetramer structures were requested, but their corresponding calculations were not run. CrystaLattE found 464 707 symmetry-unique tetramers and 270 423 replicas that were discarded. The execution of the tetramers' procedure took less than 35 h, including writing 464 707 input files to a network attached file server.

It is worth noting that these numbers of structures were determined using the monomer cutoff exclusively. In practical terms, the user would almost certainly request dimer, trimer, or tetramer cutoffs that would be equal or shorter than the monomer cutoff. In other words, no structures were being filtered out by any of these *N*-mer cutoffs; thus, the number of unique structures was the maximum for a 15 Å monomer cutoff. Figure 4 analyzes how the number of structures of each type changes with respect to the monomer cutoff.

The execution wall-clock time for each of these Psi4 HF-3c calculations varies by a few seconds. A dimer calculation elapsed about 3 s, a trimer was around 30 s, and a tetramer would be approximately 5 min. If all outputs were run sequentially, adding up all the wall-clock execution times for the whole set of dimer calculations, the total elapsed time was under 4 min. The whole set of trimers added up to under 60 h. We call this the *sequential execution time*. However, these calculations would be routinely distributed among multiple nodes allowing for significant reductions of the wall-clock time required to complete this stage. Additionally, in more typical applications, the number of trimers would be substantially reduced using a trimer cutoff. In addition, if the computations were run in priority order, the trimer computations could cease after the three-body energy appeared to be converging. Finally, after execution of all calculations, the extraction of data and computation of the MBE, what we could call the *postprocessing time*, elapsed 39 s for dimers and trimers.

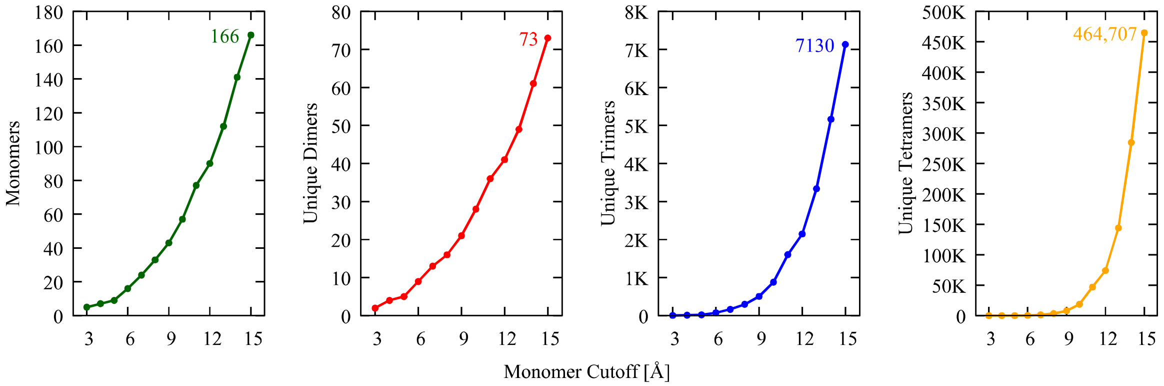

FIG. 4. Comparison of the number of structures with respect to the radius of the quasispherical supercell. Within a radius cutoff of 15 Å, 166 monomers are detected, which are then combined to form 73 unique dimers, 7130 unique trimers, and 464 707 unique tetramers. Distance-based *N*-mer cutoffs (see text and Fig. 3) may be used to limit the number of *N*-mers actually computed.

Disk usage in psithon mode varied depending on the level at which the expansion was truncated. 18.8 MB was required to write the Psi4 inputs of dimers and trimers, and 1.55 GB would be required if the nontruncated set of all tetramers consistent with a 15 Å monomer cutoff were included. After execution of all the dimer and trimer inputs, the total disk usage to store the generated outputs is about 524 MB. Again, because we were not employing any N-mer cutoffs, these are much larger than typical disk usage values.

B. Benzene crystal lattice energy with HF-3c

Figure 5 shows, in the logarithmic scale, the two-body energy for each dimer [ΔE_{IJ} of Eq. (2)] against the distance of separation of the closest-contact between atoms of the two different monomers (i.e., the *dimer cutoff* distance in CrystaLattE).

Note that there are conformations in which the closest-contact separation of a given dimer is slightly longer than the monomer cutoff. This is because the monomer cutoff is not taken from the COM of the reference monomer, but from the center of the supercell. The reference monomer is the closest one to that point, but it is not necessarily centered on the center of the supercell.

The distribution of the data in the plot resembles an exponential decay, indicating a major predominance of the closest separated dimers. The first three dimers, the ones on the first shell surrounding the reference monomer, contribute with interaction energies that are about an order of magnitude larger than any of the other structures in the plot. The two-body energies quickly vanish to <0.1 kJ mol^{-1} at distances longer than 6 Å.

In addition, the vertical variation of the points with similar separation values indicates that even though the distance between dimers is the main contributing factor to determine the strength of the two-body interaction, it is clearly not the only one. This reflects that the orientation of the monomers with respect to each other, i.e., stacked, T-shape, coplanar, etc., can introduce significant variations in the dimer interaction energy that can be as large as an order of magnitude for equally separated dimers.

Because the separation between monomers in the dimer plays a significant role in determining the two-body interaction energy, it is reasonable to order the sum using the closest-contact separation for each dimer. Figure 6 shows the partial lattice energy as the sum of each dimer contribution, as calculated from Eq. (4), against the closest-contact separation, or the distance that we use to define the dimer cutoff.

Using the cutoffs described above, the total two-body contribution to the crystal lattice energy is −60.1 kJ mol^{-1}, according to HF-3c. This value is only −4.8 kJ mol^{-1} off that reported by Kennedy *et al.*,[33] −55.3 kJ mol^{-1}, calculated using at the coupled cluster through perturvative triples [CCSD(T)] complete-basis-set (CBS) limit with a limited number of dimers. It is off by −2.5 kJ mol^{-1} with respect to the −57.64 reported by Chan and co-workers,[34] also estimating the CCSD(T)/CBS limit, but with a slightly different methodology and a more complete set of dimers. This is a remarkably small error considering the inexpensive computational cost of HF-3c compared to that of CCSD(T)/CBS.

Figure 6 also shows a rapid convergence of the total two-body contribution. The final value would be just <0.5 kJ mol^{-1} different if it was truncated at 8 Å. Besides, the first three dimers contribute −53.3 kJ mol^{-1} or roughly 89% of the total two-body contribution to the crystal lattice energy, reiterating the predominance of those dimers that form the first shell surrounding the reference monomer. Because their contribution is so large, it is likely that errors for these three dimers dominate the overall error in the two-body term. This is an important observation because it implies that a combined-method approach could be practical and accurate: the error could be reduced by employing a more sophisticated method for the first several dimers and then HF-3c for the remainder.

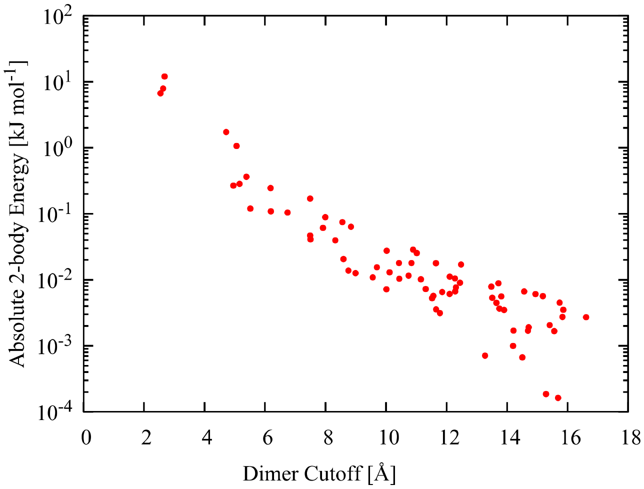

FIG. 5. Analysis of the individual dimer interaction energies. The absolute value of the energy is plotted. The two-body interaction seems to decay quasiexponentially with respect to the distance between the monomers. The interaction of the first three dimers is about an order of magnitude stronger than any of the others.

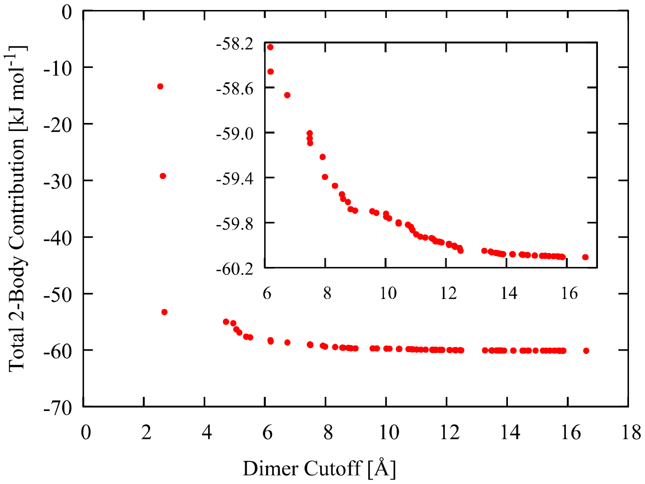

FIG. 6. Accumulation of two-body contributions to the lattice energy as a function of the distance between monomers. The first three dimers, whose monomers are separated by less than 2.5 Å, account for roughly 89% of the total two-body contribution. The subplot shows a zoomed-in view of the intermediate-separation to long-separation region. Convergence to less than 0.5 kJ mol^{-1} is achieved at a dimer cutoff of 8 Å.

Figure 7 shows, in the logarithmic scale, the three-body energy for each trimer [ΔE_{IJK} of Eq. (3)] against the distance of separation of the *longest* of the three closest-contact distances between the atoms on the three monomers (the *trimer cutoff* distance). We are using a logarithm plot to help determine the rate of convergence of the three-body energy contributions. However, here we take the logarithm of the absolute value of ΔE_{IJK} because for trimers that energy can be positive. Although there is more vertical variation than in the case of dimers, the distribution of the points in the plot also resembles an exponential decay, at least in the range below 20 Å. This points out again that the closest separated trimers are more relevant. However, the comparison with the case of dimers is more complex because there are three separation vectors between the monomers and the energy depends upon all these distances. One interesting observation is that long-range three-body interactions seem to level off at about 10^{-4} kJ mol^{-1}. This could reflect that these energies are at or below the numerical precision set for these calculations.

Figure 8 shows the partial lattice energy as the sum of each trimer contribution, also calculated from Eq. (4), against the closest-contact separation. The sum is also ordered using the longest contact separation among monomers of a trimer or what we denote as the trimer cutoff distance. The total three-body contribution to the crystal lattice energy is expected to be significantly smaller than the two-body contribution. Using HF-3c with the chosen cutoffs, the calculated value is 1.1 kJ mol^{-1}. That quantity is −2.6 kJ mol^{-1} off the 3.7 kJ mol^{-1} computed at mixed levels of theory including CCSD(T)/CBS and ATM-corrected MP2/aug-cc-pVDZ by Kennedy *et al.*[33] Although qualitatively correct, this is a larger error than that of the two-body contribution. HF-3c will capture (approximately) three-body exchange and induction/polarization, but it lacks any description of three-body dispersion, which is the dominant three-body effect in crystalline benzene.[33,79] As discussed in the literature, capturing three-body dispersion effects explicitly is challenging.[42,80,81] Simplified models, such as the ATM potential, have been shown to be a useful alternative.[33] If we apply the ATM potential to estimate the missing three-body dispersion, using a molecular ATM constant of 82 657.65 a.u. from previous estimations using Density Functional Theory-based Symmetry-Adapted Perturbation Theory [SAPT(DFT)][31] and all trimers considered in our HF-3c computations, we obtain a value of 2.7 kJ mol^{-1}. This is quite similar to the 3.2 kJ mol^{-1} estimate for three-body dispersion reported by Kennedy *et al.*[33]

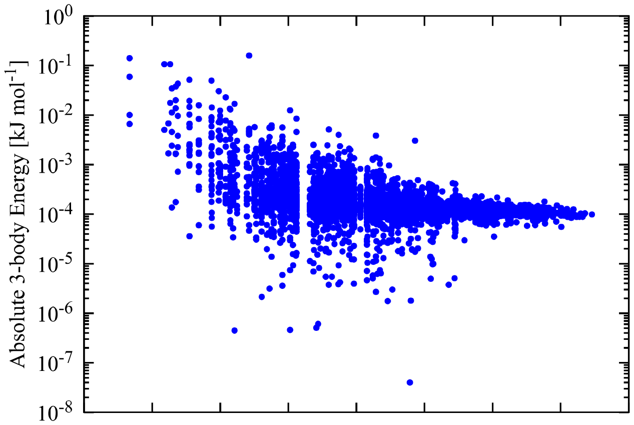

FIG. 7. Analysis of the trimer nonadditive interaction energies. The absolute value of the energy is plotted. Below a trimer cutoff of 20 Å, the nonadditive three-body energy seems to decay exponentially with respect to the longest contact distance between the monomers in a trimer. This again indicates that the most relevant contributions to the lattice energy are generated by the closest-separated trimers. However, after 20 Å, the three-body nonadditive energies seem to even out below 10^{-4} kJ mol^{-1}. The three-body interactions are generally weaker than the two-body ones.

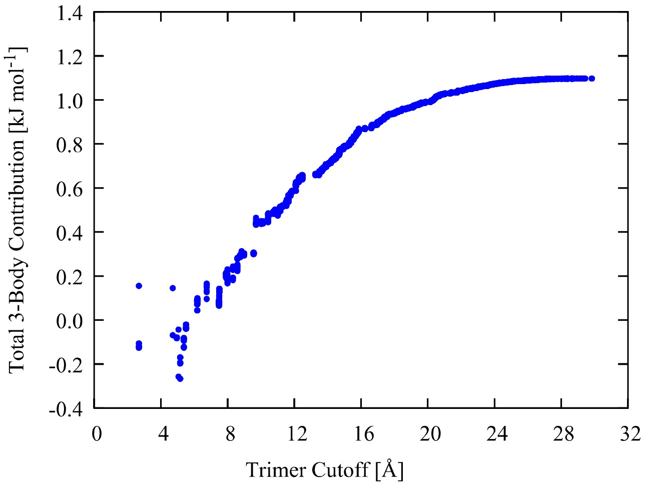

FIG. 8. Accumulation of three-body contributions to the lattice energy as a function of the longest contact distance between monomers in a trimer. Convergence to less than 0.5 kJ mol^{-1} is achieved at a trimer cutoff of 16 Å. The three-body contribution is extremely small compared to the two-body contribution for benzene.

Adding up both the dimer and trimer HF-3c contributions, the computed crystal lattice energy of benzene is −59.0 kJ mol^{-1}. Further adding the ATM estimate of three-body dispersion yields −56.3 kJ mol^{-1}. This value compares extremely well with the reference MBE-computed crystal lattice energies of −51.6 and −54.6 kJ mol^{-1} reported in 2014,[33,34] using the same 138 K crystal structure. Indeed, this level of agreement (differences of −4.7 and −1.7 kJ mol^{-1}, respectively), is remarkable, given the simplicity of the HF-3c method with respect to the highly sophisticated approaches used in the two studies from 2014. Our computed crystal lattice energy is also just −1.0 kJ mol^{-1} from the estimated experimental 0 K lattice energy, −55.3 ± 2.2 kJ mol^{-1}, obtained by Chan and co-workers from experimental data and corrected to remove zero-point effects, using a computed zero-point energy.[34] Chan and co-workers estimated that the difference in the lattice energy between the 138 K structure and the hypothetical 0 K structure, due to geometry relaxation of the crystal lattice and the gas-phase monomer, is −1.32 ± 0.1 kJ mol^{-1}.[34] If we add this correction to our computed value to estimate a 0 K lattice energy, we obtain −57.6 kJ mol^{-1}, which is 2.3 kJ mol^{-1} below the experimental 0 K lattice energy.

VI. CONCLUSIONS

We have introduced CrystaLattE, a software tool to automate the computation of lattice energies of organic crystals, given a crystallographic information file. CrystaLattE provides the possibility of achieving pleasant parallelism in a dual-level fashion by distributing the independent multithreaded calculations among the available node resources. We used CrystaLattE to compute the lattice energy of crystalline benzene with HF-3c, obtaining a value that is off by just −4.4 kJ mol^{-1} from the high-level *ab initio* reference value by Chan and co-workers.[34] Adding a correction for three-body dispersion using a simple Axilrod-Teller-Muto potential reduces this error to −1.7 kJ mol^{-1}. With this correction included, the error vs the estimated 0 K experimental lattice energy is −1.0 kJ mol^{-1}. This level of agreement is fortuitous, given the simplicity of the model, but encourages additional investigation. In future work, we will employ CrystaLattE to obtain benchmark-quality lattice energies for a wide range of molecular crystals, and using those results, we will examine the accuracy of HF-3c and various density functional approximations for lattice energy predictions.

SUPPLEMENTARY MATERIAL

See the supplementary material for complete CrystaLattE input, output, and analysis files for crystalline benzene with the HF-3c method.

ACKNOWLEDGMENTS

C.H.B. is thankful to Asem Alenaizan for his comments regarding software and to Dr. Thomas Darden for the enriching discussions about crystals and the HF-3c method. We thank both referees of this paper for insightful comments, particularly the suggestion to examine the effect of three-body dispersion in more detail. We gratefully acknowledge support from a U.S. Department of Defense HPCMP Applications Software Initiative (HASI) grant and the U.S. National Science Foundation through a Sustainable Software Infrastructure Grant (No. ACI-1449723). This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology.

APPENDIX A: NRE DESCRIPTOR AND UNIQUENESS

An interesting finding related to this study is that it is common to find different structures that have identical NREs. We present one of those cases in Fig. 9. These two trimers have NREs that match up to 12 digits of precision (1112.571 923 330 761 hartree). Yet, their distance in chemical space is significant, meaning that there is no symmetry operation that can make these two structures equivalent. A careful inspection of their conformations reveals that both trimers are formed by the same dimer. However, the third monomer is located in different positions in each trimer. In one case, the third monomer acts as a donor in a T-shape conformation with one of the other monomers, but in the other case, the third monomer is the acceptor in the T-shape. This case proves that the NRE can be used as a descriptor of dissimilarity, but it cannot determine uniqueness.

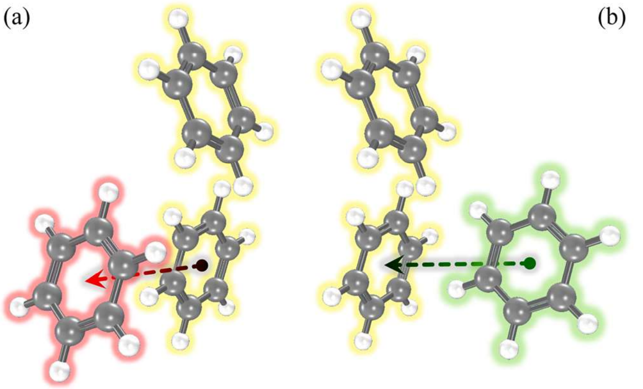

FIG. 9. The structures of two trimers (a) and (b) that have identical NRE, but are not equivalent geometries. In both trimers, there are two monomers in identical positions (yellow) and a third that is located on a different place on each trimer (red, green). In trimer (a), the monomer in the back (yellow) has it hydrogen atoms pointing toward the monomer in the front (red), forming a T-shape conformation; whereas in trimer (b), the monomer in the front (green) has its hydrogen atoms pointing toward the monomer in the back (yellow), forming an equivalent T-shape conformation.

APPENDIX B: CHEMICAL SPACE EIGENVALUES DETAILS

As already mentioned, the modified Coulomb matrix **M** [Eqs. (5) and (6)] is invariant to translations and rotations of one N-mer vs another, which is an important property when applying it to determine chemical uniqueness. If two N-mers are chemically equivalent, then the same matrix elements will be present in their Coulomb matrices **M** and **M′**. However, we have not taken steps to ensure that CrystaLattE generates chemically equivalent N-mers using the same atom ordering. Here, we show that the modified Coulomb matrices are also invariant to permutations of atom numbering.

If the ordering of atoms i and j is swapped between systems, **M** and **M′** will not be equal. To make the matrices equal, rows i and j need to be swapped as well as columns i and j. This can be achieved by right-multiplying **M** with the appropriate permutation matrix **P** to swap columns i and j and left-multiplying by the transpose of **P** to swap rows i and j,

$$\mathbf{P}^T\mathbf{M}\mathbf{P} = \mathbf{M}'. \tag{B1}$$

Because the transpose of a permutation matrix, $\mathbf{P}^T$, is the inverse of **P**, $\mathbf{P}^T\mathbf{M}\mathbf{P}$ is a similarity transform of **M**, and so the eigenvalues of **M** will be preserved despite different atom orderings in **M** and **M′**.

REFERENCES

[1] D. Hankins, J. W. Moskowitz, and F. H. Stillinger, J. Chem. Phys. **53**, 4544 (1970).
[2] H. Stoll and H. Preuß, Theor. Chim. Acta **46**, 11 (1977).
[3] J. C. White and E. R. Davidson, J. Chem. Phys. **93**, 8029 (1990).
[4] H. Stoll, Phys. Rev. B **46**, 6700 (1992).
[5] H. Stoll, J. Chem. Phys. **97**, 8449 (1992).
[6] H. Stoll, Chem. Phys. Lett. **191**, 548 (1992).

[7] S. S. Xantheas, J. Chem. Phys. **100**, 7523 (1994).

[8] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi, Chem. Phys. Lett. **313**, 701 (1999).

[9] K. Szalewicz, R. Bukowski, and B. Jeziorski, *Water in Confining Geometries* (Springer Berlin Heidelberg, 2003), pp. 7–23.

[10] B. W. Hopkins and G. S. Tschumper, J. Comput. Chem. **24**, 1563 (2003).

[11] B. W. Hopkins and G. S. Tschumper, Mol. Phys. **103**, 309 (2005).

[12] E. E. Dahlke and D. G. Truhlar, J. Chem. Theory Comput. **3**, 46 (2006).

[13] M. S. Gordon, D. G. Fedorov, S. R. Pruitt, and L. V. Slipchenko, Chem. Rev. **112**, 632 (2011).

[14] U. Góra, R. Podeszwa, W. Cencek, and K. Szalewicz, J. Chem. Phys. **135**, 224102 (2011).

[15] B. Johnson and R. J. Hinde, J. Phys. Chem. A **115**, 7112 (2011).

[16] D. M. Bates, J. R. Smith, T. Janowski, and G. S. Tschumper, J. Chem. Phys. **135**, 044123 (2011).

[17] G. J. O. Beran and S. Hirata, Phys. Chem. Chem. Phys. **14**, 7559 (2012).

[18] P. J. Bygrave, N. L. Allan, and F. R. Manby, J. Chem. Phys. **137**, 164102 (2012).

[19] M. J. Gillan, D. Alfè, P. J. Bygrave, C. R. Taylor, and F. R. Manby, J. Chem. Phys. **139**, 114101 (2013).

[20] J. Gao, J. Z. H. Zhang, and K. N. Houk, Acc. Chem. Res. **47**, 2711 (2014).

[21] M. A. Collins and R. P. A. Bettens, Chem. Rev. **115**, 5607 (2015).

[22] K. Raghavachari and A. Saha, Chem. Rev. **115**, 5643 (2015).

[23] J. Řezáč, Y. Huang, P. Hobza, and G. J. O. Beran, J. Chem. Theory Comput. **11**, 3065 (2015).

[24] F. Manby, *Accurate Condensed-Phase Quantum Chemistry* (CRC Press, 2010).

[25] S. Hirata, J. Chem. Phys. **129**, 204104 (2008).

[26] O. V. Shishkin, R. I. Zubatyuk, A. V. Maleev, and R. Boese, Struct. Chem. **25**, 1547 (2014).

[27] S. Hirata, K. Gilliard, X. He, J. Li, and O. Sode, Acc. Chem. Res. **47**, 2721 (2014).

[28] G. J. O. Beran, Chem. Rev. **116**, 5567 (2016).

[29] S. Mattsson, B. Paulus, F. A. Redeker, H. Beckers, S. Riedel, and C. Müller, Chem. - Eur. J. **25**, 3318 (2019).

[30] A. L. Ringer and C. D. Sherrill, Chem. - Eur. J. **14**, 2542 (2008).

[31] R. Podeszwa, B. M. Rice, and K. Szalewicz, Phys. Rev. Lett. **101**, 115503 (2008).

[32] G. J. O. Beran and K. Nanda, J. Phys. Chem. Lett. **1**, 3480 (2010).

[33] M. R. Kennedy, A. R. McDonald, A. E. DePrince, M. S. Marshall, R. Podeszwa, and C. D. Sherrill, J. Chem. Phys. **140**, 121104 (2014).

[34] J. Yang, W. Hu, D. Usvyat, D. Matthews, M. Schutz, and G. K.-L. Chan, Science **345**, 640 (2014).

[35] C. Müller and B. Paulus, Phys. Chem. Chem. Phys. **14**, 7605 (2012).

[36] S. R. Hall, F. H. Allen, and I. D. Brown, Acta Crystallogr., Sect. A: Found. Crystallogr. **47**, 655 (1991).

[37] H. J. Bernstein, J. C. Bollinger, I. D. Brown, S. Gražulis, J. R. Hester, B. McMahon, N. Spadaccini, J. D. Westbrook, and S. P. Westrip, J. Appl. Crystallogr. **49**, 277 (2016).

[38] D. A. Bardwell, C. S. Adjiman, Y. A. Arnautova, E. Bartashevich, S. X. M. Boerrigter, D. E. Braun, A. J. Cruz-Cabeza, G. M. Day, R. G. D. Valle, G. R. Desiraju, B. P. van Eijck, J. C. Facelli, M. B. Ferraro, D. Grillo, M. Habgood, D. W. M. Hofmann, F. Hofmann, K. V. J. Jose, P. G. Karamertzanis, A. V. Kazantsev, J. Kendrick, L. N. Kuleshova, F. J. J. Leusen, A. V. Maleev, A. J. Misquitta, S. Mohamed, R. J. Needs, M. A. Neumann, D. Nikylov, A. M. Orendt, R. Pal, C. C. Pantelides, C. J. Pickard, L. S. Price, S. L. Price, H. A. Scheraga, J. van de Streek, T. S. Thakur, S. Tiwari, E. Venuti, and I. K. Zhitkov, Acta Crystallogr., Sect. B: Struct. Sci. **67**, 535 (2011).

[39] A. M. Reilly, R. I. Cooper, C. S. Adjiman, S. Bhattacharya, A. D. Boese, J. G. Brandenburg, P. J. Bygrave, R. Bylsma, J. E. Campbell, R. Car, D. H. Case, R. Chadha, J. C. Cole, K. Cosburn, H. M. Cuppen, F. Curtis, G. M. Day, R. A. DiStasio, Jr., A. Dzyabchenko, B. P. van Eijck, D. M. Elking, J. A. van den Ende, J. C. Facelli, M. B. Ferraro, L. Fusti-Molnar, C.-A. Gatsiou, T. S. Gee, R. de Gelder, L. M. Ghiringhelli, H. Goto, S. Grimme, R. Guo, D. W. M. Hofmann, J. Hoja, R. K. Hylton, L. Iuzzolino, W. Jankiewicz, D. T. de Jong, J. Kendrick, N. J. J. de Klerk, H.-Y. Ko, L. N. Kuleshova, X. Li, S. Lohani, F. J. J. Leusen, A. M. Lund, J. Lv, Y. Ma, N. Marom, A. E. Masunov, P. McCabe, D. P. McMahon, H. Meekes, M. P. Metz, A. J. Misquitta, S. Mohamed, B. Monserrat, R. J. Needs, M. A. Neumann, J. Nyman, S. Obata, H. Oberhofer, A. R. Oganov, A. M. Orendt, G. I. Pagola, C. C. Pantelides, C. J. Pickard, R. Podeszwa, L. S. Price, S. L. Price, A. Pulido, M. G. Read, K. Reuter, E. Schneider, C. Schober, G. P. Shields, P. Singh, I. J. Sugden, K. Szalewicz, C. R. Taylor, A. Tkatchenko, M. E. Tuckerman, F. Vacarro, M. Vasileiadis, A. Vazquez-Mayagoitia, L. Vogt, Y. Wang, R. E. Watson, G. A. de Wijs, J. Yang, Q. Zhu, and C. R. Groom, Acta Crystallogr., Sect. B: Struct. Sci., Cryst. Eng. Mater. **72**, 439 (2016).

[40] M. A. Neumann, F. J. J. Leusen, and J. Kendrick, Angew. Chem., Int. Ed. **47**, 2427 (2008).

[41] R. Podeszwa, B. M. Rice, and K. Szalewicz, Phys. Chem. Chem. Phys. **11**, 5512 (2009).

[42] N. Marom, R. A. DiStasio, V. Atalla, S. Levchenko, A. M. Reilly, J. R. Chelikowsky, L. Leiserowitz, and A. Tkatchenko, Angew. Chem., Int. Ed. **52**, 6629 (2013).

[43] A. M. Lund, A. M. Orendt, G. I. Pagola, M. B. Ferraro, and J. C. Facelli, Cryst. Growth Des. **13**, 2181 (2013).

[44] A. Otero de la Roza and E. R. Johnson, J. Chem. Phys. **136**, 174109 (2012).

[45] A. M. Reilly and A. Tkatchenko, J. Chem. Phys. **139**, 024705 (2013).

[46] J. Moellmann and S. Grimme, J. Phys. Chem. C **118**, 7615 (2014).

[47] A. Nangia, Acc. Chem. Res. **41**, 595 (2008).

[48] S. L. Price, Acc. Chem. Res. **42**, 117 (2009).

[49] R. Sure and S. Grimme, J. Comput. Chem. **34**, 1672 (2013).

[50] R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, R. D. Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, J. Chem. Theory Comput. **13**, 3185 (2017).

[51] L. A. Burns, D. G. A. Smith, D. A. Sirianni, A. Alenaizan, Z. L. Glick, J. Lee, A. Lolinco, N. de Silva, and D. Matthews, Quantum Chemistry Common Driver and Databases, https://github.com/qcdb/qcdb, 2019.

[52] J. R. Hester, J. Appl. Crystallogr. **39**, 621 (2006).

[53] I. A. Kaliman and L. V. Slipchenko, J. Comput. Chem. **34**, 2284 (2013).

[54] I. A. Kaliman and L. V. Slipchenko, J. Comput. Chem. **36**, 129 (2014).

[55] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery, J. Comput. Chem. **14**, 1347 (1993).

[56] M. S. Gordon and M. W. Schmidt, *Theory and Applications of Computational Chemistry* (Elsevier, 2005), pp. 1167–1189.

[57] M. S. Gordon, M. A. Freitag, P. Bandyopadhyay, J. H. Jensen, V. Kairys, and W. J. Stevens, J. Phys. Chem. A **105**, 293 (2001).

[58] I. Adamovic and M. S. Gordon, Mol. Phys. **103**, 379 (2005).

[59] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms* (The MIT Press, 2009).

[60] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).

[61] L. A. Burns and D. G. A. Smith, QCElemental, https://github.com/MolSSI/QCElemental, 2019.

[62] W. Kabsch, Acta Crystallogr., Sect. A: Found. Adv. **32**, 922 (1976).

[63] W. Kabsch, Acta Crystallogr., Sect. A: Found. Adv. **34**, 827 (1978).

[64] H. W. Kuhn, Nav. Res. Logist. Q. **2**, 83 (1955).

[65] J. Munkres, J. Soc. Ind. Appl. Math. **5**, 32 (1957).

[66] T. Uno, *Algorithms and Computation*, Proceedings of the 8th International Symposium on Algorithms and Computation (ISAAC 97), Singapore, Singapore, December 17–19, 1997, Lecture Notes in Computer Science Vol. 1350, edited by H. W. Leong, H. Imai, and S. Jain (Department of Information Systems and Computer Science, National University of Singapore, 1997), pp. 92–101.

[67] E. Jones, T. Oliphant, P. Peterson *et al.*, SciPy: Open source scientific tools for Python, 2001, Online accessed 22 April 2019.

[68] Xunius, Bipartite matching, https://github.com/Xunius/bipartite_matching, 2017.
[69] B. M. Axilrod and E. Teller, J. Chem. Phys. **11**, 299 (1943).
[70] Y. Muto, J. Phys. Soc. Jpn. **17**, 629 (1943).
[71] S. Grimme, A. Hansen, J. G. Brandenburg, and C. Bannwarth, Chem. Rev. **116**, 5105 (2016).
[72] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, and W. A. De Jong, Comput. Phys. Commun. **181**, 1477 (2010).
[73] S. Grimme, J. G. Brandenburg, C. Bannwarth, and A. Hansen, J. Chem. Phys. **143**, 054107 (2015).
[74] G. E. Bacon, N. A. Curry, and S. A. Wilson, Proc. R. Soc. London, Ser. A **279**, 98 (1964).
[75] A. Otero-de-la Roza and E. R. Johnson, J. Chem. Phys. **137**, 054103 (2012).
[76] P. Valiron and I. Mayer, Chem. Phys. Lett. **275**, 46 (1997).
[77] S. F. Boys and F. Bernardi, Mol. Phys. **19**, 553 (1970).
[78] D. Feller, J. Chem. Phys. **96**, 6104 (1992).
[79] R. Podeszwa, J. Phys. Chem. A **112**, 8884 (2008).
[80] O. A. von Lilienfeld and A. Tkatchenko, J. Chem. Phys. **132**, 234109 (2010).
[81] J. F. Gonthier and M. Head-Gordon, J. Chem. Theory Comput. **15**, 4351 (2019).