



Interoperable Workflows by Exchanging Grid-Based Data between Quantum-Chemical Program Packages

Kevin Focke, Matteo de Santis, Mario Wolter, Jessica Martinez B, Valérie Vallet, André Severo Pereira Gomes, Malgorzata Olejniczak, Christoph Jacob

► To cite this version:

Kevin Focke, Matteo de Santis, Mario Wolter, Jessica Martinez B, Valérie Vallet, et al.. Interoperable Workflows by Exchanging Grid-Based Data between Quantum-Chemical Program Packages. The Journal of Chemical Physics, 2024, 160 (16), pp.162503. 10.1063/5.0201701 . hal-04431912

HAL Id: hal-04431912

<https://hal.science/hal-04431912>

Submitted on 30 Apr 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Interoperable Workflows by Exchanging Grid-Based Data between Quantum-Chemical Program Packages

Kevin Focke^{a,1}, Matteo De Santis^{b,2}, Mario Wolter^{a,3}, Jessica A.
Martinez B^{b,c,4}, Valérie Vallet^{b,5}, André Severo Pereira Gomes^{*,b,6},
Małgorzata Olejniczak^{*,d,7}, Christoph R. Jacob^{*,a,8}

^a Technische Universität Braunschweig, Institute of Physical and Theoretical Chemistry,
Gaußstraße 17, 38106 Braunschweig, Germany

^b Univ. Lille, CNRS, UMR 8523 – PhLAM – Physique des Lasers Atomes et Molécules,
F-59000 Lille, France

^c Department of Chemistry, Rutgers University, Newark, New Jersey, USA

^d Centre of New Technologies, University of Warsaw, S. Banacha 2c, 02-097 Warsaw, Poland

Date: March 29, 2024

Manuscript No. JCP24-SW-MIS2023-00521

Status: revised version, resubmitted to *J. Chem. Phys.*

(Special Topic “Modular and Interoperable Software for Chemical Physics”)

¹ORCID: 0009-0009-9079-1769

²ORCID: 0000-0001-7366-1780

³ORCID: 0000-0001-6948-6801

⁴ORCID: 0000-0002-9577-3449

⁵ORCID: 0000-0002-2202-3858

⁶ORCID: 0000-0002-5437-2251, E-Mail: andre.gomes@univ-lille.fr

⁷ORCID: 0000-0002-8370-9570, E-Mail: malgorzata.olejniczak@cent.uw.edu.pl

⁸ORCID: 0000-0002-6227-8476, E-Mail: c.jacob@tu-braunschweig.de

Abstract

Quantum-chemical subsystem and embedding methods require complex workflows that may involve multiple quantum-chemical program packages. Moreover, such workflows require the exchange of voluminous data that goes beyond simple quantities such as molecular structures and energies. Here, we describe our approach for addressing this interoperability challenge by exchanging electron densities and embedding potentials as grid-based data. We describe the approach that we have implemented to this end in a dedicated code, PYEMBED, currently part of a Python scripting framework. We discuss how it has facilitated the development of quantum-chemical subsystem and embedding methods, and highlight several applications that have been enabled by PYEMBED, including WFT-in-DFT embedding schemes mixing non-relativistic and relativistic electronic structure methods, real-time time-dependent DFT-in-DFT approaches, the density-based many-body expansion, and workflows including real-space data analysis and visualization. Our approach demonstrates in particular the merits of exchanging (complex) grid-based data, and in general the potential of modular software development in quantum chemistry, which hinges upon libraries that facilitate interoperability.

1 Introduction

Contemporary applications of quantum chemistry increasingly rely on complex workflows, which combine calculations with different methods and program packages. For instance, schemes for the automatic exploration of chemical reaction networks (see, e.g., Ref. 1–3) combine heuristics for the generation of possible reaction products and pathways,^{4–6} semi-empirical quantum chemistry for the generation of conformer ensembles,^{7,8} and quantum-chemical calculations at different levels of theory, possibly in combination with approaches for quantifying their uncertainty.^{9,10} Similarly, the elucidation of reaction mechanisms in homogeneous transition-metal catalysis requires intricate workflows combining distinct computational tools (for examples, see Ref. 11, 12). Complex computational workflows are further at the heart of high-throughput applications of quantum chemistry,^{13–16} which are often a prerequisite for leveraging the potential of machine learning in chemistry.^{17–22} Numerous software packages have been developed over the past two decades to facilitate such computational workflows in quantum chemistry.^{23–39}

The computational treatment of complex chemical systems such as biomolecules, molecular aggregates, or nanoscopic materials and interfaces is nowadays enabled by multiscale and multilevel methods of computational chemistry.^{40–43} Examples of such methods include QM/MM approaches,^{44–46} quantum-chemical fragmentation methods,^{47–49} ONIOM-type methods,^{50,51} and density-based subsystem and embedding methods.^{52–56} As these multiscale and multilevel methods combine rather distinct computational approaches for different parts or fragments of complex chemical systems, they commonly give rise to computational workflows that combine many individual calculations, possibly with different quantum-chemical software modules or packages — both for density-functional theory (DFT) and for various methods of wave-function theory (WFT) — and with other software tools, such as force-field codes or chemoinformatics toolkits. Furthermore, they might require exchanging data between these individual calculations beyond “simple” quantities

such as atomic coordinates or total energies. Several computational tools for realizing such multiscale workflows have been developed in the past decades for different use cases, in particular for QM/MM schemes^{57–63} and for fragmentation methods.^{64–66}

To realize the computational workflows that are required for density-based subsystem and embedding methods, some of us have initially developed the Python-based scripting framework PYADF.²³ While initially focused around the Amsterdam Density Functional (ADF) program⁶⁷ [now part of the Amsterdam Modeling Suite (AMS)], over the years PYADF has been extended to facilitate density-based subsystem and embedding methods in which different quantum-chemical methods and program packages can be combined.

In this context, the interoperability of different quantum-chemical program packages poses particular challenges. Commonly, quantum-chemical program packages (such as AMS,^{67,68} DALTON,^{69,70} DIRAC,⁷¹ NWCHEM,^{72,73} MOLCAS,^{74,75} ORCA,^{76,77} TURBOMOLE,^{78,79} and QUANTUMESPRESSO^{80,81}) have been developed as monolithic codes over many decades.⁸² Therefore, there is generally no easy access to the internals of the program packages, either because they are closed-source codes or because they are too complex for users and developers outside of their core community. However, density-based subsystem and embedding methods require exchanging information between different program packages and/or the workflow environment that go beyond those commonly provided to the user, such as integration grids or electron densities and derived properties. Unfortunately, there are no standardized interfaces for exchanging such information between different quantum-chemical program packages.

Here, we discuss how these challenges have been addressed in the PYADF scripting framework. To this end, we describe the PYEMBED module of PYADF for handling grid-based data (see Section 2) and demonstrate how it facilitates interoperable workflows in multi-scale quantum chemistry. To illustrate the capabilities of our approach, we discuss selected examples of workflows that either combine different program packages or that are agnostic

to the underlying quantum-chemical programs and methods. In particular, we highlight the development and applications of DFT-in-DFT and WFT-in-DFT embedding schemes (see Section 3.1), of real-time time-dependent DFT-in-DFT embedding schemes (see Section 3.2), and of the density-based many-body expansion (see Section 3.3). Furthermore, we discuss the integration of real-space data analysis and visualization into computational workflows (see Section 3.4). Note that the present article focuses on PYADF's functionality for handling grid-based data and does not aim at covering its full capabilities in other areas, which will be discussed elsewhere.

2 Challenges and General Design

The scripting framework PYADF allows the user to set up workflows combining several quantum-chemical calculations seamlessly. It provides classes for handling and manipulating atomic coordinates, for defining quantum-chemical calculations with a number of program packages (including ADF/AMS, DALTON, DIRAC, NWCHEM, MOLCAS, ORCA, and TURBOMOLE) as well as for generating input files for these programs, for executing the quantum-chemical calculations as well as for storing their results, and for extracting results data (such as optimized structures, energies, and properties) from the result files. The general design of PYADF is described in Ref. 23, even though its functionality has been extended significantly in the past decade.

2.1 Use Case: Density-Based Embedding

The prototypical use case that is of interest to us here are density-based subsystem and embedding calculations, in particular, DFT-in-DFT and WFT-in-DFT calculations based on the framework of subsystem-DFT and frozen-density embedding (FDE).^{52,54,56} Here, one considers a molecular system, which is divided into N subsystems. This is achieved

by partitioning the total electron density $\rho_{\text{tot}}(\mathbf{r})$ into the electron densities of these subsystems,

$$\rho_{\text{tot}}(\mathbf{r}) = \sum_I \rho_I(\mathbf{r}). \quad (1)$$

When considering the first subsystem as the subsystem of interest, its electron density $\rho_1(\mathbf{r})$ is obtained by minimizing the total energy functional. This results in Kohn-Sham (KS)-like equations for the active subsystem, in which the effect of the frozen environment of all the other subsystems (with electron densities $\rho_{J=2,\dots,N}$) is included via an effective embedding potential,⁸³

$$v_{\text{emb}}^{(1)}[\rho_1, \rho_{\text{tot}}](\mathbf{r}) = \sum_{I=2}^N v_{\text{nuc}}^{(I)}(\mathbf{r}) + v_{\text{Coul}}[\rho_{\text{tot}} - \rho_1](\mathbf{r}) + \frac{\delta E_{\text{xc}}[\rho_{\text{tot}}]}{\delta \rho_{\text{tot}}(\mathbf{r})} - \frac{\delta E_{\text{xc}}[\rho_1]}{\delta \rho_1(\mathbf{r})} + \frac{\delta T_s[\rho_{\text{tot}}]}{\delta \rho_{\text{tot}}(\mathbf{r})} - \frac{\delta T_s[\rho_1]}{\delta \rho_1(\mathbf{r})}. \quad (2)$$

Here, $v_{\text{nuc}}^{(I)}(\mathbf{r})$ is the potential of the nuclei in subsystem I , $v_{\text{Coul}}[\rho](\mathbf{r})$ is the Coulomb potential of electron density ρ , $E_{\text{xc}}[\rho]$ is the exchange–correlation functional, and $T_s[\rho]$ is the kinetic energy functional. The latter two are commonly evaluated using suitable approximate density functionals.

For DFT-in-DFT embedding calculations, this embedding potential is included in the KS equations. Note that it depends on the electron density of the active subsystem, i.e., it needs to be updated during the self-consistent field interactions. In WFT-in-DFT calculations, the embedding potential is included in the one-electron Hamiltonian, which requires evaluating the matrix elements,

$$V_{\mu\nu}^{\text{emb}} = \langle \chi_\mu | v_{\text{emb}} | \chi_\nu \rangle = \int \chi_\mu^*(\mathbf{r}) v_{\text{emb}}(\mathbf{r}) \chi_\nu(\mathbf{r}) d^3r, \quad (3)$$

where χ_μ are the basis functions. For further details on the inclusion of the FDE embedding potential in WFT calculations, see Refs. 40,84–87. Here, one commonly replaces the dependence of the embedding potential on the electron density of the active subsystem by a fixed density (linearized embedding potential⁸⁸).

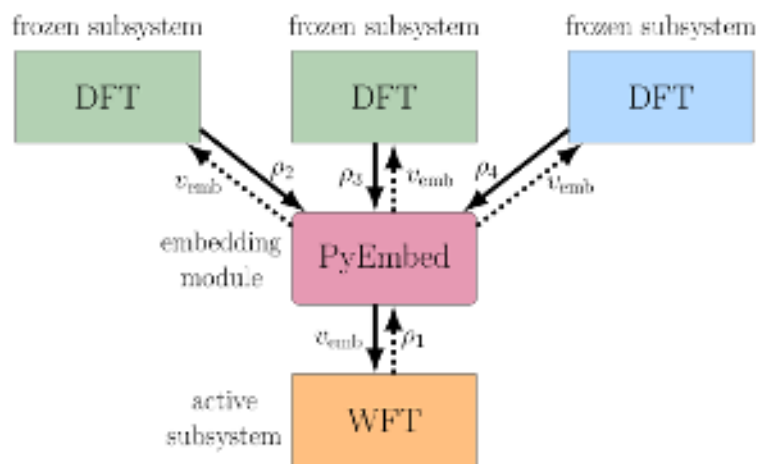


Figure 1: Illustration of a modular approach to WFT-in-DFT calculations. Each subsystem is treated in a separate quantum-chemical calculation (shown as rectangles), possibly using different program packages (indicated by the colors of the rectangles). The electron densities from the individual calculations serve as input to PYEMBED, which calculates the embedding potential that is subsequently imported into the WFT calculation of the active subsystem. In freeze-and-thaw calculations, the roles of the different subsystems can be interchanged (indicated by the dashed arrows).

Several quantum-chemical program packages provide integrated implementation of DFT-in-DFT embedding, e.g., ADF/AMS,⁸⁹ Serenity,^{90,91} CP2K,⁹² and eQE.^{93,94} These control the calculations for the different subsystems themselves and use their internal data structures to exchange data between them. In this case, algorithms and routines available within these programs can be used to evaluate the different terms of the embedding potential and their matrix elements. An integrated implementation of WFT-in-DFT methods is available in KOALA.⁹⁵

However, such an integrated approach makes it difficult to combine the unique capabilities of different program packages in a single DFT-in-DFT or WFT-in-DFT workflow. An alternative modular approach, which is facilitated by the PYEMBED module, is illustrated in Fig. 1. Here, DFT calculations for the frozen subsystems are performed individually, possibly using different methods and/or program packages for different frozen subsystems (see green and blue rectangles at the top of the figure). These calculations provide the electron densities of the subsystems, from which the PYEMBED module calculates

the embedding potential according to Eq. (2) (see central box), which is subsequently imported into the WFT calculation for the active subsystem, again using a different program package (see orange box at the bottom). In freeze-and-thaw calculations, roles of the subsystems can be interchanged, and the electron density from the WFT calculation (or an approximation thereof) can be used to obtain an embedding potential for the initially frozen subsystems, leading to updated frozen electron densities that can, in turn, be used to calculate an updated embedding potential for the WFT calculation.

Such a modular approach to DFT-in-DFT and WFT-in-DFT calculations poses a challenge, as it requires clearly defined interfaces between the quantum-chemical program packages and PYEMBED that facilitate interoperability. In particular, the quantum-chemical program packages, on the one hand, need to be able to export the electron density and related quantities, and on the other hand, must provide the capability to import an external embedding potential. In particular the latter will generally require modifying the quantum-chemical program packages themselves, but the required changes should be as lightweight as possible.

Here, we chose to address these challenges by exchanging the densities and potentials as grid-based data. That is, we employ a suitable three-dimensional grid of points $\{\mathbf{r}_i\}$, and express the electron densities and related quantities as well as embedding potentials on these grid points. Usually, the grid points coincide with those of a suitable numerical integration grid (e.g., a Becke grid⁹⁶), which makes it possible to evaluate the integrals over the embedding potential [Eq. (3)] by numerical integration:

$$V_{\mu\nu}^{\text{emb}} = \langle \chi_\mu | v_{\text{emb}} | \chi_\nu \rangle \approx \sum_i \chi_\mu^*(\mathbf{r}_i) v_{\text{emb}}(\mathbf{r}_i) \chi_\nu(\mathbf{r}_i). \quad (4)$$

To this end, the embedding potential is also evaluated at the individual grid points from the values of the electron densities and related properties at the grid points, which are exported from the quantum-chemical program packages on these grid points. In addition to the electron density itself, the first and second derivatives of the electron density at

the grid points are required to calculate the functional derivatives of $E_{xc}[\rho]$ and of $T_s[\rho]$ when using generalized-gradient approximations (GGAs), and the Coulomb potential of the electron densities,

$$v_{\text{Coul}}[\rho](\mathbf{r}) = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d^3r, \quad (5)$$

should also be provided on the grid points by the quantum-chemical program packages. Furthermore, the nuclear potential $v_{\text{nuc}}(\mathbf{r})$ at each grid point is required, but can be easily calculated directly from the atomic coordinates.

Exchanging grid-based data facilitates interoperability, as it defines a clear interface that is independent of the quantum-chemical methods that are used, of the employed basis functions and the specific basis sets, and of the inner workings of the individual quantum-chemical program packages. In the following, we will present the primary components of the PYEMBED module that allow for the exchange of grid-based data between quantum-chemical program packages.

While our focus here is on calculating embedding potentials using PYEMBED, it is worth noting that exchanging grid-based data can extend to more complex quantities. For instance, it can involve derivatives of the embedding potential, essential for computing kernel contributions to linear response properties.⁹⁷

2.2 Handling of Grid-Based Data

To facilitate the handling of grid-based quantities in Python, we have developed a library of `GridFunction` classes, which provides the required data structures (see Fig. 2). Simply speaking, a `GridFunction` contains references to the associated integration grid and the relevant data on these grid points.

Here, the integration grid is represented by an instance of a `Grid` class, which encaps-

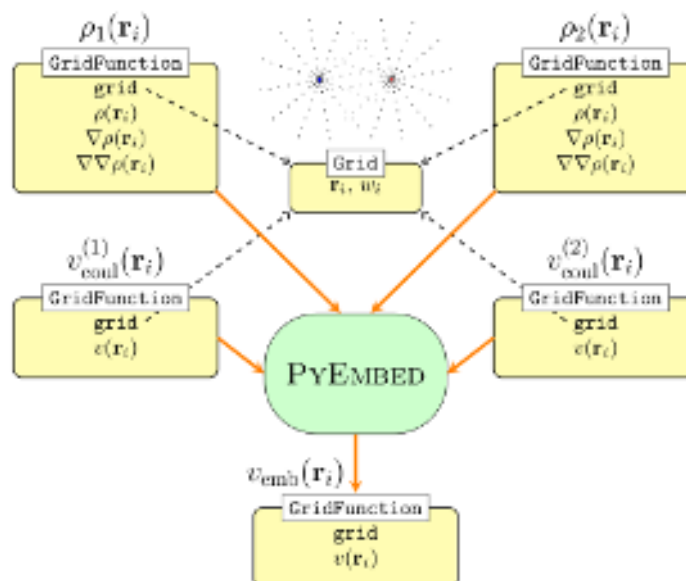


Figure 2: Overview of the general design of PYEMBED, for the example of the evaluation of the embedding potential $v_{\text{emb}}(\mathbf{r})$ from the fragments electron densities $\rho_1(\mathbf{r})$ and $\rho_2(\mathbf{r})$ as well as the corresponding Coulomb potentials $v_{\text{Coul}}^{(1)}(\mathbf{r})$ and $v_{\text{Coul}}^{(2)}(\mathbf{r})$. The grid-based representations of the densities and potentials are stored as instances of **GridFunction**, each referencing the associated **Grid**. See text for further details.

ulates the coordinates of the grid points as well as their weights. Different types of grids, such as evenly-spaced cubic grids (commonly used for visualization) and irregular molecular integration grids (as used in molecular quantum chemistry) are supported. PYEMBED interfaces with PySCF^{98,99} to generate suitable Becke integration grids based on the molecular structure, independent of external program packages. The accuracy of the generated Becke grid can be tuned according to the options available in PySCF.

Grid-based data (values on the grid points) are generally stored in NUMPY arrays. Different subclasses of **GridFunction** are tailored to specific common use cases. For instance, **GridFunctionDensityWithDerivatives** can encapsulate the electron density $\rho(r_i)$ (stored as one-dimensional NUMPY array of size N), its gradient $\nabla\rho(r_i)$ (stored as $N \times 3$ array), and if necessary its second derivative $\nabla\nabla\rho(r_i)$ (stored as $N \times 6$ array). The possibility to represent spin unrestricted densities is also available.

All `GridFunctions` provide methods for mathematical operations on them, such as addition and multiplication by scalars, as well as for numerical integration. These methods provide an interface that is unified across all types of grid functions, and are implemented efficiently via `NUMPY`. Furthermore, all `GridFunctions` include functionality for reading and writing the grid and data to file in various formats, including `NUMPY`'s^{100,101} text and binary formats, as well as `HDF5`.¹⁰² Finally, routines for interpolating grid-based quantities between different grids are also available.

2.3 Generation of Grid-Based Data

The available methods for reading `GridFunctions` from files provide a natural interface to external program packages and can be used to import electron densities and related quantities, provided the program packages allow for the export of grid-based data in a suitable file format. Currently, `PYADF` supports the import from `AMS`'s tape files as well as from text files written by `DALTON` and `DIRAC`. Implementing interfaces to other program packages and file formats as well as to other scalar, vector and tensor fields beyond density-related quantities (see Sect. 3.4) is straightforward.

Another approach for obtaining the values of the electron density and related quantities on the grid points proceeds via the molecular orbital coefficients optimized in the quantum-chemical calculation. Together with information on the basis functions used, this allows for the calculation of the electron density and related quantities inside `PYEMBED`. However, despite efforts to develop a standardized data format for the exchange of the required basis set and orbital / wavefunction information (such as `Q5Cost`,¹⁰³ `QCSchema`,¹⁰⁴ and `TREXIO`¹⁰⁵), only a few quantum-chemical program packages currently provide such a standardized interface. Further complications may arise if calculations take into account relativistic effects^{106,107} such as spin-orbit coupling,¹⁰⁸ since then one-electron functions may be expressed as complex-valued functions, or as quaternions.^{109,110} However, if one

remains within the non-relativistic theory, the file format used by the MOLDEN program¹¹¹ (originally developed several decades ago), provides a *de facto* standard, as such files can be obtained from most program packages.

Therefore, we have implemented a **DensityEvaluator** module that can process MOLDEN files generated by quantum-chemical program packages using Gaussian-type orbitals. This module makes use of PYSCF to read in the basis functions, molecular orbital coefficients, as well as occupation numbers and uses these to calculate the electron density, its derivatives, and the corresponding electrostatic potential on the points of an integration grid. These quantities are provided as **GridFunction** objects inside PYEMBED. We note that the use of such a non-standardized legacy file format can cause problems, in particular because different program packages implement the MOLDEN file format inconsistently. Further details are described in Ref. 112.

Currently, PYEMBED supports the generation of electron densities and related quantities via the **DensityEvaluator** MOLDEN file interface from DALTON, NWCHEM, ORCA, and TURBOMOLE. Extending this interface to modern, standardized file formats for the exchange of molecular orbital / wavefunction information^{103–105} will be straightforward once such interfaces are available in the respective quantum-chemical program packages.

2.4 Calculation of Embedding Potential

At the core of PYEMBED is the **EmbPot** module, which performs the calculation of the embedding potential as given in Eq. (2). In this module, the nuclear potential of the frozen subsystems is evaluated directly from the corresponding atomic coordinates, whereas the Coulomb potential of the frozen subsystems needs to be provided as input as a suitable **GridFunction** object. For evaluating the nonadditive kinetic and exchange–correlation potentials, it takes the electron density (and possibly its first and second derivatives) of

the active and frozen subsystems as input, again in the form of suitable `GridFunction` objects. To this end, the functional derivatives of the selected approximate functionals are evaluated with the help of the XCFUN library.^{113,114} For the nonadditive xc and kinetic energy functionals, both LDA functionals (requiring only the electron densities of the subsystems) as well as GGA functionals (additionally requiring the first and second derivatives of the subsystem electron densities) are currently supported. Note that for GGA functionals, the local embedding potential is evaluated explicitly, such that the matrix elements of the embedding potential can be evaluated directly according to Eq. (4) (i.e., the derivatives of the basis functions are not needed).

As a result, this provides a `GridFunction` representation of the embedding potential, which can be imported into the quantum-chemical calculation for the active subsystem. The evaluation of the matrix elements of the embedding potential [see Eq. (3)] by numerical integration generally needs to be performed inside the respective codes, which also need to provide an interface for importing this embedding potential from file. Such interfaces are currently publicly available in AMS/ADF, DALTON, DIRAC, NWChem, MOLCAS, and QUANTUMESPRESSO. We note that with the modular approach described here, the integration grid used inside the quantum-chemical program packages in the case of DFT calculations and those used for the evaluation of the embedding potential and for the evaluation of the matrix elements of the embedding potential according to Eq. (4) are generally different. Therefore, both grid have to be chosen appropriately.

3 Applications

3.1 DFT-in-DFT and WFT-in-DFT embedding

The exchange of grid-based data has been extensively employed by some of us in applications of WFT-in-DFT, employing a linearized embedding potential. In the case one is interested in energies for ground or excited states, the linearized scheme has been found to be rather accurate, since electron densities from WFT and DFT are often very similar.^{84,85,87}

One recent example is found in the simulations of valence binding energies of halides in water droplets,¹¹⁵ such as the one shown in Fig. 3a. In this work, the halide binding energies have been calculated with DIRAC using the relativistic equation of motion coupled cluster for ionization energies (EOM-IP), with the embedding potential being obtained after preparatory DFT-in-DFT calculations with ADF. Due to the need to account for finite temperature effects, these CC-in-DFT calculations have been carried out for a set of 100 snapshots originating from classical molecular dynamics simulations with polarizable force fields. With this setup, the overall computational cost for the calculation of each snapshot was roughly equivalent to that of the DFT-in-DFT calculations (instead of the much more computationally expensive EOM-IP calculation), while results for the halide binding energies were found to deviate from experiment by about 0.1 eV, a value comparable to that obtained with state of the art periodic Green's functions calculations.¹¹⁶

A similar protocol, based on an ensemble of preparatory DFT-in-DFT calculations followed by a WFT-in-DFT calculation, has been employed to obtain the core electron binding energies associated with the chlorine 2s and 2p levels for chloride in water droplets as well as chloride and HCl on ice surfaces (see Fig. 3b).¹¹⁷ The WFT-in-DFT calculations were again carried out with DIRAC, but now employing the core-valence separation (CVS)

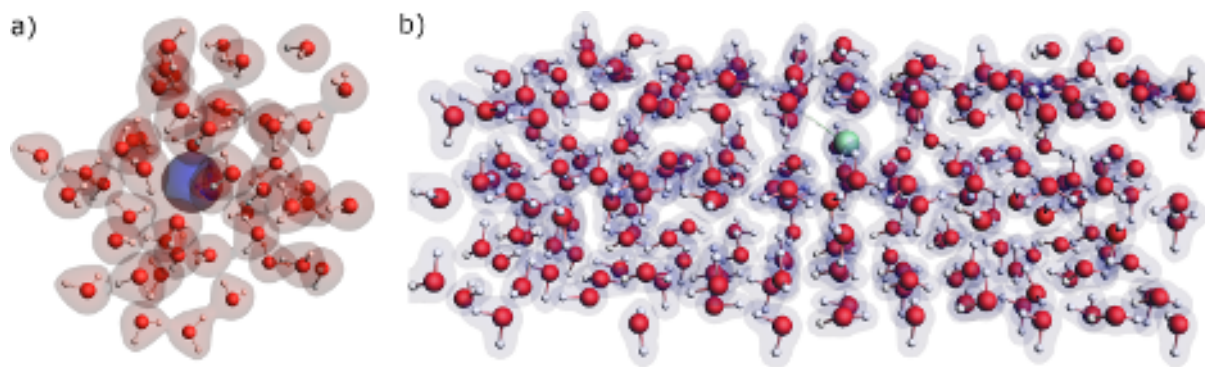


Figure 3: Structural models of (a) a halide in a 50-water droplet used in the WFT-in-DFT calculations reported in Ref. 115 (Figure available from <https://doi.org/10.5281/zenodo.1477102> under a CC-BY license); and (b) HCl adsorbed onto an amorphous ice surface used in the WFT-in-DFT calculations reported in Ref. 117 (Figure available from <https://zenodo.org/doi/10.5281/zenodo.10591687> under a CC-BY license). In both cases the highlighted volumes indicate the electron densities for the individual subsystems, obtained from the preparatory DFT-in-DFT calculations.

approximation to the EOM-IP problem. As with the valence electron binding energies, for chloride these simulations compared quite favorably to experiment in both environments. For HCl, on the other hand, preliminary findings point to the need to improve the structural models to accurately capture the predissociation tendency of HCl in contact with ice surfaces.

More recently, DFT-in-DFT calculations combining the ADF code and the four-component code BERTHA¹¹⁸ through the use of the PYBERTHA¹¹⁹ framework (which in turn employs PYEMBED as a backend for communicating with ADF) have been carried out to investigate the performance of FDE in describing the effects of solvation of gold clusters [for species such as $\text{Au}_8(\text{H}_2\text{O})_{10}$ or $\text{Au}_4(\text{H}_2\text{O})_{80}$] and of confinement on superheavy elements (Rn, Cn, Fl, Og) encapsulated in C_{60} molecules.¹²⁰ For the case of superheavy elements, it was found that embedding potentials were largely transferable between atoms, and provided a more physical description of the cage than the previously employed potentials consisting of a square well. A key implementation detail in PYBERTHA for achieving high efficiency is the expression of the embedding potential on the basis of fit functions, instead of orbital pairs.¹²⁰

The applications outlined above also serve to underscore that when different subsystems are treated with different Hamiltonians, exchanging grid-based data has a practical (if not fundamental) advantage over data exchange through orbital manipulations: since water molecules or C_{60} are made up of light elements for which spin-orbit coupling is not important, preparatory DFT-in-DFT calculations could safely be carried out with the scalar relativistic ZORA Hamiltonian (decreasing the overall cost of calculations), and the resulting embedding potential imported into four-component calculations without any particular difficulty. We are not aware of any other implementation that allows for such a mixture.

Grid-based data exchange has also been deployed for relativistic DFT-in-DFT calculations of response properties. In a recent application, the X-ray absorption spectra of the uranyl tetrachloride molecule ($Cs_2UO_2Cl_4$) has been obtained from the calculation of dipole polarizabilities within the damped response formalism,¹²¹ and in line with what had been found for valence excitations,⁸⁷ embedding calculations in which the chlorides are represented by an embedding potential can reproduce very well the spectral features for the uranium M_4 and L_3 edges, and the O K-edge.

Another example of DFT-in-DFT response calculations is that of magnetic properties such as NMR shieldings, indirect spin-spin coupling constants, and magnetizabilities.⁹⁷ In this case, the embedding was determined self-consistently at the four-component level, followed by that of perturbed densities, required for the calculation of kernel and coupling contributions which are essential for linear response properties.

It should be noted that the Fortran implementation in DIRAC closely followed the design philosophy of PYEMBED, with the definition of `GridFunction` containing grid-based data for particular subsystems, and operations that can be carried out on them. This makes for a natural connection to the PYADF and PYEMBED frameworks, and easily allows the exchange between subsystems (as DIRAC is not capable of handling multiple fragments

in the same execution).

The magnetic properties embedding code has been used to investigate the solvent effects on NMR shieldings of the molybdate (MnO_4^{2-}) ion,¹²² with temperature effects being taken into account by averaging the property calculations over 100 snapshots from AIMD simulations of molybdate with 20 water molecules.¹²³ One key finding from these studies is that despite the operators involved in the magnetic perturbations exhibiting an effective r^{-2} dependence (suggesting strong locality), the response properties still show non-negligible contributions from longer-range interactions with solvent molecules. These findings, along with the challenges in understanding the interplay between long-range and short-range contributions, have motivated some of us to pursue the development of real-space data analysis and visualization, discussed below.

3.2 Real-Time TDDFT-in-DFT Simulations

The applications of FDE discussed above for calculating molecular properties rely in one way or another on perturbation (response) theory. Apart from the fact that response theory will break down in the case of strong external perturbations (e.g., in the simulation of processes such as high harmonic generation), and that they are unable to provide detained information in the attosecond regime, a practical drawback of response theory is the need to calculate first or higher-order derivatives of the embedding potential.

An alternative approach lies in the so-called real-time (rt) methods, in which the time-dependent Schrödinger (or Dirac) equation is solved explicitly. In the case of Kohn-Sham DFT, this translates into

$$i \frac{\partial \psi_i(\mathbf{r}, t)}{\partial t} = \left(-\frac{\nabla_i^2}{2} + v_{\text{eff}}^{\text{KS}}(\mathbf{r}, t) + v_{\text{emb}}(\mathbf{r}, t) \right) \psi_i(\mathbf{r}, t) \quad (6)$$

where from time-dependent orbitals $\psi_i(\mathbf{r}, t)$ one obtains the time-dependent density matrix $\mathbf{D}_{\mu\nu}(t)$ and consequently the time-dependent electron density can be expressed as $\rho(\mathbf{r}, t) =$

$\sum_{\mu\nu} \mathbf{D}_{\mu\nu}(t) \chi_{\mu}(\mathbf{r}) \chi_{\nu}(\mathbf{r})$, with $\chi_{\mu}(\mathbf{r})$ being, for instance, atom-centered basis functions. This equation can be recast as the *Liouville-von Neumann* (LvN) equation^{124,125}

$$i \frac{\partial \mathbf{D}(t)}{\partial t} = \mathbf{F}(t) \mathbf{D}(t) - \mathbf{D}(t) \mathbf{F}(t), \quad (7)$$

with the time-dependent Fock matrix $\mathbf{F}(t)$ now includes the embedding potential. Molecular properties can then be obtained from a simulation of the time evolution of the dipole moment or another suitable operator, and this signal can be subsequently Fourier transformed to obtain for instance an excitation spectrum.

One consequence of this formulation for the density-based embedding theories,^{53,126} is that the embedding potential now formally becomes time-dependent, even if only subsystem I evolve in time (in analogy to the time-independent FDE case), due to the dependence of the embedding potential on both electron densities through the non-additive term (in what follows we only discuss results employing non-relativistic Hamiltonians, but note that a four-component code for rt-TDDFT-in-DFT calculations based on the PYBERTHA framework is currently under development).

In practice, recalculating the embedding potential at each time step, especially considering that typical simulations require thousands to tens of thousands of time steps, can be computationally very expensive and, in some cases, prohibitive. However, in a first implementation with atom-centered Gaussian basis sets employing the PYEMBED framework and combining the ADF and PSI4 codes, some of us have shown¹²⁶ that it is nevertheless possible to avoid recalculating the embedding potential over a certain (problem-dependent) number of time steps, and with that reduce the computational cost of rt-TDDFT-in-DFT FDE calculations, making it feasible to investigate the effect of the solvent on spectra with it.

In spite of the computational savings introduced in Ref. 126, calculations with relatively large frozen density region remained rather time-consuming, putting application to sys-

tems requiring large solvation shells (such as the halides discussed above) out of reach. At the same time, even for halides FDE had been shown to fail to properly describe the spectra of species such as fluoride,¹¹⁵ which interact strongly with their first solvation shell.

In order to overcome these two difficulties, we have recently introduced¹²⁷ an embedding rt-TDDFT implementation based on the PYEMBED and PYBERTHA-RT¹¹⁹ frameworks that combines the real-time^{128,129} Block-Orthogonalized Manby-Miller Embedding (BOMME)¹³⁰ and FDE¹²⁶ approaches. This hybrid approach aims to reduce computational cost further while accurately treating strong interactions in the vicinity of a species of interest (BOMME). Moreover, it can incorporate long-range interactions (FDE), which are crucial for describing excitation processes accurately.

We also explored a simple way to speed up calculations by taking advantage of the GPU offloading capabilities of optimized tensor libraries like PYTORCH and TENSORFLOW. Our early stage implementation, described in Ref. 127, involved replacing certain NUMPY operations, such as Fock matrix formation in time propagation, and constructing the matrix representation of the embedding potential, which are all performed in our own Python codes. Further work still needs to be done to fully realize the potential of GPU acceleration, in particular by enabling GPU offloading in more components of our implementation, as the generation of the non-additive potentials (`get_nadd_pot`) with PYEMBED.

The performance analysis summarized in Fig. 4, illustrates the impact of GPU offloading. In this figure we can see that the total wall time is reduced when going from the CPU-only implementation to that with PYTORCH, with a further reduction with TENSORFLOW. This is mostly due to a significant reduction in the portion of the total wall time spent on building the atomic orbital (AO) matrix representation of the embedding potential (`embpot2mat`), from 65.8% in the original CPU-based implementation (making it the bottleneck) to 22.6% for PYTORCH and further down to 1.8% for TENSORFLOW. With these

speedups and the lack of GPU offloading, `get_nadd_pot` then becomes the bottleneck for the tensor libraries, representing 61.4% of walltime for PYTORCH and 49.4% for TENSORFLOW. The percentage of time spent on Fock matrix formation (`mo_fock_mid_forward`) also becomes a bit more important for the tensor libraries, though they are rather similar (22.6% of walltime for PYTORCH and 25.9% for TENSORFLOW). For TENSORFLOW other routines such as `traceback_utils` start to become important for the timings, though at this stage we consider them as artifacts of profiling runs, and we intend to explore this matter further once the modifications to `get_nadd_pot` are carried out.

Taken together, these results provide a good first indication of the potential to speed up python-based scientific code by offloading costly operations onto GPU-aware frameworks, with minimal code changes in the scientific application.

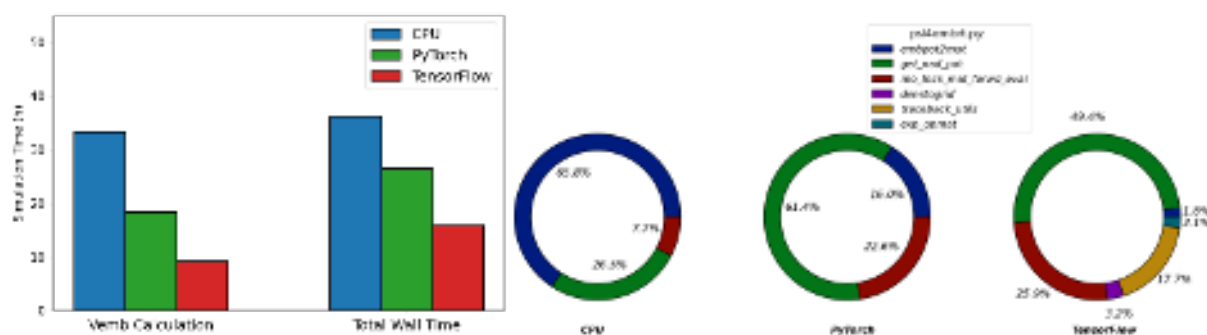


Figure 4: Comparison of the performance of rt-TDDFT-in-DFT simulations with and without GPU offloading employing the PYTORCH and TENSORFLOW frameworks. Left: comparison of total simulation time (wall time) required to carry out 56000 time steps, and the aggregate of time spent on evaluating the embedding potential. Right: breakdown of the time spent on the functions: `embpot2mat` calculates matrix elements of the embedding potential in the electronic structure code; `get_nadd_pot` calculates the embedding potential on the grid with PYEMBED, `mo_fock_mid_forward` extrapolates of Fock matrices during the propagation; `denstogrid` calculates the electron density on the grid. See Ref. 127 for a detailed discussion. The benchmarks were carried out using 64 OpenMP threads on an Intel(R) Xeon(R) Gold 6230R CPU @ 2.10GHz for the CPU runs, and 1 OpenMP thread (on the same type of CPU) and 1 NVIDIA A100-PCIE-40GB GPU for the GPU runs (Figure available from <https://zenodo.org/doi/10.5281/zenodo.10591687> under a CC-BY license).

In parallel to these developments, we have recently started to explore how far one can

take the idea of not updating the embedding potentials at each timestep. To this end, in Fig. 5 we present results of a simulation revisiting the acetone in water system of Ref. 126 in which we compare two limiting cases: the update of the embedding potential at each timestep as done previously, and the opposite situation in which the embedding potential is not updated at all during the time propagation. These preliminary results seem to indicate that for this system, which is a case of relatively weak interactions between the active system and its environment, this very crude approximation yields results which are quite close to the original ones. This point requires further investigations, as it may be that this approximation will be suitable for a few low-lying transitions but break down for high energy transitions such as those in the X-ray region.

In the case of heavy atoms (Rn, Cn, Fl, Og) encapsulated in C_{60} , discussed above for time-independent calculations, we observed the transferability of the embedding potential. We explored here whether that could hold true also for time-dependent case, by considering trifluoroacetone, which can be regarded as a modified acetone. We calculated its absorption spectrum utilizing embedding potentials without updates (corresponding to the assumption of weak interactions with the environment for the fluorinated species), one obtained for the acetone in water system, and another calculated for the trifluoroacetone in water system. As shown in Fig. 5b, the spectra for the two calculations show a semi-quantitative agreement. The conditions of transferability of embedding potentials in the case of confined molecular systems will be addressed in future works.

3.3 Density-Based Many-Body Expansion

Another example that has been enabled by the possibility to import and process grid-based quantities inside PYEMBED is the development of the density-based many-body expansion (db-MBE) by some of us.^{112,131,132} The db-MBE offers an accurate and efficient fragmentation method for the quantum-chemical treatment of molecular clusters. Its

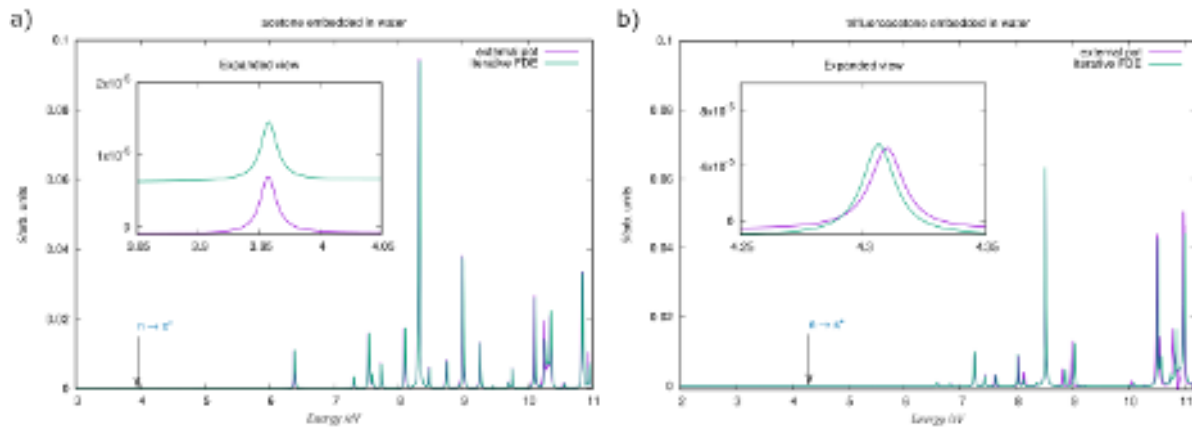


Figure 5: Absorption spectra (a) of acetone in water and obtained using the time-dependent embedding potential (green) evaluated at each time step (denoted as iterative FDE) with the same simulation setup as in Ref. 126. The violet trace (“external pot”) has been obtained here for simulations that closely match those in Ref. 126, with the difference that there is no update of the embedding potential using the time propagation; and (b) of trifluoroacetone in water, embedded in external static potentials. We show in violet the spectrum obtained with a potential originating from a calculation of acetone in water, and in green the spectrum obtained with the with an embedding potential determined for trifluoroacetone in water.

starting point is the conventional many-body expansion (MBE),^{48,133,134} in which the total energy of a molecular cluster is approximated as [energy-based many-body expansion (eb-MBE)],

$$E_{\text{tot}} \approx E_{\text{eb-MBE}}^{(n)} = \sum_I E_I^{(1)} + \sum_{I < J} \Delta E_{IJ}^{(2)} + \cdots + \sum \Delta E_{IJK\dots}^{(n)}, \quad (8)$$

where $E_I^{(1)}$ is the energy of the I -th fragment, $\Delta E_{IJ}^{(2)} = E_{IJ}^{(2)} - E_I^{(1)} - E_J^{(1)}$ is the interaction energy for the dimer of fragments I and J , and so forth.

The db-MBE seeks to improve upon the eb-MBE by calculating the density-based correction,

$$E_{\text{db-corr}}^{(n)} = E_{\text{tot}} \left[\rho_{\text{tot}}^{(n)}(\mathbf{r}) \right] - E_{\text{tot}}^{(n)}, \quad (9)$$

where

$$\rho_{\text{tot}}(\mathbf{r}) \approx \rho_{\text{tot}}^{(n)}(\mathbf{r}) = \sum_I \rho_I^{(1)}(\mathbf{r}) + \sum_{I < J} \Delta \rho_{IJ}^{(2)}(\mathbf{r}) + \cdots + \sum \Delta \rho_{IJK\dots}^{(n)}(\mathbf{r}), \quad (10)$$

is the many-body expansion of the total electron density, $E_{\text{tot}}[\rho]$ is the DFT total energy functional, and $E_{\text{tot}}^{(n)}$ is its many-body expansion. With this density-based correction, an improved n -body approximation to the total energy of the molecular cluster is obtained as $E_{\text{db-MBE}}^{(n)} = E_{\text{eb-MBE}}^{(n)} + E_{\text{db-corr}}^{(n)}$. Previous studies have demonstrated the excellent accuracy of db-MBE in predicting both absolute and relative energies of water and ion-water clusters, even at the two-body expansion level.^{132,135}

At first order (i.e., using only the electron densities of the monomers), the density-based correction can be calculated as

$$\begin{aligned}
 E_{\text{corr}}^{(1)} = & \sum_{I \neq J} \int \rho_I^{(1)}(\mathbf{r}) v_{\text{nuc}}^{(J)}(\mathbf{r}) d^3r + \sum_{I \neq J} \frac{1}{2} \int \rho_I^{(1)}(\mathbf{r}) v_{\text{Coul}}[\rho_J^{(1)}](\mathbf{r}) d^3r \\
 & + E_{\text{xc}}^{\text{nadd}}[\{\rho_I^{(1)}\}] + T_s^{\text{nadd}}[\{\rho_I^{(1)}\}] + \sum_{I < J} E_{\text{NN}}^{(IJ)},
 \end{aligned} \tag{11}$$

where the non-additive kinetic energy and exchange–correlation functionals

$$T_s^{\text{nadd}}[\{\rho_I^{(1)}\}] = T_s[\sum_I \rho_I^{(1)}] - \sum_I T_s[\rho_I^{(1)}] \tag{12}$$

$$E_{\text{xc}}^{\text{nadd}}[\{\rho_I^{(1)}\}] = E_{\text{xc}}[\sum_I \rho_I^{(1)}] - \sum_I E_{\text{xc}}[\rho_I^{(1)}], \tag{13}$$

are evaluated using suitable approximate density functionals, and where $E_{\text{NN}}^{(IJ)}$ is the interaction energy between the nuclei of fragments I and J .

With representations of the monomers' electron densities (as well as their derivatives) and nuclear and Coulomb potentials available, all terms in Eq. (11) can be evaluated using numerical integration.¹³¹ We have integrated a db-MBE module in PYADF that makes use of the PYEMBED module as described above and evaluates the density-based correction independent of external quantum-chemical program packages (see top part of Fig. 6). Again, the density functionals are evaluated using the XCFUN library.^{113,114}

For higher orders ($n > 1$), the density-based correction is given by

$$\begin{aligned}
 E_{\text{corr}}^{(n)} = & \left(V_{\text{nuc}}[\rho_{\text{tot}}^{(n)}] - V_{\text{nuc}}^{(n)} \right) + \left(J[\rho_{\text{tot}}^{(n)}] - J^{(n)} \right) \\
 & + T_s^{\text{nadd},(n)}[\{\rho_I\}, \{\rho_{IJ}\}, \dots] + E_{\text{xc}}^{\text{nadd},(n)}[\{\rho_I\}, \{\rho_{IJ}\}, \dots],
 \end{aligned} \tag{14}$$

which is also evaluated within the db-MBE module of PYADF using the corresponding n -mer densities and potentials.

The nuclear attraction and Coulomb energy functionals, $V_{\text{nuc}}[\rho]$ and $J[\rho]$ are calculated for the n -body expansion of the electron density [cf. Eq. (10)] using numerical integration, where the Coulomb energy is evaluated by integrating the electron density with the corresponding Coulomb potential. To this end, the many-body expansions of the electron density and of the Coulomb potential are assembled inside the db-MBE module of PYADF from the grid-based representations of the individual n -mer densities and potentials.

In Eq. (14), $V_{\text{nuc}}^{(n)}$ and $J^{(n)}$ refer to the many-body expansion of the nuclear attraction and Coulomb energies, respectively, are defined in analogy to Eq. (8). The individual energy contributions required for these terms can either be extracted from the underlying quantum-chemical calculations for the n -mers, or can be evaluated by numerical integration in the db-MBE module from the electron densities and Coulomb potentials obtained from each of the n -mer calculation. Finally, the n -body nonadditive kinetic and exchange–correlation energy functionals, $T_s^{\text{nadd},(n)}$ and $E_{\text{xc}}^{\text{nadd},(n)}$ are defined in analogy to Eq. (12) and (13) (see Ref. 131 for details) and are evaluated using XCFUN.

Our implementation of the db-MBE, facilitated by PYADF and PYEMBEd, exemplifies the modular development of novel quantum-chemical fragmentation methods. The db-MBE requires the calculation of a density-based correction, which depends not only on energies from the quantum-chemical calculations for the individual fragments, but also on the resultant electron densities. While such computations typically fall within the purview of quantum-chemical software packages, they can be realized within an external Python module through the grid-based exchange of electron densities and related quantities.

Given that the db-MBE module solely requires grid-based representations of the electron density and the Coulomb potential as its input (besides the atomic coordinates

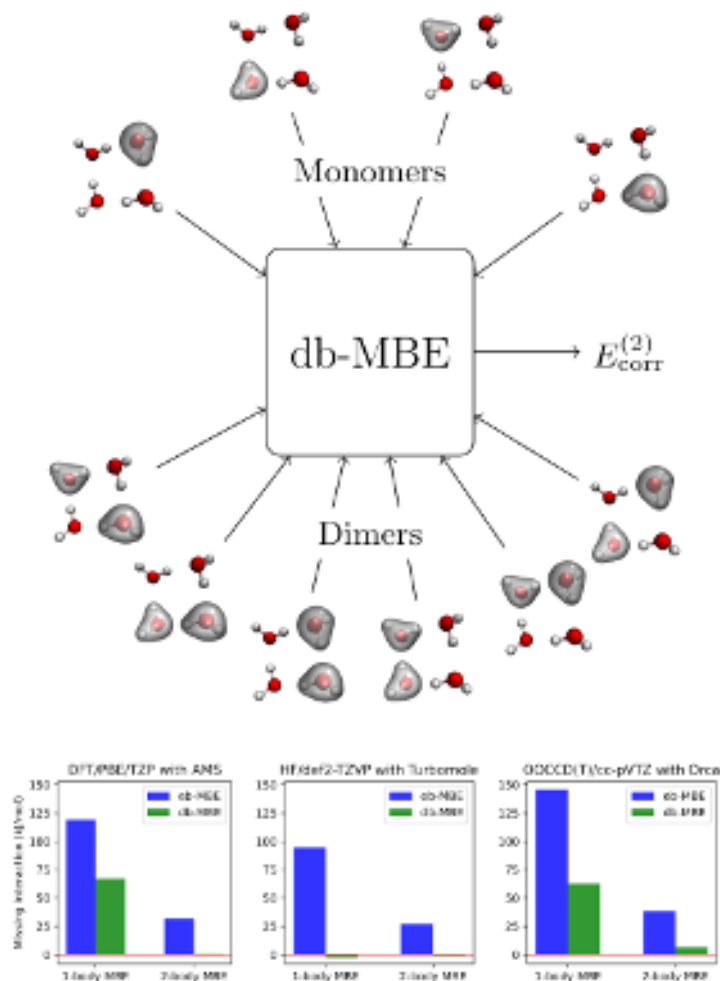


Figure 6: Top: Illustration of the db-MBE for the example of a water tetramer. The electron densities and related quantities are fed into the db-MBE module of PYADF as grid-based data. The db-MBE calculation uses these to calculate the density-based correction energy $E_{\text{corr}}^{(2)}$. Bottom: Error in interaction energy obtained for the eb-MBE and db-MBE at first and second order, using fragment calculations with three different quantum-chemical methods and program packages. The calculations have been performed as described in Ref. 112.

of the fragments), it operates independently of the specific type of underlying quantum-chemical calculations. Therefore, the db-MBE can take densities generated using different quantum-chemical methods in different program packages (see bottom part of Fig. 6. This versatility has enabled the extension of the db-MBE from DFT calculations to coupled-cluster methods.¹¹²

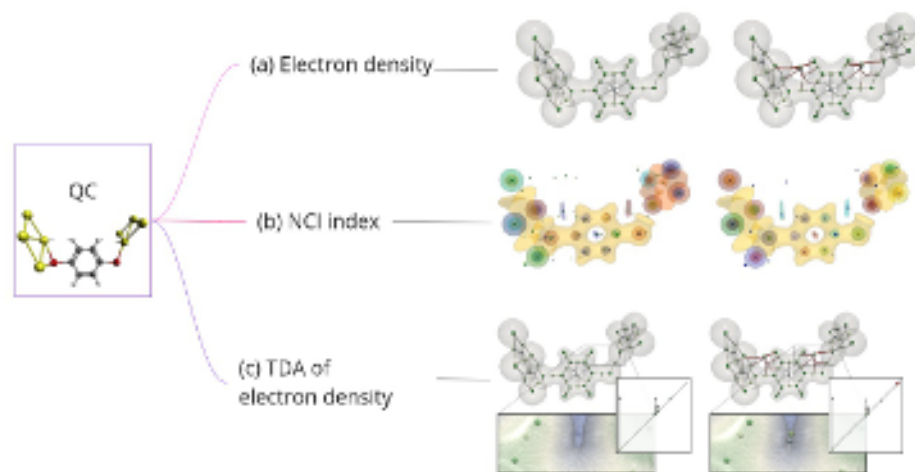


Figure 7: The selection of the correct molecular descriptor and the analysis method can significantly affect the analysis results: The topological analysis of the electron density of the $\text{Au}_4\text{-S-C}_6\text{H}_4\text{-S'-Au}'_4$ complex (a) by QTAIM and TDA suggests relativity-induced non-covalent interactions between Au and H atoms (manifested by the absence/presence of the bond critical points in the non-relativistic/relativistic case, respectively, demonstrated on the left and right figures). Their further inspection with the TDA tools (c) shows that these bond critical points are insignificant, and their importance is further reduced with the improvement of the quantum chemistry model. A better molecular descriptor for uncovering weak interactions is the NCI index (b), which unambiguously points to the presence of the bond critical points irrespective of the relativity level applied in QM calculations. Example after Ref. 139.

3.4 Real-Space Data Analysis and Visualization

Another important area of application for workflows that involve the exchange of grid-based data is the analysis and visualization of quantum-chemical calculation. Such combined methods can be particularly insightful — they adopt the “give us insight (not) and numbers” mindset,^{136–138} which has been driving the development of numerous analysis tools to accompany quantum-chemical calculations.

Specifically, we highlight techniques applied to *molecular descriptors*, which can be represented as scalar, vector, or tensor fields and discretized on grids in physical space. Such molecular descriptors can be provided by quantum-chemical program packages by the interfaces described above (passed as `GridFunctions`, see Fig. 7), and if necessary, the

modules described in Section 2 facilitate their processing, manipulation, and export to software tools for their analysis and manipulation.

The most obvious example of such a molecular descriptor is the electron density, $\rho(\mathbf{r})$. Plotting the electron density in real space as, for instance, a three-dimensional surface for a selected isovalue, can demonstrate the “shape” of a molecule or a specific molecular fragment (depending on the isovalue), point to nuclear positions and identify chemical bonds (see Fig. 7a). More elaborated electron density functions designed to capture specific information can highlight a particular feature of the molecular system. For instance, the electron density Laplacian $\Delta\rho(\mathbf{r})$ identifies electron localization domains.¹⁴⁰ The reduced density gradient, $s(\mathbf{r}) \propto |\nabla\rho(\mathbf{r})|/\rho(\mathbf{r})$, captures weak non-covalent interactions much better than the electron density. For this reason, its visualization (popularized as the NCI index^{141,142}) often complements electron density plots in studies on chemical interactions (see Fig. 7b). Such an information-centric approach to chemical simulations bloomed into the conceptual DFT theory¹⁴³ and Quantum Chemistry Topology (QCT)¹⁴⁴ domains, which largely focus on developing such molecular descriptors.

Efficient visualization techniques for such fields can be broadly classified as geometrical or feature-based approaches (for more detailed classification schemes, see, e.g., Ref. 145). Geometrical approaches are based on plotting the data either directly on the grids (as points or vectors) or by various functions that represent the data, for instance, via contours (scalar data) or integral curves (vector data). However, while these methods illustrate the underlying data intuitively and efficiently, they “transfer the burden of feature recognition to human brains”.¹⁴⁶ Moreover, such methods come with various difficulties, which emerge especially for more complex data, such as the selection of meaningful isosurfaces (scalar data) or the choice of seeding points for streamlines (vector data), to name but a few. The other group of methods focuses on extracting specific features of the data, which are independent of the choice of their visual representation and, therefore, can lead to

more reliable qualitative conclusions. An example is the topology-based visualization,¹⁴⁷ which, for instance, applied to a scalar field, results in the *topological skeleton* of the field composed of the critical points and separatrices (lines and planes “joining” these points).

Such topological methods have been widely adopted in computational chemistry. In particular, the whole QCT domain is based on the analysis of scalar and vector fields through their gradient fields. Specifically, it applies the theory of dynamical systems to the studied field gradient and interprets the extracted topological objects (e.g., critical points) in terms of chemical concepts (e.g., chemical bonds). The well-known example of a QCT method is the topological analysis of the electron density in molecular systems, pioneered by Bader and popularized as the Quantum Theory of Atoms in Molecules (QTAIM).¹⁴⁸ However, the practical applications of QCT methods to chemistry problems are often burdened with technical difficulties, and their value is reduced by ambiguous interpretations.

Modern analysis approaches, such as the Topological Data Analysis (TDA), can address many of such challenges. TDA is a new research field at the interface between mathematics and computer science, rooted in sound theoretical settings such as Morse theory¹⁴⁹ and Persistent Homology (PH).¹⁵⁰ TDA constitutes an appealing framework for extracting the topological features in the data, measuring their salience (what allows for discriminating important features from non-essential ones), and for multi-scale data analysis (for examples from chemistry, see Refs. 151–156). As such, it is particularly useful to study the manifestation of weak phenomena in the data representing both weak and strong physical features. A good illustration of this aspect is the analysis of non-covalent interactions in small gold complexes, $\text{Au}_4\text{-S-C}_6\text{H}_4\text{-S'-Au}'_4$, based on the presence of bond critical points of the electron density in the region between interacting atoms: while QTAIM attributed their appearance to the relativistic effects,¹⁵⁷ TDA demonstrated that these critical points are non-essential electron density features.¹³⁹

If the analysis outcome remains inconclusive, a good strategy is to reconsider the selection

of a molecular descriptor for the studied phenomenon — in the cited case, further studies employing the reduced density gradient scalar field confirmed the presence of non-covalent interactions between Au–H irrespective of the inclusion of relativistic effects in the quantum chemistry model (Fig. 7c). The choice of an appropriate function for analysis is also crucial for understanding specific features of descriptors represented by higher-rank tensor fields. For example, the extraction of axial and toroidal vortices in the “omega” scalar field derived from the magnetically-induced current density tensor by TDA tools¹⁵⁸ serves as a good illustration. While the primary focus of real-space analysis revolves around data visualization, other aspects such as quantification (e.g., through data integration), space partitioning, and data clustering can now also be efficiently addressed by TDA tools and libraries implementing them, such as the Topological Toolkit (TTK).^{159,160}

4 Conclusions and Outlook

Realizing complex workflows in quantum chemistry that involve different quantum-chemical program packages presents a challenge, particularly in the context of quantum-chemical subsystem and embedding methods. Quantum chemistry software packages are typically developed as stand-alone codes that prioritize numerical efficiency, accuracy, and adaptability to parallel architectures, but they often lack standardized interfaces for interoperability. However, achieving interoperability requires the seamless exchange of complex data between different program packages.

In this context, we have introduced an approach implemented in the PYEMBED module of the PYADF scripting framework to facilitate the exchange of electron densities, related properties, and embedding potentials between various quantum-chemical program packages. We have opted to utilize grid-based data and developed Python modules specifically for this purpose. Leveraging grid data for information exchange significantly simplifies

the interfaces between program packages, as it is agnostic to internal data structures and technical intricacies such as basis functions and basis sets. The PYEMBED module outlined here establishes clear data and file formats for grid-based information exchange, streamlining the integration of program packages into workflows to the creation of clean interfaces that furnish the necessary grid-based data.

While grid-based data simplifies data exchange, standardization of data and file formats, such as adopting a standardized format based on HDF5, would greatly enhance the broader acceptance of the modular approach outlined here. Furthermore, establishing clear and standardized interfaces in quantum-chemical program packages for results data, including molecular orbital coefficients, will further improve interoperability. In the approach presented here, utilizing the MOLDEN file format, despite its antiquated nature and potential limitations in meeting modern software development needs, offers the advantage of enabling the generation of grid-based data outside of quantum-chemical program packages, thus allowing this process to be performed within PYEMBED. This flexibility enhances the versatility of the modular approach and streamlines the integration of grid-based data into quantum-chemical workflows.

The Python modules discussed here are presently incorporated into the PyADF scripting framework but are also designed to be utilized independently. We intend to release PYEMBED as a standalone library in the near future. Additionally, we plan to introduce support for GPU offloading using machine learning frameworks such as PYTORCH or TENSORFLOW. This integration has the potential to yield substantial speed improvements, particularly when processing grid-based data, aligning with our recent efforts to accelerate rt-TDDFT-in-DFT calculations.¹²⁷

We have showcased numerous applications made possible by interoperable workflows that rely on the exchange of grid-based data between quantum-chemical program packages. Primarily, PYEMBED facilitates DFT-in-DFT and WFT-in-DFT embedding workflows by

integrating various program packages, thereby granting access to a wide array of quantum-chemical methods. The modular nature of this approach is especially well-suited for implementing customized freeze-and-thaw schemes involving different program packages.

In a broader sense, PYEMBED streamlines the rapid prototyping of subsystem and embedding methods. This capability has been exemplified in the development of real-time time-dependent density functional theory (rt-TDDFT) embedding schemes, where different program packages must interact at each time step. Similarly, the creation of the density-based many-body expansion (db-MBE) has been made possible solely through the modular and interoperable approach outlined here. These instances signify a larger paradigm shift, wherein the innovation of novel quantum-chemical methods shifts away from monolithic codes to lightweight Python scripts that manipulate and amalgamate externally provided data. We anticipate that this evolving paradigm will gain broader acceptance in the years to come (see also Refs. 82,161).

Finally, the utilization of grid-based data streamlines the incorporation of data analysis and visualization tools into quantum chemistry workflows. This facilitates the development of customized workflows that seamlessly integrate analysis tools to adjust the computational model as required, akin to the modeling cycle proposed in Ref. 162. These adaptive workflows will need to encompass steps related to evaluating the accuracy of models and data, which are pivotal for their optimization and refinement. The integration of real-space data analysis could enhance numerous stages of such efficient modeling cycles.

The integration of grid-based data, including descriptors utilized in QSAR/QSAP¹⁶³ or from conceptual DFT,¹⁴³ into computational workflows also presents new opportunities for high-throughput applications. These applications can serve as the foundation for machine learning in computational chemistry.^{164,165}

Data Availability

The most recent version of the PYADF scripting framework, including the PYEMBED module are the related functionality described in this article, is available at <https://github.com/chjacob-tubs/pyadf-releases/>. Code and data for the applications reviewed here are available in the cited publications.

Author Contributions

Kevin Focke: Methodology (equal), Software (equal), Visualization (lead), Writing – Original Draft (supporting), Writing – Review and Editing (equal). **Matteo De Santis:** Methodology (equal), Software (equal), Visualization (supporting), Writing – Original Draft (supporting), Writing – Review and Editing (equal). **Mario Wolter:** Methodology (equal), Software (supporting), Supervision (supporting), Writing – Review and Editing (equal). **Jessica Martinez:** Methodology (equal), Software (equal), Visualization (supporting), Writing – Review and Editing (equal). **Valérie Vallet:** Conceptualization (supporting), Supervision (supporting), Methodology (supporting), Writing – Review and Editing (equal). **André Severo Pereira Gomes:** Conceptualization (lead), Methodology (supporting), Software (equal), Supervision (equal), Writing – Original Draft (lead), Writing – Review and Editing (equal). **Małgorzata Olejniczak:** Conceptualization (lead), Methodology (supporting), Software (equal), Supervision (equal), Writing – Original Draft (lead), Writing – Review and Editing (equal). **Christoph R. Jacob:** Conceptualization (lead), Methodology (supporting), Software (equal), Supervision (equal), Writing – Original Draft (lead), Writing – Review and Editing (equal).

Conflicts of Interest

The authors have no conflicts to disclose.

Acknowledgments

C.R.J. and A.S.P.G. gratefully acknowledge funding from the Franco–German project CompRIXS (Agence nationale de la recherche ANR-19-CE29-0019, Deutsche Forschungsgemeinschaft JA 2329/6-1). C.R.J. and M.W. acknowledge funding from the Deutsche Forschungsgemeinschaft for the development of PYADF (Project Suresoft, JA 2329/7-1). M.D.S., V.V., and A.S.P.G acknowledge funding from projects CPER WaveTech, Labex CaPPA (Grant No. ANR-11-LABX-0005-01), the I-SITE ULNE project OVERSEE and MESONM International Associated Laboratory (LAI) (Grant No. ANR-16-IDEX-0004), as well support from the French national supercomputing facilities (Grant Nos. DARI A0130801859, A0110801859). J.A.M.B. acknowledges support from the Chateaubriand Fellowship of the Office for Science & Technology of the Embassy of France in the United States, and the National Science Foundation under grants No. CHE-2154760 and OAC-2117429. M.O. acknowledges support from the Polish National Science Centre (NCN) (grant number 2020/38/E/ST4/00614).

References

- [1] A. L. Dewyer, A. J. Argüelles, and P. M. Zimmerman, *Methods for exploring reaction space in molecular systems*, WIREs Comput. Mol. Sci. **8**, e1354 (2018).
- [2] J. P. Unsleber and M. Reiher, *The Exploration of Chemical Reaction Networks*, Annu. Rev. Phys. Chem. **71**, 121 (2020).
- [3] A. Baiardi, S. A. Grimm, M. Steiner, P. L. Türtcher, J. P. Unsleber, T. Weymuth, and M. Reiher, *Expansive Quantum Mechanical Exploration of Chemical Reaction Paths*, Acc. Chem. Res. **55**, 35 (2022).

- [4] D. Rappoport, C. J. Galvin, D. Y. Zubarev, and A. Aspuru-Guzik, *Complex Chemical Reaction Networks from Heuristics-Aided Quantum Chemistry*, J. Chem. Theory Comput. **10**, 897 (2014).
- [5] M. Bergeler, G. N. Simm, J. Proppe, and M. Reiher, *Heuristics-Guided Exploration of Reaction Mechanisms*, J. Chem. Theory Comput. **11**, 5712 (2015).
- [6] D. Rappoport and A. Aspuru-Guzik, *Predicting Feasible Organic Reaction Pathways Using Heuristically Aided Quantum Chemistry*, J. Chem. Theory Comput. **15**, 4099 (2019).
- [7] C. Bannwarth, S. Ehlert, and S. Grimme, *GFN2- xTB —An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions*, J. Chem. Theory Comput. **15**, 1652 (2019).
- [8] P. Pracht, F. Bohle, and S. Grimme, *Automated exploration of the low-energy chemical space with fast quantum chemical methods*, Phys. Chem. Chem. Phys. **22**, 7169 (2020).
- [9] G. N. Simm and M. Reiher, *Error-Controlled Exploration of Chemical Reaction Networks with Gaussian Processes*, J. Chem. Theory Comput. **14**, 5238 (2018).
- [10] J. Proppe and M. Reiher, *Mechanism Deduction from Noisy Chemical Reaction Networks*, J. Chem. Theory Comput. **15**, 357 (2019).
- [11] M. Podewitz, S. Sen, and M. R. Buchmeiser, *On the Origin of E-Selectivity in the Ring-Opening Metathesis Polymerization with Molybdenum Imido Alkylidene N-Heterocyclic Carbene Complexes*, Organometallics **40**, 2478 (2021).
- [12] R. A. Talmazan, J. Refugio Monroy, F. del Río-Portilla, I. Castillo, and M. Podewitz, *Encapsulation Enhances the Catalytic Activity of C-N Coupling: Reaction Mechanism of a Cu(I)/Calix[8]arene Supramolecular Catalyst*, ChemCatChem **14**, e202200662 (2022).
- [13] A. Jain et al., *Commentary: The Materials Project: A materials genome approach to accelerating materials innovation*, APL Mater. **1**, 011002 (2013).
- [14] J. Hachmann et al., *Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry – the Harvard Clean Energy Project*, Energ. Environ. Sci. **7**, 698 (2014).
- [15] E. O. Pyzer-Knapp, C. Suh, R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, and A. Aspuru-Guzik, *What Is High-Throughput Virtual Screening? A Perspective from Organic Materials Discovery*, Annu. Rev. Mater. Res. **45**, 195 (2015).

- [16] J. P. Unsleber et al., *High-throughput ab initio reaction mechanism exploration in the cloud with automated multi-reference validation*, J. Chem. Phys. **158**, 084803 (2023).
- [17] H. J. Kulik, *Making machine learning a useful tool in the accelerated discovery of transition metal complexes*, WIREs Comput. Mol. Sci. **10**, e1439 (2020).
- [18] R. Pollice et al., *Data-Driven Strategies for Accelerated Materials Design*, Acc. Chem. Res. **54**, 849 (2021).
- [19] A. Nandy, C. Duan, M. G. Taylor, F. Liu, A. H. Steeves, and H. J. Kulik, *Computational Discovery of Transition-metal Complexes: From High-throughput Screening to Machine Learning*, Chem. Rev. **121**, 9927 (2021).
- [20] C. D. Rankine and T. J. Penfold, *Progress in the Theory of X-ray Spectroscopy: From Quantum Chemistry to Machine Learning and Ultrafast Dynamics*, J. Phys. Chem. A **125**, 4276 (2021).
- [21] T. Kirschbaum, B. von Seggern, J. Dzubiella, A. Bande, and F. Noé, *Machine Learning Frontier Orbital Energies of Nanodiamonds*, J. Chem. Theory Comput. **19**, 4461 (2023).
- [22] M. Eckhoff, J. V. Diedrich, M. Mücke, and J. Proppe, *Quantitative Structure–Reactivity Relationships for Synthesis Planning: The Benzhydrylium Case*, J. Phys. Chem. A **128**, 343 (2024).
- [23] Ch. R. Jacob, S. M. Beyhan, R. E. Bulo, A. S. P. Gomes, A. W. Götz, K. Kiewisch, J. Sikkema, and L. Visscher, *PyADF — A scripting framework for multiscale quantum chemistry*, J. Comput. Chem. **32**, 2328 (2011).
- [24] J. Yang and M. P. Waller, *JACOB: A Dynamic Database for Computational Chemistry Benchmarking*, J. Chem. Inf. Model. **52**, 3255 (2012).
- [25] A. Jain et al., *FireWorks: a dynamic workflow system designed for high-throughput applications*, Concurr. Comput. Pract. Exp. **27**, 5037 (2015).
- [26] J. Arshad, A. Hoffmann, S. Gesing, R. Grunzke, J. Krüger, T. Kiss, S. Herres-Pawlis, and G. Terstyanszky, *Multi-level meta-workflows: new concept for regularly occurring tasks in quantum chemistry*, J. Cheminformatics **8**, 58 (2016).
- [27] A. H. Larsen et al., *The atomic simulation environment—a Python library for working with atoms*, J. Phys.: Condens. Matter **29**, 273002 (2017).
- [28] Y. Guan, V. M. Ingman, B. J. Rooks, and S. E. Wheeler, *AARON: An Automated Reaction Optimizer for New Catalysts*, J. Chem. Theory Comput. **14**, 5249 (2018).

- [29] F. Zapata, L. Ridder, J. Hidding, Ch. R. Jacob, I. Infante, and L. Visscher, *QMflows: A Tool Kit for Interoperable Parallel Workflows in Quantum Chemistry*, J. Chem. Inf. Model. **59**, 3191 (2019).
- [30] S. P. Huber et al., *AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance*, Sci. Data **7**, 300 (2020).
- [31] M. Uhrin, S. P. Huber, J. Yu, N. Marzari, and G. Pizzi, *Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows*, Comput. Mat. Sci. **187**, 110086 (2021).
- [32] J. George, *Automation in DFT-based computational materials science*, Trends Chem. **3**, 697 (2021).
- [33] V. M. Ingman, A. J. Schaefer, L. R. Andreola, and S. E. Wheeler, *QChASM: Quantum chemistry automation and structure manipulation*, WIREs Comput. Mol. Sci. **11**, e1510 (2021).
- [34] D. G. A. Smith et al., *Quantum Chemistry Common Driver and Databases (QCDB) and Quantum Chemistry Engine (QCEngine): Automation and interoperability among computational chemistry programs*, J. Chem. Phys. **155**, 204801 (2021).
- [35] J. P. Unsleber, S. A. Grimm, and M. Reiher, *Chemoton 2.0: Autonomous Exploration of Chemical Reaction Networks*, J. Chem. Theory Comput. **18**, 5393 (2022).
- [36] E. Zurich, *SCINE: Software for Chemical Interaction Networks*, 2024, URL: <https://scine.ethz.ch/download/>.
- [37] C. B. Hicks and T. J. Martinez, *Massively Scalable Workflows for Quantum Chemistry: BIGCHEM and CHEMCLOUD*, ChemRxiv Preprint (2023), DOI: 10.26434/chemrxiv-2023-bn0jv-v2.
- [38] P. O. Dral et al., *MLatom 3: Platform for machine learning-enhanced computational chemistry simulations and workflows*, (2023), arXiv:2310.20155 [physics], DOI: 10.48550/arXiv.2310.20155.
- [39] A. S. Rosen et al., *Jobflow: Computational Workflows Made Simple*, J. Open Source Softw. **9**, 5995 (2024).
- [40] A. S. P. Gomes and Ch. R. Jacob, *Quantum-chemical embedding methods for treating local electronic excitations in complex chemical systems*, Annu. Rep. Prog. Chem., Sect. C **108**, 222 (2012).
- [41] M. S. Gordon, editor, *Fragmentation: Toward Accurate Calculations on Complex Molecular Systems*, Wiley, Hoboken, NJ, 1st edition, 2017.

- [42] L. O. Jones, M. A. Mosquera, G. C. Schatz, and M. A. Ratner, *Embedding Methods for Quantum Chemistry: Applications from Materials to Life Sciences*, J. Am. Chem. Soc. **142**, 3281 (2020).
- [43] M. Jansen, N. N. T. Minh, E. D. Hedegård, and C. König, Quantum-derived embedding schemes for local excitations, in *Chemical Modelling*, pages 24–60, Royal Society of Chemistry, 2022, DOI: 10.1039/9781839169342-00024.
- [44] H. M. Senn and W. Thiel, *QM/MM Methods for Biomolecular Systems*, Angew. Chem. Int. Ed. **48**, 1198 (2009).
- [45] S. F. Sousa, A. J. M. Ribeiro, R. P. P. Neves, N. F. Brás, N. M. F. S. A. Cerqueira, P. A. Fernandes, and M. J. Ramos, *Application of quantum mechanics/molecular mechanics methods in the study of enzymatic reaction mechanisms*, WIREs Comput. Mol. Sci. **7**, e1281 (2017).
- [46] R. P. Magalhães, H. S. Fernandes, and S. F. Sousa, *Modelling Enzymatic Mechanisms with QM/MM Approaches: Current Status and Future Challenges*, Isr. J. Chem. **60**, 655 (2020).
- [47] K. Raghavachari and A. Saha, *Accurate Composite and Fragment-Based Quantum Chemical Models for Large Molecules*, Chem. Rev. **115**, 5643 (2015).
- [48] J. M. Herbert, *Fantasy versus reality in fragment-based quantum chemistry*, J. Chem. Phys. **151**, 170901 (2019).
- [49] J. Liu and X. He, *Recent advances in quantum fragmentation approaches to complex molecular and condensed-phase systems*, WIREs Comput. Mol. Sci. **13**, e1650 (2023).
- [50] L. W. Chung et al., *The ONIOM Method and Its Applications*, Chem. Rev. **115**, 5678 (2015).
- [51] P. Seeber, S. Seidenath, J. Steinmetzer, and S. Gräfe, *Growing Spicy ONIOMs: Extending and generalizing concepts of ONIOM and many body expansions*, WIREs Comput. Mol. Sci. **13**, e1644 (2023).
- [52] Ch. R. Jacob and J. Neugebauer, *Subsystem density-functional theory*, WIREs Comput. Mol. Sci. **4**, 325 (2014).
- [53] A. Krishtal, D. Sinha, A. Genova, and M. Pavanello, *Subsystem density-functional theory as an effective tool for modeling ground and excited states, their dynamics and many-body interactions*, J. Phys.: Condens. Matter **27**, 183202 (2015).
- [54] T. A. Wesolowski, S. Shedge, and X. Zhou, *Frozen-Density Embedding Strategy for Multilevel Simulations of Electronic Structure*, Chem. Rev. **115**, 5891 (2015).

- [55] S. J. R. Lee, M. Welborn, F. R. Manby, and T. F. Miller, *Projection-Based Wavefunction-in-DFT Embedding*, Acc. Chem. Res. **52**, 1359 (2019).
- [56] Ch. R. Jacob and J. Neugebauer, *Subsystem density-functional theory (update)*, WIREs Comput. Mol. Sci. **14**, e1700 (2024).
- [57] S. Metz, J. Kästner, A. A. Sokol, T. W. Keal, and P. Sherwood, *ChemShell—a modular software package for QM/MM simulations*, WIREs Comput. Mol. Sci. **4**, 101 (2014).
- [58] O. Weingart, A. Nenov, P. Altoè, I. Rivalta, J. Segarra-Martí, I. Dokukina, and M. Garavelli, *COBRAMM 2.0 — A software interface for tailoring molecular electronic structure calculations and running nanoscale (QM/MM) simulations*, J. Mol. Model. **24**, 271 (2018).
- [59] Y. Lu, M. R. Farrow, P. Fayon, A. J. Logsdail, A. A. Sokol, C. R. A. Catlow, P. Sherwood, and T. W. Keal, *Open-Source, Python-Based Redevelopment of the ChemShell Multiscale QM/MM Environment*, J. Chem. Theory Comput. **15**, 1317 (2019).
- [60] B. Zhang, D. Altarawy, T. Barnes, J. M. Turney, and H. F. Schaefer, *Janus: An Extensible Open-Source Software Package for Adaptive QM/MM Methods*, J. Chem. Theory Comput. **15**, 4362 (2019).
- [61] J. P. Götze, Y.-W. Pi, S. Petry, F. Langkabel, J. F. Witte, and O. Lemke, *A user-friendly, Python-based quantum mechanics/Gromacs interface: gm_x2qm_{mm}*, Int. J. Quantum Chem. **121**, e26486 (2021).
- [62] S. Martí, *QMCube (QM3): An all-purpose suite for multiscale QM/MM calculations*, J. Comput. Chem. **42**, 447 (2021).
- [63] Y. Lu et al., *Multiscale QM/MM modelling of catalytic systems with ChemShell*, Phys. Chem. Chem. Phys. **25**, 21816 (2023).
- [64] H. L. Woodcock, B. T. Miller, M. Hodoscek, A. Okur, J. D. Larkin, J. W. Ponder, and B. R. Brooks, *MSCALE: A General Utility for Multiscale Modeling*, J. Chem. Theory Comput. **7**, 1208 (2011).
- [65] A. V. Odinokov, N. O. Dubinets, and A. A. Bagaturyants, *pyEFP: Automatic decomposition of the complex molecular systems into rigid polarizable fragments*, J. Comput. Chem. **39**, 807 (2018).
- [66] D. R. Broderick and J. M. Herbert, *Scalable generalized screening for high-order terms in the many-body expansion: Algorithm, open-source implementation, and demonstration*, J. Chem. Phys. **159**, 174801 (2023).

- [67] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, and T. Ziegler, *Chemistry with ADF*, J. Comput. Chem. **22**, 931 (2001).
- [68] Software for Chemistry and Materials, Amsterdam, AMS, Amsterdam Modelling Suite, Version 2021.201, 2021, URL: <http://www.scm.com>.
- [69] K. Aidas et al., *The Dalton quantum chemistry program system*, WIREs Comput. Mol. Sci. **4**, 269 (2014).
- [70] J. M. H. Olsen et al., *Dalton Project: A Python platform for molecular- and electronic-structure simulations of complex systems*, J. Chem. Phys. **152**, 214115 (2020).
- [71] T. Saue et al., *The DIRAC code for relativistic molecular calculations*, J. Chem. Phys. **152**, 204104 (2020).
- [72] H. J. J. v. Dam, W. A. d. Jong, E. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, and M. Valiev, *NWChem: scalable parallel computational chemistry*, WIREs Comput. Mol. Sci. **1**, 888 (2011).
- [73] E. Aprà et al., *NWChem: Past, present, and future*, J. Chem. Phys. **152**, 184102 (2020).
- [74] F. Aquilante et al., *Modern quantum chemistry with [Open]Molcas*, J. Chem. Phys. **152**, 214117 (2020).
- [75] G. Li Manni et al., *The OpenMolcas Web: A Community-Driven Approach to Advancing Computational Chemistry*, J. Chem. Theory Comput. **19**, 6933 (2023).
- [76] F. Neese, F. Wennmohs, U. Becker, and C. Riplinger, *The ORCA quantum chemistry program package*, J. Chem. Phys. **152**, 224108 (2020).
- [77] F. Neese, *Software update: The ORCA program system—Version 5.0*, WIREs Comput. Mol. Sci. **12**, e1606 (2022).
- [78] S. G. Balasubramani et al., *TURBOMOLE: Modular program suite for ab initio quantum-chemical and condensed-matter simulations*, J. Chem. Phys. **152**, 184107 (2020).
- [79] Y. J. Franzke et al., *TURBOMOLE: Today and Tomorrow*, J. Chem. Theory Comput. **19**, 6859 (2023).
- [80] P. Giannozzi et al., *Quantum ESPRESSO toward the exascale*, J. Chem. Phys. **152**, 154105 (2020).

- [81] I. Carnimeo et al., *Quantum ESPRESSO: One Further Step toward the Exascale*, J. Chem. Theory Comput. **19**, 6992 (2023).
- [82] Ch. R. Jacob, *How Open Is Commercial Scientific Software?*, J. Phys. Chem. Lett. **7**, 351 (2016).
- [83] T. A. Wesolowski and A. Warshel, *Frozen Density Functional Approach for ab Initio Calculations of Solvated Molecules*, J. Phys. Chem. **97**, 8050 (1993).
- [84] A. S. P. Gomes, Ch. R. Jacob, and L. Visscher, *Calculation of local excitations in large systems by embedding wave-function theory in density-functional theory*, Phys. Chem. Chem. Phys. **10**, 5353 (2008).
- [85] S. Höfener, A. Severo Pereira Gomes, and L. Visscher, *Molecular properties via a subsystem density functional theory formulation: A common framework for electronic embedding*, J. Chem. Phys. **136**, 044104 (2012).
- [86] S. Höfener and L. Visscher, *Calculation of electronic excitations using wave-function in wave-function frozen-density embedding*, J. Chem. Phys. **137**, 204120 (2012).
- [87] A. S. P. Gomes, Ch. R. Jacob, F. Réal, L. Visscher, and V. Vallet, *Towards systematically improvable models for actinides in condensed phase: the electronic spectrum of uranyl in $\text{Cs}_2\text{UO}_2\text{Cl}_4$ as a test case*, Phys. Chem. Chem. Phys. **15**, 15153 (2013).
- [88] M. Dułak, J. W. Kamiński, and T. A. Wesolowski, *Linearized orbital-free embedding potential in self-consistent calculations*, Int. J. Quantum Chem. **109**, 1886 (2009).
- [89] Ch. R. Jacob, J. Neugebauer, and L. Visscher, *A flexible implementation of frozen-density embedding for use in multilevel simulations*, J. Comput. Chem. **29**, 1011 (2008).
- [90] J. P. Unsleber, T. Dresselhaus, K. Klahr, D. Schnieders, M. Böckers, D. Barton, and J. Neugebauer, *Serenity: A subsystem quantum chemistry program*, J. Comput. Chem. **39**, 788 (2018).
- [91] N. Niemeyer et al., *The subsystem quantum chemistry program Serenity*, WIREs Comput. Mol. Sci. **13**, e1647 (2023).
- [92] S. Andermatt, J. Cha, F. Schiffmann, and J. VandeVondele, *Combining Linear-Scaling DFT with Subsystem DFT in Born–Oppenheimer and Ehrenfest Molecular Dynamics Simulations: From Molecules to a Virus in Solution*, J. Chem. Theory Comput. **12**, 3214 (2016).
- [93] A. Genova, D. Ceresoli, A. Krishtal, O. Andreussi, R. A. DiStasio, and M. Pavanello, *eQE: An open-source density functional embedding theory code for the condensed phase*, Int. J. Quantum Chem. **117**, e25401 (2017).

- [94] W. Mi, X. Shao, A. Genova, D. Ceresoli, and M. Pavanello, *eQE 2.0: Subsystem DFT beyond GGA functionals*, Comput. Phys. Commun. **269**, 108122 (2021).
- [95] S. Höfener, *The KOALA program: Wavefunction frozen-density embedding*, Int. J. Quantum Chem. **121**, e26351 (2021).
- [96] A. D. Becke, *A multicenter numerical integration scheme for polyatomic molecules*, J. Chem. Phys. **88**, 2547 (1988).
- [97] M. Olejniczak, R. Bast, and A. S. P. Gomes, *On the calculation of second-order magnetic properties using subsystem approaches in a relativistic framework*, Phys. Chem. Chem. Phys. **19**, 8400 (2017).
- [98] Q. Sun et al., *PySCF: the Python-based simulations of chemistry framework*, WIREs Comput. Mol. Sci. **8**, e1340 (2018).
- [99] Q. Sun et al., *Recent developments in the PySCF program package*, J. Chem. Phys. **153**, 024109 (2020).
- [100] T. Oliphant et al., NUMPY — A python library for numerical computations, URL: <http://www.scipy.org/NumPy>.
- [101] C. R. Harris et al., *Array programming with NumPy*, Nature **585**, 357 (2020).
- [102] The HDF Group, Hierarchical Data Format 5, Version 1.14.0, 1997, URL: <https://www.hdfgroup.org/HDF5/>.
- [103] S. Borini et al., *FORTTRAN Interface for Code Interoperability in Quantum Chemistry: The Q5Cost Library*, J. Chem. Inf. Model. **47**, 1271 (2007).
- [104] D. G. A. Smith, D. Altarawy, L. A. Burns, M. Welborn, L. N. Naden, L. Ward, S. Ellis, B. P. Pritchard, and T. D. Crawford, *The MolSSI QCArchive project: An open-source platform to compute, organize, and share quantum chemistry data*, WIREs Comput. Mol. Sci. **11**, e1491 (2021).
- [105] E. Posenitskiy et al., *TREXIO: A file format and library for quantum chemistry*, J. Chem. Phys. **158**, 174801 (2023).
- [106] K. G. Dyall and K. Fægri Jr., *Introduction to Relativistic Quantum Chemistry*, Oxford University Press, New York, 2007.
- [107] P. Pyykkö, *Relativistic Effects in Chemistry: More Common Than You Thought*, Annu. Rev. Phys. Chem. **63**, 45 (2012).
- [108] T. Saue, *Relativistic Hamiltonians for Chemistry: A Primer*, ChemPhysChem **12**, 3077 (2011).

- [109] T. Saue and H. J. A. Jensen, *Quaternion symmetry in relativistic molecular calculations: The Dirac–Hartree–Fock method*, J. Chem. Phys. **111**, 6211 (1999).
- [110] B. Senjean, S. Sen, M. Repisky, G. Knizia, and L. Visscher, *Generalization of Intrinsic Orbitals to Kramers-Paired Quaternion Spinors, Molecular Fragments, and Valence Virtual Spinors*, J. Chem. Theory Comput. **17**, 1337 (2021).
- [111] G. Schaftenaar and J. H. Noordik, *Molden: a pre- and post-processing program for molecular and electronic structures*, J. Comput.-Aided Mol. Des. **14**, 123 (2000).
- [112] K. Focke and Ch. R. Jacob, *Coupled-Cluster Density-Based Many-Body Expansion*, J. Phys. Chem. A **127**, 9139 (2023).
- [113] U. Ekström et al., XCFun: Exchange-Correlation functionals with arbitrary order derivatives, 2019, URL: <https://github.com/dftlibs/xcfun>.
- [114] U. Ekström et al., XCFun: Exchange-Correlation functionals with arbitrary order derivatives, Version 2.2.1, 2020, DOI: 10.5281/zenodo.4269992, URL: <https://github.com/dftlibs/xcfun/tree/v2.1.1>.
- [115] Y. Bouchafra, A. Shee, F. Réal, V. Vallet, and A. Severo Pereira Gomes, *Predictive Simulations of Ionization Energies of Solvated Halide Ions with Relativistic Embedded Equation of Motion Coupled Cluster Theory*, Phys. Rev. Lett. **121**, 266001 (2018).
- [116] A. P. Gaiduk, M. Govoni, R. Seidel, J. H. Skone, B. Winter, and G. Galli, *Photoelectron Spectra of Aqueous Solutions from First Principles*, J. Am. Chem. Soc. **138**, 6912 (2016).
- [117] R. A. Opoku, C. Toubin, and A. S. P. Gomes, *Simulating core electron binding energies of halogenated species adsorbed on ice surfaces and in solution via relativistic quantum embedding calculations*, Phys. Chem. Chem. Phys. **24**, 14390 (2022).
- [118] L. Belpassi, M. De Santis, H. M. Quiney, F. Tarantelli, and L. Storchi, *BERTHA: Implementation of a four-component Dirac–Kohn–Sham relativistic framework*, J. Chem. Phys. **152**, 164118 (2020).
- [119] M. De Santis, L. Storchi, L. Belpassi, H. M. Quiney, and F. Tarantelli, *Py-BERTHART: A Relativistic Real-Time Four-Component TDDFT Implementation Using Prototyping Techniques Based on Python*, J. Chem. Theory Comput. **16**, 2410 (2020).
- [120] M. De Santis, D. Sorbelli, V. Vallet, A. S. P. Gomes, L. Storchi, and L. Belpassi, *Frozen-Density Embedding for Including Environmental Effects in the Dirac–Kohn–Sham Theory: An Implementation Based on Density Fitting and Prototyping Techniques*, J. Chem. Theory Comput. **18**, 5992 (2022).

- [121] W. A. Misael and A. Severo Pereira Gomes, *Core Excitations of Uranyl in $\text{Cs}_2\text{UO}_2\text{Cl}_4$ from Relativistic Embedded Damped Response Time-Dependent Density Functional Theory Calculations*, Inorg. Chem. **62**, 11589 (2023).
- [122] L. Halbert, M. Olejniczak, V. Vallet, and A. S. P. Gomes, *Investigating solvent effects on the magnetic properties of molybdate ions (MoO_4^{2-}) with relativistic embedding*, Int. J. Quantum Chem. **120**, e26207 (2020).
- [123] T. T. Nguyen, J. Jung, X. Trivelli, J. Trébosc, S. Cordier, Y. Molard, L. Le Pollès, C. J. Pickard, J. Cuny, and R. Gautier, *Evaluation of ^{95}Mo Nuclear Shielding and Chemical Shift of $[\text{Mo}_6\text{X}_{14}]^{2-}$ Clusters in the Liquid Phase*, Inorg. Chem. **54**, 7673 (2015).
- [124] J. Jakowski and K. Morokuma, *Liouville–von Neumann molecular dynamics*, J. Chem. Phys. **130**, 224106 (2009).
- [125] X. Li, S. M. Smith, A. N. Markevitch, D. A. Romanov, R. J. Levis, and H. B. Schlegel, *A time-dependent Hartree–Fock approach for studying the electronic optical response of molecules in intense fields*, Phys. Chem. Chem. Phys. **7**, 233 (2005).
- [126] M. De Santis, L. Belpassi, Ch. R. Jacob, A. Severo Pereira Gomes, F. Tarantelli, L. Visscher, and L. Storchi, *Environmental Effects with Frozen-Density Embedding in Real-Time Time-Dependent Density Functional Theory Using Localized Basis Functions*, J. Chem. Theory Comput. **16**, 5695 (2020).
- [127] J. A. Martinez B, M. De Santis, M. Pavanello, V. Vallet, and A. S. P. Gomes, *Solvation effects on halides core spectra with Multilevel Real-Time quantum embedding*, (2024), arXiv:2401.14548 [physics].
- [128] K. J. Koh, T. S. Nguyen-Beck, and J. Parkhill, *Accelerating Realtime TDDFT with Block-Orthogonalized Manby–Miller Embedding Theory*, J. Chem. Theory Comput. **13**, 4173 (2017).
- [129] M. De Santis, V. Vallet, and A. S. P. Gomes, *Environment Effects on X-Ray Absorption Spectra With Quantum Embedded Real-Time Time-Dependent Density Functional Theory Approaches*, Front. Chem. **10** (2022).
- [130] F. Ding, F. R. Manby, and T. F. Miller, *Embedded Mean-Field Theory with Block-Orthogonalized Partitioning*, J. Chem. Theory Comput. **13**, 1605 (2017).
- [131] D. Schmitt-Monreal and Ch. R. Jacob, *Frozen-density embedding-based many-body expansions*, Int. J. Quantum Chem. **120**, e26228 (2020).
- [132] D. Schmitt-Monreal and Ch. R. Jacob, *Density-Based Many-Body Expansion as an Efficient and Accurate Quantum-Chemical Fragmentation Method: Application to Water Clusters*, J. Chem. Theory Comput. **17**, 4144 (2021).

- [133] J. Cui, H. Liu, and K. D. Jordan, *Theoretical Characterization of the $(\text{H}_2\text{O})_{21}$ Cluster: Application of an n -body Decomposition Procedure*, J. Phys. Chem. B **110**, 18872 (2006).
- [134] R. M. Richard, K. U. Lao, and J. M. Herbert, *Understanding the many-body expansion for large systems. I. Precision considerations*, J. Chem. Phys. **141**, 014108 (2014).
- [135] S. Schürmann, J. R. Vornweg, M. Wolter, and Ch. R. Jacob, *Accurate quantum-chemical fragmentation calculations for ion–water clusters with the density-based many-body expansion*, Phys. Chem. Chem. Phys. **25**, 736 (2023).
- [136] C. A. Coulson, *Present State of Molecular Structure Calculations*, Rev. Mod. Phys. **32**, 170 (1960).
- [137] W. Thiel, *Theoretical Chemistry—Quo Vadis?*, Angew. Chem. Int. Ed. **50**, 9216 (2011).
- [138] F. Neese, M. Atanasov, G. Bistoni, D. Maganas, and S. Ye, *Chemistry and Quantum Mechanics in 2019: Give Us Insight and Numbers*, J. Am. Chem. Soc. **141**, 2814 (2019).
- [139] M. Olejniczak, A. Severo Pereira Gomes, and J. Tierny, *A Topological Data Analysis perspective on noncovalent interactions in relativistic calculations*, Int. J. Quantum Chem. **120**, e26133 (2020).
- [140] J. Contreras-García and W. Yang, *Perspective: Chemical Information Encoded in Electron Density*, Acta Phys.-Chim. Sin. **34**, 567 (2018).
- [141] E. R. Johnson, S. Keinan, P. Mori-Sánchez, J. Contreras-García, A. J. Cohen, and W. Yang, *Revealing Noncovalent Interactions*, J. Am. Chem. Soc. **132**, 6498 (2010).
- [142] R. Laplaza, F. Peccati, R. A. Boto, C. Quan, A. Carbone, J.-P. Piquemal, Y. Maday, and J. Contreras-García, *NCIPLOT and the analysis of noncovalent interactions using the reduced density gradient*, WIREs Comput. Mol. Sci. **11**, e1497 (2021).
- [143] P. Geerlings, F. De Proft, and W. Langenaeker, *Conceptual Density Functional Theory*, Chem. Rev. **103**, 1793 (2003).
- [144] R. Chauvin, C. Lepetit, B. Silvi, and E. Alikhani, editors, *Applications of Topological Methods in Molecular Chemistry*, volume 22 of *Challenges and Advances in Computational Chemistry and Physics*, Springer International Publishing, Cham, 2016, DOI: 10.1007/978-3-319-29022-5.

- [145] D. R. Lipsa, R. S. Laramée, S. J. Cox, J. C. Roberts, R. Walker, M. A. Borkin, and H. Pfister, *Visualization for the Physical Sciences*, Comput. Graph. Forum **31**, 2317 (2012).
- [146] W. Wang, W. Wang, and S. Li, *From numerics to combinatorics: a survey of topological methods for vector field visualization*, J.Vis. **19**, 727 (2016).
- [147] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. De Floriani, G. Scheuermann, H. Hagen, and C. Garth, *A Survey of Topology-based Methods in Visualization*, Comput. Graph. Forum **35**, 643 (2016).
- [148] R. F. W. Bader, *A quantum theory of molecular structure and its applications*, Chem. Rev. **91**, 893 (1991).
- [149] J. W. Milnor, *Morse Theory*, Princeton University Press, 1963.
- [150] H. Edelsbrunner and J. L. Harer, *Computational Topology: An Introduction*, American Mathematical Society, 2009.
- [151] K. Xia, *Persistent homology analysis of ion aggregations and hydrogen-bonding networks*, Phys. Chem. Chem. Phys. **20**, 13448 (2018).
- [152] K. Wu, Z. Zhao, R. Wang, and G.-W. Wei, *TopP-S: Persistent homology-based multi-task deep neural networks for simultaneous predictions of partition coefficient and aqueous solubility*, J. Comput. Chem. **39**, 1444 (2018).
- [153] M. Pirashvili, L. Steinberg, F. Belchi Guillaumon, M. Niranjana, J. G. Frey, and J. Brodzki, *Improved understanding of aqueous solubility modeling through topological data analysis*, J. Cheminformatics **10**, 54 (2018).
- [154] P. Hermosilla, J. Estrada, V. Guallar, T. Ropinski, À. Vinacua, and P.-P. Vázquez, *Physics-Based Visual Characterization of Molecular Interaction Forces*, IEEE Trans. Vis. Comput. Graph. **23**, 731 (2017).
- [155] D. Günther, R. A. Boto, J. Contreras-Garcia, J.-P. Piquemal, and J. Tierny, *Characterizing Molecular Interactions in Chemical Systems*, IEEE Trans. Vis. Comput. Graph. **20**, 2476 (2014).
- [156] H. Bhatia, A. G. Gyulassy, V. Lordi, J. E. Pask, V. Pascucci, and P.-T. Bremer, *TopoMS: Comprehensive topological exploration for molecular and condensed-matter systems*, J. Comput. Chem. **39**, 936 (2018).
- [157] J. S. M. Anderson, J. I. Rodríguez, P. W. Ayers, D. E. Trujillo-González, A. W. Götz, J. Autschbach, F. L. Castillo-Alvarado, and K. Yamashita, *Molecular QTAIM Topology Is Sensitive to Relativistic Corrections*, Chem. Eur. J. **25**, 2538 (2019).

- [158] M. Olejniczak and J. Tierny, *Topological data analysis of vortices in the magnetically-induced current density in LiH molecule*, Phys. Chem. Chem. Phys. **25**, 5942 (2023).
- [159] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux, *The Topology ToolKit*, IEEE Trans. Vis. Comput. Graph. **24**, 832 (2018), URL: <https://topology-tool-kit.github.io/>, DOI: 10.1109/TVCG.2017.2743938.
- [160] T. Bin Masood et al., *An Overview of the Topology ToolKit*, Nykoping, Sweden, 2019.
- [161] S. Lehtola, *A call to arms: Making the case for more reusable libraries*, J. Chem. Phys. **159**, 180901 (2023).
- [162] J. B. Rommel, *From Prescriptive to Predictive: an Interdisciplinary Perspective on the Future of Computational Chemistry*, (2021), arXiv:2103.02933 [physics], DOI: 10.48550/arXiv.2103.02933.
- [163] M. Karelson, V. S. Lobanov, and A. R. Katritzky, *Quantum-Chemical Descriptors in QSAR/QSPR Studies*, Chem. Rev. **96**, 1027 (1996).
- [164] P. O. Dral, *Quantum Chemistry in the Age of Machine Learning*, J. Phys. Chem. Lett. **11**, 2336 (2020).
- [165] B. Dou, Z. Zhu, E. Merkurjev, L. Ke, L. Chen, J. Jiang, Y. Zhu, J. Liu, B. Zhang, and G.-W. Wei, *Machine Learning Methods for Small Data Challenges in Molecular Science*, Chem. Rev. **123**, 8736 (2023).