

i-PI 3.0: a flexible and efficient framework for advanced atomistic simulations

Yair Litman,¹ Venkat Kapil,^{1,2,3} Yotam M. Y. Feldman,⁴ Davide Tisi,⁵ Tomislav Begušić,⁶ Karen Fidanyan,⁷ Guillaume Fraux,⁵ Jacob Higer,⁸ Matthias Kellner,⁵ Tao E. Li,⁹ Eszter S. Pósz,⁷ Elia Stocco,⁷ George Trenins,⁷ Barak Hirshberg,⁴ Mariana Rossi,⁷ and Michele Ceriotti^{5,*}

¹*Y. Hamied Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge, CB2 1EW, UK*

²*Dept. of Physics and Astronomy, University College London, 17-19 Gordon St, London WC1H 0AH, UK*

³*Thomas Young Centre & London Centre for Nanotechnology, 19 Gordon St, London WC1H 0AH, UK*

⁴*School of Chemistry, Tel Aviv University, Tel Aviv 6997801, Israel.*

⁵*Laboratory of Computational Science and Modeling, Institut des Matériaux, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland*

⁶*Div. of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125, USA*

⁷*MPI for the structure and Dynamics of Matter, Hamburg, Germany*

⁸*School of Physics, Tel Aviv University, Tel Aviv 6997801, Israel.*

⁹*Department of Physics and Astronomy, University of Delaware, Newark, Delaware 19716, USA*

(Dated: July 11, 2024)

Atomic-scale simulations have progressed tremendously over the past decade, largely due to the availability of machine-learning interatomic potentials. These potentials combine the accuracy of electronic structure calculations with the ability to reach extensive length and time scales. The i-PI package facilitates integrating the latest developments in this field with advanced modeling techniques, thanks to a modular software architecture based on inter-process communication through a socket interface. The choice of Python for implementation facilitates rapid prototyping but can add computational overhead. In this new release, we carefully benchmarked and optimized i-PI for several common simulation scenarios, making such overhead negligible when i-PI is used to model systems up to tens of thousands of atoms using widely adopted machine learning interatomic potentials, such as Behler-Parinello, DeePMD and MACE neural networks. We also present the implementation of several new features, including an efficient algorithm to model bosonic and fermionic exchange, a framework for uncertainty quantification to be used in conjunction with machine-learning potentials, a communication infrastructure that allows deeper integration with electronic-driven simulations, and an approach to simulate coupled photon-nuclear dynamics in optical or plasmonic cavities.

I. INTRODUCTION

The atomic-scale modeling of molecules and materials is a constantly evolving field driven by the interest of researchers in discovering new fundamental phenomena, interpreting complex experimental results, and, together with an increasing number of commercial enterprises, improving, discovering, and developing new materials.^[1–6] Several computational developments in recent years extended dramatically the problems that can be addressed by atomistic simulations, far exceeding the steady improvement of the underlying hardware. Arguably, the largest contribution to such fast progress can be attributed to the development of machine learning interatomic potentials (MLIPs). MLIPs approach the accuracy of the underlying *ab initio* data, but at a small fraction of the computational cost. They have allowed access to system sizes and time scales which were previously only accessible to empirical interatomic potentials.^[7–10] This has driven a renewed interest in complex simulation algorithms that were thought inapplicable for stud-

ies that require a first-principles description of interatomic interactions, such as the investigation of catalytic reactions^[11–13] or of the thermodynamic stability of materials.^[14–17] However, the lack of easy-to-use implementations of such approaches has become an important factor that hinders a more widespread adoption

Since its first release in 2014^[18], one of i-PI’s primary objectives has been “lowering the implementation barrier to bring state-of-the-art sampling and atomistic modeling techniques to *ab initio* and empirical interatomic potentials programs”. The first version of i-PI was designed for path integral molecular dynamics (PIMD) simulations, hence the code name. Since then, and thanks to a community effort, it has evolved into a “universal force engine for advanced molecular simulations”, including both standard and advanced simulations algorithms, as described in the following sections.^[19] In the last few years, our efforts have adapted to the changing needs of the field. While continuing with the implementation of new and advanced features, we have improved the scalability of the code backend, as required by newly available and computationally efficient MLIPs. We also adopted automatic testing workflows to strengthen code stability and enhance our reproducibility standards.

* michele.ceriotti@epfl.ch

i-PI’s capabilities as a versatile tool for atomistic simulations can be demonstrated by the wide range of applications in which it has been employed. These examples range from the dynamics of charge density wave materials [20], interpretation of neutron scatter experiments,[21] and nuclear motion associated to electronic excitations,[22–24] to the elucidation of the thermodynamic properties of aqueous electrons,[25–27] the calculation of first principles phase diagrams,[28–30] and the study of aqueous solutions and interfaces.[31–36] Moreover, by virtue of its modular structure, i-PI is already being used as a simulation engine with newly developed MLIP architectures [9, 37, 38] making it easy to use them in combination with advanced sampling techniques. It can also be used as a toolbox to combine different simulation “modes” to design sophisticated modeling workflows that involve multiple types of interatomic potentials, free-energy-sampling software[39] as well as post-processing of trajectory data.[40]

This manuscript reports on the progress made in the last 5 years since the release of version 2.0, by describing the newly adopted maintenance strategies, the improved performance, and the newly-implemented features. The manuscript is organised as follows: In Sec. II, we start with a short review of the code architecture, followed by a description of the improvements made to the code’s robustness and usability and a list of the main features available in this release. In Sec. III, we present a detailed discussion of the code efficiency for production and large-scale simulations while in Sec. IV, we describe some selected new features since the last code release discussing their implementation and examples of their application. In Sec. V, we showcase how the modularity and flexibility of i-PI help the implementation of advanced simulation setups that would otherwise require new codes to be written and Sec. VI concludes the article.

II. PROGRAM OVERVIEW

i-PI is a force engine that operates with a client-server paradigm. i-PI acts as a server in command of the evolution of the nuclear positions while one or more instances of the client code handle the evaluation of energy, forces, and possibly other properties for individual configurations. Designed to be universal, i-PI’s architecture is agnostic to the identity of the force providers, with a general and flexible backend that accommodates a wide range of client codes.

The code is distributed under a dual GPL and MIT license, and it is written in Python 3, a high-level, general-purpose interpreted programming language known for its emphasis on code readability. This choice facilitates rapid prototyping of new ideas and relatively easy code maintenance when compared to compiled languages traditionally used in scientific computing. Additionally, the availability of highly optimised numerical libraries allows Python-based software to deliver competitive per-

formance. Indeed, even computing-intensive electronic structure codes have released Python implementations in recent years.[41, 42] A detailed discussion of the code efficiency and improvements since the previous release is presented in Sec. III.

The communication between i-PI and the client codes is carried out through sockets, enabling local connections by shared-memory inter-process communication (UNIX domain sockets) or between two hosts connected to a local network or to the Internet (TCP/IP sockets). During the initialization, i-PI creates the sockets and awaits the connection of one or more clients. When the connection is established, the simulation loop starts: i) i-PI communicates the nuclear positions and the cell lattice vectors to the active clients, ii) the clients evaluate the required properties – typically energies, forces, and stresses – and send these back, and iii) i-PI uses this data to displace the atoms and to calculate the new nuclear positions, restarting the loop. As i-PI is designed to be client-agnostic, clients must be initialized independently before they can connect to the sockets. Importantly, the identity and the number of atoms must be defined consistently in the client code according to its own input mechanism and remain unchanged throughout the simulation.

A schematic representation of the code structure and the server-client communication is presented in Fig. 1. i-PI is structured in a modular way that represents the underlying physics of the target simulation as faithfully as possible. To achieve this, the code is organized around the *System* class which encodes all the information related to the physical system, such as the number and identity of atoms, the ensemble to be sampled, the number of replicas in the path integral simulations, the initialization procedure, and the algorithm for evolving the nuclear positions. Forces are managed through the *Forces* class, which can combine multiple force components, each of which is computed following the strategy specified in a *Forcefield* class. This two-layer approach is particularly advantageous for algorithms where the total forces and energies are obtained as the sum of terms computed at different accuracy levels,[24, 43–47] or for different portions of the system.[48] Simulation techniques that require the evolution of many systems simultaneously make use of the *SMotion* (Systems Motion) class. This class enables the definition of evolution steps that combine the (possibly many) different systems, facilitating, for example, exchanges between system replicas as done in replica exchange simulations, or the application of biases, like in metadynamics. Finally, estimators can be defined within the code or computed by the provided post-processing tools or user-generated scripts. An updated list of code features is reported in Section II C.

A. Code philosophy and development model

Many available client codes: i-PI is connected to a multitude of electronic structure and machine learn-

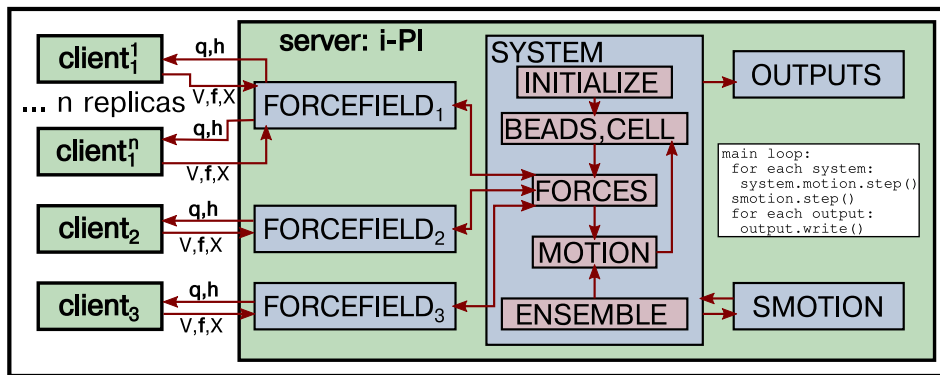


FIG. 1. Schematic representation of the i-PI code structure and the server-client communication. The physical system is defined by one or more replicas (beads), and the sampling conditions (e.g. temperature and pressure) by an *Ensemble* class. The forces acting on the atoms are constructed based on a combination of energy contributions evaluated by one or more instances of the *Forcefield* class (three in this example). Each *Forcefield* instance communicates with a different client sending positions (\mathbf{q}) and lattice vectors (\mathbf{h}), and receiving energy (V), forces (\mathbf{f}), stresses and possible extra properties (X) as a JSON formatted string. In simulations with multiple replicas, each *Forcefield* instance can accept connections from several clients, to achieve parallel evaluation. The *Motion* class determines the evolution of the atoms (e.g. time integration, or geometry optimization), while “system motion” classes (*SMotion*) can act on multiple systems, e.g. to carry out replica exchange simulations. The output class handles writing the output and restart files.

ing codes, either directly or through its ASE [49] or LAMMPS [50] interfaces. This wide-ranging connectivity means that a single implementation of an advanced simulation method is readily available to all the users of these codes, which significantly amplifies the potential impact of each contribution to i-PI. Furthermore, this capability enables a direct comparison between different client codes within an identical simulation setup.

Ease of implementation of new interfaces: Adding a new interface to i-PI is simple. For codes that provide a Python interface, this can be implemented by adding a small new file to the Python driver directory (`drivers/py/pes/`). For compiled software, the driver loop can usually be rather self-contained, modelled after the molecular dynamics or geometry optimization loop. The FORTRAN driver distributed with i-PI provides a good example, as well as utilities to open and communicate over sockets that can be reused in third-party codes, while the LAMMPS[50] implementation of `fix_ipi` can be used as a template of a C++ interface. In all cases, the structure (atom numbers and types) should be initialized *before* communication is established, and that i-PI does not guarantee that successive calls provide similar structures (as in a MD trajectory) which might require some care when the force evaluator makes a continuity assumption for neighbor list updates or wavefunction extrapolation.

Lower development and maintenance effort: In scenarios where a developer aims to use a newly implemented algorithm for nuclear motion with different types of codes, such as a density-functional theory (DFT) plane wave code to study periodic systems, or a quantum chemistry code to study gas phase molecules at a higher electronic structure level, two separate implementations would be required. By integrating the algorithm into i-

PI, only one implementation is needed and there is only one code to maintain.

Automatic (re)evaluation of physical quantities: The infrastructure of i-PI automatically manages updates of energy, forces, stresses, and other structure-dependent properties by establishing dependencies between physically related variables. This architecture enables developers to concentrate solely on the algorithm’s implementation making the development process more focused and efficient.

Participative development model: The modular structure of the code simplifies the contribution of self-contained implementations of new methods. These methods are explicitly listed in a dedicated section of the documentation, giving developers visibility and recognition for their contributions. Many of the applications presented in this paper are examples of this model.

Code Maintenance and Long-term Sustainability: As with most open-source projects, i-PI is developed and maintained by a community of volunteer contributors. Several of the authors of this manuscript, who are core developers and hold permanent or long-term academic positions, are involved in the governance of the code. These people have administrator rights on the main repository, which reduces the risk that the code base will be abandoned. We encourage power users to also help maintain the core infrastructure and the documentation; in preparation for this release, we have dedicated a special focus on improving the development infrastructure, as we discuss in the next Section.

B. Code robustness and usability

The increased number of i-PI users and contributors since the last release required enhancing the maintainability, robustness, and standardization of the code. These improvements involve five distinct aspects:

Documentation: We refactored the documentation of i-PI, which is now generated using `Sphinx`, is deployed automatically when new features are merged to the `main` branch, and is linked to the main code webpage [51]. The documentation has been updated to reflect the current status of the code, and a section about how to contribute new features to the code has been added to reduce the barrier for new developers. An automatic parsing of the Python docstrings throughout i-PI ensures a complete listing of all possible keywords and the corresponding explanation, which is automatically updated as soon as new features are implemented and merged in the main branch.

Examples: We have completely restructured the provided examples to make them easier to navigate and as intuitive as possible for newcomers. The examples are now organized into three directories, namely *features*, *clients*, and *hpc_scripts*. The *features* directory displays extensive examples of the different implemented functionalities, all of which can be run locally using the built-in Python or Fortran driver, without requiring additional software installation. The *clients* directory includes examples tailored to specific driver codes, highlighting the unique syntax and tags needed for an adequate setup when used in connection to i-PI. At the moment, we provide examples for ASE,[49] CASTEP,[52] CP2K,[53] LAMMPS,[50] DFTB+,[54] ffsGDML,[55] FHI-aims,[56] librascal,[57] Quantum ESPRESSO,[58–60] VASP,[61] elphmod,[62] and YAFF.[63] We reiterate that many other codes can also be connected to i-PI through their ASE and LAMMPS interfaces. Finally, motivated by the number of user queries on this topic, the *hpc_scripts* directory provides examples with the corresponding submission scripts to run the simulations on high-performance computing platforms.

Demonstrations and tutorials: We have added a *demos* directory that contains examples of more complicated setups, to illustrate the application of i-PI in realistic simulation scenarios, combining several of the features showcased in the *examples* directory. Some of these demonstration examples also cover some of the use of the pre- and post-processing scripts accompanying the code. Users specifically interested in learning the theory of path integral simulations and obtaining hands-on practice using i-PI are referred to the online courses developed with this goal.[64]

Continuous integration: We have implemented continuous integration workflows to automatically test for the code’s functionality. One automatic workflow reviews all the simulations in the *examples* directory by running them for a few steps and quickly verifying if they are functioning. A second workflow performs regression tests. In these cases, the simulations are performed for

the required amount of steps and the simulation outputs are compared against pre-computed reference values. Together, these workflows guarantee that any new modification to the code neither alters the results of simulations that are tested nor breaks the provided examples. Additional workflows, less critical for the functioning of the code, focus on checking the formatting syntax using Flake8 [65] and Black [66], or managing the updates of PyPI releases.

Developers’ documentation and workflows We have included and updated the development guidelines, that are discussed in the main documentation, as well as in the `CONTRIBUTING.md` file in the main repository. Contributors are encouraged to use the standard *git* workflows, such as creating a fork and opening a pull request, and expect code review that will involve both the correctness of the algorithm and the consistency with i-PI’s coding style and quality.

C. Main Features

We conclude this Section by listing the main features currently available in i-PI. A comprehensive list of features, indicating the main contributors, is maintained as part of the documentation.

1. Features available in the 2.0 release:

Classical and path integral molecular dynamics: in the *NVE*, *NVT*, *NPT* and constant stress (NsT) ensembles.

Thermostats: stochastic velocity rescaling,[67] path-integral Langevin equation thermostats,[68] generalized Langevin equation (GLE) thermostats, including the optimal sampling,[69, 70] quantum, [71], δ ,[72] hot-spot, [73] and PIGLET thermostats,[74] GLE thermostats for use with path-integral correlation functions,[75] fast forward Langevin thermostat,[76] Langevin sampling for noisy and/or dissipative forces.[77, 78]

Barostats: isotropic barostats,[18, 79] anisotropic barostats,[80] and its path integral adaptations.[81, 82]

Geometry optimizers: steepest descent, conjugate gradient, L-BFGS, trust radius BFGS, nudged elastic band [34, 83, 84] and mode following saddle point search.[85]

Enhanced sampling and free energy calculations: thermodynamic integration,[86] metadynamics, replica exchange MD,[87] and quantum alchemical transformations.[88, 89]

Multiple time-stepping (MTS): MTS in real time and imaginary time, also known as ring polymer contraction.[43, 44, 46, 48]

Approximate quantum dynamics calculations: ring polymer molecular dynamics (RPMD),[90, 91] centroid molecular dynamics (CMD),[92, 93] thermostatted

RPMD,[75, 94], ring polymer instanton rates and tunnelling splittings.[95]

Advanced path integral simulations and estimators: fourth-order path integrals,[96] perturbed path integrals,[97] the scaled-coordinate heat capacity estimator,[98] isotope fractionation estimations,[99] particle momentum distribution,[100] reweighted fourth-order path integral MD.[101, 102]

2. Features added after the 2.0 release:

Efficient integrators: BAOAB splitting [47], and Cayley integrator [103] for larger PIMD timesteps.

Barostats: MTTK fully flexible barostat [104], MTS implementation, and barostat implementation for the Suzuki-Chin [47] PIMD.

Advanced path integral simulations and estimators: Suzuki-Chin double virial operator estimator for calculating heat capacity [47] and Open chain path integrals [105] for estimating momentum dependent operators

Vibrational analysis: through the harmonic Hessian, independent mode framework, the vibrational self-consistent field, and self-consistent phonons.[106]

Particle exchange Monte Carlo: for efficient sampling of poorly ergodic mixtures [107]

Path Integral Molecular Dynamics (PIMD) for indistinguishable particles: Methods for simulating bosons,[108] with improved quadratic scaling,[109] and fermions, through a reweighting procedure.[110]

Cavity MD (CavMD): Simulation of systems confined in optical or plasmonic cavities.[111, 112]

Nuclear propagation by additional dynamical variables: non-adiabatic tunnelling rates in metallic systems,[113, 114] molecular dynamics with time-dependent external fields.

Ensemble models: committee models, baseline energy model and uncertainty estimation.[107, 115]

Extracting dynamical information from thermostatted simulations: for vibrational spectra and diffusion coefficients from PIGLET thermostatted simulations [116]

Non-linear spectroscopy: Evaluation of non-linear spectroscopic observables by equilibrium non-equilibrium simulations for classical and quantum nuclei.[117–119]

T_e path integral coarse-graining simulations (PIGS) simulations: for approximate quantum dynamics using molecular dynamics on an MLIP representing the centroid potential of mean force [120, 121].

III. EFFICIENCY OF LARGE-SCALE SIMULATIONS

i-PI’s socket-based communication and Python architecture offer convenience and flexibility to implement new and advanced simulation algorithms, facilitating integration with other codes. This flexibility, however, comes

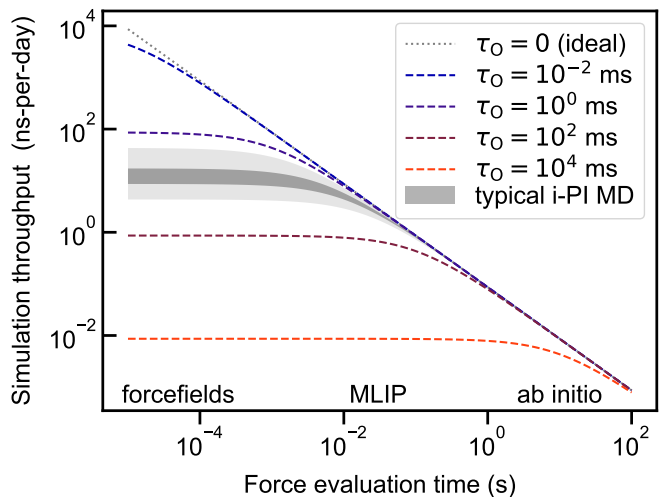


FIG. 2. Simplified picture of the impact of a given overhead τ_O on the simulation throughput (expressed in ns/day, assuming a time step of 1 fs) as a function of the force evaluation time, the lower the impact a given τ_O on the simulation throughput. An overhead of about 10 ms (typical for a moderately complicated simulation with i-PI 2.6) is negligible when used with *ab initio* force evaluations, but becomes significant for the most efficient (or heavily parallelised) MLIPs.

at the price of a computational overhead associated with UNIX- and TCP-socket communications, and Python’s inherent inefficiency compared to lower-level languages such as Fortran, C, and C++.

To analyse i-PI’s overhead on the performance of the simulations, it is instructive to separate the intrinsic wall-clock time needed to evaluate energy and forces, τ_F , and the time overhead associated only to i-PI, τ_O , such that the length of simulation that can be obtained for a certain computational effort is proportional to $1/(\tau_F + \tau_O)$. The initial focus of i-PI was on *ab initio* force providers, where τ_F is of the order of several seconds, and therefore an overhead of $\tau_O \sim 10$ ms was inconsequential (see Fig. 2). However, with MLIPs becoming the primary tool for large-scale, reactive calculations, an overhead of several ms can become the performance bottleneck. Therefore, we have improved the efficiency of i-PI for this release.

A. Estimating the overhead of i-PI

The cost per MD step of i-PI simulations can be roughly modelled as a constant term plus a per-atom contribution. This overhead can be estimated as $\tau_O \approx A + B N_a$. Note that, for such a wide range of system sizes, it is not advisable to determine the coefficients by linear regression because the intercept would be affected by large relative errors. Therefore, we determined A as the time per step of the smaller simulation (8 atoms) and B as the time per step per atom of the largest one (2^{16}

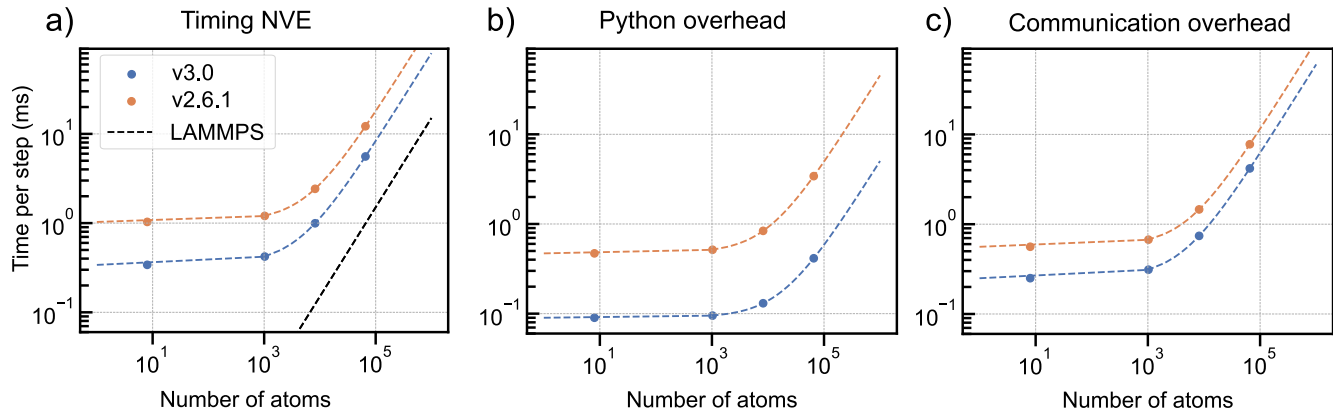


FIG. 3. (a) System size dependence of the timing per step when performing a NVE simulation of an ideal gas using LAMMPS, and when using i-PI with a dummy driver that returns zero energy and forces, communicating through UNIX sockets. The cost of a simulations run with i-PI is broken down into (b) a base Python overhead (obtained by performing the i-PI simulation with no forcefield) and the communication overhead (c), that is obtained by subtracting the Python overhead from the cost of a simulation that does communicate with the dummy driver. Markers correspond to actual simulations while the dashed lines are fits to $\tau_O = A + BN_a$, where N_a is the number of atoms. All simulations were run on nodes with Intel Xeon E5-2690 v3 @ 2.60GHz processors using a single core.

					A (ms)		B ($\mu\text{s}/N_a$)	
type	replicas	socket	ensemble	thermostat	i-PI 2.6	i-PI 3.0	i-PI 2.6	i-PI 3.0
MD	1	none	NVE	none	0.47	0.09	0.06	0.02
MD	1	UNIX	NVE	none	1.03	0.34	0.17	0.08
		UNIX socket communication overhead			0.56	0.25	0.11	0.06
MD	1	TCP/IP	NVE	none	21.54	1.26	0.64	0.10
		TCP/IP socket communication overhead			21.07	1.20	0.56	0.08
PIMD	2	none	NPT	pile_l	1.39	0.47	0.94	0.60
PIMD	2	UNIX	NPT	pile_l	2.62	1.27	1.32	0.76
		UNIX socket communication overhead			1.23	0.80	0.38	0.16
PIMD	2	TCP/IP	NPT	pile_l	32.64	2.45	1.68	0.77
		TCP/IP socket communication overhead			31.25	1.98	0.74	0.17

TABLE I. Comparison between the timings for the tests in the `ipi_tests/profiling` directory with i-PI version 3.0.0-beta or 2.6.3. The coefficients A and B can be used to estimate the overhead for a given number of atoms N_a , as $\tau_O = A + BN_a$. In the table, (PI)MD indicate simulations performed with (path integral) MD; “socket” indicates the type of communication, *none* corresponding to simulations that avoid entirely force evaluations. “pile_l” stands for the path-integral Langevin thermostat.[68]. The purpose of this table is to make the communication overhead of MD and PIMD runs evident. The numbers are obtained by subtracting the UNIX or TCP/IP socket values from the respective line where socket is *none*. All the simulations were run on nodes with 2 Intel(R) Xeon(R) Platinum 8360Y processors using a single core.

atoms).

We start by analysing the simplest possible case of a constant-energy simulation of an ideal gas, with no forces acting between atoms and therefore $\tau_F \approx 0$. (Fig. 3). In this case, the advantage of a C++ implementation such as that in LAMMPS is evident. For the largest system sizes, LAMMPS is about 10 times faster than i-PI, but for a few atoms the difference is much larger: LAMMPS can perform millions of steps per second, whereas i-PI 2.6 saturates at about 1000 steps per second (Fig. 3a).

It is also instructive to partition τ_O into a contribution

associated with the Python overhead and the one associated with the socket communication. We estimated the former by disabling force evaluation altogether, which can be achieved by setting `weight="0"` as an attribute of the `<force>` components. We estimated the latter by subtracting the Python overhead from the overall cost of a simulation communicating with a dummy driver that simply returned zero energy and forces (Fig. 3b-c).

Fig. 3 also compares the computational efficiency of the stable 2.6 version with that of the current 3.0 release. In the latter, we performed several optimizations

type	replicas	socket	ensemble	thermostat	A (ms)		B ($\mu\text{s}/N_a$)	
					i-PI 2.6	i-PI 3.0	i-PI 2.6	i-PI 3.0
MD	1	none	NVT	svr	0.56	0.14	0.07	0.03
MD	1	none	NVT	langevin	0.86	0.22	0.24	0.21
MD	1	none	NPT	langevin	1.45	0.47	0.37	0.32
MD	1	none	NPT	svr	1.13	0.38	0.22	0.13
PIMD	32	none	NPT	langevin	2.73	0.67	17.21	12.97
REMD	4×6	none	SC-NPT	langevin	27.80	16.86	11.29	12.50

TABLE II. Comparison between the timings for additional tests in the `ipi_tests/profiling` directory with i-PI version 3.0.0-beta or 2.6.3, presented in the same way as in Table I. In the table, (PI)MD indicate simulations performed with (path integral) MD; Thermostats are chosen between stochastic velocity rescaling, *svr* and [122] plain Langevin. The last line corresponds to a complicated setup combining path integral and replica-exchange molecular dynamics (REMD) with 4 beads and 6 ensemble replicas and a finite-difference evaluation of Suzuki-Chin (SC) high-order path integrals.[96, 123] All the simulations were run avoiding force evaluations entirely (socket = *none*), on nodes with 2 Intel(R) Xeon(R) Platinum 8360Y processors using a single core. The communication timings reported in Table I should be added to the timings reported here for a realistic estimation, keeping into account, when relevant, the number of replicas.

to reduce the overhead associated with accessing the i-PI `depend` objects,[18] substituted explicit loops with array operations, and removed other obvious bottlenecks from the implementation. This reduced the overhead of the NVE integrator by a factor of 3-5, and the communication overhead by a factor of 2.

In Table I, we report the values of the A and B coefficients for different simulation setups enabling the calculation of the communication overhead, while in Table II, we report the values of the A and B associated with the Python overhead for different nuclear motion algorithms. Thus, the total overall i-PI overhead, τ_O , can be obtained by adding the TCP/IP or UNIX socket timings from Table I to the algorithm-specific times from Table II. In all cases, the new release dramatically reduces both constants so that for most classical MD simulations and up to a few 1000s of atoms, one can now expect an overhead of a few ms per step. It is thus possible to perform tens of millions of simulation steps per day when disregarding the force evaluation costs.

The Python overhead is larger when considering simulations with multiple replicas and/or complicated setups that involve, e.g. high-order path integration or replica exchange, as evidenced in Table II. Often, however, these advanced simulation techniques decrease the overall cost of a calculation by reducing the number of force evaluations necessary to reach the desired statistical accuracy. This is a key advantage of using i-PI over faster, but less feature-rich, alternatives. An example, also relevant to classical MD trajectories, is the use of BAOAB integrators,[124] which allow doubling the time step in simulations of liquid water. A technical detail we also want to highlight from Table I is the $> 20\times$ reduction of the constant overhead for TCP/IP communication. This was achieved by disabling Nagle’s algorithm,[125] which was found to interfere negatively with the transfer of small messages in the i-PI communication protocol. This change has to be enacted also on the client side, as currently done in the driver codes we distribute with

i-PI, so we urge developers of packages that contain an i-PI interface to apply similar changes.

B. Efficient performance of i-PI on realistic parallelised simulations

Even though this release significantly cuts the i-PI overhead τ_O , and we are committed to further reducing it in the future, it is important to determine whether and when it matters in practical use cases. To this end, we consider the task of simulating 4096 water molecules at room temperature, in the canonical *NVT* ensemble, using a stochastic velocity rescaling thermostat. Based on Table I, we expect that using i-PI 3.0 should entail an overhead of 1.3 ms when using a UNIX socket, and 2.5 ms when using a TCP/IP socket, as necessary for simulations running across many nodes.

We start by considering the LAMMPS implementation of a DeePMD neural network, that is used both to perform MD directly using its LAMMPS implementation,[7, 127, 128] and as a driver for running classical MD and path integral molecular dynamics with i-PI. On the hardware we use for this test, detailed in the caption of Fig. 4, the evaluation of energies and forces on one CPU requires approximately 10 s/step, which is several orders of magnitude larger than the estimated τ_O .

In practical scenarios, however, the MLIP calculation will be heavily parallelized, either using a multi-CPU implementation or using one or more GPUs. As shown in Fig. 4, i-PI and LAMMPS having essentially equivalent performance up to the highest parallelization level for CPUs (up to 4 nodes, 288 cores), consistent with the small predicted τ_O . The GPU implementation of DeePMD is much faster and can be parallelized efficiently up to about 8 GPUS (4 nodes), for which (with a time step of 0.5 fs) LAMMPS clocks almost 2.5 ns of simulation per day. In our initial benchmarks we observed the empirical τ_O with LAMMPS DeePMD to be up to 10

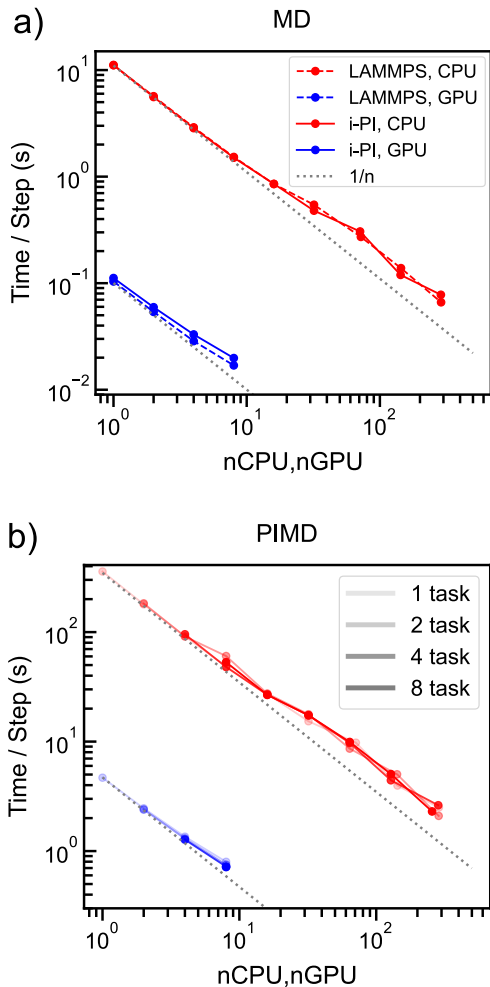


FIG. 4. Timings for 1 step of NVT dynamics with Langevin thermostatting, for a box containing 4096 water molecules at ambient temperature and density, using a DeePMD model,^[126] using the LAMMPS implementation.^[127, 128] Simulations were run on nodes with 2 Intel(R) Xeon(R) Platinum 8360Y, with 36 cores each (CPU) and nodes with 2 NVIDIA V100 cards (GPU). Parallelization is based on MPI, leveraging the LAMMPS domain decomposition. Dotted lines are guides for the eye corresponding to an ideal $1/n$ scaling. (a) Timings for MD simulations. Dashed and full lines correspond to simulations performed with LAMMPS, and i-PI using LAMMPS/DeePMD as a driver, respectively. (b) Timings for PIMD simulations with 32 replicas. The shading of the curves indicates the number of independent LAMMPS tasks that are connected to i-PI, and the number of processing units indicates the sum across all tasks.

times larger than when using the dummy driver, which we could trace to unnecessary reconstructions of the neighbor list at the LAMMPS side of the interface. We could optimize the LAMMPS interface of the driver, minimizing the number of neighbor list updates, and reducing the observed overhead to about 3 ms for the 8-GPU run. This is still almost 2 times larger than predicted, because of the additional bookkeeping that is needed to

interface i-PI with a feature-rich code such as LAMMPS, in comparison to the minimalistic dummy driver. Nevertheless, these optimizations have reduced τ_O to less than 20% of the overall simulation step even for the highly efficient multi-GPU setup, allowing to perform more than 2 ns of simulation per day with a relatively large number of atoms. For a PIMD setup, i-PI allows exploiting an additional level of parallelism by running multiple clients in parallel. Given that this is a trivial type of parallelization, it can improve the scaling whenever the client code shows less-than-ideal scaling, as observed for the GPU implementation of DeePMD in Fig. 4b, even though the speedup is minimal in this case.

					throughput (ns/day)	
model	architecture	$n_{\text{CPU}}/n_{\text{GPU}}$	$n_{\text{H}_2\text{O}}$	τ_{F}/s	direct	w/i-PI
DeePMD	CPU (Xeon)	288	4096	0.066	0.65	0.56
DeePMD	GPU (V100)	8	4096	0.017	2.54	2.16
MACE	GPU (A100)	1	4096	1.47	0.03	0.03
MACE	GPU (A100)	1	64	0.047	0.94	0.92
BPNN	CPU (Xeon)	768	4096	0.015	2.79	2.10

TABLE III. An overview of the expected throughput for *NVT* MD simulations of liquid water (with a time step of 0.5 fs) for different system sizes, hardware, number of processing units ($n_{\text{CPU}}/n_{\text{GPU}}$) and MLIPs. DeePMD simulation timings correspond to the highest level of parallelization in Fig. 4. The Simulations were run on nodes with 2 Intel(R) Xeon(R) Platinum 8360Y, with 36 cores each (CPU) and nodes with 2 NVIDIA V100 cards (GPU); MACE simulations use the ASE driver with the MACE-MP-0 model(S),^[9] single precision and were run on one A100 GPU card. Behler-Parrinello neural network (BPNN) simulations use the model from Ref. ^[129], and were run using the n2p2 interface in LAMMPS, on a Intel Xeon E5-2690 v3 @ 2.60GHz.

C. Near-ideal performance of i-PI with popular MLIPs

These experiments show that, in practical use cases, the overhead introduced by i-PI is negligible or small up to high levels of parallelization of the MLIP code – although one should reassess these conclusions depending on the hardware, the specific MLIP implementation, and the type of simulation performed. As shown in Fig. 2, a practical recipe to determine what throughput one can expect out of the combination of i-PI with your favourite machine learning code is to determine the overhead τ_O , either from Table I or by running a short simulation with a dummy driver. A dummy calculation is also suitable to tune the simulation parameters (and potentially to help remove bottlenecks from other simulation pathways – some examples of profiling runs can be found in the *test* directory). The cost of evaluating energy and forces, τ_{F} , also depends on the type of MLIP, system size and the

hardware of choice. Thus, by comparing τ_O and τ_F , it is possible to determine whether τ_O is significant.

As shown in Table III, for both small and medium-sized supercells, and for DeePMD, n2p2 and MACE models, a standard constant-temperature MD simulation is limited by τ_F . Even though, as shown by the case of LAMMPS, the overhead of a specific client might be larger than that of the minimalistic driver distributed with i-PI, this type of analysis helps determine whether i-PI is a suitable choice for the modeling task, or whether one should look for (or implement) a more performant, compiled-language implementation. In summary, using i-PI as a driver does not prevent reaching a few million energy evaluations per day, corresponding to several ns for typical MD time steps. In addition, it allows access to advanced simulation methods, such as path integral molecular dynamics, as well as facile interfacing with new MLIPs implementations, which might make i-PI the preferred choice even in cases in which it entails a small performance penalty.

IV. DESCRIPTION OF SELECTED FEATURES

A. Bosonic and Fermionic Path Integral Molecular Dynamics

Exchange effects, arising from bosonic and fermionic statistics, are important in simulations of phenomena such as superfluidity,[130] supersolidity,[131] and superconductivity. However, ordinary PIMD simulations assume that the atoms are distinguishable particles and obey Boltzmann statistics. The biggest challenge in going beyond this assumption and performing PIMD simulations of indistinguishable particles is including all the possible identical particle permutations in the partition function. These are taken into account by connecting rings of permuted particles to form larger polymers.[132] Since there are $N!$ permutations of N particles, the simulation, in principle, should sum over the spring forces from $N!$ configurations, which quickly becomes intractable.

Version 3.0 of i-PI supports PIMD simulations that include exchange effects, using an algorithm introduced by Hirshberg *et al.*[108, 109] In this approach, PIMD simulations of bosons scale quadratically,[109] rather than exponentially, with N . This is achieved by replacing the standard ring polymer potential with $V_B^{(N)}$, which is defined by the recurrence relation,

$$e^{-\beta V_B^{(N)}} = \frac{1}{N} \sum_{k=1}^N e^{-\beta (V_B^{(N-k)} + E_N^{(k)})}. \quad (1)$$

Here, $E_N^{(k)}$ is the spring potential of a ring polymer connecting the particles $N - k + 1, \dots, N$ in a cycle, and $V_B^{(0)} = 0$. This potential and the forces it induces are computed in quadratic time with N and in linear time with the number of beads.[109]

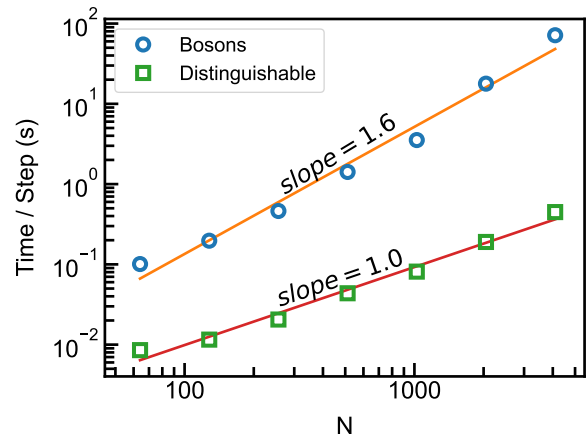


FIG. 5. Scaling with the number of particles of the propagation of the spring forces in Cartesian coordinates, for distinguishable and bosonic trapped, non-interacting systems. Both simulations were run with the `bab` propagator with `nmts=10`. Simulations were run on a node with two AMD EPYC 7763 Processors with 64 cores each, blocking the entire node for every run and using a single i-PI force-field client.

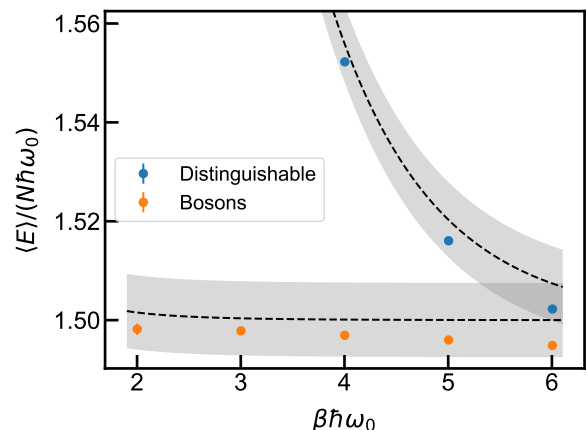


FIG. 6. The energy per particle of $N = 512$ trapped non-interacting bosons as a function of temperature (indicated by the dimensionless number $\beta \hbar \omega_0$, where ω_0 is the harmonic trap constant), with $P = 42$ beads. The dashed lines indicate the analytical results and the shaded areas indicate an error region of $\pm 0.5\%$.

The i-PI implementation allows the efficient simulation of systems composed of ~ 1000 particles. A comparison of the scalability of bosonic and standard PIMD simulations in i-PI is presented in Fig. 5, using the same integrator (`bab`, see below) in both cases, for a system of trapped non-interacting particles. The bosonic simulation has quadratic scaling while the distinguishable one scales linearly. For $N = 64$ particles, the cost of bosonic exchange effects is approximately 12 times that of standard PIMD, and for $N = 1024$ it is only 44 times slower with the same integrator.

Performing bosonic PIMD in i-PI is

straightforward and involves adding the following section to the xml input file:

```
<normal_modes propagator='bab'>
  <bosons>...</bosons>
  <nmts>M</nmts>
</normal_modes>
```

The bosonic atoms are listed using the `bosons` tag. This can be done as a list of atoms (zero-based), such as `<bosons id='index'> [0, 1, 2] </bosons>`, or by a list of labels, such as `<bosons id='label'> ['He'] </bosons>`. At the moment, the integration should be done in Cartesian coordinates (`normal_modes propagator='bab'`), because an analytic expression for the normal-modes propagator of the free ring polymer part of the Hamiltonian is not known for bosons. Because the frequencies of the springs are typically higher than the physical frequencies, the use of multiple-time stepping in the Cartesian integrator is recommended. [133] This is done by setting the `nmts` tag (standing for the number of iterations for each multiple-time stepping level) to `M`, where `M` is the ratio between the time step used for the physical forces and that used for the spring term.[19] In Fig. 6, we present the average energy of 512 trapped, non-interacting cold bosons. By comparing with distinguishable particle simulations, the relevance of exchange effects becomes evident.

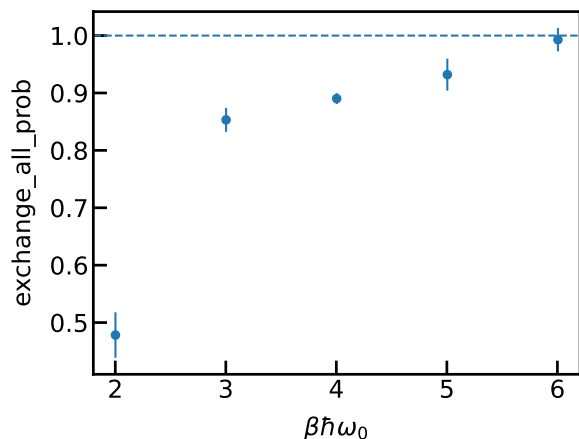


FIG. 7. The probability of the ring polymer configuration that connects all $N = 16$ non-interacting trapped particles at different temperatures (indicated by the dimensionless number $\beta\hbar\omega_0$, where ω_0 is the harmonic trap constant).

A way to assess the extent of exchange effects in bosonic simulations is to analyse the following probabilities, which can be accessed through the properties tag in i-PI: (i) `exchange_distinct_prob`, which is the probability of the ring-polymer configuration corresponding to distinguishable particles:

$$\frac{1}{N! \cdot e^{-\beta V_B^{(N)}}} e^{-\beta \left(\sum_{i=1}^N E_i^{(1)} \right)}, \quad (2)$$

where $E_i^{(1)}$ is the spring potential of the ring polymer connecting the beads of particle i in a separate ring. This quantity tends to 1 in the regimes where the exchange is negligible. (ii) `exchange_all_prob`, which is the probability of the ring-polymer configuration connecting *all* the bosons, scaled by $1/N$:

$$\frac{1}{e^{-\beta V_B^{(N)}}} e^{-\beta \cdot E_N^{(N)}}, \quad (3)$$

where $E_N^{(N)}$ is the spring potential of the ring polymer connecting all the particles $1 \rightarrow 2 \dots \rightarrow \dots \rightarrow N \rightarrow 1$ in one large ring. This quantity is high in regimes of effective exchange between all the particles. In Fig. 7, we show the average of `exchange_all_prob` in converged simulations of trapped cold noninteracting bosons as a function of temperature. This quantity tends to 1 in the limit $T \rightarrow 0$, $\beta \rightarrow \infty$, in which all permutations are equally probable. [108]

Several estimators require modifications when performing bosonic PIMD simulations. For example, the primitive estimator for the kinetic energy (`kinetic_td`) is modified according to Eq. 1 (as shown in Eq. 2 in the SI of Hirshberg *et al.* [108]) while the standard virial kinetic energy estimator can be used for bosons in confined systems. On the other hand, the centroid-virial kinetic estimator is not applicable due to the lack of an appropriate centroid definition.

Lastly, fermionic statistics can be obtained by first performing bosonic simulations and then reweighting the results.[110, 134, 135] This is done by using the fermionic sign estimator (the `fermionic_sign` property of i-PI). For every position-dependent operator, the corresponding fermionic estimator is obtained from the bosonic estimator by multiplying with the fermionic sign, and dividing by the average fermionic sign in the simulation (see Eq. (9) of Ref. [110]). The average sign is always positive but for simulations at low temperatures or with large N , where the sign problem is severe, it gets close to zero. Fig. 8 illustrates the behaviour of the fermionic energy estimator for $N = 3$ non-interacting, trapped fermions at several temperatures. While the distribution of the bosonic energy estimator throughout the simulation is always positive, the fermionic energy estimator changes sign, its distribution is bi-modal around $E = 0$, and more symmetric at lower temperatures. The near-complete cancellations of positive and negative contributions make the mean fermionic energy harder to converge in comparison to the bosonic case.

B. Nuclear propagation by additional dynamical variables

i-PI 3.0 can employ numerical quantities provided by the driver (beyond the usual forces and stress arrays) to run advanced types of dynamics. This supplementary communication is achieved through an additional string

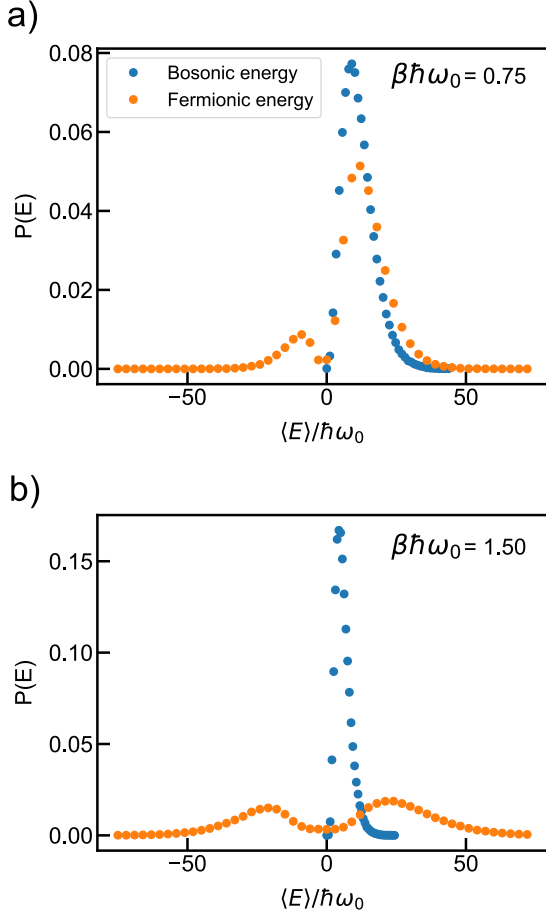


FIG. 8. The distribution of bosonic and fermionic energy estimator, $P(E)$, for $N = 3$ trapped non-interacting particles at two representative temperatures indicated by the dimensionless number $\beta\hbar\omega_0$, where ω_0 is the harmonic trap constant: (a) high temperature exhibiting mild cancellations between the two peaks of the bimodal distribution of the fermionic energy estimator, and (b) low temperature where the cancellation is near-complete and the sign problem is more severe.

that can be passed in the `extras` field of `i-PI` within the same socket communication. When such string is JSON-formatted it is automatically parsed, and the additional quantities returned by the driver are made available in the code and accessible as a dictionary. More importantly, when the data is numeric, it is added as a property of the force object, guaranteeing a seamless integration with other `i-PI` features, and becoming available to be used in any type of algorithm. In the next section, we present two examples of advanced simulations that employ electronic structure quantities as “additional” variables to modify the atomic forces and therefore alter the nuclear dynamics.

1. Non-adiabatic tunnelling rates in metallic systems

The vanishing band gap in metals allows nuclear movements to induce electronic excitations, even at relatively low velocities.[136, 137] These effects are not captured by the Born-Oppenheimer approximation and, therefore, are collectively referred to as non-adiabatic effects (NAEs). NAEs in metals are qualitatively different from the ones found in molecules due to the continuous manifold of available empty electronic states present in the former, and thus, they require a different set of theoretical approaches. A frequently used technique to simulate these types of NAEs is to decorate the otherwise Born-Oppenheimer dynamics with an electronic frictional kernel and a stochastic force.[138–140] The electronic friction formalism has been derived from first principles starting from the exact factorization of the wave function, initially at 0 K,[141, 142] and more recently at finite temperatures.[143] Moreover, in the case of a theory of non-interacting electrons, such as DFT, the friction kernel can be computed explicitly from *ab initio* calculations.[144]

When studying the kinetics of light atom diffusing in metals, nuclear quantum effects (NQEs) can become as important as NAEs. To address this complex problem some of the authors have recently derived a theoretical framework that can capture both types of effects.[113] The method called ring polymer instanton with explicit friction (RPI-EF) was derived by combining the semiclassical instanton rate theory in its path integral discretized form,[145] with the *ab initio* electronic friction formalism. The framework is general enough to incorporate the spatial and frequency dependence of the friction tensor and, owing to the computational efficiency of the parent methods, [146, 147] it is suitable for first-principles calculations of reaction rates in complex, high-dimensional systems.[114]

The RPI-EF method is algorithmically similar to classical Eyring transition rate theory and requires the search of a first-order saddle point. In this case, however, the saddle point is not the reaction transition state but rather an imaginary time trajectory that makes the Euclidean action stationary. This special trajectory is known as instanton and can be interpreted as the main tunnelling pathway.[148] In the limiting case of a position-independent friction kernel (see a more general expression including the position dependence of the friction kernel in Ref. [113]) the discretized Euclidean action, S_P can be expressed as

$$S_P = S_P^{\text{sys}} + \hbar \frac{\beta}{P} \sum_{l=-P/2+1}^{P/2} \sum_{i=1}^{3N} \frac{\tilde{\eta}(\omega_l)\omega_l}{2} (Q_i^{(l)})^2, \quad (4)$$

where P is the number of discretization points or beads, N is the number of atoms, β is the inverse temperature, $Q^{(l)}$ represents the free ring-polymer normal mode coordinates, ω_l are the free ring-polymer (RP) normal mode frequencies, and $\tilde{\eta}$ is the Laplace transform of the

(electronic) friction kernel. S_P is related to the potential energy of the ring polymer U_P by $S_P/\hbar = \beta_P U_P$. The RPI-EF calculation consists of a series of geometry optimizations on U_P using progressively more discretization points until the converged instanton pathway is achieved. A detailed description of the individual steps involved in RPI and RPI-EF simulations, as well as practical guidelines, can be found elsewhere.[114, 149, 150]

The i-PI implementation of RPI-EF builds upon the RPI implementation from the previous i-PI release. The primary distinction regards the additional terms that contain the friction kernel. In the case of position-dependent friction, these terms require updates at each step of the geometry optimization, which has been made possible by the automatic parsing of JSON-formatted data returned by the client. The friction terms are enabled by setting the friction flag inside the instanton block, `frictionSD`. Since the theory assumes that the frequency dependence of the friction kernel is the same for all positions, the user needs to provide the normalized spectral density, $\tilde{\eta}(\omega)$, that can be specified with the flag `fric_spec_dens` and the energy at which it has been normalized to one by using the flag `fric_spec_dens_ener`. A minimal example of the section `motion` to run RPI-EF simulations with i-PI is

```
<motion mode='instanton'>
  <instanton mode='rate'>
    <friction>true</friction>
    <fric_spec_dens mode='file'>
      fric_spec_dens.dat
    </fric_spec_dens>
    <fric_spec_dens_ener> 0.1
  </fric_spec_dens_ener>
  <hessian_update>powell</hessian_update>
</instanton>
</motion>
```

In Fig. 9, we present the instanton geometry obtained for a 1D double well potential without including any friction, including a constant friction profile, and including an analytical position-dependent friction profile. The inclusion of constant friction causes the particle to behave more classically, resulting in the shrinkage of the instanton geometry when compared to the calculation without friction. The result with position-dependent friction shows that the instanton geometry avoids regions with large friction. As discussed in Refs. [114, 151], this indicates that non-adiabatic effects can steer the main tunnelling pathway and, consequently, steer quantum dynamics.

2. Dynamics with time-dependent external field

Recent experiments have shown that the excitation of infrared-active phonons through THz laser pulses leads to a vast phenomenology in different kinds of materials.[152] These phenomena are non-thermally driven by micro-

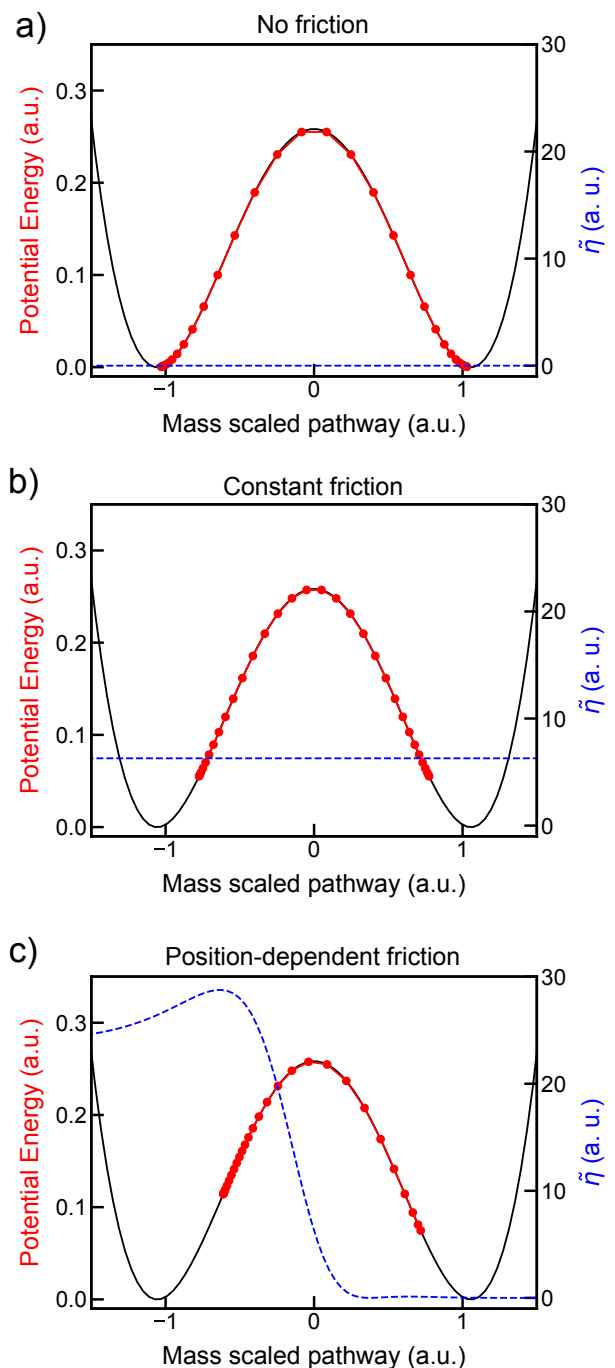


FIG. 9. Instanton pathways obtained in double-well potential with a) no friction, b) constant friction and c) position-dependent friction. The instanton pathways, potential energy surface and friction profiles, $\tilde{\eta}$, are depicted as red dotted lines, black solid lines and blue dashed lines, respectively. The analytical expression of the potential and friction profiles can be found in Ref. [114].

scopic processes rather than originating from ultrafast heating effects.[153] The possibility for driving materials, ranging from traditional solids to liquids and interfaces, across phase transitions and into metastable states that are not accessible at thermal equilibrium, has opened the possibility of dynamical materials design.[154, 155]

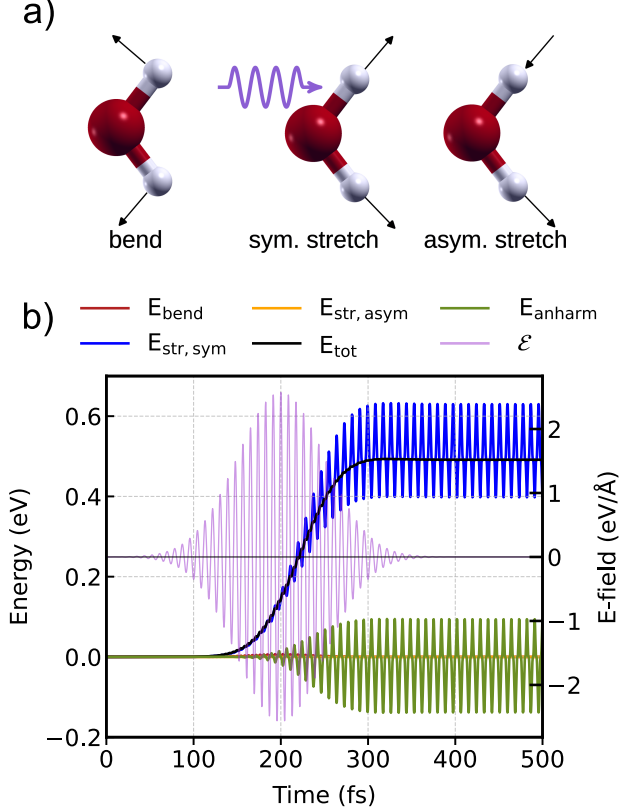


FIG. 10. (a) Vibrational modes of a water monomer: bending E_{bend} , symmetric-stretch $E_{\text{str, sym}}$, and asymmetric-stretch $E_{\text{str, asym}}$. (b) Contributions to the total energy along a driven trajectory. Time series of the harmonic contribution to the energy of different vibrational modes, the total energy E_{tot} , and the anharmonic contribution E_{anharm} of a water monomer along a driven trajectory. The $E_{\text{str, sym}}$ mode (blue) is resonant with the applied electric field \mathcal{E} (shown in purple). The contributions from $E_{\text{str, asym}}$ (yellow) and E_{bend} (red) remain close to zero throughout.

Many of these phenomena rely on the existence of anharmonic coupling between different vibrational modes that lead to the so-called nonlinear effects in vibrational dynamics.[156, 157] A popular approach to simulate these experiments and capture the associated nonlinear dynamics is based on the electric dipole approximation (EDA). Within EDA, one can assume that the changes induced by the electric field \mathcal{E} on the electronic structure are not significant, and expand the electric enthalpy function of a system around the vanishing-field condition up to first order. This leads to the following expression for the (external field dependent) potential

energy

$$\mathcal{U}(\mathcal{E}, \mathbf{q}) \simeq V(\mathbf{q})|_{\mathcal{E}=0} - \mathcal{E} \cdot \boldsymbol{\mu}(\mathbf{q}). \quad (5)$$

where $\boldsymbol{\mu}$ is the dipole moment of the system in the absence of the electric field and \mathbf{q} represents the atomic positions. Eq. 5 can be evaluated with a variety of theories. For example, when employing DFT, evaluating the expression above would require a full self-consistent convergence of the Kohn-Sham (KS) energy and the dipole corresponding to this converged electronic density. In periodic systems, the dipole would have to be expressed in terms of the system polarization, calculated with the Berry-phase formalism.[158]

Assuming a time-dependent electric field, $\mathcal{E}(t)$, the effect of the external field manifests itself on the nuclear dynamics through the nuclear forces, \mathbf{F} ,

$$\mathbf{F} = -\frac{\partial \mathcal{U}(\mathcal{E}, \mathbf{q})}{\partial \mathbf{q}} = -\frac{\partial V(\mathbf{q})}{\partial \mathbf{q}} \Big|_{\mathcal{E}=0} + \mathcal{E}(t) \cdot \frac{\partial \boldsymbol{\mu}}{\partial \mathbf{q}}, \quad (6)$$

where the first term corresponds to the usual forces in the absence of an external field. The second term can be conveniently expressed in terms of the *dynamical charges*, or Born effective charges (BEC) $Z_{\alpha j}^*$, defined as,

$$Z_{\alpha j}^* = \frac{1}{e} \frac{\partial \mu_{\alpha}}{\partial q_j}, \quad (7)$$

where e is the elementary charge (1 in atomic units) $\alpha = x, y, z$ runs through the Cartesian components of the total dipole and $j = 1, \dots, 3N$ runs through the degrees of freedom of the system. With this definition, the last term in Eq. 6, henceforth \mathbf{F}^{ext} , reads

$$\mathbf{F}_j^{\text{ext}}(t) = \sum_{\alpha=x,y,z} \mathcal{E}_{\alpha}(t) Z_{\alpha j}^*. \quad (8)$$

For a fixed atomic configuration, the BEC can be computed by finite differences of the dipole values with respect to the atomic positions. In the context of DFT, it is also possible to exploit density functional perturbation theory (DFPT).[159, 160]

The i-PI implementation for driven vibrational motion is based on the definition of a time-dependent integrator for the equations of motion, called `EDAIIntegrator`, built on top of the existing integrators. The role of this integrator is simply to add $\mathbf{F}^{\text{ext}}(t)$ to the momenta within the Verlet algorithm. This was implemented in a new *motion* class of i-PI, called `DrivenDynamics`, which inherits all the attributes and methods of the *Dynamics* class.

We implemented a plane-wave linearly-polarized electric field with a Gaussian envelope function, of the following form,

$$\mathcal{E}(t) = \mathcal{E}_{\text{amp}} \cos(\omega t) \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (9)$$

where ω is the frequency of the plane-wave, \mathcal{E}_{amp} is the polarization vector of the electric field, and m and σ^2 are the mean and variance of the Gaussian envelope function, respectively. Other types of field profiles can be added straightforwardly.

The BECs and dipoles are communicated by the client at every new nuclear configuration through the `i-PI extras` field as a JSON formatted string. An example of the communication of such a client is provided in the Python driver code distributed with `i-PI` (see `driverdipole`). Alternatively, BECs can be provided as a `txt` file and kept fixed during the dynamics.

An example of the section `motion` to run dynamics driven by a Gaussian-enveloped electric field with fixed BECs reads

```
<motion mode='driven_dynamics'>
<driven_dynamics mode='eda-nve'>
  <timestep units='femtosecond'> 0.3 </timestep>
  <efield>
    <amp units='atomic_unit'> [0,5e-2,0] </amp>
    <freq units='thz'> 115 </freq>
    <peak units='picosecond'> 0.2 </peak>
    <sigma units='picosecond'> 0.05 </sigma>
  </efield>
  <bec mode='file'> bec.txt </bec>
</driven_dynamics>
</motion>
```

In this example, the `amp`, `freq`, `peak`, and `sigma` tags refer to \mathcal{E}_{amp} , ω , m and σ as defined in Eq. 9, respectively. The file `bec.txt` contains a 2D array of shape $(3N, 3)$. The columns correspond to the Cartesian components of the dipole and the rows to the degrees of freedom j in Eq. 7. The current implementation only supports the NVE ensemble, but can be easily combined with other ensembles.

As a benchmark example, a single water monomer has been driven using fixed BECs and the Partridge-Schwenke water model.[161] The parameters adopted for the electric field are the ones shown in the input example above. The resulting trajectory, initialized with vanishing velocities has then been projected along the 3 vibrational modes of the system, which have been computed using the harmonic vibrational analysis provided by `i-PI`. The vibrational modes are shown in Fig. 10a. The oscillating frequency of the field at around 3836 cm^{-1} (115 THz) was chosen to be resonant with the symmetric-stretch mode.

Fig. 10b shows the energy given by the sum of the three harmonic vibrational mode energies (E_{bend} , $E_{\text{sym.stretch}}$, $E_{\text{asym.stretch}}$). E_{anharmon} is computed as the difference between the total energy and the sum of the harmonic energies. The profile of the applied field is also shown in Fig. 10. As expected, the resonant vibrational mode is excited and the energy of the system increases while the driving field assumes non-negligible values. Afterwards, the field decays to zero but the system remains excited due to the lack of a dissipation mechanism. By comparing the different components of the energy, it is

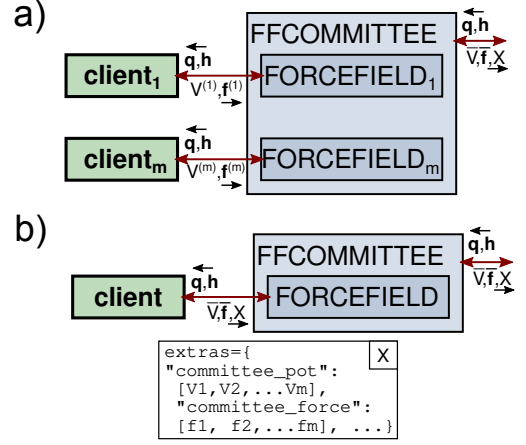


FIG. 11. Schematic representation of the implementation of the `FFCommittee` class, that combines the members of an ensemble model to evaluate the mean potential and force, that are used to propagate the dynamics. The statistics of the ensemble are stored in a JSON-formatted `extras` string, depicted in the figure with the `X` label. This class can handle two types of models. The first one, depicted in panel a, works by collecting data from one forcefield per committee member, which is suitable for fully independent models. Alternatively, a single forcefield can compute all ensemble members, returning them to `i-PI` as a JSON string with the same syntax (panel b), which is then used to compute the error. This second mode is particularly suitable for shallow models with a high degree of weight sharing.

evident that the harmonic contribution is dominated by the symmetric-stretch mode and that in this simple example the anharmonicity plays a relatively minor role.

C. Ensemble models and error estimation

This version of `i-PI` implements error estimation for MLIPs based on ensembles. The idea is that, for each configuration A , multiple potentials $V^{(k)}(A)$ are computed, together with the associated forces. The simulation is then evolved based on the mean of the ensemble $\bar{V}(A) = \frac{1}{n_{\text{ens}}} \sum_k V^{(k)}(A)$, where n_{ens} is the number of potentials in the ensemble, and the error is estimated as the standard deviation $\sigma^2(A)$. While, in general, ensemble approaches entail a large computational overhead because n_{ens} potentials have to be evaluated at each time step, there are cases in which this overhead can be largely avoided, as e.g. when employing linear or kernel methods,[162] or when applying a high-level of weight sharing between different neural networks e.g. using a “shallow-ensemble” on the last-layer.[115, 163–167] `i-PI` provides two alternative modes to simulate nuclear (thermo)dynamics including ensemble-based uncertainty estimates, depending on the characteristics of the underlying framework.

The first entails independent models that are run as

separate clients, by specifying multiple socket connectors as part of a `<ffcommittee>` tag. For instance,

```
<ffcommittee name='my_committee'>
  <ffsocket name='driver-1'>
    [...]
  </ffsocket>
  <ffsocket name='driver-2'>
    [...]
  </ffsocket>
  [...]
</ffcommittee>
```

The associated class takes care of collecting potentials, forces and stresses from the various sockets and averaging them. Even though a `ffcommittee` section contains multiple sockets, one should use a single `<force>` block referring to the name of the committee (`my_committee` in this example) in the definition of the system. A linear committee calibration scaling [162] can also be applied with the keyword `<alpha>`.

In the second mode, which is suitable when using a shallow ensemble model with weight sharing, a single client computes all the ensemble members. In this case, the client passes the average energies, forces and virials to i-PI in the standard way and also passes the ensemble information as a JSON string into the `extras` field. The string contains the list of potentials, forces, and virials into fields named `committee_pot`, `committee_force`, `committee_virial`. We provide a reference implementation of such a client in the FORTRAN driver distributed with i-PI. It is then sufficient to specify a single client in the committee section

```
<ffcommittee name='my_committee'>
  <ffsocket name='driver'>
    [...]
  </ffsocket>
  <parse_json> True </parse_json>
</ffcommittee>
```

and to activate the flag `parse_json` that instructs the committee class to parse the ensemble data out of the string. Also in this case, the `forces` section should refer to "my_committee" and not to the name of the inner forcefield.

For both modes, committee potentials, forces, as well as the computed uncertainty, are collected into an *extras* dictionary corresponding to the same schema used for the JSON string. The different diagnostics can then be printed out using the generic output features, e.g. specifying `extra_type='committee_pot'` to print the energies of all members of the ensemble.

The committee forcefield implements a rudimentary active-learning mechanism. By specifying an output filename and an error threshold, every structure with a predicted uncertainty above the threshold encountered at any point of a calculation will be stored so it can be used to improve the MLIP model in an iterative manner. It is also possible to build a *weighted baseline* model,[107]

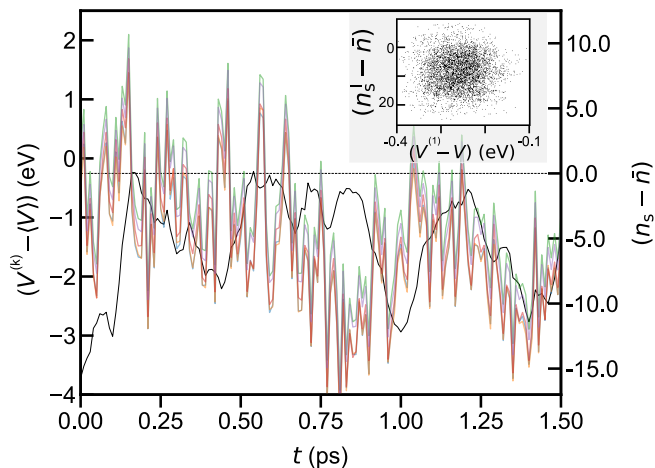


FIG. 12. Trajectories of the various energy predictions of the DPOSE [115] committee model (plotted in contrasting colors) and of the order parameter that determines the pinning potential, Eq. 11 (black). The inset shows the near-Gaussian correlation plot between the order parameter and the deviation of one of the ensemble members from the mean potential.

in which one forcefield, whose name is specified in the `<baseline_name>` flag, is interpreted as a reference potential, to which the model described by a committee is added as a correction. The baseline is assumed to have a constant uncertainty σ_{base} , that can be set with the option `<baseline_uncertainty>`. Rather than simply computing the overall potential as $V = V_{\text{base}} + \bar{V}$, the weighted baseline potential is computed as

$$V(A) = V_{\text{base}}(A) + \frac{\sigma_{\text{base}}^2}{\sigma_{\text{base}}^2 + \sigma^2(A)} \bar{V}(A) \quad (10)$$

which uses the correction when the predicted error is smaller than the baseline error, and switches to the more reliable (although less accurate for structures that are well-predicted by the committee) baseline potential.

1. Computing the solid-liquid chemical potential for water

We discuss a representative example of the use of `ffcommittee` to compute errors of the solid-liquid chemical potential $\Delta\mu$ for an ice/water system, that showcases several of the advanced features in i-PI. This type of simulation can be used to estimate the melting point of water T_m , by repeating the procedure at different temperatures and finding the temperature at which $\Delta\mu = 0$. As shown in Ref. 107, once a collection of $\Delta\mu$ values associated with the different ensemble members has been obtained, it is easy to determine the error on T_m by ensemble propagation.

For this example, we use a Behler-Parrinello-style neural network,[168] based on smooth-overlap of atomic positions (SOAP)[169] features and trained on the dataset

of Ref. 14. The model generates an ensemble of 5 energy predictions, differing only by the last-layer weights and trained by optimizing the negative log-likelihood following the direct propagation of shallow ensembles (DPOSE) protocol.[115] We run a simulation with 336 water molecules, arranged in an elongated simulation cell containing coexisting water and ice. A restraining *pinning potential* to an anchor point, \bar{n} ,

$$V_{\text{pin}}(A) = \frac{1}{2}k(n_S(A) - \bar{n})^2 \quad (11)$$

is applied using a suitable collective variable that estimates the number of molecules in a solid-like environment, $n_S(A)$, [170] as implemented in PLUMED[39] and here used through its interface with i-PI. At the melting point, the potential restrains the solid fraction to the target value \bar{n} , whereas away from equilibrium the mean deviation is proportional to the solid-liquid molar free-energy difference, $\Delta\mu \propto \langle n_S(A) - \bar{n} \rangle$. Further details on the setup can be inferred from the example files included in the `demos/ensemble-deltamu` directory. It should be noted that both size and duration of the simulations are insufficient to obtain a converged estimate of the melting point, as this is only intended as a demonstrative example.

Fig. 12 shows a small segment of a trajectory obtained with the DPOSE model[115]. The mean potential undergoes thermal fluctuations, and the ensemble of predictions has a varying spread along the trajectory. Similar to what was observed in Ref. 115, the members of the ensemble differ mostly by a rigid shift, reflecting a systematic bias in the prediction of the potential. Meanwhile, the order parameter fluctuates about a value that is slightly below the target, indicating that the simulation temperature (290 K) is above the melting point for this water model – at least for this under-converged supercell size. There is no obvious pattern in the correlation between the total potential energy and the order parameter: at this temperature, and for such a comparatively small system, random fluctuations dominate over the enthalpy change associated with the changes in the solid fraction. Importantly, the cross-correlations between the solid fraction and the ensemble potential shift ($V^{(k)}(A) - \bar{V}(A)$) show approximately Gaussian behaviour (see inset in Fig. 12). This simplifies the estimation of the average deviation $\langle (n_S(A) - \bar{n}) \rangle$ from a single trajectory driven by \bar{V} . The basic idea is to use Torrie-Valleau reweighting,[171] i.e. weighting each snapshot by $e^{-\beta(V^{(k)}(A) - \bar{V}(A))}$.

As shown in Fig. 13, the cumulative averages computed with the different weighting factors converge to different values – providing an ensemble of predictions for $\langle n_S - \bar{n} \rangle_k$. The sharp, sudden changes in the cumulative average reflect the presence of very large weighting factors, that occur whenever the deviations of individual predictions from the mean potential are large relative to $k_B T$. The resulting statistical inefficiency[101] would become worse as the system size is increased, eventually

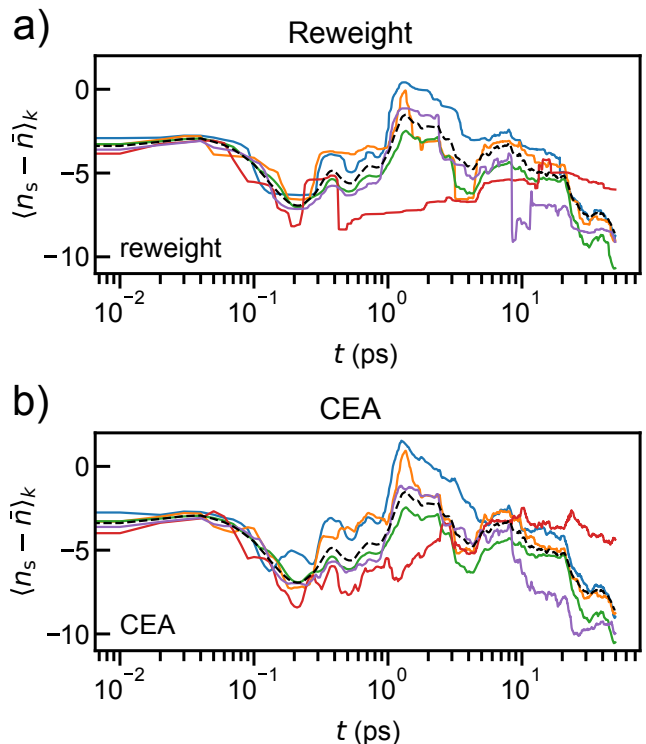


FIG. 13. Cumulative sums of the reweighted mean $\langle n_S - \bar{n} \rangle_k$ computed using the k -th ensemble member (plotted in contrasting colors). The dashed black line indicates the cumulative average of n_S computed without reweighting. Panel a) shows the average computed with Boltzmann reweighting while panel b) uses a statistically-better-behaved cumulant expansion approximation (CEA).

making the procedure completely ineffective. Following Ref. 107, one can use instead a cumulant expansion approximation (CEA) that estimates reweighted average under the assumption that the target observable and the potential are joint Gaussians, and yields

$$\langle n_S \rangle_k \approx \langle n_S \rangle - \beta [\langle n_S (V^{(k)} - \bar{V}) \rangle - \langle n_S \rangle \langle (V^{(k)} - \bar{V}) \rangle]. \quad (12)$$

This approximation introduces a systematic error because the distribution is *not* exactly Gaussian, but as can be seen in Fig. 12, the qualitative behaviour reflects the one seen for the direct reweighting. The statistical error is well-behaved also for large simulations, making the CEA a more generally applicable strategy. We provide a commented example to perform this kind of simulation in the `demos/ensemble-deltamu`; the post-processing needed to evaluate the reweighted averages is implemented in the `i-pi-committee-reweight` script.

By repeating the simulation at different temperatures one can determine the temperature dependence of $\Delta\mu$ for each ensemble member separately, and use the separately-computed values of T_m to estimate the error in the determination of the melting point. It is worth noting that in Ref. 107 the melting point was determined to be 290 ± 5 K, with a setup similar to the one discussed here,

but using a deep ensemble of models, fitted on the same dataset. This estimate is consistent with our observation that $\langle n_s - \bar{n} \rangle$ at 290 K is very close to zero. The discrepancy with Ref. 14, where the classical melting point for the reference DFT was determined by free-energy perturbation to be at ≈ 275 K can be attributed to finite-size effects, that tend to destabilise the liquid and increase the melting point.

D. Cavity Molecular Dynamics for Polaritonics

For molecules confined in optical cavities, molecular transitions may form polaritons, hybrid light-matter states with cavity photon modes.[172–176] Of particular interest is the vibrational strong coupling regime,[177, 178] in which vibrational polaritons are formed when a molecular vibrational mode is near resonant with an IR cavity photon mode. Under vibrational strong coupling, many molecular properties, including chemical reaction rates and energy transfer rates, can be significantly modified in the liquid phase.[179–181] For a better understanding of these exciting experiments, both the molecules and the cavity photons need to be propagated self-consistently with realistic parameters. This requirement is beyond the scope of standard molecular dynamics schemes, whereby the time-dependent dynamics of the photon field are not explicitly propagated.

The 3.0 version of the i-PI package provides an efficient method for simulating vibrational strong coupling: the CavMD approach.[111, 112] Within the framework of CavMD, both nuclei and cavity photons in the IR domain are treated classically, propagated by the following light-matter Hamiltonian under the ground electronic state: $H_{\text{QED}} = H_{\text{M}} + H_{\text{F}}$, [111, 182] where H_{M} denotes the standard molecular Hamiltonian composed of the kinetic and potential terms, and

$$H_{\text{F}} = \sum_{\mathbf{k}, \lambda} \frac{\tilde{p}_{\mathbf{k}, \lambda}^2}{2m_{\mathbf{k}, \lambda}} + \frac{1}{2} m_{\mathbf{k}, \lambda} \omega_{\mathbf{k}, \lambda}^2 \left(\tilde{q}_{\mathbf{k}, \lambda} + \frac{\tilde{\varepsilon}_{\mathbf{k}, \lambda} d_{\lambda}}{m_{\mathbf{k}, \lambda} \omega_{\mathbf{k}, \lambda}^2} \right)^2. \quad (13)$$

Here, $\tilde{p}_{\mathbf{k}, \lambda}$, $\tilde{q}_{\mathbf{k}, \lambda}$, $\omega_{\mathbf{k}, \lambda}$, and $m_{\mathbf{k}, \lambda}$ denote the momentum, position, frequency, and auxiliary mass for the cavity photon mode defined by a wave vector \mathbf{k} and polarization direction $\boldsymbol{\xi}_{\lambda}$ with $\mathbf{k} \cdot \boldsymbol{\xi}_{\lambda} = 0$. For example, when the cavity is placed along the z direction, $\lambda = x, y$. d_{λ} is the electronic ground-state dipole moment for the molecular system projected along the direction of $\boldsymbol{\xi}_{\lambda}$, and $\tilde{\varepsilon}_{\mathbf{k}, \lambda}$ represents the effective light-matter coupling strength.

The basic implementation strategy of CavMD is as follows. The cavity photon modes are stored in i-PI as **L** (light) atoms. The coordinates and velocities of both the nuclei and the cavity photon modes are propagated in i-PI. A new *FFCavPhSocket* class, which inherits from the original *FFSocket* class, evaluates the forces of both the nuclei and the cavity photon modes. Similar to *FFSocket*, *FFCavPhSocket* calls external client codes to evaluate the

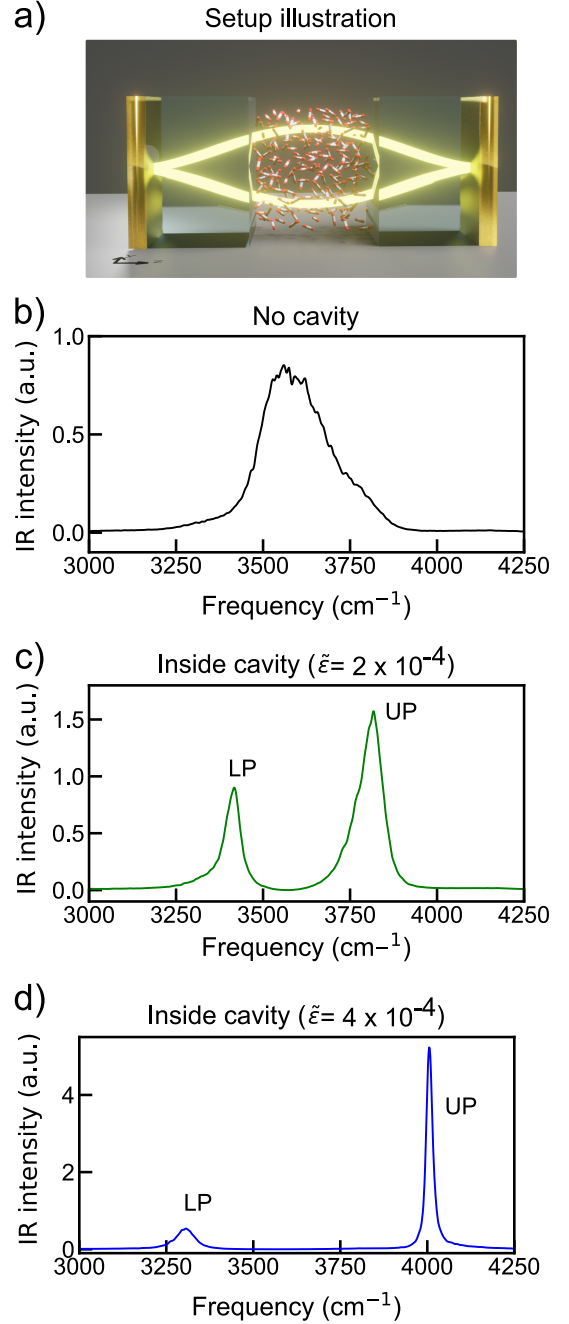


FIG. 14. CavMD simulations: Panel (a) illustrates the simulation setup of CavMD, whereby a large ensemble of molecules is confined in an optical cavity under vibrational strong coupling. The rest of the panels are adapted from Ref. 111, with permission from the authors, and show the IR spectra of the OH stretch mode of liquid water without a cavity and inside a cavity with different light-matter coupling strength, $\tilde{\varepsilon}$. LP and UP stand for lower and upper polariton, respectively.

nuclear forces associated with H_M , via the socket interface. After obtaining this component of the total forces, *FFCavPhSocket* performs additional calculations to obtain the photonic forces and cavity modifications on the nuclear forces as prescribed by Eq. 13. Because this implementation encapsulates the cavity-related calculations within the *FFCavPhSocket* class, many advanced features in i-PI, including RPMD, [111] and ring-polymer instantons, can be directly applied to study vibrational strong coupling without additional modifications of the i-PI code.

The CavMD implementation was used to simulate the IR spectra of liquid water under vibrational strong coupling.[111] The key i-PI input section to simulate vibrational strong coupling is as follows:

```
<ffcavphsocket name='lammps' mode='unix'>
  <address>h2o-cl-cavmd</address>
  <apply_photon> True </apply_photon>
  <E0> 4e-4 </E0>
  <omega_c units='inversecm'>
    3550.0
  </omega_c>
  <charge_array> [...] </charge_array>
</ffcavphsocket>
```

In the *FFCavPhSocket* section above, a cavity mode with frequency $\omega_c = 3550 \text{ cm}^{-1}$ was coupled to liquid water with an effective light-matter coupling strength $\tilde{\epsilon} = 4 \times 10^{-4}$ a.u. Under the default setting, the cavity mode is polarised along both the x and y directions. The partial charges of all molecular atoms are given in the *charge_array* section of *FFCavPhSocket*, and are used to evaluate the molecular dipole moment in Eq. (13). In Fig. 14 we show a schematic picture of the setup along with the IR spectra of water inside and outside the cavity. As shown in Fig. 14c, via classical CavMD simulations, the wide OH band of liquid water splits into a pair of polariton states, the upper polariton (UP) and the lower polariton (LP). Beyond this study, CavMD was also used to explore novel energy transfer and chemical reaction pathways under vibrational strong coupling. [183–185]

V. SHOWCASE OF ADVANCED SIMULATION SETUPS

A. Path Integral Coarse-Grained Simulations

While PIMD has demonstrated success in incorporating quantum nuclear motion for equilibrium properties, extracting accurate dynamical information from path-integral simulations is a longstanding challenge [186]. For instance, in estimating the vibrational spectra, established path-integral approximations to quantum dynamics such as with RPMD [90], CMD [92, 93] and variants [75, 94, 187–189], present artefacts such as unphysical shifted or broadened peaks.[94, 190] In addition, the computational overhead of path-integral simulations and

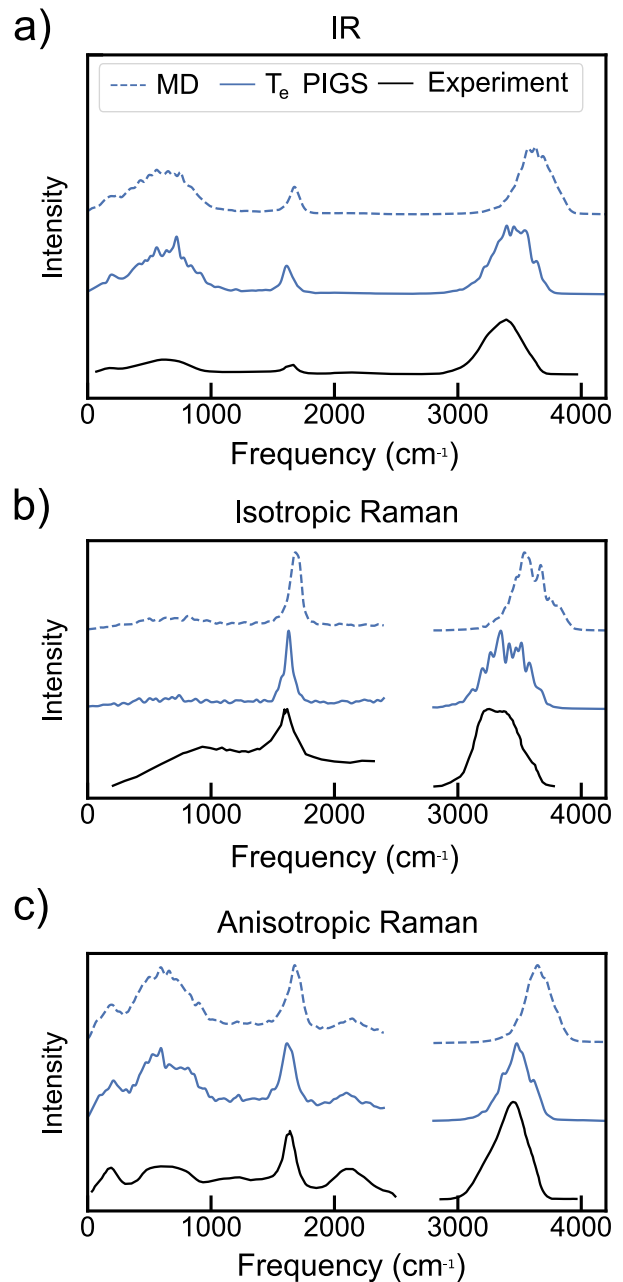


FIG. 15. IR, isotropic Raman, and anisotropic Raman vibrational spectrum of water at 300 K using the T_e PIGS method. Adapted from Ref. 38 with permission (CC BY 4.0 Licence).

the impact of the artefacts become more pronounced with lowering temperature.

Musil *et al.* [120] recently introduced a machine learning framework to significantly enhance the accuracy and efficiency of approximating quantum dynamics based on CMD. Their approach, referred to as T_e PIGS, has two components. The PIGS method is introduced to estimate an effective potential with an MLIP, which represents the centroid potential of the mean force, to drive the dynamics of the centroid. Second, the T_e *ansatz* estimates

the potential of mean force at an elevated temperature to effectively address the “curvature problem” of standard CMD [190]. This *ansatz* attenuates the spurious redshift in the centroid-based vibrational spectra of high-frequency quantized modes and results in a temperature-transferable effective potential for approximate quantum dynamics. Combining these components, the T_e PIGS method yields accurate modeling of vibrational spectra of large systems via inexpensive classical MD on an effective centroid MLIP. Together with an accurate potential energy surface, we observe quantitative agreement with experiments for IR, Raman and sum-frequency generation vibrational spectra in bulk and interfacial water [121].

This approach is easily implemented using the flexible functionality of i-PI as demonstrated for the calculation of accurate IR and Raman spectra of liquid water (see Fig. 15) using the MACE-OFF23(S) general organic MLIP [38] and an ML model for the polarization and polarizability of bulk water.[121] A fully documented example with scripts to generate the spectra is provided in the i-PI *demos* directory. For a given potential energy surface $V(\mathbf{q})$, with \mathbf{q} the Cartesian positions, the T_e PIGS ML centroid potential of mean force can be trained using a single high-temperature PIMD simulation. The training data includes the centroid positions $\bar{\mathbf{q}} = \sum_{j=1}^P \mathbf{q}^{(j)}/P$ with $\mathbf{q}^{(j)}$ the position of the j -th replica, the instantaneous estimator of the mean centroid force $\bar{\mathbf{f}}_c = \sum_{j=1}^P \mathbf{f}^{(j)}/P$, where $\mathbf{f}^{(j)}$ is the force acting on the j -th replica, and the physical force acting on the centroid $\bar{\mathbf{f}}_p = -\partial V(\bar{\mathbf{q}})/\partial \bar{\mathbf{q}}$. The elevated temperature was set to $T_e = 500$ K, and the centroid potential of mean force was fit from a 10 ps long path integral simulation with $P = 8$, with the centroid positions and forces sampled every 10 fs (20 timesteps). The centroid positions and instantaneous forces ($\bar{\mathbf{q}}$ and $\bar{\mathbf{f}}_c$) can be printed out as follows,

```
<trajectory filename='xc' stride='20'
  format='xyz' cell_units='ase'>
  x_centroid{ase}
</trajectory>
<trajectory filename='centroid_force'
  stride='20' format='ase'>
  f_centroid
</trajectory>
```

The quantity $\bar{\mathbf{f}}_p$ can be calculated in a post-processing step with a “replay” simulation, which allows the calculation and output of properties on a list of centroid positions. Alternatively, the physical force on the centroid can be printed on the fly by defining a “dummy” force component with a zero weight that receives centroid positions using,

```
<forces>
[... ]
<force forcefield='maceoff23' weight='0'
  nbeads='1'>
</force>
</forces>
```

Printing out this force component requires a single line:

```
<trajectory filename='physical_force'
  stride='20' format='ase'>
  forces_component_raw(1)
</trajectory>
```

With this data, a force matching method can be used to obtain the T_e PIGS MLIP by regressing the centroid force against the centroid position. This can be done transparently using standard MLIP packages like *n2p2* [191], *nequip* [192], *mace* [193], by replacing physical positions and forces, as is typically done while training MLIPs, with positions and forces associated with the centroid. To simplify the training problem, we advise to learn the difference between $\bar{\mathbf{f}}_c$ and $\bar{\mathbf{f}}_p$. The scripts necessary for dataset generation, curation and training with *mace* are provided in the i-PI *demos* directory.

After the MLIP is trained in this fashion, performing a T_e PIGS simulation is as simple as running classical molecular dynamics on the sum of the potential energy surface $V(\mathbf{q})$ and the T_e PIGS MLIP. This can also be done in i-PI by exploiting again the modularity of forces. We only need to define two forcefield sockets for the potential energy surface and the T_e PIGS MLIP:

```
<ffsocket name='maceoff23' mode='unix'
  pbc='false'>
  <address> driver </address>
</ffsocket>
<ffsocket name='maceoff23-pigs' mode='unix'
  pbc='false'>
  <address> driver-pigs </address>
</ffsocket>
```

The addition of the potentials, along with the calculation of all derivatives like the force and stress, is done under the hood by defining the forces of the system class comprising two force components that communicate with the forcefield sockets:

```
[... ]
<forces>
  <force forcefield='maceoff23'> </force>
  <force forcefield='maceoff23-pigs'>
  </force>
</forces>
[... ]
```

The remaining steps for estimating IR and Raman spectra, as shown in Fig. 15, include predicting the total polarization and polarizability tensors and estimating their time correlation functions as detailed in Ref. 121. The scripts for these steps are in the *demos* directory.

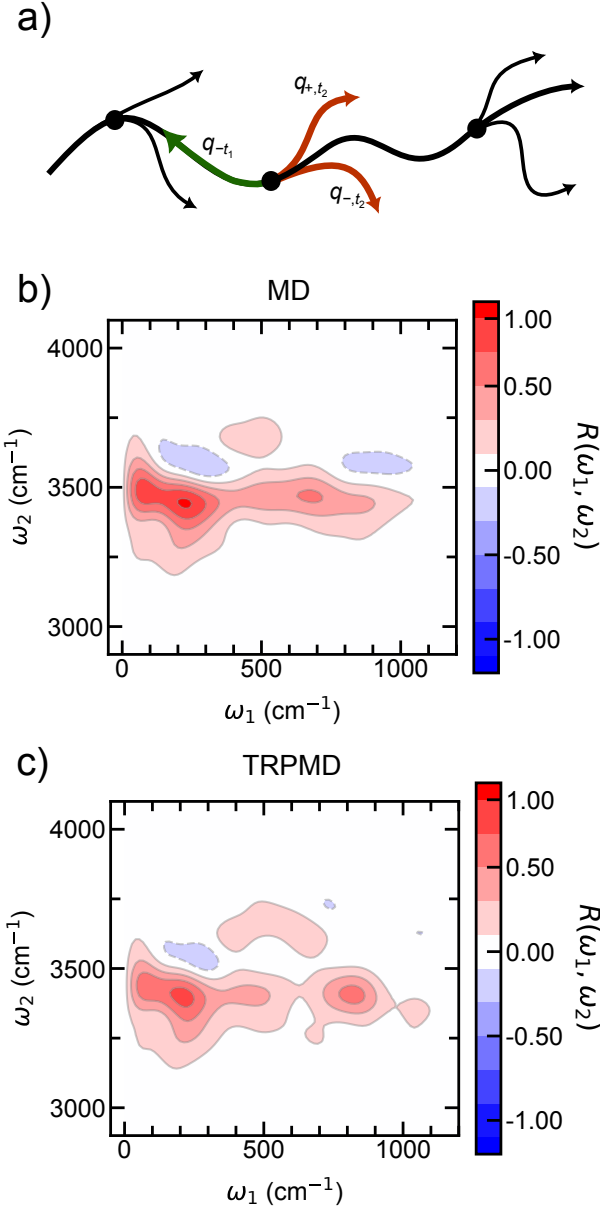


FIG. 16. Two-dimensional IR-Raman spectroscopy: Panel (a) shows the schematic of the equilibrium-nonequilibrium (RP)MD approach (see text). Panel (b) and (c) depict the frequency-dependent response function corresponding to the two-dimensional THz-IR-Raman (also called THz-IR-visible [194]) spectrum of liquid water, computed using classical MD and TRPMD, respectively. Figure adapted from Ref. 119, with permission (CC-BY Licence).

B. 2D IR-Raman spectra of liquid water

Nonlinear, multi-time response functions are associated with different flavours of time-resolved and multidimensional spectroscopic techniques.[195] Here, we show an example provided in the i-PI *demos* directory, that simulates the two-dimensional IR-Raman spectrum of

liquid water in which the sample interacts first with two mid- or far-IR pulses through a dipole interaction and then with a near-IR or visible pulse through a Raman process.[194, 196] The spectrum is obtained as a two-dimensional Fourier sine transform of the two-time response function

$$R(t_1, t_2) = -\frac{1}{\hbar^2} \langle [\hat{\alpha}(t_1 + t_2), \hat{\mu}(t_1)], \hat{\mu}(0) \rangle \quad (14)$$

where $\hat{\alpha}$ is the polarizability operator and $\hat{\mu}$ the dipole moment. The example presented in `demos/2D-IR-Raman` implements the equilibrium-nonequilibrium RPMD method of Refs. 118 and 119, which approximates the quantum-mechanical response function of Eq. 14. As a special case when the number of ring-polymer beads $P = 1$, the method corresponds to the MD approach of Hasegawa and Tanimura.[117] In passing, we note that other methods based on imaginary-time path integrals have been reported to calculate non-linear response functions using approximations to fully-symmetrized multi-time Kubo correlation functions.[197–199]

The equilibrium-nonequilibrium method requires sampling the initial distribution as in a typical PIMD simulation and running three trajectories for each initial phase-space point (q_0, p_0) —one backward equilibrium trajectory along which we evaluate the dipole moment and two nonequilibrium trajectories that are initialized with a momentum kick, i.e., starting from $(q_{\pm,0} = q_0, p_{\pm,0} = p_0 \pm \varepsilon \mu'(q_0)/2)$ (see schematics in Fig. 16a). The two-time response function is then computed as

$$R^{(2)}(t_1, t_2) = \frac{\beta}{\varepsilon} \langle [\alpha(q_{+,t_2}) - \alpha(q_{-,t_2})] \dot{\mu}(q_{-t_1}) \rangle, \quad (15)$$

where $\langle \cdot \rangle$ now represents a classical average over the ring-polymer equilibrium distribution at inverse temperature β . ε is a parameter that controls the magnitude of the perturbation (“kick”) applied to the nonequilibrium trajectories. Position-dependent operators $\alpha(q)$ and $\mu(q)$ in Eq. 15 are evaluated as averages over P beads of the ring polymer, like in the case of linear RPMD correlation functions.[90]

The evaluation of nonequilibrium trajectories, two-time response function, and two-dimensional spectrum is implemented in the scripts provided in `demos/2D-IR-Raman`. The `noneqm-traj.py` script uses checkpoint files and z -dipole nuclear gradients stored along a longer RPMD trajectory to run multiple shorter nonequilibrium trajectories (see Fig. 16a). In this way, the long RPMD trajectory serves both for sampling initial conditions and computing the response function. To properly sample the canonical ensemble, the whole calculation is typically repeated with independent RPMD trajectories. The `noneqm-response.py` script then computes the response function by reading in the dipole moments and polarizabilities along the long equilibrium trajectory and nonequilibrium trajectories. Finally, two-dimensional IR-Raman spectroscopy strongly depends on

the nonlinearity of the polarizabilities and dipole moments (electrical anharmonicity). To study these and other phenomena that are sensitive to such nonlinearities, we implemented a simple truncated dipole-induced-dipole model for water in i-PI’s internal Fortran driver (option `-m water_dip_pol`). Classical and TRPMD two-dimensional IR-Raman spectra of liquid water at 300 K are displayed in Fig. 16b showing that NQEs play a minor role in this observable. Overall, this example illustrates how minor scripting and modifications outside of the core i-PI code can be used to implement new methods and study various types of dynamical properties.

C. Melting of Charge Density Waves in 2D Materials

Downfolded lattice models aim to reduce the complexity of the full electronic problem by treating only a few low-energy electronic states (around the Fermi energy) explicitly and defining a simplified Hamiltonian where all other high-energy states are integrated out.[200] The parameters required for this dimensionality reduction can be obtained directly from *ab initio* calculations. There are several ways to include the effect of screening of the high-energy electrons on the interaction terms related to the explicit degrees of freedom of the downfolded Hamiltonian,[201] and making appropriate choices in this respect is necessary to reproduce the fully *ab initio* potential energy surface of certain materials.[20]

i-PI has recently been employed in this context, through its new interface with the `elphmod` code.[62] With `elphmod`, it is possible to set up downfolded models for the electronic interactions and for the electron-nuclear interactions, the latter being far less common to find in literature. Having access to the nuclear degrees of freedom in the Hamiltonian makes it possible to calculate nuclear forces efficiently, in the order of milliseconds per MD step per atom.[20]

In connection with i-PI, this capability allows the simulation of different types of molecular dynamics with such downfolded models. While MLIPs could play a similar role, there is an added value of physical interpretability when employing these models. Moreover, even if downfolded models cannot be applied to all types of materials, for suitable systems they are straightforward to parametrize.

In Ref. 20, large-scale replica-exchange and path-integral replica exchange simulations were performed with the i-PI/`elphmod` combination in order to study the charge-density-wave (CDW) phase transition in monolayer 1H-TaS₂. The downfolded model was obtained from PBE density-functional theory calculations. In particular, the phase transition between the 3×3 CDW and the symmetric structure was studied. We note that capturing the nuclear dynamics involved in such phase transitions calls for techniques that include non-perturbative anharmonic effects such as (PI)MD. In addition, as ex-

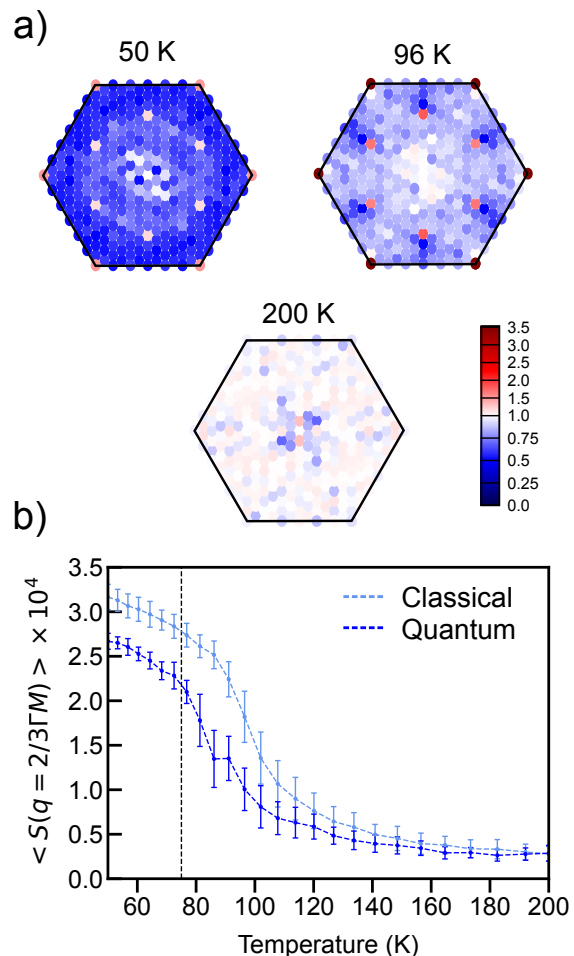


FIG. 17. (a) Ratio of the structure factors obtained from classical (S_{CL}) and quantum path-integral (S_{PI}) simulations S_{CL}/S_{PI} performed on a 18×18 supercell of monolayer 1H-TaS₂. A value close to 1 (indicated in white) corresponds to minimal differences between classical and quantum simulations. The peaks at $\mathbf{q} = 2/3\Gamma M$ and $\mathbf{q} = K$ for $T = 50$ K are characteristic of the 3×3 CDW phase. (b) Temperature-dependent structure factor $\langle S(\mathbf{q}) \rangle_T$ at the characteristic CDW wavevector $\mathbf{q} = 2/3\Gamma M$ obtained from classical MD replica-exchange MD (light blue) and path integral replica-exchange MD (dark blue). The experimental value (for the bulk) is represented by the vertical black dashed line. The shift of the inflection point of the PIMD curve toward the experimental value is due to nuclear quantum effects. Figure and caption adapted from Ref. 20 (CC-BY Licence).

PLICITLY shown in Ref. 20, finite-size effects on the phase transition are significant for small unit cells of these systems, calling for potentials that are efficient and accurate enough to allow the simulation of larger unit cells containing thousands of atoms.

Experimentally, the “melting” temperature of the 3×3 CDW in bulk TaS₂ is measured at around 75 K.[202] While the phase-transition temperature is not known for the monolayer, it is accepted that it should not strongly

deviate from the bulk temperature. In Fig. 17, we show the ratio between classical and quantum \mathbf{q} -resolved structure factors in panel (a) and the intensity of the CDW 3×3 diffraction peaks with varying temperature in panel (b). The inflection point of these curves is shown to correlate with a peak in heat capacity calculated from the simulations. Nuclear quantum effects decrease the transition temperature by around 20 K, bringing it in closer agreement with the experimental estimate.

Such simulations open the path for a closer analysis of particular phonon modes that contribute to the CDW phase transition, an analysis of the co-existence of different CDW phases at metastable regions, and the mapping of more complex CDW phase diagrams. Given the close connection between CDW phases and superconductivity, this is a promising field of study. An example of how to set up and run the `elphmod` code in connection with `i-PI` is included in the `examples/clients` directory.

VI. CONCLUSIONS

In this paper, we described the current capabilities of the `i-PI` code. The fundamental reasoning behind the design decisions made during its early development, namely an architecture that allows rapid prototyping in Python, and utilities to facilitate implementing complicated sampling algorithms, still prove to be very useful. However, some of the underlying assumptions made in the past are no longer valid. For example, the widespread adoption of MLIPs has dramatically reduced the cost of energy evaluation relative to explicit electronic-structure calculations, making the overhead associated with the evolution of the atomic coordinates a more pressing concern. In this paper, we showed that relatively superficial changes in the implementation of some of the core classes in `i-PI` substantially accelerated the most severe bottlenecks, ensuring that `i-PI` can now be used in combination with state-of-the-art MLIPs without adding significant overhead, even when pushing the energy evaluation to high levels of parallelism.

At the same time, the modular design of `i-PI` makes it easy to implement radically new modeling ideas, bringing algorithmic speed-ups and new useful features to the codebase. A bosonic PIMD implementation with quadratic scaling, the implementation of the ring polymer instanton with explicit friction method, a framework for dynamics driven by an electric field, a framework for simulations of light-matter coupling, and a better interface with ensembles of MLIPs including uncertainty estimation, are the five examples we highlight. Many useful, and sophisticated, simulation workflows can also be realized without major implementation efforts, by simply combining existing methods with external scripts and suitable post-processing. We demonstrate this with three applications: path integral coarse-grained simulations for vibrational spectra, calculations of the 2D-IR spectrum of water, and the melting of 2D charge-density waves.

As MLIPs become faster and enable simulations with larger lengths and time scales, it might be necessary to revisit some of the deeper architectural decisions of `i-PI` to further reduce the overhead inherent to the serial Python-based implementation. The work that was made in preparation for this release demonstrates that these kinds of optimizations can coexist with a modular design, that supports the fast implementation of new and exciting methods for advanced molecular simulations.

ACKNOWLEDGMENTS

Y.L. has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project number 467724959. V.K. acknowledges support from the Ernest Oppenheimer Early Career Fellowship and the Sydney Harvey Junior Research Fellowship, Churchill College, University of Cambridge. V.K. is grateful for computational support from the Swiss National Supercomputing Centre under project s1209, the UK national high-performance computing service, ARCHER2, for which access was obtained via the UKCP consortium and the EPSRC grant ref EP/P022561/1, and the Cambridge Service for Data Driven Discovery (CSD3). T.B. acknowledges financial support from the Swiss National Science Foundation through the Early Postdoc Mobility Fellowship (grant number P2ELP2-199757). B.H. acknowledges support by the USA-Israel Binational Science Foundation (grant No. 2020083) and the Israel Science Foundation (grants No. 1037/22 and 1312/22). Y.F. was supported by Schmidt Science Fellows, in partnership with the Rhodes Trust. T.E.L. is supported by start-up funds from the University of Delaware Department of Physics and Astronomy. M.R. and E.S. acknowledge computer time from the Max Planck Computing and Data Facility (MPCDF), funding from the IMPRS-UFASST program and the Lise-Meitner Excellence program. M.R. thanks Jan Berges for a careful read of section V C. M.C., M.K. and D.T. acknowledge funding from the Swiss National Science Foundation (SNSF) under the projects CRSII5.202296 and 200020.214879. M.C. and D.T. also acknowledge the support from the MARVEL National Centre of Competence in Research (NCCR). M.C. acknowledges funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme Grant No. 101001890-FIAMMA. We thank Eli Fields for helpful comments.

VII. DATA AVAILABILITY

`i-PI` can be obtained downloaded from <https://github.com/i-pi/i-pi>. The tag `v3.0.0-beta` was used to generate the examples and benchmarks reported in this manuscript. The data required to reproduce the figures and benchmark simulations are available at

- [1] O. A. von Lilienfeld and K. Burke, Retrospective on a decade of machine learning for chemical discovery, *Nature Communications* **11**, 4895 (2020).
- [2] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Machine learning for molecular and materials science, *Nature* **559**, 547 (2018).
- [3] H. J. Kulik, T. Hammerschmidt, J. Schmidt, S. Botti, M. A. L. Marques, M. Boley, M. Scheffler, M. Todorović, P. Rinke, C. Oses, A. Smolyanyuk, S. Curtarolo, A. Tkatchenko, A. P. Bartók, S. Manzhos, M. Ihara, T. Carrington, J. Behler, O. Isayev, M. Veit, A. Grisafi, J. Nigam, M. Ceriotti, K. T. Schütt, J. Westermayr, M. Gastegger, R. J. Maurer, B. Kalita, K. Burke, R. Nagai, R. Akashi, O. Sugino, J. Hermann, F. Noé, S. Pilati, C. Draxl, M. Kuban, S. Rigamonti, M. Scheidgen, M. Esters, D. Hicks, C. Toher, P. V. Balachandran, I. Tamblyn, S. Whitelam, C. Bellinger, and L. M. Ghiringhelli, Roadmap on machine learning in electronic structure, *Electronic Structure* **4**, 023004 (2022).
- [4] V. L. Deringer, M. A. Caro, and G. Csányi, Machine learning interatomic potentials as emerging tools for materials science, *Advanced Materials* **31**, 1902765 (2019).
- [5] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, Machine learning unifies the modeling of materials and molecules, *Science Advances* **3**, e1701816 (2017).
- [6] Q. Tao, P. Xu, M. Li, and W. Lu, Machine learning for perovskite materials design and discovery, *npj Computational Materials* **7**, 23 (2021).
- [7] D. Lu, H. Wang, M. Chen, L. Lin, R. Car, W. E, W. Jia, and L. Zhang, 86 pflops deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy, *Computer Physics Communications* **259**, 107624 (2021).
- [8] J. Behler, First principles neural network potentials for reactive simulations of large molecular and condensed systems, *Angewandte Chemie International Edition* **56**, 12828 (2017).
- [9] I. Batatia, P. Benner, Y. Chiang, A. M. Elena, D. P. Kovács, J. Riebesell, X. R. Advincula, M. Asta, M. Avaylon, W. J. Baldwin, F. Berger, N. Bernstein, A. Bhowmik, S. M. Blau, V. Carare, J. P. Darby, S. De, F. D. Pia, V. L. Deringer, R. Elijošius, Z. El-Machachi, F. Falcioni, E. Fako, A. C. Ferrari, A. Genreith-Schriever, J. George, R. E. A. Goodall, C. P. Grey, P. Grigorev, S. Han, W. Handley, H. H. Heenen, K. Hermansson, C. Holm, J. Jaafar, S. Hofmann, K. S. Jakob, H. Jung, V. Kapil, A. D. Kaplan, N. Karimitari, J. R. Kermode, N. Kroupa, J. Kullgren, M. C. Kuner, D. Kuryla, G. Liepuoniute, J. T. Margraf, I.-B. Magdău, A. Michaelides, J. H. Moore, A. A. Naik, S. P. Niblett, S. W. Norwood, N. O'Neill, C. Ortner, K. A. Persson, K. Reuter, A. S. Rosen, L. L. Schaaf, C. Schran, B. X. Shi, E. Sivonxay, T. K. Stenczel, V. Svahn, C. Sutton, T. D. Swinburne, J. Tilly, C. van der Oord, E. Vargumbrich, T. Vegge, M. Vondrák, Y. Wang, W. C. Witt, F. Zills, and G. Csányi, A foundation model for atomistic materials chemistry (2024), [arXiv:2401.00096](https://arxiv.org/abs/2401.00096).
- [10] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, and K.-R. Müller, Machine learning force fields, *Chemical Reviews* **121**, 10142 (2021).
- [11] A. Bruix, J. T. Margraf, M. Andersen, and K. Reuter, First-principles-based multiscale modelling of heterogeneous catalysis, *Nature Catalysis* **2**, 659 (2019).
- [12] P. Schlexer-Lamoureux, K. T. Winther, J. A. Garrido-Torres, V. Streibel, M. Zhao, M. Bajdich, F. Abild-Pedersen, and T. Bligaard, Machine learning for computational heterogeneous catalysis, *ChemCatChem* **11**, 3581 (2019).
- [13] S. Ma and Z.-P. Liu, Machine learning for atomic simulation and activity prediction in heterogeneous catalysis: Current status and future, *ACS Catalysis* **10**, 13213 (2020).
- [14] B. Cheng, E. A. Engel, J. Behler, C. Dellago, and M. Ceriotti, Ab initio thermodynamics of liquid and solid water, *Proceedings of the National Academy of Sciences* **116**, 1110 (2019).
- [15] V. L. Deringer and G. Csányi, Machine learning based interatomic potential for amorphous carbon, *Physical Review B* **95**, 094203 (2017).
- [16] S. Wengert, G. Csányi, K. Reuter, and J. T. Margraf, A hybrid machine learning approach for structure stability prediction in molecular co-crystal screenings, *Journal of Chemical Theory and Computation* **18**, 4586 (2022).
- [17] V. Kapil and E. A. Engel, A complete description of thermodynamic stabilities of molecular crystals, *Proceedings of the National Academy of Sciences* **119**, 10.1073/pnas.2111769119 (2022).
- [18] M. Ceriotti, J. More, and D. E. Manolopoulos, i-pi: A python interface for ab initio path integral molecular dynamics simulations, *Computer Physics Communications* **185**, 1019 (2014).
- [19] V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bivenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. Van Speybroeck, and M. Ceriotti, i-pi 2.0: A universal force engine for advanced molecular simulations, *Computer Physics Communications* **236**, 214 (2019).
- [20] A. Schobert, J. Berges, E. G. C. P. van Loon, M. A. Sentef, S. Brener, M. Rossi, and T. O. Wehling, Ab initio electron-lattice downfolding: Potential energy landscapes, anharmonicity, and molecular dynamics in charge density wave materials, *SciPost Phys.* **16**, 046 (2024), [arXiv:2303.07261](https://arxiv.org/abs/2303.07261).
- [21] T. M. Linker, A. Krishnamoorthy, L. L. Daemen, A. J. Ramirez-Cuesta, K. Nomura, A. Nakano, Y. Q. Cheng, W. R. Hicks, A. I. Kolesnikov, and P. D. Vashishta, Neutron scattering and neural-network quantum molecular dynamics investigation of the vibrations of ammonia along the solid-to-liquid transition, *Nature Communi-*

- cations **15**, [10.1038/s41467-024-48246-9](https://doi.org/10.1038/s41467-024-48246-9) (2024).
- [22] M. Jacobs, K. Fidanyan, M. Rossi, and C. Cocchi, Impact of nuclear effects on the ultrafast dynamics of an organic/inorganic mixed-dimensional interface, *Electronic Structure* (2024).
- [23] Y. Liu, R. Long, W.-H. Fang, and O. V. Prezhdo, Nuclear quantum effects prolong charge carrier lifetimes in hybrid organic–inorganic perovskites, *Journal of the American Chemical Society* **145**, 14112–14123 (2023).
- [24] L. Gigli, D. Tisi, F. Grasselli, and M. Ceriotti, Mechanism of charge transport in lithium thiophosphate, *Chemistry of Materials* **36**, 1482 (2024).
- [25] J. Lan, V. Kapil, P. Gasparotto, M. Ceriotti, M. Iannuzzi, and V. V. Rybkin, Simulating the ghost: quantum dynamics of the solvated electron, *Nature Communications* **12**, 766 (2021).
- [26] J. Lan, V. V. Rybkin, and A. Pasquarello, Temperature dependent properties of the aqueous electron, *Angewandte Chemie International Edition* **61**, e202209398 (2022).
- [27] F. Novelli, K. Chen, A. Buchmann, T. Ockelmann, C. Hoberg, T. Head-Gordon, and M. Havenith, The birth and evolution of solvated electrons in the water, *Proceedings of the National Academy of Sciences* **120**, [10.1073/pnas.2216480120](https://doi.org/10.1073/pnas.2216480120) (2023).
- [28] B. Cheng, G. Mazzola, C. J. Pickard, and M. Ceriotti, Evidence for supercritical behaviour of high-pressure liquid hydrogen, *Nature* **585**, 217 (2020).
- [29] V. Kapil, C. Schran, A. Zen, J. Chen, C. J. Pickard, and A. Michaelides, The first-principles phase diagram of monolayer nanoconfined water, *Nature* **609**, 512 (2022).
- [30] S. L. Bore and F. Paesani, Realistic phase diagram of water from “first principles” data-driven quantum simulations, *Nature Communications* **14**, [10.1038/s41467-023-38855-1](https://doi.org/10.1038/s41467-023-38855-1) (2023).
- [31] Y. Litman, K.-Y. Chiang, T. Seki, Y. Nagata, and M. Bonn, Surface stratification determines the interfacial water structure of simple electrolyte solutions, *Nature Chemistry* **16**, 644 (2024).
- [32] Y. Litman, J. Lan, Y. Nagata, and D. M. Wilkins, Fully first-principles surface spectroscopy with machine learning, *The Journal of Physical Chemistry Letters* **14**, 8175 (2023).
- [33] S. Shepherd, J. Lan, D. M. Wilkins, and V. Kapil, Efficient Quantum Vibrational Spectroscopy of Water with High-Order Path Integrals: From Bulk to Interfaces, *The Journal of Physical Chemistry Letters* , 9108 (2021).
- [34] K. Fidanyan, G. Liu, and M. Rossi, Ab initio study of water dissociation on a charged Pd(111) surface, *The Journal of Chemical Physics* **158**, 094707 (2023), [2212.08855](https://doi.org/10.1063/1.5022121).
- [35] M. de la Puente and D. Laage, How the acidity of water droplets and films is controlled by the air–water interface, *Journal of the American Chemical Society* **145**, 25186–25194 (2023).
- [36] K. Inoue, Y. Litman, D. M. Wilkins, Y. Nagata, and M. Okuno, Is unified understanding of vibrational coupling of water possible? hyper-raman measurement and machine learning spectra, *The Journal of Physical Chemistry Letters* **14**, 3063 (2023).
- [37] S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Sauceda, A. Tkatchenko, and K.-R. Müller, Accurate global machine learning force fields for molecules with hundreds of atoms, *Science Advances* **9**, [10.1126/sciadv.adf0873](https://doi.org/10.1126/sciadv.adf0873) (2023).
- [38] D. P. Kovács, J. H. Moore, N. J. Browning, I. Batafia, J. T. Horton, V. Kapil, W. C. Witt, I.-B. Magdău, D. J. Cole, and G. Csányi, *MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules* (2023), [arXiv:2312.15211](https://arxiv.org/abs/2312.15211).
- [39] M. Bonomi, D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello, PLUMED: A portable plugin for free-energy calculations with molecular dynamics, *Computer Physics Communications*. **180**, 1961 (2009).
- [40] K. Rossi, V. Jurásková, R. Wischert, L. Garel, C. Corminboeuf, and M. Ceriotti, Simulating Solvation and Acidity in Complex Mixtures with First-Principles Accuracy: The Case of CH₃SO₃H and H₂O₂ in Phenol, *Journal of Chemical Theory and Computation* **16**, 5139 (2020).
- [41] J. J. Mortensen, GPAW: An open Python package for electronic structure calculations, *The Journal of Chemical Physics* **160**, 092503 (2024).
- [42] Q. Sun, X. Zhang, S. Banerjee, P. Bao, M. Barbry, N. S. Blunt, N. A. Bogdanov, G. H. Booth, J. Chen, Z.-H. Cui, J. J. Eriksen, Y. Gao, S. Guo, J. Hermann, M. R. Hermes, K. Koh, P. Koval, S. Lehtola, Z. Li, J. Liu, N. Mardirossian, J. D. McClain, M. Motta, B. Mussard, H. Q. Pham, A. Pulkin, W. Purwanto, P. J. Robinson, E. Ronca, E. R. Sayfutyarova, M. Scheurer, H. F. Schurkus, J. E. T. Smith, C. Sun, S.-N. Sun, S. Upadhyay, L. K. Wagner, X. Wang, A. White, J. D. Whitfield, M. J. Williamson, S. Wouters, J. Yang, J. M. Yu, T. Zhu, T. C. Berkelbach, S. Sharma, A. Y. Sokolov, and G. K.-L. Chan, Recent developments in the PySCF program package, *The Journal of Chemical Physics* **153**, 024109 (2020).
- [43] T. E. Markland and D. E. Manolopoulos, An efficient ring polymer contraction scheme for imaginary time path integral simulations, *The Journal of Chemical Physics* **129**, 024105 (2008).
- [44] T. E. Markland and D. E. Manolopoulos, A refined ring polymer contraction scheme for systems with electrostatic interactions, *Chem. Phys. Lett.* **464**, 256 (2008).
- [45] D. Tisi, F. Grasselli, L. Gigli, and M. Ceriotti, Thermal conductivity of Li₃PS₄ solid electrolytes with ab initio accuracy, *Phys. Rev. Mater.* **8**, 065403 (2024).
- [46] V. Kapil, J. VandeVondele, and M. Ceriotti, Accurate molecular dynamics and nuclear quantum effects at low cost by multiple steps in real and imaginary time: Using density functional theory to accelerate wavefunction methods, *The Journal of Chemical Physics* **144**, 054111 (2016).
- [47] V. Kapil, J. Wieme, S. Vandenbrande, A. Lammaire, V. Van Speybroeck, and M. Ceriotti, Modeling the Structural and Thermal Properties of Loaded Metal–Organic Frameworks. An Interplay of Quantum and Anharmonic Fluctuations, *Journal of Chemical Theory and Computation* **15**, 3237 (2019).
- [48] Y. Litman, D. Donadio, M. Ceriotti, and M. Rossi, Decisive role of nuclear quantum effects on surface mediated water dissociation at finite temperature, *The Journal of Chemical Physics* **148**, 102320 (2018).
- [49] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dulak, J. Friis, M. N.

- Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, The atomic simulation environment—a python library for working with atoms, *Journal of Physics: Condensed Matter* **29**, 273002 (2017).
- [50] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Computer Physics Communications*. **271**, 108171 (2022).
- [51] *i-PI official documentation*: <https://ipi-code.org/i-pi/index.html>.
- [52] S. J. Clark, M. D. Segall, C. J. Pickard, P. J. Hasnip, M. I. J. Probert, K. Refson, and M. C. Payne, First principles methods using castep, *Zeitschrift für Kristallographie - Crystalline Materials* **220**, 567 (2005).
- [53] T. D. Kühne, M. Iannuzzi, M. Del Ben, V. V. Rybkin, P. Seewald, F. Stein, T. Laino, R. Z. Khalullin, O. Schütt, F. Schiffmann, D. Golze, J. Wilhelm, S. Chulkov, M. H. Bani-Hashemian, V. Weber, U. Borštnik, M. Taillefumier, A. S. Jakobovits, A. Lazaro, H. Pabst, T. Müller, R. Schade, M. Guidon, S. Andermatt, N. Holmberg, G. K. Schenter, A. Hehn, A. Bussy, F. Belleflamme, G. Tabacchi, A. Glöß, M. Lass, I. Bethune, C. J. Mundy, C. Plessl, M. Watkins, J. VandeVondele, M. Krack, and J. Hutter, CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations, *The Journal of Chemical Physics* **152**, 194103 (2020).
- [54] B. Hourahine, B. Aradi, V. Blum, F. Bonafé, A. Buccheri, C. Camacho, C. Cevallos, M. Y. Deshayé, T. Dumitrică, A. Dominguez, S. Ehlert, M. Elstner, T. van der Heide, J. Hermann, S. Irle, J. J. Kranz, C. Köhler, T. Kowalczyk, T. Kubař, I. S. Lee, V. Lutsker, R. J. Maurer, S. K. Min, I. Mitchell, C. Negre, T. A. Niehaus, A. M. N. Niklasson, A. J. Page, A. Pecchia, G. Penazzi, M. P. Persson, J. Řezáč, C. G. Sánchez, M. Sternberg, M. Stöhr, F. Stuckenberg, A. Tkatchenko, V. W.-z. Yu, and T. Frauenheim, DFTB+, a software package for efficient approximate density functional theory based atomistic simulations, *The Journal of Chemical Physics* **152**, 124101 (2020).
- [55] S. Chmiela, H. E. Sauceda, I. Poltavsky, K.-R. Müller, and A. Tkatchenko, sgdml: Constructing accurate and data efficient molecular force fields using machine learning, *Computer Physics Communications* **240**, 38 (2019).
- [56] V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, and M. Scheffler, Ab initio molecular simulations with numeric atom-centered orbitals, *Computer Physics Communications*. **180**, 2175 (2009).
- [57] F. Musil, M. Stricker, A. Goscinski, F. Giberti, M. Veit, T. Junge, G. Fraux, M. Ceriotti, R. Cersonsky, M. Willatt, and A. Grisafi, cosmo-epfl/librascal, zenodo. <https://doi.org/10.5281/zenodo.4526063> (2021).
- [58] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, Quantum espresso: a modular and open-source software project for quantum simulations of materials, *Journal of Physics: Condensed Matter* **21**, 395502 (2009).
- [59] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. B. Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. D. Corso, S. de Gironcoli, P. Delugas, R. A. D. Jr, A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Küçükbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. O. de-la Roza, L. Paulatto, S. Poncé, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, and S. Baroni, Advanced capabilities for materials modelling with quantum espresso, *Journal of Physics: Condensed Matter* **29**, 465901 (2017).
- [60] P. Giannozzi, O. Baseggio, P. Bonfà, D. Brunato, R. Car, I. Carnimeo, C. Cavazzoni, S. de Gironcoli, P. Delugas, F. Ferrari Ruffino, A. Ferretti, N. Marzari, I. Timrov, A. Urru, and S. Baroni, Quantum espresso toward the exascale, *The Journal of Chemical Physics* **152**, 154105 (2020).
- [61] G. Kresse and J. Furthmüller, Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set, *Physical Review B* **54**, 11169 (1996).
- [62] J. Berges, A. Schobert, E. G. C. P. van Loon, M. Rösner, and T. O. Wehling, elphmod: Python modules for electron-phonon models, Zenodo (2017), doi:10.5281/zenodo.5919991.
- [63] S. V. T. Verstraelen, L. Vanduyfhuys and S. M. J. Rogge, Yaff, yet another force field, <https://molmod.ugent.be/software/>.
- [64] The “Path Integral Methods in Atomistic Modelling” is hosted at <https://app.courseware.epfl.ch/learning/course/course-v1:EPFL+X+2022/home> and code tutorials at <https://github.com/i-pi/piqm2023-tutorial/>.
- [65] See <https://flake8.pycqa.org/> for more information about Flake8.
- [66] See <https://github.com/psf/black> for more information about Black.
- [67] G. Bussi, D. Donadio, and M. Parrinello, Canonical sampling through velocity rescaling, *The Journal of Chemical Physics* **126**, 14101 (2007).
- [68] M. Ceriotti, M. Parrinello, T. E. Markland, and D. E. Manolopoulos, Efficient stochastic thermostating of path integral molecular dynamics., *The Journal of Chemical Physics* **133**, 124104 (2010).
- [69] M. Ceriotti, G. Bussi, and M. Parrinello, Langevin Equation with Colored Noise for Constant-Temperature Molecular Dynamics Simulations, *Physical Review Letters* **102**, 020601 (2009).
- [70] M. Ceriotti, G. Bussi, and M. Parrinello, Colored-Noise Thermostats à la Carte, *Journal of Chemical Theory*

- and Computation **6**, 1170 (2010).
- [71] M. Ceriotti, G. Bussi, and M. Parrinello, Nuclear Quantum Effects in Solids Using a Colored-Noise Thermostat, *Physical Review Letters* **103**, 30603 (2009).
- [72] M. Ceriotti and M. Parrinello, The δ -thermostat: selective normal-modes excitation by colored-noise Langevin dynamics, *Procedia Computer Science* **1**, 1607 (2010).
- [73] R. Dettori, M. Ceriotti, J. Hunger, C. Melis, L. Colombo, and D. Donadio, Simulating Energy Relaxation in Pump-Probe Vibrational Spectroscopy of Hydrogen-Bonded Liquids, *Journal of Chemical Theory and Computation* **13**, 1284 (2017).
- [74] M. Ceriotti and D. E. Manolopoulos, Efficient First-Principles Calculation of the Quantum Kinetic Energy and Momentum Distribution of Nuclei, *Physical Review Letters* **109**, 100604 (2012).
- [75] M. Rossi, V. Kapil, and M. Ceriotti, Fine tuning classical and quantum molecular dynamics using a generalized Langevin equation, *The Journal of Chemical Physics* **148**, 102301 (2018).
- [76] M. Hijazi, D. M. D. Wilkins, and M. Ceriotti, Fast-forward Langevin dynamics with momentum flips, *The Journal of Chemical Physics* **148**, 184109 (2018).
- [77] F. R. Krajewski and M. Parrinello, Linear scaling electronic structure calculations and accurate statistical mechanics sampling with noisy forces, *Physical Review B* **73**, 041105 (2006).
- [78] T. D. Kühne, M. Krack, F. R. Mohamed, and M. Parrinello, Efficient and accurate car-parrinello-like approach to born-oppenheimer molecular dynamics, *Physical Review Letters* **98**, 066401 (2007).
- [79] G. Bussi, T. Zykova-Timan, and M. Parrinello, Isothermal-isobaric molecular dynamics using stochastic velocity rescaling, *The Journal of Chemical Physics* **130**, 074101 (2009).
- [80] P. Raiteri, J. D. Gale, and G. Bussi, Reactive force field simulation of proton diffusion in bazo3 using an empirical valence bond approach, *Journal of Physics: Condensed Matter* **23**, 334213 (2011).
- [81] G. J. Martyna, A. Hughes, and M. E. Tuckerman, Molecular dynamics algorithms for path integrals at constant pressure, *The Journal of Chemical Physics* **110**, 3275 (1999).
- [82] V. Kapil, J. Wieme, S. Vandenbrande, A. Lemaire, V. Van Speybroeck, and M. Ceriotti, Modeling the structural and thermal properties of loaded metal-organic frameworks. an interplay of quantum and anharmonic fluctuations, *Journal of Chemical Theory and Computation* **15**, 3237 (2019).
- [83] G. Henkelman, B. P. Uberuaga, and H. Jónsson, A climbing image nudged elastic band method for finding saddle points and minimum energy paths, *The Journal of Chemical Physics* **113**, 9901 (2000).
- [84] G. Henkelman and H. Jónsson, Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points, *The Journal of Chemical Physics* **113**, 9978 (2000).
- [85] J. Nichols, H. Taylor, P. Schmidt, and J. Simons, Walking on potential energy surfaces, *The Journal of Chemical Physics* **92**, 340 (1990).
- [86] M. Rossi, P. Gasparotto, and M. Ceriotti, Anharmonic and Quantum Fluctuations in Molecular Crystals: A First-Principles Study of the Stability of Paracetamol, *Physical Review Letters* **117**, 115702 (2016).
- [87] R. Petraglia, A. Nicolai, M. M. D. Wodrich, M. Ceriotti, and C. Corminboeuf, Beyond static structures: Putting forth REMD as a tool to solve problems in computational organic chemistry, *Journal of Computational Chemistry* **37**, 83 (2016).
- [88] J. Liu, R. S. Andino, C. M. Miller, X. Chen, D. M. Wilkins, M. Ceriotti, and D. E. Manolopoulos, A Surface-Specific Isotope Effect in Mixtures of Light and Heavy Water, *The Journal of Physical Chemistry C* **117**, 2944 (2013).
- [89] B. Cheng, J. Behler, and M. Ceriotti, Nuclear Quantum Effects in Water at the Triple Point: Using Theory as a Link Between Experiments, *The Journal of Physical Chemistry Letters* **7**, 2210 (2016).
- [90] I. R. Craig and D. E. Manolopoulos, Quantum statistics and classical mechanics: Real time correlation functions from ring polymer molecular dynamics, *The Journal of Chemical Physics* **121**, 3368 (2004).
- [91] S. Habershon, D. E. Manolopoulos, T. E. Markland, and T. F. Miller, Ring-polymer molecular dynamics: quantum effects in chemical dynamics from classical trajectories in an extended phase space., *Annual review of physical chemistry* **64**, 387 (2013).
- [92] J. Cao and G. A. Voth, A new perspective on quantum time correlation functions, *The Journal of Chemical Physics* **99**, 10070 (1993).
- [93] J. Cao and G. A. Voth, The formulation of quantum statistical mechanics based on the Feynman path centroid density. IV. Algorithms for centroid molecular dynamics, *The Journal of Chemical Physics* **101**, 6168 (1994).
- [94] M. Rossi, M. Ceriotti, and D. E. Manolopoulos, How to remove the spurious resonances from ring polymer molecular dynamics., *The Journal of Chemical Physics* **140**, 234116 (2014).
- [95] Y. Litman, J. O. Richardson, T. Kumagai, and M. Rossi, Elucidating the nuclear quantum dynamics of intramolecular double hydrogen transfer in porphycene, *Journal of the American Chemical Society* **141**, 2526 (2019).
- [96] V. Kapil, J. Behler, and M. Ceriotti, High order path integrals made easy, *The Journal of Chemical Physics* **145**, 234103 (2016).
- [97] I. Poltavsky and A. Tkatchenko, Modeling quantum nuclei with perturbed path integral molecular dynamics, *Chemical Science* **7**, 1368 (2016).
- [98] T. M. Yamamoto, Path-integral virial estimator based on the scaling of fluctuation coordinates: Application to quantum clusters with fourth-order propagators, *The Journal of Chemical Physics* **123**, 104101 (2005).
- [99] M. Ceriotti and T. E. Markland, Efficient methods and practical guidelines for simulating isotope effects., *The Journal of Chemical Physics* **138**, 014112 (2013).
- [100] L. Lin, J. A. Morrone, R. Car, and M. Parrinello, Displaced Path Integral Formulation for the Momentum Distribution of Quantum Particles, *Physical Review Letters* **105**, 110602 (2010).
- [101] M. Ceriotti, G. A. R. Brain, O. Riordan, and D. E. Manolopoulos, The inefficiency of re-weighted sampling and the curse of system size in high order path integration, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **468**, 2 (2011).
- [102] S. S. Jang and G. A. Voth, Applications of higher order composite factorization schemes in imaginary time path integral simulations, *The Journal of Chemical Physics*

- 115, 7832 (2001).
- [103] R. Korol, N. Bou-Rabee, and T. F. Miller, Cayley modification for strongly stable path-integral and ring-polymer molecular dynamics, *The Journal of Chemical Physics* **151**, 10.1063/1.5120282 (2019).
- [104] G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, Explicit reversible integrators for extended systems dynamics, *Molecular Physics* **87**, 1117–1157 (1996).
- [105] V. Kapil, A. Cuzzocrea, and M. Ceriotti, Anisotropy of the Proton Momentum Distribution in Water, *The Journal of Physical Chemistry B* **122**, 6048 (2018).
- [106] V. Kapil, E. Engel, M. Rossi, and M. Ceriotti, Assessment of approximate methods for anharmonic free energies, *Journal of Chemical Theory and Computation* **15**, 5845 (2019).
- [107] G. Imbalzano, Y. Zhuang, V. Kapil, K. Rossi, E. A. Engel, F. Grasselli, and M. Ceriotti, Uncertainty estimation for molecular dynamics and sampling, *The Journal of Chemical Physics* **154**, 074102 (2021).
- [108] B. Hirshberg, V. Rizzi, and M. Parrinello, Path integral molecular dynamics for bosons, *Proceedings of the National Academy of Sciences* **116**, 21445 (2019).
- [109] Y. M. Y. Feldman and B. Hirshberg, Quadratic scaling bosonic path integral molecular dynamics, *The Journal of Chemical Physics* **159**, 154107 (2023).
- [110] B. Hirshberg, M. Invernizzi, and M. Parrinello, Path integral molecular dynamics for fermions: Alleviating the sign problem with the Bogoliubov inequality, *The Journal of Chemical Physics* **152**, 171102 (2020).
- [111] T. E. Li, J. E. Subotnik, and A. Nitzan, Cavity Molecular Dynamics Simulations of Liquid Water under Vibrational Ultrastrong Coupling, *Proc. Natl. Acad. Sci.* **117**, 18324 (2020).
- [112] T. E. Li, A. Nitzan, S. Hammes-Schiffer, and J. E. Subotnik, Quantum Simulations of Vibrational Strong Coupling via Path Integrals, *The Journal of Physical Chemistry Letters* **13**, 3890 (2022).
- [113] Y. Litman, E. S. Pócs, C. L. Box, R. Martinazzo, R. J. Maurer, and M. Rossi, Dissipative tunneling rates through the incorporation of first-principles electronic friction in instanton rate theory. I. Theory, *The Journal of Chemical Physics* **156**, 194106 (2022).
- [114] Y. Litman, E. S. Pócs, C. L. Box, R. Martinazzo, R. J. Maurer, and M. Rossi, Dissipative tunneling rates through the incorporation of first-principles electronic friction in instanton rate theory. II. Benchmarks and applications, *The Journal of Chemical Physics* **156**, 194107 (2022).
- [115] M. Kellner and M. Ceriotti, [Uncertainty quantification by direct propagation of shallow ensembles](#) (2024).
- [116] V. Kapil, D. M. Wilkins, J. Lan, and M. Ceriotti, Inexpensive modeling of quantum dynamics using path integral generalized Langevin equation thermostats, *The Journal of Chemical Physics* **152**, 124104 (2020).
- [117] T. Hasegawa and Y. Tanimura, Calculating fifth-order Raman signals for various molecular liquids by equilibrium and nonequilibrium hybrid molecular dynamics simulation algorithms, *The Journal of Chemical Physics* **125**, 074512 (2006).
- [118] T. Begušić, X. Tao, G. A. Blake, and I. Miller, Thomas F., Equilibrium–nonequilibrium ring-polymer molecular dynamics for nonlinear spectroscopy, *The Journal of Chemical Physics* **156**, 131102 (2022).
- [119] T. Begušić and G. A. Blake, Two-dimensional infrared-raman spectroscopy as a probe of water’s tetrahedrality, *Nature Communications* **14**, 1950 (2023).
- [120] F. Musil, I. Zaporozhets, F. Noé, C. Clementi, and V. Kapil, Quantum dynamics using path integral coarse-graining, *The Journal of Chemical Physics* **10.1063/5.0120386** (2022).
- [121] V. Kapil, D. P. Kovács, G. Csányi, and A. Michaelides, First-principles spectroscopy of aqueous interfaces using machine-learned electronic and quantum nuclear effects, *Faraday Discuss.* **249**, 50 (2024).
- [122] G. Bussi, D. Donadio, and M. Parrinello, Canonical sampling through velocity rescaling, *The Journal of Chemical Physics* **126**, 014101 (2007).
- [123] M. Suzuki, Hybrid exponential product formulas for unbounded operators with possible applications to Monte Carlo simulations, *Physics Letters A* **201**, 425 (1995).
- [124] B. Leimkuhler and C. Matthews, Robust and efficient configurational molecular sampling via Langevin dynamics, *The Journal of Chemical Physics* **138** (2013).
- [125] J. Nagle, Congestion control in ip/tcp internetworks, *SIGCOMM Comput. Commun. Rev.* **14**, 11–17 (1984).
- [126] J. Han, L. Zhang, R. Car, and W. E, Deep potential: A general representation of a many-body potential energy surface, *Communications in Computational Physics* **23**, 629 (2018).
- [127] H. Wang, L. Zhang, J. Han, and W. E, Deepmd-kit: A deep learning package for many-body potential energy representation and molecular dynamics, *Computer Physics Communications* **228**, 178 (2018).
- [128] J. Zeng, D. Zhang, D. Lu, P. Mo, Z. Li, Y. Chen, M. Rynik, L. Huang, Z. Li, S. Shi, Y. Wang, H. Ye, P. Tuo, J. Yang, Y. Ding, Y. Li, D. Tisi, Q. Zeng, H. Bao, Y. Xia, J. Huang, K. Muraoka, Y. Wang, J. Chang, F. Yuan, S. L. Bore, C. Cai, Y. Lin, B. Wang, J. Xu, J.-X. Zhu, C. Luo, Y. Zhang, R. E. A. Goodall, W. Liang, A. K. Singh, S. Yao, J. Zhang, R. Wentzcovitch, J. Han, J. Liu, W. Jia, D. M. York, W. E, R. Car, L. Zhang, and H. Wang, DeePMD-kit v2: A software package for deep potential models, *The Journal of Chemical Physics* **159**, 054801 (2023).
- [129] P. Ravindra, X. R. Advincula, C. Schran, A. Michaelides, and V. Kapil, [A quasi-one-dimensional hydrogen-bonded monolayer ice phase](#) (2023), arXiv:2312.01340 [cond-mat].
- [130] D. M. Ceperley, Path integrals in the theory of condensed helium, *Reviews of Modern Physics* **67**, 279 (1995).
- [131] C. W. Myung, B. Hirshberg, and M. Parrinello, Prediction of a supersolid phase in high-pressure deuterium, *Physical Review Letters* **128**, 045301 (2022).
- [132] D. Chandler and P. G. Wolynes, Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids, *The Journal of Chemical Physics* **74**, 4078 (1981).
- [133] M. E. Tuckerman, B. J. Berne, G. J. Martyna, and M. L. Klein, Efficient molecular dynamics and hybrid Monte Carlo algorithms for path integrals, *The Journal of Chemical Physics* **99**, 2796 (1993), https://pubs.aip.org/aip/jcp/article-pdf/99/4/2796/19121187/2796_1_online.pdf.
- [134] T. Dornheim, M. Invernizzi, J. Vorberger, and B. Hirshberg, Attenuating the fermion sign problem in path integral Monte Carlo simulations using the Bogoliubov

- inequality and thermodynamic integration, *The Journal of Chemical Physics* **153**, 234104 (2020).
- [135] T. Dornheim, P. Tolias, S. Groth, Z. A. Moldabekov, J. Vorberger, and B. Hirshberg, Fermionic physics from ab initio path integral Monte Carlo simulations of fictitious identical particles, *The Journal of Chemical Physics* **159**, 164113 (2023).
- [136] A. M. Wodtke, Electronically non-adiabatic influences in surface chemistry and dynamics, *Chem. Soc. Rev.* **45**, 3641 (2016).
- [137] N. Kavokine, M.-L. Bocquet, and L. Bocquet, Fluctuation-induced quantum friction in nanoscale water flows, *Nature* **602**, 84 (2022).
- [138] M. Head-Gordon and J. C. Tully, Molecular dynamics with electronic frictions, *The Journal of Chemical Physics* **103**, 10137 (1995).
- [139] W. Dou, G. Miao, and J. E. Subotnik, Born-oppenheimer dynamics, electronic friction, and the inclusion of electron-electron interactions, *Physical Review Letters* **119**, 046001 (2017).
- [140] W. Dou and J. E. Subotnik, Perspective: How to understand electronic friction, *The Journal of Chemical Physics* **148**, 230901 (2018).
- [141] R. Martinazzo and I. Burghardt, Quantum theory of electronic friction, *Physical Review A* **105**, 052215 (2022).
- [142] R. Martinazzo and I. Burghardt, Quantum dynamics with electronic friction, *Physical Review Letters* **128**, 206002 (2022).
- [143] R. Martinazzo and I. Burghardt, Quantum hydrodynamics of coupled electron-nuclear systems (2023), [arXiv:2310.08766](https://arxiv.org/abs/2310.08766).
- [144] R. J. Maurer, M. Askerka, V. S. Batista, and J. C. Tully, Ab initio tensorial electronic friction for molecules on metal surfaces: Nonadiabatic vibrational relaxation, *Physical Review B* **94**, 115432 (2016).
- [145] J. O. Richardson, Ring-polymer instanton theory, *International Reviews in Physical Chemistry* **37**, 171 (2018).
- [146] Y. Litman and M. Rossi, Multidimensional hydrogen tunneling in supported molecular switches: The role of surface interactions, *Physical Review Letters* **125**, 216001 (2020).
- [147] C. L. Box, W. G. Stark, and R. J. Maurer, Ab initio calculation of electron-phonon linewidths and molecular dynamics with electronic friction at metal surfaces with numeric atom-centred orbitals, *Electronic Structure* **5**, 035005 (2023).
- [148] J. O. Richardson and S. C. Althorpe, Ring-polymer molecular dynamics rate-theory in the deep-tunneling regime: Connection with semiclassical instanton theory, *The Journal of Chemical Physics* **131**, 214106 (2009).
- [149] A. N. Beyer, J. O. Richardson, P. J. Knowles, J. Rommel, and S. C. Althorpe, Quantum tunneling rates of gas-phase reactions from on-the-fly instanton calculations, *The Journal of Physical Chemistry Letters* **7**, 4374 (2016).
- [150] Y. Litman, *Tunneling and Zero-Point Energy Effects in Multidimensional Hydrogen Transfer Reactions: From Gas Phase to Adsorption on Metal Surfaces*, Ph.D. thesis, Freie Universität Berlin (2020).
- [151] O. Bridge, P. Lazzaroni, R. Martinazzo, M. Rossi, S. C. Althorpe, and Y. Litman, Quantum rates in dissipative systems with spatially varying friction (2024), [arXiv:2405.00512](https://arxiv.org/abs/2405.00512).
- [152] P. Salén, M. Basini, S. Bonetti, J. Hebling, M. Krasilnikov, A. Y. Nikitin, G. Shamuilov, Z. Tibai, V. Zhaunerchyk, and V. Goryashko, Matter manipulation with extreme terahertz light: Progress in the enabling thz technology, *Physics Reports* **836-837**, 1 (2019).
- [153] A. de la Torre, D. M. Kennes, M. Claassen, S. Gerber, J. W. McIver, and M. A. Sentef, Colloquium: Nonthermal pathways to ultrafast control in quantum materials, *Reviews of Modern Physics* **93**, 041002 (2021).
- [154] A. S. Disa, T. F. Nova, and A. Cavalleri, Engineering crystal structures with light, *Nature Physics* **17**, 1087 (2021).
- [155] D. M. Juraschek, M. Fechner, and N. A. Spaldin, Ultrafast structure switching through nonlinear phononics, *Physical Review Letters* **118**, 054101 (2017).
- [156] A. Subedi, A. Cavalleri, and A. Georges, Theory of nonlinear phononics for coherent light control of solids, *Physical Review B* **89**, 220301 (2014).
- [157] M. Först, C. Manzoni, S. Kaiser, Y. Tomioka, Y. Tokura, R. Merlin, and A. Cavalleri, Nonlinear phononics as an ultrafast route to lattice control, *Nature Physics* **7**, 854 (2011).
- [158] N. A. Spaldin, A beginner's guide to the modern theory of polarization, *Journal of Solid State Chemistry* **195**, 2 (2012), polar Inorganic Materials: Design Strategies and Functional Properties.
- [159] S. Baroni, S. de Gironcoli, A. Dal Corso, and P. Giannozzi, Phonons and related crystal properties from density-functional perturbation theory, *Reviews of Modern Physics* **73**, 515 (2001).
- [160] H. Shang, N. Raimbault, P. Rinke, M. Scheffler, M. Rossi, and C. Carbogno, All-electron, real-space perturbation theory for homogeneous electric fields: theory, implementation, and application within dft, *New Journal of Physics* **20**, 073040 (2018).
- [161] H. Partridge and D. W. Schwenke, The determination of an accurate isotope dependent potential energy surface for water from extensive ab initio calculations and experimental data, *The Journal of Chemical Physics* **106**, 4618 (1997).
- [162] F. Musil, M. J. Willatt, M. A. Langovoy, and M. Ceriotti, Fast and Accurate Uncertainty Estimation in Chemical Machine Learning, *Journal of Chemical Theory and Computation* **15**, 906 (2019).
- [163] B. Lakshminarayanan, A. Pritzel, and C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in *Advances in Neural Information Processing Systems*, Vol. 30, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).
- [164] G. Kunapuli, *Ensemble Methods for Machine Learning* (Manning, 2023).
- [165] A. Ashukha, A. Lyzhov, D. Molchanov, and D. Vetrov, Pitfalls of in-domain uncertainty estimation and ensemble in deep learning (2021), [arXiv:2002.06470](https://arxiv.org/abs/2002.06470).
- [166] R. Rahaman and A. Thiery, Uncertainty quantification and deep ensembles, in *Advances in Neural Information Processing Systems*, Vol. 34, edited by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan (Curran Associates, Inc., 2021) pp. 20063–20075.
- [167] F. Bigi, S. Chong, M. Ceriotti, and F. Grasselli, A pre-

- diction rigidity formalism for low-cost uncertainties in trained neural networks (2024), [arXiv:2403.02251](https://arxiv.org/abs/2403.02251).
- [168] J. Behler and M. Parrinello, Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces, *Physical Review Letters* **98**, 146401 (2007).
- [169] A. P. Bartók, R. Kondor, and G. Csányi, On representing chemical environments, *Physical Review B* **87**, 184115 (2013).
- [170] U. R. Pedersen, F. Hummel, and C. Dellago, Computing the crystal growth rate by the interface pinning method., *The Journal of Chemical Physics* **142**, 44104 (2015).
- [171] G. M. Torrie and J. P. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling, *Journal of Computational Physics* **23**, 187 (1977).
- [172] R. F. Ribeiro, L. A. Martínez-Martínez, M. Du, J. Campos-Gonzalez-Angulo, and J. Yuen-Zhou, Polariton Chemistry: Controlling Molecular Dynamics with Optical Cavities, *Chemical Science* **9**, 6325 (2018).
- [173] T. E. Li, B. Cui, J. E. Subotnik, and A. Nitzan, Molecular Polaritonics: Chemical Dynamics Under Strong Light–Matter Coupling, *Annu. Rev. Phys. Chem.* **73**, 43 (2022).
- [174] J. Fregoni, F. J. Garcia-Vidal, and J. Feist, Theoretical Challenges in Polaritonic Chemistry, *ACS Photonics* **9**, 1096 (2022).
- [175] A. Mandal, M. A. Taylor, B. M. Weight, E. R. Koessler, X. Li, and P. Huo, Theoretical Advances in Polariton Chemistry and Molecular Cavity Quantum Electrodynamics, *Chemical Reviews* **123**, 9786 (2023).
- [176] M. Ruggenthaler, D. Sidler, and A. Rubio, Understanding Polaritonic Chemistry from Ab Initio Quantum Electrodynamics, *Chemical Reviews* **123**, 11191 (2023).
- [177] A. Shalabney, J. George, J. Hutchison, G. Pupillo, C. Genet, and T. W. Ebbesen, Coherent Coupling of Molecular Resonators with a Microcavity Mode, *Nature Communications* **6**, 5981 (2015).
- [178] J. P. Long and B. S. Simpkins, Coherent Coupling between a Molecular Vibration and Fabry–Perot Optical Cavity to Give Hybridized States in the Strong Coupling Limit, *ACS Photonics* **2**, 130 (2015).
- [179] A. Thomas, J. George, A. Shalabney, M. Dryzhakov, S. J. Varma, J. Moran, T. Chervy, X. Zhong, E. Devaux, C. Genet, J. A. Hutchison, and T. W. Ebbesen, Ground-State Chemical Reactivity under Vibrational Coupling to the Vacuum Electromagnetic Field, *Angew. Chemie Int. Ed.* **55**, 11462 (2016).
- [180] A. Thomas, L. Lethuillier-Karl, K. Nagarajan, R. M. A. Vergauwe, J. George, T. Chervy, A. Shalabney, E. Devaux, C. Genet, J. Moran, and T. W. Ebbesen, Tilting a Ground-State Reactivity Landscape by Vibrational Strong Coupling, *Science* **363**, 615 (2019).
- [181] B. Xiang, R. F. Ribeiro, M. Du, L. Chen, Z. Yang, J. Wang, J. Yuen-Zhou, and W. Xiong, Intermolecular Vibrational Energy Transfer Enabled by Microcavity Strong Light–Matter Coupling, *Science* **368**, 665 (2020).
- [182] J. Flick, M. Ruggenthaler, H. Appel, and A. Rubio, Atoms and Molecules in Cavities, from Weak to Strong Coupling in Quantum-Electrodynamics (QED) Chemistry, *Proc. Natl. Acad. Sci.* **114**, 3026 (2017).
- [183] T. E. Li, A. Nitzan, and J. E. Subotnik, Cavity Molecular Dynamics Simulations of Vibrational Polariton-Enhanced Molecular Nonlinear Absorption, *The Journal of Chemical Physics* **154**, 094124 (2021).
- [184] T. E. Li, A. Nitzan, and J. E. Subotnik, Energy-Efficient Pathway for Selectively Exciting Solute Molecules to High Vibrational States via Solvent Vibration-Polariton Pumping, *Nature Communications* **13**, 4203 (2022).
- [185] T. E. Li and S. Hammes-Schiffer, QM/MM Modeling of Vibrational Polariton Induced Energy Transfer and Chemical Dynamics, *Journal of American Chemical Society* **145**, 377 (2023).
- [186] S. C. Althorpe, Path-integral approximations to quantum dynamics, *The European Physical Journal B* **94**, 155 (2021).
- [187] M. J. Willatt, M. Ceriotti, and S. C. Althorpe, Approximating Matsubara dynamics using the planetary model: Tests on liquid water and ice, *The Journal of Chemical Physics* **148**, 102336 (2018).
- [188] G. Trenins, M. J. Willatt, and S. C. Althorpe, Path-integral dynamics of water using curvilinear centroids, *The Journal of Chemical Physics* **151**, 054109 (2019).
- [189] C. Haggard, V. G. Sadhasivam, G. Trenins, and S. C. Althorpe, Testing the quas centroid molecular dynamics method on gas-phase ammonia, *The Journal of Chemical Physics* **155**, 174120 (2021).
- [190] A. Witt, S. D. Ivanov, M. Shiga, H. Forbert, and D. Marx, On the applicability of centroid and ring polymer path integral molecular dynamics for vibrational spectroscopy., *The Journal of Chemical Physics* **130**, 194510 (2009).
- [191] A. Singraber, J. Behler, and C. Dellago, Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials, *Journal of Chemical Theory and Computation* **15**, 1827 (2019).
- [192] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nature Communications* **13**, 2453 (2022).
- [193] I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, and G. Csanyi, MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields, *Advances in Neural Information Processing Systems* **35**, 11423 (2022).
- [194] M. Grechko, T. Hasegawa, F. D’Angelo, H. Ito, D. Turchinovich, Y. Nagata, and M. Bonn, Coupling between intra- and intermolecular motions in liquid water revealed by two-dimensional terahertz-infrared-visible spectroscopy, *Nature Communications* **9**, 885 (2018).
- [195] S. Mukamel, *Principles of Nonlinear Optical Spectroscopy* (Oxford University Press, 1999).
- [196] P. Seliya, M. Bonn, and M. Grechko, On selection rules in two-dimensional terahertz–infrared–visible spectroscopy, *The Journal of Chemical Physics* **160**, 034201 (2024).
- [197] K. A. Jung, P. E. Videla, and V. S. Batista, Ring-polymer, centroid, and mean-field approximations to multi-time Matsubara dynamics, *The Journal of Chemical Physics* **153**, 124112 (2020).
- [198] K. A. Jung, P. E. Videla, and V. S. Batista, Inclusion of nuclear quantum effects for simulations of nonlinear spectroscopy, *The Journal of Chemical Physics* **148**, 244105 (2018).
- [199] Z. Tong, P. E. Videla, K. A. Jung, V. S. Batista, and X. Sun, Two-dimensional Raman spectroscopy of

- Lennard-Jones liquids via ring-polymer molecular dynamics, *The Journal of Chemical Physics* **153**, 034117 (2020).
- [200] F. Aryasetiawan and F. Nilsson, *Downfolding Methods in Many-Electron Theory* (AIP Publishing LLC, 2022) <https://pubs.aip.org/book-pdf/12252618/9780735422490.pdf>.
- [201] J. Berges, N. Girotto, T. Wehling, N. Marzari, and S. Ponc e, Phonon self-energy corrections: To screen, or not to screen, *Physical Review X* **13**, 041009 (2023).
- [202] X.-M. Zhao, K. Zhang, Z.-Y. Cao, Z.-W. Zhao, V. V. Struzhkin, A. F. Goncharov, H.-K. Wang, A. G. Gavriliuk, H.-K. Mao, and X.-J. Chen, Pressure tuning of the charge density wave and superconductivity in $2h - \text{TaS}_2$, *Physical Review B* **101**, 134506 (2020).