The Journal of Chemical Physics

RESEARCH ARTICLE | JULY 27 2023

Machine-learned acceleration for molecular dynamics in CASTEP

Special Collection: Software for Atomistic Machine Learning

Tamás K. Stenczel ; Zakariya El-Machachi ; Guoda Liepuoniute ; Joe D. Morrow ; Albert P. Bartók; Matt I. J. Probert ; Gábor Csányi ; Volker L. Deringer

J. Chem. Phys. 159, 044803 (2023)
https://doi.org/10.1063/5.0155621

AIP Publishing

Machine-learned acceleration for molecular dynamics in CASTEP

Cite as: J. Chem. Phys. 159, 044803 (2023); doi: 10.1063/5.0155621
Submitted: 21 April 2023 • Accepted: 3 July 2023 •
Published Online: 27 July 2023

Tamás K. Stenczel,[1] Zakariya El-Machachi,[2] Guoda Liepuoniute,[1] Joe D. Morrow,[2] Albert P. Bartók,[3,4] Matt I. J. Probert,[5] Gábor Csányi,[1,a)] and Volker L. Deringer[2,a)]

AFFILIATIONS

[1] Engineering Laboratory, University of Cambridge, Cambridge CB2 1PZ, United Kingdom
[2] Department of Chemistry, Inorganic Chemistry Laboratory, University of Oxford, Oxford OX1 3QR, United Kingdom
[3] Department of Physics, University of Warwick, Coventry CV4 7AL, United Kingdom
[4] Warwick Centre for Predictive Modelling, School of Engineering, University of Warwick, Coventry CV4 7AL, United Kingdom
[5] School of Physics, Engineering and Technology, University of York, York YO10 5DD, United Kingdom

Note: This paper is part of the JCP Special Topic on Software for Atomistic Machine Learning.

[a)] **Authors to whom correspondence should be addressed:** gc121@cam.ac.uk and volker.deringer@chem.ox.ac.uk

ABSTRACT

Machine learning (ML) methods are of rapidly growing interest for materials modeling, and yet, the use of ML interatomic potentials for new systems is often more demanding than that of established density-functional theory (DFT) packages. Here, we describe computational methodology to combine the CASTEP first-principles simulation software with the on-the-fly fitting and evaluation of ML interatomic potential models. Our approach is based on regular checking against DFT reference data, which provides a direct measure of the accuracy of the evolving ML model. We discuss the general framework and the specific solutions implemented, and we present an example application to high-temperature molecular-dynamics simulations of carbon nanostructures. The code is freely available for academic research.

I. INTRODUCTION

Molecular dynamics (MD) simulations have become a key research technique in physics, chemistry, and materials science. Quantum-mechanical methods, most commonly density-functional theory (DFT), make it possible to accurately predict forces on atoms, which in turn can be used to integrate Newton's equation of motion and, therefore, to drive high-fidelity simulations of molecular systems and condensed phases.[1] This type of simulation, sketched in Fig. 1(a), is often referred to as first-principles or "*ab initio*" MD (AIMD): because it is based on the laws of quantum mechanics, no prior knowledge of the system is required as long as a reasonable initial structural model is chosen. Such AIMD simulations can reach system sizes of several hundreds of atoms, but remain computationally highly expensive.

Machine learning (ML) has opened up an increasingly popular alternative route in this regard.[2–6] The key idea is to fit ML surrogate models of the potential-energy surface of a given system based on a relatively small number of DFT reference computations. Despite the growing popularity and widespread use, there is still a significant barrier from the users' perspective compared to how widely and routinely DFT is used day-to-day. One bottleneck has been the development of suitable reference databases, which initially had to be constructed and curated with substantial user input. Examples for ML potentials, fitted to extensive databases of that type, are Gaussian approximation potential (GAP) models for carbon[7] or silicon[8] or the ANI series of neural-network potentials for organic molecules.[9,10] Once trained, these general-purpose models enable predictions outside of the direct scope of their training: for example, resolving the intricate structure of amorphous phosphorus[11] without having included it explicitly in the reference database. We also note recent work in the field on creating widely applicable, "universal" pre-trained neural-network potentials, which are beginning to be tested for very different chemistries.[12–14] For all those

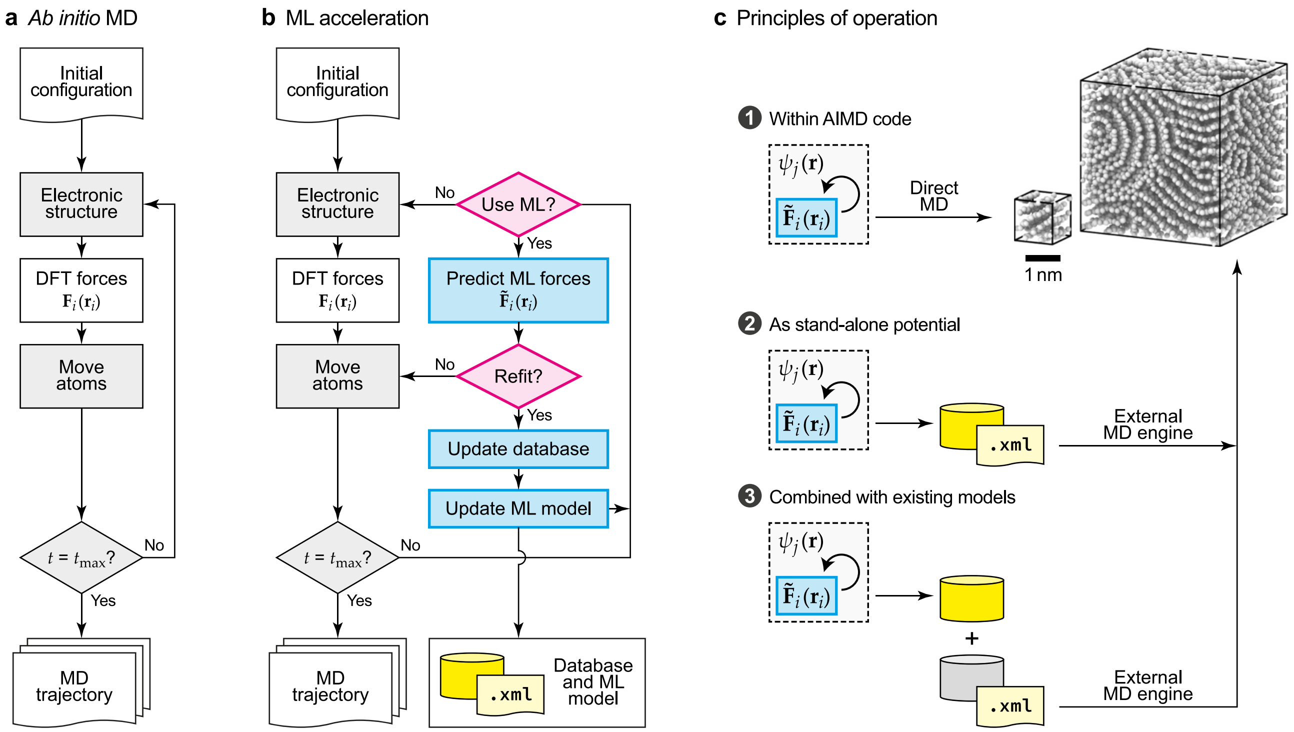


FIG. 1. Machine-learned acceleration for *ab initio* MD. (a) Simplified flowchart of the overall methodology in AIMD, as implemented in CASTEP and other DFT-based simulation software. Solving the electronic-structure problem yields DFT-based forces $\mathbf{F}_i$ for the *i*th atom. Those are used to move the atoms (by integrating Newton's equations of motion), and the process is repeated until the simulation time, *t*, reaches its user-defined maximum, $t_{\max}$. (b) Extension of this flowchart to ML-accelerated MD. In this case, the DFT forces are gradually replaced by an ML model prediction, $\tilde{\mathbf{F}}_i$, which is "learned" on the fly. The decisions whether to use DFT or ML forces, and whether to update ("refit") the ML model, can be based on different criteria. See also Ref. 18. (c) Principles of operation for ML-accelerated MD. Three possible use cases are sketched; relevant system sizes are illustrated using MD snapshots of 200-atom and 10 000-atom carbon systems (cf. Sec. V).

approaches, there is a substantial requirement for human and computational time in the first place, yet a significant possible benefit once the methodology is used in practice.

A somewhat complementary approach is now to use the underlying DFT code not just for labeling the data (i.e., computing reference energies and forces) but also to generate the structural data themselves "on the fly." In other words, the ML potential fit can be incorporated directly into the AIMD workflow [Fig. 1(b)]—starting an MD simulation with DFT forces, then accelerating it substantially with an on-the-fly fitted ML potential model where this is possible, and updating the model with new reference data as required. This way, there is no requirement for expert knowledge from the user, even for unseen systems. This type of approach was pioneered for the Vienna *Ab initio* Simulation Package (VASP),[15,16] which now offers ML potential model fitting on the fly.[17–20] We review emerging applications in more depth below.

In the present work, we describe a computational framework that allows for the direct interfacing of the DFT-based simulation software CASTEP (Ref. 21) with ML potential fitting. The key methodological steps described herein are (i) the integration of relevant functionality into the CASTEP code itself and (ii) a stand-alone decision-making code that connects to external software for the fitting of ML models. The code is designed to be interoperable with different ML potential fitting frameworks, whether already established in the field, newly emerging, or envisioned in the future: the practical aspects will change, but we expect the overarching ideas to remain the same. We illustrate these developments with a case study of a potential that is "learned" by our hybrid CASTEP + ML scheme and then taken out and used for a larger, stand-alone simulation.

II. METHODOLOGY

A. Machine-learned acceleration for *ab initio* MD

Figure 1 provides an overview of the methodology: the regular AIMD workflow is shown in simplified terms in Fig. 1(a), and the inclusion of the ML model requires additional actions (in blue) and decision points (magenta), as shown in Fig. 1(b). We note that this flowchart is deliberately generic, and the questions when to "Use ML?" and when to "Refit?" in Fig. 1(b) can be answered in multiple ways—one of which is discussed below (Sec. II C).

Irrespective of how exactly an on-the-fly fitted ML potential is created, there are different principles by which it can be used in computational practice [Fig. 1(c)]. It can be applied directly within the AIMD code to accelerate simulations of systems with typically a few hundred atoms. This direct acceleration is exemplified in

Fig. 1(c) by showing a 200-atom disordered carbon structure. However, once an on-the-fly model has been generated, it can also be used subsequently as a stand-alone potential[22–24]—evaluating it with an external MD engine, rather than within the AIMD code, thereby enabling much larger-scale simulations. Finally, one might expect in future work to take an on-the-fly generated fitting database (yellow) and combine it with other, existing datasets (gray), together yielding a combined ML potential for a more complex system. An example for the second principle is shown in Fig. 1(c) for a 10 000-atom carbon structure; we will discuss this specific example in Sec. V.

On-the-fly ML potential fitting methodology has been implemented in VASP[17,18] using kernel-based regression models similar to the ones used in GAP fitting and the associated Bayesian error estimation to define the decision criteria. The capability of the methodology was initially demonstrated for temperature-dependent structural distortions in a hybrid halide perovskite[17] and for melting-point predictions,[18] respectively. The utility of the approach was then shown for a wider range of research questions: for simulations of thermal properties that would be computationally highly expensive at the full DFT level[25,26] and for beyond-DFT computations for bulk and surface systems.[27,28] Once the ML potential has been fitted in this way, it can be taken out and used in larger-scale, stand-alone simulations [Fig. 1(c)], as has been demonstrated in Refs. 22–24. We mention in passing that there are analogies to other "on-the-fly" schemes, which similarly aim to accelerate MD simulations, although in different ways: by combining force-field and quantum-mechanical simulations[29,30] and by making external DFT calls during an ML-driven simulation (often referred to as "active learning").[31–35] In the following discussion, we take "on-the-fly" to refer to one of the workflows sketched in Figs. 1(b) and 1(c).

B. Gaussian approximation potentials

In the present work, we primarily fit ML potentials using the Gaussian approximation potential (GAP) framework and the QUIP and `gap_fit` codes. GAPs were initially introduced by Bartók *et al.* in Ref. 36, and many aspects of the methodology have been established in a series of papers on tungsten,[37] carbon,[38] and silicon.[8] The underlying theory of Gaussian process regression (GPR), as well as the design choices that are relevant in computational practice, has been reviewed in Ref. 39.

In brief, the aim of GPR models is to predict some target quantity, $\tilde{y}(\mathbf{x})$, at a location, $\mathbf{x}$. This prediction is made by comparing the new location to all N points in the training,

$$\tilde{y}(\mathbf{x}) = \sum_{i=1}^{N} c_i k(\mathbf{x}, \mathbf{x}_i) \equiv \mathbf{c}^{\mathsf{T}} \mathbf{k}(\mathbf{x}), \tag{1}$$

where c_i are the fitting coefficients and k is a kernel (or similarity) function.

The choice of k is, therefore, the first important decision to be made in GAP fitting. k is typically given by the Smooth Overlap of Atomic Position (SOAP) kernel.[40,41] The most relevant SOAP hyperparameters are the radial cutoff, r_{cut}, and the smoothness used for the neighbor density, σ_{atom}. The convergence of the neighbor-density expansion with respect to its local basis is controlled by setting maximum values for the radial and angular quantum numbers, n and l, respectively.

The training task in GPR fitting is in finding the coefficients, c_i, in Eq. (1). Given reference data labels, $\mathbf{y}$, the coefficients are obtained by minimizing a loss function, conveniently expressed in the notation as above,

$$\mathbf{c} = (\mathbf{K}_{NN} + \mathbf{\Sigma})^{-1}\mathbf{y}, \tag{2}$$

where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{\Sigma}$ adds the GPR regularization term(s).

In practice, there are two important points beyond the simple formula above. First, in the context of interatomic potentials, y is a local energy term, but this is not normally available from quantum mechanics directly—it is instead observed through the total energy and its derivatives, which changes the expression above (see Ref. 39). Second, only a limited number of representative points, M, is used in GAP fitting, leading to a sparse GPR model fitted according to

$$\mathbf{c} = [\mathbf{K}_{MM} + \mathbf{K}_{MN}\mathbf{\Sigma}^{-1}\mathbf{K}_{NM}]^{-1}\mathbf{K}_{MN}\mathbf{\Sigma}^{-1}\mathbf{y}, \tag{3}$$

where $M \ll N$ and the cost of *prediction* now depends on M rather than N, making the evaluation of a GAP independent of the size of its training database. Therefore, M becomes another important aspect to control the quality of the potential. It is usually chosen large enough to approach convergence with respect to M for general-purpose models;[8] for on-the-fly-fitted models, as generated here, the value will need to be smaller to speed up both fitting and evaluation. The choice of optimized parameters for "low-cost" on-the-fly ML potentials is under ongoing study in our laboratories.

C. Checking criteria

ML models accelerate AIMD by replacing the expensive *ab initio* force calculation with faster ML forces when possible. There is an inherent trade-off in accuracy between the two methods, requiring monitoring of the error of the ML model. This gives us two requirements for the design of the code: (1) minimizing the number of *ab initio* force calls throughout the simulation and (2) keeping the accuracy of the entire simulation within a specified bound. To fulfill these requirements, a decision-making procedure is introduced before force calculations [Fig. 1(b)], determining whether ML or *ab initio* forces should be used for the next step. We note that this decision can, in principle, be based on any structural or model-derived information, as long as that information is predictive of the error of the evolving ML model.

We have made the design choice to use direct checking against *ab initio* data to decide whether or not a potential needs to be refitted. This makes the approach applicable to all ML potential fitting frameworks, irrespective of whether they have internal confidence estimates available. At MD steps where checks are performed, both the *ab initio* and ML forces are evaluated, and the disagreement between the two is checked against user-supplied tolerance values. The new *ab initio* observation gathered can be used for updating the model either at all such observations or only when tolerances are not met. As there is a non-negligible cost associated with this refitting step, we only perform model training on occasions where the user-defined tolerances are not met. In principle, the model can be updated more often, though this is only economical if an existing model can be extended without needing to re-train from scratch—e.g., for neural-network-based models.

Figures 2(a) and 2(b) illustrate two possible interval modes for checking the accuracy of the model. The basis of both presented methods is a direct comparison between the CASTEP DFT results (which serve as the ground truth), on one hand, and the ML model prediction, on the other hand. The user can set tolerances for energy, force, and stress errors against which the evolving ML potential is assessed. The interface allows for running a number of initial DFT steps to gather data, indicated by blue shading, followed by a fit ("training") of the ML model to those data. This is useful for starting from scratch, i.e., without any previous ML potential model, as well as for updating a model to the structure/calculation at hand in cases where previously gathered *ab initio* data are supplied.

The checking then proceeds at specified points during the MD run. The simplest solution is pre-defining a fixed interval for the DFT re-evaluation and running a constant number of ML-driven steps in between. This is illustrated in Fig. 2(a): the only choice to make here is the number of MD time steps per interval, denoted as `n_fix` in the input file. We implemented an adaptive scheme as well [Fig. 2(b)], in which the number of MD steps between DFT force calculations is adapted according to the accuracy of the model observed: the previous interval is increased when tolerances are met and decreased otherwise, together with refitting the model. We use a scaling factor (keyword `factor` in the input file) by which to multiply or divide the interval, keeping it between a lower and upper bound of steps to take (keywords `n_min` and `n_max`, respectively).

Figure 2(c) illustrates the application of adaptive checking intervals for aluminum oxide, Al_2O_3. In this simple experiment, we take a 2 × 2 × 1 (120-atom) supercell expansion of the corundum-type unit cell of Al_2O_3 and remove one Al and one O atom each, as a prototype for more extended simulations of vacancy migration. We plot the energy (top) and force (bottom) errors observed during the first 1 ps of the MD run, indicating points where both DFT checks were passed in blue and those where at least one was not passed in red. Vertical lines indicate the refitting events. Detailed data, including the associated input files and the results of a longer simulation, are provided in the repository accompanying this paper (see the Data availability statement).

III. IMPLEMENTATION

In our implementation, we connect CASTEP[21] as the AIMD software to a custom and standalone Python program, which we call `hybrid-md`. Within the present study, we primarily use the GAP code for potential fitting and evaluation (Sec. II B). We emphasize, however, that `hybrid-md` can be interfaced with other ML potential fitting and evaluation frameworks as well, for example, through a custom Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) interface described below, which we expect to be relevant in the future. We refer to the overall framework as "CASTEP + ML" and to the specific potential models fitted in the present work as "CASTEP-GAP" models.

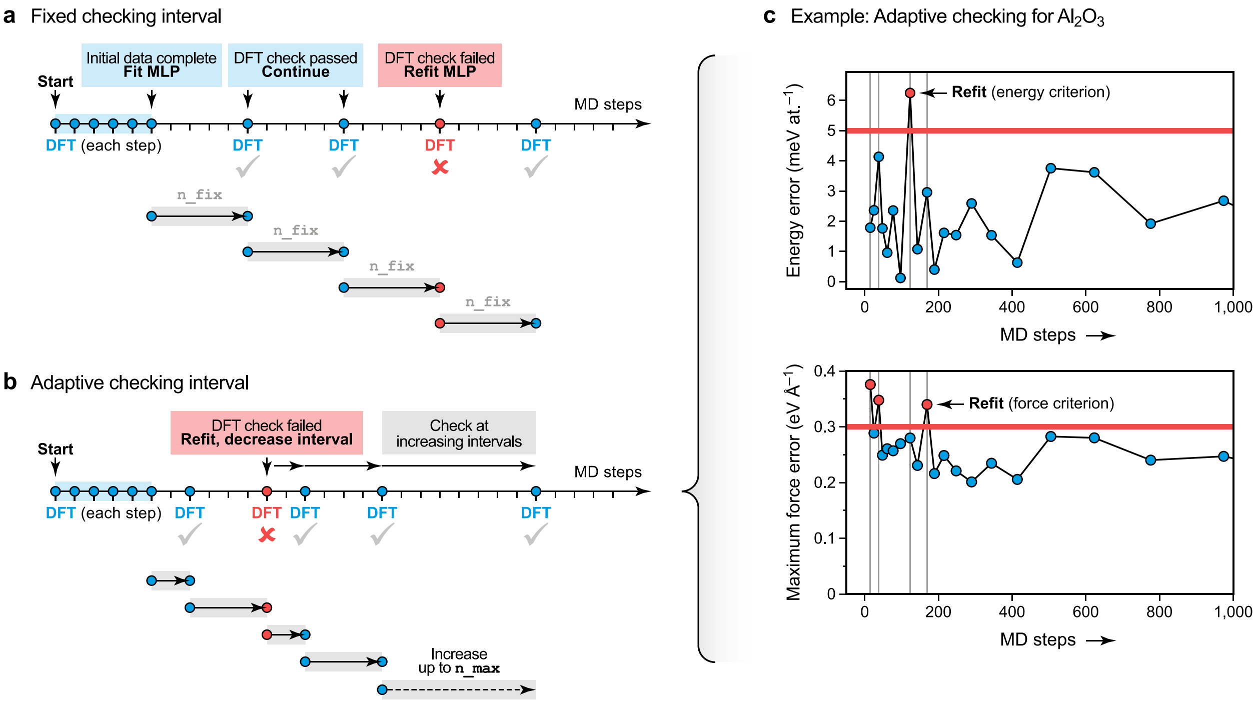


FIG. 2. Checking intervals. (a) and (b) Fixed and adaptive intervals for checking the quality of the simulation through a DFT call and refitting if necessary. (c) Energy and force error evolution during an MD simulation of Al_2O_3. This illustrates how during the simulation two criteria are tested: that of a maximum energy error of 5 meV per atom and that of a maximum force component error of 0.3 eV Å^{-1}. When one or both of these criteria are met, the potential is refitted, as indicated by vertical gray lines.

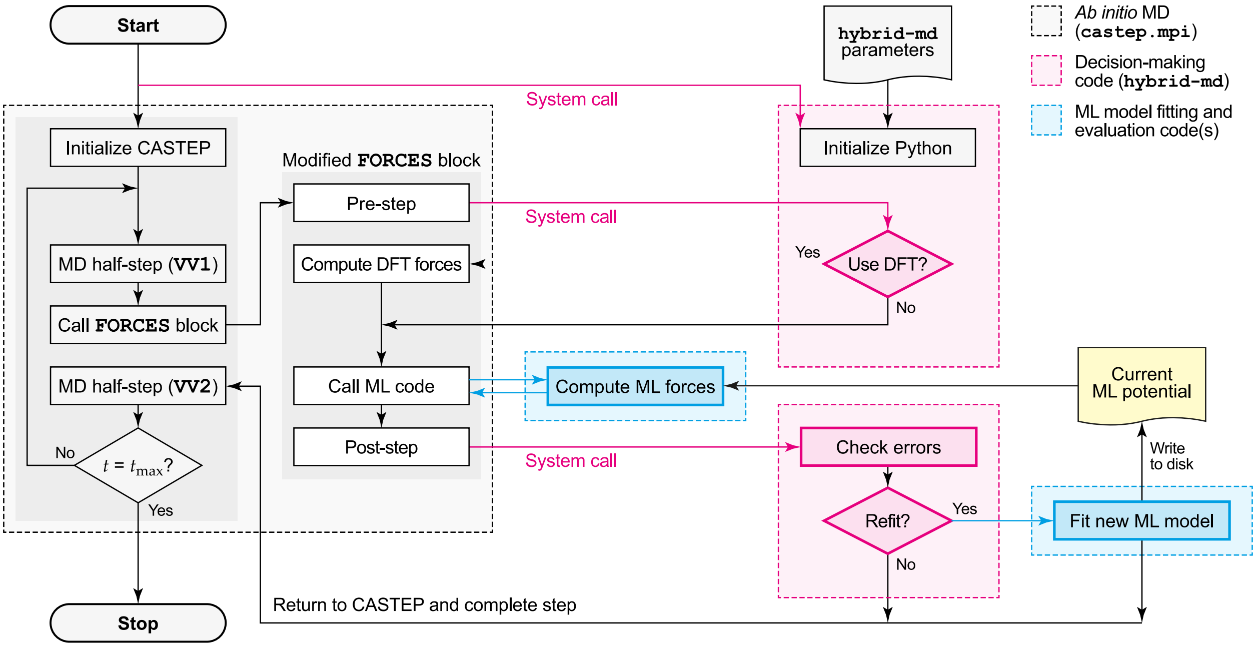


FIG. 3. Implementation of the decision-making code. The left-hand side shows a simplified flowchart of the overall methodology in CASTEP, which iterates over a number of Velocity Verlet (VV) steps and evaluates forces on atoms during the cycles. The dashed box, shaded in gray, encloses all steps that are being carried out within CASTEP (normally executed through `castep.mpi`). The close-up shows how the `FORCES` routine is being modified here to enable interfacing with the hybrid MD code. Functionality within the decision-making code, which is accessed separately through system calls, is highlighted in magenta; functionality of ML model fitting and evaluation codes is highlighted in blue.

Figure 3 provides an overview of the implementation, separated according to those tasks that are carried out within CASTEP (light gray), within the decision-making code described here (magenta), and by external ML potential fitting and evaluation software (blue).

A. GAP interface

We provide an interface in the CASTEP Fortran code to evaluate any QUIP-compatible interatomic potential model, including GAP. We designed a simple interface of the Python code as a shell executable that can be used by the CASTEP code via system calls. Decisions are passed as shell exit codes, parsed by the Fortran code. The behavior of the decision-making code is controlled by an input file; we show an example in Listing 1.

The model updates are provided by the Python code, building on the results of the DFT calculations being written to xyz files by CASTEP. The model fitting itself is carried out using the built-in fitting executable of GAP, which fits a new model from scratch each time there is an update in the data. Again, we note that the code is designed to enable extension and future interfaces with other fitting frameworks as well; we show initial examples to this effect below. In Fig. 3, we refer to generic tasks rather than specific ML potential software to emphasize the general nature of the approach.

LISTING 1. Example input file for the hybrid-md code.

```
can_update: true
num_initial_steps: 10

adaptive_method_parameters:
  n_min: 1
  n_max: 10000
  factor: 2

tolerances:
  ediff: 0.01    # eV/atom
  fmax: null     # eV/A
  frmse: null    # eV/A
  vmax: null     # eV/atom (virial)

refit:
  previous_data: [
      "init.xyz",
  ]
  default_sigma: "0.002 0.2 0.2 0.0"
  descriptor_str: <string>
  extra_gap_opts: "sparse_jitter = 1.0e-8"
  e0: "-148.314002"
```

The Python code is intended to be self-contained and provide the single interface for non-core development. This is favorable from a licensing point of view (the CASTEP license is more restrictive than GAP's ASL license) as well as for the efficiency of development: if the user wishes, they can change the Python code for the specific research question they are addressing or add a new custom feature, while not having to obtain, change, and recompile the source of the DFT code. The latter would be a clear issue on high-performance computing (HPC) systems, for example, on top of the typically higher barrier of entry into Fortran programming (compared to Python).

B. LAMMPS interface

We further extended CASTEP with an interface to call LAMMPS[42] as a force calculator, which provides access to a wider variety of ML models through optional packages in LAMMPS. For this work, we used the `ML-PACE` and `ML-MACE` packages in LAMMPS, making it possible to evaluate Atomic Cluster Expansion (ACE) models[43] and equivariant message-passing neural-network models based on ACE (MACE),[44] respectively.

LAMMPS uses domain decomposition for parallelism, splitting the simulation cell between Message Passing Interface (MPI) processes for neighbor list and force calculations. Our interface implements supplying the entire MPI communicator to LAMMPS or communicating only on a single process. The latter is essential for models using graphics processing unit (GPU) accelerators, such as MACE, where splitting the cell is counter-productive. In the future, extension for supporting further patterns is possible, e.g., one MPI process per GPU for a multi-GPU model evaluation.

C. Model training extensions

User-defined model training logic can be used for training ACE and MACE models. The import path of the required code can be specified in the input file of the acceleration program. ACE model training can be carried out using the `ACE1pack` Julia library, executing a training script similarly to how the `gap_fit` program is called for training GAP models. Training is performed from scratch (no re-use of previous model weights) on the central processing unit (CPU). The MACE Python package is used directly for training MACE models on a GPU. Incremental training is possible, where initially a longer optimization of model weights is used, and later, only short fine-tuning is carried out with new observations. We note that while this functionality is provided as part of the current code implementation, further work is required for an in-depth analysis of generally applicable and robust settings. All following discussion, therefore, will be based on the GAP framework, which is more firmly established at the time of this writing.

IV. NUMERICAL PERFORMANCE

We present example runs to assess the numerical performance of the method. We tested the speed of CASTEP MD runs, comparing standard AIMD with GAP-accelerated runs as described above. We chose two chemically rather different systems to provide representative examples of different envisioned use cases: defective crystalline Al_2O_3 (ordered structure, primarily ionic–covalent bonding) and liquid Si (highly disordered structure, primarily metallic bonding). Both simulations started from scratch with no ML model or data, and both used the adaptive checking interval scheme described in Sec. II C. The results are summarized in Table I.

TABLE I. Numerical performance of CASTEP + ML runs using the GAP framework. We show MD tests for Al_2O_3 and liquid Si model systems (see the text), specifying the fraction of simulation time spent in self-consistent field (SCF) evaluations (*ab initio* forces), model training (`gap_fit`), and the rest in ML forces and system overhead. The speedup is shown compared to the number of traditional AIMD steps one can perform within the same wall time on the same hardware.

	Crystalline Al_2O_3	Liquid Si
Time in SCF	35.3%	2.5%
Time in `gap_fit`	1.0%	3.0%
Time in GAP forces	63.7%	94.5%
(Incl. system & `hybrid-md`)		
CASTEP + ML MD steps	100 000	
AIMD steps in same time	≈200	≈5, 400
ML speedup	≈500×	≈20×

For the example run on Al_2O_3 characterized in Fig. 2(c), we examined the timing data to assess the numerical performance of the method. In this specific simulation, 35.3% of the overall runtime was required for the CASTEP self-consistent field loop (that is, for the DFT evaluation), 1.0% for running the external `gap_fit` program, and 63.7% for MD: system, decision making, and GAP model evaluation. A total of 57 DFT evaluations were carried out, and eight of those led to `gap_fit` calls [four of those during the first 1 ps of the simulation; see the gray horizontal lines in Fig. 2(c)]. Hence, over a total of 100 k MD steps, we had an average of 1750 GAP-MD steps per DFT computation. AIMD on the same system and input structure has been tested (same HPC system and settings): about 200 MD steps can be done within the same time, which means a speedup of about 500× with the ML acceleration.

For the equivalent test for liquid Si, a total of 36 DFT evaluations (2.5% of overall runtime) and 12 `gap_fit` calls (3.0%) were carried out over 100 k MD steps on a system size of 250 atoms (2563 per DFT computation, 94.5% of total time). ~5400 AIMD steps can be carried out within the same walltime using the same basis set and k-point sampling (250 eV, Γ point only), which amounts to a speedup of ≈20×. The liquid Si simulation was completed in two parts of duration 90 and 10 ps (the latter exemplifying the "restart" functionality). In the final 10 ps, one refit was performed. We note that for both test systems, only a small fraction of the computational time was required for the (re-) fitting of the ML models.

V. CASE STUDY: GRAPHITIZATION OF AMORPHOUS CARBON

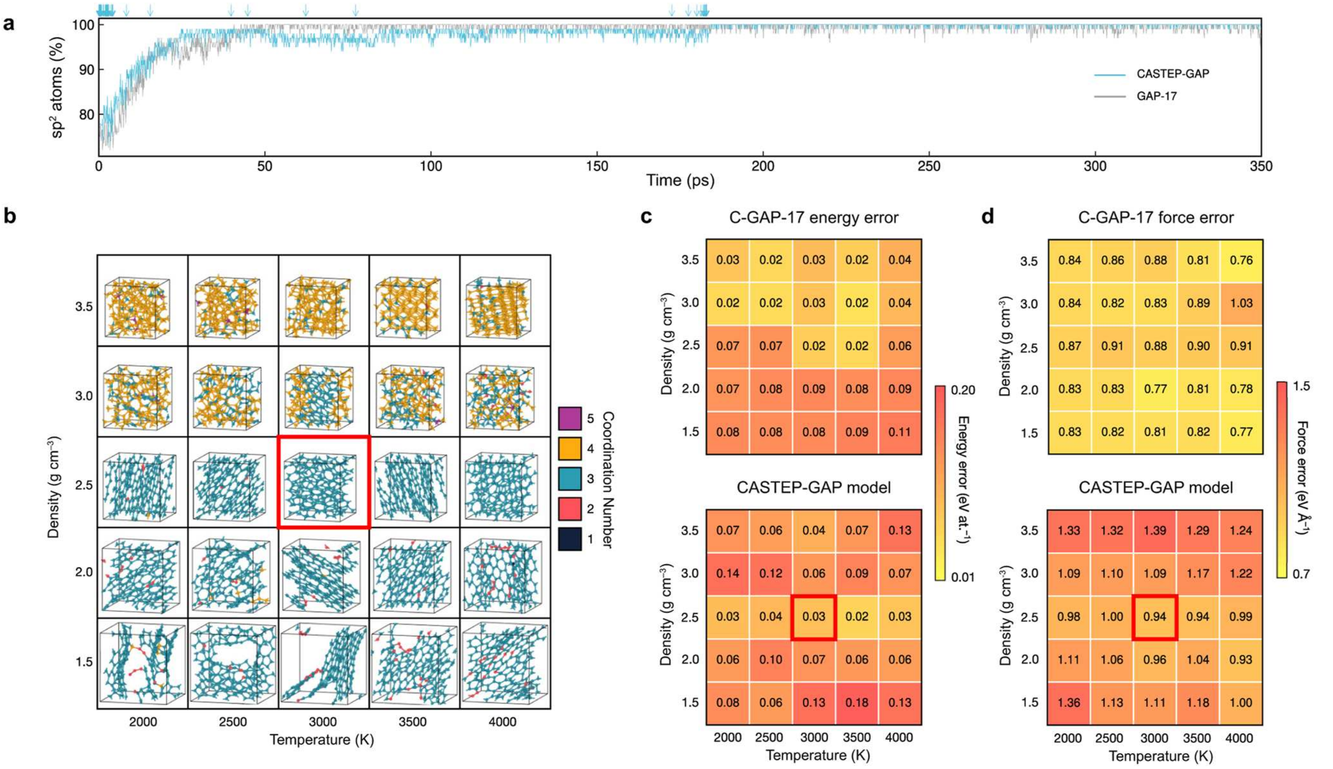
We present a case study for a complex, disordered material, as an example of a simulation that will ultimately require larger simulation cell sizes than would be amenable to DFT at all. This section, therefore, exemplifies the second "mode of operation" in Fig. 1(c): generating a potential with CASTEP + ML, which is then taken out and used to drive stand-alone simulations. Our application case is amorphous carbon (a-C), whose structures can have pore radii of several nm, and a-C films can be described in deposition simulations.[45,46] The ML-driven computational modeling of a-C materials is relevant for diverse applications, e.g., in energy storage: porous carbon structures can host metal ions (for battery anodes), molecular species (for supercapacitors), and so on.[47–51]

The thermal annealing of a-C structures is a benchmark challenge for empirical and ML interatomic potentials.[52–54] Starting from a disordered precursor, these simulations involve the breaking and forming of bonds typically over hundreds of thousands of MD steps, resulting in a gradually more ordered network that tends toward an sp^2-rich, graphitic structure at lower densities (1.5–2.0 g cm^{-3}) or an sp^3-rich, diamond-like structure at higher densities (3.0–3.5 g cm^{-3}). Meeting this benchmark is challenging for carbon potentials because of the diverse atomic environments in the amorphous state.

As mentioned above, on-the-fly fitting needs a different strategy compared to a "typical" GAP fit to an existing database of structures. Here, rather than starting from the existing C-GAP-17 database,[38] we show that it is possible to start with a small amount of fitting data sampled at relatively few sparse points when fitting on-the-fly and yet achieve accuracy to within tens of meV for a specific use-case. While this error would normally be considered relatively high for ML potentials,[55] we note that modeling a-C is challenging due to its complex configurational space and that C-GAP-17 has been shown to be a robust model for a-C.[54] CASTEP-GAP produces potentials of comparable quality, but will be less transferable due to the much smaller training database.

For this test, we took a 200-atom a-C structure generated by melt-quenching with C-GAP-17 and annealed it at 3000 K for 350 ps using two different methods: (i) direct MD with C-GAP-17 and (ii) CASTEP MD accelerated with on-the-fly GAP fitting. We chose a density of 2.5 g cm^{-3}, slightly higher than that of graphite, but not high enough to form a dense diamond-like structure—it might, therefore, be viewed as one of the more challenging cases. The `hybrid-md` settings used are those in Listing 1; technical details are given in the Appendix.

Figure 4(a) shows that the reference C-GAP-17 simulation and the new CASTEP + ML run exhibit qualitatively similar trends and physically correct behavior with respect to the fraction of sp^2 atoms. The latter increases rapidly during the first 25 ps, followed by a



C-GAP-17 energy error

Density (g cm^{-3}) \ Temperature (K)	2000	2500	3000	3500	4000
3.5	0.03	0.02	0.03	0.02	0.04
3.0	0.02	0.02	0.03	0.02	0.04
2.5	0.07	0.07	0.02	0.02	0.06
2.0	0.07	0.08	0.09	0.08	0.09
1.5	0.08	0.08	0.08	0.09	0.11

CASTEP-GAP model

Density (g cm^{-3}) \ Temperature (K)	2000	2500	3000	3500	4000
3.5	0.07	0.06	0.04	0.07	0.13
3.0	0.14	0.12	0.06	0.09	0.07
2.5	0.03	0.04	0.03	0.02	0.03
2.0	0.06	0.10	0.07	0.06	0.06
1.5	0.08	0.06	0.13	0.18	0.13

C-GAP-17 force error

Density (g cm^{-3}) \ Temperature (K)	2000	2500	3000	3500	4000
3.5	0.84	0.86	0.88	0.81	0.76
3.0	0.84	0.82	0.83	0.89	1.03
2.5	0.87	0.91	0.88	0.90	0.91
2.0	0.83	0.83	0.77	0.81	0.78
1.5	0.83	0.82	0.81	0.82	0.77

CASTEP-GAP model

Density (g cm^{-3}) \ Temperature (K)	2000	2500	3000	3500	4000
3.5	1.33	1.32	1.39	1.29	1.24
3.0	1.09	1.10	1.09	1.17	1.22
2.5	0.98	1.00	0.94	0.94	0.99
2.0	1.11	1.06	0.96	1.04	0.93
1.5	1.36	1.13	1.11	1.18	1.00

FIG. 4. Graphitization of amorphous carbon as a benchmark for ML-accelerated MD. (a) Evolution of the count of sp^2 (threefold-connected) atoms at 3000 K for a 200-atom system at a density of 2.5 g cm^{-3}. Annealing within CASTEP + ML (on-the-fly GAP model fit; blue line) and C-GAP-17 in LAMMPS (gray line) shows similar profiles starting from the same structure. Blue arrows indicate refitting events within CASTEP + ML. (b) A matrix visualizing annealed carbon structures obtained using C-GAP-17, which are here used to create a DFT benchmark set. Structures are color-coded by the atomic coordination numbers. Higher-density structures appear more diamond-like in nature, whereas lower-density structures appear graphitic. The central structure is highlighted in red: this corresponds to the density and temperature settings chosen for the CASTEP + ML simulation. (c) Energy error matrix comparing potentials with DFT. *Top*: Energy RMSE of C-GAP-17 on 10 unique structures at 500 ps. *Bottom*: Same for the CASTEP-GAP potential. (d) Force error matrix comparing potentials with DFT. *Top*: Force RMSE of C-GAP-17 on 10 unique structures at 500 ps. *Bottom*: Same for the CASTEP-GAP potential.

slower increase until converging at ∼180 ps, up to numerical fluctuations. The simulation driven by the original C-GAP-17 model shows a similar profile while reaching convergence faster, at around 60 ps. Figure 4(a) indicates by arrows when the CASTEP-GAP model was refit, with a total of 255 refits out of 404 DFT checks throughout the simulation. Several refitting events occurred near the beginning of the run; they then gradually spread out according to the adaptive refitting scheme, up until around 170 ps where an increasing number of refitting events occurs. After ∼180 ps, the simulation required restarting, and at that point, refitting was manually disabled due to the perceived stability of the potential, allowing for the simulation to continue with the ML potential without further refitting.

To analyze the quality of the resulting potential, we introduce here a systematic benchmark for disordered carbon across the range of densities and degrees of ordering [Fig. 4(b)]. These structures consist of 200 atoms and were generated using C-GAP-17 following the same annealing protocol. Ten independent structures were DFT-labeled per density–temperature run at the end of the annealing process (500 ps) for a total of 250 uncorrelated cells (50 000 atoms) in this benchmark set. The DFT parameters were adjusted for the benchmark set to match that used in C-GAP-17[38] for direct comparison; we note that, in future work, we expect to re-label the dataset at higher levels of DFT. The root mean square error (RMSE) against DFT for the energies and forces were computed for both the CASTEP-GAP model [Figs. 4(c) and 4(d), bottom] and for C-GAP-17 (top). At the chosen temperature and density, the energy accuracy of the CASTEP-GAP model is within 30 meV/at., and the force error is below 1 eV/Å, as highlighted by a square in Fig. 4(c) (bottom). C-GAP-17 reaches an energy error of ≈20 meV/at. and a force error below 0.9 eV/at., being fitted with many more SOAP sparse points and a larger database. As a result of the latter, C-GAP-17 is better suited to capturing a range of amorphous structures as indicated by the lower energy errors across most of the structures with no anomalous results [Fig. 4(c), top]. This is further illustrated by the force errors observed in C-GAP-17 [Fig. 4(d), top] where the majority of errors are below 1 eV/Å, compared with the CASTEP-GAP model [Fig. 4(d), top], which has relatively poor force accuracy since force errors were not accounted for during the refitting criterion (see Listing 1). Changes in temperature and density have a significant impact on the numerical accuracy of the CASTEP-GAP potential. Notably, at densities of 1.5 and 3.5 g cm^{-3}, we observe significant errors. The reason behind these errors is the absence of higher- and lower-density structures in the CASTEP-GAP database. Consequently, diamond-like structures are evaluated with poor numerical accuracy in terms of forces when compared to C-GAP-17. Additionally, structures at lower densities display a greater variety of chemical environments (e.g., chains, pores, and low-coordination environments) that are absent from the CASTEP-GAP database. This lack of representation results in higher energy and force errors.

One centrally important aspect of on-the-fly ML potential fitting is the ability to extract the potential for use in an external MD engine to run larger simulations.[22,23] We exemplify this approach here: we scaled up the system size to 10 000 atoms [Figs. 5(a) and 5(b)] and generated an a-C precursor at 2.5 g cm^{-3} using the same protocol as for the 200-atom one. The CASTEP-GAP potential was then used in LAMMPS[42] to anneal the 10 000-atom structure at 3000 K for 350 ps. The final annealed structure is shown in Fig. 5(b), color-coded by SOAP similarity to diamond,[40] where a value of 1 indicates a local chemical environment identical to that in bulk diamond. Qualitatively, the structure in Fig. 5(b) shows the expected warped graphitic layers at this density, where it is too dense to form flat, ordered graphitic sheets, but not dense enough to form a bulk tetrahedral (diamond-like) structure. Notably, there are locally two regions with that characteristic, highlighted in yellow in Fig. 5(b). This type of behavior has been observed with C-GAP-17 previously (along with other empirical potentials)[54,57] and ascribed to numerical artifacts arising from interactions between the cutoff function and the second-neighbor peak in the radial distribution function.[57]

For comparison, the same 10 000-atom structure was also annealed using C-GAP-17. The shortest-path ring counts for both

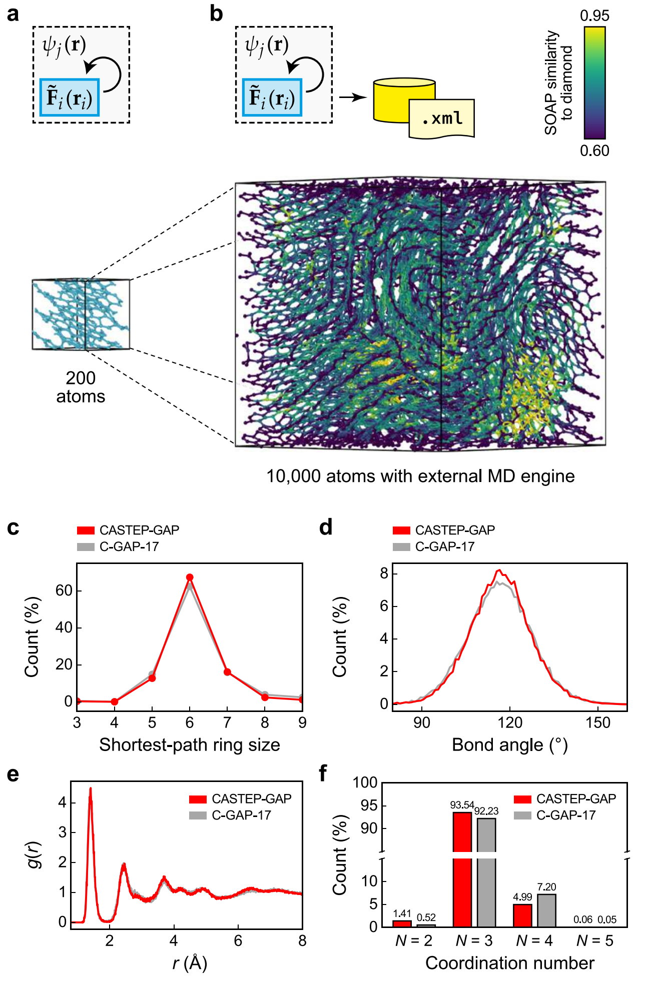


FIG. 5. Using the on-the-fly generated potential to simulate a 10 000-atom graphitization. (a) 200-atom structure annealed using CASTEP + ML. (b) 10 000-atom structure generated using the on-the-fly potential within LAMMPS. The structure is color coded according to per-atom SOAP similarity to bulk diamond. (c) Shortest-path ring size count as determined using the R.I.N.G.S software.[56] (d) Bond-angle distribution. (e) Radial distribution function. (f) Coordination number count. The results in (c)–(f) are shown for a C-GAP-17-driven simulation (gray) and for a comparable simulation driven by the on-the-fly-generated CASTEP + ML potential (red).

structures give nearly identical trends, where the structure generated from the on-the-fly potential has slightly more six-membered rings [Fig. 5(c)]. Both bond angle distributions [Fig. 5(d)] are centered around 120°, emphasizing the presence of sp^2 environments; both radial distribution functions [Fig. 5(e)] have a first peak at ≈1.41 Å, consistent with the experimental bond length in graphene (1.42 Å),[58,59] along with a second, slightly broader peak centered at 2.44 Å, corresponding to next-nearest neighbors, and a smaller peak at 3.77 Å. Finally, an assessment of coordination numbers [Fig. 5(f)] shows that both annealed structures have a high percentage of sp^2 environments (>92%). However, the structure generated with the on-the-fly potential has fewer sp^3 environments compared to the C-GAP-17 one. Note that there is a very small amount of five-coordinated atoms in both structures (0.06% and 0.05%, respectively); for a discussion of five-coordinated atoms in simulations of amorphous carbon, see Ref. 46.

Overall, the on-the-fly ML potential shows good performance in a scaled-up simulation using an external MD engine. As demonstrated in Fig. 5, the potential yields structural metrics that are comparable to the predictions of C-GAP-17 while requiring only a fraction of the fitting cost. Additionally, little human effort was expended for the construction of the training database and the fitting procedure itself. Numerical validation across densities and temperatures [Fig. 4(c)] showed that the generated potential performs similarly to C-GAP-17 for some configurations, but has a higher error for others for which it has not been "trained." We emphasize that to be considered fully validated (for production simulations), the robustness of the potential would now need to be more comprehensively tested—e.g., by repeating the same structural validation across various densities and temperatures.

VI. CONCLUSIONS

Combining DFT-driven molecular dynamics with ML acceleration can, in principle, offer the best of two worlds: accurate and reliable data for chemically complex systems "out of the box" and fast simulations at the same time. We have here described the practical implementation of a hybrid AIMD + ML framework, with the aim to provide a link between the CASTEP first-principles simulation code on one hand and any user-defined fitting method on the other hand. The connection is made using a standalone Python code to facilitate usability and future modifications. The developments described herein are implemented in CASTEP as of version 22.1, which is freely available for academic research.

One major direction for future work in this area relates to the choice of error prediction and uncertainty quantification algorithms. For now, we have focused on using generic energy and force error criteria to determine at which points of the simulation to refit the model. This means that the method can, in principle, be interfaced with any type of ML potential model beyond GAP: one may envision an extension to fast linear-fitted potentials (e.g., Refs. 60–62) or to recently developed equivariant neural-network models,[44,63] and interfaces for this purpose have begun to be implemented in the code (Sec. III C). Beyond this scheme, it would be interesting in future work to explore the comparison between the direct DFT evaluation, on one hand, and predictive error estimators and uncertainty quantification, on the other hand—the latter having been successfully used in Refs. 17 and 34, for example.

In addition to providing a reference for the implementation of CASTEP + ML, our work touches on more generally relevant points about the future development and adaptation of ML potential models in the wider community. One point is the democratization of the usage of those models: initially having required substantial expert knowledge, it is now increasingly possible to investigate a given material with the same ease that characterizes popular AIMD codes. A second point is the expected capability of those approaches not "just" to speed up MD but to provide efficient sampling for developing more generic and wider-ranging fitting databases. The ML potentials generated by an on-the-fly scheme can be extracted and then run on their own, with much more efficient simulations and larger system sizes than if they were integrated in the AIMD loop directly. We expect that these overarching directions will be of interest to the materials modeling community in the years ahead.

ACKNOWLEDGMENTS

T.K.S. acknowledges the support from the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement No. 957189 (BIG-MAP project) and the UKCP Consortium's Code Development Internship Scheme 2020 and assistance by Dr. William C. Witt with ACE & MACE model compilation and training. We are grateful for the support from the EPSRC Centre for Doctoral Training in Theory and Modeling in Chemical Sciences (TMCS) under Grant No. EP/L015722/1 (Z.E.-M.). This paper conforms to the RCUK data management requirements. G.L. was supported through the EPSRC Centre for Doctoral Training in Graphene Technology, Grant No. EP/L016087/1, and Versarien ltd. J.D.M. acknowledges funding from the EPSRC Centre for Doctoral Training in Inorganic Chemistry for Future Manufacturing (OxICFM), Grant No. EP/S023828/1. A.P.B. and M.I.J.P. acknowledge the support from the CASTEP-USER project, funded by the Engineering and Physical Sciences Research Council under Grant Agreement No. EP/W030438/1. A.P.B. also acknowledges the support from the NOMAD Centre of Excellence (European Commission Grant Agreement No. ID 951786). We are grateful for the computational support from the UK Materials and Molecular Modeling Hub, which is partially funded by EPSRC (Grant Nos. EP/P020194 and EP/T022213), as well as the UK national high performance computing service, ARCHER2, for which access was obtained via the UKCP consortium and funded by EPSRC Grant Reference Nos. EP/P022561/1 and EP/X035891/1.

AUTHOR DECLARATIONS

Conflict of Interest

A.P.B. and G.C. are listed as inventors on a patent filed by Cambridge Enterprise Ltd. related to SOAP and GAP (US patent 8843509, filed on 5 June 2009 and published on 23 September 2014). A.P.B. and M.I.J.P. declare the receipt of income from commercial sales of CASTEP by Biovia Inc. The other authors have no conflicts to disclose.

Author Contributions

Tamás K. Stenczel: Investigation (equal); Software (lead); Writing – original draft (equal). **Zakariya El-Machachi**: Investigation (equal); Validation (lead); Writing – original draft (equal). **Guoda Liepuoniute**: Investigation (supporting). **Joe D. Morrow**: Investigation (supporting). **Albert P. Bartók**: Software (supporting). **Matt I. J. Probert**: Software (supporting); Writing – review & editing (supporting). **Gábor Csányi**: Conceptualization (equal); Supervision (equal); Writing – review & editing (supporting). **Volker L. Deringer**: Conceptualization (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (lead).

DATA AVAILABILITY

The code is available as a part of the current academic release of CASTEP, for which a license may be freely obtained for academic use (http://www.castep.org/). A stand-alone version of the code is available at https://github.com/stenczelt/CASTEP-ML-data. Data supporting the findings of this study are openly available in the same repository.

APPENDIX: TECHNICAL DETAILS

In Table II, we list the hyperparameters that define the structural descriptors and the number of representative points, M (controlled by the keyword `n_sparse`), for GAP fitting for all three model systems. The full strings (“`descriptor_str`” keyword; cf. Listing 1) are contained in the input files provided (see the Data availability statement).

In carbon graphitization simulations, for the GAP fitting parameters, we set $r_{\text{cut}} = 3.7$ Å for two-body and SOAP descriptors and additionally included a shorter-ranged three-body descriptor, in analogy to the settings used for C-GAP-17.[38] We emphasize, however, that we here only used $M = 200$ for the SOAP kernel—in strong contrast to C-GAP-17 ($M = 4030$).[38] All DFT computations for carbon simulations were performed in the local density approximation.[64] For CASTEP + ML, a plane wave cutoff of 550 eV was used with a 2 × 2 × 2 k-point grid and a Gaussian smearing width of 0.1 eV. The SCF halting criterion was $\Delta E < 10^{-5}$ eV at.$^{-1}$. The DFT parameters for C-GAP-17 can be found in Ref. 38.

TABLE II. GAP descriptor hyperparameters for the three systems studied in the present work.

		Al_2O_3	l-Si	a-C
Two-body	`cutoff` (Å)	4.5	10.0	3.7
	`n_sparse`	20	30	15
Three-body	`cutoff` (Å)			3.0
	`n_sparse`			50
SOAP	`cutoff` (Å)	3.5	5.0	3.7
	`atom_sigma` (Å)	0.4	0.5	0.5
	`n_max`	8	12	8
	`l_max`	4	4	8
	`n_sparse`	1000	500	200

REFERENCES

[1] D. Marx and J. Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods* (Cambridge University Press, Cambridge, 2009).

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

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

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

[5] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik, “Machine-learned potentials for next-generation matter simulations,” Nat. Mater. **20**, 750–761 (2021).

[6] 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,” Chem. Rev. **121**, 10142–10186 (2021).

[7] P. Rowe, V. L. Deringer, P. Gasparotto, G. Csányi, and A. Michaelides, “An accurate and transferable machine learning potential for carbon,” J. Chem. Phys. **153**, 034702 (2020).

[8] A. P. Bartók, J. Kermode, N. Bernstein, and G. Csányi, “Machine learning a general-purpose interatomic potential for silicon,” Phys. Rev. X **8**, 041048 (2018).

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

[10] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, and A. E. Roitberg, “Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning,” Nat. Commun. **10**, 2903 (2019).

[11] Y. Zhou, W. Kirkpatrick, and V. L. Deringer, “Cluster fragments in amorphous phosphorus and their evolution under pressure,” Adv. Mater. **34**, 2107515 (2022).

[12] S. Takamoto, C. Shinagawa, D. Motoki, K. Nakago, W. Li, I. Kurata, T. Watanabe, Y. Yayama, H. Iriguchi, Y. Asano, T. Onodera, T. Ishii, T. Kudo, H. Ono, R. Sawada, R. Ishitani, M. Ong, T. Yamaguchi, T. Kataoka, A. Hayashi, N. Charoenphakdee, and T. Ibuka, “Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements,” Nat. Commun. **13**, 2991 (2022).

[13] C. Chen and S. P. Ong, “A universal graph deep learning interatomic potential for the periodic table,” Nat. Comput. Sci. **2**, 718–728 (2022).

[14] D. Zhang, H. Bi, F.-Z. Dai, W. Jiang, L. Zhang, and H. Wang, “DPA-1: Pretraining of attention-based deep potential model for molecular simulation,” arXiv:2208.08236 [physics.chem-ph] (2022).

[15] G. Kresse and J. Hafner, “*Ab initio* molecular dynamics for liquid metals,” Phys. Rev. B **47**, 558–561 (1993).

[16] G. Kresse and J. Furthmüller, “Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set,” Phys. Rev. B **54**, 11169–11186 (1996).

[17] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, and M. Bokdam, “Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with Bayesian inference,” Phys. Rev. Lett. **122**, 225701 (2019).

[18] R. Jinnouchi, F. Karsai, and G. Kresse, “On-the-fly machine learning force field generation: Application to melting points,” Phys. Rev. B **100**, 014105 (2019).

[19] R. Jinnouchi, F. Karsai, C. Verdi, R. Asahi, and G. Kresse, “Descriptors representing two- and three-body atomic distributions and their effects on the accuracy of machine-learned inter-atomic potentials,” J. Chem. Phys. **152**, 234102 (2020).

[20] R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, and R. Asahi, “On-the-fly active learning of interatomic potentials for large-scale atomistic simulations,” J. Phys. Chem. Lett. **11**, 6946–6955 (2020).

[21] 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,” Z. Kristallogr. **220**, 567–570 (2005).

[22] J. Lahnsteiner and M. Bokdam, “Anharmonic lattice dynamics in large thermodynamic ensembles with machine-learning force fields: $CsPbBr_3$, a phonon liquid with Cs rattlers,” Phys. Rev. B **105**, 024302 (2022).

[23] R. Jinnouchi, “Molecular dynamics simulations of proton conducting media containing phosphoric acid,” Phys. Chem. Chem. Phys. **24**, 15522–15531 (2022).

[24] R. Jinnouchi, S. Minami, F. Karsai, C. Verdi, and G. Kresse, "Proton transport in perfluorinated ionomer simulated by machine-learned interatomic potential," J. Phys. Chem. Lett. **14**, 3581–3588 (2023).

[25] P. Liu, C. Verdi, F. Karsai, and G. Kresse, "α-β phase transition of zirconium predicted by on-the-fly machine-learned force field," Phys. Rev. Mater. **5**, 053804 (2021).

[26] C. Verdi, F. Karsai, P. Liu, R. Jinnouchi, and G. Kresse, "Thermal transport and phase transitions of zirconia by on-the-fly machine-learned interatomic potentials," npj Comput. Mater. **7**, 156 (2021).

[27] P. Liu, C. Verdi, F. Karsai, and G. Kresse, "Phase transitions of zirconia: Machine-learned force fields beyond density functional theory," Phys. Rev. B **105**, L060102 (2022).

[28] P. Liu, J. Wang, N. Avargues, C. Verdi, A. Singraber, F. Karsai, X.-Q. Chen, and G. Kresse, "Combining machine learning and many-body calculations: Coverage-dependent adsorption of CO on Rh(111)," Phys. Rev. Lett. **130**, 078001 (2023).

[29] G. Csányi, T. Albaret, M. C. Payne, and A. De Vita, "'Learn on the fly': A hybrid classical and quantum-mechanical molecular dynamics simulation," Phys. Rev. Lett. **93**, 175503 (2004).

[30] J. R. Kermode, T. Albaret, D. Sherman, N. Bernstein, P. Gumbsch, M. C. Payne, G. Csányi, and A. De Vita, "Low-speed fracture instabilities in a brittle crystal," Nature **455**, 1224–1227 (2008).

[31] E. V. Podryabinkin and A. V. Shapeev, "Active learning of linearly parametrized interatomic potentials," Comput. Mater. Sci. **140**, 171–180 (2017).

[32] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, "Less is more: Sampling chemical space with active learning," J. Chem. Phys. **148**, 241733 (2018).

[33] L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E, "Active learning of uniformly accurate interatomic potentials for materials simulation," Phys. Rev. Mater. **3**, 023804 (2019).

[34] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, "On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events," npj Comput. Mater. **6**, 20 (2020).

[35] J. Vandermause, Y. Xie, J. S. Lim, C. J. Owen, and B. Kozinsky, "Active learning of reactive Bayesian force fields applied to heterogeneous catalysis dynamics of H/Pt," Nat. Commun. **13**, 5183 (2022).

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

[37] W. J. Szlachta, A. P. Bartók, and G. Csányi, "Accuracy and transferability of Gaussian approximation potential models for tungsten," Phys. Rev. B **90**, 104108 (2014).

[38] V. L. Deringer and G. Csányi, "Machine learning based interatomic potential for amorphous carbon," Phys. Rev. B **95**, 094203 (2017).

[39] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian process regression for materials and molecules," Chem. Rev. **121**, 10073–10141 (2021).

[40] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).

[41] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, "Physics-inspired structural representations for molecules and materials," Chem. Rev. **121**, 9759–9815 (2021).

[42] 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," Comput. Phys. Commun. **271**, 108171 (2022).

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

[44] I. Batatia, D. P. Kovacs, G. N. C. Simm, C. Ortner, and G. Csanyi, "MACE: Higher order equivariant message passing neural networks for fast and accurate force fields," in *Advances in Neural Information Processing Systems*, edited by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho (Curran Associates, Inc., 2022).

[45] N. Marks, "Modelling diamond-like carbon with the environment-dependent interaction potential," J. Phys.: Condens. Matter **14**, 2901 (2002).

[46] M. A. Caro, G. Csányi, T. Laurila, and V. L. Deringer, "Machine learning driven simulated deposition of carbon films: From low-density to diamondlike amorphous carbon," Phys. Rev. B **102**, 174201 (2020).

[47] V. L. Deringer, C. Merlet, Y. Hu, T. H. Lee, J. A. Kattirtzi, O. Pecher, G. Csányi, S. R. Elliott, and C. P. Grey, "Towards an atomistic understanding of disordered carbon electrode materials," Chem. Commun. **54**, 5988–5991 (2018).

[48] J.-X. Huang, G. Csányi, J.-B. Zhao, J. Cheng, and V. L. Deringer, "First-principles study of alkali-metal intercalation in disordered carbon electrode materials," J. Mater. Chem. A **7**, 19070–19080 (2019).

[49] E. H. Lahrar, A. Belhboub, P. Simon, and C. Merlet, "Ionic liquids under confinement: From systematic variations of the ion and pore sizes toward an understanding of the structure and dynamics in complex porous carbons," ACS Appl. Mater. Interfaces **12**, 1789 (2020).

[50] X. Wang, Z. Wang, F. P. García de Arquer, C.-T. Dinh, A. Ozden, Y. C. Li, D.-H. Nam, J. Li, Y.-S. Liu, J. Wicks, Z. Chen, M. Chi, B. Chen, Y. Wang, J. Tam, J. Y. Howe, A. Proppe, P. Todorović, F. Li, T.-T. Zhuang, C. M. Gabardo, A. R. Kirmani, C. McCallum, S.-F. Hung, Y. Lum, M. Luo, Y. Min, A. Xu, C. P. O'Brien, B. Stephen, B. Sun, A. H. Ip, L. J. Richter, S. O. Kelley, D. Sinton, and E. H. Sargent, "Efficient electrically powered CO_2-to-ethanol via suppression of deoxygenation," Nat. Energy **5**, 478 (2020).

[51] Y. Wang, Z. Fan, P. Qian, T. Ala-Nissila, and M. A. Caro, "Structure and pore size distribution in nanoporous carbon," Chem. Mater. **34**, 617–628 (2022).

[52] R. C. Powles, N. A. Marks, and D. W. M. Lau, "Self-assembly of sp^2-bonded carbon nanostructures from amorphous precursors," Phys. Rev. B **79**, 075430 (2009).

[53] C. de Tomas, I. Suarez-Martinez, and N. A. Marks, "Graphitization of amorphous carbons: A comparative study of interatomic potentials," Carbon **109**, 681–693 (2016).

[54] C. de Tomas, A. Aghajamali, J. L. Jones, D. J. Lim, M. J. López, I. Suarez-Martinez, and N. A. Marks, "Transferability in interatomic potentials for carbon," Carbon **155**, 624–634 (2019).

[55] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, "Performance and cost assessment of machine learning interatomic potentials," J. Phys. Chem. A **124**, 731–745 (2020).

[56] S. Le Roux and P. Jund, "Ring statistics analysis of topological networks: New approach and application to amorphous GeS_2 and SiO_2 systems," Comput. Mater. Sci. **49**, 70–83 (2010).

[57] A. Aghajamali, C. de Tomas, I. Suarez-Martinez, and N. A. Marks, "Unphysical nucleation of diamond in the extended cutoff Tersoff potential," Mol. Simul. **44**, 164–171 (2018).

[58] D. R. Cooper, B. D'Anjou, N. Ghattamaneni, B. Harack, M. Hilke, A. Horth, N. Majlis, M. Massicotte, L. Vandsburger, E. Whiteway, and V. Yu, "Experimental review of graphene," Int. Scholarly Res. Not. **2012**, 501686.

[59] G. Yang, L. Li, W. B. Lee, and M. C. Ng, "Structure of graphene and its disorders: A review," Sci. Technol. Adv. Mater. **19**, 613–648 (2018).

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

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

[62] Y. Lysogorskiy, C. van der Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, "Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon," npj Comput. Mater. **7**, 97 (2021).

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

[64] J. P. Perdew and A. Zunger, "Self-interaction correction to density-functional approximations for many-electron systems," Phys. Rev. B **23**, 5048–5079 (1981).