The Journal of Chemical Physics

AENET–LAMMPS and AENET–TINKER: Interfaces for accurate and efficient molecular dynamics simulations with machine learning potentials

Cite as: J. Chem. Phys. **155**, 074801 (2021); https://doi.org/10.1063/5.0063880
Submitted: 18 July 2021 • Accepted: 28 July 2021 • Published Online: 16 August 2021

Michael S. Chen, Tobias Morawietz, Hideki Mori, et al.

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

AENET–LAMMPS and AENET–TINKER: Interfaces for accurate and efficient molecular dynamics simulations with machine learning potentials

Cite as: J. Chem. Phys. 155, 074801 (2021); doi: 10.1063/5.0063880
Submitted: 18 July 2021 • Accepted: 28 July 2021 •
Published Online: 16 August 2021

View Online Export Citation CrossMark

Michael S. Chen,[1] Tobias Morawietz,[1] Hideki Mori,[2] Thomas E. Markland,[1,a)] and Nongnuch Artrith[3,4,5,b)]

AFFILIATIONS

[1] Department of Chemistry, Stanford University, Stanford, California 94305, USA
[2] Department of Mechanical Engineering, College of Industrial Technology, 1-27-1 Nishikoya, Amagasaki, Hyogo 661-0047, Japan
[3] Department of Chemical Engineering, Columbia University, New York, New York 10027, USA
[4] Columbia Center for Computational Electrochemistry, Columbia University, New York, New York 10027, USA
[5] Materials Chemistry and Catalysis, Debye Institute for Nanomaterials Science, Utrecht University, 3584 CG Utrecht, The Netherlands

[a)] **Electronic mail:** tmarkland@stanford.edu
[b)] **Author to whom correspondence should be addressed:** n.artrith@uu.nl

ABSTRACT

Machine-learning potentials (MLPs) trained on data from quantum-mechanics based first-principles methods can approach the accuracy of the reference method at a fraction of the computational cost. To facilitate efficient MLP-based molecular dynamics and Monte Carlo simulations, an integration of the MLPs with sampling software is needed. Here, we develop two interfaces that link the atomic energy network (ænet) MLP package with the popular sampling packages TINKER and LAMMPS. The three packages, ænet, TINKER, and LAMMPS, are free and open-source software that enable, in combination, accurate simulations of large and complex systems with low computational cost that scales linearly with the number of atoms. Scaling tests show that the parallel efficiency of the ænet–TINKER interface is nearly optimal but is limited to shared-memory systems. The ænet–LAMMPS interface achieves excellent parallel efficiency on highly parallel distributed-memory systems and benefits from the highly optimized neighbor list implemented in LAMMPS. We demonstrate the utility of the two MLP interfaces for two relevant example applications: the investigation of diffusion phenomena in liquid water and the equilibration of nanostructured amorphous battery materials.

I. INTRODUCTION

Atomistic simulations based on molecular dynamics (MD) or Monte Carlo (MC) techniques[1,2] have become standard tools for the *in silico* characterization and prediction of materials and molecular properties. Due to their ability to include realistic experimental conditions, such as solvent effects, temperature, and pressure, they are routinely employed in a wide range of research areas in both academic and industrial settings. Prominent examples are the identification of novel materials for energy applications[3–7] or the design of novel drugs.[8–11] To perform these simulations, reliable interatomic potentials (or *force fields*) are required that describe the atomic interactions.[12,13] They need to be not only accurate to capture the atomistic process of interest but also efficient to generate simulations on sufficient time and length scales using reasonable compute resources. Machine-learning potentials (MLPs),[14–26] including approaches based on artificial neural networks (ANNs), learn the interatomic interactions from accurate quantum mechanical (QM) calculations, such as density-functional theory (DFT),[27] and have shown great promise in combining accuracy and affordability, allowing to simulate complex systems under realistic conditions.[11,16,28–30]

In this work, we discuss interfaces of two popular freely available simulation codes, the TINKER molecular modeling software[31] and the *Large-scale Atomic/Molecular Massively Parallel Simulator* (LAMMPS),[32] with the atomic energy network (ænet) ANN potential package.[19] In combination, these tools enable routine long-time simulations of large and complex systems with computational cost that scales linearly with system size and can be distributed across compute cores without significant loss of efficiency.

We note that there are already several MLP methods available in LAMMPS, for example, the Spectral Neighbor Analysis Potential (SNAP)[33] and the Gaussian Approximation Potential (GAP)[15] approaches, each with their own strengths and weaknesses. Potentials trained with the GAP approach provide an intrinsic estimate of the prediction error. An advantage of the SNAP method is its computational efficiency for simple (especially elemental) compounds. The Behler–Parrinello ANN potential method[14] that is the subject of the present work is computationally more demanding but can often achieve greater accuracy.[34] Another advantage of the ANN potential method is that it can be trained on large reference datasets with millions of samples, complex structures, and many chemical elements.[22]

In Sec. II, we briefly review the ANN potential method. The basic functionality of the simulation framework and the required input for incorporating ænet ANN interatomic potentials are described in Sec. III. Section IV A illustrates the need for computationally efficient simulation tools with accurate ANN potentials as demonstrated for two examples of complex atomic systems: amorphous lithium–silicon (a-LiSi) alloys and liquid water. The computational efficiency of both interfaces is discussed in Sec. IV B. All code interfaces presented here are freely available and shared, together with the corresponding input files, user instructions, and ANNs for LiSi[35,36] and liquid water.[37,38]

II. METHODS

A. The artificial neural network (ANN) potential method

A feedforward ANN is a vector function that can be expressed as the recurrence relation

$$\mathbf{x}^l = f_a^l\left(\mathbf{w}^l \cdot \mathbf{x}^{l-1} + \mathbf{b}^l\right), \tag{1}$$

where the values of the neurons $\mathbf{x}^l$ in the (l)th layer depend on the values of the neurons $\mathbf{x}^{l-1}$ in the $(l-1)$th layer and the elements of matrix $\mathbf{w}^l$ and vector $\mathbf{b}^l$ are parameter sets called *weights* and *biases*, respectively. f_a^l is the *activation function* of layer l, which acts on each component of its input vector. It has been shown that ANNs of the form of Eq. (1) can represent any function with arbitrary accuracy if suitable non-linear activation functions are used.[39]

The initial layer of the ANN, i.e., $\mathbf{x}^0$ in Eq. (1), is given by the function input (*input layer*). The final *output layer* of the ANN corresponds to the function value. The layers between the input and the output do not have any intuitive meaning and are often referred to as *hidden layers*. Together, the dimensions of all layers define the *architecture* of the ANN, e.g., n_{in}-$n_{\text{h,1}}$-$n_{\text{h,2}}$-n_{out} is the architecture of an ANN with input dimension n_{in}, output dimension n_{out}, and two hidden layers with dimensions $n_{\text{h,1}}$ and $n_{\text{h,2}}$.

Behler and Parrinello[14] introduced a technique for the representation of interatomic potentials with ANNs by expressing the total energy E of an atomic structure as the sum of atomic energies E_i,

$$E = \sum_i^{\text{atoms}} E_i \approx \sum_t^{\text{types}} \sum_i^{\text{atoms}} \text{ANN}_t\left(\tilde{\sigma}_i^{R_c}\right), \tag{2}$$

where ANN_t is an ANN trained to predict the atomic energies of atoms with type/species t. The ANN input $\tilde{\sigma}_i^{R_c}$ is a descriptor vector of the local atomic environment of atom i within a cutoff radius of R_c.

B. Descriptors of the local atomic environment

The potential energy is invariant with respect to rotation and translation of the entire structure as well as to the permutation of equivalent atoms, and the descriptor $\tilde{\sigma}_i^{R_c}$ also needs to satisfy these symmetries. Additionally, the size of the descriptor vector determines the dimension of the ANN input layer and must not vary with the number of atoms in an atomic structure. Therefore, the Cartesian atomic coordinates cannot be directly used as ANN inputs and need to be first transformed into an invariant representation.

In the present work, we use the descriptor by Artrith, Urban, and Ceder,[40] in which the radial distribution function (RDF) and angular distribution function (ADF) of the local atomic environment are expanded in a basis set of Chebyshev polynomials. The expansion coefficients are a suitable invariant descriptor of the local atomic environment and can be used as the inputs for the ANN potential. The coefficients of the RDF expansion are given by

$$c_\alpha^{\text{pair}} = \sum_{j\neq i} T_\alpha\left(\frac{2r_{ij}}{R_c} - 1\right) f_c(r_{ij}), \tag{3}$$

where r_{ij} is the atomic distance between atoms i and j, R_c is the cutoff radius, and T_α is the Chebyshev polynomial of the first kind with order α.[40] The Chebyshev polynomials T_n are defined by a recurrence relation

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \tag{4}$$

where $T_0(x)$ and $T_1(x)$ are 1 and x, respectively. We use a cosine cutoff function defined as

$$f_c(r_{ij}) = \begin{cases} \frac{1}{2}\left[\cos\left(\frac{\pi r_{ij}}{R_c}\right) + 1\right] & (r_{ij} \leq R_c), \\ 0 & (r_{ij} > R_c). \end{cases} \tag{5}$$

The coefficients from the expansion of the ADF are similarly given by

$$c_\alpha^{\text{triple}} = \sum_{j\neq i, k\neq i,j} T_\alpha\left(\cos\theta_{ijk}\right) f_c(r_{ij}) f_c(r_{ik}), \tag{6}$$

where r_{ij}, r_{ik}, and r_{jk} are the atomic distances between atoms i, j, and k and θ_{ijk} is the angle defined by the three atoms.

C. ANN potential training

Training an ANN potential requires the optimization of the weight and bias parameters $\mathbf{w}^l$ and $\mathbf{b}^l$ of Eq. (1) to reproduce reference energies as closely as possible. In the present work, this was achieved by minimizing the cost function C, defined as follows:

$$C(\{\mathbf{w},\mathbf{b}\}) = \frac{1}{2}\sum_{t=1}^{N_s}\left[E_t^{\mathrm{ANN}}(\{\mathbf{w},\mathbf{b}\}) - E_t^{\mathrm{DFT}}\right]^2, \tag{7}$$

where N_s is the number of structures in the *training set* and E_t^{ANN} and E_t^{DFT} are potential energies of the tth structure of the ANN potential and the reference DFT calculation, respectively. The limited memory Broyden–Fletcher–Goldfarb–Shanno (LM-BFGS) method was used for the actual optimization,[41–45] and the gradients of the cost function, $\frac{\partial C}{\partial \mathbf{w}}$ and $\frac{\partial C}{\partial \mathbf{b}}$, were obtained using the conventional back-propagation technique.[46]

III. IMPLEMENTATION

The ænet package[19] is a free and open-source software for the construction (training) and evaluation of ANN-based machine-learning potentials. ænet implements a variant of the high-dimensional neural network potential method by Behler and Parrinello[14] described in Sec. II. The cost of computing ænet's structure descriptors, detailed in Sec. II B, does not scale with the number of atomic species. This enables the systematic construction of efficient and accurate ANN potentials without manual parameterization for compositions with a large number of atom types.[40] With the Chebyshev descriptor, the cumbersome setup of hand-crafted descriptors is avoided and replaced by just a single line of code in the ænet input.

The ænet package can be compiled into a library (ænetLib) that is compatible with the Python, C/C++, and FORTRAN programming languages, facilitating the integration with simulation software. A Python interface to the Atomic Simulation Environment (ASE)[47] was provided in Ref. 19. ASE is a Python framework for atomistic simulations and provides a simple application programming interface (API) together with calculators for interfacing with third-party software for the evaluation of structural energies and atomic forces. The ænet package includes an implementation of an ASE calculator linked to ænetLib. A C++ interface to the LAMMPS code was also provided in Ref 25. This interface is designed for the metallurgical analysis in structural materials, such as bcc iron. For example, this interface extends ænetLib so that the stress tensor of the entire system can be calculated using the functionality in LAMMPS.[48]

In this work, we describe two interfaces with *aenetLib* coupled to the software code written in Fortran and C++: the TINKER and LAMMPS codes, respectively. These simulation software packages are designed with different use cases in mind: TINKER is implemented in the FORTRAN programming language and is structured into small command-line utilities, each with distinct purpose (e.g., MD simulations, geometry optimizations, MC simulations, etc.). From a user perspective, TINKER is comparatively easy to learn and install from the source code, which makes it a popular choice for desktop or laptop computers. The standard distribution of TINKER is *shared-memory* parallelized with the Open Multi-Processing (OpenMP)[49] standard so that simulations can make optimal use of modern consumer-grade processors. LAMMPS, on the other hand, is a highly optimized MD software designed for massively parallel high-performance computer (HPC) systems. LAMMPS input files are scriptable, which adds flexibility but also steepens the learning curve. LAMMPS is *distributed-memory* parallelized using the message passing interface (MPI) standard and can scale to thousands of distributed processor cores. We note that a TINKER distribution for HPC systems has recently been released, named Tinker-HP.[50,51] Data reported here were, however, obtained with the standard TINKER distribution.

Figure 1 shows the general workflow for performing efficient atomistic simulations with MLPs: The Cartesian coordinates of an initial atomic structure are mapped to their corresponding total energy and atomic forces via the MLP package ænet, given an MLP that is parameterized for all elements in the system. Employing an interface coupling, the ænetLib library to either the TINKER or the LAMMPS simulation package, the system of interest is propagated to a new set of coordinates under a specified set of thermodynamic conditions. Both LAMMPS and standard TINKER were interfaced with ænet by linking against the *ænetLib* library so that ænet routines are used for evaluating structural energies and interatomic forces. In Secs. III A–III D we describe the implementation and usage of these interfaces.

Note that ænet–TINKER uses the neighbor list from ænet, which is implemented by a comparatively simple algorithm, while ænet–LAMMPS makes direct use of the highly optimized LAMMPS neighbor list.

A. Building TINKER with ænet support

Three new files were created to connect the ænet libraries with TINKER: *aenettinker.f90*, *extra.f*, and *extra1.f*.

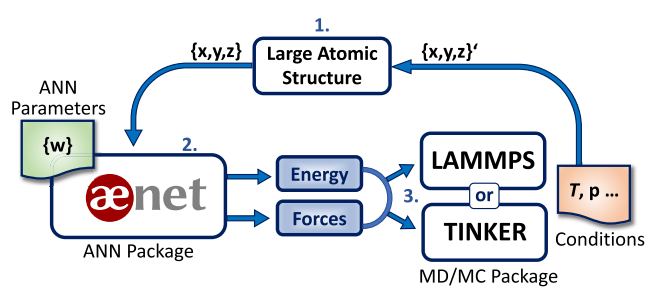

FIG. 1. Schematic overview of the ænet–LAMMPS/ænet–TINKER program interfaces for efficient atomistic simulations with artificial neural network (ANN) potentials. The Cartesian coordinates $\{x, y, z\}$ of a large (periodic or non-periodic) atomic system (1) are mapped to total energy and atomic forces by a previously trained ANN potential (2) defined by a set of model parameters $\{w\}$. Via an interface (3), between the open-source ANN code ænet,[19] using the ænet library **(ænetLib)**, and the molecular dynamics (MD)/Monte Carlo (MC) codes LAMMPS[32] or TINKER,[31] force and/or energy information are used to propagate the initial atomic positions to a new structure with coordinates $\{x, y, z\}'$ under a set of specified thermodynamic conditions, such as temperature, T, and pressure, p. The cycle repeats until a trajectory of sufficient length has been generated from which the properties of interest can be obtained.

Using an ANN potential with the TINKER MD code requires copying the following files to the TINKER source directory:

```
aenettinker.f90
extra.f
extra1.f
```

and compiling the TINKER source code using the provided *Makefile*, e.g.,

```
Makefile.aenetlib_ifort
Makefile.aenetlib_gfortran.
```

B. Running TINKER simulations using ænet potentials

To run simulations using ænet ANN potentials, the TINKER parameter file *aenet.prm* is used. It only contains the masses of all chemical species as no further information is required.

All ANN potential files have to be present in the working directory and need to follow the naming convention

```
<species>.ann,
```

where <species> is the chemical symbol (H, He, Li, etc.). In the principal TINKER key input file, ænet potentials are activated with the keyword

```
EXTRATERM only.
```

The number of threads used in parallel runs can be controlled with the keyword *OPENMP-THREADS*. We also refer to the example Tinker input file *tinker.key* on our shared repository.

C. Building LAMMPS with ænet support

Similar to the TINKER package, two new files,

```
pair_aenet.h
pair_aenet.cpp,
```

were written as part of the USER-AENET package to build ænet–LAMMPS by connecting with the ænet libraries.

To build LAMMPS with the ænet interface, simply copy the USER-AENET directory to the LAMMPS source directory and execute `make yes-user-aenet`. After that, LAMMPS can be built in the usual way, e.g., with `make mpi`.

D. Running LAMMPS simulations using ænet potentials

The ænet library files, as well as any other dependencies, need to be properly loaded (i.e., by setting the $LD_LIBRARY_PATH). The LAMMPS input script also needs to be configured so as to use the ænet *pair style* and to specify which neural network parameter (*.ann*) files to use.

A partial input example for water is provided as follows:

```
units metal
mass 1  1.007825
mass 2 15.999491
pair_style aenet H.ann O.ann
pair_coeff * *
```

The user must specify the order of the *<species>.ann* potential files such that they correspond to the mass values defined above. In the example above, element 1 is designated to be hydrogen and element 2 is designated to be oxygen. Consequently, the ænet parameter file for hydrogen (*H.ann*) needs to be specified first and then followed by the parameter file for oxygen (*O.ann*).

The selected LAMMPS unit system should match the units of the reference data used for training the MLP by ænet. In the example above, *metal* units are used, meaning that the ænet potentials were fitted to training data for which the energies were reported in electron-volts and the positions in angstroms. If, instead, the training data are provided in units of hartrees and bohrs, then the corresponding LAMMPS unit system *electron* should be chosen. Further details on the unit systems can be looked up in the LAMMPS manual.

IV. RESULTS AND DISCUSSION

A. Complex systems require efficient and accurate simulation tools

The accurate modeling of complex materials at experimental conditions often requires the use of large simulation cells containing thousands of atoms. Long simulation times on the order of nanoseconds are needed if the dynamics of the systems are of interest. In addition to these efficiency requirements, the complex bonding patterns that govern the atomic interactions need to be described by an appropriate method of sufficient accuracy. MLPs trained to accurate QM calculations can fulfill both the efficiency and the accuracy requirements. Two examples that demonstrate the need for efficient and accurate simulation tools are the transport of Li in nanostructured amorphous Si for high-energy-density batteries and diffusion phenomena in liquid water.

Diffusion in complex amorphous Li_xSi materials: Nanostructured silicon (Si) is a promising alternative to graphite as a high-capacity anode material for Li-ion batteries.[54–61] Nanoscaled materials are required to overcome mechanical limitations and avoid fracturing.[62] In addition, the rate-capability of silicon anodes is two to four orders of magnitude lower[63–65] compared to graphite anodes[66] owing to lower Li diffusivity. The lithiation of crystalline Si (c-Si) results in amorphization of the silicon crystal structure.[54] The amorphous structure of silicon anodes, the length-scale dependence of its mechanical properties, and the comparatively low lithium conductivity make the investigation of lithium transport in silicon electrodes challenging for computation and experiment. We have previously employed ANN potentials to investigate the complex phase diagram of amorphous LiSi alloys[35] and the delithiation of entire Li_xSi nanoparticles.[36] Figure 2 shows examples of different crystalline and quasi-amorphous Li_xSi structures that can occur during the lithiation and delithiation of silicon anodes, which exhibit a wide variety of structural motifs and chemical bonds.

These considerations make this system a prototypical complex material that cannot easily be modeled with either first-principles methods or conventional interatomic potentials but can be successfully described by ANN potentials.[35,36]

Diffusion in liquid water: The need for efficient simulation tools coupled to MLPs also becomes apparent when studying diffusion phenomena in complex molecular systems. One example is the transport of proton defects,[68,69] a crucial process for the

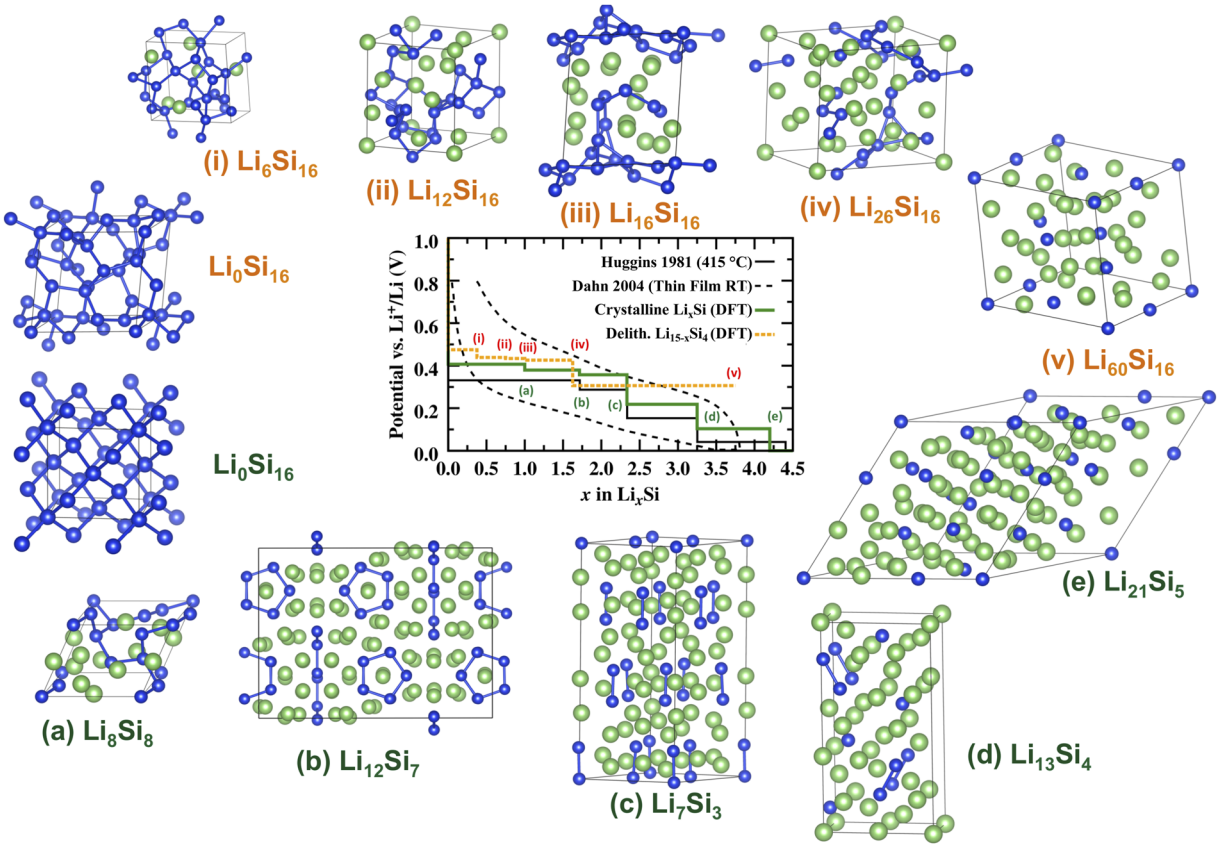

FIG. 2. Complex phases of LiSi alloys. Different crystalline and quasi-amorphous Li_xSi structures (color codes: Li = green and Si = blue) that can occur during the lithiation and delithiation of silicon anodes and exhibit a wide variety of structural motifs and chemical bonds. The relative lithium content x in the amorphous Li_xSi alloys varies during battery charge and discharge (experimental values from Refs. 52 and 53). The structure models were taken from Refs. 35 and 36.

understanding of electrochemical reactions and enzymatic mechanisms that requires a simulation method with the ability to break and form chemical bonds. *Ab initio* molecular dynamics (AIMD) simulations are, in principle, able to describe the diffusion of protons in hydrogen-bond networks but are limited to short time-scales and often neglect nuclear quantum effects.[70] As demonstrated in Fig. 3, short simulations with a length of 10s of picoseconds are not sufficient to reliably obtain proton defect diffusion coefficients in liquid water. This is a consequence of the Grotthuss mechanism in which structural diffusion occurs in a stepwise fashion with periods of rapid defect diffusion followed by rest states with little movement[71,72] that makes it necessary to follow the defect trajectories over several nanoseconds. The issue is further complicated by the fact that periodic simulation cells are usually small in size to save computing resources and therefore often contain only a single defect. Even when averaging over multiple atoms is possible, as in the case of molecular diffusion in pure liquid water, an accurate prediction and comparison to the experimentally observed property are not straightforward. While structural properties of liquid water, such as pair distribution functions, can be routinely obtained from direct AIMD simulations employing relatively small simulation cells and moderate simulation times, dynamical properties are more challenging since they have a strong system size dependence and require longer times to converge.[73]

As these two examples demonstrate, efficient tools to generate long trajectories for complex atomistic system are a prerequisite for obtaining reliable molecular and materials properties. In Sec. IV B, we show how the ænet–TINKER and ænet–LAMMPS interfaces can meet these requirements while making efficient use of modern compute hardware distributing compute load across central processing units (CPUs).

We demonstrate the computational efficiency of the combination of MLPs with the MD codes TINKER (Fig. 4) and LAMMPS (Fig. 5) for amorphous LiSi and bulk water with simulation cells of increasing size containing up to millions of atoms. ANN potentials for both systems trained to publicly available datasets[74] are provided with the implementation together with the corresponding input files and reference output to facilitate a quick set up of the simulation and guarantee reproducibility.[75] In brief, these ANNs are trained to reference DFT calculations for LiSi with the Perdew–Burke–Ernzerhof (PBE)[76] density-functional on data, covering bulk, surface, and cluster structures at various LiSi compositions; see the details in Refs. 35 and 36. The liquid water ANN was trained on liquid water structures across the full liquid temperature range described by the

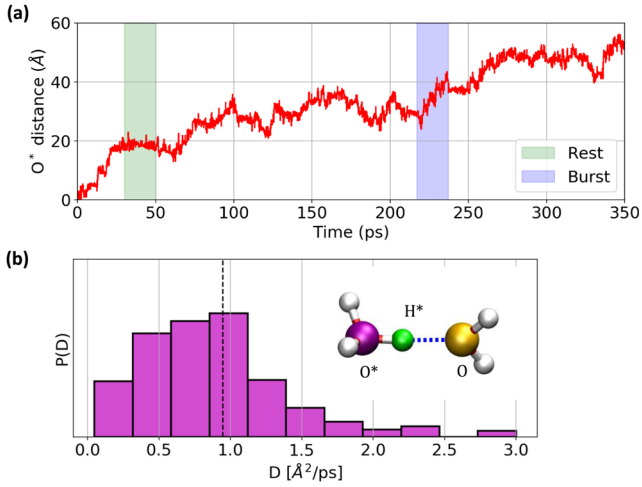

FIG. 3. Importance of long-time simulations for predicting proton defect diffusion. (a) The diffusion process of a hydronium (H_3O^+) defect in liquid water, shown as the distance of the defect position at time t_i from its position at time t_0, is characterized by alternating periods of rapid defect movement (*burst* phase highlighted in blue) and little change in position (*rest* phase highlighted in green) that can extend over 10s of picoseconds. (b) Defect diffusion coefficients obtained from 20 ps segments extracted from the 350 ps trajectory shown in (a) are distributed across a large value range, indicating that a single trajectory of that length is insufficient to estimate the proton defect diffusion coefficient. The results from periodic AIMD simulations containing 63 H_2O molecules and a single H_3O defect.[67]

revPBE-D3[76–78] density-functional. We have previously shown that an ANN potential trained to this same dataset with a different set of descriptors yields accurate results across the full liquid temperature range.[37] A recent study also found that local environments sampled in liquid water are sufficiently similar to the environments in different ice phases that an MLP trained on only liquid water configurations is able to reproduce many properties of ice.[79] For the details on the availability of potentials and datasets, see the section titled "Data Availability.[84,85]"

B. Performance of ænet–TINKER and ænet–LAMMPS interfaces

A major drawback of direct AIMD simulations is their steep increase in compute time with the number of atoms in the system. Common DFT calculations have an algorithmic scaling of order $\mathcal{O}(N^3)$ or $\mathcal{O}(N \log N)$ with the number of electrons N in the system.

To demonstrate the impact of system size on compute cost for our MLP-based simulation interfaces, we compare the compute time per atom for periodic simulation cells of crystalline LiSi, Figs. 4(a) and 5(a), and bulk H_2O, Figs. 4(c) and 5(c), ranging from a few hundred atoms all the way up to millions of atoms. All scaling tests were performed on the Columbia HPC cluster Terremoto using Dell C6420 nodes (24 cores per node) with dual Intel Xeon Gold 6126 Processors (2.6 GHz) using the Intel 2017 compiler, EDR Infiniband, and Red Hat Enterprise Linux 7.

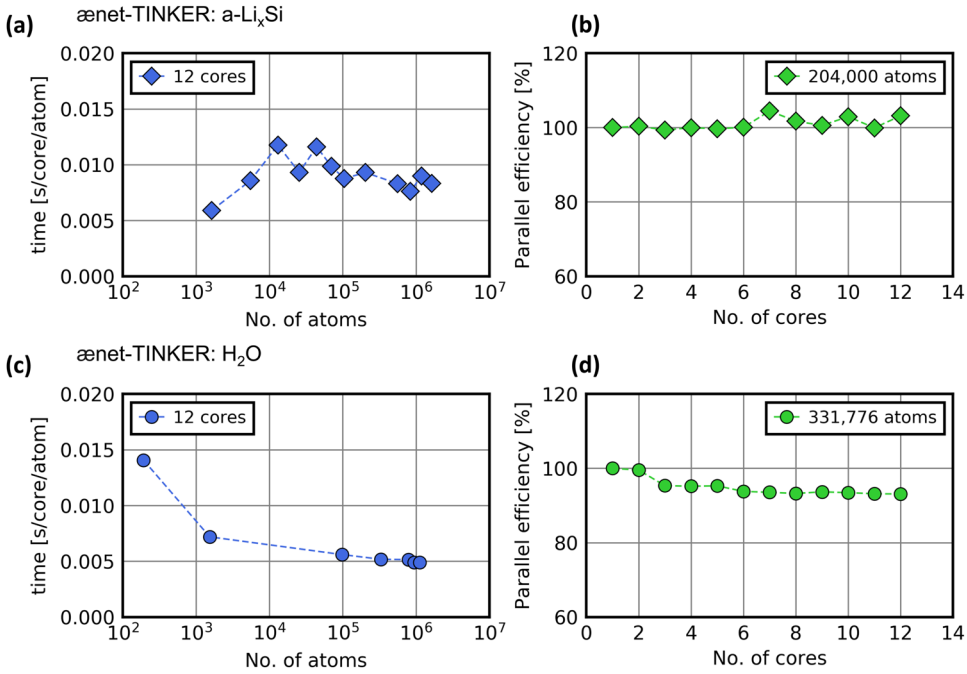

FIG. 4. Computational efficiency of molecular dynamics simulations with the ænet–TINKER interface. [(a) and (b)] *a*-LiSi and [(c) and (d)] water. Computer cluster specifications: Dell C6420 nodes with dual Intel Xeon Gold 6126 Processor (2.6 GHz) with Intel compiler 2017, EDR Infiniband, Red Hat Enterprise Linux 7.

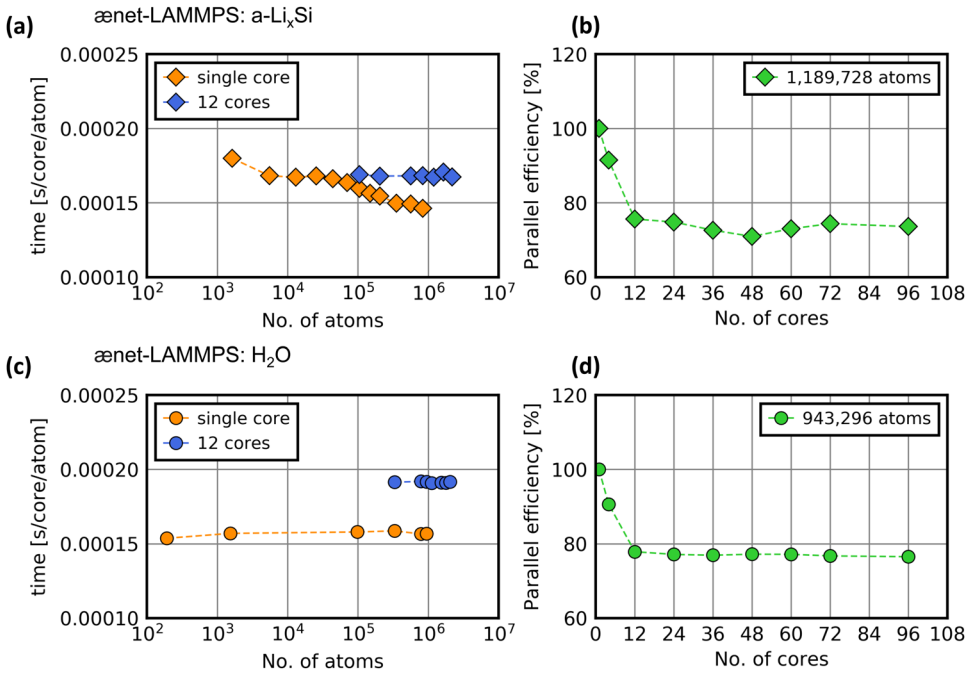

FIG. 5. Computational efficiency of molecular dynamics simulations with the ænet–LAMMPS interface. [(a) and (b)] *a*-LiSi and [(c) and (d)] water. Computer cluster specifications: Dell C6420 nodes with dual Intel Xeon Gold 6126 Processor (2.6 GHz) with Intel compiler 2017, EDR Infiniband, Red Hat Enterprise Linux 7.

As can be seen from the results in Figs. 4 and 5, computational costs for both interfaces scale strictly linearly with system size, even up to very large unit cells, which is a result of the employed ANN approach in which the total atomic interactions are represented by local, atom-centered ANNs.[14,19] Even though we evaluate performance on two very different systems (an amorphous solid and a molecular liquid), both display very similar compute requirements, with single core compute times per atoms of ~10 ms with TINKER (Fig. 4) and <1 ms for LAMMPS (Fig. 5). As discussed, the current TINKER interface is based on the original implementation making use of OpenMP parallelization. Even higher efficiency can be expected with the recently released TINKER-HP package.[50] To translate these numbers into compute times: to generate a 1 ns-long trajectory of a system of 1536 atoms (for H_2O) takes 134 h on a single compute core with ænet–LAMMPS.

We note that the overall efficiency difference of the ænet–TINKER and ænet–LAMMPS interfaces is mainly due to the evaluation of the particle neighbor lists that are used to determine which atoms are in the local atomic environment of each atom. The ænet–TINKER interface currently uses the neighbor list implemented in ænet, which is not optimized for MD simulations and is rebuilt at each step of the simulation. In contrast, the ænet–LAMMPS interface makes use of LAMMPS' highly optimized neighbor list implementation, which can reduce the computational cost for MD simulations significantly for large atomic structures.

To further reduce the wait time, the compute load can be distributed across cores. In Figs. 4(b), 4(d), 5(b), and 5(d), we report the parallel efficiency, i.e., the loss of compute power due to the distribution of tasks and the communication between the compute cores. Due to OpenMP parallelization, the ænet–TINKER simulations (Fig. 4) have an efficiency very close to 100% on a single node. For ænet–LAMMPS (Fig. 5), efficiency drops more but still stays at high values of around 70%–80% using up to 96 CPU cores on multiple nodes and with an overall better efficiency than the ænet–TINKER interface. Parallelization over multiple cores reduces the real wait time for our example system of 1536 atom to 3.6 h on 48 cores and just 1.8 h on 96 cores.

Apart from size and composition of the system of interest, the simulation performance is also dependent on the specifications of the ANN potential, which, in turn, has a direct impact on the accuracy of the calculated properties. The efficiency of ANN-based models does not depend on the size of the training dataset since the parameters that determine the potentials are only optimized once before the simulation. For ANN potentials, a main factor that determines both simulation efficiency and accuracy is the complexity of the model as defined by the number of nodes per layer. The size of the input layer is given by the number of descriptors, while the hidden layer size is a separate hyperparameter that has to be tuned during model construction.

To investigate these effects and provide guidance for the construction of new ANN potentials, we have trained two additional potentials for water with gradually decreased complexity. The optimal ANN potential model that was used for benchmarking the computational efficiency in Figs. 4 and 5 and the diffusion analysis shown in Sec. IV D has a 52-25-25-1 architecture. First, the size of the descriptor was reduced from 52 to 20 dimensions, and then, in addition, the hidden layer width was decreased from 25 to 5, i.e., the

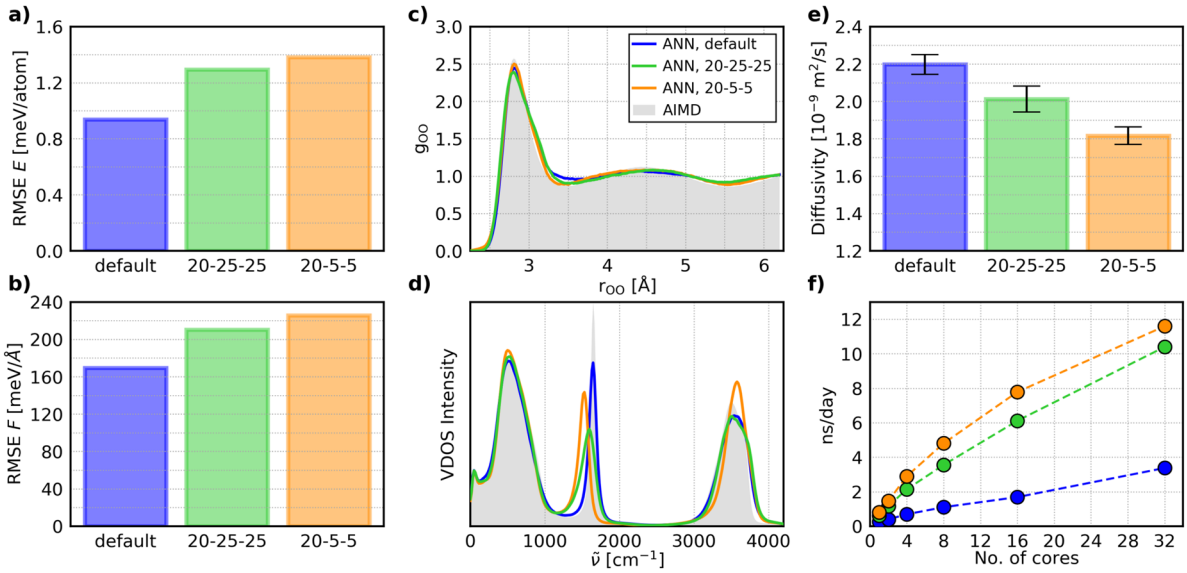

FIG. 6. Impact of model complexity on accuracy and simulation efficiency. Test set accuracy of energies (a) and forces (b) for three ANN potentials with decreasing model complexity (specified as [number of input nodes] – [number of nodes in hidden layer 1] – [number of nodes in hidden layer 2]; default model complexity: 52-25-25). Oxygen–oxygen radial distribution functions (c) and vibrational density of states (VDOS) of the hydrogen atoms (d) compared to results from direct AIMD simulations.[80] (e) System size dependent diffusion coefficients, $D(L)$, at a box size of 64 H_2O molecules. (f) Simulation time in nanoseconds accumulated on one day of compute as a function of the number of employed cores.

architecture was reduced to 20-25-25-1 and 20-5-5-1, respectively. In Fig. 6, the performance of the reduced models is compared to the optimal model (52-25-25-1) as a default.

As can be seen, the descriptor size is a major factor determining model accuracy [Figs. 6(a)–6(e)] and simulation speed [Fig. 6(f)]. While cutting the descriptor leads to a simulation speed-up of a factor of 3 [Fig. 6(f)], the energy and force accuracy decrease [Figs. 6(a) and 6(b)] and deviations in dynamics properties become visible. Although the structural properties (as exemplified by the oxygen–oxygen RDF) show only a low variance across models [Fig. 6(c)], the reduced model shows a broadening and shift of the OH bending region of the hydrogen vibrational density of states [VDOS, Fig. 6(d)]. Further simplifying the ANN potential by reducing the hidden layer width only marginally improves the efficiency but results in greater deviations of the VDOS, now also affecting the OH stretch region.

In the following, we directly demonstrate the impact of the two efficient simulation interfaces: making use of ænet–TINKER to equilibrate a large nanoparticle of LiSi and using ænet–LAMMPS to calculate the liquid water diffusivity at experimental conditions.

C. ænet–TINKER applied to LiSi

To demonstrate the feasibility of large-scale simulations of nanostructures, we performed an MD equilibration simulation of a Li_xSi nanoparticle with 11 000 atoms (8 nm diameter) at 500 K using the ænet–TINKER interface. The initial structure (truncated from the $Li_{15}Si_4$ crystal structure) and the final equilibrated structure are shown in Fig. 7. The progression of the potential energy during the MD simulation and its gliding average over 10 ps is illustrated in Fig. 8. After around 2 ns, the potential energy is within 2 meV/atom of the final structure, indicating that the system has reached thermal equilibrium. This example further illustrates the need for efficient long-time MD simulations to investigate the properties of LiSi anodes.[35,36]

D. ænet–LAMMPS applied to H_2O

As an additional demonstration of the benefits of an efficient, linear scaling MD setup for ab initio-quality simulations, we report results on molecular diffusion in liquid water, employing the

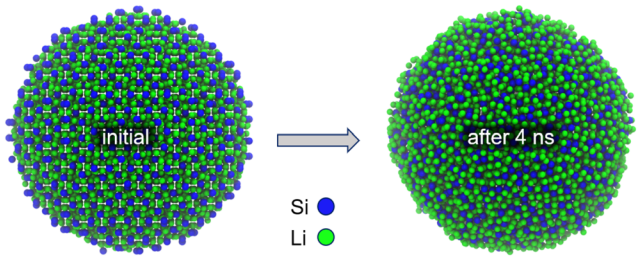

FIG. 7. LiSi nanoparticle equilibrated with the ænet–TINKER interface. Molecular dynamics (MD) simulation of a $Li_{8429}Si_{2571}$ nanoparticle structure (11 000 atoms, diameter 8 nm). Left: initial structure before MD; right: final structure after 4 ns MD simulation.

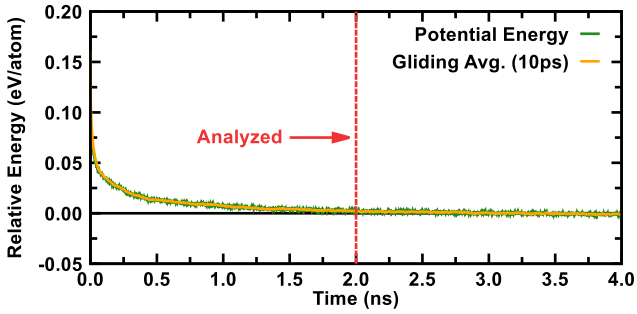

FIG. 8. Equilibration of a LiSi nanoparticle with the ænet–TINKER interface. The amorphous LiSi nanoparticle structure (11 000 atoms, structure shown in Fig. 7) has reached thermal equilibrium after 2 ns-long molecular dynamics (MD) simulations at 500 K. The MD trajectory was used to analyze properties between 2 and 4 ns. Gliding average (10 ps) of the potential energy during MD simulation of the nanoparticle model with composition $Li_{8429}Si_{2571}$ at 500 K.

ænet–LAMMPS interface. Diffusion coefficients were calculated from the mean square displacement of the oxygen positions

$$D = \lim_{t\to\infty} \frac{1}{6}\frac{d}{dt}\left\langle |r(t) - r(0)|^2 \right\rangle. \tag{8}$$

As discussed above, the accurate prediction of dynamic properties, such as diffusion coefficients, is particularly challenging due to their system size dependence resulting from hydrodynamic interactions.[81] For the small system sizes still affordable by direct AIMD, this can lead do deviations between the diffusion coefficient computed at finite system size and the infinite system limit that should be compared to the experiment.[82] A common solution is to apply a correction formula that extrapolates the diffusion coefficient at finite system size, $D(L)$, to the infinite system size limit,

$$D(\infty) = D(L) + \frac{\xi}{6\pi\beta}\frac{1}{\eta L}, \tag{9}$$

with $\beta = (k_B T)^{-1}$, the length of the cubic simulation cell L, the shear viscosity η, and a constant $\xi = 2.837\,297$.[81,82] However, this formula requires a value for the viscosity that is often taken from the experiment and might not correspond to the value one would obtain from simulations with a particular computational method (density-functional, interatomic potential, or force field).

Using the ænet–LAMMPS interface, we performed MD simulations with increasingly larger simulation cells (containing up to 3072 atoms with a total time of 1 ns per system) to estimate the diffusion coefficient of liquid water at experimental conditions. As shown in Fig. 9, the size-independent diffusion coefficient $D(\infty)$ that should be compared to the experiment is obtained directly by extrapolating the diffusivity obtained from finite simulation cells to infinite system size employing the linear relationship between diffusion and inverse box length. Compared to the estimate obtained with the correction formula, Eq. (9), the $D(\infty)$ value from direct extrapolation [line fit to $D(L)$ vs $1/L$] is lower, indicating that the viscosity of revPBE-based ANN water is higher than the experimental one at 300 K.

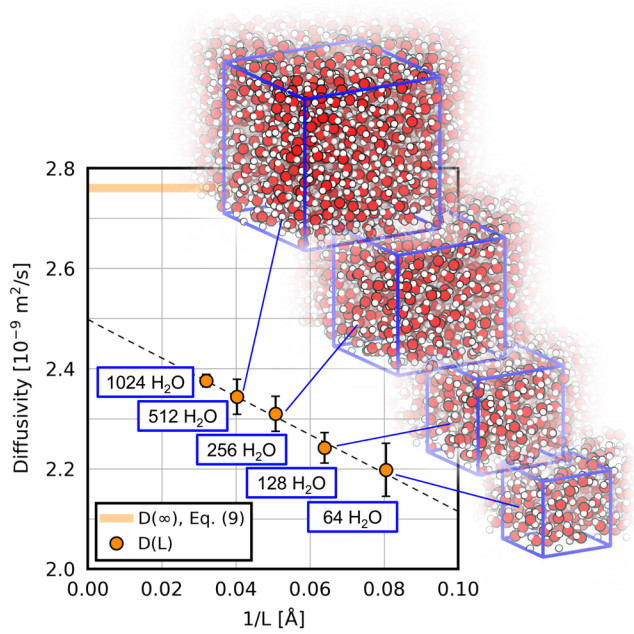

FIG. 9. Calculation of the liquid water diffusivity with the ænet–LAMMPS interface. Liquid water diffusion coefficients as a function of the inverse length of the simulation cell, $1/L$, (shown as the orange circle) obtained from ANN simulations with increasingly larger simulations cells. Error bars represent the standard error of the mean from a set of diffusion coefficients obtained from five independent runs (of length 200 ps each) per system size. Extrapolation to infinite system size employing the linear relationship between diffusion and inverse box length yields an estimate of the diffusion coefficient at experimental conditions $D(\infty) = 2.5 \times 10^{-9}$ m^2/s (dashed line at $1/L = 0.0$). A corrected diffusion coefficient based on the ANN results for 64 H_2O obtained with the correction formula of Eq. (9) using the experimental viscosity value is displayed as the orange shaded line.

V. CONCLUSION

To realize the full potential of machine-learning accelerated atomistic simulations, MLPs need to be made compatible with existing highly optimized simulation software. Here, we reported interfaces of the ænet ANN potential software with the LAMMPS and TINKER simulation packages for accurate and efficient simulations of complex atomic systems. We detailed the installation and usage and discussed implementation differences. Both interfaces facilitate large-scale molecular dynamics simulations on modern parallel computer architectures with single step compute times of (sub-)milliseconds for both systems, an amorphous material and a molecular liquid. The shared-memory parallelized ænet–TINKER interface shows nearly perfect parallel efficiency on a single computer node. The distributed-memory parallelized ænet–LAMMPS interface scales with excellent efficiency (~80%) on multi-node architectures as tested here with up to ~100 computer cores. The ænet–LAMMPS interface further generally shows a higher efficiency than the ænet–TINKER interface owing to the highly optimized neighbor list implementation in LAMMPS. The ænet–TINKER interface currently uses ænet's internal neighbor list that is not optimized for MD simulations, and the efficiency of the

interface for large structures could be further improved by making use of TINKER's own implementation. Both interfaces make it possible to simulate atomic structures with several hundreds of thousands to millions of atoms with an accuracy close to first-principles calculations. Further improvement in the efficiency of such simulations can be expected in the future when support for graphical processor units (GPUs) becomes available.

AUTHORS' CONTRIBUTIONS

M.S.C. and T.M. contributed equally to this work.

ACKNOWLEDGMENTS

M.S.C., T.M., and T.E.M. thank Dr. Ondrej Marsalek for discussions on the simulation of diffusion phenomena in liquid water and Austin Atsango for discussions on the proton analysis. M.S.C. and T.E.M. were supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award No. DE-SC0020203. T.E.M. also acknowledges support from the Camille Dreyfus Teacher–Scholar Awards Program. T.M. acknowledges financial support from the DFG (Grant No. MO 3177/1-1). N.A. acknowledges the support from the Columbia Center for Computational Electrochemistry (CCCE) and is grateful for discussions with the CCCE team, including Dr. Alexander Urban, Dr. Richard A. Friesner, and Dr. David R. Reichman from Columbia University and Dr. James Stevenson, Dr. Leif Jacobson, and Dr. Steven Dajnowicz from Schrödinger, Inc. N.A. acknowledges the Extreme Science and Engineering Discovery Environment (XSEDE), which was supported by the National Science Foundation under Grant No. ACI-1053575 (Allocation No. DMR140005), for supporting the development of the Atomic Energy Network (ænet) package. AENET–LAMMPS and AENET–TINKER scaling tests used HP computing resources from Columbia University's Shared Research Computing Facility project, which was supported by the NIH Research Facility Improvement, Grant No. 1G20RR030893-01, and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract No. C090171, both awarded on 15 April 2010.

DATA AVAILABILITY

The reference datasets of the crystalline/amorphous LiSi and liquid water systems are openly available in the Materials Cloud repository at https://doi.org/10.24435/materialscloud:dx-ct, reference number materialscloud:2020.92. These datasets contain atomic structures and interatomic forces in the XCrySDen[83] structure format (XSF), and total energies are included as additional meta information.

REFERENCES

[1] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications* (Elsevier, 2001).
[2] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, 2017).
[3] S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, "The high-throughput highway to computational materials design," Nat. Mater. **12**, 191–201 (2013).
[4] A. Jain, Y. Shin, and K. A. Persson, "Computational predictions of energy materials using density functional theory," Nat. Rev. Mater. **1**, 15004 (2016).
[5] A. Urban, D.-H. Seo, and G. Ceder, "Computational understanding of Li-ion batteries," npj Comput. Mater. **2**, 16002 (2016).
[6] Z. W. Seh, J. Kibsgaard, C. F. Dickens, I. Chorkendorff, J. K. Nørskov, and T. F. Jaramillo, "Combining theory and experiment in electrocatalysis: Insights into materials design," Science **355**, eaad4998 (2017).
[7] A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, "Structure prediction drives materials discovery," Nat. Rev. Mater. **4**, 331–348 (2019).
[8] W. L. Jorgensen, "The many roles of computation in drug discovery," Science **303**, 1813–1818 (2004).
[9] J. H. Van Drie, "Computer-aided drug design: The next 20 years," J. Comput.-Aided Mol. Des. **21**, 591–601 (2007).
[10] M. Aminpour, C. Montemagno, and J. A. Tuszynski, "An overview of molecular modeling for drug discovery with specific illustrative examples of applications," Molecules **24**, 1693 (2019).
[11] T. Morawietz and N. Artrith, "Machine learning-accelerated quantum mechanics-based atomistic simulations for industrial applications," J. Comput.-Aided Mol. Des. **35**, 557–586 (2021).
[12] C. A. Becker, F. Tavazza, Z. T. Trautt, and R. A. Buarque de Macedo, "Considerations for choosing and using force fields and interatomic potentials in materials science and engineering," Curr. Opin. Solid State Mater. Sci. **17**, 277–283 (2013).
[13] W. L. Jorgensen and J. Tirado-Rives, "Potential energy functions for atomic-level simulations of water and organic and biomolecular systems," Proc. Natl. Acad. Sci. U. S. A. **102**, 6665–6670 (2005).
[14] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).
[15] 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).
[16] N. Artrith, T. Morawietz, and J. Behler, "High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide," Phys. Rev. B **83**, 153101 (2011).
[17] K. V. J. Jose, N. Artrith, and J. Behler, "Construction of high-dimensional neural network potentials using environment-dependent atom pairs," J. Chem. Phys. **136**, 194111 (2012).
[18] V. Botu and R. Ramprasad, "Adaptive machine learning framework to accelerate *ab initio* molecular dynamics," Int. J. Quantum Chem. **115**, 1074–1083 (2015).
[19] N. Artrith and A. Urban, "An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO_2," Comput. Mater. Sci. **114**, 135–150 (2016).
[20] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," Multiscale Model. Simul. **14**, 1153–1173 (2016).
[21] A. Khorshidi and A. A. Peterson, "Amp: A modular approach to machine learning in atomistic simulations," Comput. Phys. Commun. **207**, 310–324 (2016).
[22] 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).
[23] O. T. Unke and M. Meuwly, "A reactive, scalable, and transferable model for molecular energies from a neural network approach based on local information," J. Chem. Phys. **148**, 241708 (2018).
[24] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "SchNet – A deep learning architecture for molecules and materials," J. Chem. Phys. **148**, 241722 (2018).
[25] H. Mori and T. Ozaki, "Neural network atomic potential to investigate the dislocation dynamics in bcc iron," Phys. Rev. Mater. **4**, 040601 (2020).
[26] A. M. Miksch, T. Morawietz, J. Kästner, A. Urban, and N. Artrith, "Strategies for the construction of machine-learning potentials for accurate and efficient atomic-scale simulations," Mach. Learn.: Sci. Technol. **2**, 031001 (2021).
[27] K. Burke, "Perspective on density functional theory," J. Chem. Phys. **136**, 150901 (2012).
[28] N. Artrith, B. Hiller, and J. Behler, "Neural network potentials for metals and oxides - First applications to copper clusters at zinc oxide," Phys. Status Solidi B **250**, 1191–1203 (2013).

[29]N. Artrith, "Machine learning for the modeling of interfaces in energy storage and conversion materials," J. Phys.: Energy **1**, 032002 (2019).
[30]T. Mueller, A. Hernandez, and C. Wang, "Machine learning for interatomic potential models," J. Chem. Phys. **152**, 050902 (2020).
[31]J. W. Ponder and F. M. Richards, "An efficient Newton-like method for molecular mechanics energy minimization of large molecules," J. Comput. Chem. **8**, 1016–1024 (1987).
[32]S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," J. Comput. Phys. **117**, 1–19 (1995).
[33]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).
[34]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).
[35]N. Artrith, A. Urban, and G. Ceder, "Constructing first-principles phase diagrams of amorphous LixSi using machine-learning-assisted sampling with an evolutionary algorithm," J. Chem. Phys. **148**, 241711 (2018).
[36]N. Artrith, A. Urban, Y. Wang, and G. Ceder, "Atomic-scale factors that control the rate capability of nanostructured amorphous Si for high-energy-density batteries," arXiv:1901.09272 [cond-mat, physics:physics] (2019).
[37]T. Morawietz, O. Marsalek, S. R. Pattenaude, L. M. Streacker, D. Ben-Amotz, and T. E. Markland, "The interplay of structure and dynamics in the Raman spectrum of liquid water over the full frequency and temperature range," J. Phys. Chem. Lett. **9**, 851–857 (2018).
[38]T. Morawietz, A. S. Urbina, P. K. Wise, X. Wu, W. Lu, D. Ben-Amotz, and T. E. Markland, "Hiding in the crowd: Spectral signatures of overcoordinated hydrogen-bond environments," J. Phys. Chem. Lett. **10**, 6067–6073 (2019).
[39]G. Cybenko, "Approximation by superpositions of a sigmoidal function," Math. Control, Signals, Syst. **2**, 303–314 (1989).
[40]N. Artrith, A. Urban, and G. Ceder, "Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species," Phys. Rev. B **96**, 014112 (2017).
[41]C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 1. General considerations," IMA J. Appl. Math. **6**, 76–90 (1970).
[42]R. Fletcher, "A new approach to variable metric algorithms," Comput. J. **13**, 317–322 (1970).
[43]D. Goldfarb, "A family of variable-metric methods derived by variational means," Math. Comput. **24**, 23 (1970).
[44]D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," Math. Comput. **24**, 647 (1970).
[45]D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Math. Program. **45**, 503–528 (1989).
[46]Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, 2nd ed., edited by G. Montavon, G. B. Orr, and K.-R. Müller (Springer Berlin Heidelberg, Berlin, Heidelberg, 2012), pp. 9–48.
[47]A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—A Python library for working with atoms," J. Phys.: Condens. Matter **29**, 273002 (2017).
[48]A. P. Thompson, S. J. Plimpton, and W. Mattson, "General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions," J. Chem. Phys. **131**, 154107 (2009).
[49]L. Dagum and R. Menon, "OpenMP: An industry standard API for shared-memory programming," IEEE Comput. Sci. Eng. **5**, 46–55 (1998).
[50]L. Lagardère, L.-H. Jolly, F. Lipparini, F. Aviat, B. Stamm, Z. F. Jing, M. Harger, H. Torabifard, G. A. Cisneros, M. J. Schnieders, N. Gresh, Y. Maday, P. Y. Ren, J. W. Ponder, and J.-P. Piquemal, "TINKER-HP: A massively parallel molecular dynamics package for multiscale simulations of large complex systems with advanced point dipole polarizable force fields," Chem. Sci. **9**, 956–972 (2018).
[51]O. Adjoua, L. Lagardère, L.-H. Jolly, A. Durocher, T. Very, I. Dupays, Z. Wang, T. J. Inizan, F. Célerse, P. Ren, J. W. Ponder, and J.-P. Piquemal, "Tinker-HP: Accelerating molecular dynamics simulations of large complex systems with advanced point dipole polarizable force fields using GPUs and multi-GPUs systems," J. Chem. Theory Comput. **17**, 2034–2053 (2021).
[52]C. J. Wen and R. A. Huggins, "Chemical diffusion in intermediate phases in the lithium-silicon system," J. Solid State Chem. **37**, 271–278 (1981).
[53]T. D. Hatchard and J. R. Dahn, "*In situ* XRD and electrochemical study of the reaction of lithium with amorphous silicon," J. Electrochem. Soc. **151**, A838 (2004).
[54]M. T. McDowell, S. W. Lee, W. D. Nix, and Y. Cui, "25th anniversary article: Understanding the lithiation of silicon and other alloying anodes for lithium-ion batteries," Adv. Mater. **25**, 4966–4985 (2013).
[55]X. Su, Q. Wu, J. Li, X. Xiao, A. Lott, W. Lu, B. W. Sheldon, and J. Wu, "Silicon-based nanomaterials for lithium-ion batteries: A review," Adv. Energy Mater. **4**, 1300882 (2014).
[56]M. R. Zamfir, H. T. Nguyen, E. Moyen, Y. H. Lee, and D. Pribat, "Silicon nanowires for Li-based battery anodes: A review," J. Mater. Chem. A **1**, 9566 (2013).
[57]Y. Domi, H. Usui, A. Ando, K. Nishikawa, and H. Sakaguchi, "Analysis of the Li distribution in Si-based negative electrodes for lithium-ion batteries by soft X-ray emission spectroscopy," ACS Appl. Energy Mater. **3**, 8619–8626 (2020).
[58]Y. Domi, H. Usui, N. Ieuji, K. Nishikawa, and H. Sakaguchi, "Lithiation/delithiation properties of lithium silicide electrodes in ionic-liquid electrolytes," ACS Appl. Mater. Interfaces **13**, 3816–3824 (2021).
[59]S. Cangaz, F. Hippauf, F. S. Reuter, S. Doerfler, T. Abendroth, H. Althues, and S. Kaskel, "Enabling high-energy solid-state batteries with stable anode interphase by the use of columnar silicon anodes," Adv. Energy Mater. **10**, 2001320 (2020).
[60]C. Cao, I. I. Abate, E. Sivonxay, B. Shyam, C. Jia, B. Moritz, T. P. Devereaux, K. A. Persson, H.-G. Steinrück, and M. F. Toney, "Solid electrolyte interphase on native oxide-terminated silicon anodes for Li-ion batteries," Joule **3**, 762–781 (2019).
[61]B. Zhu, G. Liu, G. Lv, Y. Mu, Y. Zhao, Y. Wang, X. Li, P. Yao, Y. Deng, Y. Cui, and J. Zhu, "Minimized lithium trapping by isovalent isomorphism for high initial Coulombic efficiency of silicon anodes," Sci. Adv. **5**, eaax0651 (2019).
[62]B. Wang, X. Li, B. Luo, L. Hao, M. Zhou, X. Zhang, Z. Fan, and L. Zhi, "Approaching the downsizing limit of silicon for surface-controlled lithium storage," Adv. Mater. **27**, 1526–1532 (2015).
[63]N. Ding, J. Xu, Y. X. Yao, G. Wegner, X. Fang, C. H. Chen, and I. Lieberwirth, "Determination of the diffusion coefficient of lithium ions in nano-Si," Solid State Ionics **180**, 222–225 (2009).
[64]J. Xie, N. Imanishi, T. Zhang, A. Hirano, Y. Takeda, and O. Yamamoto, "Li-ion diffusion in amorphous Si films prepared by RF magnetron sputtering: A comparison of using liquid and polymer electrolytes," Mater. Chem. Phys. **120**, 421–425 (2010).
[65]J. Li, X. Xiao, F. Yang, M. W. Verbrugge, and Y.-T. Cheng, "Potentiostatic intermittent titration technique for electrodes governed by diffusion and interfacial reaction," J. Phys. Chem. C **116**, 1472–1478 (2012).
[66]M. Park, X. Zhang, M. Chung, G. B. Less, and A. M. Sastry, "A review of conduction phenomena in Li-ion batteries," J. Power Sources **195**, 7904–7929 (2010).
[67]J. A. Napoli, O. Marsalek, and T. E. Markland, "Decoding the spectroscopic features and time scales of aqueous proton defects," J. Chem. Phys. **148**, 222833 (2018).
[68]G. A. Voth, "Computer simulation of proton solvation and transport in aqueous and biomolecular systems," Acc. Chem. Res. **39**, 143–150 (2006).
[69]D. Marx, A. Chandra, and M. E. Tuckerman, "Aqueous basic solutions: Hydroxide solvation, structural diffusion, and comparison to the hydrated proton," Chem. Rev. **110**, 2174–2216 (2010).
[70]T. E. Markland and M. Ceriotti, "Nuclear quantum effects enter the mainstream," Nat. Rev. Chem. **2**, 0109 (2018).

[71] A. Hassanali, F. Giberti, J. Cuny, T. D. Kühne, and M. Parrinello, "Proton transfer through the water gossamer," Proc. Natl. Acad. Sci. U. S. A. **110**, 13723–13728 (2013).
[72] Y.-L. S. Tse, C. Knight, and G. A. Voth, "An analysis of hydrated proton diffusion in *ab initio* molecular dynamics," J. Chem. Phys. **142**, 014104 (2015).
[73] T. D. Kühne, M. Krack, and M. Parrinello, "Static and dynamical properties of liquid water from first principles by a novel Car-Parrinello-like approach," J. Chem. Theory Comput. **5**, 235–241 (2009).
[74] M. S. Chen, T. Morawietz, T. E. Markland, and N. Artrith, "AENET-LAMMPS and . AENET-TINKER: Interfaces for accurate and efficient molecular dynamics simulations with machine learning potentials," Mater. Cloud Arch. **2020**, 92.
[75] N. Artrith, K. T. Butler, F.-X. Coudert, S. Han, O. Isayev, A. Jain, and A. Walsh, "Best practices in machine learning for chemistry," Nat. Chem. **13**, 505–508 (2021).
[76] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," Phys. Rev. Lett. **77**, 3865–3868 (1996).
[77] Y. Zhang and W. Yang, "Comment on "Generalized gradient approximation made simple"," Phys. Rev. Lett. **80**, 890 (1998).
[78] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, "A consistent and accurate *ab initio* parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu," J. Chem. Phys. **132**, 154104 (2010).
[79] B. Monserrat, J. G. Brandenburg, E. A. Engel, and B. Cheng, "Liquid water contains the building blocks of diverse ice phases," Nat. Commun. **11**, 5757 (2020).
[80] O. Marsalek and T. E. Markland, "Quantum dynamics and spectroscopy of *ab initio* liquid water: The interplay of nuclear and electronic quantum effects," J. Phys. Chem. Lett. **8**, 1545–1551 (2017).
[81] B. Dünweg and K. Kremer, "Molecular dynamics simulation of a polymer chain in solution," J. Chem. Phys. **99**, 6983–6997 (1993).
[82] I.-C. Yeh and G. Hummer, "System-size dependence of diffusion coefficients and viscosities from molecular dynamics simulations with periodic boundary conditions," J. Phys. Chem. B **108**, 15873–15879 (2004).
[83] A. Kokalj, "XCrySDen—A new program for displaying crystalline structures and electron densities," J. Mol. Graphics Model. **17**, 176–179 (1999).
[84] **Code availability**: The ænet source code and its documentation are openly available from the ænet website (http://ann.atomistic.net) or from GitHub (https://github.com/atomisticnet/aenet). The implementation and examples, including ANN potentials, for crystalline/amorphous LiSi and liquid water are also openly available on GitHub at: 1. ænet–LAMMPS code (https://github.com/atomisticnet/aenet-lammps); 2. ænet–TINKER code (https://github.com/atomisticnet/aenet-tinker).
[85] **Tutorials**: Jupyter notebooks with tutorials demonstrating the usage of the TINKER and LAMMPS interfaces for three different example systems can be found at

- ænet–tinker (amorphous LiSi): https://github.com/atomisticnet/aenet-tinker/tree/master/tutorial;
- ænet–lammps (liquid bulk water): https://github.com/atomisticnet/aenet-lammps/tree/master/tutorial;
- ænet–lammps (BCC iron): https://github.com/HidekiMori-CIT/aenet-lammps/tree/master/tutorial.