Topical Review

Deep potentials for materials science

Tongqi Wen[1] ![ORCID], **Linfeng Zhang**[2,3], **Han Wang**[4,5,*], **Weinan E**[3,6,7] and **David J Srolovitz**[1,8,*]

[1] Department of Mechanical Engineering, The University of Hong Kong, Hong Kong, Hong Kong Special Administrative Region of China
[2] DP Technology, Beijing, People's Republic of China
[3] AI for Science Institute, Beijing, People's Republic of China
[4] Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing, People's Republic of China
[5] HEDPS, CAPT, College of Engineering, Peking University, Beijing, People's Republic of China
[6] School of Mathematical Sciences, Peking University, Beijing, People's Republic of China
[7] Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ, United States of America
[8] International Digital Economy Academy (IDEA), Shenzhen, People's Republic of China

E-mail: wang_han@iapcm.ac.cn and srol@hku.hk

CrossMark

Abstract

To fill the gap between accurate (and expensive) *ab initio* calculations and efficient atomistic simulations based on empirical interatomic potentials, a new class of descriptions of atomic interactions has emerged and been widely applied; i.e. machine learning potentials (MLPs). One recently developed type of MLP is the deep potential (DP) method. In this review, we provide an introduction to DP methods in computational materials science. The theory underlying the DP method is presented along with a step-by-step introduction to their development and use. We also review materials applications of DPs in a wide range of materials systems. The DP Library provides a platform for the development of DPs and a database of extant DPs. We discuss the accuracy and efficiency of DPs compared with *ab initio* methods and empirical potentials.

Supplementary material for this article is available online

Keywords: deep potential, atomistic simulation, machine learning potential, neural network

1. Introduction

Atomistic simulations are playing an increasingly important role in materials science and changing how research in this heavily-experimental field is conducted [1]. All atomistic simulations in materials modelling require the input of some

form of potential energy surface (PES) to describe how atoms interact; from which atomic forces are determined. The most accurate way to obtain the PES, within the Born-Oppenheimer approximation [2], is by solving the Schrödinger equation based on a quantum mechanical treatment of the electronic structure for fixed atomic nuclei coordinates [3]. However, in the most-widely applied electronic structure approach, density function theory (DFT) [4], the scaling is normally $\propto N^3$ where N is the number of atoms. This scaling makes DFT very costly in applications to large materials systems (over 1000 atoms) and for long simulation times (e.g. nanoseconds in molecular dynamics, MD). A widely applied solution to this efficiency problem is to develop empirical interatomic

---

* Authors to whom any correspondence should be addressed.

Future perspectives

Machine learning potentials (MLPs) have emerged as a new approach to achieve a balance between accuracy, generality, and efficiency in description of the energetics of atomic ensembles for atomistic simulations. The deep potential (DP) method, one such MLP, is developing rapidly and has been applied to a wide range of materials systems in recent years. While this review focuses on the theory behind the DP method, provides practical guidance for DP development and application, provides a diverse set of examples and a comprehensive discussion of the accuracy and efficiency of DPs, it also outlines a series of promising directions for future development in the short- to medium-terms. These include (1) the development of improved descriptors for better predictability (e.g. three-body embeddings and hybrid descriptors), (2) improved (and more automated) approaches for DP training and specialisation for particular classes of applications, (3) further optimisation for increased computational speed in applications. Additional opportunities lie in the area of DPs for multi-component alloy systems and the expansion of the collection of DPs to include all elements.

potentials [5–10], which describe the relationship between atom positions and system energy by assuming an analytical functional relationship, often based upon physical and chemical insights. Although large-scale, long-time atomistic simulations may readily be performed using empirical interatomic potentials, the accuracy of the atomistic simulations in describing real materials systems is often limited by the assumptions inherent to these empirical descriptions. In this context, one faces the dilemma that quantum mechanics methods are highly accurate but extremely inefficient for such atomistic simulations while empirical interatomic potentials are efficient, but commonly of limited accuracy.

Many approaches have been proposed to strike a balance between accuracy and efficiency in atomistic simulations. One approach to overcoming the low efficiency of widely-used DFT methods was the development of the ONESTEP program [11] in which plane-wave DFT calculations are performed on parallel computers, leading to computational costs that are linear in the number of atoms. DFT calculations have also been implemented on GPU machines [12–15] leading to accelerations by a factor of more than 20 compared to CPU machines [15]. The accuracy of empirical interatomic potentials can be improved, to some extent, by developing analytical functional forms that include many additional parameters to account for additional physical insights. A typical example is the modified embedded atom method (MEAM) potential [10] which extends the embedded atom method (EAM) [9] by considering the angular nature of electron density distributions. However, this tradeoff leads to a decrease in computational speed of the MEAM potentials compared to the simpler EAM potentials; such angular potentials are also more difficult to accelerate than simpler potentials. Nonetheless, the progress from these two approaches are encouraging and have led to many applications. Nevertheless, as materials systems of interest become larger, more complex and demands on predictability more severe, obtaining a better balance between accuracy and efficiency is both an urgent and challenging problem. This challenge requires fundamentally new approaches,

rather than tweaking existing techniques. New insights may be garnered from recent progress in other disciplines.

Machine learning (ML) is well known for its surprising successes in, for example, pattern recognition and, as a result many different ML methods were developed in recent years [16–18]. ML potentials may be viewed as versatile descriptions of the PES parameterised using a flexible ML-based analytical form. The flexibility of ML potential increases the representability (accuracy) compared to empirical interatomic potentials and the analytical form significantly improves the efficiency relative to DFT calculations. In this perspective, ML potentials, fit to DFT results, have the potential for achieving DFT accuracy and empirical interatomic potential efficiency. Developing such ML potentials is a challenge.

Since the pioneering work of Blank *et al* [19] where ML neural network methods for describing the PES were first introduced, a variety of ML potentials have been proposed. Behler and Parrinello (BP) [20] introduced ML neural network potential (NNP) approach in which radial and angular symmetry functions are used as atomic environment descriptors; NNP has found applications in bulk silicon [20], carbon [21], TiO_2 [22] and many other materials [23]. BP-NNPs were categorised into four generations [23] for which there are several recent reviews [23–25]. Schütt *et al* [26, 27] developed SchNet and the SchNetPack package based upon a neural network framework to model the chemical properties and PES of molecular materials. Many other NNPs exist and are popular for different material systems [28–31]. Apart from NNP and SchNet, several other types of ML potentials were introduced; Gaussian approximation potentials [32–35], moment tensor potentials (MTP) [36–38], spectral neighbour analysis potentials [39–41], gradient-domain ML [42, 43], etc [44]. A comprehensive comparison of the major ML potentials in terms of accuracy and efficiency can be found in a recent paper [45].

Deep potentials (DP) [46–49] are of the NNP type that were first published in 2017. This approach has been used extensively for different material systems. The underlying theory has also developed continuously pushing these potentials to increasingly favourable combinations of accuracy and efficiency. Recently, DP has been applied to MD simulations of more than 100 million atoms with *ab initio* accuracy on a state-of-the-art supercomputer [50]. This is a good example of the power of integrating physical modelling, ML and high-performance computing. In this review, we focus on the application of DP in materials science and discuss a vision for future DPs. The paper is organised as follows. In section 2, we review the basic theory underlying the DP method (section 2.1), demonstrate the steps for developing DPs and their application for atomistic simulation (section 2.2), introduce the extant software for DP development (section 2.3) and the DP Library (section 2.4), and discuss how to make DP more practical for atomistic simulations (section 2.5). Section 3 lists many examples of DP applications in materials science, covering elemental bulk systems (section 3.1), multi-element bulk systems (section 3.2), aqueous systems (section 3.3), and other applications (section 3.4). Next,

we discuss the efficiency and accuracy of DPs in practice (section 4.1) and a comparison of the computational speed of DP versus other approaches (section 4.2). We conclude section 5 with an assessment of where DP is going in the near future.

2. Deep potential

2.1. Theory

Consider a system of N atoms, where the total energy of the system is denoted by E, the atomic coordinates by $\mathcal{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_i, \ldots, \boldsymbol{r}_N\}$ and $\{r_{i1}, r_{i2}, r_{i3}\}$ are the three Cartesian components of the vector position of atom i, $\boldsymbol{r}_i$. The potential energy E is a function of all atom coordinates, i.e. $E = E(\mathcal{R})$, and can be accurately determined from first principles calculations. The force on atom i is the negative gradient of the potential energy with respect to its atomic coordinates ($\boldsymbol{F}_i = \{F_{i1}, F_{i2}, F_{i3}\}$):

$$\boldsymbol{F}_i = -\nabla_{\boldsymbol{r}_i} E. \tag{1}$$

Periodic boundary conditions are often applied in atomistic simulations; we denote the cell vectors by a matrix $h = \{h_{\alpha\beta}\}$, where $h_{\alpha\beta}$ is the βth component of the αth cell vector. The virial tensor is then defined by $\boldsymbol{\Xi} = \{\Xi_{\alpha\beta}\}$ with:

$$\Xi_{\alpha\beta} = -\frac{\partial E}{\partial h_{\gamma\alpha}} h_{\gamma\beta}. \tag{2}$$

Training of the ML potential belongs to the category of classical supervised ML. We first obtain the total energy, atomic forces and virial tensors of a number of different system configurations described by the atomic coordinates and use these data as the training labels. Then the ML potential is trained on these labels. We denote a ML potential as $E^w(\mathcal{R})$ where ω is the set of trainable parameters of the model. The force $\boldsymbol{F}^w(\mathcal{R})$ and the virial tensor $\boldsymbol{\Xi}^w(\mathcal{R})$ of the ML potential are derived from E^ω by equations (1) and (2), respectively. The derivatives can be obtained analytically if the potential $E^w(\mathcal{R})$ is differentiable with respect to $\mathcal{R}$. In this case, the training of the ML potential over the training labels can be regarded as a minimisation process of the loss function:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{k \in \mathcal{B}} \left(p_e \mathcal{L}_e^{(k)} + p_f \mathcal{L}_f^{(k)} + p_v \mathcal{L}_v^{(k)} \right), \tag{3}$$

$$\mathcal{L}_e^{(k)} = \frac{1}{N} \left| E(\mathcal{R}^{(k)}) - E^w(\mathcal{R}^{(k)}) \right|^2, \tag{4}$$

$$\mathcal{L}_f^{(k)} = \frac{1}{3N} \sum_{i\alpha} \left| F_{i\alpha}(\mathcal{R}^{(k)}) - F_{i\alpha}^w(\mathcal{R}^{(k)}) \right|^2, \tag{5}$$

$$\mathcal{L}_v^{(k)} = \frac{1}{9N} \sum_{\alpha\beta} \left| \Xi_{\alpha\beta}(\mathcal{R}^{(k)}) - \Xi_{\alpha\beta}^w(\mathcal{R}^{(k)}) \right|^2, \tag{6}$$

where $\mathcal{B}$ is a mini-batch of the training data and $|\mathcal{B}|$ is the number of configurations in this batch. p_e, p_f, and p_v are the prefactors of the energy, forces, and virials in the loss function, which are defined by user or adjusted according to the learning rate in the training process [48].

In the ML potential, the extensibility of the total energy is preserved upon decomposition into atomic energies as follows:

$$E^w = \sum_{i=1}^{N} E_i^w = \sum_{i=1}^{N} E_i^w(\mathcal{R}_i), \tag{7}$$

where E_i^w is the energy we associate with atom i. In the ML potential, as well as many classical interatomic potentials (such as EAM potentials), a common assumption is that the energy of atom i depends only on its atomic coordinate and local environment. Consider r_c as a pre-defined cutoff radius (the choice of r_c depends on the atomic interaction characteristics of the material), the local environment of atom i is the collection of the relative positions of all neighbouring atoms whose distance to atom i is smaller than r_c. This collection of near neighbours is denoted as $\mathcal{N}_{r_c}(i) = \{j | r_{ij} = |\boldsymbol{r}_{ij}| \leqslant r_c\}$. We define the cardinality of the set $\mathcal{N}_{r_c}(i)$ as N_i and use the environment matrix $\mathcal{R}_i$ with N_i rows and 3 columns to represent the local environment of atom i, where row j of $\mathcal{R}_i$ is the relative position of atoms i and j:

$$(\mathcal{R}_i)_j = (\boldsymbol{r}_{ij}). \tag{8}$$

The local (neighbourhood) dependence of the atomic energy is an assumption. There are, however, non-local (long-range) interactions, arising mainly from Coulombic interactions within the electron density distribution. For metallic systems, the local dependence assumption is reasonable as a result of shielding effects. For homogeneous materials, the long-range interactions attenuates rapidly with increasing atomic separation such that a sufficiently large r_c can always satisfy the local dependence assumption. For materials where long-range interactions dominate, these must be explicitly considered in the model construction. Although there are research reports which introduce long-range interaction in the model construction [28, 30, 51–59], there is still no widely accepted method which can handle this interaction appropriately. In the DP method, we focus on the most common situation, in which E_i^w is well-described by the local interaction assumption. Also note, that when the system is extended (rather than molecules), we explicitly involve periodic boundary conditions in the usual way; for more details, see, e.g. [60].

The total energy of the material system is invariant under a set of symmetry operations that may include translations, rotations, and permutations:

$$E(\mathcal{R}) = E(U\mathcal{R}), \tag{9}$$

where U is the symmetry operation on the atomic coordinates. For the ML potential model, the most common approach is to insure that:

$$E_i^w(\mathcal{R}_i) = E_i^w(U\mathcal{R}_i). \tag{10}$$

To achieve this energy invariance, descriptors of the atomic environment are introduced that are invariant upon these symmetry operations performed on the atomic coordinates:

$$\mathcal{D}(\mathcal{R}_i) = \mathcal{D}(U\mathcal{R}_i). \tag{11}$$

The atomic energy can thus be written as:

$$E_i^w(\mathcal{R}_i) = \mathcal{F}(\mathcal{D}(\mathcal{R}_i)), \tag{12}$$

where $\mathcal{F}$ is the fitting method employed for the deep neural network (DNN) in the DP construction. The smoothness of the formalism is also important for calculating atomic forces and virial tensors from derivatives. This implies that the descriptors should be of sufficiently high resolution so as to distinguish intrinsically different local atomic environments.

Different types of ML potentials employ different descriptors. Some examples are the symmetry functions from BP [20], the Smooth Overlap of Atomic Positions from Bartok *et al* [34], the SchNet 'descriptors' from Schütt *et al* [26, 27], and the moment tensor from Shapeev [36]. In the following, we introduce the DNN and then focus on the construction and physical meanings of the descriptors employed in DPs.

2.1.1. Deep neural network.   Deep neural networks (DNN) and kernel methods are the two dominant methods employed in constructing ML potentials. Kernel-based methods usually construct descriptors explicitly, while DNN methods can be used to 'learn' descriptors. These are both broad areas, so we refer interested readers to a standard reference [18]. Due to the robust fitting ability of DNN for high-dimensional and nonlinear properties, the DP uses DNNs both as a fitting method for $\mathcal{F}$ and also to construct descriptors of local atomic environments (as described below, section 2.1.3)

The DNN is used as the fitting net $\mathcal{F}$ that maps the descriptors $\mathcal{D}_i$ to local atomic energies:

$$\mathcal{F}(\mathcal{D}_i) = \mathcal{L}_O \circ \mathcal{L}_P \circ \cdots \circ \mathcal{L}_k \circ \cdots \circ \mathcal{L}_1(\mathcal{D}_i), \tag{13}$$

where $\mathcal{L}_k$ is the mapping from layer $k-1$ to k (P hidden layers in total) in the neural network. Each mapping is composed of a linear and a non-linear transformation as follows:

$$\boldsymbol{d}_k = \mathcal{L}_k(\boldsymbol{d}_{k-1}) = \phi(\mathcal{W}_k \cdot \boldsymbol{d}_{k-1} + \boldsymbol{b}_k), \tag{14}$$

where $\boldsymbol{d}_k \in \mathbb{R}^{n_k}$ represents the state of the neurons in layer k, and n_k is the number of neurons. The weight matrix $\mathcal{W}_k \in \mathbb{R}^{n_k \times n_{k-1}}$ and bias vector $\boldsymbol{b}_k \in \mathbb{R}^{n_k}$ are trainable parameters in the neuron network. ϕ is a non-linear activation function, e.g. a hyperbolic tangent. $\mathcal{L}_O$ is the output mapping from the last hidden layer to the output layer and is usually a linear function with trainable parameters. From equation (14), we see that the smoothness of thee DNN is determined by the activation function. The addition of 'skip' connections can improve the accuracy [61]:

$$\boldsymbol{d}_k = \boldsymbol{d}_{k-1} + \phi(\mathcal{W}_k \cdot \boldsymbol{d}_{k-1} + \boldsymbol{b}_k). \tag{15}$$

Quantification of the representability of a DNN is an active area of research. Barron [62, 63] proved that a neural network with only 1 hidden layer ($P = 1$) and an arbitrary number of neurons can approximate a class of functions with arbitrary precision. Many researchers have subsequently investigated the approximation ability of DNNs with $P > 1$ [64–67] and explained why the DNN is more successful (and widely used) than wide neural networks (many neurons in one hidden layer). E *et al* [68, 69] explained and demonstrated why DNNs are suitable for high dimensional problems.

2.1.2. Descriptors for the non-smooth DP.   There are two classes of descriptors for DPs, namely non-smooth [48, 49] and smooth mappings of the atomic coordinates [70]. The concept behind the non-smooth descriptor is to set up a local coordinate frame for every atom and its neighbours inside the cutoff distance r_c and then sort neighbour atoms according to the distance to the centre atom. This helps preserve the translational, rotational, and permutational symmetries of the atomic environment. Construction of the local frame is as follows. First, two atoms are chosen from the neighbours of atom i: $a(i) \in \mathcal{N}_{r_c}(i)$, $b(i) \in \mathcal{N}_{r_c}(i)$ such that atoms i, $a(i)$, and $b(i)$ are not colinear. We define the rotational matrix as follows and the elements in each column are the basis vectors in the local coordinate system:

$$\mathcal{R}(\boldsymbol{r}_{ia(i)}, \boldsymbol{r}_{ib(i)}) = \begin{pmatrix} \boldsymbol{e}(\boldsymbol{r}_{ia(i)}) \\ \boldsymbol{e}\left[\boldsymbol{r}_{ib(i)} - \frac{\boldsymbol{r}_{ia(i)} \cdot \boldsymbol{r}_{ib(i)}}{\boldsymbol{r}_{ia(i)} \cdot \boldsymbol{r}_{ia(i)}} \boldsymbol{r}_{ia(i)}\right] \\ \boldsymbol{e}(\boldsymbol{r}_{ia(i)} \times \boldsymbol{r}_{ib(i)}) \end{pmatrix}^T, \tag{16}$$

where $\boldsymbol{e}(\boldsymbol{r}) = \boldsymbol{r}/|\boldsymbol{r}|$ is the normalised vector of $\boldsymbol{r}$. Then the local coordinates $\boldsymbol{r'}_{ij} = (x'_{ij}, y'_{ij}, z'_{ij})$ can be transformed from the global coordinates $\boldsymbol{r}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ according to:

$$(x'_{ij}, y'_{ij}, z'_{ij}) = (x_{ij}, y_{ij}, z_{ij}) \cdot \mathcal{R}(\boldsymbol{r}_{ia(i)}, \boldsymbol{r}_{ib(i)}). \tag{17}$$

The local coordinates can be used directly to construct the descriptor by using radial and/or angular information:

$$\{D_{ij}\} = \begin{cases} \left\{ \frac{1}{r_{ij}}, \frac{x'_{ij}}{r_{ij}}, \frac{y'_{ij}}{r_{ij}}, \frac{z'_{ij}}{r_{ij}} \right\}^{\text{sort}}_{j \in \mathcal{N}_{r_c}(i)}, & \text{full info.} \\ \\ \left\{ \frac{1}{r_{ij}} \right\}^{\text{sort}}_{j \in \mathcal{N}_{r_c}(i)}, & \text{radial-only} \end{cases} \tag{18}$$

where the superscript 'sort' implies sorting the atoms according to their inverse distance to atom i (i.e. $1/r_{ij}$). When a neighbour atom is far away from the centre atom, the radial-only information can be considered rather than both angular and radial information.

A schematic of the descriptor for the non-smooth DP is shown for a water molecule in figure 1. The red and blue spheres denote oxygen and hydrogen atoms, $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ is the local frame of atom i and its neighbour atom j, and $(x'_{ij}, y'_{ij}, z'_{ij})$ are the Cartesian components of $\boldsymbol{r}_{ij}$ in this local frame. The descriptor $\mathcal{D}_{ij}$ may contain only radial (distance) information
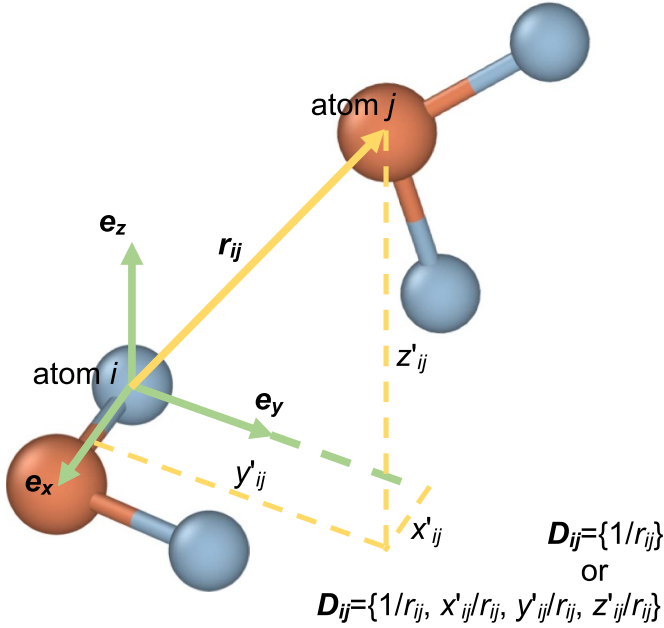
Figure 1. Schematic of the non-smooth DP descriptor for a water molecule. The red and blue spheres denote oxygen and hydrogen atoms, respectively.

Figure 2. Workflow for constructing the smooth DP descriptors.

and/or angles. In the water molecule example, neighbours of atom i are first sorted according to their chemical species (oxygen first, then hydrogen). Then within the same species, the inverse distances to atom i are used to sort the atoms. Finally, $\mathcal{D}_{ij}$ is applied to the sorted input data for atom i.

The advantage of non-smooth descriptor is that all neighbour information is preserved. However, due to uncertainty in the choice of neighbour atoms $a(i)$ and $b(i)$, the descriptor is non-smooth. In practice, $a(i)$ is picked as the nearest neighbour and $b(i)$ as the second nearest neighbour. Continuous change in atom positions can thus result in a discontinuous change of the atom number, the local frame, and the local coordinates. In addition, the sorting operation for the other neighbours introduces additional discontinuities in the descriptor and its derivatives (see equation (18)).

2.1.3. Descriptors for the smooth DP. The workflow for the DP-Smooth Edition (DP-SE) model [70] descriptor is shown in figure 2. There are three major steps to construct the sub-network for the atomic energy using this descriptor. First, the environment matrix for atom i, equation (8) is reformed as:

$$(\mathcal{R}_i)_j = s(r_{ij}) \times \left(\frac{x_{ij}}{r_{ij}}, \frac{y_{ij}}{r_{ij}}, \frac{z_{ij}}{r_{ij}}\right), \quad (19)$$

where $s(r_{ij})$ is a continuous and differentiable function:

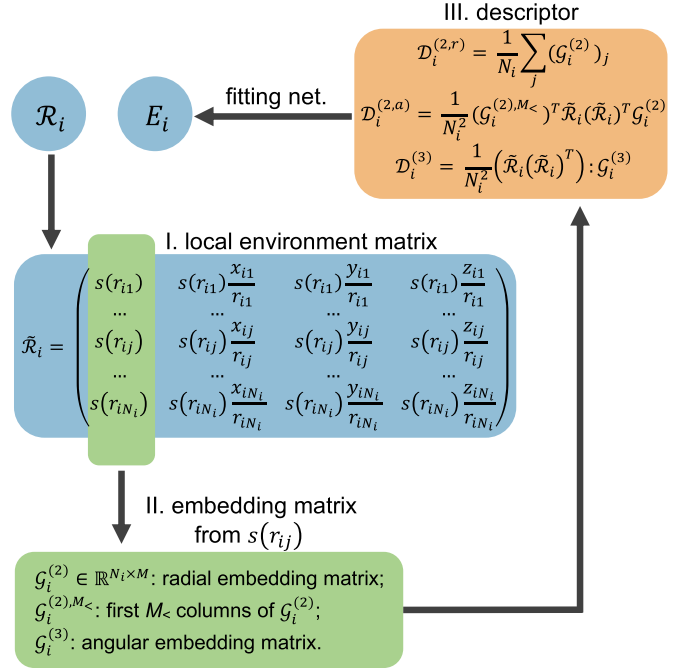$$s(r_{ij}) = \begin{cases} \frac{1}{r_{ij}}, & r_{ij} < r_{cs} \\ \frac{1}{r_{ij}} f_c(r_{ij}), & r_{cs} < r_{ij} < r_c, \\ 0, & r_{ij} > r_c, \end{cases} \quad (20)$$

and where $f_c(r_{ij})$ is a smooth switching function decaying from 1 at r_{cs} to 0 at r_c. DP uses a 5th order polynomial for this purpose:

$$f_c(r) = u^3(-6u^2 + 15\,u - 10) + 1, \; u = \frac{r - r_{cs}}{r_c - r_{cs}}. \quad (21)$$

Using this f_c yields a second order differentiable $s(r)$ with continuous derivatives at r_{cs} and r_c. The augmented matrix $\tilde{\mathcal{R}}_i \in \mathbb{R}^{N_i \times 4}$ (N_i neighbours of atom i) is then constructed from $\mathcal{R}_i$ by adding a column and each row is defined as:

$$(\tilde{\mathcal{R}}_i)_j = s(r_{ij}) \times \left(1, \frac{x_{ij}}{r_{ij}}, \frac{y_{ij}}{r_{ij}}, \frac{z_{ij}}{r_{ij}}\right). \quad (22)$$

Next, the two-body embedding matrix $\mathcal{G}_i^{(2)} \in \mathbb{R}^{N_i \times M}$ is constructed from the first column in $\tilde{\mathcal{R}}_i$ and each row is defined as:

$$\left(\mathcal{G}_i^{(2)}\right)_j = \left(G_1^{(2)}(s(r_{ij}), Z_j), ..., G_M^{(2)}(s(r_{ij}), Z_j)\right), \quad (23)$$

where the vector $\left(G_1^{(2)}, ..., G_M^{(2)}\right)$ is a DNN mapping from a scalar input $s(r_{ij})$ and Z_j is the input type of neighbour atom j. The three-body embedding tensor is constructed as:

$$\left(\mathcal{G}_i^{(3)}\right)_{j,k} = \left(G_1^{(3)}((\theta_i)_{jk}, Z_j, Z_k), ..., G_M^{(3)}((\theta_i)_{jk}, Z_j, Z_k)\right), \quad (24)$$

where the angular information with atom i at the centre and atoms j and k as neighbours is the inner product of rows j and k of $\mathcal{R}_i$:

$$(\theta_i)_{jk} = (\mathcal{R}_i)_j \cdot (\mathcal{R}_i)_k, \tag{25}$$

and the vector $(G_1^{(3)}, \ldots, G_M^{(3)})$ is represented by a DNN.

Finally, from equations (23) and (24), smooth descriptors can be constructed from the two-body and three-body embedding matrix and tensor:

$$\mathcal{D}_i^{(2,r)} = \frac{1}{N_i} \sum_j \left(\mathcal{G}_i^{(2)} \right)_j, \tag{26}$$

$$\mathcal{D}_i^{(2,a)} = \frac{1}{N_i^2} \left(\mathcal{G}_i^{(2),M_<} \right)^T \tilde{\mathcal{R}}_i \left(\tilde{\mathcal{R}}_i \right)^T \mathcal{G}_i^{(2)}, \tag{27}$$

$$\mathcal{D}_i^{(3)} = \frac{1}{N_i^2} \left(\tilde{\mathcal{R}}_i (\tilde{\mathcal{R}}_i)^T \right) : \mathcal{G}_i^{(3)}, \tag{28}$$

where $\mathcal{G}_i^{(2),M_<}$ are the first $M_<$ columns of $\mathcal{G}_i^{(2)}$. The two-body embedding $\mathcal{D}^{(2,r)}$ only depends on the radial distance between neighbouring atoms. While the two-body embedding $\mathcal{D}^{(2,a)}$ depends on the coordinates of the neighbour atoms, the embedding term (equation (23)) only relies on the atom separation. Three-body embedding $\mathcal{D}_i^{(3)}$ considers the angle between neighbour atoms in the embedding term (equation (24)). In this case, from the point of view of descriptor accuracy and resolution, $\mathcal{D}_i^{(3)} > \mathcal{D}_i^{(2,a)} > \mathcal{D}_i^{(2,r)}$. Training the descriptor with higher resolution is more difficult. In practice, one, two or all of the descriptors above can be used in a hybrid format. For example, one may define the descriptor as $\mathcal{D} = \left(\mathcal{D}_i^{(3)}, \mathcal{D}_i^{(2,a)} \right)$, where the $\mathcal{D}_i^{(3)}$ with a smaller cut-off describes near-neighbour configurations, while $\mathcal{D}_i^{(2,a)}$ with a larger cut-off describes the environment further away.

2.2. Developing and applying DPs

After illustrating the descriptors and DNN fitting in the DP method, we now outline the main steps we apply to develop a DP: preparing training datasets, the training process, and model validation. We also explain how to apply DPs in atomistic simulations.

2.2.1. Preparing training datasets. The preparation of the training data for a DP has two parts: (1) providing atomic configurations, (i.e. the coordinates of atoms and the cell shape tensor) and (2) labelling (i.e. calculating the energy, atomic forces and virial tensor for the configuration). Labelling is done through a DFT calculation; e.g. well-established DFT packages such as the Vienna *Ab initio* Simulation Package (VASP) [71, 72], Quantum ESPRESSO (QE) [73], and Atomic-orbital Based Ab-initio Computation at UStc (ABACUS) [74]. Since DPs are trained using a DFT training set, the accuracy of a DP will never exceed that of its training

data. The errors in DFT calculations have two main sources. The first is the error introduced by the approximate form of the exchange-correlation functional. This type of error may be reduced by moving up the Jacob's ladder [75] of increasingly accurate exchange correlation functionals (usually at higher computational cost). It is also possible to use the high order post-Hartree Fock methods like the Møller-Plesset perturbation [76], coupled cluster [77], and configuration interaction [78] for labelling. The other source of error in DFT calculations is numerical, i.e. error introduced by the numerical discretisation of wave functions in real and k spaces and convergence. This type of error can be systematically controlled with the use of more complete basis sets (increasing the energy cutoff in plane-wave approximations), reducing k-space mesh spacing and using stricter stop criteria for self-consistent field iterations. Generally speaking, increasing label quality implies larger computational demands. One usually seeks for a balance between quality and cost.

The training dataset is another critical issue for generating a DP; here, the two main issues are (1) completeness and (2) compactness. By completeness, we mean the training datasets need to sample the relevant configuration space as completely as possible. Increasing the diversity in the members of the training dataset helps increase the transferability of the DP. By compactness, we mean the training data should be the minimal subset of the sampled configurations from which a model with uniform accuracy on the sampled configurations is trained. This is important for minimising DFT computational time. Different approaches have been used to sample the configuration space including MD simulations, genetic algorithms [79, 80], enhanced sampling methods [81], active learning [82] and concurrent learning schemes [83, 84]. Among these, the concurrent learning scheme we refer to as the deep potential generator (DP-GEN) is found to be particularly effective at generating training datasets that satisfy both the completeness and compactness conditions (see section 2.3.3).

2.2.2. Training process. We employ the DeePMD-kit package [49] to train a DP. A python package, dpdata (web link 1 in section S2 of the supplementary material (SM) available online at stacks.iop.org/MF/1/022601/mmedia) is used together with the DeePMD-kit to transform the DFT labels from DFT software to the data format accepted by the DeePMD-kit package. DeePMD-kit and dpdata package details are discussed in section 2.3.

There are several important issues in the training process. First, the DP should be trained such as to avoid under-fitting and over-fitting. Under-fitting implies that the DP performs poorly for both the training and validating datasets. Increasing the number of fitting parameters (wider or deeper DNN) or adjusting the ML algorithms can help avoid under-fitting. Under-fitting is easily detected in the training process. On the other hand, over-fitting suggests that a DP is very good at reproducing the training data, but poor at prediction. This may be solved by increasing the training dataset size, decreasing
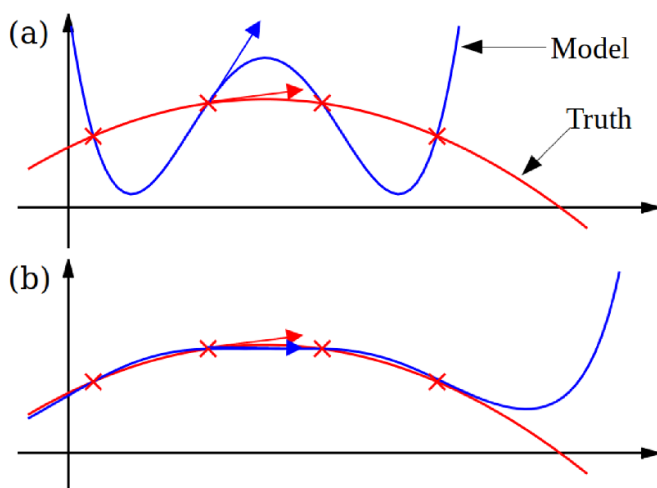
Figure 3. Schematic depiction of an over-fit ML model (a) and a properly fit ML model (b). The ground truth and ML model are denoted by the red and blue lines. The training data are denoted by red ×. The over-fit values are close to the ground truth for the training data, while the gradients deviate from the ground truth (red and blue arrows). Both the values and gradients of the properly fit model are close to the ground truth for the training data.

the number of fitting parameters, or using the force and virial as labels (see below). Over-fitting is more difficult to identify; hence, comprehensive testing is necessary.

Second, while training may involve the energy, force and virial labels (normally from DFT calculations), not all of these are necessary. However, it is strongly recommended to use force labels because (a) force has more information: $3N$ vs. 1 compared to the energy label and (b) training with gradient information helps avoid over-fitting. The schematic plot in figure 3 demonstrates that the gradients of an over-fit model deviate from the ground truth, thus training with forces (gradients of energy with respect to coordinates) and virials (gradients of energy with respect to the cell tensor) as labels helps avoid over-fitting.

Third, there are many tunable hyper-parameters in the training process in the DeePMD-kit, including neural network size, learning rate, prefactors in the loss functions (equation (3)). In practice, we observe that the quality of the DP is not very sensitive to these hyper-parameters and the default settings in the DeePMD-kit routinely provide reasonable accuracy. In some cases, high accuracy for energy is required and 'fine tuning' with larger energy prefactors helps. A more detailed discussion of the hyper-parameters in DeePMD-kit is in section 2.3.2.

2.2.3. Model validation. After the training process is complete, it is advisable to validate the obtained DP to decide whether additional training datasets are required prior to atomistic simulation applications. There are two main approaches to test the DP performance. (1) We can construct a small set of datasets which are unrelated to the training datasets. The DP is used to predict the total energy, atomic forces, and virial

tensors of the constructed datasets and the results are compared with DFT results for the same atomic configuration. Experience shows that the root mean square error (RMSE) for energy and forces for a good DP should be smaller than ~ 10 meV atom^{-1} and ~ 100 meV Å^{-1}. It is not uncommon to achieve ~ 1 meV atom^{-1} and <50 meV Å^{-1} errors for energy and forces. (2) The DP can be applied to calculate properties of interest to be compared with DFT. For example, elastic constants are important properties for structural applications that can be obtained both from DFT and DP MD. Application of DPs in atomistic simulations are discussed below and the Autotest package in DP-GEN conveniently calculates several different properties (see section 2.3.3).

2.2.4. Model inference. Model inference (a commonly used term in the ML community), is the process of providing live data to the ML model to obtain an output. In the context of ML models for the potential energy surface, the inference means taking the configuration (atom coordinates and cell tensor) as input to calculate the energy, force and virial tensor. With the interfaces provided by the DeePMD-kit package, one can easily make inferences about the DP in a Python or C++ programming environment. This makes it possible to use DPs in various molecular simulation tasks such as MD, Monte Carlo, geometric optimisation, by interfacing DeePMD-kit with molecular simulation packages, such as the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [85], the atomic simulation environment (ASE) [86], i-PI [87], and GROMACS [88].

2.3. Software for DP development

There are three main packages for DP development: (1) dpdata for converting the output of DFT software to the data format accepted by DeePMD-kit; (2) DeePMD-kit for DP training and inference; (3) DP-GEN for efficient sampling and labelling of the training data. The DP-GEN package integrates with DeePMD-kit and DFT calculation packages to automatically generate and test DPs.

2.3.1. dpdata. dpdata is a python package which converts and manipulates training data in different DFT package formats to the compressed format used by DeePMD-kit. A typical dpdata workflow is as follows. (1) Load data from data files. Data files can be written in one of the following package formats: VASP [71, 72], LAMMPS [85], Gaussian [89], SIESTA [90], CP2K [91], QE [73], FHI-aims [92], QUIP (web link 2 in section S2 of the SM), PWmat [14, 15], AMBER [93], GROMACS [88], and ABACUS [74]. (2) Data may be manipulated through operations including replication of atom configurations in a supercell, perturbation of the cell vector and atom positions, and replacement of a number of one type of atoms with others. (3) Output data in a format of one of the aforementioned software. Here we give a simple example of transforming the OUTCAR file from VASP into a training dataset that DeePMD-kit can read:

```
1 import dpdata
2 d_outcar = dpdata.LabeledSystem("OUTCAR")
3 d_outcar.to("deepmd/npy", "dpmd_raw")
```

After execution of these commands, a directory named `dpmd_raw` is created that stores the `numpy` compressed data appropriate for DeePMD-kit use. The dpdata package source code and manual is available at web link 1 in section S2 of the SM.

2.3.2. DeePMD-kit.

The DeePMD-kit package [49] (first made publicly in November 2018) is available on GitHub (web link 3 in section S2 of the SM) under the GNU Lesser General Public License (LGPL). DeePMD-kit has continued to evolve since first becoming available. The DeePMD-kit package interfaces with TensorFlow [94] to make the training and inference codes more efficient and automated. The training is integrated and made available via a command line interface. Model inference is provided through C++ and Python interfaces, which accept atom positions and the cell tensor and return the energy, force and virial for this configuration. These interfaces may be used by MD and molecular simulation packages written in C, C++ or Python, as discussed in section 2.2.4. The DeePMD-kit supports GPU accelerated training and inference. When interfaced with the LAMMPS MD package, parallel and distributed computations accelerated by GPUs are available. (On Summit, one of the most powerful supercomputers in the world, DeePMD-kit has pushed the limit of MD with *ab initio* accuracy to 100 million atoms and achieved a peak performance of 91 PetaFLOPS in double precision (45.5% of the theoretical peak) [50]). The DeePMD-kit package is now well-developed, been installed over 30 000 times (GitHub) and received over 650 GitHub stars.

DeePMD-kit can now be easily installed via off-line packages and package managers such as Conda and Docker. A detailed introduction can be found at web link 4 in section S2 of the SM. The brief release history and key milestones of DeePMD-kit can be found in the SM.

2.3.2.1. Training and testing of DPs.

After installation of DeePMD-kit package, we follow the path in section 2.2 to develop a DP. First, dpdata is used to generate the training datasets to be read by DeePMD-kit package (see section 2.3.1). Then, DP training is started with the following command:

```
1 dp train input.json
```

where `input.json` is the input script controlling the training process. If the training program stops, it can be restarted with

```
1 dp train--restart model.ckpt input.json
```

where `model.ckpt` is the checkpoint file storing the existing model and training status. A detailed explanation of the output files and training (validation) errors can be accessed at web link 5 in section S2 of the SM.

After the training process, the architecture and the parameters of the DP can be abstracted from the checkpoint file and saved in the DP file in the Google's protobuf format

```
1 dp freeze -o graph.pb
```

where `graph.pb` is the DP file. The DP file can be used for testing and model inference via the Python and C++ interfaces. For example, the DP can be used to give energy, forces, and virial errors on a designated datasets:

```
1 dp test -m graph.pb -s /path/to/system -n 30
```

where the tested model follows `-m` tag, `-s` gives the path to the tested system, and `-n` is the number of frames in the tested system.

2.3.2.2. Fine-tuning a DP.

The adjustment of hyper-parameters in ML is never a trivial issue. Fortunately, the efficiency of training a DP is not very sensitive to hyper-parameters and the default settings in DeePMD-kit usually yield a DP of reasonable accuracy. Detailed documentation of all available training parameters can be accessed at web link 6 in section S2 of the SM. On occasion, the hyper-parameters settings may not be satisfactory; e.g. if one wants a DP with higher energy and virial accuracy without a large penalty in force accuracy, fine-tuning of the hyper-parameters may be necessary.

To fine-tune a DP, one may adjust the prefactors of the energy (p_e), forces (p_f), and virials (p_v) in the loss function in equation (3) and the learning rate. DeePMD-kit implements an exponentially decaying learning rate and dynamically adjusts prefactors:

$$r_l(t) = r_l^0 e^{t/t_d}, \tag{29}$$

$$p(t) = p^{\text{limit}} \left[1 - \frac{r_l(t)}{r_l^0} \right] + p^{\text{start}} \left[\frac{r_l(t)}{r_l^0} \right], \tag{30}$$

where t and t_d denote the training step and the typical time-scale of the learning rate decay, respectively. In equation (30), $r_l(t)$ and r_l^0 are the learning rate at training step t and the learning rate at the beginning, respectively. DeePMD-kit lets the user set the start learning rate r_l^0, number of training steps T_{train} and the expected learning rate at the end of training $r_l(T_{\text{train}})$, then determines the parameter t_d automatically. The prefactors gradually change from p^{start} to p^{limit} in the training process. The default settings of the start and stop learning rates are 10^{-3} and 10^{-8}, respectively, and those of the prefactors are $p_e^{\text{start}} = 0.02$, $p_e^{\text{limit}} = 1$, $p_f^{\text{start}} = 1000$, $p_f^{\text{limit}} = 1$, $p_v^{\text{start}} = 0.02$, $p_v^{\text{limit}} = 1$. At the beginning of training, the force prefactor dominates the loss function, and the DP is mainly trained to minimise the force error. At the end of training the force prefactor decreases, while the energy and virial prefactors increase, so the training is in more balanced on minimising the energy, force and virial errors. It was argued that this dynamic prefactor scheme is usually more efficient than constant prefactors [48]. The model

trained with this setting is usually very accurate in predicting atomic forces, but is not always satisfactory in predicting the energy and virial tensor.

Model fine-tuning restarts training. In the fine-tuning, the model parameter are initialised from the DP trained with the default setting, the start learning rate is decreased to 10^{-4} while the stop learning rate is kept, and the loss prefactors set to $p_e^{\text{start}} = 10$, $p_e^{\text{limit}} = 100$, $p_f^{\text{start}} = 1$, $p_f^{\text{limit}} = 1$, $p_v^{\text{start}} = 10$, $p_v^{\text{limit}} = 100$. The idea behind this setting is to inherit model parameters from a DP with high force accuracy, and to train the model with significantly larger energy and virial prefactors to focus the training on the energy and virial accuracy. The learning rate is decreased to slow the forgetting of the model parameters of the original model and to preserve the force accuracy. If the numerical error in the labels is low, then the fine-tuning usually leads to a significant improvement on the energy and virial accuracy with little decrement to the force accuracy [95].

Another hyper-parameter critical to the accuracy of the DP is the number of training steps. Different settings may be employed for different DP purposes. In the concurrent learning procedure, (for example DP-GEN see section 2.3.3), 0.4–2 million training steps should suffice. If the DP is intended for high-accuracy atomistic simulations, over 8 million training steps are often employed for from-scratch training or fine-tuning.

Finally, the `atom_ener` parameter may be used to specify the energy of an atom in the vacuum for each atom. For metals and alloys, the cohesive energy calculated from DFT does not always match experiment. For example, for Ti, the cohesive energy calculated from DFT is 5.34 eV atom^{-1} [96], while experiment gives 4.85 eV atom^{-1} [97]. Because DP is fitted to DFT data, this command helps to correct the DFT 'errors' for the isolated atom. Accurate cohesive energy relative to experiment is critical for DP in some applications, such as the fracture behaviour of metals and alloys.

2.3.2.3. Interfacing with third-party packages. After a DP is trained and tested by DeePMD-kit, application requires linking with other software. DeePMD-kit provides Python and C++ interfaces for model inference, which is helpful for calculating energy, atomic forces, and virial tensors with input atomic coordinates and the cell tensor. DeePMD-kit also interfaces with ASE [86], LAMMPS [85], i-PI [87], and GROMACS [88] for DP-based atomistic simulations. The detailed use of DP for these can be found at web link 7 in section S2 of the SM.

Here, we explicitly discuss using DPs in LAMMPS. The installation package for DeePMD-kit already incorporates the latest stable version of LAMMPS. Running MD or other molecular simulations with DP in LAMMPS is very simple. The user only needs to add two lines of commands specifying the interatomic interaction with the LAMMPS command `pair_style` and `pair_coeff` to the LAMMPS input script:

```
1 pair_style    deepmd graph.pb
2 pair_coeff    * *
```

where `graph.pb` is the DP file. The user need not change any other part of the input script to conduct DP-based molecular simulations in LAMMPS.

2.3.3. Deep potential generator. Deep Potential GENerator (DP-GEN) is a software package that implements the concurrent learning framework for generating high quality DPs.

2.3.3.1. Basic concepts and framework. As discussed in section 2.2.1, generating a complete and compact dataset is critical for training high quality DPs. A straightforward way to generate data is to run finite-temperature *ab initio* MD (AIMD) simulations and use the configurations along the trajectories with labels (energy, forces, and virial tensors) as training datasets. However, this method of data generation is not particularly efficient. First, AIMD simulations are computationally expensive since labels are calculated at each time step by DFT. Second, computationally affordable AIMD simulations are typically very short and do not effectively explore phase space. Many important phenomena, such as phase transformations are difficult to be observed on AIMD time scales. Also configurations generated at successive MD time steps are extremely similar to one the other.

The DP-GEN concurrent learning scheme is designed to overcome these difficulties by generating a much more complete and compact training dataset for DP training. The DP-GEN is an iterative scheme. In each iteration it performs exploration, labelling and training. The training dataset is gradually enriched, and the quality of the DP improves on each iteration. The DP-GEN scheme is deemed converged when all relevant configurations are explored and the DP is uniformly accurate on the explored configurations. The three steps in each DP-GEN iteration are as follows.

(a) Exploration. We start with an ensemble of DPs. One of these DPs is used in the exploration step (figure 4(b)) to efficiently sample the relevant configuration space. The sampler typically performs several DP MD simulations (different initial configurations) for a set of thermodynamics conditions. In principle, sampling can also be performed using Monte Carlo simulations, enhanced sampling MD simulations [98], and any molecular simulation method that explores configuration space.

The ML potential models cannot be 'extrapolated' to the configurations that they are not trained on [34], thus the sampling should generate training datasets that explore the relevant configuration space as completely as possible. In the context of the DP-GEN scheme, the configurations in the training dataset are proposed in the exploration step, thus it should explore the relevant configurations as complete as possible by exploiting the high efficiency of the DP itself. The design of the exploration strategy depends on the applications for which the DP is intended. For example, a DP for liquid water properties need not fully explore ice configurations. A DP for chemical reactions may need enhanced sampling techniques to explore reaction pathways usually not available by standard MD simulations [99].
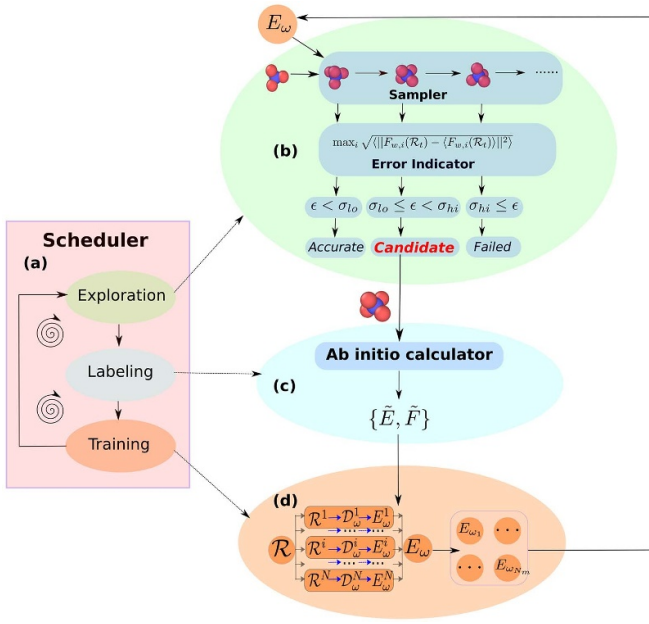
Figure 4. Schematic illustration of the DP-GEN scheme. (a) The DP-GEN scheduler runs iteratively, performing three steps: exploration, labelling, and training. (b) In the exploration step, different structures are sampled using DP MD and an error indication is applied to choose the candidates for labelling. (c) In the labelling step, DFT calculations are performed on the candidates to obtain the energy, forces, and virial tensor. (d) In the training process, DeePMD-kit package is used to train a new DP based on the initial datasets and candidates from each iteration. Reprinted from [84], Copyright (2020), with permission from Elsevier.

For each explored configuration, the difference between the DP prediction and the ground truth (i.e. error) is *estimated* from the ensemble of models without referring to any DFT calculation. The error indicator is defined as the maximal deviation of the forces predicted by the ensemble of models. More precisely, the model deviation is:

$$\epsilon_t = \max_i \sqrt{\langle ||\boldsymbol{F}_{w,i}(\mathcal{R}_t) - \langle \boldsymbol{F}_{w,i}(\mathcal{R}_t)\rangle||^2 \rangle}, \qquad (31)$$

where $\boldsymbol{F}_{\omega,i}(\mathcal{R}_t)$ is the force on atom i predicted by model E_ω and the ensemble average $\langle \cdots \rangle$ is taken over the ensemble of models. The ensemble average $\langle \boldsymbol{F}_{\omega,i}(\mathcal{R}_t)\rangle$ can be approximated as:

$$\langle \boldsymbol{F}_{\omega,i}(\mathcal{R}_t)\rangle = \frac{1}{N_m} \sum_{\alpha=1}^{N_m} \boldsymbol{F}_{\omega_\alpha,i}(\mathcal{R}_t), \qquad (32)$$

where N_m is the number of models in the ensemble (usually 4). Calculation of the model deviation requires N_m model inferences. Such an error estimate (over all explored configurations) provides an efficient measure of the model deviation for each configuration. Calculating 'real' errors by comparing with DFT calculations would be extremely prohibitive.

How the model deviation indicates the error is shown in the schematic figure 5. Four deep learning (DL) models (potentials), DL 1 to 4, are trained from the same training data (blue crosses) with random initialisation of the model parameters.
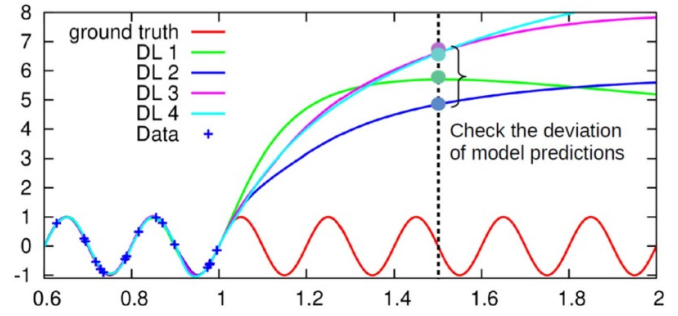

Figure 5. Visualisation of the model deviation. DL 1 to 4 indicates 4 Deep Learning models trained from the existing data (blue crosses).

At times, DL1 to DL4 all show small training error in the region 'covered' by the training data, but the predicting error is large in the region far away from the training data. The standard deviation of the DL model predictions follows exactly the same trend. In the region 'covered' by the training data, all models/potentials are trained against the same target, thus their predictions agree with each other, while in the region far from the dataset the different DL potentials (training from different initialisations lead to different potentials) yield different predictions. Therefore, the model deviation is a good indicator of the DL prediction error.

Upper and lower trust levels, σ_{hi} and σ_{lo}, are used to select configurations for labelling. When the error indicator (ε) is smaller than σ_{lo}, it suggests that the atomic forces of all atoms in the configuration are accurately predicted and there is no need to do DFT labelling for this configuration. When $\epsilon \geqslant \sigma_{hi}$, the potential behaves poorly and DFT labelling will not be performed, because such explored configurations may be non-physical (e.g. overlapping atoms) due to poor potential quality. Only when $\sigma_{lo} \leqslant \epsilon < \sigma_{hi}$, will the configuration be selected as candidates for DFT labelling. The user may also set the maximal number of structures in the candidate set to perform DFT labelling; this leads to a randomly sampling to form the labelled dataset.

(b) Labelling. In the labelling step (figure 4(c)), a DFT software package is called to calculate the energy, force and virial tensor on the selected configurations. The labelled data are *added* to the existing training dataset.

(c) Training. In the training step (figure 4(d)), an ensemble of new potentials is trained based on the same training dataset but with different and random initialisation of the potential parameters. Relatively short trainings are performed at this step, because the configurations are selected for labelling by the force criteria, and reasonably accurate DP forces can be obtained with relatively short trainings. Experience shows that 400 000 to 2 000 000 training steps are appropriate. One may also initialise the potential parameters in the ensemble of potentials from the previous iteration, and start the training in the next iteration with a relatively small initial learning rate (e.g. 10^{-4}), and with a bias to the new training data.

(d) Initial datasets for starting the DP-GEN workflow. The DP-GEN workflow introduced above is automatic, but human intervention is required at the beginning when no DP exists. Generally, we generate some initial datasets and train an

ensemble of initial DPs to kick off the workflow. In metals and alloys, the generation of initial bulk datasets usually consists of the following steps. (a) Start from a supercell $(2 \times 2 \times 2)$ of crystal unit cells (BCC, FCC, HCP). (b) Compress and dilate the supercell uniformly to cover a range of densities. (c) Randomly perturb the atom coordinates and supercell vectors. (d) Randomly replace atom elements to generate alloys. (e) Start from different structures and do a short AIMD at low temperature (100 K). (f) Collect the DFT results and generate initial training datasets. The initial datasets are not particularly important as long as they reasonably cover the starting structures of DP MD in the exploration step to avoid breakdown of the DP MD within the first few steps. This generating initial dataset method is simply one possibility; other methods such as generating initial structures according to crystal symmetry in genetic algorithm [80] are also applicable.

2.3.3.2. A practical guide. Here we provide a practical guide for generating a general-purpose DP for a bulk metal or alloy using the DP-GEN scheme.

(a) Generation of initial datasets. Initial datasets are needed to generate the first ensemble of DPs. Because initial datasets are only used to start DP-GEN, datasets with only several hundred different configurations should suffice. The accuracy of the DPs is not sensitive to the choice of the initial datasets and would be gradually improved as iterations increase assuming that the DP-GEN run parameters are set appropriately.

(b) Exploration. The exploration strategy is not unique and is determined by the user according to the applications of interest. Here, we focus on the general exploration strategy using, again, bulk metals or alloys as our example. DP-based MD simulations at different temperatures and pressures are employed to explore the configurational space. For efficient sampling, we increase the temperature during the exploration. If the highest melting point of the element in an alloy is T_m, we usually divide the temperature range from 50 K to $2T_m$ into 4 regions. (a) [50 K, (0.1 , 0.2 , 0.3 , 0.4)T_m] (b) [0.5, 0.6, 0.7, 0.8, 0.9]T_m; (c) [1.0, 1.1, 1.2, 1.3, 1.4]T_m; (d) [1.5, 1.6, 1.7, 1.8, 1.9]T_m. For each temperature region, the pressure range is varied over [0.001, 0.01, 0.1, 1, 5, 10, 20, 50] kBar. In this case, there are $5 \times 8 = 40$ different MD conditions in each temperature region. Other parameters for the DP MD simulations is the MD simulation duration and the number of different initial structures. At the beginning of each new temperature region, the number of MD step should be small (500 or 1000) and the number of starting structures should also be small (e.g. 5 each for different distorted crystal supercells). This is because the DPs have little 'knowledge' of this new temperature region such that the expected accuracy is low. As the iteration number increases within a temperature region, the MD runs are longer and the number of starting structures is increased.

After the MD runs, candidate systems are selected based upon the trust levels σ_{lo} and σ_{hi}. The key idea is to make DP-GEN converge within each temperature region. If σ_{lo} and σ_{hi} are too strict, for example σ_{lo} is too low, the accurate ratio $(\epsilon < \sigma_{lo})$ stays low (smaller than 90%) in the temperature region. If σ_{hi} is too high, too many structures are selected as

Table 1. Trust levels σ_{lo} and σ_{hi} employed in DP-GEN for several systems.

System	$[\sigma_{lo}, \sigma_{hi}]$
Mg [83]	[0.03,0.13]
Al [83] & Al-Mg [83]	[0.05,0.15]
Cu [84]	[0.05,0.20]
Mg-Al-Cu [100]	[0.05,0.20]
Ti [101]	[0.10,0.25] at $T < 1.5T_m$[a] for bulk exploration and [0.15,0.30] elsewhere
W [102]	[0.20,0.35]
Ag-Au [103]	[0.05,0.20]
water [95]	[0.15,0.25] in first 24 iterations [0.18,0.32] in iterations 25 to 32 [0.20,0.35] in iterations 33 to 36
SiC [104]	[0.15,0.30]
Li$_{10}$(Ge,Si, or Sn)P$_2$S$_{12}$ [105]	[0.12,0.25]

[a] T_m is 1941 K, which is the experimental melting point for Ti.

candidates for labelling and the ratio of the failed structure $(\epsilon \geqslant \sigma_{hi})$ is very low. The values of σ_{lo} and σ_{hi} used in our previous work for different systems (see table 1); σ_{hi} is normally 0.15–0.30 eV Å^{-1} higher than σ_{lo}. In principle, σ_{lo} should be higher than the lowest error that a DP can achieve (error introduced by data, fitting ability of optimiser, representability of the DP). σ_{hi} should be set so as to eliminate unphysical configurations from the dataset, since the convergence of the DFT calculations for such configurations is difficult to obtain. See [84] for additional guidance.

(c) Labelling. After candidate structures are selected from the exploration step, users can choose the maximal number of structures sent to DFT labelling in each iteration. On the one hand, if there is a large number of structures for DFT labelling in one iteration, the computational cost for labelling would be high and this dataset of configurations may be redundant, leading to wasted computational resources. On the other hand, if the number of structures for DFT labelling is too small (e.g. one) in one iteration, the datasets after DP-GEN run would be very compact, requiring many iterations, which again would be computationally expensive for training in each iteration. In this case, the maximal number of structures for DFT labelling in one iteration should be set appropriately, we typically set this number to $\sim$100.

(d) Training. The prefactor of the virial tensor in the loss function (equation (3)) is usually set to zero in the DP-GEN training loop for two reasons. First, only the model deviation of the forces is used to select candidates in the exploration step; this is irrelevant for the virial tensors. Second, longer training is needed to obtain a DP with good accuracy on both forces and virial tensors. The training step in each iteration of DP-GEN should be small, so we do not initially train on the virial tensors.

(e) DP-GEN convergence. In principle, DP-GEN is considered converged when the model deviation (ϵ) on each structure is smaller than the trust level σ_{lo}. However, there are often some (very low probability) configurations with undesirable accuracy in long enough MD simulations. These

configurations generally do not improve the DP performance. In this case, DP-GEN allows users to set the convergence criterion at which level the DP is regarded as converged. For example, the user can skip the labelling step when the percentage of configurations with model deviation smaller than σ_{lo} is larger than 99%. After convergence, long training steps (usually over 8 000 000 steps) are performed on all DP-GEN loop datasets by including labelled energy, forces, and virial tensors. Comprehensive tests of the resulting model are then performed and the Autotest in DP-GEN package is applied to perform different property tests (see below). The DP obtained after careful DP-GEN run and long training is normally good for general purposes, but does not guarantee good performance for subtle properties. Specialisation for these properties is required and the details of specialisation are presented in section 2.5.2.

2.3.3.3. Autotest. After one or a series of DPs is trained, we employ the Autotest package, to calculate a simple set of properties and/or perform tests for comparison with DFT and/or empirical interatomic potentials (EAM, MEAM, and etc). Because the DP was obtained by fitting DFT calculation results, the degree of agreement between DP and DFT for a series of properties is usually excellent; if not, this comparison provides a guide for further training and parameter settings. Autotest, as part of the DP-GEN package, has standardised the calculation of some critical physical properties and provides a series of reliable benchmark testing to better evaluate the performance of the DPs. As of this writing, Autotest includes the calculation of the following set of properties (additional properties are added continuously): (a) equilibrium structural parameters (relaxation), (b) equation of state (eos), (c) elastic constants (elastic), (d) vacancy formation energy (vacancy), (e) interstitial formation energy (interstitial), and (f) surface formation energy (surface).

Autotest can also use LAMMPS or VASP to reproduce and refine previous calculation results. The current package is mainly targeted for simple crystal structures (metals and alloys), but is extensible for users to implement new features. The detailed manual and framework of Autotest package is available at web link 8 in section S2 of the SM.

2.4. The DP Library

The DP-GEN package provides a relatively automatic routine for generating DPs, and the DP Library project is a place for sharing and publishing the DPs and the training datasets. First, DP Library is a place for model sharing, which is similar to other interatomic potential repositories like the NIST (web link 9 in section S2 of the SM) and OpenKIM (web link 10 in section S2 of the SM). Second, DP Library provides an opportunity for data sharing. One can contribute and download DFT datasets used to train a published DP. If a DP needs to be refined or improved, one can first add new data to the downloaded training datasets from DP Library and then perform training using DeePMD-kit. For example, one may develop a DP for A-B alloys based on the training data for element A and element B, instead of generating the datasets by him/herself.

Figure 6. Elements for which DPs are currently available in the DP Library (web link 11 in section S2 of the SM) are indicated in black.

The settings (usually settings to use a DFT software) used to generate the dataset are asked to provide, so the new data can be generated in the same way as the downloaded dataset. The shared training data makes the published DPs reproducible and improvable. DP Library can be accessed at web link 11 in section S2 of the SM and an image of the website is shown in figure 6. In the periodic table, the available DP for elements is in black and details about the DP and DFT data can be found by just several clicks. For more details, please refer to the manual of DP Library at web link 12 in section S2 of the SM.

2.5. Efficiency and accuracy of DPs for applications

To make DP more practical for applications in different materials systems, efficiency and accuracy must be balanced.

Although DP is very much faster than DFT, it is still much slower than empirical interatomic potentials like EAM and MEAM. This is not surprising given the vast number of parameters in the neural net of the DP (compared to simple empirical potentials such as EAM). Hence, optimisation of the efficiency in use of the DP can greatly enhance the practical usability. Recently, a highly optimised GPU version of DeePMD-kit pushed the limit of MD with *ab initio* accuracy to 100 million atoms [50, 106] (recognised by the 2020 Gordon Bell Prize). Further optimisations are still possible based on advances in neural networks.

DPs generated by the example exploration protocol (section 2.3.3) may not be accurate for all applications. For example, the Ti DP yields screw $\langle \mathbf{a} \rangle$ dislocation properties that are not consistent with experiment (or DFT). For some particular applications, the DP must be tuned or specialised for important, yet subtle properties. This may be accomplished through specialisation, as discussed below [101].

We first introduce DP model compression which can easily accelerate DP by a factor of 4 to 18, based on experience. Next, we discuss DP specialisation and consider the example of Ti, mentioned above.

2.5.1. DP compression. The most computationally intensive part of using the DP is the evaluation of the embedding net (equation (23) or (24)) and the assembly of the descriptor

by equations (26)–(28) [107]. The goal of DP compression is to reduce the computational and memory overhead associated with the embedding net with little loss of accuracy. The evaluation of the DP involved the mapping from a scalar to a vector of dimension M, each dimension of which can be approximated by a piece-wise fifth order interpolating polynomials. The range of the embedding net is first discretised by nodes $x_1, x_2, \ldots, x_l, \ldots, x_{L+1}$, and the lengths of intervals (tabulation step) are assumed to be the same. On each interval, e.g. $[x_l, x_{l+1})$, the 6 fitting parameters of the fifth-order polynomials are uniquely determined by matching the value and the first and second derivatives of the polynomial to those of the embedding net on nodes x_l and x_{l+1}. This approximation is referred to as *tabulation*, because the parameters of the polynomials are stored in a table, and use involves simply a look-up the table for the polynomial when evaluating the embedding net. The accuracy of the tabulation is controlled by the tabulation step; a size of 0.01 leads to errors in energy and force smaller than 10^{-7} eV atom^{-1} and 10^{-6} eV Å^{-1}. After tabulation, the multiplication between the embedding and the environment matrices becomes the bottleneck. The embedding matrix is stored in the memory after it is computed and then loaded from memory to the register for the matrix multiplication. This requires significant I/O with the memory. This can be eliminated by merging the tabulation and the multiplication; i.e. once one component of the embedding net is computed by tabulation, it is immediately multiplied with the environment matrix pre-loaded in the register, and accumulated to the result. With this optimisation, the multiplication between the embedding net and the redundant zeros in the environment matrix can be avoided.

The DP compression was benchmarked for Cu, H_2O and Al-Cu-Mg ternary alloy DPs [107]. The model inference was accelerated by 9.7, 4.3 and 18.0 times on a CPU and by 9.7, 3.7 and 16.2 times on an Nvidia V100 GPU for Cu, H_2O and Al-Cu-Mg, respectively. The maximum number of atoms handled by one GPU also increased from 12, 49 and 5×10^3 to 129, 246 and 61×10^3, respectively.

Figures 7(a) and (b) display the speed comparison of the compressed Ti DP [101] with an EAM [108], and an MEAM potential [109] on a CPU and GPU machine. Note that the DP has a larger radius cutoff distance than EAM and MEAM in this case. On CPUs, the compressed DP is 200-300 times slower than EAM potentials and 30-40 times slower than the MEAM potential. On GPUs, the compressed DP is 20-30 times slower than the EAM potential (MEAM is currently not ported to GPU in LAMMPS). All potentials show a linear scaling with the number of atoms. Because of this linearity and speed, the compressed DP can be used to perform large scale MD simulations to investigate a wide range of properties with *ab initio* accuracy; e.g. defect properties, phase transformations, and transport coefficients.

DP compression is supported in the DeePMD-kit package (releases beyond 2.0.0) and the compressed model can be easily generated using the following command:
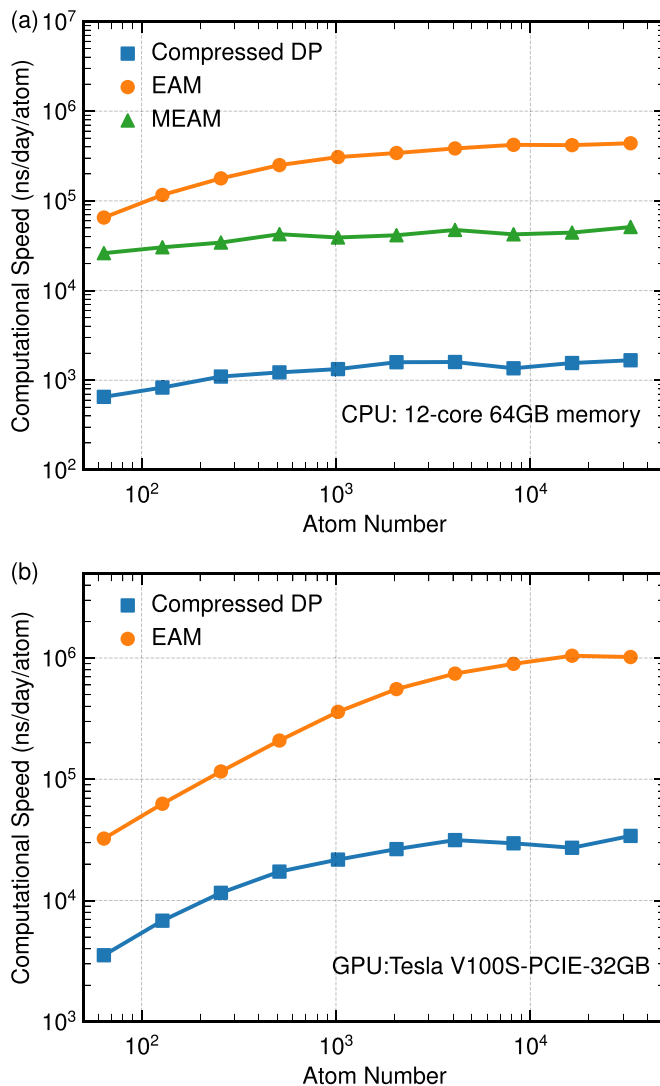

Figure 7. A comparison of the speed of MD simulations using the compressed Ti DP, an EAM, and/or an MEAM potential on (a) CPU and (b) GPU systems [101].

```
1 dp compress -i graph.pb -o graph-compress.pb
```

Using the optional -s flag, followed by the tabulation step with the default value of 10^{-2} typically gives very accurate compressed DPs (compared with the original). It is recommended that the user compare the values of a few key properties from the original and compressed DPs.

2.5.2. Specialisation. The DP generated from the DP-GEN scheme described above may not be sufficiently accurate for some complex phenomena and specialisation of this general-purpose DP may be required. This is not surprising since the exploration strategy may not provide a sufficient sampling of the relevant local structures that are inherent to the complex phenomena of interest. Therefore, some special structures should be added to the training process to better represent the
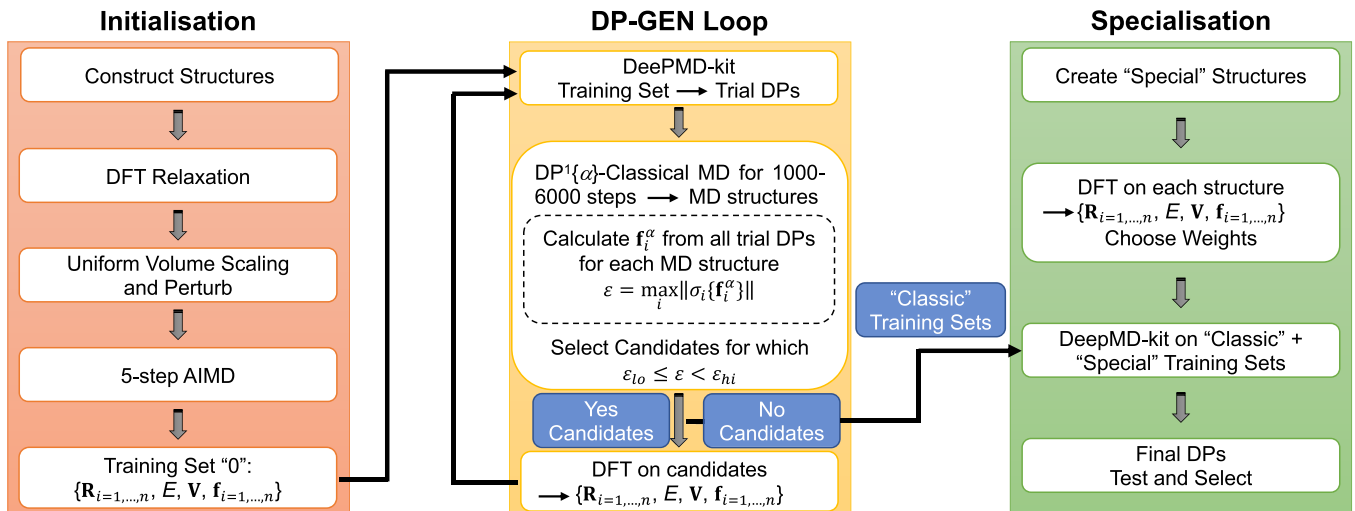
Figure 8. The workflow for specialising a general DP. $\mathbf{R}_i$ is the atomic coordinate of atom i, E is the total energy of one configuration, $\mathbf{V}$ is the virial (stress) tensor of one configuration, $\mathbf{f}_i$ is the force on atom i, n is the number of atoms in one configuration, DP^1 is the first ensemble of trial DPs, α labels the αth DP in the ensemble, σ is the standard deviation, and ε_{lo} and ε_{hi} are two thresholds in DP-GEN. The specialisation step is shown in the green box. Reprinted by permission from Springer Nature Customer Service Centre GmbH: [Nature] [npj Computational Materials] [101].

requisite subtle properties. The workflow for specialising DP is shown in figure 8. The initialisation and DP-GEN loop steps were discussed in section 2.3.3.

In the specialisation step, 'Special' structures are first created based on the special properties/structures of interest. For example, if the DP elastic constants are not sufficiently accurate, 'Special' structures may be created corresponding to a range of different crystal deformations. If the specialised DP is to describe dislocation properties in complex crystal, then 'Special' structures that include sheared structures akin to those used in determining the generalised stacking fault energy γ-lines [110] may be of use. DFT calculations are then performed on these 'Special' structures and the energies, atomic forces, and virial tensors along with the configuration form the 'Special' training sets. The 'Special' training sets then combine with the 'Classic' training sets from DP-GEN loop and weights are chosen for the 'Special' training sets. The default weight is 1 but this weighting of the 'Special' training sets should be increased because (1) the number of 'Special' training sets are usually much smaller than those in the 'Classic' training sets, and (2) experience shows this may be necessary to properly reproduce the special properties. Next, DeePMD-kit package is applied to retrain the DP on all of the training sets. Finally, the specialised DP is further tested to insure the best overall performance.

Here, we show an example of DP specialisation for the mechanical response of Ti [101]. Figures 9(a) and (b) present a comparison of different empirical interatomic potentials with DFT on the γ-lines of Basal and Prism planes in HCP Ti. Illustrations of the Basal and Prism planes of HCP Ti are shown in the insets of figures 9(c) and (d). All of these empirical potentials yield inaccurate γ-line profiles and/or generally predict low stacking fault energy (the first minimum at $\sim$0.33 along the x-axis) on the Basal plane with respect to DFT. These

empirical potentials are not specialised for dislocation properties. When general-purpose DP is obtained from DP-GEN scheme, it also shows similar systematic inadequacy as the other empirical potentials: low Basal and high Prism stacking fault energy, which prohibits the use of a generalised DP for investigating dislocation properties in HCP Ti. Specialisation is then performed by adding the structures in the red dashed box as well as the origin in figures 9(c), (d) and the structures along γ-lines in figure 9(e), all with a weight of 100. The DP γ-lines in figures 9(c)–(e) are overall in good agreement with DFT. In particular, the stable stacking fault energy order Basal > Prism > Pyramidal I narrow (the stable stacking fault energy on this plane will decrease significantly after full relaxation compared to that from γ-line) is different from previous empirical potentials in figures 9(a) and (b) and follows the trends observed in the DFT data. Based on this, the relative energies of the screw $\langle \mathbf{a} \rangle$ dislocation on the Prism and Pyramidal I planes are shown in figure 9(f). The screw $\langle \mathbf{a} \rangle$ dislocation is more stable on Pyramidal I plane than on Prism plane for DP, which agrees with previous DFT calculations and experimental measurements [113]. We refer readers to [101] for more details. As an additional example where the difference between DP and empirical potentials is particularly vivid; we consider the case of the screw $\langle c+a \rangle$ dislocation core on the Pyramidal I plane of HCP Ti, as determined using DFT, DP [101], and the most widely used interatomic potential [109] in figure 10. The DP-determined core-structure shares the key features of the DFT-determined core structure; a long extended region and an asymmetric compact, localised core. The DP- and empirical interatomic potential-determined dislocation move very differently. This example of Ti DP demonstrates the importance of specialisation of the DP and the superior flexibility of DP over other empirical potentials.
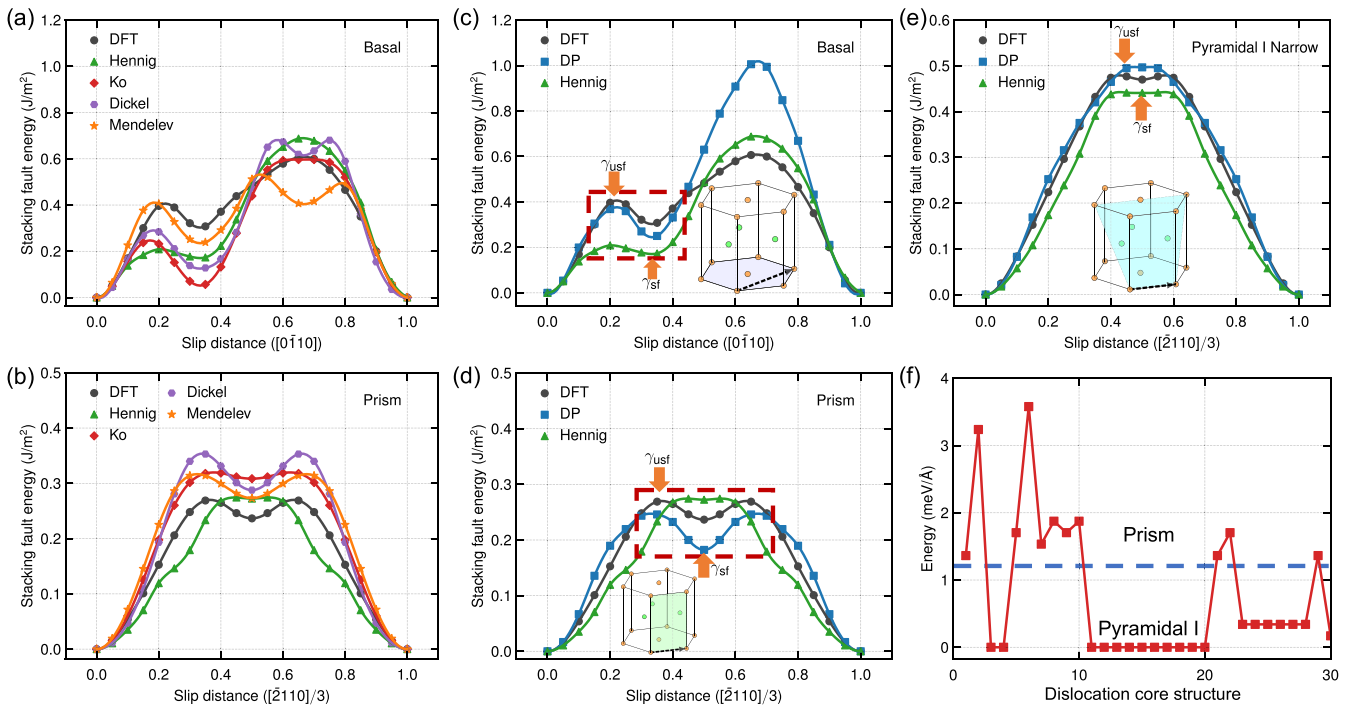
Figure 9. The generalised stacking fault energy (γ-lines) on the (a), (c) Basal, (b), (d) Prism, and (e) Pyramidal I plane of HCP Ti calculated using DFT, empirical potentials (Hennig [109], Ko [111], Dickel [112], and Mendelev ('Ti3' in [108])), and a specialised DP. The configurations in the dashed red box and at zero slip (origin) are included in the training sets. All configurations on the Pyramidal I narrow γ-line (e) are included in the training dataset. (f) Each point denotes the energy of a simulation cell containing a screw $\langle \mathbf{a} \rangle$ dislocation core from a 600 K MD simulation after quenches to zero temperature (each point is one picosecond apart in the MD simulation). Two types of screw dislocation cores are observed: one delocalised onto a Prism plane (above the blue line) and one on Pyramidal I plane (below the blue line). Some of the cores are slightly distorted leading to small variations in the energy. See [101] for details.
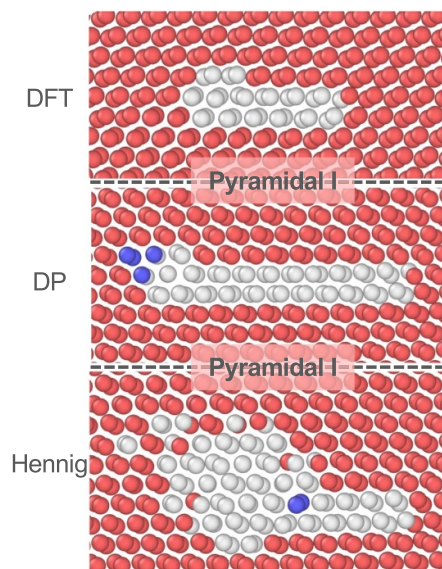

Figure 10. The screw $\langle c + a \rangle$ dislocation core structure on the Pyramidal I plane of HCP Ti determined via DFT, DP [101], and the Hennig MEAM potential [109]. The red, blue, and white shaded atoms denote local HCP, BCC, and indeterminate atomic environments.

3. DP applications in materials science

In the past three years, DPs have been applied in a number of systems in materials science including (1) elemental bulk systems, (2) multi-element bulk systems, (3) aqueous systems, (4) molecular systems and clusters, and (5) surfaces and low-dimensional systems. Table 2 shows a list of the material systems to which DPs have been applied (as of the writing of

Table 2. Example applications of DP in materials science.

System	Reference
Elemental bulk systems	
Al	[83, 114–118]
Mg	[83]
Cu	[84]
Ti, W	[101, 102]
Ag, Au	[103, 119, 120]
Li	[121]
Be	[122]
Ga	[123]
Sb	[124]
C	[125]
Si	[126, 127]
P	[98]
Multi-element bulk systems	
Al-Mg, Al-Cu-Mg	[83, 100, 128, 129]
Al-Cu, Al-Zn-Mg	[130, 131]
Al-Cu-Ni	[132]
Ag-Au	[103, 119]
Pd-Si, Nb_5Si_3, $Zr_{77}Rh_{23}$, Bi_2Te_3	[133–136]
$Al_{90}X_{10}$ (X = Tb, Cr, or Ce)	[137–140]
$(Pd, Pt)_x(Ge, Sn, Pb)_y$	[141]
P_2Sn_5	[142]
Silica, silicate	[143–146]
SiC	[104, 147]
B_4C	[148]
Molten salt LiF, FLiBe, and chloride	[149–157]
Li or Na-based battery materials	[105, 158–162]
TiO_2	[163]
β-Ga_2O_3	[164]
Ferroelectrics HfO_2	[165]
Ag_2S	[166]
MoS_2	[167]
SnSe	[168]
$Zr_{1-x}W_xB_2$	[169]
$(Hf_{0.2}Zr_{0.2}Ta_{0.2}Nb_{0.2}Ti_{0.2})X$ (X = C or B_2)	[170, 171]
Aqueous systems	
Water	[95, 172–183]
Zinc ion in water	[184]
Water–vapour interface	[185, 186]
Water–TiO_2 interface	[187]
Ice	[188, 189]
Molecular systems and clusters	
Organic molecules	[99, 190–195]
Metal and alloy clusters	[119, 196]
Surfaces and low-dimensional systems	
Metal and alloy surfaces	[103, 119, 129]
Graphane	[125, 197]
Monolayer In_2Se_3	[198]
2D Co-Fe-B	[199]

this paper). We choose several examples from each category to briefly discuss the corresponding DP application and how DP aids materials science research.

3.1. Elemental bulk systems

To date, DPs have been applied to a wide-range of pure systems, including Al, Mg, Cu, Ti, W, Ga, C, Si, as shown in table 2. Al was the first metal system to which DP was applied and a general-purpose DP developed [83]. It accurately reproduces the lattice parameter, elastic constants, vacancy and interstitial formation energies, surface energies, twin and stacking fault energies, melting point, enthalpy of fusion, diffusion coefficient. The general-purpose DP for other metal elements is also accurate for the same properties. For other properties, not included in the training datasets, DP is in better agreement with DFT than MEAM [10] for phonon dispersion relations, equations of state, and the liquid state radial distribution function [83]. Based on this, Wang *et al* [114] smoothly interpolated the Ziegler-Biersack-Littmark (ZBL) screened nuclear repulsion potential with a DP to obtained a DP-ZBL model for irradiation damage simulations, surpassing the widely adopted ZBL MEAM [200] or EAM [201] potentials. Later, a DP was developed for warm dense Al to simulate ion dynamics near the hydrodynamic limit [115], structural and dynamic properties [116], and electronic and ionic thermal conductivities [117]. A DP was also developed for high temperature and high pressure liquid Al to calculate shear velocity [118].

Dislocation properties play important roles in the plastic response of most structural materials, including Ti and W. The specialised Ti DP accurately depicts the γ-lines on different planes (figures 9(c)–(e)) and the screw $\langle \mathbf{a} \rangle$ dislocation core energy ordering between the Prism and Pyramidal I planes (figure 9(f)). In addition, the screw $\langle \mathbf{a} \rangle$ dislocation core structures on Prism and Pyramidal I planes are also in surprisingly good agreement with DFT results [113]. Because the dislocation core structures could not be explicitly included into DFT training datasets, the example of Ti DP shows that surrogate properties can be used to optimise DPs for dislocation properties of complex HCP systems. While the dislocation core structures of BCC W are not as complex as in HCP Ti (because it only adopts a compact core), the Peierls barrier for BCC W is difficult to reproduce with other empirical potentials (cf DFT results [102]). The DP-SE2 model with only a two-body embedding descriptor yields a very poor prediction on the Peierls barrier, but the DP-HYB that hybridises descriptors with two-body and three-body embeddings reproduces this property very accurately [102]. The Ti and W examples demonstrate that DPs can accurately describe dislocation core structures and Peierls barrier of BCC and HCP metals.

The DP approach has also been applied to Ag and Au (widely used in catalytic applications). Andolina *et al* [119] and Wang *et al* [103] developed DP for Ag and Au that is accurate at lattice parameters, elastic constants, surface formation energies, interstitial and vacancy formation energies, etc. Furthermore, Andolina *et al* [119] got accurate adsorption energy and diffusion barriers for adatoms on {100}, {110}, and {111} compared to DFT results. Wang *et al* [103] presented a comprehensive study of the Au {111} surface reconstruction using a DP that yields excellent agreement with DFT results. From another perspective, Chen *et al* [120] used DP

MD to illustrate the dynamics compression process of Au. The developed DP could accurately reproduce the experimentally determined phase boundaries and the short-to-medium range orders are proposed to reduce the Gibbs free energies of the shocked structures. The examples above for Ag and Au validate the applications of DP in both catalytic and shock compression areas.

In addition, DP has also been applied in many other elemental bulk systems and here we only list some of the examples in table 2. For Li, which is an important element for battery, Jiao *et al* [121] developed DP to reveal self-healing mechanisms in a large Li-metal system. For Ga, Niu *et al* [123] used DP to construct the phase diagram of liquid Ga, α-Ga, β-Ga, and Ga-II, in good agreement with experimental results. In addition, the local structure of liquid Ga and the nucleation process into α-Ga, β-Ga were also studied [123]. For carbon, a DP was developed by Wang *et al* [125] to simulate the structural properties of 12 different bulk and low-dimensional carbon structures. For Si, which has both covalent and metallic bonding behaviour, the first DP was trained based on datasets generated by classical metadynamics simulations [126]. The DP was then applied to study the crystallisation and the free energy surface between liquid and solid. Many thermodynamics properties near the critical point were found to be close to experimental data. Li *et al* [127] trained a DP on DFT data of silicon in the crystalline, liquid, and amorphous phases and thermal conductivity was accurately reproduced. DPs were used to study a liquid-liquid phase transition in P, where DP established the main features of the liquid phase diagram [98]. In particular, DP phase diagrams of Ga and liquid P clearly indicate how DPs accurately describe the PES of different phases and transitions between them.

The applications in elemental bulk systems are among the first applications of DPs in materials science. Their successes for different crystal structures and various classes of properties (mechanical, catalytic, irradiation properties, phase transformation, thermal conductivity, etc) encouraged their extension to multi-element bulk systems and increasingly complex phenomena.

3.2. Multi-element bulk systems

Al-Mg was the first alloy system for which an accurate DP was developed [83]. This DP was used to describe the 28 crystalline Al-Mg alloys structures in the Materials Project (MP) database [202]; include accurate prediction of formation energies, equilibrium volumes, elastic constants, vacancy and interstitial formation energies, and unrelaxed surface energies. Wang [128] applied this Al-Mg DP and validated its reliability for crystal structure prediction by using DP+CALYPSO. This work interfaces the DeePMD-kit package and crystal structure prediction software (e.g. CALYPSO [203], USPEX [80], and Pychemia (web link 13 in section S2 of the SM)). Andolina *et al* [129] developed a Al-Mg DP based on the original DP to investigate anisotropic surface segregation. Based on the Al-Mg DP, a Al-Cu-Mg ternary DP was developed for the entire compositional space [100]. 2.73 billion alloy configurations were explored in the DP-GEN process. The resulting

DP yields more accurate results for energetic, mechanical, and defect properties of 58 crystalline structures as compared with MEAM potentials [10]. The multi-component DP approach can readily be applied to high-entropy alloys for which adequate empirical potentials are difficult to obtain.

Bourgeois *et al* [130] built an Al-Cu DP to simulate the aggregation of vacancies around embedded θ' precipitates and investigated the nucleation of this strengthening phase onto a template structure. An Al-Mg-Zn DP was developed [131] and applied to confirm the co-segregation of Mg and Zn atoms at a precipitate and matrix interface. DPs have also been found to be powerful and promising for the prediction of the structure and dynamics of metallic liquids, glasses, and quasi-crystal [133–140]. The DP developed for Pd-Si accurately represented the structure of liquid and crystal structures, melting points, and glass-forming ability at compositions near Pd_3Si and Pd_9Si_2 (more accurately than existing EAM potential) [133]. Accurate liquid structure and dynamic properties were also obtained with DPs for Nb_5Si_3 [134], $Zr_{77}Rh_{23}$ [135], and Bi_2Te_3 systems [136].

Tang *et al* [137–140] performed DP MD simulations of a series Al-based alloys; we focus now on Al-Cr quasicrystals [139] as an example and a demonstration of how DPs can be used together with experimental studies. Dendritic growth of metastable quasicrystals were observed in the $Al_{13}Cr_2$ approximant phase (formed from $Al_{90}Cr_{10}$ thin film) by pulsed laser deposition [139] which is structurally similar to quasicrystal of the $Al_{13}Cr_2$ matrix. The Al-Cr DP was used to simulate the quenching of the $Al_{90}Cr_{10}$ alloy from 2200 to 700 K at 10^{11} K s^{-1}. There are three types of 13-atom icosahedra in the approximant $Al_{13}Cr_2$ phase and one icosahedral Al-Cr quasicrystal motif. These 4 icosahedral motifs are similar despite slightly different Cr-Al bond lengths. All 4 types of 13-atom icosahedral motifs were Cr-centred. The icosahedral motif appears in both the quasicrystal and approximant structures, which results in the survival of the 13-atom icosahedron after laser irradiation. This was observed in both simulation and experiment; the success of the DP was attributed to the excellent reproduction of the liquid structure in the DP-based simulations.

A DP was developed to simulate liquid and glassy silica which proved to have satisfactory accuracy based upon a relatively small training dataset [143]. Other DPs were developed to calculate transport properties of silicate in the mantle [144–146]. DPs were also employed in large-scale calculations of thermodynamic, transport, and structural properties in different molten salts [149–157].

Another exciting class of DP applications was for Li and/or Na-based battery materials [105, 158–162]; here we focus on the example of $Li_{10}GeP_2S_{12}$-type superionic conductors [105]. The DP-GEN scheme was used to generate DPs for three solid-state electrolyte materials ($Li_{10}GeP_2S_{12}$, $Li_{10}SiP_2S_{12}$, and $Li_{10}SnP_2S_{12}$) and applied to diffusion over a wide temperature range with ~1000 atoms. The predicted diffusion coefficients slightly overestimated the experimental values but were within the experimental uncertainty. These DP-based simulations provided a starting point for large size scale and long time scale MD investigations of solid-state electrolyte materials.

Additional DPs were developed for a wide-range of other multi-element bulk systems, including metal oxide [163–165], metal sulfide [166, 167], thermoelectric SnSe materials [168], metal borides [169, 171], and metal carbide [170] systems.

3.3. Aqueous systems

Since the original DP paper [48], water has been widely studied with DPs. Ko *et al* [172] applied DPs to perform extensive sampling of thermal and nuclear quantum fluctuations on an accurate PES. In particular, a DP was used to investigate isotopic effects on structural properties of liquid water (H_2O and D_2O). To understand the experimentally observed isotope effect in the x-ray absorption spectra between liquid H_2O and D_2O, DP-based, path-integral MD simulations were performed [174]. A more comprehensive study by the same group [177] examined isotope effects on radial distribution functions, O-O-O triplet angular distributions, density and found that the DP-based simulations were consistent with the experimental observations. Recently, Calio *et al* [183] performed DP-based simulations to interpret experimental observations on the hydrated excess proton in water.

Sommers *et al* [173] trained a DP to predict the polarisability of liquid water with *ab initio* accuracy in order to calculate the Raman spectra in long time scale. Gartner *et al* [175] trained a DP for water to examine the questions of the existence of a liquid-liquid transition in water. Andreani *et al* [176] combined neutron scattering experiments and DP MD to investigate hydrogen dynamics in supercritical water. The vibrational densities of states observed in DP MD showed coupling between intramolecular vibrational and intermolecular librational and rotational motions. Piaggi *et al* [178] used a DP to investigate ice nucleation in water, hexagonal ice, and cubic ice and obtained quantitative agreement between DP and experiment (better than the state-of-the-art semi-empirical potentials). A more complete description of the phase equilibrium between different phases of H_2O was achieved through constructing the DP phase diagram from low temperature and pressure to $\sim$2400 K and $\sim$50 GPa [95]. Although both the DP and empirical TIP4P/2005 potential [204] work well at low and intermediate pressures, DP outperforms TIP4P/2005 at higher pressure compared to experimental results, especially at the phase boundaries between ice VIII, VII, and VI (figure 11). Unlike in experiments, the TIP4P/2005 predicts a first-order transition from ice VII to a plastic phase (no such phase transformation occurs using DP). This is a milestone for DP considering the importance of water, the vast range of temperatures and pressures, and the high accuracy required for free energy representation.

Tisi *et al* [179] calculated the thermal conductivity of water using both DFT (SCAN) and DP. Interestingly, both approaches yield the same conductivity which was $\sim$50% higher than the experimental value. Zhang *et al* [180] applied DP to help improve the exchange functional in DFT from SCAN to SCAN0 for water. Similarly, Torres *et al* [181] evaluated the errors of DFT-based simulations on structural and dynamical properties due to time- and size-scale limitations by using DP MD. In these two examples, the high efficiency
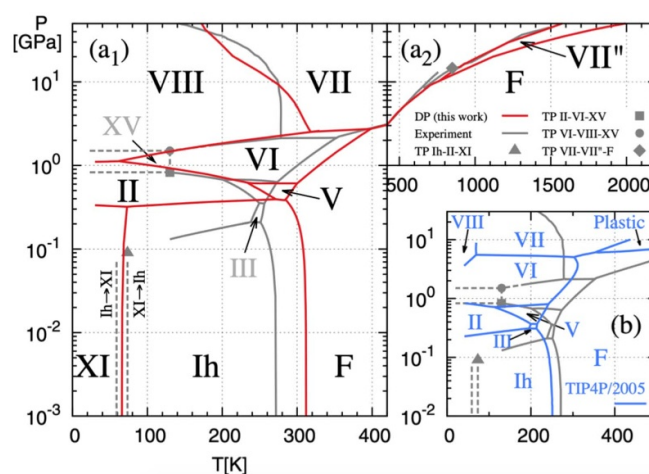

Figure 11. Phase diagram of water. (a_1) DP model (red solid lines) and experiment (gray solid lines) for $T < 420$ K. (a_2) Phase diagram at high T and P. (b) Phase diagram of TIP4P/2005 [204] water. Reprinted figure with permission from [95], Copyright (2021) by the American Physical Society.

and accuracy of DP provided rapid screening of different properties to feedback into DFT exchange-correlation functional optimisation. Shi *et al* [182] extended DP to produce accurate molecular multipole moments in the bulk and near interfaces consistent with AIMD simulations. These moments were used to compute the electrostatic potential at the centre of a molecular-sized hydrophobic cavity in water.

The DP approach has also been used in a wide range of aqueous systems. Xu *et al* [184] developed a DP to perform MD study of zinc ions in liquid water. The experimentally observed zinc-water radial distribution function, as well as the x-ray absorption near edge structure spectrum, was well-reproduced by the DP MD simulation. Recently, the Limmer group [185, 186] applied DP to study liquid-vapour interfaces. They found that the DP yielded accurate interfacial properties by incorporating explicit models of the slowly varying long-ranged interactions and training neural networks only on the short-ranged components [185]. In addition, they trained a DP for solvated N_2O_5 and bulk ambient water and applied DP MD and importance sampling to study the uptake of N_2O_5 into an aqueous aerosol [186]. In contrast to the previous understanding that the uptake process occurs within the bulk of an aerosol, interfacial processes dominate the uptake process due to facile hydrolysis at the liquid-vapour interface and competitive re-evaporation. This work not only brings new insights to a long-standing questions, but also extends the application of DPs to the liquid-vapour interface. Other examples of DP applications in aqueous systems include TiO_2-water interfaces [187], the ice Ih/XI transition [188], and dynamical states of high-pressure ice VII [189].

3.4. Other systems

Here we briefly list several applications of DPs to other classes of systems including molecular systems, clusters, surfaces, and low-dimensional systems. Jiang *et al* [190] developed DPs

for sulfuric acid-sulfuric acid, dimethylamine-dimethylamine, and sulfuric aciddimethylamine organic molecular systems to investigate the atmospheric aerosol nucleation process. Zeng *et al* [191] trained a DP based on a dataset for the pyrolysis of *n*-dodecane and performed a reactive DP MD simulation to reveal the detailed pyrolysis mechanism, in good agreement with experiment. Chen *et al* [192] used a DP to accurately represent the ground- and excited-state PES of CN$_2$NH. This DP accurately reproduced excited-state topological structures, photo-isomerisation paths, and conical band structure intersections. Yang *et al* [99] first generated training datasets through active learning with enhanced sampling and then developed a DP to study the urea decomposition process in water. Wang *et al* [194] presented a data-driven coarse-grained simulation of polymers in solution and validated the accuracy of this method with DPs to construct a coarse-grained potential. Pan *et al* [195] extended the DP-approach to incorporate external electrostatic potentials in a molecular system; the resultant DP was accurate for energies and forces of representative configurations along the Menshutkin and chorismate mutase reactions pathways.

A study of metal and alloy clusters and surfaces demonstrated a conflict between Al bulk and cluster energies [196]. This indicates that the compromise between properties are, on occasion, necessary and suggest that DPs should be developed for target properties (this is the specialisation discussed above). Andolina *et al* applied a DP to study the nucleation and growth of seeded core–shell Ag and Au clusters [119] and predict the anisotropic surface segregation for Al-Mg alloys [129]. Wang *et al* [103] successfully applied DPs in Ag-Au, Au {111} surface reconstruction and segregation of Au on the Ag-Au nanoalloy surfaces.

Achar *et al* [197] proposed a DP for graphane and showed that it outperforms empirical interatomic potentials for phonon density of states, thermodynamic properties, velocity autocorrelation function, and stress–strain curve up to the yield point. Wu *et al* [198] developed a DP for In$_2$Se$_3$ monolayers and used it to predict a range of thermodynamic properties of In$_2$Se$_3$ polymorphs and lattice dynamics of ferroelectric In$_2$Se$_3$ with *ab initio* accuracy. Chen *et al* [199] applied a DP to simulate the synthesis of amorphous CoFeB during a rapid cooling process. The applications of DPs in low-dimensional systems are in their early stage, but existing evidence suggests that the DP method is promising for the simulation of low-dimensional materials.

4. Accuracy and efficiency of DPs in practice

The summary of recent applications of the DP method in different systems suggests the wide-applicability and high accuracy of DPs. Nonetheless, it is appropriate to return to the competition between DP accuracy vs the efficiency of DP-based simulations. On the accuracy issue, we reflect on what we have learned from the applications performed to-date and our experience, to address: (1) How good is DP? (2) What have we learned? (3) When can we rely on DPs and when can we not? On the efficiency issue, we summarise how

fast DPs are by comparison with other approaches from pair potentials to DFT.

4.1. Accuracy

4.1.1. How good are DPs?. Broadly speaking, DPs are more accurate than other types of empirical interatomic potentials; this was the main reason behind the development of the DP method and its application to a wide range of systems. The improvement in accuracy often leads to 'qualitatively' new results. We return to two examples where DPs are 'qualitatively' better. The first example is metal Ti [101]. Previous experiments and DFT calculations [113] confirmed that the screw ⟨**a**⟩ dislocation is more stable on the Pyramidal I than on the Prism plane. DFT calculations, using pseudopotentials with different numbers of valence electrons [205], showed that the dislocation core energy of the screw ⟨**a**⟩ dislocation on the Pyramidal I plane is 18.4 meV b^{-1} lower than that on the Prism plane. This energy ordering is a prerequisite for the phenomena of screw ⟨**a**⟩ dislocation 'locking' and 'unlocking' observed in experiment at 150 K [113]. While the screw ⟨**a**⟩ dislocation is more stable on the Pyramidal I plane, it has a high energy barrier for glide; hence the screw is 'locked' on this plane. However, the screw ⟨**a**⟩ has a much lower glide barrier on the higher energy Prism plane; hence on this plane, the screw 'unlocks' and glides easily. Unfortunately, all extant empirical Ti potentials predict the incorrect energy ordering; i.e. the ⟨**a**⟩ screw is more stable on the Prism plane (including the generally excellent MEAM empirical potential [109]), in 'qualitative' disagreement with experiment. The powerful representability and flexibility of the DP method allows the Ti DP to reproduce this important feature of defects in Ti and enables new insights through large-scale MD simulations.

Another example of 'qualitative' improvements made possible by a DP is the calculation of the phase diagram of water [95]. TIP4P/2005 is one of the most accurate empirical water models available today for phase diagram prediction [95]. At high temperature and pressure, TIP4P/2005 predicts a first-order transition from ice VII to a plastic phase, in which the BCC oxygen sublattice coexists with freely rotating molecules. This prediction has not been experimentally confirmed. The DP for water predicts a super-ionic ice VII in that region, in agreement with recent experimental observations [206]. DP also better reproduces the phase boundaries between ice VI, VII, and VIII, at high pressures better than TIP4P/2005. These are two of many examples of where the improved quantitative predictions of DP enable 'qualitatively' correct phenomena not accessible through other empirical interatomic potentials.

The accuracy of DP is routinely compared with DFT results especially when benchmarks from widely accepted empirical interatomic potentials do not exist. In most examples (and our own experience), the agreement between energy and force (RMSE) values obtained using a DP and DFT is commonly smaller than 10 meV atom^{-1} and 100 meV Å^{-1}, respectively. In some systems, such as the water phase diagram, the error is typically ∼1 meV molecule^{-1}. This accuracy should meet the

requirements for most applications. In particular, almost half of the examples listed in table 2 focus on non-crystalline systems (liquids, amorphous systems; including liquid metallic alloys, and metallic glasses). DP-based predictions are especially accurate for liquid properties (DFT references) on liquid structures, diffusion coefficients, thermal conductivities.

While the flexibility of DPs is desirable for describing complex potential energy surfaces, the ultimate accuracy of DPs are often limited by the accuracy of the DFT training set. Since DFT is itself an approximation, perfect agreement with nature is not expected. Improvements are possible with improved exchange and correlation DFT functionals; climbing the Jacob's ladder from LDA to GGA to meta-GGA to hybrid-GGA to fully non-local approaches [75]. In practice, the ultimate limit of DP accuracy can often be associated with the DP training sets, the accuracies of which are often associated with the choice of the DFT functionals. This choice is often dictated by the associated computational cost which rises rapidly on climbing the rungs of the Jacob's ladder.

4.1.2. What have we learned?. The two examples, above, demonstrate that DPs can lead to very high accuracy results; comparable to the underlying DFT approaches and higher than the vast majority of empirical interatomic potentials. The examples, above, also show us that DPs can be employed to add new understanding and insight in situations which were previously inaccessible to other computational approaches and experimental observation. The DP approach is also useful in the development of new *ab initio* methods and pseudopotentials [180, 181]. Finally, the examples showed how DPs can be readily specialised to describe phenomena for which general purpose DPs do not suffice.

At early stages of DP development and for simple applications (e.g. simulation of liquid state structure) where the demands on accuracy are not too high, sampling efficiency is not so critical. For such simple applications, an initial DP based on a small set of AIMD trajectory training sets, often leads to efficient and accurate results. However, sampling efficiency is critical when a general-purpose DP is required or target properties are subtle (e.g. where energy differences between phases are very small or some defect properties). AIMD may not adequately/efficiently sample atomic configurations that represent those associated with properties of interest. This may be addressed by starting with the general purposed DP developed through DP-GEN and tweaking it through identification of property-appropriate and incorporation in the specialisation step, as described above. The specialisation process varies with systems and properties of interest and may be viewed as the art of tweaking DPs based upon physical understanding.

4.1.3. When can we rely on a DP?. After performing our normal suite of property testing on a DP (e.g. see [101]), experience shows that such DPs yield reliable results in atomistic simulations—especially compared with DFT calculations that are of insufficient spatial scales or empirical interatomic potentials that are of insufficient accuracy. Critical issues for

all ML potentials are representability and transferability. Representability implies the ability of the functional form to accurately reproduce the target properties. Transferability is the ability of a potential to describe the properties which were not included in the training process.

The DP approach usually performs well from a representability perspective; DPs are usually able to provide fits that adequately represent all of the training datasets. In some cases DP failed to distinguish similar configurations (e.g. the configurations along the transition path of the screw dislocation in BCC W); the representability of the DP can be improved by using more expressive descriptors, such as a three-body embedding descriptor [102]. On the other hand, transferability can be non-trivial and subtle for DPs. Transferability can be classified as *in-distribution* and *out-of-distribution* transferability. In-distribution or out-of-distribution transferability is the ability of model to interpolate within or extrapolate out of the sampled configuration distribution, respectively. The in-distribution transferability of DPs trained with the DP-GEN scheme is generally quite good, providing reliable and accurate predictions of configurations similar to those in the sampled distribution. However, ML potentials will fail in the out-of-distribution transferability where the explored configuration is 'far' from those sampled configurations used in training [34]. A simple example of an out-of-distribution issue is a DP trained using only liquid datasets; such a DP normally shows poor transferability with respect to crystal datasets because there is little overlap between the liquid and solid configuration distributions. For defect properties, we do not know *a priori* whether defect configurations represent an in- or out-of the distribution with respect to those sampled by DP-GEN. Model deviation (equation (31)) serves as a good indicator of the transferability of a DP (without the need for additional DFT calculations). In the case of transferability failure, the DP can be specialised by adding configurations to the training set that more closely represent the configuration of interest; this is converting out-of-distribution transferability failure to in-distribution transferability agreement.

In the cases when a user is not confident whether a DP is transferable, the DP can be used in conjunction with model deviation after validating against relevant DFT or experimental benchmarks. DPs are, like other empirical potentials subject to the adages that interatomic potentials 'will work only before they fail' or 'will work until they do not'. From this perspective, the replacement of DFT by ML potentials including DP are not completely reliable; DFT will remain the method of choice where very high accuracy property prediction is necessary. The combination of DFT and DP provides a practical strategy for the needs of the materials science community, providing the implicit trade-offs between accuracy and computational efficiency.

4.2. Efficiency

Using the previous DP application examples, we see that (1) on CPUs, compressed DPs are faster than DFT by a factor of over 10^6 and slower than empirical interatomic potentials such as EAM (MEAM) by $\sim$100 (10) times; (2) on GPUs,

DP compression model is slower than potentials like EAM by, of order 10 times. Of course, the actual efficiency is application dependent (especially for comparison with DFT). Compressed DPs can be faster than the original DPs by a factor of over 10 and consume an order of magnitude less memory. Additional optimisation of the neural networks at the heart of DPs is possible by optimisation of different operators on the computational graph and through hardware changes [107]. Both DPs and empirical potentials show linear scaling with the number of atoms on both CPU and GPU machines. This linear scaling is the enabler of large-scale atomistic simulations. The lower speed of DPs, compared with empirical potentials, is reasonable considering the vast number of parameters in DPs (often at the order of 10^5). Because DPs are and will continue to be slower than empirical potentials, simpler, empirical potentials will continue to play an important role in materials science. The 'competition' between MLs and simpler, empirical potentials drives the continuous improvement of each. The ML potential community continues to focus on improving the potential efficiency (computational speed), while empirical potential development continues to develop new formalism increasing accuracy. The concept of 'ML potentials guided by the physics in the empirical potentials' is also an exciting area; e.g. see the recent work of Mishin *et al* [31, 207].

5. Conclusions and outlook

With increasing need for atomistic simulations with higher accuracy, larger length scales, longer time scales and computational efficiency, ML-based interatomic potentials are rapidly gaining acceptance in the broad materials science community. This is especially true in areas where the phenomena of interest are subtle and those in which the material system is complex. In this review, we examined the DP approach (for ML potentials); summarising the basic theory, how to develop DPs and apply DP-development software and database, how to make DPs more efficient in applications, how to specialise DPs for subtle application, reviewed several DP applications, and discussed DP accuracy and efficiency. After several years of evolution, the DP method is now relatively mature, yet continuing to improve in both accuracy and efficiency within an open-source community framework. We envision the DP method to continue developing in the coming years and the continued expansion of the data base of useable DPs.

Continued development will likely proceed along several avenues. The first is the development of new and more intelligent descriptors for better predictability. We see from the W example [102] that the Peierls barrier (the barrier for dislocation glide) can only be accurately reproduced by expanding the DP descriptor to include three-body embedding. We suspect that such examples will continue to arise as DPs are expanded to include a broader set of applications in different materials. Another issue is related to magnetism; how can magnetic moment degrees of freedom be incorporated into DPs? Empirical EAM and MEAM potentials deal with this issue through a set of assumptions and approximations. More

intelligent descriptors also improve the ease of DP training. For example, questions arise for the current hybrid descriptors, with two-body and three-body embeddings about how much to weight three-body embeddings in the hybrid descriptors. Current strategies in this area tend to be based largely on empirical experience; hence, there are opportunities to transform this into a machine-driven process.

An important second area of development will be improvement of the automation of DP training and specialisation. In DeePMD-kit and DP-GEN software, different settings do, on occasion, influence the performance of trained DPs. Although we presented our experience on choosing these settings in section 2.3, ideally this experience should be replaced in future generations of the DeePMD-kit and DP-GEN software. The automatic selection of the trust levels in DP-GEN is already in the testing stage. The specialisation step should also be more automated to reduce user intervention in determining: (1) what types of specialisation datasets are needed? (2) how many specialisation datasets are needed to combine with DP-GEN datasets? (3) when to include specialisation datasets and how to modify DeePMD-kit and DP-GEN settings? More automated training and specialisation schemes would accelerate the development of new DPs for more systems and applications.

A third area of DP future development is further optimisation of the computational speed of DP. Currently, many empirical potentials are faster than DP (at least a factor of 10) which leads potential users to prefer empirical potentials when accuracy demands are not high. In principle, DPs cannot be faster than empirical potentials considering the vast number of parameters involved, but decreasing the computation efficiency loss in using a DP can change the speed—accuracy tradeoff and enable more effective material simulations. As seen above, the greatly improved accuracy of DPs over most empirical potentials opens the door to applications where empirical potentials are simply 'qualitatively' incorrect.

A fourth area is the combination of DP and transformer [208]. Transformers were introduced in 2017 and are now commonly used in the fields of natural language processing (NLP) and computer vision. Large pre-trained models based on the transformer architecture (e.g. BERT [209] and GPT-3 [210]) are changing the research paradigm of NLP. Traditionally, models for a specific task are trained from scratch. The Transformer approach is based upon fine-tuning pre-trained models such that new records are being established for specific tasks [211]. For DP, pre-trained models are trained using a large database based upon different elements in the periodic table; learning the embedding mapping from the element type to the abstract space. For example, when we have training datasets for binary systems and want to train a DP model for a ternary system, the traditional step would be using DPGEN to generate datasets for ternary system from scratch. However, with the introduction of pre-trained binary system DP models, the number of additional datasets required to train a ternary system DP model is greatly decreased. From this perspective, pre-trained models can accelerate the training process and decrease the computational time required for generating the requisite training datasets.

Finally, the open-source DP Library (a database for DP including training datasets, training schemes, DP, and testing results) should be greatly expanded to include most of the periodic table, figure 6 and alloys. This will be an on-going effort requiring contributions from the entire user group. Another task is to make the DP Library easier to use. Both of these will be enabled by improvements in the openness of scientific computing community (e.g. appropriate acknowledgment of contributions).

The development of the DP approach (and related ML potential approaches) represents an important milestone for the field of atomistic simulations of materials that rests on advances in ML technology and descriptors of atomic environments. DPs routinely provide high (near DFT) accuracy with reasonable computational efficiency, as compared with empirical potentials. The accuracy and efficiency of DPs open the door to 'qualitatively' new applications of atomistic simulation.

Acknowledgments

ORCID iD

Tongqi Wen ⓘ https://orcid.org/0000-0003-1692-2512

References

[1] Hafner J 2000 Atomic-scale computational materials science *Acta Mater.* **48** 71

[2] Born M and Oppenheimer R 1927 Zur quantentheorie der molekeln *Ann. Phys., Lpz.* **389** 457

[3] Dirac P A M 1929 Quantum mechanics of many-electron systems *Proc. R. Soc.* A **123** 714

[4] Kohn W and Sham L J 1965 Self-consistent equations including exchange and correlation effects *Phys. Rev.* **140** A1133

[5] Verlet L 1967 Computer "experiments" on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules *Phys. Rev.* **159** 98

[6] Zwanzig R W 1954 High-temperature equation of state by a perturbation method. I. Nonpolar gases *J. Chem. Phys.* **22** 1420

[7] Tersoff J 1989 Modeling solid-state chemistry: interatomic potentials for multicomponent systems *Phys. Rev.* B **39** 5566

[8] Vink R, Barkema G, van der Weg W and Mousseau N 2001 Fitting the Stillinger-Weber potential to amorphous silicon *J. Non-Cryst. Solids* **282** 248

[9] Daw M S and Baskes M I 1984 Embedded-atom method: derivation and application to impurities, surfaces and other defects in metals *Phys. Rev.* B **29** 6443

[10] Baskes M I 1992 Modified embedded-atom potentials for cubic materials and impurities *Phys. Rev.* B **46** 2727

[11] Prentice J C A *et al* 2020 The ONETEP linear-scaling density functional theory program *J. Chem. Phys.* **152** 174111

[12] Hacene M, Anciaux-Sedrakian A, Rozanska X, Klahr D, Guignon T and Fleurat-Lessard P 2012 Accelerating VASP electronic structure calculations using graphic processing units *J. Comput. Chem.* **33** 2581

[13] Hutchinson M and Widom M 2012 VASP on a GPU: application to exact-exchange calculations of the stability of elemental boron *Comput. Phys. Commun.* **183** 1422

[14] Jia W, Cao Z, Wang L, Fu J, Chi X, Gao W and Wang L-W 2013 The analysis of a plane wave pseudopotential density functional theory code on a GPU machine *Comput. Phys. Commun.* **184** 9

[15] Jia W, Fu J, Cao Z, Wang L, Chi X, Gao W and Wang L-W 2013 Fast plane wave density functional theory molecular dynamics calculations on multi-GPU machines *J. Comput. Phys.* **251** 102

[16] Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)

[17] Jordan M I and Mitchell T M 2015 Machine learning: trends, perspectives and prospects *Science* **349** 255

[18] Mahesh B 2020 Machine learning algorithms-a review *Int. J. Sci. Res.* **9** 381

[19] Blank T B, Brown S D, Calhoun A W and Doren D J 1995 Neural network models of potential energy surfaces *J. Chem. Phys.* **103** 4129

[20] Behler J and Parrinello M 2007 Generalized neural-network representation of high-dimensional potential-energy surfaces *Phys. Rev. Lett.* **98** 146401

[21] Khaliullin R Z, Eshet H, Kühne T D, Behler J and Parrinello M 2011 Nucleation mechanism for the direct graphite-to-diamond phase transition *Nat. Mater.* **10** 693

[22] Artrith N and Urban A 2016 An implementation of artificial neural-network potentials for atomistic materials simulations: performance for TiO_2 *Comput. Mater. Sci.* **114** 135

[23] Behler J 2021 Four generations of high-dimensional neural network potentials *Chem. Rev.* **121** 10037

[24] Behler J 2016 Perspective: machine learning potentials for atomistic simulations *J. Chem. Phys.* **145** 170901

[25] Behler J 2017 First principles neural network potentials for reactive simulations of large molecular and condensed systems *Angew. Chem., Int. Ed.* **56** 12828

[26] Schütt K T, Sauceda H E, Kindermans P-J, Tkatchenko A and Müller K-R 2018 SchNet—a deep learning architecture for molecules and materials *J. Chem. Phys.* **148** 241722

[27] Schütt K T, Kessel P, Gastegger M, Nicoli K A, Tkatchenko A and Müller K-R 2019 SchNetPack: a deep learning toolbox for atomistic systems *J. Chem. Theory Comput.* **15** 448

[28] Ghasemi S A, Hofstetter A, Saha S and Goedecker S 2015 Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network *Phys. Rev.* B **92** 045131

[29] Hy T S, Trivedi S, Pan H, Anderson B M and Kondor R 2018 Predicting molecular properties with covariant compositional networks *J. Chem. Phys.* **148** 241745

[30] Unke O T and Meuwly M 2019 Physnet: a neural network for predicting energies, forces, dipole moments and partial charges *J. Chem. Theory Comput.* **15** 3678

[31] Purja Pun G P, Batra R, Ramprasad R and Mishin Y 2019 Physically informed artificial neural networks for atomistic modeling of materials *Nat. Commun.* **10** 2339

[32] Bartók A P, Payne M C, Kondor R and Csányi G 2010 Gaussian approximation potentials: the accuracy of

quantum mechanics, without the electrons *Phys. Rev. Lett.* **104** 136403

[33] Dragoni D, Daff T D, Csányi G and Marzari N 2018 Achieving DFT accuracy with a machine-learning interatomic potential: thermomechanics and defects in bcc ferromagnetic iron *Phys. Rev. Mater.* **2** 013808

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

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

[36] Shapeev A V 2016 Moment tensor potentials: a class of systematically improvable interatomic potentials *Multiscale Model. Simul.* **14** 1153

[37] Podryabinkin E V and Shapeev A V 2017 Active learning of linearly parametrized interatomic potentials *Comput. Mater. Sci.* **140** 171

[38] Podryabinkin E V, Tikhonov E V, Shapeev A V and Oganov A R 2019 Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning *Phys. Rev.* B **99** 064114

[39] Chen C, Deng Z, Tran R, Tang H, Chu I-H and Ong S P 2017 Accurate force field for molybdenum by machine learning large materials data *Phys. Rev. Mater.* **1** 043603

[40] Li X-G, Hu C, Chen C, Deng Z, Luo J and Ong S P 2018 Quantum-accurate spectral neighbor analysis potential models for Ni-Mo binary alloys and fcc metals *Phys. Rev.* B **98** 094104

[41] Deng Z, Chen C, Li X-G and Ong S P 2019 An electrostatic spectral neighbor analysis potential for lithium nitride *npj Comput. Mater.* **5** 75

[42] Sauceda H E, Chmiela S, Poltavsky I, Müller K-R and Tkatchenko A 2019 Molecular force fields with gradient-domain machine learning: construction and application to dynamics of small molecules with coupled cluster forces *J. Chem. Phys.* **150** 114102

[43] Chmiela S, Sauceda H E, Poltavsky I, Müller K-R and Tkatchenko A 2019 sGDML: constructing accurate and data efficient molecular force fields using machine learning *Comput. Phys. Commun.* **240** 38

[44] Unke O T, Chmiela S, Sauceda H E, Gastegger M, Poltavsky I, Schütt K T, Tkatchenko A and Müller K-R 2021 Machine learning force fields *Chem. Rev.* **121** 10142

[45] Zuo Y *et al* 2020 Performance and cost assessment of machine learning interatomic potentials *J. Phys. Chem.* A **124** 731

[46] Han J, Zhang L, Car R and E W 2017 Deep potential: a general representation of a many-body potential energy surface (arXiv:1707.01478 [physics.comp-ph])

[47] Han J, Zhang L, Car R and E W 2018 Deep potential: a general representation of a many-body potential energy surface *Commun. Comput. Phys.* **23** 629

[48] Zhang L, Han J, Wang H, Car R and E W 2018 Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics *Phys. Rev. Lett.* **120** 143001

[49] Wang H, Zhang L, Han J and E W 2018 Deepmd-kit: a deep learning package for many-body potential energy representation and molecular dynamics *Comput. Phys. Commun.* **228** 178

[50] Jia W, Wang H, Chen M, Lu D, Lin L, Car R, E W and Zhang L 2020 Pushing the limit of molecular dynamics with *ab initio* accuracy to 100 million atoms with machine learning *SC20: Int. Conf. for High Performance Computing, Networking, Storage and Analysis* pp 1–14

[51] Artrith N, Morawietz T and Behler J 2011 High-dimensional neural-network potentials for multicomponent systems: applications to zinc oxide *Phys. Rev.* B **83** 153101

[52] Bereau T, Andrienko D and von Lilienfeld O A 2015 Transferable atomic multipole machine learning models for small organic molecules *J. Chem. Theory Comput.* **11** 3225

[53] Bereau T, DiStasio R A, Tkatchenko A and von Lilienfeld O A 2018 Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning *J. Chem. Phys.* **148** 241706

[54] Nebgen B, Lubbers N, Smith J S, Sifain A, Lokhov A, Isayev O, Roitberg A, Barros K and Tretiak S 2018 Transferable molecular charge assignment using deep neural networks (arXiv:1803.04395 [physics.chem-ph])

[55] Sifain A E, Lubbers N, Nebgen B T, Smith J S, Lokhov A Y, Isayev O, Roitberg A E, Barros K and Tretiak S 2018 Discovering a transferable charge assignment model using machine learning *J. Phys. Chem. Lett.* **9** 4495

[56] Ko T W, Finkler J A, Goedecker S and Behler J 2021 A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer *Nat. Commun.* **12** 398

[57] Ko T W, Finkler J A, Goedecker S and Behler J 2021 General-purpose machine learning potentials capturing nonlocal charge transfer *Acc. Chem. Res.* **54** 808

[58] Grisafi A and Ceriotti M 2019 Incorporating long-range physics in atomic-scale machine learning *J. Chem. Phys.* **151** 204105

[59] Grisafi A, Nigam J and Ceriotti M 2021 Multi-scale approach for the prediction of atomic scale properties *Chem. Sci.* **12** 2078

[60] Frenkel D and Smit B 2002 *Understanding Molecular Simulation From Algorithms to Applications* (New York: Academic)

[61] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. Conf. on Computer Vision and Pattern Recognition (CVPR)*

[62] Barron A 1993 Universal approximation bounds for superpositions of a sigmoidal function *IEEE Trans. Inf. Theory* **39** 930

[63] Barron A R 1994 Approximation and estimation bounds for artificial neural networks *Mach. Learn.* **14** 115

[64] Liang S and Srikant R 2017 Why deep neural networks for function approximation? (arXiv:1610.04161 [cs.LG])

[65] Telgarsky M 2016 benefits of depth in neural networks *29th Conf. on Learning Theory (Proc. Machine Learning Research) (PMLR)* vol 49, ed V Feldman, A Rakhlin and O Shamir (New York: Columbia University) pp 1517–39

[66] Yarotsky D 2017 Error bounds for approximations with deep relu networks *Neural Netw.* **94** 103

[67] Lu J, Shen Z, Yang H and Zhang S 2021 Deep network approximation for smooth functions *SIAM J. Math. Anal.* **53** 5465

[68] E W, Ma C and Wu L 2019 *A priori* estimates of the population risk for two-layer neural networks *Commun. Math. Sci.* **17** 1407–25

[69] E W, Ma C and Wu L 2022 The barron space and the flow-induced function spaces for neural network models *Constructive Approx.* **55** 369–406

[70] Zhang L, Han J, Wang H, Saidi W, Car R and E W 2018 End-to-end symmetry preserving inter-atomic potential energy model for finite and extended systems *Advances in Neural Information Processing Systems* vol 31, ed S Bengio, H Wallach, H Larochelle, K Grauman, N Cesa-Bianchi and R Garnett (Curran Associates, Inc.) pp 4436–46

[71] Kresse G and Furthmüller J 1996 Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set *Comput. Mater. Sci.* **6** 15

[72] Kresse G and Furthmüller J 1996 Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set *Phys. Rev.* B **54** 11169

[73] Giannozzi P *et al* 2009 QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials *J. Phys.: Condens. Matter.* **21** 395502

[74] Chen M, Guo G-C and He L 2010 Systematically improvable optimized atomic basis sets for *ab initio* calculations *J. Phys.: Condens. Matter.* **22** 445501

[75] Perdew J P and Schmidt K 2001 Jacob's ladder of density functional approximations for the exchange-correlation energy *Conf. Proc.* vol 577 p 1

[76] Møller C and Plesset M S 1934 Note on an approximation treatment for many-electron systems *Phys. Rev.* **46** 618

[77] Čížek J 1996 On the correlation problem in atomic and molecular systems. Calculation of wavefunction components in ursell-type expansion using quantum-field theoretical methods *J. Chem. Phys.* **45** 4256

[78] Fano U 1961 Effects of configuration interaction on intensities and phase shifts *Phys. Rev.* **124** 1866

[79] Deaven D M and Ho K M 1995 Molecular geometry optimization with a genetic algorithm *Phys. Rev. Lett.* **75** 288

[80] Glass C W, Oganov A R and Hansen N 2006 USPEX-evolutionary crystal structure prediction *Comput. Phys. Commun.* **175** 713

[81] Laio A and Parrinello M 2002 Escaping free-energy minima *Proc. Natl Acad. Sci.* **99** 12562

[82] Cohn D, Atlas L and Ladner R 1994 Improving generalization with active learning *Mach. Learn.* **15** 201

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

[84] Zhang Y, Wang H, Chen W, Zeng J, Zhang L, Wang H and E W 2020 DP-GEN: a concurrent learning platform for the generation of reliable deep learning based potential energy models *Comput. Phys. Commun.* **253** 107206

[85] Plimpton S 1995 Fast parallel algorithms for short-range molecular dynamics *J. Comput. Phys.* **117** 1

[86] Larsen A H *et al* 2017 The atomic simulation environment—a python library for working with atoms *J. Phys.: Condens. Matter.* **29** 273002

[87] Ceriotti M, More J and Manolopoulos D E 2014 i-PI: a python interface for *ab initio* path integral molecular dynamics simulations *Comput. Phys. Commun.* **185** 1019

[88] Van Der Spoel D, Lindahl E, Hess B, Groenhof G, Mark A E and Berendsen H J C 2005 Gromacs: fast, flexible and free *J. Comput. Chem.* **26** 1701

[89] Gaussian 16 Revision C.01 2016 Gaussian Inc. Wallingford CT

[90] Soler J M, Artacho E, Gale J D, García A, Junquera J, Ordejón P and Sánchez-Portal D 2002 The SIESTA method for *ab initio* order-N materials simulation *J. Phys.: Condens. Matter.* **14** 2745

[91] Kühne T D *et al* 2020 Cp2k: an electronic structure and molecular dynamics software package—quickstep: efficient and accurate electronic structure calculations *J. Chem. Phys.* **152** 194103

[92] Blum V, Gehrke R, Hanke F, Havu P, Havu V, Ren X, Reuter K and Scheffler M 2009 *Ab initio* molecular simulations with numeric atom-centered orbitals *Comput. Phys. Commun.* **180** 2175

[93] Case D A, Cheatham III T E, Darden T, Gohlke H, Luo R, Merz Jr. K M, Onufriev A, Simmerling C, Wang B and Woods R J 2005 The amber biomolecular simulation programs *J. Comput. Chem.* **26** 1668

[94] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (software available from www.tensorflow.org/)

[95] Zhang L, Wang H, Car R and E W 2021 Phase diagram of a deep potential water model *Phys. Rev. Lett.* **126** 236001

[96] Schimka L, Gaudoin R, Klimeš J, Marsman M and Kresse G 2013 Lattice constants and cohesive energies of alkali, alkaline-earth and transition metals: random phase approximation and density functional theory results *Phys. Rev.* B **87** 214102

[97] Kittel C 2005 *Introduction to Solid State Physics* 8th edn (New York: Wiley)

[98] Yang M, Karmakar T and Parrinello M 2021 Liquid-liquid critical point in phosphorus *Phys. Rev. Lett.* **127** 080603

[99] Yang M, Bonati L, Polino D and Parrinello M 2022 Using metadynamics to build neural network potentials for reactive events: the case of urea decomposition in water *Catal. Today* **387** 143

[100] Jiang W, Zhang Y, Zhang L and Wang H 2021 Accurate Deep Potential model for the Al-Cu-Mg alloy in the full concentration space *Chin. Phys.* B **30** 050706

[101] Wen T, Wang R, Zhu L, Zhang L, Wang H, Srolovitz D J and Wu Z 2021 Specialising neural network potentials for accurate properties and application to the mechanical response of titanium *npj Comput. Mater.* **7** 206

[102] Wang X, Wang Y, Zhang L, Dai F and Wang H 2021 A tungsten deep potential with high accuracy and generalization ability based on a newly designed three-body embedding formalism (arXiv:2111.04281 [cond-mat.mtrl-sci])

[103] Wang Y, Zhang L, Xu B, Wang X and Wang H 2022 A generalizable machine learning potential of Ag-Au nanoalloys and its application to surface reconstruction, segregation and diffusion *Modelling Simul. Mater. Sci. Eng.* **30** 025003

[104] Fu B, Sun Y, Zhang L, Wang H and Xu B 2021 Deep learning inter-atomic potential for thermal and phonon behaviour of silicon carbide with quantum accuracy (arXiv:2110.10843 [cond-mat.mtrl-sci])

[105] Huang J, Zhang L, Wang H, Zhao J, Cheng J and E W 2021 Deep potential generation scheme and simulation protocol for the $Li_{1}0GeP_{2}S_{1}2$-type superionic conductors *J. Chem. Phys.* **154** 094703

[106] Lu D, Wang H, Chen M, Lin L, Car R, E W, Jia W and Zhang L 2021 86 PFLOPS deep potential molecular dynamics simulation of 100 million atoms with *ab initio* accuracy *Comput. Phys. Commun.* **259** 107624

[107] Lu D, Jiang W, Chen Y, Zhang L, Jia W, Wang H and Chen M 2021 DP train, then DP compress: model compression in deep potential molecular dynamics (arXiv:2107.02103 [physics.comp-ph])

[108] Mendelev M, Underwood T and Ackland G 2016 Development of an interatomic potential for the simulation of defects, plasticity and phase transformations in titanium *J. Chem. Phys.* **145** 154102

[109] Hennig R, Lenosky T, Trinkle D, Rudin S and Wilkins J W 2008 Classical potential describes martensitic phase transformations between the α, β and ω titanium phases *Phys. Rev.* B **78** 054121

[110] Vítek V 1968 Intrinsic stacking faults in body-centred cubic crystals *Phil. Mag.* **18** 773

[111] Ko W-S, Grabowski B and Neugebauer J 2015 Development and application of a Ni-Ti interatomic potential with high predictive accuracy of the martensitic phase transition *Phys. Rev.* B **92** 134107

[112] Dickel D, Barrett C, Carino R, Baskes M and Horstemeyer M 2018 Mechanical instabilities in the modeling of phase transitions of titanium *Modelling Simul. Mater. Sci. Eng.* **26** 065002

[113] Clouet E, Caillard D, Chaari N, Onimus F and Rodney D 2015 Dislocation locking versus easy glide in titanium and zirconium *Nat. Mater.* **14** 931

[114] Wang H, Guo X, Zhang L, Wang H and Xue J 2019 Deep learning inter-atomic potential model for accurate irradiation damage simulations *Appl. Phys. Lett.* **114** 244101

[115] Zeng Q, Yu X, Yao Y, Gao T, Chen B, Zhang S, Kang D, Wang H and Dai J 2021 *Ab initio* validation on the connection between atomistic and hydrodynamic description to unravel the ion dynamics of warm dense matter *Phys. Rev. Res.* **3** 033116

[116] Liu Q, Lu D and Chen M 2020 Structure and dynamics of warm dense aluminum: a molecular dynamics study with density functional theory and deep potential *J. Phys.: Condens. Matter.* **32** 144002

[117] Liu Q, Li J and Chen M 2021 Thermal transport by electrons and ions in warm dense aluminum: a combined density functional theory and deep potential study *Matter Radiat. Extremes* **6** 026902

[118] Cheng Y *et al* 2021 Deep-learning potential method to simulate shear viscosity of liquid aluminum at high temperature and high pressure by molecular dynamics *AIP Adv.* **11** 015043

[119] Andolina C M, Bon M, Passerone D and Saidi W A 2021 Robust, multi-length-scale, machine learning potential for Ag-Au bimetallic alloys from clusters to bulk materials *J. Phys. Chem.* C **125** 17438

[120] Chen B, Zeng Q, Wang H, Zhang S, Kang D, Lu D and Dai J 2021 Atomistic mechanism of phase transition in shock compressed gold revealed by deep potential (arXiv:2006.13136 [cond-mat.mtrl-sci])

[121] Jiao J 2021 Self-healing mechanism of lithium in lithium metal batteries (arXiv:2106.10979 [cond-mat.mtrl-sci])

[122] Zhang Y, Gao C, Liu Q, Zhang L, Wang H and Chen M 2020 Warm dense matter simulation via electron temperature dependent deep potential molecular dynamics *Phys. Plasmas* **27** 122704

[123] Niu H, Bonati L, Piaggi P M and Parrinello M 2020 *Ab initio* phase diagram and nucleation of gallium *Nat. Commun.* **11** 1

[124] Shi M, Li J, Tao M, Zhang X and Liu J 2021 Artificial intelligence model for efficient simulation of monatomic phase change material antimony *Mater. Sci. Semicond. Process.* **136** 106146

[125] Wang J, Shen H, Yang R, Xie K, Zhang C, Chen L, Ho K-M, Wang C-Z and Wang S 2022 A deep learning interatomic potential developed for atomistic simulation of carbon materials *Carbon* **186** 1

[126] Bonati L and Parrinello M 2018 Silicon liquid structure and crystal nucleation from *ab Initio* deep metadynamics *Phys. Rev. Lett.* **121** 265701

[127] Li R, Lee E and Luo T 2020 A unified deep neural network potential capable of predicting thermal conductivity of silicon in different phases *Mater. Today Phys.* **12** 100181

[128] Wang H, Zhang Y, Zhang L and Wang H 2020 Crystal structure prediction of binary alloys via deep potential *Front. Chem.* **8** 895

[129] Andolina C M, Wright J G, Das N and Saidi W A 2021 Improved Al-Mg alloy surface segregation predictions with a machine learning atomistic potential *Phys. Rev. Mater.* **5** 083804

[130] Bourgeois L, Zhang Y, Zhang Z, Chen Y and Medhekar N V 2020 Transforming solid-state precipitates via excess vacancies *Nat. Commun.* **11** 1

[131] Cheng B, Zhao X, Zhang Y, Chen H, Polmear I and Nie J-F 2020 Co-segregation of Mg and Zn atoms at the planar η_1-precipitate/Al matrix interface in an aged Al-Zn-Mg alloy *Scr. Mater.* **185** 51

[132] Ryltsev R E and Chtchelkatchev N M 2021 Deep machine learning potentials for multicomponent metallic melts: development, predictability and compositional transferability (arXiv:2110.14006 [cond-mat.mtrl-sci])

[133] Wen T *et al* 2019 Development of a deep machine learning interatomic potential for metalloid-containing Pd-Si compounds *Phys. Rev.* B **100** 174101

[134] Wang Q, Zhai B, Wang H P and Wei B 2021 Atomic structure of liquid refractory Nb_5Si_3 intermetallic compound alloy based upon deep neural network potential *J. Appl. Phys.* **130** 185103

[135] Guo Y *et al* 2019 Bergman-type medium range order in amorphous Zr77Rh23 alloy studied by *ab initio* molecular dynamics simulations *J. Alloys Compd.* **790** 675

[136] Guo Y *et al* 2019 Temperature dependence of structural, dynamical and electronic properties of amorphous Bi_2Te_3: an *ab initio* study *New J. Phys.* **21** 093062

[137] Tang L, Yang Z J, Wen T Q, Ho K M, Kramer M J and Wang C Z 2020 Development of interatomic potential for Al-Tb alloys using a deep neural network learning method *Phys. Chem. Chem. Phys.* **22** 18467

[138] Tang L, Yang Z, Wen T, Ho K M, Kramer M J and Wang C Z 2021 Short- and medium-range orders in Al90Tb10 glass and their relation to the structures of competing crystalline phases *Acta Mater.* **204** 116513

[139] Han I, McKeown J T, Tang L, Wang C-Z, Parsamehr H, Xi Z, Lu Y-R, Kramer M J and Shahani A J 2020 Dynamic observation of dendritic quasicrystal growth upon laser-induced solid-state transformation *Phys. Rev. Lett.* **125** 195503

[140] Tang L, Ho K M and Wang C Z 2021 Molecular dynamics simulation of metallic Al-Ce liquids using a neural network machine learning interatomic potential *J. Chem. Phys.* **155** 194503

[141] Daniels C L, Liu D-J, Adamson M A S, Knobeloch M and Vela J 2021 Azo(xy) vs aniline selectivity in catalytic nitroarene reduction by intermetallics: experiments and simulations *J. Phys. Chem.* C **125** 24440

[142] Zhang C, Sun Y, Wang H-D, Zhang F, Wen T-Q, Ho K-M and Wang C-Z 2021 Crystallization of the P_3Sn_4 phase upon cooling P_2Sn_5 liquid by molecular dynamics simulation using a machine learning interatomic potential *J. Phys. Chem.* C **125** 3127

[143] Balyakin I A, Rempel S V, Ryltsev R E and Rempel A A 2020 Deep machine learning interatomic potential for liquid silica *Phys. Rev.* E **102** 052125

[144] Deng J and Stixrude L 2021 Thermal conductivity of silicate liquid determined by machine learning potentials *Geophys. Res. Lett.* **48** e2021GL093806

[145] Luo H, Karki B B, Ghosh D B and Bao H 2021 Anomalous behavior of viscosity and electrical conductivity of $MgSiO_3$ melt at mantle conditions *Geophys. Res. Lett.* **48** e2021GL093573

[146] Luo H, Karki B B, Ghosh D B and Bao H 2021 Deep neural network potentials for diffusional lithium isotope fractionation in silicate melts *Geochim. Cosmochim. Acta* **303** 38

[147] Chen W and Li L-S 2021 The study of the optical phonon frequency of 3C-SiC by molecular dynamics simulations with deep neural network potential *J. Appl. Phys.* **129** 244104

[148] An Q 2021 Mitigating amorphization in superhard boron carbide by microalloying-induced stacking fault formation *Phys. Rev. Mater.* **5** 103602

[149] Rodriguez A, Lam S and Hu M 2021 Thermodynamic and transport properties of LiF and FLiBe molten salts with deep learning potentials *ACS Appl. Mater. Interfaces* **13** 55367–79

[150] Liang W, Lu G and Yu J 2021 Theoretical prediction on the local structure and transport properties of molten alkali chlorides by deep potentials *J. Mater. Sci. Technol.* **75** 78

[151] Liang W, Lu G and Yu J 2020 Molecular dynamics
    simulations of molten magnesium chloride using
    machine-learning-based deep potential *Adv. Theory Simul.*
    3 2000180

[152] Pan G, Chen P, Yan H and Lu Y 2020 A DFT accurate
    machine learning description of molten ZnCl2 and its
    mixtures: 1. Potential development and properties
    prediction of molten ZnCl2 *Comput. Mater. Sci.*
    185 109955

[153] Pan G, Ding J, Du Y, Lee D-J and Lu Y 2021 A DFT
    accurate machine learning description of molten ZnCl2
    and its mixtures: 2. Potential development and properties
    prediction of ZnCl2-NaCl-KCl ternary salt for CSP
    Comput. Mater. Sci. **187** 110055

[154] Liang W, Lu G and Yu J 2021 Machine-learning-driven
    simulations on microstructure and thermophysical
    properties of MgCl2-KCl eutectic *ACS Appl. Mater.
    Interfaces* **13** 4034

[155] Bu M, Liang W, Lu G and Yu J 2021 Local structure
    elucidation and properties prediction on KCl-CaCl2
    molten salt: a deep potential molecular dynamics study
    Sol. Energy Mater. Sol. Cells **232** 111346

[156] Zhao J, Liang W and Lu G 2021 Theoretical prediction on
    the redox potentials of rare-earth ions by deep potentials
    Ionics **27** 2079

[157] Zhang J, Fuller J and An Q 2021 Coordination and
    thermophysical properties of transition metal
    chlorocomplexes in LiCl-KCl eutectic *J. Phys. Chem.* B
    125 8876

[158] Xu N, Shi Y, He Y and Shao Q 2020 A deep-learning
    potential for crystalline and amorphous Li-Si alloys *J.
    Phys. Chem.* C **124** 16278

[159] Marcolongo A, Binninger T, Zipoli F and Laino T 2019
    Simulating diffusion properties of solid-state electrolytes
    via a neural network potential: performance and training
    scheme (arXiv:1910.10090 [physics.comp-ph])

[160] Gupta M K, Ding J, Osti N C, Abernathy D L, Arnold W,
    Wang H, Hood Z and Delaire O 2021 Fast Na diffusion
    and anharmonic phonon dynamics in superionic Na$_3$PS$_4$
    Energy Environ. Sci. **14** 6554

[161] Li H-X, Zhou X-Y, Wang Y-C and Jiang H 2021 Theoretical
    study of Na+ transport in the solid-state electrolyte
    Na$_3$OBr based on deep potential molecular dynamics
    Inorg. Chem. Front. **8** 425

[162] Lin M, Liu X, Xiang Y, Wang F, Liu Y, Fu R, Cheng J and
    Yang Y 2021 Unravelling the fast alkali-ion dynamics in
    paramagnetic battery materials combined with NMR and
    deep-potential molecular dynamics simulation *Angew.
    Chem., Int. Ed.* **60** 12547

[163] Calegari Andrade M F and Selloni A 2020 Structure of
    disordered TiO$_2$ phases from *ab initio* based deep neural
    network simulations *Phys. Rev. Mater.* **4** 113803

[164] Li R, Liu Z, Rohskopf A, Gordiz K, Henry A, Lee E and
    Luo T 2020 A deep neural network interatomic potential
    for studying thermal conductivity of β-Ga$_2$O$_3$ *Appl. Phys.
    Lett.* **117** 152102

[165] Wu J, Zhang Y, Zhang L and Liu S 2021 Deep learning of
    accurate force field of ferroelectric HfO$_2$ *Phys. Rev.* B
    103 024108

[166] Balyakin I and Sadovnikov S 2022 Deep learning potential
    for superionic phase of Ag$_2$S *Comput. Mater. Sci.*
    202 110963

[167] Wang H, Guo X and Xue J 2020 Deep-learning interatomic
    potential for irradiation damage simulations in MoS$_2$ with
    ab initial accuracy (arXiv:2010.09547 [cond-mat.mtrl-sci])

[168] Guo D, Li C, Li K, Shao B, Chen D, Ma Y, Sun J, Cao X,
    Zeng W and Chang X 2021 The thermoelectric
    performance of new structure SnSe studied by quotient

graph and deep learning potential *Mater. Today Energy*
    20 100665

[169] Dai F-Z, Wen B, Xiang H and Zhou Y 2020 Grain boundary
    strengthening in ZrB2 by segregation of W: atomistic
    simulations with deep learning potential *J. Eur. Ceram.
    Soc.* **40** 5029

[170] Dai F-Z, Wen B, Sun Y, Xiang H and Zhou Y 2020
    Theoretical prediction on thermal and mechanical
    properties of high entropy (Zr0.2Hf0.2Ti0.2Nb0.2Ta0.2)C
    by deep learning potential *J. Mater. Sci. Technol.*
    43 168

[171] Dai F-Z, Sun Y, Wen B, Xiang H and Zhou Y 2021
    Temperature dependent thermal and elastic properties of
    high entropy (Ti0.2Zr0.2Hf0.2Nb0.2Ta0.2)B2: molecular
    dynamics simulation by deep learning potential *J. Mater.
    Sci. Technol.* **72** 8

[172] Ko H-Y, Zhang L, Santra B, Wang H, E W, DiStasio Jr R A
    and Car R 2019 Isotope effects in liquid water via deep
    potential molecular dynamics *Mol. Phys.* **117** 3269

[173] Sommers G M, Calegari Andrade M F, Zhang L, Wang H and
    Car R 2020 Raman spectrum and polarizability of liquid
    water from deep neural networks *Phys. Chem. Chem.
    Phys.* **22** 10592

[174] Zhang C, Zhang L, Xu J, Tang F, Santra B and Wu X 2020
    Isotope effects in x-ray absorption spectra of liquid water
    Phys. Rev. B **102** 115155

[175] Gartner T E, Zhang L, Piaggi P M, Car R,
    Panagiotopoulos A Z and Debenedetti P G 2020
    Signatures of a liquid-liquid transition in an *ab initio* deep
    neural network model for water *Proc. Natl Acad. Sci.*
    117 26040

[176] Andreani C, Romanelli G, Parmentier A, Senesi R,
    Kolesnikov A I, Ko H-Y, Calegari Andrade M F and
    Car R 2020 Hydrogen dynamics in supercritical water
    probed by neutron scattering and computer simulations *J.
    Phys. Chem. Lett.* **11** 9461

[177] Xu J, Zhang C, Zhang L, Chen M, Santra B and Wu X 2020
    Isotope effects in molecular structures and electronic
    properties of liquid water via deep potential molecular
    dynamics based on the SCAN functional *Phys. Rev.* B
    102 214113

[178] Piaggi P M, Panagiotopoulos A Z, Debenedetti P G and
    Car R 2021 Phase equilibrium of water with hexagonal
    and cubic ice using the SCAN functional *J. Chem. Theory
    Comput.* **17** 3065

[179] Tisi D, Zhang L, Bertossa R, Wang H, Car R and Baroni S
    2021 Heat transport in liquid water from first-principles
    and deep-neural-network simulations (arXiv:2108.10850
    [cond-mat.mtrl-sci])

[180] Zhang C, Tang F, Chen M, Xu J, Zhang L, Qiu D Y,
    Perdew J P, Klein M L and Wu X 2021 Modeling liquid
    water by climbing up Jacob's ladder in density functional
    theory facilitated by using deep neural network potentials
    J. Phys. Chem. B **125** 11444

[181] Torres A, Pedroza L S, Fernandez-Serra M and Rocha A R
    2021 Using neural network force fields to ascertain the
    quality of *ab initio* simulations of liquid water *J. Phys.
    Chem.* B **125** 10772

[182] Shi Y, Doyle C C and Beck T L 2021 Condensed phase water
    molecular multipole moments from deep neural network
    models trained on *ab initio* simulation data *J. Phys. Chem.
    Lett.* **12** 10310

[183] Calio P B, Li C and Voth G A 2021 Resolving the structural
    debate for the hydrated excess proton in water *J. Am.
    Chem. Soc.* **143** 18672

[184] Xu M, Zhu T and Zhang J Z H 2019 Molecular dynamics
    simulation of zinc ion in water with an *ab initio* based
    neural network potential *J. Phys. Chem.* A **123** 6587

[185] Niblett S P, Galib M and Limmer D T 2021 Learning intermolecular forces at liquid-vapor interfaces *J. Chem. Phys.* **155** 164101

[186] Galib M and Limmer D T 2021 Reactive uptake of N_2O_5 by atmospheric aerosol is dominated by interfacial processes *Science* **371** 921

[187] Andrade M F C, Ko H-Y, Zhang L, Car R and Selloni A 2020 Free energy of proton transfer at the water-TiO_2 interface from *ab initio* deep potential molecular dynamics *Chem. Sci.* **11** 2335

[188] Piaggi P M and Car R 2021 Enhancing the formation of ionic defects to study the ice Ih/XI transition with molecular dynamics simulations *Mol. Phys.* **119** e1916634

[189] Ye Q-J, Zhuang L and Li X-Z 2021 Dynamic nature of high-pressure ice VII *Phys. Rev. Lett.* **126** 185501

[190] Jiang S, Liu Y-R, Huang T, Feng Y-J, Wang C-Y, Wang Z-Q and Huang W 2021 Towards fully *ab initio* simulation of atmospheric aerosol nucleation (arXiv:2107.04802 [physics.atm-clus])

[191] Zeng J, Zhang L, Wang H and Zhu T 2021 Exploring the chemical space of linear alkane pyrolysis via deep potential generator *Energy Fuels* **35** 762

[192] Chen W-K, Liu X-Y, Fang W-H, Dral P O and Cui G 2018 Deep learning for nonadiabatic excited-state dynamics *J. Phys. Chem. Lett.* **9** 6702

[193] Zhang L, Wang H and E W 2018 Reinforced dynamics for enhanced sampling in large atomic and molecular systems *J. Chem. Phys.* **148** 124113

[194] Wang S, Ma Z and Pan W 2020 Data-driven coarse-grained modeling of polymers in solution with structural and dynamic properties conserved *Soft Matter* **16** 8330

[195] Pan X, Yang J, Van R, Epifanovsky E, Ho J, Huang J, Pu J, Mei Y, Nam K and Shao Y 2021 Machine-learning-assisted free energy simulation of solution-phase and enzyme reactions *J. Chem. Theory Comput.* **17** 5745

[196] Tuo P, Ye X B and Pan B C 2020 A machine learning based deep potential for seeking the low-lying candidates of Al clusters *J. Chem. Phys.* **152** 114105

[197] Achar S K, Zhang L and Johnson J K 2021 Efficiently trained deep learning potential for graphane *J. Phys. Chem.* C **125** 14874

[198] Wu J, Bai L, Huang J, Ma L, Liu J and Liu S 2021 Accurate force field of two-dimensional ferroelectrics from deep learning *Phys. Rev.* B **104** 174107

[199] Chen H, Chen J, Ning P, Chen X, Liang J, Yao X, Chen D, Qin L, Huang Y and Wen Z 2021 2D heterostructure of amorphous CoFeB coating black phosphorus nanosheets with optimal oxygen intermediate absorption for improved electrocatalytic water oxidation *ACS Nano* **15** 12418

[200] Pascuet M and Fernández J 2015 Atomic interaction of the MEAM type for the study of intermetallics in the Al-U alloy *J. Nucl. Mater.* **467** 229

[201] Jacobsen K W, Norskov J K and Puska M J 1987 Interatomic interactions in the effective-medium theory *Phys. Rev.* B **35** 7423

[202] Jain A *et al* 2013 Commentary: The materials project: a materials genome approach to accelerating materials innovation *APL Mater.* **1** 011002

[203] Wang Y, Lv J, Zhu L and Ma Y 2012 Calypso: a method for crystal structure prediction *Comput. Phys. Commun.* **183** 2063

[204] Aragones J L, Conde M M, Noya E G and Vega C 2009 The phase diagram of water at high pressures as obtained by computer simulations of the tip4p/2005 model: the appearance of a plastic crystal phase *Phys. Chem. Chem. Phys.* **11** 543

[205] Poschmann M, Asta M and Chrzan D C 2018 Convergence of calculated dislocation core structures in hexagonal close packed titanium *Modelling Simul. Mater. Sci. Eng.* **26** 014003

[206] Queyroux J-A *et al* 2020 Melting curve and isostructural solid transition in superionic ice *Phys. Rev. Lett.* **125** 195501

[207] Mishin Y 2021 Machine-learning interatomic potentials for materials science *Acta Mater.* **214** 116980

[208] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need (available at: https://doi.org/10.48550/ARXIV.1706.03762)

[209] Devlin J, Chang M-W, Lee K and Toutanova K 2019 Bert: pre-training of deep bidirectional transformers for language understanding *Proc. 2019th Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (Long and Short Papers Vol 1)* (Minneapolis, MN: Association for Computational Linguistics) pp 4171–86

[210] Brown T *et al* 2020 Language models are few-shot learners *Adv. Neural Inf. Process. Syst.* **33** 1877–901

[211] Min B, Ross H, Sulem E, Veyseh A P B, Nguyen T H, Sainz O, Agirre E, Heinz I and Roth D 2021 Recent advances in natural language processing via large pre-trained language models: a survey (available at: https://doi.org/10.48550/ARXIV.2111.01243)