

Full length article

Machine learning is funny but physics makes the money: How machine-learning potentials can advance computer-aided materials design in metallurgy

M. Hodapp

Materials Center Leoben Forschung GmbH (MCL), Leoben (AT), Austria

ARTICLE INFO

Keywords:

Computer-aided materials design
Mechanical properties
Atomistic simulations
Machine-learning potentials
Active learning
High-throughput screening

ABSTRACT

Over the past decade, machine learning interatomic potentials (MLIPs) have advanced many areas of computational materials science and chemistry. In metallurgy, however, substantial work remains to be done when computing *critical mechanical properties of metals* (e.g., strength, ductility, hardness, etc.). This is due to three distinct characteristics of metals that are, in this combined form, not necessarily relevant for predicting other properties (e.g., elastic constants). First, metals are inherently multiscale and, so, simulating realistic microstructures may require millions of atoms. On the other hand, to be predictive, MLIPs must be trained on quantum-mechanical simulations that are limited to a few hundred atoms, which raises the question of transferability to larger systems. Second, there are quantum-mechanical effects, such as magnetism, that are not yet to full extent accounted for in the functional form of state-of-the-art MLIPs. Third, the search space of metallic alloys spans over billions of compositions, and screening for the best possible alloy is like searching for the needle in the haystack without sophisticated high-throughput methods.

While the concept of MLIPs is extremely promising regarding quantitative accuracy and automation, these challenges must be addressed in the years to come in order to lift computer-aided materials design to the next level. This review attempts to provide a state-of-the-art survey of ongoing developments towards achieving this goal.

1. Introduction and overview

The main challenge in predicting mechanical properties of metals is their *multiscale nature*. This implies that a computational materials design process must take into account effects on all length scales, from angstroms to meters, and neglecting one of the length scales can lead to an incomplete understanding of the material behavior (see [1] for a comprehensive overview of this topic). A complete (though simplified for compactness) workflow of a *holistic* computer-aided materials design process is schematically depicted in Fig. 1 and consists of the following stages:

(i) the experimental stage,

where we analyze the atomic-scale behavior in response to mechanical loading (e.g., using transmission electron microscopy) to relate the behavior of crystalline defects to deformation,

(ii) the simulation stage,

where we attempt to translate the experimental picture into an atomic-scale model that can be

(iii) the modeling stage,

(iv) the screening stage,

used to investigate the relevant mechanism(s) that *drive* the defects using molecular dynamics simulations,

where we attempt to translate the outcome of the simulation(s) into a simpler model that allows us to understand the mechanical behavior in terms of material properties (e.g., cohesive energies, surface energies, stacking fault energies, etc.) and other tuning parameters (e.g., the size and density of precipitates),

where we feed the model developed in the previous stage with loads of data to find the best possible composition(s)

E-mail address: maxludwig.hodapp@mcl.at.

<https://doi.org/10.1016/j.commatsci.2023.112715>

Received 1 August 2023; Received in revised form 22 November 2023; Accepted 1 December 2023

Available online 7 December 2023

0927-0256/© 2023 Elsevier B.V. All rights reserved.

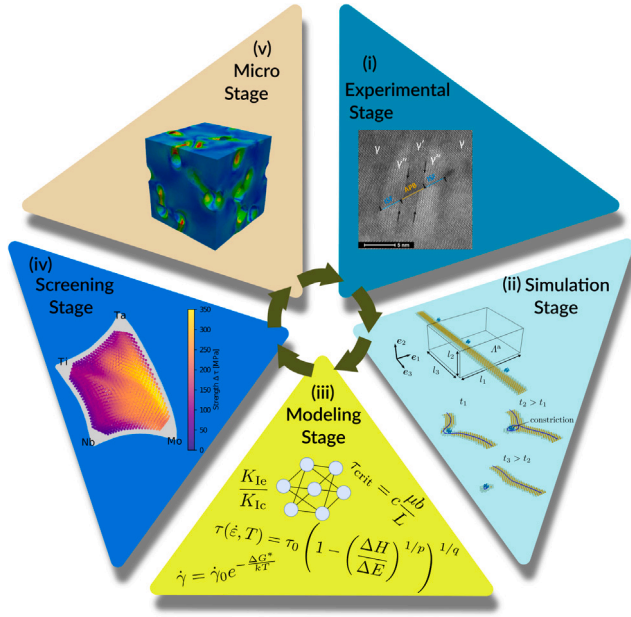


Fig. 1. Schematic workflow of a computer-aided materials design process. The figure in stage (i) is reproduced from [2] (Figure 7 therein) with permission from Elsevier.

(e.g., the ones with the best possible strength-vs-ductility ratios) for our application of interest, where we use information from stages (i)–(iii) to develop higher-scale models and methods (e.g., phase-field or discrete dislocation dynamics methods) to simulate microstructures in order to investigate the collective behavior of defects.

The information is then fed back to the experimental stage to investigate whether the favorable alloy compositions (a) are technically feasible, and, if yes, (b) improve on existing alloys.

The present work focuses on simulation methods for the stages (ii) and (iv). Ideally, simulations in those stages would be carried out using molecular dynamics where interatomic interaction is described by means of empirical interatomic potentials (EIPs) since this is the fastest way to simulate atomic-scale mechanisms. However, EIPs are not accurate enough to quantitatively describe multicomponent systems and complex chemical interactions and, hence, cannot be used for screening. At their best, they may be able to suggest a qualitative mechanism—but often not even that (see below).

An accurate method is plane-wave Density Functional Theory (in the following just DFT) [3] that computes the total energy of a configuration of atoms $\{r_i\}$ in terms of the electron density ρ as

$$\Pi^{\text{qm}} = \Pi^{\text{qm}}(\{r_i\}, \rho) = \langle \phi | \mathcal{H}(\{r_i\}, \rho) | \phi \rangle, \quad (1)$$

where the ϕ 's are the (orbital) wave functions, and \mathcal{H} is the Hamiltonian operator. However, we are actually interested in the ground-state energy, which requires minimizing (1) with respect to ρ . Moreover, each time we want to compute Π^{qm} for some ρ during the minimization, we need to solve an eigenvalue problem that scales *cubically* with the number atoms in $\{r_i\}$. DFT can therefore not be used to solve large-scale problems with tens of thousands of atoms and is mostly used for screening tasks that require configurations with not more than a few hundred atoms to be computed. But also for screening DFT quickly

reaches its limitations, in particular, when additional effects (e.g., relaxation, chemical ordering, etc.) over a wide range of compositions need to be considered.

Over the past two decades, machine learning models, in the field-specific language usually called *machine-learning interatomic potentials* (MLIPs), emerged as a new alternative for bringing quantitative accuracy into atomistic simulations, albeit several of the key ideas that are now commonly associated with MLIPs have already been developed about a decade earlier [4–7]—still within the framework of EIPs (cf., Fig. 2). The term “machine-learning potential” (or “machine-learning force field”) seems to originate from the physical chemistry community where it first appeared around the mid 2000s (see, e.g., [8] for a recent review), and throughout the past ~15 years MLIPs were primarily pioneered by works of Behler and Parrinello [9], Bartók et al. [10], Thompson et al. [11], Shapeev [12], Drautz [13], or Pun et al. [14], towards *large-scale atomistic simulations of metals*. Contrary to EIPs, they assume a very general functional form

$$\Pi^{\text{mlip}} = \Pi^{\text{mlip}}(\{r_i\}, \theta) \quad (2)$$

in terms of the atomic positions (and possibly other variables, such as the atom types) and a vector of free parameters θ . The generality allows MLIPs to provably approximate any (local) DFT energy (1) (cf., [12]), but comes at a price: besides certain symmetry conditions, there is, contrary to EIPs, usually not much physics contained in (2), and, therefore, parameterizing (2) becomes a delicate task. Parameterizing (2) is done by constructing a training set $\mathcal{T} = \{\Pi_i^{\text{qm}}, \{f_k^{\text{qm}}\}, \{\sigma_i^{\text{qm}}\}\}$ that contains the energy Π_i^{qm} , forces $\{f_k^{\text{qm}}\}$, and stresses σ_i^{qm} , coming from many (possibly thousands of) single-point DFT calculations. The parameters are then obtained by minimizing the loss function

$$\mathcal{L}(\theta, \mathcal{T}) = \sum_i \left(w_e \left(\Pi_i^{\text{mlip}} - \Pi_i^{\text{qm}} \right)^2 + w_f \sum_k \left\| f_k^{\text{mlip}} - f_k^{\text{qm}} \right\|^2 + w_s \left\| \sigma_i^{\text{mlip}} - \sigma_i^{\text{qm}} \right\|^2 \right),$$

where w_e , w_f , and w_s , are weights. Due to the lack of physics, to be predictive, the training set must be as representative as possible for the entire configurational space in which the MLIP is intended to be used, with some additional “physical wisdom” applied in order to improve the MLIP’s capabilities of extrapolation. Constructing a MLIP that is both *general* and *predictive* is usually referred to the (still open) *problem of transferability*. However, the good news is: there are certain guidelines that, if followed, allow to construct sufficiently general MLIPs that are good enough for many new applications that cannot be approached with EIPs, or DFT alone. Explaining them is one of the goals of the present review.

To give an example for a success story of MLIPs: one problem that has bothered the computational materials community for a long time was the problem of EIPs not being able to simultaneously predict several important properties of screw dislocations in body-centered cubic (bcc) metals, i.e., the compact core structure, the Peierls potential, and the correct Peierls stress and glide plane (cf., e.g., [15]). The properties of screw dislocations of course influence the motion of the dislocations, and the interaction with other types of defects, which is essential for simulating microstructures, and, so, results obtained using EIPs are not reliable. MLIPs have solved this problem once and for all (cf., [16,17]).

The present review article attempts to discuss the current state-of-the-art methodology behind MLIPs, and how MLIPs can be integrated into the process of computer-aided materials design exemplified in Fig. 1, with an emphasis on the author’s own contributions. In particular, we will discuss how to develop (Section 2), train (Section 3), and use (Section 4), MLIPs. An overview of available MLIP software packages and training sets, in addition to a summary with best practices on how to construct MLIPs is given in Section 5. Some selected open problems are presented in Section 6.

As such, the present review article does *not* attempt to cover an exhaustive literature review on MLIPs, but rather attempts to provide

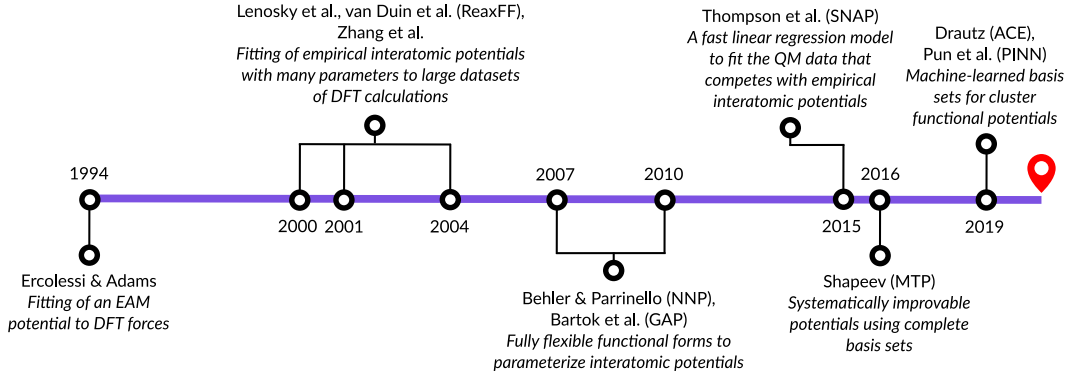


Fig. 2. Timeline with some of the most significant milestones for the development of MLIPs towards large-scale atomistic simulations of metals.

a concise overview of the topic, intended for prospective users as well as developers. There are other recent and more detailed review articles on each of the individual topics discussed here, e.g., by Musil et al. [18] on the analysis of MLIP descriptors, by Shapeev et al. [19] on active learning algorithms for training MLIPs, and by Wang et al. [20] on some overview of state-of-the-art MLIPs for alloys.

2. Machine-learning interatomic potentials

2.1. Basic notions and notation

Before proceeding, we fix some notation: let r_i be the position of an atom and $\{r_i, z_i\} = \{r_i\}_{i=1,\dots,N}$ be a configuration of N atoms, where z_i is a vector that specifies the atom type; the length of z_i is equal to the total number of atom types in our configuration. We assume a lexicographical order of types $t = \{0, 1, 2, \dots\}$ and define $z_{i\alpha} = \delta_{\alpha t_i}$, where t_i is the type of atom i . Each atom i is assumed to interact with all other M atoms in its neighborhood that we denote as $\mathcal{N} = \{r_{ij}, z_{ij}, z_j\}_{j=1,\dots,M}$, where $r_{ij} = r_j - r_i$ are the interatomic distances. We then assume that the total energy of a configuration Π is a sum of per-atom contributions \mathcal{E}

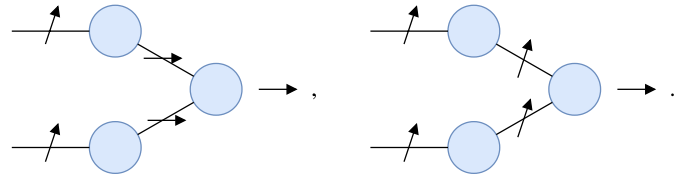
$$\Pi(\{r_i\}) = \sum_{\mathcal{N} \in \{r_i, z_i\}} \mathcal{E}(\mathcal{N}).$$

Constructing a good \mathcal{E} such that Π systematically approximates Π^{qm} down to the typical noise in numerical DFT codes is one of the central parts of developing MLIPs.

Descriptors and features. The terms feature and descriptor are often used interchangeably. We will use the term feature in the following, unless we refer to well-established terms. A feature is a per-atom quantity $F = F(\mathcal{N})$, usually a tensor, that depends on variables that describe the atomic environment. The simplest features are pairwise features composed of radial basis functions $Q = Q(|r_{ij}|)$, such that $F_{\text{pair}} = \sum_j Q(|r_{ij}|)$, that can be used to construct pair potentials. The radial basis functions must have a smooth cut-off such that \mathcal{E} remains smooth when atoms are entering or leaving the neighborhood \mathcal{N} . Pair potentials are almost never enough to model a more complex material behavior, so we also need features that take into account the angle between bonds, i.e., three-body features $F_{3\text{-body}} = \sum_{j,k} f(r_{ij}, r_{ik})$ or terms with an even higher body-order, e.g., $F_{4\text{-body}} = \sum_{j,k,l} g(r_{ij}, r_{ik}, r_{il})$. Features with body-order higher than tree are almost never used explicitly because computing them scales exponentially with the body-order. Most state-of-the-art MLIPs therefore use a separation of variables by making the Ansatz $\tilde{F}_{3\text{-body}} = \sum_j f'(r_{ij}) \sum_k f''(r_{ik})$, etc. Using this Ansatz, one can construct features of arbitrary body-order by multiplying functions that only depend on pairwise interactions and, so, computing $\tilde{F}_{3\text{-body}}$ scales linear with the number of atoms in the neighborhood. Features that describe multicomponent systems can be constructed by letting the radial basis depend on the atom types,

i.e., $Q_{\alpha\beta}(|r_{ij}|) = Q(|r_{ij}|)z_{i\alpha}z_{j\beta}$. In the following, we will simply use one index n , in machine learning usually called “channel”, that comprises all radial features (including different types of radial basis functions), unless $z_{i\alpha}z_{j\beta}$ is explicitly featured in F .

Invariance and equivariance. A good \mathcal{E} needs to encode the model symmetries, namely, invariance under permutation between chemically equivalent atoms, and invariance under Euclidean transformations (translation, rotation, and reflection). All potentials referenced in the present article are permutation and translation invariant, so we do not discuss it further. One possibility to account for invariance under actions of $O(3)$, the group of reflections and rotations, is by making the feature tensors, activation functions, etc., invariant under actions of this group. However, this might constrain the flexibility of the functional form too much. Another closely related concept that allows for a higher flexibility is *equivariance*, where we require that a *covariant transformation* $x \mapsto f(x)$ commutes with the action of the group g , i.e., $f(g \circ x) = g \circ f(x)$. The difference between invariant and equivariant transformations can be schematically explained as follows: suppose all inputs and outputs of a network are two-dimensional vectors \rightarrow , then, if these input vectors are rotated to \nearrow , a network with invariant and a (neural) network with equivariant layers can be visualized as



Usually one of the outputs of the last (hidden) layer must be *contravariant* so that the final output is *invariant* under the action of g . Most state-of-the-art MLIPs discussed below use equivariant transformations of the inputs.

2.2. First-generation potentials

MLIPs can roughly be grouped into three main classes of potentials: (1) neural network potentials, (2) Gaussian process-based potentials, and (3) polynomial potentials. In the following, we discuss some of the most popular and well-established MLIPs from each of these classes that have been proposed over the past decade.

2.2.1. Neural network potentials

Neural Network Potentials (NNPs), initially proposed by Behler and Parrinello [9], have been the first class of MLIPs. In its original form, it contained the two-body and three-body features, later also referred to as the “Behler-Parrinello symmetry functions”,

$$F_{2\text{-body},n}^{\text{nnp}} = G^1(\mathcal{N}) = \sum_j Q_n(|r_{ij}|),$$

and

$$F_{3\text{-body},n}^{\text{nnp}} = G^2(\mathcal{N}) = \sum_{j,k} Q_n(|r_{ij}|) Q_n(|r_{ik}|) Q_n(|r_{jk}|) f(|r_{ij}|, |r_{ik}|, r_{ij}^\top \cdot r_{ik}).$$

The per-atom energy of NNPs is then the output

$$\mathcal{E}^{\text{nnp}} = \mathcal{E}^{\text{nnp}}(\{F_{2\text{-body},n}^{\text{nnp}}\}, \{F_{3\text{-body},n}^{\text{nnp}}\})$$

of an artificial neural network. Obviously, all features are O(3)-invariant, and so is \mathcal{E}^{nnp} . NNPs still appears to be the largest class of MLIPs, with newer versions being continuously developed, e.g., SchNet [21], ANI-1 [22], DeepMD [23], or physically informed neural network (PINN) potentials [14].

In principle, one could also consider features with a higher body-order, however, this would not be very efficient due to the exponential dependence on the body-order (see above). Newer variants of NNPs use transformations of pairwise features, e.g., the convolutional neural networks (CNNs), as proposed by Batzner et al. [24], use filters of the form

$$\mathbb{F}_{n\ell m} = \mathbb{F}_{n\ell m}(r_{ij}) = Q_n(|r_{ij}|) Y_{\ell m}(\hat{r}_{ij}),$$

where $Y_{\ell m}$ are the spherical harmonics basis functions that depend on the angular part \hat{r}_{ij} of r_{ij} . An output of a convolutional layer is then produced as

$$v_{n\ell_3 m_3} = \sum_{\substack{\ell_1, \ell_2, \\ m_1, m_2}} C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3} \left(\sum_j \mathbb{F}_{n\ell_2 m_2}(r_{ij}) \right) F_{n\ell_1 m_1}^{\text{cnn}}, \quad (3)$$

where $C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3}$ are the Clebsch–Gordan coefficients (e.g., [25]) that contract the filter with the input features $F_{n\ell_1 m_1}^{\text{cnn}}$ in an *equivariant* manner so that the output of the neural network is SO(3)-invariant (cf., Section 2.1). To account for the full O(3)-invariance, we remark that, under reflections, spherical harmonics transform from $Y_{\ell m}(\hat{r}_{ij})$ to $(-1)^\ell Y_{\ell m}(-\hat{r}_{ij})$, so we only need to make sure that the sum of all ℓ -indices corresponding to the filters is even. Note that in [24], a CNN layer also features a message-passing mechanism (cf., Section 2.3.1) that we omitted in (3) for the sake of explaining the equivariant contractions more clearly.

2.2.2. Gaussian process-based potentials

The historically second main class of MLIPs are the Gaussian process-based potentials that have been pioneered by Bartók et al. [10]. Potentials from this class are kernel-based. That is, the per-atom energy \mathcal{E} is composed of functions $k(\mathcal{N}, \mathcal{N}_\alpha)$, the kernels, that measure the distance between some neighborhood \mathcal{N} and neighborhoods \mathcal{N}_α from the training set, instead of building the MLIP's functional form directly from the features. The arguably most popular representative of this class is the Gaussian Approximation Potential (GAP) that uses the “smooth overlap of atomic positions” (SOAP) kernel [26]. The SOAP kernel measures the distance between smooth per-atom densities

$$\rho(r) = \rho(\mathcal{N}, r) = \sum_j Q_n(|r_{ij}|) e^{-\frac{(|r| - |r_{ij}|)^2}{2\sigma^2}},$$

and additionally integrates them over all reflections and rotations to ensure its invariance over O(3) such that

$$\tilde{k}(\mathcal{N}, \mathcal{N}') = \left(\int_{\text{O}(3)} \left(\int \rho(r) \rho'(Qr) dr \right) dQ \right)^\zeta, \quad (4)$$

where $\rho'(Qr) = \rho(\mathcal{N}', Qr)$, and $\zeta \geq 1$ is a parameter that defines the body-order of GAP. The per-atom energy of GAP can then be written as follows

$$\mathcal{E}(\mathcal{N}) = \sum_\alpha \theta_\alpha k(\mathcal{N}, \mathcal{N}_\alpha),$$

where $k(\mathcal{N}, \mathcal{N}_\alpha) = \tilde{k}(\mathcal{N}, \mathcal{N}_\alpha) / \sqrt{\tilde{k}(\mathcal{N}, \mathcal{N}) \tilde{k}(\mathcal{N}_\alpha, \mathcal{N}_\alpha)}$ is the normalized SOAP kernel.

In [26], it was shown that an efficient way to compute (4) is to approximate the density by expanding it in terms of spherical harmonics

$$\rho(r) = \sum_{n,\ell,m} c_{n\ell m} Q_n(|r|) Y_{\ell m}(\hat{r}).$$

With such an expansion, a SOAP kernel for, e.g., $\zeta = 3$ can then be written as

$$\tilde{k}(\mathcal{N}, \mathcal{N}') = \sum_{\substack{n_1, n_2, n_3, \\ \ell_1, \ell_2, \ell_3}} \left(\sum_{m_1, m_2, m_3} C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3} c_{n_1 \ell_1 m_1} c_{n_2 \ell_2 m_2} c_{n_3 \ell_3 m_3}^\dagger \right) \left(\sum_{m'_1, m'_2, m'_3} C_{\ell'_1 m'_1 \ell'_2 m'_2}^{\ell'_3 m'_3} c'_{n'_1 \ell'_1 m'_1} c'_{n'_2 \ell'_2 m'_2} c'_{n'_3 \ell'_3 m'_3}^\dagger \right).$$

2.2.3. Polynomial potentials

The third main class are the polynomial potentials, which, as their name suggests, are constructed by taking polynomials of the feature tensors. To the author's best knowledge, the first MLIP that could be termed “polynomial potential” is the spectral neighbor analysis potential (SNAP) of Thompson et al. [11]. SNAP uses the feature tensors

$$F_{n\ell m}^{\text{snap}} = \sum_j Q_n(|r_{ij}|) Y_{\ell m}(\hat{r}_{ij})$$

in order to construct SO(3)-invariant basis functions using the SO(3) bispectrum coefficients

$$B_\alpha(\mathcal{N}) = \sum_{m_1, m_2, m_3} C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3} F_{n_1 \ell_1 m_1}^{\text{snap}} F_{n_2 \ell_2 m_2}^{\text{snap}} F_{n_3 \ell_3 m_3}^{\text{snap}^\dagger}(\mathcal{N}),$$

with the multi-index $\alpha = (n_1 n_2 n_3 \ell_1 \ell_2 \ell_3)$. Again, as for the CNNs described in Section 2.2.1, $\ell_1 + \ell_2 + \ell_3$ must be even to ensure the full O(3)-invariance of the B_α 's. The per-atom energy is then given by a linear combination of those basis functions

$$\mathcal{E}^{\text{snap}}(\mathcal{N}) = \sum_\alpha \theta_\alpha B_\alpha(\mathcal{N}).$$

An alternative for the basis functions B_α proposed in [11] are the SO(4) bispectrum coefficients, which require hyperspherical harmonics basis functions.

Shortly after SNAP, Shapeev proposed his moment tensor potentials (MTPs) [12] that can be considered as a generalization of SNAP to arbitrary body-order. The basis functions B_α^{mtp} of MTPs are constructed by contracting combinations of the moment tensor descriptors

$$M_{\mu, \nu} = \sum_j Q_\mu(|r_{ij}|) \underbrace{(r_{ij} \otimes \cdots \otimes r_{ij})}_{\nu \text{ times}} \quad (5)$$

to a scalar. Linear MTPs are then constructed in the same way as SNAP, i.e., $\mathcal{E}^{\text{mtp}}(\mathcal{N}) = \sum_\alpha \theta_\alpha B_\alpha^{\text{mtp}}(\mathcal{N})$.

Another popular class of polynomial MLIPs are the atomic cluster expansion (ACE) potentials developed by Drautz [13]. ACE is a more direct generalization of SNAP than MTP: instead of using moment tensors, it also uses a spherical harmonics basis set, so their feature tensors are the same, i.e., $F^{\text{ace}} = F^{\text{snap}}$. The per-atom energy of an, e.g., five-body ACE potential can then be written as follows

$$\mathcal{E}^{\text{ace}}(\mathcal{N}) = \mathcal{E}^{\text{snap}}(\mathcal{N}) + \sum_{\substack{m_1, m_2, m_3, \\ m_4, m_5}} C_{\ell_1 m_1 \ell_2 m_2}^{\ell_3 m_3} C_{\ell_3 m_3 \ell_4 m_4}^{\ell_5 m_5} F_{\ell_1 m_1 n_1}^{\text{ace}} F_{\ell_2 m_2 n_2}^{\text{ace}} F_{\ell_3 m_3 n_3}^{\text{ace}} F_{\ell_4 m_4 n_4}^{\text{ace}} F_{\ell_5 m_5 n_5}^{\text{ace}}.$$

Higher body-order terms can be added by chaining additional Clebsch–Gordan coefficients and feature tensors.

2.2.4. All potentials are equal but some are more equal than others

In [27], we have made an attempt to compare MLIPs graphically using tensor network diagrams [28]. In Fig. 3, we show the five MLIPs presented above using this notation.

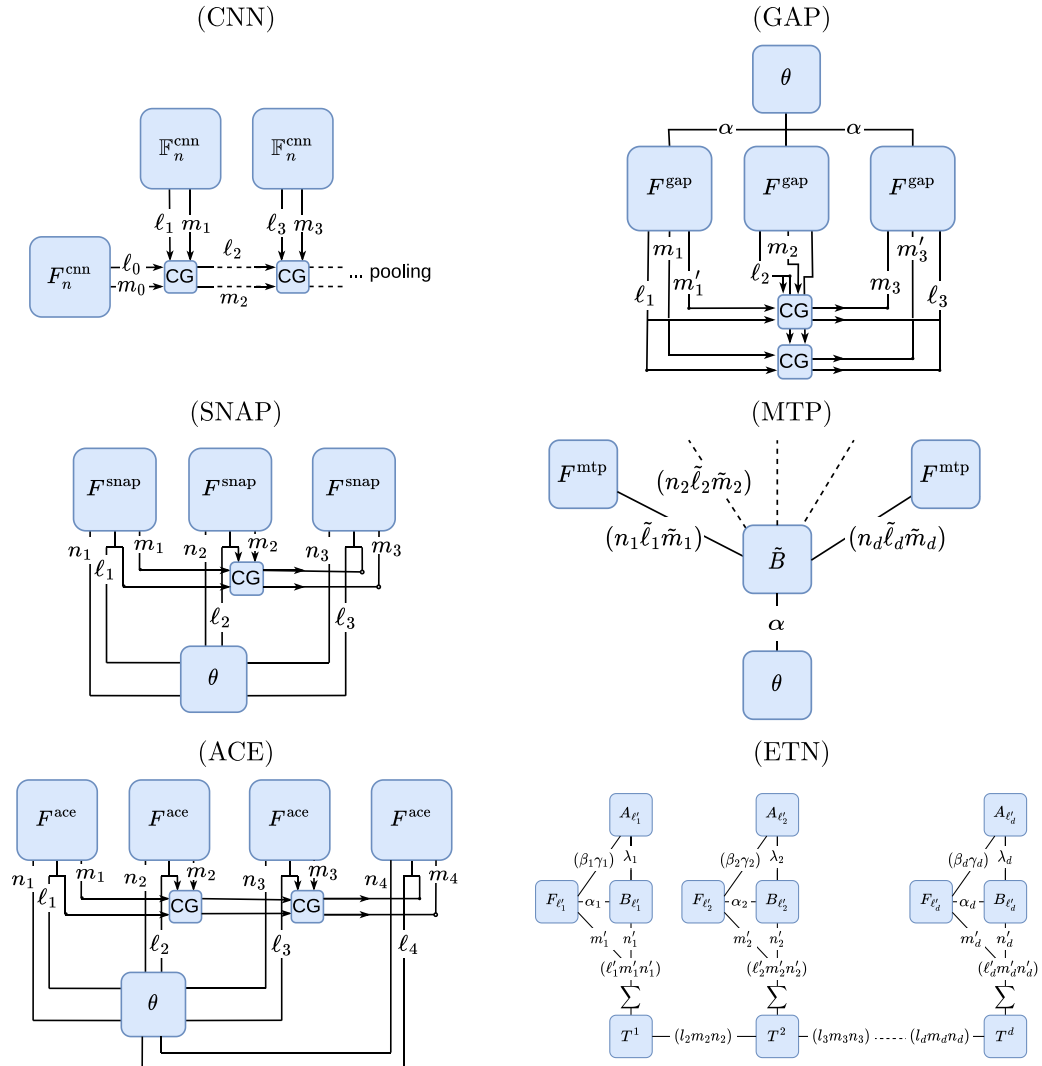


Fig. 3. MLIPs described in Sections 2.2.1–2.2.3 in their tensor network representation. The tensors are represented with boxes, their dimensions are represented with links, and each diagram represents the result of contractions (two connected tensor dimensions implies that these dimensions are contracted). For CNNs, GAP, SNAP, and ACE, the direction of the lines specifies the form of the tensor (co- or contravariant). MTPs and ETNs use a real basis where such a distinction is not necessary (all tensors are covariant). The diagrams of SNAP and ACE are reproduced from [27] (equation (B7) & (B8) therein).

By comparing the tensor network notation, we observe that the contraction of the angular basis set is very much the same for CNNs, SNAP, and ACE. On the other hand, SNAP and ACE use a linear combination of basis functions (each combination of the n 's and ℓ 's is a basis function), while CNNs have a more complicated architecture that includes further (nonlinear) layers not shown in Fig. 3.

We may also cast GAP into a tensor network notation if we interpret the coefficients that arise from the expansion of the atomic densities as feature tensors

$$F_{\ell m m'}^{\text{gap}} = \sum_n c_{n\ell m} c_{n\ell' m'}.$$

Interestingly, GAP also uses a spherical harmonics basis but involves two Clebsch–Gordan coefficients in the four-body SOAP kernel (cf., Fig. 3) instead of one as SNAP does.

The main exception is MTP, which uses moment tensors as features instead of spherical harmonics to construct a complete basis set. Mathematically, the only difference between the two is that, for a moment tensor $M_{\mu,\nu}$, all dimensions of the tensor are rotated simultaneously and, hence, their scalar contractions are always rotational invariant. On the contrary, spherical harmonics get re-expanded and rotate during

the contraction to a scalar—and it is the job of the Clebsch–Gordan coefficients to ensure that this scalar is invariant under actions of $\text{SO}(3)$.

To make this more apparent, we may re-interpret the moment tensor descriptors (5) as feature tensors (cf., [27])

$$F_{n\ell\tilde{m}}^{\text{mtp}} = Q_n(\|r_{ij}\|) \underbrace{(r_{ij} \otimes \dots \otimes r_{ij})}_{\tilde{\ell} \text{ times}}_{(\tilde{\ell}\tilde{m})},$$

which includes all $M_{\mu,\nu}$'s contained in the basis functions, reshaped to a two-dimensional tensor; we have chosen its indices to be $\tilde{\ell}$ and \tilde{m} to highlight the similarity with respect to the spherical harmonics basis. The per-atom energy can then be written as

$$\mathcal{E}^{\text{mtp}} = \sum_{\alpha} \theta_{\alpha} \tilde{B}_{\alpha n_1 \tilde{\ell}_1 \tilde{m}_1 \dots n_d \tilde{\ell}_d \tilde{m}_d} F_{n_1 \tilde{\ell}_1 \tilde{m}_1} \dots F_{n_d \tilde{\ell}_d \tilde{m}_d}.$$

Here, \tilde{B}_{α} is an integer tensor that encodes all $\text{O}(3)$ -invariant contractions of the features for a given basis function α .

In summary, it appears that the way of MLIPs incorporating symmetries has been mostly settled. The main difference between the MLIPs lies in the way how the channels are composed and contracted. For example, GAP, SNAP, MTP, and ACE, use multilinear contractions, while CNNs use more complicated nonlinear dependencies. Zuo et al.

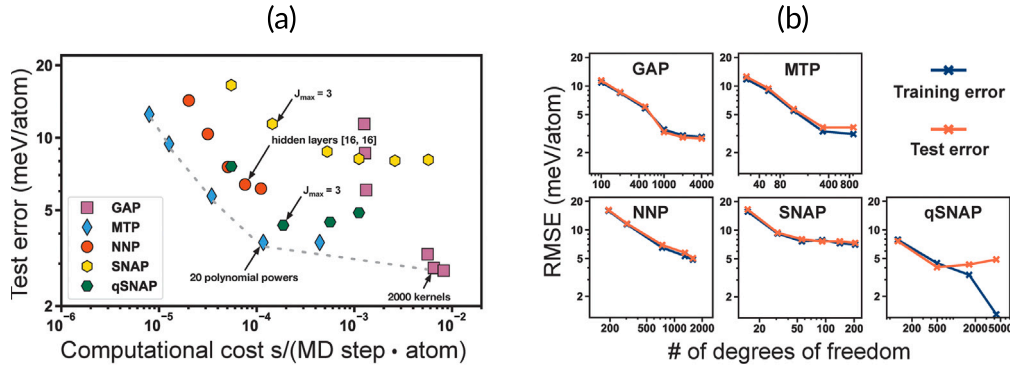


Fig. 4. Benchmark test from [29] comparing the performance of various MLIPs described in Section 2.2. (a) MLIP test error vs. computational cost for a training set comprising molybdenum; the gray line corresponds to the Pareto front indicating the optimal trade-off between accuracy and computational cost. (b) MLIP test error vs. number of free parameters.

Source: The figure has been reprinted (adapted) from [29] (Figure 2 therein) with permission from ACS.

[29] have published a benchmark study that compares the accuracy-vs-computational cost ratio between the Behler–Parrinello–NNPs, GAP, SNAP, and MTP, for the unary systems lithium, molybdenum, copper, nickel, silicon, and germanium, which showed favorable performance of the polynomial potential MTP (cf., Fig. 4). Later, Lysogorskiy et al. [30] published another benchmark study using different training sets for copper and silicon, and obtained comparable performance to that of MTP for the other polynomial potential ACE. However, two years have passed since then and many new advancements to improve the generality and performance of MLIPs are in progress, which calls for a new comparison. We discuss some of those advancements in the following.

2.3. Next-generation potentials

2.3.1. Expanding the feature space: magnetic moments, electronic temperature, long-range interactions, ...

All previous MLIPs have been proposed for multicomponent systems, yet, studying systems that depend on more complex quantum-mechanical phenomena requires expanding the feature space beyond atomic positions and types. Some possible extensions to improve the generality of MLIPs are discussed below.

Magnetic moments. To include magnetic effects into \mathcal{E} requires to enrich the neighborhoods to $\mathcal{N} = \{\mathbf{r}_{ij}, \mathbf{z}_i, \mathbf{z}_j, \mathbf{s}_i, \mathbf{s}_j\}$, where \mathbf{s}_i and \mathbf{s}_j are the spins of the i th atom and all j atoms in its neighborhood. A way of introducing collinear spins (only absolute values of \mathbf{s}_i and \mathbf{s}_j are considered as degrees of freedom) into the functional forms of MTPs has been proposed by Novikov et al. [31] and successfully applied to study properties of α -iron with DFT accuracy. The moment tensor descriptor for (unary) collinear magnetic MTPs can be written as

$$M_{\mu,\nu} = M_{(n\gamma\eta),\nu} = \sum_j Q_n(\|\mathbf{r}_{ij}\|) Q_{\gamma}^{\text{spin}}(\|\mathbf{s}_i\|) Q_{\eta}^{\text{spin}}(\|\mathbf{s}_j\|) \underbrace{(\mathbf{r}_{ij} \otimes \cdots \otimes \mathbf{r}_{ij})}_{v \text{ times}},$$

where the Q^{spin} 's are radial basis functions for the spins (they are not forced to go to zero beyond some cut-off radius as opposed to the Q 's). A paramagnetic model was also proposed in [31] by building outer products of spin vectors $\mathbf{s}_i \otimes \mathbf{s}_j \otimes \cdots \otimes \mathbf{s}_i \otimes \mathbf{s}_j$ into the moment tensors $M_{\mu,\nu}$, but has not been implemented. An analogous way of integrating magnetic degrees of freedom into ACE has been proposed in [32,33].

Electronic temperature. Another example for quantum-mechanical phenomena is the kinetic energy of electrons that becomes in particular relevant for solids at high temperature, liquid metals, or plasmas. The kinetic energy of electrons can be built into the MLIP's functional form by adding the electronic temperature T as an additional feature to the per-atom energies such that $\mathcal{E} = \mathcal{E}(\mathcal{N}, T)$. MLIPs that include the electronic temperature as features are, e.g., NNPs [34,35].

Message passing. Message passing is a mechanism that can be considered as a more efficient way of including long-range interactions beyond the cut-off radius—but without actually increasing it (see [36] for a review). The main idea of message passing is to pass features of neighboring atoms to the central atom, which technically creates new, higher-level, features. This passing can be repeated multiple times to increase the interaction range. For example, in CNNs, this is implemented by letting $F_{n\ell_1 m_1}^{\text{cnn}}$ in (3) also depend on \mathcal{N} . Other message passing algorithms have been developed for, e.g., NNPs and ACE [37,38].

2.3.2. Compressing the feature space

One problem that comes along with expanding the feature space is its exponential increase with the number of features. This can be most intuitively explained for polynomial potentials: polynomial MLIPs can be encoded with tensors, i.e., if their functional is a d th order polynomial, it can be written as follows

$$\theta_{i_1 \dots i_n}^1 \dots i_1^d \dots i_n^d F_{i_1 \dots i_n}^1 \dots i_1^d \dots i_n^d, \quad (6)$$

where $\theta_{i_1 \dots i_n}^1 \dots i_1^d \dots i_n^d$ is the parameter tensor (i.e., the tensor of polynomial coefficients), and F is the feature tensor. The total number of parameters for this representation is $\bar{m}^{\bar{d}n}$, where \bar{m} is the average size over all dimensions of the feature tensor, which makes it obviously very inefficient when increasing the number of features. Therefore, we urgently need to find reduced representations of (6) when adding more physics to the potential. Mathematically, a reduced representation of (6) can be motivated as follows. Suppose our feature tensor has two dimensions and our model depends linear on it so that it can be expressed as $\theta_{ij} F_{ij}$. Another way of representing our parameter tensor is by using two matrices $\underline{A} \in \mathbb{R}^{\bar{m} \times r}$, and $\underline{B} \in \mathbb{R}^{r \times \bar{m}}$, such that $\theta_{ij} F_{ij} = A_{ik} B_{kj} F_{ij}$. Now, if we can represent $\theta_{ij} F_{ij}$ still sufficiently well by making the rank r much smaller than \bar{m} , θ_{ij} is low-rank and our feature space scales effectively linear with \bar{m} . Physically, this means that there is a enough similarity between the features. In machine learning, such a contraction scheme is considered to be a type of “autoencoder”, a more general concept of feature compression that may also account for nonlinear dependencies between the features.

Feature contraction had been built into multicomponent MTPs [39] from the very beginning by contracting the radial and type features to some new vector, i.e.,

$$M_{\mu,\nu} = \sum_j c_{\mu\alpha\beta} Q_n(\|\mathbf{r}_{ij}\|) z_{i\alpha} z_{j\beta} \underbrace{(\mathbf{r}_{ij} \otimes \cdots \otimes \mathbf{r}_{ij})}_{v \text{ times}},$$

before the whole moment tensor descriptor is contracted to a basis function. For the Gaussian process-based potentials, a contraction scheme similar to that of MTPs has been developed by Lopanitsyna et al. [40]. For GAP, Darby et al. [41] developed a compression scheme that

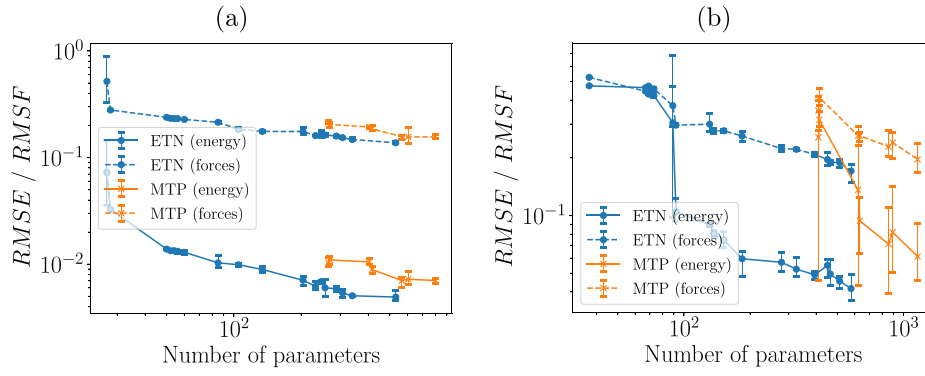


Fig. 5. Root-mean-square errors for the energies ($RMSE$) and forces ($RMSF$) for ETN potentials (8) and MTPs as a function of the number of parameters for (a) the Mo-Nb-Ta-W dataset from [43], and for (b) the Mo-Nb-Ta-V-W dataset from [44]; the horizontal bars correspond to the minimum and maximum values of several training runs with different initializations of the parameters.

Source: The figures are reproduced from [27] (Figure 7 & 8 therein).

exploits symmetries in the power spectrum and bispectrum coefficients. Moreover, further compressed versions of GAP and ACE have been recently proposed by Darby et al. [42] using the canonical polyadic decomposition.

A general framework of representing high-dimensional tensors in a data-efficient format are tensor networks, e.g., tensor trains, hierarchical Tucker, PEPS, etc. (cf., [45,46]). The power of tensor networks lies in their construction: they can be built entirely from contractions of *order-3 tensors*, which heavily simplifies generic implementations and automation of operations (e.g., automatic differentiation, tensor factorization, etc.). *Equivariant tensor networks (ETNs)*, in which each tensor contraction is an *equivariant transformation* that accounts for the necessary model symmetries, have been pioneered in quantum physics (cf., [47]), and early hints that ETNs could be a favorable concept for representing atomic environments can be found in [48]. An equivariant transformation is an *equivariant order-3 tensor* \underline{T} , and, according to the Wigner–Eckhart Theorem, any \underline{T} with three multi-indices $\{(n_i \ell_i m_i)\}_{i=1,\dots,3}$, can be factorized to

$$T_{(n_1 \ell_1 m_1)(n_2 \ell_2 m_2)(n_3 \ell_3 m_3)} = \theta_{(n_1 \ell_1)(n_2 \ell_2)(n_3 \ell_3)} C_{(\ell_1 m_1)(\ell_2 m_2)(\ell_3 m_3)}, \quad (7)$$

where $\theta_{(n_1 \ell_1)(n_2 \ell_2)(n_3 \ell_3)}$ contains the degrees of freedom, and $C_{(\ell_1 m_1)(\ell_2 m_2)(\ell_3 m_3)}$ is the Clebsch–Gordan coefficient that defines the symmetry group $O(3)$ in our case of interatomic potentials. Analogous to standard tensor networks, ETNs can be built from sequences of tensor contractions of the following form

$$\begin{aligned} u_{(n_1 \ell_1 m_1)}^2 &= T_{(n_1 \ell_1 m_1)(n_2 \ell_2 m_2)(n_3 \ell_3 m_3)}^1 v_{(n_2 \ell_2 m_2)}^1 u_{(n_3 \ell_3 m_3)}^1, \\ u_{(n_1 \ell_1 m_1)}^3 &= T_{(n_1 \ell_1 m_1)(n_2 \ell_2 m_2)(n_3 \ell_3 m_3)}^2 v_{(n_2 \ell_2 m_2)}^2 u_{(n_3 \ell_3 m_3)}^2, \\ &\dots \end{aligned}$$

where the u 's and v 's are covariant vectors. This has led us to propose *ETN potentials* [27] that are built entirely from such sequences of operations. For example, a tensor train representation of atomic energies can be written as follows

$$\begin{aligned} \mathcal{E}^{\text{etn}} &= \left(T_{(n'_1 \ell'_1 m'_1)(n_1 \ell_1 m_1)}^1 F_{(n'_1 \ell'_1 m'_1)}^{\text{etn}} \right) \left(T_{(n_1 \ell_1 m_1)(n'_2 \ell'_2 m'_2)(n_2 \ell_2 m_2)}^2 F_{(n'_2 \ell'_2 m'_2)}^{\text{etn}} \right) \\ &\dots \left(T_{(n_{d-1} \ell_{d-1} m_{d-1})(n'_d \ell'_d m'_d)}^d F_{(n'_d \ell'_d m'_d)}^{\text{etn}} \right). \end{aligned} \quad (8)$$

in which the *channels* n_1, n_2, \dots, n_d *naturally emerge as tensor ranks*. Moreover, we have developed a multilinear compression scheme in

which the radial features are contracted before entering the tensor train leading to

$$F_{(n \ell m)}^{\text{etn}} = \sum_j \left(B_{n \alpha \ell} Q_{\alpha}(|r_{ij}|) (A_{\lambda \beta \gamma \ell} z_{i \beta} z_{j \gamma}) \right) Y_{\ell m}(\hat{r}_{ij}).$$

This enables learning similarities between *all* radial features from the data through the parameter tensors A and B . The number of parameters of ETN potentials is proportional to $d(n + \bar{m})\bar{r}^2$, where \bar{r} is average rank over all \underline{T} 's, which makes them obviously more efficient than the representation (6) if \bar{r} can be kept small. In practice, ETN potentials require 2–3 times fewer parameters than MTPs (even though MTPs already contain some feature compression, cf., Fig. 4(b)) on training sets for four- and five-component alloys to reach the same accuracy (cf., Fig. 5), and we anticipate that this ratio will be much more pronounced when adding more features, such as spins, in favor of ETNs.

3. How to construct a good training set?

3.1. The problem of transferability

Having decided on the functional form, the main challenge that remains in order to create a good MLIP is the construction of the training set. This is especially challenging because we are, on the one hand, interested in using the MLIP in simulations with thousands of atoms—on the other hand, we are only able to compose our training set out of configurations of at most a few hundred atoms because we cannot afford to compute more with DFT. There are, in principle, three ways to approach the problem of transferability: (1) by attempting to create very general training set, (2) by adding more physics to the functional form so that the potential can better extrapolate to unseen structures, and (3) by active learning, an algorithmic approach that allows us to identify new unseen configurations on which the potential needs to be trained while already running the simulations.

The first approach is conceptually the easiest one and should be preferred if the physical problem to be solved is known in advance. For example, if we want to simulate screw dislocations in bcc metals, we may even “handcraft” a training set composed of bulk configurations, configurations with stacking faults, and configurations with dislocation dipoles (those are small enough because dislocation cores in bcc metals are usually compact). And the early works by Szlachta et al. [16], Dragoni et al. [49] on α -iron using GAPs have shown that this leads to

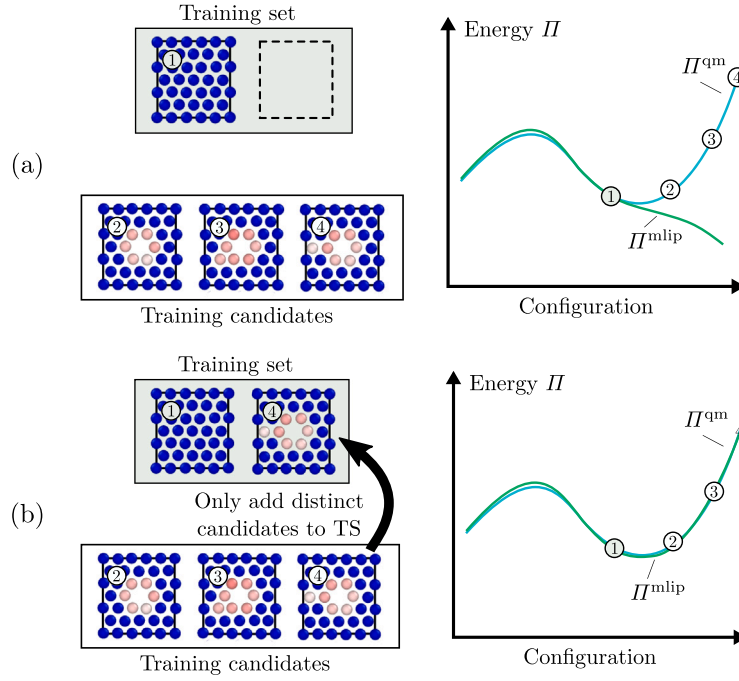


Fig. 6. Schematic illustration of the functioning of active learning.

good potentials that can be used to simulate screw dislocation motion in cells with more than 10 000 atoms [50]. However, other works in which the same potential has been used showed that it did not very well extrapolate to other types of defects that were not included in the training set, e.g., edge dislocations [51], or cracks [52].

A new approach to the problem of constructing general training sets has been recently proposed by Poul et al. [53] who generated their training sets by sampling configurations from all possible space groups. They showed that their approach allows to construct MTPs for pure magnesium that perform well on various (bulk) defects (e.g., stacking faults or grain boundaries), although those defects have not been included in the training set explicitly. How this approach extends to more complex types of defects, e.g., curved dislocations, or cracks, and, in particular, to multicomponent systems is yet unclear.

The second approach is considerably more difficult since adding more physics may easily spoil the potential in terms of generality. Some works, e.g., by Pun et al. [14], have pursued this approach by adding activation functions that take the form of bond-order potentials to a neural network architecture and shown that this can lead to a strongly improved transferability compared to the original NNP by Behler and Parrinello [9]. A similar approach has been proposed by Drautz [13] who used ACE to parameterize the electron densities in Finnis–Sinclair-type potentials. An alternative approach in this direction could be autoencoders (cf., Section 2.3.2) that would be able to learn dependencies between features and/or data, and, hence, would be able to infer from those dependencies to unseen structures.

3.2. Active learning

One of the major goals of machine learning is the automation of tasks and processes. This line of thought is not only useful in view of automating the process of developing MLIPs, but, in fact, crucial if we want to make sure that a MLIP performs sufficiently well on large-scale configurations that we cannot validate anymore with DFT. Conceptually, active learning is a way to achieve exactly this.

The idea behind active learning can be intuitively understood as follows. Suppose we want to simulate some crystalline material and we have a MLIP trained on the training set shown in Fig. 6(a). This

potential is able to approximate the energy in the configurational space up to some point ①, but gets worse and worse when approaching the remaining points ②, ③, and ④. However, we do not know the DFT energy in general, so, the job of active learning is now to check whether the potential is inaccurate for some configuration—ideally without doing any additional DFT calculations—and add this configuration to the training set. Moreover, the number of added configurations should be as small as possible so that the potential can still interpolate the DFT energy (in the figure, it suffices to add configuration ④).

In the field of atomistic modeling, a related approach termed “learning on-the-fly” has been developed by De Vita and Car [54], Csányi et al. [55], Li et al. [56] in which an interatomic potential is compared to the quantum-mechanical forces after a fixed number of time steps, and re-fitted if the potential’s forces become inaccurate. The main difference between these two approaches is that, in active learning, we usually *estimate* the accuracy of the interatomic potential *without additional quantum-mechanical calculations*—again, this is crucial here because computing quantum-mechanical forces explicitly is too expensive for large-scale simulations.

Active learning can be formulated as follows. Suppose we want to estimate whether an unknown configuration $\{r_i\}^*$ needs to be added to the training set. The measure based on which active learning makes this decision is called the *model uncertainty* $\gamma = \gamma(\{r_i\}^*)$, and the algorithm that computes γ is called “query strategy”. There are many different ways of formulating such a query strategy (cf., [57]) and the most popular ones that have been successfully adapted to work with MLIPs are described in the following.

Query-by-committee. The query-by-committee strategy used, e.g., by Behler [58] or Zhang et al. [59] for NNPs trains an ensemble of N MLIPs with different random initialization of the parameters, and measures the error by computing the variance between the different model predictions of, e.g., the total energy Π_j of a configuration $\{r_i\}^*$

$$\gamma_{\text{qbc}} = \frac{1}{N} \sum_{j=1}^N \left(\Pi_j(\{r_i\}^*) - \bar{\Pi}(\{r_i\}^*) \right)^2,$$

where $\bar{\Pi}$ is the mean energy. If the variance of one configuration exceeds some threshold (typically about a few meV per atom), this configuration is added to the training set. While being straightforward

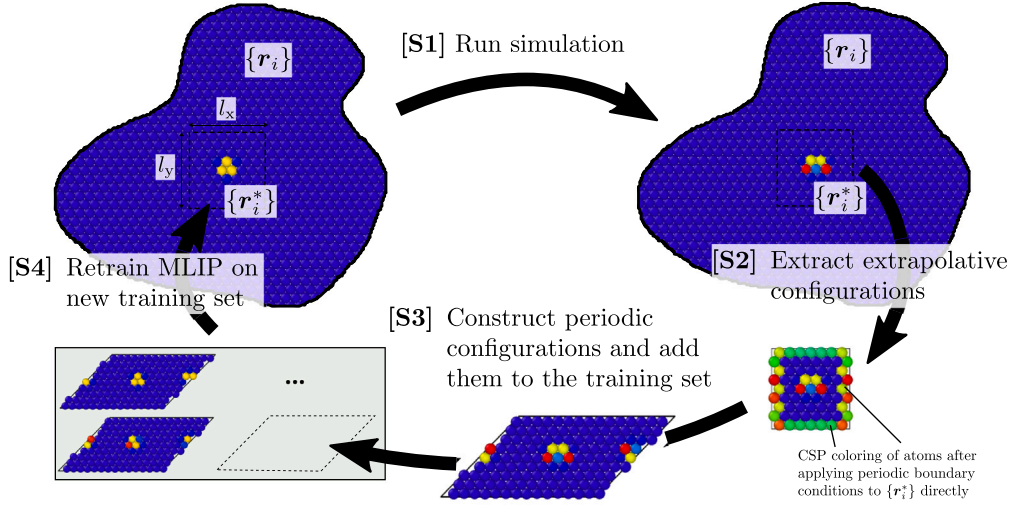


Fig. 7. Schematic illustration of the active learning algorithm for large-scale simulations proposed in [17].

Source: The figure is reproduced from [17] (Figure 1 therein) with permission from IOP.

to implement, the downside of query-by-committee is the necessity to train *and* evaluate several potentials which can become very costly when running large-scale molecular dynamics simulations.

Bayesian inference. A more efficient query strategy than query-by-committee that does not require to train an ensemble of models is Bayesian inference which is built into the Gaussian process-based potentials. For example, Jinnouchi et al. [60] and Vandermause et al. [61] have developed algorithms that compute the predictive variance for each neighborhood \mathcal{N}^* in $\{r_i\}^*$

$$\gamma_{\text{bi}}(\mathcal{N}^*) = k(\mathcal{N}^*, \mathcal{N}^*) - \underline{k}^T(\mathcal{N}^*, \underline{\mathcal{N}}) \cdot \left(\underline{k}^{-1}(\underline{\mathcal{N}}, \underline{\mathcal{N}}) \underline{k}(\underline{\mathcal{N}}, \mathcal{N}^*) \right), \quad (9)$$

where $\underline{\mathcal{N}}$ is the vector of neighborhoods contained in the training set, and \underline{k} is the covariance matrix (to which we may add some regularization). As for query-by-committee, if the maximum variance over all neighborhoods in the configuration exceeds some threshold of the order of one meV per atom, this configuration is added to the training set. Typically the total number of neighborhoods contained in the training set is too large for \underline{k}^{-1} being computable efficiently. Hence, only a representative subset of $\underline{\mathcal{N}}$ is used, which can be computed using matrix decomposition methods such as CUR/CGR [62].

D-optimality. For MTPs, Shapeev's group developed an algorithm similar to Bayesian inference based on the D-optimality criterion [63, 64]. D-optimality works as follows. Suppose for a moment that we have as many parameters as configurations in the training set, we then form the information matrix \underline{A} by computing the derivative of, e.g., the total energy of each configuration in the training set with respect to the parameters, i.e., $A_{jk} = \partial_{\theta_j} \Pi(\{r_i\}_k)$. The model uncertainty is then maximum change in the determinant of \underline{A} if $\{r_i\}^*$ would be added to the training set, which can be computed as

$$\gamma_{\text{do}} = \max_i \left(\underline{b}^T \underline{A}^{-1} \right)_i, \quad \text{where } b_j = \partial_{\theta_j} \Pi(\{r_i\}^*).$$

If the change in the determinant of \underline{A} is small, then $\gamma_{\text{do}} \approx 1$ and the MLIP is assumed to predict energies, forces, and stresses, of $\{r_i\}^*$ sufficiently well. Usually the MLIP's prediction become unreliable when $\gamma_{\text{do}} > 10$. For D-optimality, the model uncertainty is also called the *extrapolation grade* because, mathematically, γ_{do} has the meaning of an extrapolation. Usually we have many more training configurations than parameters, so, to form a quadratic \underline{A} , we select those configurations that maximize linear independence between its row vectors. An efficient algorithm to do this is the maxvol algorithm [65], which is also based on the CUR/CGR decomposition.

We remark that active learning can not only help to estimate the model uncertainty when running a simulation, but also when creating general training sets. A way to create better training sets with the help of active learning is by running molecular dynamics simulations on the already computed training configurations, and, while running the simulations, extract configurations with high uncertainty and add them to the training set. Montes De Oca Zapiain et al. [66], Oord et al. [67], Kulichenko et al. [68] have developed even more efficient algorithms that bias the dynamics towards regions with high uncertainty in order to further improve the efficiency of the training stage and the diversity of the training set.

3.3. Active learning during large-scale simulations

Active learning solves the problem of transferability when the number of atoms is small enough so that single-point DFT calculations, composing the training set, can be performed on the entire configuration. However, the situation becomes more complicated for large-scale simulations. In principle, we may still compute a per-atom model uncertainty and extract a small subset of atoms when the model uncertainty exceeds some threshold. However, DFT requires periodic boundary conditions to be imposed on the extracted configurations which creates artificial neighborhoods at the cell boundaries that do not occur in our large-scale simulation (cf., [S2] in Fig. 7).

To that end, we have developed an algorithm that converts the extracted configurations to periodic ones, before computing them with DFT and adding them to the training set, by mirroring the displacements at the periodic cell boundaries [17]. In [17], we have applied this algorithm to simulate screw dislocation motion in bcc tungsten, and very recently we have extended it to dissociated dislocations with arbitrary character angles in face-centered cubic metals [69]. Moreover, Zhang et al. [52] have shown that the same idea can also be applied to cracks, which also possess certain symmetries that allow one to construct DFT cells without too artificial neighborhoods (cf., Fig. 8).

Constructing training sets this way is supported by recent analysis of Ortner and Wang [72], Wang et al. [73] who proved that the MLIP's error with respect to DFT converges exponentially with the size of the training configurations (if the MLIP approximates the training set sufficiently well). Their training configurations were constructed using *periodic arrays* of screw dislocations, and this implies that, if the training configurations already *contain the impact of the exact boundary conditions*—as it is the case when training directly

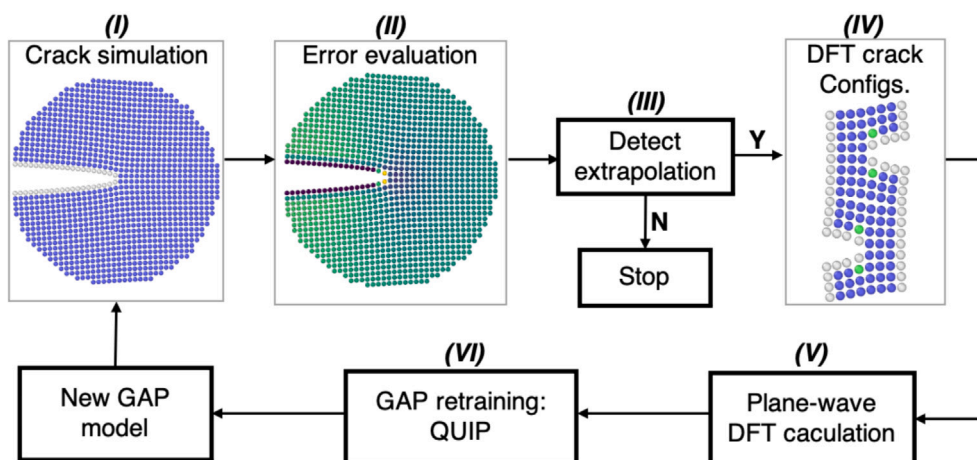


Fig. 8. Application of the active learning algorithm for large-scale simulations proposed in [17] to cracks. Source: The figure is reproduced from [52] (Figure 3 therein) with permission from the corresponding author.

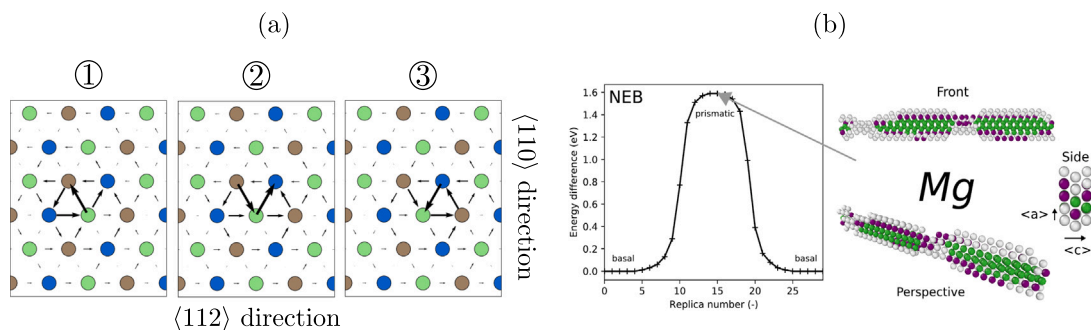


Fig. 9. (a) Visualization of the dislocation core structure while running the active learning algorithm from Fig. 7 using a differential displacement map at different stages during motion along the glide plane, i.e., ① before, ② during, and ③ after crossing the first Peierls barrier; thereby, the dislocation core structure correctly changes from easy to intermediate, and back again to easy, without admitting any artificial split-core configuration. The figure is reproduced from [17] (Figure 6 therein) with permission from IOP. (b) Transition path for prismatic slip in magnesium using an NNP [70]; during the transition, the dislocation does not dissociate consistent with experiments. Source: The figure is reproduced from [71] (graphical abstract) with permission from ACS.

on neighborhoods extracted from the large-scale simulations—then these training configurations should already compose a nearly optimal training set.

In principle, such an algorithm as shown in Fig. 7 allows to run molecular dynamics *completely autonomously* by letting active learning collect the right training data during the simulations. Nonetheless, while this approach appears to work well in situations when the extracted configurations are smooth at the periodic boundaries, it is still an open problem how to handle general situations when this is not the case. Of course, one can always compute clusters similar to those used in QM/MM [74], but this requires additional buffer and/or vacuum regions which heavily increases the computational cost over the mirroring approach. Moreover, with clusters, we may not be able to accurately train MLIPs on energies; although Grigorev et al. [75] have shown that this might not be a problem when the potential has already been sufficiently trained on energies contained in the initial training set. Another approach is to “squeeze” the clusters containing extrapolative neighborhoods into periodic configurations and optimize the atoms at the cell boundaries to mitigate artificial neighborhoods [76,77].

4. Now, what can machine-learning potentials do better than empirical potentials?

4.1. Predicting mechanisms

One of the shortcomings of using EIPs in atomistic simulations is their inability to make quantitative predictions in general. However, being quantitatively accurate is crucial for computer-aided materials design in order to predict trends between alloys (e.g., for energy barriers, or mechanical properties) and mechanisms. In the following, we give examples for a few success stories where MLIPs have advanced the state-of-the-art of atomistic simulations over EIPs.

One prominent example of EIPs showing artifacts in atomistic simulations is screw dislocation motion in bcc tungsten (in fact, in bcc transition metals in general); EIPs often predict metastable split-core configurations that are inconsistent with DFT. Our active learning algorithm (Section 3.3) was able to automatically train an MTP during screw dislocation motion, and this MTP retains the compact core structure during motion consistent with DFT (cf., Fig. 9(a)). Another example is prismatic slip in magnesium. Here, EIPs also tend to predict

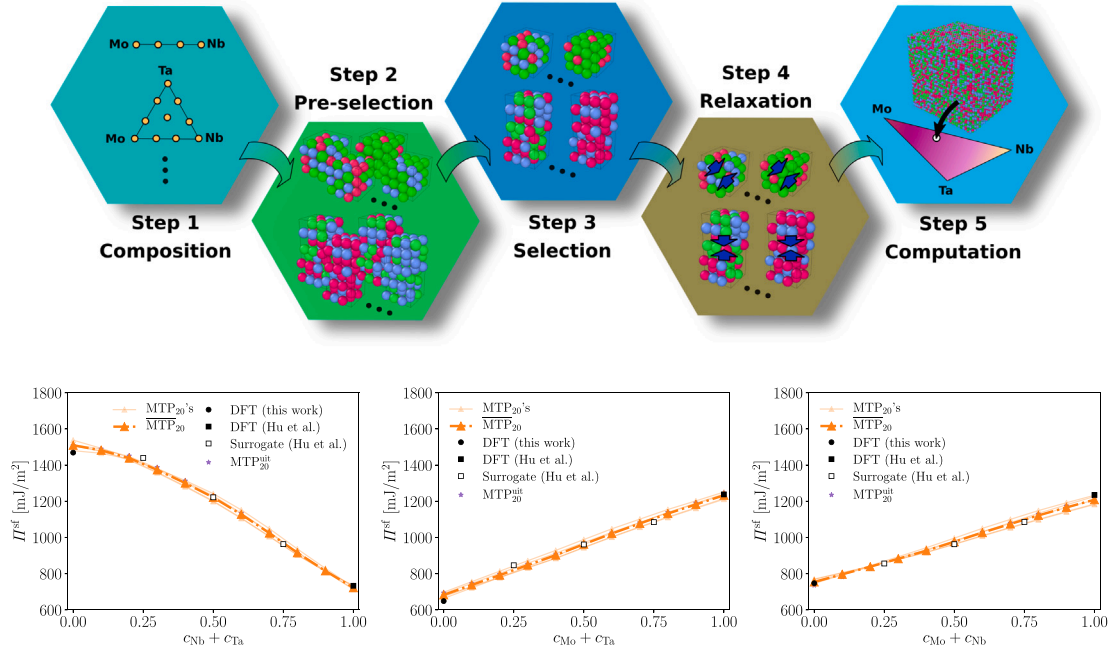


Fig. 10. (Top) Illustration of the training algorithm from [78] for constructing MLIPs for random alloys. (Bottom) Level-20 MTPs trained with this algorithm above predict the DFT stacking fault energy of Mo-Nb-Ta over the entire composition space practically without error. Source: Both figures are reproduced from [78] (Figure 2 & 10 therein) with permission from APS.

a dissociation of $\langle a \rangle$ basal screw dislocations, while Stricker et al. [70], Stricker and Curtin [71] have shown that NNPs predict a compact core structure that is consistent with DFT and real experiments (cf., Fig. 9(b)).

Numerous MLIPs have also been developed for alloys. For example, Li et al. [43], Yin et al. [79] developed SNAPs and MTPs for the Mo-Nb-Ta-W medium-entropy alloy and showed that, for this class of alloys, strength can be, unlike in unary bcc metals, controlled by edge dislocations, which has also been confirmed by real experiments [80]. Again, this would have likely been difficult to predict with EIPs since they often admit artificial split-core configurations (see above).

The list of examples could be continued for a while, but this would, however, go far beyond the scope of the present article. A more comprehensive list of state-of-the-art MLIPs for alloys can be found, e.g., in the review article of Wang et al. [20].

4.2. Predictive high-throughput screening of alloy properties or: DFT-accurate molecular dynamics à la carte

The previous examples mainly concern problems that require *general-purpose* potentials for large-scale simulations of microstructures, e.g., polycrystals. To find new alloys with exotic properties, it is, however, in general necessary to screen over an enormous number of compositions of the order of billions, or more, to find those compositions that meet a certain design criterion. Screening over such an amount of compositions is not possible using large-scale simulations and requires reduced models that depend on simpler variables, e.g., material properties, such as the elastic constants, cohesive energies, or misfit volumes.

To date, research on data mining has mainly been focused on developing machine learning models that approximate the material properties of alloys over the composition space to reduce the number of necessary DFT calculations (e.g., [81,82]). However, computing accurate average properties of, e.g., high-entropy alloys already requires ~ 30 DFT calculations. Including relaxation, the amount of single-point

DFT calculations likely sums up to already more than one thousand—for only one composition! Clearly, this approach is not efficient for computing material properties such as stacking fault energies that require ~ 100 atoms per cell. Moreover, it is *infeasible* for computing more involved properties such as dislocation-solute interaction energies that are typically required in strengthening models (cf., Varvenne et al. [83]), and, on top, investigating chemical ordering via Monte-Carlo simulations. To that end, a transfer learning approach seems preferable in which most of the DFT calculations can be replaced by a *special-purpose potential*, i.e., a potential whose only purpose is to predict one material property.

In [78], we have developed a training algorithm for constructing MTPs for random alloys using the example of unstable stacking fault energies in the bcc Mo-Nb-Ta medium-entropy alloy and shown that ~ 100 single-point DFT calculations suffice to predict the DFT reference results (see Fig. 10). In [84], we have applied this algorithm also to surface energies and shown that the algorithm predicts the alloy's ductility using a criterion [85] based on the Rice-Thomson model [86] with DFT accuracy, requiring only a few hundred single-point DFT calculations (including the stacking fault calculations). Moitzi et al. [87] combined the ductility model with a solute-strengthening model [83] into a Bayesian multi-objective optimization framework that allows to predict Pareto-optimal compositions, i.e., compositions that have optimal ductility-vs-strength ratios. Finding *all* Pareto-optimal compositions for a four-component high-entropy alloy only required ~ 700 – 800 single-point DFT calculations using our training algorithm, thus enabling now a tractable, ab initio-based high-throughput screening of Pareto-optimal multicomponent alloys over the entire composition space.

5. Getting started







5.1. MLIP software and training sets

Most of the implementations of the potentials that are featured in this article are available open source and the corresponding packages are listed in Table 1 (a). Typically, each of these packages contain at

Table 1

(a) Software packages that implement the MLIPs discussed in this work. (b) Training sets that have been constructed in the works discussed in the sections 2–4.

(a)			
Package name	MLIP	URL	
n2p2 [9,88]	NNP	github.com/CompPhysVienna/n2p2	
DeePMD-kit [89]	Deep NNP	github.com/deepmodeling/deepmd-kit	
Nequip [24]	CNN	github.com/mir-group/nequip	
QUIP [10,90]	GAP	github.com/libAtoms/QUIP	
FitSNAP [91]	SNAP	github.com/FitSNAP/FitSNAP	
MLIP-2 [92]/MLIP-3 [93]	MTP	gitlab.com/ashapeev/mlip-2/ gitlab.com/ashapeev/mlip-3	
pyace [30]	ACE	github.com/ICAMS/python-ace	
MACE [38]	message passing ACE	github.com/ACEsuit/mace	

(b)			
Material	Training set (i.a.)	Application(s)	Repository
α -Fe [49,52]	Crack tips	Dislocations [50], cracks [52]	
Mo-Nb-Ta-W [43]	MD snapshots	polycrystals [43], Dislocations [79]	
Mg [70]	Planar defects	Cracks [70], dislocations [71]	
Mo-Nb-Ta-V-W [44]	Liquid phases	Dislocation loops	
W-He [75]	Dislocations	Dislocation-He interaction	
Mg [53]	All crystal systems	Planar defects	

least the (low-level) implementation of the corresponding potential and functionalities to train it. Well then, which package should one use?

It is difficult to say which package to use as this heavily depends on the requirements of the user—and none of the packages contains all features discussed in Sections 2 and 3. As common features, all packages contain interfaces to DFT codes, such as VASP [94], in order to construct the training sets, and to LAMMPS [95] in order to use the potentials in large-scale atomistic simulations. NNPs and the polynomial potentials SNAP, MTP, and ACE, are very fast and have been mostly used for large-scale simulations of metals and their alloys (e.g., [43,70,79]). GAP has also been extensively applied to metals (e.g., [44,49]), but is generally reported to be slower than the other potentials (cf., Fig. 4); although there are ways to accelerate GAP (cf., [44,96]). Some packages implement more advanced mechanisms, such as interactions between atomic features using attention or message passing mechanisms (DeePMD, MACE), that can improve the performance of MLIPs. Some packages implement more advanced features, such as magnetic degrees of freedom (MLIP-3, pyace).

Moreover, it should be emphasized that *active learning* works strikingly well for polynomial potentials, possibly because extrapolation of raw polynomials is conceptually easy to define from a mathematical point of view. In particular, the D-optimal selection algorithm proposed by Podryabinkin and Shapeev [63] presently appears to be one of the most efficient *and* best-validated methods to build training sets for MTPs with a minimal amount of DFT calculations.

The available training sets that have been constructed in the works discussed in Sections 2–4 are listed in Table 1 (b). This list is, obviously, by no means comprehensive, but covers different *flavors* of how various groups approach the problem of generating transferable training sets. A more complete list of existing potentials and their corresponding training sets can be found in other recent and more focused review articles, e.g., by Freitas and Cao [97], or Wang et al. [20].

All of the author's contributions reported in Section 4 have been implemented using the MLIP-2 and MLIP-3 packages.

5.2. Best practices (from the author)

Each of the MLIP packages listed in Table 1 (a) contains tutorials and/or examples that should provide prospective users with a good starting point for “exploring MLIPs”. Below, we give some additional “tips and tricks” on how to overcome the first two obstacles when using MLIPs: how to select the hyperparameters and how to train a robust potential. For the third obstacle—how to *validate* results of a computational study using MLIPs—we refer to the recent work of Morrow et al. [98].

5.2.1. Selection of the hyperparameters

Several hyperparameters are common to all potentials, such as the cut-off radius, the choice of the radial basis, and the weighting/regularization of energies, forces, and stresses, in the loss function. Other common hyperparameters are the polynomial degree of the angular features (for a spherical harmonics basis this corresponds to the maximum number of angular momenta) and the body-order, except for SNAP where the body-order is fixed to four.¹ Hyperparameters that are unique to each of the individual potentials from Section 2 are briefly discussed in the following.

NNP. In principle, the hyperparameters of NNPs comprise everything what modern neural networks have to offer: the number of hidden layers, the number of neurons, the type of activation function(s), etc.

GAP. For GAP, we additionally have to choose the total number of kernels.

MTP. For MTPs, the polynomial degree and the body-order are increased comparably using one single hyperparameter called the *level* (cf., e.g., [39]).

PINN/ACE. Instead of using generic activation functions, Drautz [13] and Pun et al. [14] suggested to use the ACE basis or neural

¹ For NNPs, MTPs, ACE potentials, and ETNs, the body-order can be controlled by how often a feature tensor F is used as an input to the network (cf., Fig. 3); For GAP, the body-order can be controlled by using kernels with different ζ 's (cf., Eq. (4)).

networks as an input to cluster functional potentials, such as bond-order potentials, which can, therefore, be considered as physically motivated activation functions.

ETN. ETNs additionally parameterize the ranks of the order-3 tensors that compose the tensor network.

For some hyperparameters, there exist universally good choices that yield sufficiently accurate results for most applications. For example, for non-magnetic metals, the cut-off radius can safely be fixed to about two times the shortest lattice spacing. Further, a radial basis (e.g., Chebyshev or Bessel) of size 8–10 is usually a reasonable choice. Moreover, a relative weighting between energies and forces w_f/w_e in the range of 0.1–0.01, and between energies and stresses w_s/w_e in the range of 0.01–0.001, has been found to work well in practice; for more detailed information on how to judiciously choose the weights for various applications, we refer to, e.g., [92] (Section 2.2 therein).

For other hyperparameters, it may not be possible to find universally good values a priori because they depend more strongly on the training set. For example, it was found in [27] that molecules benefit from a higher body-order, while metallic alloys benefit from a higher polynomial degree. In this case, one may do a grid search to obtain the optimal hyperparameters (cf., [29]). However, for the next-generation neural network and tensor network potentials, a grid search is not efficient because these classes of potentials contain too many hyperparameters. For such MLIPs with many hyperparameters, it is more tractable to do a *greedy search* starting from a potential with low complexity, e.g., a pair potential, and minimize the loss function with respect to the computational cost using a stochastic gradient descent algorithm (cf., [27]).

5.2.2. Training protocol

Below, we exemplify the essential steps of a workflow for a computational study using MLIPs.

1. First analyze the physical problem to be solved. Which quantum effects are supposed to be relevant? All MLIPs presented in Section 2 can simulate multicomponent systems, some also have more advanced features that allow to generalize the model to simulate more complex quantum-mechanical effects (cf., Section 2.3.1).
2. Next construct the initial training set. To that end, first analyze the problem to be simulated. For example, if the problem consists of simulating predefined defects, it likely suffices to construct *special-purpose potentials*. Some algorithms that can do this fully automatically have been discussed in Sections 3.3 and 4.2. If the problem consists of simulating microstructures, the reader is referred to Section 3.1, which discusses some strategies to build *general-purpose potentials*.
3. Extract some configurations from the training set to be used as validation and test sets (usually the ratio should be approximately 70:15:15). Train the potential on the remaining configurations in the training set and optimize the hyperparameters on the test set until the error on the validation set is as small as desired. In practice, a potential with root-mean-square errors for energies of the order of 1 meV per atom and for forces of the order of 10 meV/Å is usually considered to be as good enough.
4. Start the simulation(s) and leave active learning switched on to measure the model uncertainties. Algorithms that can update the training set while running large-scale simulations if the model uncertainties become too high in certain regions have been discussed in Sections 3.2 and 3.3.

The last two steps should ideally be alternated during the entire lifetime of the potential to achieve the best possible accuracy; whether this is necessary depends on the application. For example, if the mechanism to be simulated is unknown in advance, relevant neighborhoods are likely not yet present in the initial training set and active learning becomes

inevitable. At first glance, this appears cumbersome, but note that this process can to large extent be automated.

6. Open problems

Given the vast amount of literature appearing on the topic of MLIPs, the present article might already be partially outdated by the time of its publication. However, there are certain challenges that are likely going to persist throughout the years to come. Some of the—in the author's view—most important ones are listed below.

Influence of magnetism. Most DFT studies are done for non-magnetic materials, however, magnetism can have a strong influence on the material properties, for example, on the stacking fault energy that influences plasticity on larger scales (see the study on face-centered cubic iron by Bleskov et al. [99]). While several MLIPs feature collinear spins, it is still an open question how to train such potentials efficiently. To that end, we have recently proposed a methodology for training magnetic MTPs [100] on constrained DFT [101] that allows us to use active learning on spins. Extending this methodology to paramagnetism and, moreover, *exploring* the impact of magnetism on material properties is still in its infancy.

Multi-defect interactions. It is not sufficiently well understood how to train a MLIP to be used for simulating multi-defect interactions, e.g., partial dislocations, dislocation-grain boundary interactions, etc. While different training strategies were presented in Section 3 that usually allow one to create robust MLIPs, quantifying the MLIP's uncertainty on configurations that contain multiple defects remains a challenge because one cannot afford even one single-point DFT calculation for such configurations. This is in particular challenging when far-field contributions to the atomic energies become important.

Database for training sets and benchmark problems. A successful database for EIPs is the OpenKIM project (www.openkim.org). MLIPs, however, follow a different philosophy in how they are created and used: instead of fixing the MLIP's parameters once and for all, they are continuously allowed to be refined if the MLIP is used in parts of the configurational space on which it has not been trained. Hence, what is now needed is a *database for training sets* rather than potentials. The first initiative in this direction is ColabFit Exchange [102]. In addition, to better compare MLIPs for applications in metallurgy requires curated datasets that cover the whole bandwidth of crystalline defects (vacancies, grain boundaries, dislocations, cracks, etc.). Creating such datasets for benchmarking potentials is a common practice in molecular modeling, but still lacking in computational metallurgy.

Automation of workflows. While many works have been devoted to developing methods that automate the construction of training sets, creating a workflow that fully automates molecular dynamics is still very challenging. For example, a scenario in which a potential is first pre-trained on some DFT data, and then on simulations by extracting any missing configurations while running the simulations, yields a highly heterogeneous workflow that simultaneously requires different levels of parallelism (e.g., for running the molecular dynamics simulations, for DFT, and for training the potential), different input parameters (e.g., the number of k-points), etc., all of which should be decided on fully automatically during the workflow. Another aspect that requires sophisticated workflows is integrated material modeling, that is, e.g., when combining models that predict alloys with an optimal strength-vs-ductility ratio, and models that predict the phase stability of those alloys.

Accuracy of DFT. While DFT is, with a bit of exaggeration, usually considered as the ground truth of atomic-scale modeling, it has certain deficiencies, e.g., in various popular DFT codes not predicting the experimental shear modulus for several transition metals (cf., [103]). Another property that requires further investigation is the Peierls stress in α -iron which DFT predicts to be 2–3 times higher than the experimental estimates, although we remark that it is still unclear whether

this difference is due to DFT itself, or due to the assumption of computing the Peierls as the onset of motion of a single dislocation in an effectively infinite crystal (cf., [104]). Instead of attempting to improve the accuracy of DFT, another possibility could be mixing DFT with more accurate data coming from all-electron quantum-mechanical calculations or experiments.

CRediT authorship contribution statement

M. Hodapp: Conceptualization, Investigation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

I would like to thank my advisers Felix Fritzen, Bill Curtin, Guillaume Anciaux, and Alex Shapeev, without whose support I would not have come to the point of writing this article.

Writing of the present article is financially supported under the scope of the COMET program within the K2 Center “Integrated Computational Material, Process and Product Engineering (IC-MPPE)” (Project No. 886385), which is highly acknowledged. This program is supported by the Austrian Federal Ministries for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) and for Labour and Economy (BMAW), represented by the Austrian Research Promotion Agency (FFG), and the federal states of Styria, Upper Austria and Tyrol.

References

- [1] E.B. Tadmor, R.E. Miller, *Modeling Materials: Continuum, Atomistic, and Multiscale Techniques*, Cambridge University Press, 2011, OCLC: ocn751740304.
- [2] C.H. Zenk, L. Feng, D. McAllister, Y. Wang, M.J. Mills, Shearing mechanisms of co-precipitates in IN718, *Acta Mater.* 220 (2021) 117305, <http://dx.doi.org/10.1016/j.actamat.2021.117305>.
- [3] P.E. Blöchl, Projector augmented-wave method, *Phys. Rev. B* 50 (24) (1994) 17953–17979, <http://dx.doi.org/10.1103/PhysRevB.50.17953>.
- [4] F. Ercolessi, J.B. Adams, Interatomic potentials from first-principles calculations: The force-matching method, *Europhys. Lett. (EPL)* 26 (8) (1994) 583–588, <http://dx.doi.org/10.1209/0295-5075/26/8/005>.
- [5] T.J. Lenosky, B. Sadigh, E. Alonso, V.V. Bulatov, T.D.D.L. Rubia, J. Kim, A.F. Voter, J.D. Kress, Highly optimized empirical potential model of silicon, *Modelling Simul. Mater. Sci. Eng.* 8 (6) (2000) 825–841, <http://dx.doi.org/10.1088/0965-0393/8/6/305>.
- [6] A.C.T.V. Duin, S. Dasgupta, F. Lortat, W.A. Goddard, Reaxff: A reactive force field for hydrocarbons, *J. Phys. Chem. A* 105 (41) (2001) 9396–9409, <http://dx.doi.org/10.1021/jp004368u>.
- [7] Q. Zhang, T. Çağın, A. Van Duin, W.A. Goddard, Y. Qi, L.G. Hector, Adhesion and nonwetting-wetting transition in the Al / α - Al₂O₃ interface, *Phys. Rev. B* 69 (4) (2004) 045423, <http://dx.doi.org/10.1103/PhysRevB.69.045423>.
- [8] O.T. Unke, S. Chmiela, H.E. Sauceda, M. Gastegger, I. Poltavsky, K.T. Schütt, A. Tkatchenko, K.-R. Müller, Machine learning force fields, *Chem. Rev.* 121 (16) (2021) 10142–10186, <http://dx.doi.org/10.1021/acs.chemrev.0c01111>.
- [9] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, *Phys. Rev. Lett.* 98 (14) (2007) 146401, <http://dx.doi.org/10.1103/PhysRevLett.98.146401>.
- [10] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons, *Phys. Rev. Lett.* 104 (13) (2010) 136403, <http://dx.doi.org/10.1103/PhysRevLett.104.136403>.
- [11] A. Thompson, L. Swiler, C. Trott, S. Foiles, G. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, *J. Comput. Phys.* 285 (2015) 316–330, <http://dx.doi.org/10.1016/j.jcp.2014.12.018>.
- [12] A.V. Shapeev, Moment tensor potentials: A class of systematically improvable interatomic potentials, *Multiscale Model. Simul.* 14 (3) (2016) 1153–1173, <http://dx.doi.org/10.1137/15M1054183>.
- [13] R. Drautz, Atomic cluster expansion for accurate and transferable interatomic potentials, *Phys. Rev. B* 99 (1) (2019) 014104, <http://dx.doi.org/10.1103/PhysRevB.99.014104>.
- [14] G.P.P. Pun, R. Batra, R. Ramprasad, Y. Mishin, Physically informed artificial neural networks for atomistic modeling of materials, *Nature Commun.* 10 (1) (2019) 2339, <http://dx.doi.org/10.1038/s41467-019-10343-5>.
- [15] M.-C. Marinica, L. Ventelon, M.R. Gilbert, L. Provile, S.L. Dudarev, J. Marian, G. Bencteux, F. Willaime, Interatomic potentials for modelling radiation defects and dislocations in tungsten, *J. Phys.: Condens. Matter* 25 (39) (2013) 395502, <http://dx.doi.org/10.1088/0953-8984/25/39/395502>.
- [16] W.J. Szlachta, A.P. Bartók, G. Csányi, Accuracy and transferability of Gaussian approximation potential models for tungsten, *Phys. Rev. B* 90 (10) (2014) <http://dx.doi.org/10.1103/PhysRevB.90.104108>.
- [17] M. Hodapp, A. Shapeev, In operando active learning of interatomic interaction during large-scale simulations, *Mach. Learn.: Sci. Technol.* 1 (4) (2020) 045005, <http://dx.doi.org/10.1088/2632-2153/aba373>.
- [18] F. Musil, A. Grisafi, A.P. Bartók, C. Ortner, G. Csányi, M. Ceriotti, Physics-inspired structural representations for molecules and materials, *Chem. Rev.* 121 (16) (2021) 9759–9815, <http://dx.doi.org/10.1021/acs.chemrev.1c00021>, [arXiv:2101.04673](https://arxiv.org/abs/2101.04673).
- [19] A. Shapeev, K. Gubaev, E. Tsybalov, E. Podryabinkin, Active learning and uncertainty estimation, in: K.T. Schütt, S. Chmiela, O.A. von Lilienfeld, A. Tkatchenko, K. Tsuda, K.-R. Müller (Eds.), *Machine Learning Meets Quantum Physics*, Springer International Publishing, 2020, pp. 309–329, http://dx.doi.org/10.1007/978-3-030-40245-7_15.
- [20] F. Wang, H.-H. Wu, L. Dong, G. Pan, X. Zhou, S. Wang, R. Guo, G. Wu, J. Gao, F.-Z. Dai, X. Mao, Atomic-scale simulations in multi-component alloys and compounds: A review on advances in interatomic potential, *J. Materials Science & Technology* 165 (2023) 49–65, <http://dx.doi.org/10.1016/j.jmst.2023.05.010>.
- [21] K.T. Schütt, P.-J. Kindermans, H.E. Sauceda, S. Chmiela, A. Tkatchenko, K.-R. Müller, SchNet: A continuous-filter convolutional neural network for modeling quantum interactions, in: *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS '17*, Curran Associates Inc., 2017, pp. 992–1002, event-place: Long Beach, California, USA.
- [22] J.S. Smith, O. Isayev, A.E. Roitberg, ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost, *Chem. Sci.* 8 (4) (2017) 3192–3203, <http://dx.doi.org/10.1039/C6SC05720A>.
- [23] L. Zhang, J. Han, H. Wang, R. Car, W. E, Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics, *Phys. Rev. Lett.* 120 (14) (2018) 143001, <http://dx.doi.org/10.1103/PhysRevLett.120.143001>.
- [24] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J.P. Mailoa, M. Kornbluth, N. Molinari, T.E. Smidt, B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, *Nature Commun.* 13 (1) (2022) 2453, <http://dx.doi.org/10.1038/s41467-022-29939-5>.
- [25] D.A. Varshalovich, A.N. Moskalev, V.K. Khersonskii, *Quantum Theory of Angular Momentum*, World Scientific, 1988, <http://dx.doi.org/10.1142/0270>.
- [26] A.P. Bartók, R. Kondor, G. Csányi, On representing chemical environments, *Phys. Rev. B* 87 (18) (2013) 184115, <http://dx.doi.org/10.1103/PhysRevB.87.184115>.
- [27] M. Hodapp, A. Shapeev, Equivariant tensor networks, (arXiv:2304.08226) 2023, [arXiv:2304.08226](https://arxiv.org/abs/2304.08226).
- [28] J.C. Bridgeman, C.T. Chubb, Hand-waving and interpretive dance: An introductory course on tensor networks, *J. Phys. A: Math. Theoret.* 50 (22) (2017) 223001, <http://dx.doi.org/10.1088/1751-8121/aa6dc3>.
- [29] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A.V. Shapeev, A.P. Thompson, M.A. Wood, S.P. Ong, Performance and cost assessment of machine learning interatomic potentials, *J. Phys. Chem. A* 124 (4) (2020) 731–745, <http://dx.doi.org/10.1021/acs.jpca.9b08723>.
- [30] Y. Lysogorskiy, C.v.d. Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammer-schmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, R. Drautz, Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon, *npj Comput. Mater.* 7 (1) (2021) 97, <http://dx.doi.org/10.1038/s41524-021-00559-9>.
- [31] I. Novikov, B. Grabowski, F. Körmann, A. Shapeev, Magnetic moment tensor potentials for collinear spin-polarized materials reproduce different magnetic states of bcc Fe, *npj Comput. Mater.* 8 (1) (2022) 13, <http://dx.doi.org/10.1038/s41524-022-00696-9>.
- [32] R. Drautz, Atomic cluster expansion of scalar, vectorial, and tensorial properties including magnetism and charge transfer, *Phys. Rev. B* 102 (2) (2020) 024104, <http://dx.doi.org/10.1103/PhysRevB.102.024104>.
- [33] M. Rinaldi, M. Mrovec, A. Bochkarev, Y. Lysogorskiy, R. Drautz, Non-collinear magnetic atomic cluster expansion for iron, (arXiv:2305.15137) 2023, [http://arxiv.org/abs/2305.15137](https://arxiv.org/abs/2305.15137) [cond-mat].
- [34] Y. Zhang, C. Gao, Q. Liu, L. Zhang, H. Wang, M. Chen, Warm dense matter simulation via electron temperature dependent deep potential molecular dynamics, *Phys. Plasmas* 27 (12) (2020) 122704, <http://dx.doi.org/10.1063/5.0023265>.

- [35] J.A. Ellis, L. Fiedler, G.A. Popoola, N.A. Modine, J.A. Stephens, A.P. Thompson, A. Cangi, S. Rajamanickam, Accelerating finite-temperature Kohn-Sham density functional theory with deep neural networks, *Phys. Rev. B* 104 (3) (2021) 035120, <http://dx.doi.org/10.1103/PhysRevB.104.035120>.
- [36] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, in: *Proceedings of the 34th International Conference on Machine Learning - Vol. 70, ICML '17, JMLR.org*, 2017, pp. 1263–1272, Place: Sydney, NSW, Australia Number of pages: 10.
- [37] S. Takamoto, S. Izumi, J. Li, TeaNet: Universal neural network interatomic potential inspired by iterative electronic relaxations, *Comput. Mater. Sci.* 207 (2022) 111280, <http://dx.doi.org/10.1016/j.commatsci.2022.111280>.
- [38] I. Batatia, D.P. Kovacs, G. Simm, C. Ortner, G. Csányi, MACE: Higher order equivariant message passing neural networks for fast and accurate force fields, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), *Advances in Neural Information Processing Systems*, Vol. 35, Curran Associates, Inc., 2022, pp. 11423–11436.
- [39] K. Gubaev, E.V. Podryabinkin, G.L. Hart, A.V. Shapeev, Accelerating high-throughput searches for new alloys with active learning of interatomic potentials, *Comput. Mater. Sci.* 156 (2019) 148–156, <http://dx.doi.org/10.1016/j.commatsci.2018.09.031>.
- [40] N. Lopanitsyna, G. Fraux, M.A. Springer, S. De, M. Ceriotti, Modeling high-entropy transition metal alloys with alchemical compression, *Phys. Rev. Mater.* 7 (4) (2023) 045802, <http://dx.doi.org/10.1103/PhysRevMaterials.7.045802>.
- [41] J.P. Darby, J.R. Kermode, G. Csányi, Compressing local atomic neighbourhood descriptors, *npj Comput. Mater.* 8 (1) (2022) 166, <http://dx.doi.org/10.1038/s41524-022-00847-y>.
- [42] J.P. Darby, D.P. Kovács, I. Batatia, M.A. Caro, G.L.W. Hart, C. Ortner, G. Csányi, Tensor-reduced atomic density representations, *Phys. Rev. Lett.* 131 (2) (2023) 028001, <http://dx.doi.org/10.1103/PhysRevLett.131.028001>.
- [43] X.-G. Li, C. Chen, H. Zheng, Y. Zuo, S.P. Ong, Complex strengthening mechanisms in the NbMoTaW multi-principal element alloy, *npj Comput. Mater.* 6 (1) (2020) 70, <http://dx.doi.org/10.1038/s41524-020-0339-0>.
- [44] J. Byggmästar, K. Nordlund, F. Djurabekova, Modeling refractory high-entropy alloys with efficient machine-learned interatomic potentials: defects and segregation, *Phys. Rev. B* 104 (10) (2021) 104101, <http://dx.doi.org/10.1103/PhysRevB.104.104101>, [cond-mat.physics](https://arxiv.org/abs/cond-mat.physics).
- [45] A. Cichocki, N. Lee, I.V. Oseledets, A.-H. Phan, Q. Zhao, D. Mandic, Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: Perspectives and challenges PART 1, *Found. Trends® Mach. Learn.* 9 (4) (2016) 249–429, <http://dx.doi.org/10.1561/22000000059>, [arXiv:1609.00893](https://arxiv.org/abs/1609.00893).
- [46] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I.V. Oseledets, M. Sugiyama, D. Mandic, Tensor networks for dimensionality reduction and large-scale optimizations. Part 2 applications and future perspectives, *Found. Trends® Mach. Learn.* 9 (6) (2017) 249–429, <http://dx.doi.org/10.1561/22000000067>, [arXiv:1708.09165](https://arxiv.org/abs/1708.09165).
- [47] R. Orús, Tensor networks for complex quantum systems, *Nat. Rev. Phys.* 1 (9) (2019) 538–550, <http://dx.doi.org/10.1038/s42254-019-0086-7>.
- [48] D.E. Sommer, S.T. Dunham, Entangling solid solutions: Machine learning of tensor networks for materials property prediction, (arXiv:2203.09613) 2022, [cond-mat.physics](https://arxiv.org/abs/cond-mat.physics).
- [49] D. Dragoni, T.D. Daff, G. Csányi, N. Marzari, Achieving DFT accuracy with a machine-learning interatomic potential: Thermomechanics and defects in bcc ferromagnetic iron, *Phys. Rev. Mater.* 2 (1) (2018) <http://dx.doi.org/10.1103/PhysRevMaterials.2.013808>.
- [50] F. Maresca, D. Dragoni, G. Csányi, N. Marzari, W.A. Curtin, Screw dislocation structure and mobility in body centered cubic Fe predicted by a Gaussian approximation potential, *npj Comput. Mater.* 4 (1) (2018) 69, <http://dx.doi.org/10.1038/s41524-018-0125-4>.
- [51] M.R. Feller, A.M.Z. Tan, L.G. Hector, D.R. Trinkle, Geometries of edge and mixed dislocations in bcc Fe from first-principles calculations, *Phys. Rev. Mater.* 2 (11) (2018) <http://dx.doi.org/10.1103/PhysRevMaterials.2.113605>.
- [52] L. Zhang, G. Csányi, E. van der Giessen, F. Maresca, Atomistic fracture in bcc iron revealed by active learning of Gaussian approximation potential, (arXiv:2208.05912) 2022, [cond-mat](https://arxiv.org/abs/cond-mat).
- [53] M. Poul, L. Huber, E. Bitzek, J. Neugebauer, Systematic atomic structure datasets for machine learning potentials: Application to defects in magnesium, *Phys. Rev. B* 107 (10) (2023) 104103, <http://dx.doi.org/10.1103/PhysRevB.107.104103>.
- [54] A. De Vita, R. Car, A novel scheme for accurate md simulations of large systems, *MRS Proc.* 491 (1997) 473, <http://dx.doi.org/10.1557/PROC-491-473>.
- [55] G. Csányi, T. Albaret, M.C. Payne, A. De Vita, “Learn on the fly”: A hybrid classical and quantum-mechanical molecular dynamics simulation, *Phys. Rev. Lett.* 93 (17) (2004) <http://dx.doi.org/10.1103/PhysRevLett.93.175503>.
- [56] Z. Li, J.R. Kermode, A. De Vita, Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces, *Phys. Rev. Lett.* 114 (9) (2015) <http://dx.doi.org/10.1103/PhysRevLett.114.096405>.
- [57] B. Settles, Active learning literature survey, 2010, p. 67.
- [58] J. Behler, Representing potential energy surfaces by high-dimensional neural network potentials, *J. Phys.: Condens. Matter* 26 (18) (2014) 183001, <http://dx.doi.org/10.1088/0953-8984/26/18/183001>.
- [59] L. Zhang, D.-Y. Lin, H. Wang, R. Car, W. E, Active learning of uniformly accurate inter-atomic potentials for materials simulation, *Phys. Rev. Mater.* 3 (2) (2019) 023804, <http://dx.doi.org/10.1103/PhysRevMaterials.3.023804>, [arXiv:1810.11890](https://arxiv.org/abs/1810.11890).
- [60] R. Jinnouchi, F. Karsai, G. Kresse, On-the-fly machine learning force field generation: Application to melting points, *Phys. Rev. B* 100 (1) (2019) 014105, <http://dx.doi.org/10.1103/PhysRevB.100.014105>.
- [61] J. Vandermause, S.B. Torrisi, S. Batzner, et al., On-the-fly active learning of inter-pretatable bayesian force fields for atomistic rare events, *npj Comput. Mater.* 6 (1) (2020) 20, <http://dx.doi.org/10.1038/s41524-020-0283-z>.
- [62] S.A. Goreinov, E.E. Tyrtshnikov, N.L. Zamarashkin, A theory of pseudoskeleton approximations, *Linear Algebra Appl.* 261 (1997) 1–21.
- [63] E.V. Podryabinkin, A.V. Shapeev, Active learning of linearly parametrized interatomic potentials, *Comput. Mater. Sci.* 140 (2017) 171–180, <http://dx.doi.org/10.1016/j.commatsci.2017.08.031>.
- [64] K. Gubaev, E.V. Podryabinkin, A.V. Shapeev, Machine learning of molecular properties: locality and active learning, *J. Chem. Phys.* 148 (24) (2018) 241727, <http://dx.doi.org/10.1063/1.5005095>, [arXiv:1709.07082](https://arxiv.org/abs/1709.07082).
- [65] S.A. Goreinov, I.V. Oseledets, D.V. Savostyanov, E.E. Tyrtshnikov, N.L. Zamarashkin, How to find a good submatrix, in: *Matrix Methods: Theory, Algorithms and Applications*, World scientific, 2010, pp. 247–256, http://dx.doi.org/10.1142/9789812836021_0015.
- [66] D. Montes De Oca Zapiani, M.A. Wood, N. Lubbers, C.Z. Pereyra, A.P. Thompson, D. Perez, Training data selection for accuracy and transferability of interatomic potentials, *npj Comput. Mater.* 8 (1) (2022) 189, <http://dx.doi.org/10.1038/s41524-022-00872-x>.
- [67] C.V.D. Oord, M. Sachs, D.P. Kovács, C. Ortner, G. Csányi, Hyperactive learning for data-driven interatomic potentials, *npj Comput. Mater.* 9 (1) (2023) 168, <http://dx.doi.org/10.1038/s41524-023-01104-6>.
- [68] M. Kulichenko, K. Barros, N. Lubbers, Y.W. Li, R. Messerly, S. Tretiak, J.S. Smith, B. Nebgen, Uncertainty-driven dynamics for active learning of interatomic potentials, *Nature Comput. Sci.* 3 (3) (2023) 230–239, <http://dx.doi.org/10.1038/s43588-023-00406-5>.
- [69] L. Mismetti, M. Hodapp, Automated ab initio-accurate atomistic simulations of dissociated dislocations, (arXiv:2311.01830) 2023, [cond-mat.physics](https://arxiv.org/abs/cond-mat.physics).
- [70] M. Stricker, B. Yin, E. Mak, W.A. Curtin, Machine learning for metallurgy II. A neural-network potential for magnesium, *Phys. Rev. Mater.* 4 (10) (2020) 103602, <http://dx.doi.org/10.1103/PhysRevMaterials.4.103602>.
- [71] M. Stricker, W.A. Curtin, Prismatic slip in magnesium, *J. Phys. Chem. C* 124 (49) (2020) 27230–27240, <http://dx.doi.org/10.1021/acs.jpcc.0c09665>.
- [72] C. Ortner, Y. Wang, A framework for a generalization analysis of machine-learned interatomic potentials, *Multiscale Model. Simul.* 21 (3) (2023) 1053–1080, <http://dx.doi.org/10.1137/22M152267X>.
- [73] Y. Wang, S. Patel, C. Ortner, A theoretical case study of the generalisation of machine-learned potentials, (arXiv:2311.01664) 2023, [physics](https://arxiv.org/abs/physics).
- [74] E.V. Podryabinkin, A.G. Kvashnin, M. Asgarpour, I.I. Maslennikov, D.A. Ovsyannikov, P.B. Sorokin, M.Y. Popov, A.V. Shapeev, Nanohardness from first principles with active learning on atomic environments, *J. Chem. Theory Comput.* 18 (2) (2022) 1109–1121, <http://dx.doi.org/10.1021/acs.jctc.1c00783>.
- [75] P. Grigorev, A.M. Goryaeva, M.-C. Marinica, J.R. Kermode, T.D. Swinburne, Calculation of dislocation binding to helium-vacancy defects in tungsten using hybrid ab initio-machine learning methods, *Acta Mater.* 247 (2023) 118734, <http://dx.doi.org/10.1016/j.actamat.2023.118734>.
- [76] L.C. Erhard, J. Rohrer, K. Albe, V.L. Deringer, Modelling atomic and nanoscale structure in the silicon-oxygen system through active machine learning, (arXiv:2309.03587) 2023, [cond-mat](https://arxiv.org/abs/cond-mat).
- [77] F.N. Jalilov, E.V. Podryabinkin, A.R. Oganov, A.V. Shapeev, A.G. Kvashnin, Mechanical properties of single and polycrystalline solids from machine learning, (arXiv:2309.15868) 2023, [cond-mat.physics](https://arxiv.org/abs/cond-mat.physics).
- [78] M. Hodapp, A. Shapeev, Machine-learning potentials enable predictive and tractable high-throughput screening of random alloys, *Phys. Rev. Mater.* 5 (11) (2021) 113802, <http://dx.doi.org/10.1103/PhysRevMaterials.5.113802>.
- [79] S. Yin, Y. Zuo, A. Abu-Odeh, H. Zheng, X.-G. Li, J. Ding, S.P. Ong, M. Asta, R.O. Ritchie, Atomistic simulations of dislocation mobility in refractory high-entropy alloys and the effect of chemical short-range order, *Nature Commun.* 12 (1) (2021) 4873, <http://dx.doi.org/10.1038/s41467-021-25134-0>.
- [80] C. Lee, F. Maresca, R. Feng, Y. Chou, T. Ungar, M. Widom, K. An, J.D. Poplawsky, Y.-C. Chou, P.K. Liaw, W.A. Curtin, Strength can be controlled by edge dislocations in refractory high-entropy alloys, *Nature Commun.* 12 (1) (2021) 5474, <http://dx.doi.org/10.1038/s41467-021-25807-w>.
- [81] Y.-J. Hu, A. Sundar, S. Ogata, L. Qi, Screening of generalized stacking fault energies, surface energies and intrinsic ductile potency of refractory multicomponent alloys, *Acta Mater.* 210 (2021) 116800, <http://dx.doi.org/10.1016/j.actamat.2021.116800>.
- [82] C. Tandoc, Y.-J. Hu, L. Qi, P.K. Liaw, Mining of lattice distortion, strength, and intrinsic ductility of refractory high entropy alloys, *npj Comput. Mater.* 9 (1) (2023) 53, <http://dx.doi.org/10.1038/s41524-023-00993-x>.
- [83] C. Varvenne, A. Luque, W.A. Curtin, Theory of strengthening in fcc high entropy alloys, *Acta Mater.* 118 (2016) 164–176, <http://dx.doi.org/10.1016/j.actamat.2016.07.040>.

- [84] I. Novikov, O. Kovalyova, A. Shapeev, M. Hodapp, AI-accelerated materials informatics method for the discovery of ductile alloys, *J. Mater. Res.* 37 (21) (2022) 3491–3504, <http://dx.doi.org/10.1557/s43578-022-00783-z>.
- [85] E. Mak, B. Yin, W. Curtin, A ductility criterion for bcc high entropy alloys, *J. Mech. Phys. Solids* 152 (2021) 104389, <http://dx.doi.org/10.1016/j.jmps.2021.104389>.
- [86] J.R. Rice, Dislocation nucleation from a crack tip: An analysis based on the peierls concept, *J. Mech. Phys. Solids* 40 (2) (1992) 239–271, [http://dx.doi.org/10.1016/S0022-5096\(05\)80012-2](http://dx.doi.org/10.1016/S0022-5096(05)80012-2).
- [87] F. Moitzi, L. Romaner, M. Hodapp, A.V. Ruban, O.E. Peil, Ab initio framework for deciphering trade-off relationships in multi-component alloys, (arXiv:2311.12642) 2023, [cond-mat, physics:physics](https://arxiv.org/abs/2311.12642).
- [88] A.M. Tokita, J. Behler, How to train a neural network potential, *J. Chem. Phys.* 159 (12) (2023) 121501, <http://dx.doi.org/10.1063/5.0160326>.
- [89] J. Zeng, D. Zhang, D. Lu, P. Mo, Z. Li, Y. Chen, M. Rynik, L. Huang, Z. Li, S. Shi, Y. Wang, H. Ye, P. Tuo, J. Yang, Y. Ding, Y. Li, D. Tisi, Q. Zeng, H. Bao, Y. Xia, J. Huang, K. Muraoka, Y. Wang, J. Chang, F. Yuan, S.L. Bore, C. Cai, Y. Lin, B. Wang, J. Xu, J.-X. Zhu, C. Luo, Y. Zhang, R.E.A. Goodall, W. Liang, A.K. Singh, S. Yao, J. Zhang, R. Wentzcovitch, J. Han, J. Liu, W. Jia, D.M. York, W. E, R. Car, L. Zhang, H. Wang, DeepMD-kit v2: A software package for deep potential models, *J. Chem. Phys.* 159 (5) (2023) 054801, <http://dx.doi.org/10.1063/5.0155600>.
- [90] S. Klawohn, J.P. Darby, J.R. Kermode, G. Csányi, M.A. Caro, A.P. Bartók, Gaussian approximation potentials: Theory, software implementation and application examples, *J. Chem. Phys.* 159 (17) (2023) 174108, <http://dx.doi.org/10.1063/5.0160898>.
- [91] A. Rohskopf, C. Sievers, N. Lubbers, M.A. Cusentino, J. Goff, J. Janssen, M. McCarthy, D.M. De Oca Zapiain, S. Nikolov, K. Sargsyan, D. Sema, E. Sikorski, L. Williams, A.P. Thompson, M.A. Wood, FitSNAP: Atomistic machine learning with LAMMPS, *J. Open Source Softw.* 8 (84) (2023) 5118, <http://dx.doi.org/10.21105/joss.05118>.
- [92] I.S. Novikov, K. Gubaev, E.V. Podryabinkin, A.V. Shapeev, The MLIP package: moment tensor potentials with MPI and active learning, *Mach. Learn.: Sci. Technol.* 2 (2) (2021) 025002, <http://dx.doi.org/10.1088/2632-2153/abc9fe>.
- [93] E. Podryabinkin, K. Garifullin, A. Shapeev, I. Novikov, MLIP-3: Active learning on atomic environments with moment tensor potentials, *J. Chem. Phys.* 159 (8) (2023) 084112, <http://dx.doi.org/10.1063/5.0155887>.
- [94] G. Kresse, J. Furthmüller, Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set, *Phys. Rev. B* 54 (16) (1996) 11169–11186, <http://dx.doi.org/10.1103/PhysRevB.54.11169>.
- [95] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. In 'T Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, R. Shan, M.J. Stevens, J. Tranchida, C. Trott, S.J. Plimpton, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Comput. Phys. Comm.* 271 (2022) 108171, <http://dx.doi.org/10.1016/j.cpc.2021.108171>.
- [96] A. Glielmo, C. Zeni, A. De Vita, Efficient nonparametric *n*-body force fields from machine learning, *Phys. Rev. B* 97 (18) (2018) 184307, <http://dx.doi.org/10.1103/PhysRevB.97.184307>.
- [97] R. Freitas, Y. Cao, Machine-learning potentials for crystal defects, *MRS Commun.* 12 (5) (2022) 510–520, <http://dx.doi.org/10.1557/s43579-022-00221-5>.
- [98] J.D. Morrow, J.L.A. Gardner, V.L. Deringer, How to validate machine-learned interatomic potentials, *J. Chem. Phys.* 158 (12) (2023) 121501, <http://dx.doi.org/10.1063/5.0139611>.
- [99] I. Bleskov, T. Hickel, J. Neugebauer, A. Ruban, Impact of local magnetism on stacking fault energies: A first-principles investigation for fcc iron, *Phys. Rev. B* 93 (21) (2016) 214115, <http://dx.doi.org/10.1103/PhysRevB.93.214115>.
- [100] A.S. Kotykhov, K. Gubaev, M. Hodapp, C. Tantardini, A.V. Shapeev, I.S. Novikov, Constrained DFT-based magnetic machine-learning potentials for magnetic alloys: A case study of Fe–Al, *Sci. Rep.* 13 (1) (2023) 19728, <http://dx.doi.org/10.1038/s41598-023-46951-x>.
- [101] X. Gonze, B. Seddon, J.A. Elliott, C. Tantardini, A.V. Shapeev, Constrained density functional theory: A potential-based self-consistency approach, *J. Chem. Theory Comput.* 18 (10) (2022) 6099–6110, <http://dx.doi.org/10.1021/acs.jctc.2c00673>.
- [102] J.A. Vita, E.G. Fuemmeler, A. Gupta, G.P. Wolfe, A.Q. Tao, R.S. Elliott, S. Martiniani, E.B. Tadmor, ColabFit exchange: Open-access datasets for data-driven interatomic potentials, *J. Chem. Phys.* 159 (15) (2023) 154802, <http://dx.doi.org/10.1063/5.0163882>.
- [103] X. Gao, F. Tian, H. Song, Tests on the accuracy and scalability of the full-potential DFT method based on multiple scattering theory, *Front. Chem.* 8 (2020) 12.
- [104] R. Freitas, M. Asta, V.V. Bulatov, Quantum effects on dislocation motion from ring-polymer molecular dynamics, *npj Comput. Mater.* 4 (1) (2018) 55, <http://dx.doi.org/10.1038/s41524-018-0112-9>.