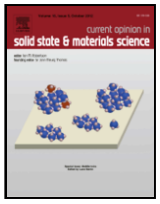
Current Opinion in Solid State & Materials Science xxx (xxxx) 101214

Contents lists available at ScienceDirect

Current Opinion in Solid State & Materials Science

journal homepage: www.elsevier.com/locate/cossms



A practical guide to machine learning interatomic potentials – Status and future

Ryan Jacobs [a, *], Dane Morgan [a, *], Siamak Attarian [a], Jun Meng [a], Chen Shen [a], Zhenghao Wu [ac], Clare Yijia Xie [c], Julia H. Yang [b, c], Nongnuch Artrith [d], Ben Blaiszik [e, f], Gerbrand Ceder [g, h], Kamal Choudhary [i], Gabor Csanyi [j], Ekin Dogus Cubuk [k], Bowen Deng [g, h], Ralf Drautz [l], Xiang Fu [m], Jonathan Godwin [n], Vasant Honavar [o, p, q, r], Olexandr Isayev [s, t], Anders Johansson [c], Stefano Martiniani [u, v, w], Shyue Ping Ong [x], Igor Poltavsky [y], KJ Schmidt [e, f], So Takamoto [z], Aidan P. Thompson [aa], Julia Westermayr [ab], Brandon M. Wood [m]

[a] *Department of Materials Science and Engineering, University of Wisconsin-Madison, Madison, WI 55705, United States*
[b] *Harvard University Center for the Environment, Harvard University, Cambridge, MA 02138, United States*
[c] *John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, United States*
[d] *Materials Chemistry and Catalysis, Debye Institute for Nanomaterials Science, Utrecht University, Utrecht 3584 CG, the Netherlands*
[e] *Globus, University of Chicago, Chicago, IL, United States*
[f] *Data Science and Learning Division, Argonne National Laboratory, Lemont, IL, United States*
[g] *Department of Materials Science and Engineering, University of California, Berkeley, CA 94720, United States*
[h] *Materials Sciences Division, Lawrence Berkeley National Laboratory, CA 94720, United States*
[i] *Material Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899, United States*
[j] *Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, United Kingdom*
[k] *Google DeepMind, Mountain View, CA, United States*
[l] *Interdisciplinary Centre for Advanced Materials Simulation (ICAMS), Ruhr-Universität Bochum 44780 Bochum, Germany*
[m] *Fundamental AI Research (FAIR) at Meta, United States*
[n] *Orbital Materials, London, England, United Kingdom*
[o] *Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, United States*
[p] *College of Information Sciences and Technology, The Pennsylvania State University, University Park, PA, United States*
[q] *Artificial Intelligence Research Laboratory, The Pennsylvania State University, University Park, PA, United States*
[r] *Center for Artificial Intelligence Foundations and Scientific Applications, The Pennsylvania State University, University Park, PA, United States*
[s] *Department of Chemistry, Mellon College of Science, Carnegie Mellon University, Pittsburgh, PA 15213, United States*
[t] *Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, United States*
[u] *Courant Institute of Mathematical Sciences, New York University, New York, NY 10003, United States*
[v] *Center for Soft Matter Research, Department of Physics, New York University, New York, NY 10003, United States*
[w] *Simons Center for Computational Physical Chemistry, Department of Chemistry, New York University, New York, NY 10003, United States*
[x] *Aiiso Yufeng Li Family Department of Chemical and Nano Engineering, University of California, San Diego, 9500 Gilman Dr. Mail Code #0448, La Jolla, CA 92093, United States*
[y] *Department of Physics and Materials Science, University of Luxembourg L-1511 Luxembourg, Luxembourg*
[z] *Preferred Networks, Inc., Otemachi Bldg., 1-6-1 Otemachi, Chiyoda-ku, Tokyo 100-0004, Japan*
[aa] *Center for Computing Research, Sandia National Laboratories, Albuquerque, New Mexico 87185*
[ab] *Wilhelm-Ostwald-Institut für Physikalische und Theoretische Chemie, Universität Leipzig, Germany*
[ac] *Department of Chemistry and Materials Science, Xi'an Jiaotong-Liverpool University, Suzhou 215123, Jiangsu, P. R. China*

A B S T R A C T

The rapid development and large body of literature on machine learning potentials (MLPs) can make it difficult to know how to proceed for researchers who are not experts but wish to use these tools. The spirit of this review is to help such researchers by serving as a practical, accessible guide to the state-of-the-art in MLPs. This review paper covers a broad range of topics related to MLPs, including (i) central aspects of how and why MLPs are enablers of many exciting advancements in molecular modeling, (ii) the main underpinnings of different types of MLPs, including their basic structure and formalism, (iii) the potentially transformative impact of universal MLPs for both organic and inorganic systems, including an overview of the most recent advances, capabilities, downsides, and potential applications of this nascent class of MLPs, (iv) a practical guide for estimating and understanding the execution speed of MLPs, including guidance for users based on hardware availabil-

* Corresponding authors.
E-mail addresses: rjacobs3@wisc.edu (R. Jacobs), ddmorgan@wisc.edu (D. Morgan).

https://doi.org/10.1016/j.cossms.2025.101214
Received 9 January 2025; Accepted 13 January 2025
1359-0286/

ity, type of MLP used, and prospective simulation size and time, (v) a manual for what MLP a user should choose for a given application by considering hardware resources, speed requirements, energy and force accuracy requirements, as well as guidance for choosing pre-trained potentials or fitting a new potential from scratch, (vi) discussion around MLP infrastructure, including sources of training data, pre-trained potentials, and hardware resources for training, (vii) summary of some key limitations of present MLPs and current approaches to mitigate such limitations, including methods of including long-range interactions, handling magnetic systems, and treatment of excited states, and finally (viii) we finish with some more speculative thoughts on what the future holds for the development and application of MLPs over the next 3–10+ years.

1. Introduction

This paper was inspired by the workshop "Machine Learning Potentials – Status and Future (MLP-SAFE)", which was held online on July 17–19, 2023. It represents select themes and key points we thought would be of particular interest to the broader materials science and chemistry communities. The rapid development and large body of literature on machine learning potentials (MLPs) (sometimes also called machine learning force fields) can make it difficult to know how to proceed for researchers who are not experts but wish to use these tools. The spirit of this paper is to help such researchers by serving as a practical, accessible guide to the state-of-the-art in MLPs. We aim to keep deep mathematics and formalism to a minimum, as such details can be readily found in other excellent reviews and references therein.[1–9] In contrast, we believe that guidance on the general landscape of MLPs- their practical use, trade-offs, pros and cons for particular problems, timing, and how to get started running them is still challenging to learn from the literature. We note that a recent Comment published by Ko and Ong and a Perspective from Duignan both highlight many of the same topics addressed in this Review,[10,11] albeit more briefly and at a higher level, and we therefore feel the present work serves as a complementary, more in-depth examination of the present state of MLPs from a practical perspective. Our target audience is technically literate material scientists and chemists, with a background in molecular modeling, but not MLP experts. Therefore, we do not discuss technical details of, for example, basis function expansions, but do provide guidance on how to understand the broad differences between approaches (e.g., atomic cluster expansion (ACE) vs. graph neural networks (GNNs)) and the benefits and tradeoffs of using different approaches. This paper will provide a high-level guide on the key fundamental aspects needed to understand the landscape of MLPs, including their enormous potential range of applications, general frameworks, typical workflows (including fitting and/or using pre-fit MLPs), speed and accuracy, supporting infrastructure, and some guidance on MLP choice.

This review paper covers a broad range of topics related to MLPs and is organized as follows. In **Sec. 2**, we provide a list of MLPs discussed throughout this review, including their abbreviations and key references to original work. In **Sec. 3**, we outline the central aspects of how and why MLPs are enablers of many exciting advancements in molecular modeling. In **Sec. 4**, we discuss the main underpinnings of different types of MLPs, including their basic structure and formalism (**Sec. 4.1**), the differences between MLPs using explicit featurization approaches of the atomic environments vs. implicit approaches leveraging graph neural networks (**Sec. 4.2**) and details of the explicit and implicit approaches more specifically in **Sec. 4.3** and **Sec 4.4**, respectively. In **Sec. 5**, we highlight the potentially transformative impact of universal MLPs (U-MLPs) for both organic and inorganic systems, including an overview of the most recent advances, capabilities, downsides, and potential applications of this nascent class of MLPs. In **Sec. 6**, we provide a practical guide for estimating and understanding the execution speed of MLPs, including guidance for users based on hardware availability, type of MLP used, and prospective simulation size and time. Next, **Sec. 7** functions as a practical manual for what MLP a user should choose for a given application by considering hardware resources (**Sec. 7.1**), speed requirements (**Sec. 7.2**), energy and force accuracy requirements (**Sec. 7.3**), as well as guidance for choosing pre-trained potentials (**Sec. 7.4**), and fitting a new potential from scratch (**Sec. 7.5** and **Sec. 7.6**). Discussion in **Sec. 8** centers around MLP infrastructure, including sources of training data, pre-trained potentials, and hardware resources for training. **Sec. 9** summarizes some key limitations of present MLPs and current approaches to mitigate such limitations, including methods of including long-range interactions, handling magnetic systems, and treatment of excited states. Finally, we conclude in **Sec. 10** with some more speculative thoughts on what the future holds for the development and application of MLPs over the next 3–10 + years.

2. A list of MLPs

In the following discussions, we will often refer to MLPs by their acronyms. To help clarify the meaning and appropriate citations for these MLPs we here summarize the names, acronyms, and standard citations of the MLPs that are discussed in this paper. Note that this is not meant to serve as a comprehensive list of existing MLPs.

Accurate NeurAl networK engINe for Molecular Energies (ANAKIN-ME, ANI for short): [12].
Allegro: [13].
Atomic Cluster Expansion (ACE): [14].
Atomic Energy Network (ænet): [15,16].
Atomistic Line Graph Neural Network-based Force Field (ALIGNN-FF): [17].
Atoms-In-Molecules Network 2 (AIMNet2): [18].
Behler-Parrinello Neural Network (BP-NN, or BP)[19].
Crystal Hamiltonian Graph Neural Network (CHGNet)[20].
Deep Molecular Dynamics (DeepMD): [21,22].
Elemental Spatial Density Neural Network Force Field (Elemental-SDNNFF): [23].
EquiformerV2-OMAT24: [24].
Fast Learning of Atomistic Rare Events (FLARE): [25].
Gaussian Approximation Potential (GAP): [26].
Graph-based Pre-trained Transformer Force Field (GPTFF): [27].
Graph Networks for Materials Exploration (GNoME): [28].
Graph Atomic Cluster Expansion (grACE): [29].
Mattersim: [30].
ACE with message passing (MACE): [31].
MACE foundation model (MACE-MP-0): [32].
MACE-OFF23 potential for organics (MACE-OFF23): [33].
Moment Tensor Potential (MTP): [34].
Neural Equivariant Interatomic Potential (NequIP): [35].
Orb: [36].
PreFerred Potential (PFP): [37].
Scalable EquiVariance-Enabled Neural NETwork (SevenNet): [38].
SchNet: [39].
Spectral Neighbor Analysis Potential (SNAP): [40].
Three-body Materials Graph Network (M3GNet): [41].
Ultra-Fast Force Fields (UF3) potential: [42].

3. What makes MLPs so exciting?

For this paper, we will define an MLP as a function that takes as input a set of atoms with positions $\{x_i, y_i, z_i\}$ and element types $\{n_i\}$ and maps this atomic configuration to a total energy E for that set of atoms i. The MLP therefore serves as a potential energy surface (PES) function. The MLP generally also provides forces, which are spatial derivatives of the PES generated by the MLP. The forces are generally available through a formal derivative expression that can be derived from the MLP and no numerical differentiation of $E\{x_i, y_i, z_i\}$ is required. A simi-

lar situation occurs for stresses. We note that some of the presently best-performing MLPs are trained separately on energies and forces, and are nonconservative in the sense that the forces are not directly calculated by differentiating the PES[24,30,36]. The purpose of an MLP is to enable efficient calculation of material properties, typically using molecular dynamics (MD), for myriad applications ranging from understanding and predicting chemical reactions to designing stronger metal alloys to developing more effective drugs. We note that here we define a "material" to mean any collection of atoms, from crystals to gasses to molecules. Throughout this work, we consider a model to be an MLP if it can provide energies and forces (regardless if these quantities are connected analytically through differentiation or obtained from separate models), and the model is capable of performing MD simulations.

Historically, atomistic simulation of materials has been divided into two very different approaches. On the one hand, *ab initio* molecular dynamics (AIMD) has enabled high accuracy simulations of small numbers of atoms, providing rich insight into the structural, thermodynamic, and transport properties of materials at the very smallest scales. On the other hand, classical molecular dynamics simulations with physics-based potentials (PBPs) have enabled researchers to qualitatively study how atomic interactions drive the emergence of diverse phenomena on much larger scales. For a long time, these two approaches were disconnected. AIMD was incapable of achieving the scale needed to observe many phenomena of scientific interest, while PBP-based MD could not provide accurate representations of specific materials. The emergence of MLPs has revolutionized the practice of atomistic simulations by bridging this disconnect. By leveraging massively parallel computing resources and flexible parallel simulations frameworks such as LAMMPS,[43] it is now possible to directly simulate large-scale emergent phenomena in specific materials with accuracy that approaches that of AIMD.

MLPs differ from traditional PBPs in that MLPs utilize a highly flexible approach to represent the PES function (e.g., a neural network), typically taken from the machine learning (ML) community. In contrast, PBPs use a highly constrained functional form guided by physics (e.g., a Lennard-Jones or Born-Meyer potential). The categories of PBPs vs. MLPs are somewhat arbitrary and inexact, as there is really a continuum of possible approaches between the extreme limits of a purely physical set of equations with almost no fitting parameters (a pure PBP) and a purely numerical fit done with almost no physical guidance (a pure MLP). An overview of the different general approaches for constructing PBPs and MLPs is provided in Fig. 1. Starting from the physics

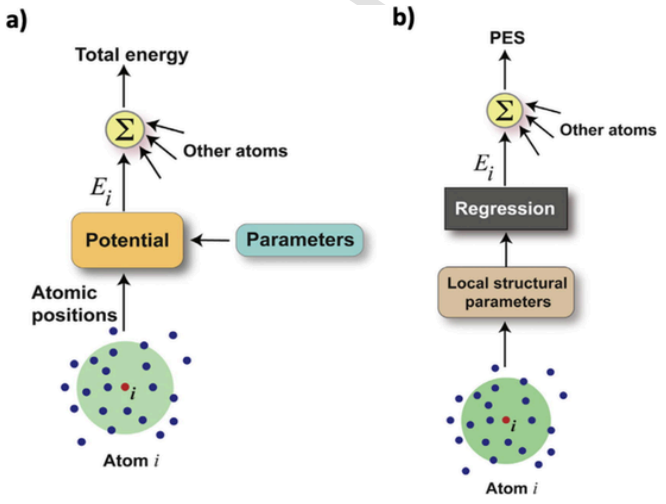


Fig. 1. Overview of approaches for generating (a) physics-based potentials and (b) machine learning-based potentials. Adapted with permission from Ref. [47].

limit, PBPs can incorporate increasingly flexible functions to become more like ML models, e.g., as has been done in the very flexible forms for pair interactions in the Embedded Atom Method (EAM) potentials. [44,45] Conversely, starting from the pure ML side, MLPs can be made more like PBPs by introducing physically-motivated terms to the PES representation, e.g., adding in a Ziegler-Biersack-Littmark repulsive interaction to ensure that atoms do not behave unphysically when close together, as is available in several MLP training packages.[21,22,46] In addition, many intermediate approaches are possible, e.g., as discussed in the review by Mishin.[4] Here, we will follow the standard convention of referring to any potential that uses traditional ML featurization or modeling approaches as an MLP.

MLPs have an advantage vs. PBPs because their flexible functional form can fit essentially arbitrarily complex atomistic scale potential energy landscapes. We note that by "energy landscape" we mean the ground state Born-Oppenheimer surface, as is generally produced by *ab initio* calculations. One of the main disadvantages of MLPs vs. PBPs is that MLPs require a lot of training data to learn the physics of the system. However, as *ab initio* data continues to become more plentiful, more accurate, easier to obtain, and codified in standard databases (e.g., the MPtrj database [20] contained in the Materials Project and the Open Materials 24 (OMAT24) database released by Meta [24]) the high training data requirements of MLPs become increasingly easy to meet, giving MLPs a notable and increasing advantage over PBPs. We can think of MLPs today as an improved version of traditional PBPs, but with greater accuracy and more flexibility to model complex systems, at the expense of higher computational cost (depending on the type of MLP used). Very flexible and accurate PBP functional forms are often difficult to develop because they require significant domain expertise and physical insight to construct.

When using MLPs, it is important to note that the level of improvement in accuracy and number of elements modeled vs. using PBPs appears to be so great that the introduction of MLPs is more revolutionary than evolutionary. The physical functional forms in PBPs, for all their ingenuity, almost always do not have sufficient complexity to quantitatively model the necessary behavior of interacting atoms across all the conditions of interest, which often contain many complex changes in bonding and charge state. In contrast, modern MLPs can capture many chemical changes of interest provided adequate training data is available. We stress that MLPs are not fundamentally limited in any particular way, e.g., to only metallic vs. ionic systems, or to only nonreactive vs. chemically reactive processes. While this is a good initial perspective for those new to MLPs, there are definitely some constraints on present MLP capabilities, and we enumerate some of the major present limitations of MLPs in **Sec. 9**. Distinctions that were often essential to determining the form and applicability of PBPs, e.g., organics vs. inorganics, bond-breaking / reactive vs. not, metallic vs. ionic vs. covalent, are often not particularly important for whether an MLP is applicable. Furthermore, the accuracy of MLPs is typically on the scale of a few to tens of meV/atom, which is often an order of magnitude better than typical PBPs.[1,48] Additionally, MLPs are straightforward to iteratively improve and can be fixed if they show undesirable errors by adding more training data.[49] While PBPs can be iteratively improved as well, doing so is more difficult than improving MLPs, because instead of just providing more diverse training data, more fundamental changes to the underlying functional forms may be needed, which requires significant expertise to do properly. Finally, MLPs with excellent testing errors are quite easy to fit (typically ranging from just days to a couple of months for a system comprising a few elements for a graduate student with the necessary skills), and good pre-fit potentials, including ones covering large parts of chemical and structural space, are becoming widely available, e.g., as seen with the recent development of Universal MLPs (U-MLPs) (see **Sec. 5**). Given all the advantages of MLPs, it seems possible that MLPs will be easy enough to train for most systems that they may at least partially replace *ab initio* calculations in applications needing

just forces and energies. Even partial replacement of *ab initio* calculations will dramatically accelerate many kinds of molecular modeling, but one notable example is that quantum mechanics-based AIMD might be almost entirely replaced by MLP MD. This replacement of AIMD with MLP MD would make similar time and length scales to those studied with AIMD accessible with orders of magnitude less compute time. Such an increased efficiency is an important change as a significant amount of the compute time used in *ab initio* simulations is devoted to running AIMD. Perhaps more importantly, the use of MLP MD would unlock gains of orders of magnitude in accessible length and time scales vs. AIMD for many systems, enabling the study of new physical regimes inaccessible with AIMD. There is growing evidence that we will be able to develop quantitative U-MLPs, something like the foundational models in computer vision and language machine learning, which can directly, or with some fine tuning, provide almost instant access to quantum mechanical accuracy on almost any chemical system at the scale of millions to billions of atoms and for microsecond or longer timescales. [20,28,32,41] Thus, MLPs may dramatically enhance the capabilities of molecular simulations, significantly impacting chemistry, biology, materials science and engineering, physics, and many other disciplines. The necessary understanding, methods, and tools exist today to enable non-experts to apply MLPs to practical problems, and it is reasonable to expect an explosion of use across many fields of science in the next few years. However, there are still significant challenges to realizing the full potential of MLPs, including refining the best features and architectures, developing optimal training strategies, finding ways to include additional physics (e.g., long-range interactions), scaling up to universal potentials, and successfully developing and adopting potentials for many complex systems of interest.

4. Understanding the types of MLPs – basic formalisms

In this section, we discuss the basic formalism behind MLPs. The goal of this discussion is to provide a qualitative description to help guide users in understanding what aspects control the key properties users care about, which include e.g., (1) human vs. computational MLP training limitations, (2) MLP speed of execution, (3) MLP accuracy, (4) MLP ease of use, and (5) appropriateness of an MLP to specific problems. Detailed mathematical descriptions of MLP formalisms can be found in many other reviews.[1–9] This section provides a high-level overview of the basic construction of an MLP (**Sec. 4.1**), discussion of the construction and use cases of MLPs created by explicitly featurizing atomic positions with specific functional forms (**Sec. 4.3**), discussion of the construction of MLPs created implicitly through featurizing by graph neural network approaches (**Sec. 4.4**), the general differences between these two approaches (**Sec. 4.2**), and, finally, the unification of these two approaches into a single over-arching MLP framework (**Sec. 4.5**). We stress that this section was written to reflect the historical development of different MLP formalisms, where we discuss differences between various approaches which we believe accurately portrays how the community has thought of MLP development until recently. However, these previously perceived differences between various MLP formalisms appears to be collapsing into a single over-arching formalism, which we discuss in more detail in the following subsections.

4.1. The basic structure of an MLP

Almost all MLPs have the same qualitative structure, although the details of the implementation differ between MLP types. The idea behind this structure is that for use in an MLP, the local environment of all atoms must be represented by some set of numbers, or features, which we will call its atomic environment featurization (AEF). In Fig. 1**B**, this is described as "local structural parameters". The AEF is built in a manner such that it can be represented as a manageable set of numbers, then that featurization is fed into an ML model. The potential accuracy of the model depends on how well these features and the model can capture the local environments, and, generally, larger sets of features are better able to capture environments (this is sometimes referred to as an AEF that is more "expressive").

4.2. Explicit vs. implicit MLPs

Determining how to distinctly categorize different MLP approaches is challenging. This complication is the result of the multiple different ways researchers approach the featurization portion of MLP development, and, as discussed below, how greater understanding in the field has prompted the convergence of various approaches, making the boundary between MLP approaches more nebulous. However, we think that a helpful distinction at present is to consider MLPs as being based on "explicit AEF" vs. "implicit AEF". We note that the designation of explicit vs. implicit AEF is analogous to what others, such as Schütt et al., have previously called "handcrafted" vs. "learned" representations.[39] By explicit AEF MLPs, we mean MLPs that define an explicit set of features for each element. Explicit AEFs are the type of potentials that were first invented by Behler and Parrinello[19] and have dominated MLPs until quite recently, where the specific formulations of these explicit AEF MLPs are discussed below in **Sec. 4.3**. In contrast, implicit AEF MLPs are MLPs that define a set of features or chemical descriptors which are learned, rather than pre-defined. Implicit AEFs result in learned features (sometimes called "embeddings") of the atoms and bonds comprising a material. MLPs employing implicit AEFs will generally involve more ML architectural complexity, potentially making them harder or slower to use, train, and execute. Of particular importance is that the learned features from implicit AEFs can scale with number of different chemical species much more efficiently than those used in most explicit AEF MLPs, and it is therefore this category of implicit AEFs that is almost always used for modeling many elements (e.g., > 5). As of this writing, implicit AEF MLPs are almost entirely based on deep learning approaches for learning effective features. For example, implicit AEF MLPs include all of the graph neural network (GNN) approaches (e.g., M3Gnet,[41] NequIP,[35] etc.) and the newest implementations of DeepMD.[21,22] Therefore, we will usually just refer to implicit AEF methods as deep learning methods, although these two categories are technically distinct.[39]In the text below, we will refer to explicit and implicit or deep learning-type MLPs when the above distinction is useful.

4.3. Explicit AEF type MLPs

In this section, we describe the explicit construction of the AEF. The standard way to treat the mathematical representation of atom types and positions is to consider each atom as having an energy given by the atom type and its local environment (the positions and element types of nearby atoms). For this description, we refer to a given atom under consideration as the *target atom* (atom *i* in Fig. 1). The initial AEF for a target atom is generally constructed by writing the local atomic environment as a set of densities for a given atom type and then expanding that density function using a basis set consisting of radial and angular functions (for example, Bessel and spherical harmonic functions, respectively). The explicit AEF is most effective when it respects the symmetries of materials, which typically include permutations, translation, and rotation. A symmetry-aware representation can be created by taking tensor products of the initial AEF over the target atom and its near neighbors. These tensor products can then be combined to create a set of values that are covariant (i.e., change in a structured and predictable way) with symmetry operations. The final ML model then operates on these tensors, generally to predict a single scalar energy. It is possible and quite common to just keep scalar-covariant, generally called *invariant*, features, which can then be used in almost any ML model, provided the ML model is continuously differentiable.

Once an AEF is established, any atom and its environment can be mapped onto a vector, which can then be used as an input feature in a standard ML model. We call the simple passing of the AEF as features into the ML model the "explicit AEF approach". A graphical overview of the explicit AEF approach is given in Fig. 2. The training target values for the ML model are typically a set of energies and forces. These energies and forces could come from any source but are almost always taken from a large set of *ab initio* calculations, such as density functional theory (DFT). Any DFT cell calculation that provides energies and forces, including stable structures, structures calculated during atomic relaxations, and structures calculated during AIMD are potentially of use. Then, the model parameters are estimated by standard regression methods. One difference from typical regression problems is that the training data are not just simple functions of the AEFs. First, the forces on a single target atom are often given as the derivative of the ML function, and so in this case the fitting loss function must have a term that depends on the derivative of the ML model function. It is worth noting that some MLPs are trained on energies and forces separately, e.g., NN-based MLPs trained without forces, where comparable accuracy can be obtained by increasing the training dataset size.[51,52] Second, the training energies are each total energies for a whole set of atoms comprising a molecule or crystal unit cell (almost no *ab initio* methods allow easy formal decomposition of the total energy into values for each atom), and so the fitting loss function typically must have a term that depends on the sum of energies of all the atoms in each calculated configuration. Note that some MLP code packages will fit to other properties as well, e.g., stress tensor, virial, polarizability, etc.[21,34] These additional properties can all be included in the fitting with regression approaches like those just described but with adjustment to the loss function. Assuming one is using an established MLP code repository or package, these manipulations should be automatic and thus largely invisible to the user, and one can consider the MLP fitting qualitatively as fitting a simple regression problem. As with any regression, there are many possible ML models available. The most widely used models for MLPs, listed in approximate order of their conceptual simplicity, decreasing speed of fitting and execution, and increasing accuracy are: Linear Regression (LR) (simplest, fastest, least accurate), Gaussian Process Regression (GPR), and Neural Networks (NNs) (complex, slowest, most accurate). We note that the speed of fitting GPR is highly dependent on the dataset size, and, for small datasets, GPR can have accuracies exceeding those of NN approaches, which tend to excel for problems involving large datasets.[1] Here, we exclude GNNs as they are discussed separately in **Sec. 4.4** in the context of implicit AEFs. From the above discussion, it is worth noting that some highly popular ML models used in regression problems, such as random forests, are not suitable for MLPs because they do not possess continuous derivatives.

There is no universal answer to which ML model is best for MLPs, but with good featurization, LR and GPR have both proven to work very well and are generally simpler to fit than NNs. Many of the most widely used MLPs can be described with this explicit AEF framework. Specifically, the original Behler-Parrinello potential used atom-centered symmetry functions (ACSFs) as AEFs and a NN ML model,[19] the Gaussian Approximation Potential (GAP) uses the Smooth Overlap of Atomic Positions (SOAP) approach to construct AEFs and a GPR ML model,[26] the Spectral Neighbor Analysis Potential (SNAP) used hyperspherical bispectrum functions (HBFs) as AEFs and a LR ML model,[40] the Moment Tensor Potentials (MTP) used moment tensor functions (MTFs) as AEFs and a LR ML model,[34] and the Atomic Cluster Expansion (ACE) uses the product of radial functions and spherical harmonics as its AEF and a LR ML model.[14] It should be noted that it has recently been realized that the ACE formalism is a superset of most other methods, meaning that ACSFs, SOAP, HBFs, and MTFs are all specific cases of ACE.[14] Note that this does not make these other potentials irrelevant, since any given potential may represent specific choices that are exceptionally efficient to train or execute, but it does help to realize that ACE appears to be a comprehensive formalism for expressing state-of-the-art explicit AEFs for MLPs. Moreover, it is possible to combine any of these AEFs with any ML model. For example, the FitSNAP software[46] allows SNAP and ACE featurizations to be combined with PyTorch and JAX models.[53].

While the explicit AEF formalism is very effective, it has until recently had a significant scaling problem[13] which we describe here.

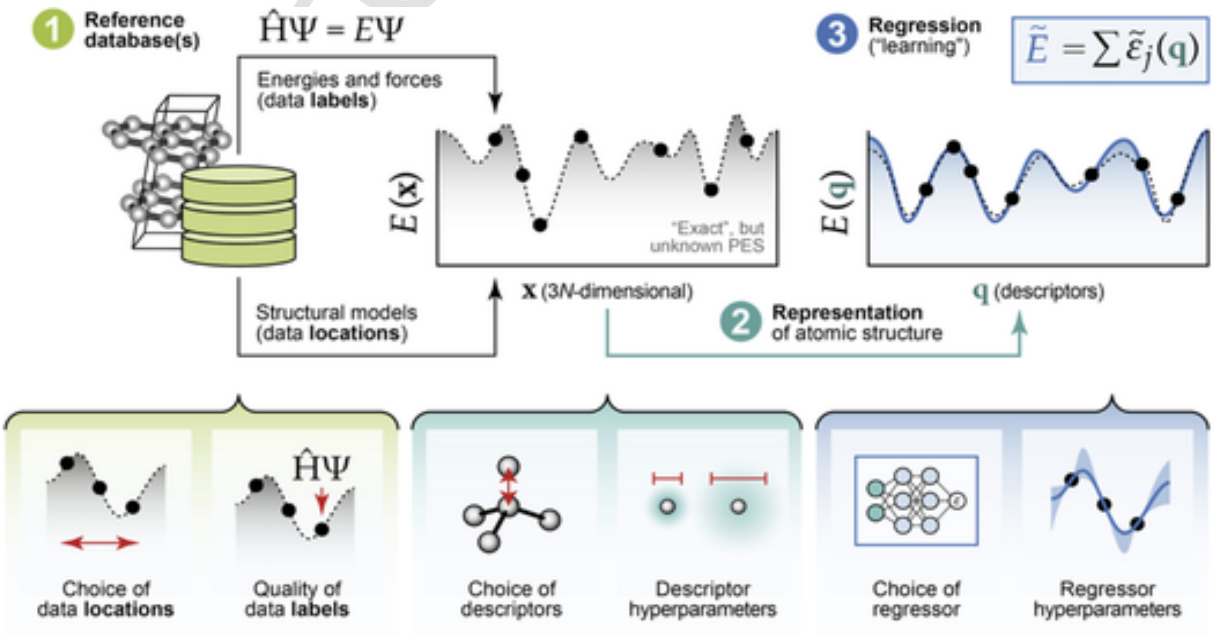


Fig. 2. An overview of the explicit AEF approach of making an MLP, including acquiring reference data from *ab initio* calculations, choosing a featurization approach to represent the local chemical environments, and an ML regression model to map the chemical environments to energies and forces. Adapted with permission from Ref. [50].

We note this argument is based on the ACE basis construction, but it is quite general and similar issues occur in other related explicit AEF formalisms. Let N_b be the number of basis functions used to expand the density of one species around a target atom and S be the number of species. Then there are $N_{total\_basis} = N_b \times S$ total basis functions for one target atom. Let ν be the number of atomic sites we couple to in tensor products (here $\nu + 1$ is called the bond order, and $\nu + 1 = 2$ gives pair information, $\nu + 1 = 3$ gives 3-body information, and so on). For a given bond order, there are order $O((N_b \times S)^{\nu})$ basis functions. For a typical bond order of 3, this gives quadratic scaling with the number of basis functions and species, which can become quite slow for a complex basis and for large numbers of species. The species scaling typically limits most explicit AEF potentials to approximately 5 or fewer species.

4.4. Implicit AEF and GNN MLPs

There are a few solutions to the issue of poor of scaling for explicit AEFs, particularly with the number of species. The general approach to overcoming this scaling problem is to instead use an implicit AEF to embed the chemical space in a learned feature vector (i.e., an embedding) that can effectively represent different chemistries without explicitly developing basis functions for each one. This approach appears to work very well, dramatically reducing the complexity of treating different species. The exact reason this works is not totally clear, but likely is because the properties of different elements are not independent, and their interactions in subclusters inform their more complex cluster interactions (e.g., pair couplings can dominate the energy of a cluster of 10 different atom types). Probably the most widely used approach that provides efficient embedding (as well as has other potential advantages and disadvantages) are GNNs, discussed more below. However, there are other approaches. For example, the DeepMD[21,22] MLP represents the local environment as embedding vectors that are constructed by a neural network based on some or all of local atom distances, angles, and types. The weights of the embedding network are trained during fitting, making the AEF an implicit function of the coordinates that is learned during training and allowing DeepMD to fit many elements. A number of papers have recently shown how features in standard explicit AEF MLPs might be manipulated to reduce the scaling with species, where such an approach by Lopanitsyna, et al. is illustratively named "chemical compression".[54–56] Darby et al. in particular has shown that linear embedding of the elements into a fixed dimensional vector space corresponds formally to tensor-decomposition, and as the dimension increases will converge to the uncompressed result.[54] Artrith et al. showed that element-specific weights allow constant-size AEF vectors irrespective of the number of chemical elements and demonstrated the method for up to 11 species[16] and a similar approach was independently proposed by Gastegger et al.[57] An outstanding example of the power of these approaches is the graph ACE (grACE) method and package, which are now available and appear to provide excellent scaling with the number of elements while achieving high accuracy.[29] These recent papers and emerging packages suggest that soon the chemical scaling issues associated with the explicit AEF approach may be greatly reduced or removed altogether.

A GNN is an NN architecture that operates on graphs, where graphs are collections of nodes and the connections between them (called edges). Perhaps not surprisingly, a graph is an excellent way to think about interacting atoms, where nodes are mapped to atoms and edges are mapped to reasonably near neighbor bonds. GNNs and the graph representation provide a somewhat different approach to constructing an AEF with some clear advantages vs. the explicit AEF approaches discussed above, and therefore have become a very popular approach for MLPs. In a graph, sets of embeddings are associated with each node and/or edge, and these embeddings can be mapped to properties of the atoms by the GNN. GNNs iteratively update the embeddings of a target node/bond through learned mappings of connected node/bond embeddings onto the target node/bond, with the connections determined by the graph structure. Each one of the updates is typically done in one layer of the GNN. These updates are also given structural information like bond lengths or more detailed AEF parametrizations. Because GNNs encode the features of atoms and bonds through a learned mapping to embedded features, these features can potentially represent the chemistry and structure much more effectively than the basis function tensor products in the standard explicit AEF MLP described in **Sec. 4.3**. In particular, compared to explicit AEFs, these embeddings appear to avoid the explosion in complexity and resultant scaling problems with number of species noted in **Sec. 4.3**. Thus, GNNs using implicit AEFs appear to have the ability to scale to almost arbitrary numbers of chemical components. Not all GNNs are equivalent. For example, recent GNNs are often so-called E(3) equivariant NNs (e.g., NequIP[3541], MACE[32], TeaNet[58]), which work with highly expressive equivariant tensor representations of atomic environments and operate on them to preserve the proper symmetries. Such GNNs appear to be particularly data-efficient in fitting. Also, most GNN MLPs effectively couple a widening range of atoms/bonds to a target atom/bond at each layer of the GNN. The multiple layers needed to get good convergence often effectively couple atoms 3–4 nm apart. This coupling can be advantageous for capturing longer range interactions, e.g., as shown for M3GNet in comparison with MTP potentials.[41] However, this coupling of 3–4 nm is much longer than typical ranges of direct physical interaction in almost all PBPs and explicit AEF MLPs (which are almost always 1 nm or less) and can lead to significant memory and parallelization issues. Therefore, researchers are now exploring more local equivariant NN approaches, e.g., Allegro,[13] which has excellent scalability with multiple processors.

4.5. Unifying explicit and implicit AEFs

It is worth noting that all of these MLP methods are increasingly appearing to be different aspects of a single general MLP approach. As discussed above, explicit and implicit AEFs were developed largely independently. Explicit AEFs focus on local descriptions of atomic energy obtained by the interaction with all neighbors within a cutoff distance. Implicit AEFs recursively incorporate via message passing information about atoms that can be several cutoff distances away. The messages are assembled from the local atomic environment within a cutoff distance and then employed for the computation of the energy of another atom. From the viewpoint of explicit AEFs, message passing modifies the character of an atom. In an explicit AEF, neighboring atoms are characterized by their positions and chemical species. In an implicit AEF, neighboring atoms are characterized by further attributes collected from the atomic environment. For example, this makes a carbon atom on a surface different from a carbon atom in the bulk. In equivariant neural networks the additional attributes are vectors and tensors, which essentially give the carbon atom an environmentally dependent, non-spherical character.

ACE provides a complete basis for the local atomic environment. Applied to the local atomic environment of neighboring atoms, ACE facilitates formally complete messages.[59] Recursive application of ACE in an implicit AEF is multi ACE (MACE).[31].

However, while intuitive, it is not necessary to take an iterative evaluation as the starting point. In fact, the ACE basis was extended to incorporate more general graph basis functions.[29] In this setting, the ACE basis functions build on star graphs, whereas the more general graph basis functions on tree graphs. In complete analogy to ACE, in graph ACE the energy or any other local or semilocal property is written as a linear combination of graph ACE basis functions, i.e. in an explicit AEF. Only for an efficient numerical evaluation of graph ACE functions and by employing tensor decomposition along the graph ACE basis functions, an iterative evaluation is employed. This iterative evaluation

comprises message passing equivariant neural networks such as NequIP,[35] MACE, i.e. it corresponds to implicit AEFs.

This facilitates the following understanding. AEFs can be formulated in an explicit way. Here, graph ACE provides a general and complete representation. Explicit AEFs with graphs that have two or more layers can be transformed to implicit AEFs for numerically efficient evaluation, resulting in message passing neural networks. In practice, even single layer explicit AEFs are evaluated iteratively for numerical efficiency,[60] which means that they should fall into the implicit AEF category, too.

Therefore, the distinction between explicit and implicit AEFs reflects the history of the development of AEFs in the past years more than their actual structure. To the best of our knowledge, all AEFs can be represented in an explicit way. Iterative evaluation for numerical efficiency leads to implicit representations of AEFs. These results increasingly suggest that we may be converging on a single general formalism for MLPs, and the seemingly very different approaches in use today are actually specific choices within the general formalism. Such understanding will hopefully allow the community to extract the approaches that are simultaneously optimized to be the most efficient for training, and fast and most accurate for prediction.

5. Universal MLPs

To date, the vast majority of MLPs are trained on a limited domain of chemical or materials systems. This amounts to an MLP that represents a particular materials family (e.g., perovskite oxides, 2D MXenes, etc.) or particular chemical system (e.g., the Li-Co-Mn-Ni-O composition space) well, but is not transferable in the sense that these MLPs are unable to extrapolate to accurately model new elements or structure types that are not present in the specific training data. As discussed in **Sec. 4** above, part of the reason researchers focus on small numbers of chemical species is related to the explicit AEF approach used, many of which do not scale well to include more than ~ 5 species. The creation of accurate, highly general MLPs that cover many more elements and conditions than typical MLPs is highly desirable as it would produce a potential with the widest possible domain of applicability, enabling the study of the statics and dynamics of many types of chemically complex systems, potentially for long simulation time scales. In thinking about scaling up MLPs to more chemical species, we propose that it is useful to distinguish a few categories of MLPs, specifically:

1. Targeted MLPs (T-MLPs). These are the typical MLPs that cover approximately 1–10 (usually < 5) elements and are typically under some constraints of chemistry, structure, or phase (e.g., oxides with certain compositions, multiple elements in a fixed crystal structure for high entropy alloys, or molten (liquid) phase salts) although these latter constraints can be quite few or potentially even none.
2. Universal MLPs (U-MLPs). These attempt to cover a large number of species under different levels of constraints, e.g., transition metal oxides in solid form or organic molecules with select heavy elements. These typically cover 10–100 elements and could range in conditions, from a very strong constraint such as a specific crystal lattice to allowing almost any atomic configuration. Obviously, the MLP would be considered more universal as more elements are included and fewer constraints on the considered chemical or material structures are made. It can be useful to consider these MLPs in two categories, which we call semi-universal-MLPs (SU-MLPs) and true U-MLPs. Both require a method that can scale well with number of species and target a large number of species. However, SU-MLPs focus on a select domain, e.g., transition metal oxides in solid form or organic molecules with select heavy elements. A good example of an SU-MLP is the recent AIMNet2,[18] which targets molecular and macromolecular structures and is applicable to species containing up to 14 chemical elements in both neutral and charged states, making it valuable for modeling the majority of non-metallic compounds. As another example of a SU-MLP, the work of Rodriguez et al. built the Elemental Spatial Density Neural Network Force Field (Elemental-SDNNFF), which produces accurate forces for Heusler alloys constituting 55 different elements and accurate predictions of phonon properties.[23] A third example is the SuperSalt potential from Chen et al.,[183] which models M−Cl molten salts for 11 cations M, and was shown to be significantly more accurate than the MACE-MP0 U-MLP for these materials. In contrast, U-MLPs attempt to cover a very large fraction or even almost all of the periodic table with atoms potentially in any arrangement. Even for U-MLPs, it is typical to exclude elements that are very impractical or intractable to study, e.g., Nobelium (atomic number 102 or anything with an atomic number above 103). Thus, the relevant portion of the periodic table for materials and chemistry is generally up to about 100 elements. U-MLPs typically cover over 50 elements and may accurately model solids, liquids, and molecular structures. A good example of this class is the recent M3GNet potential from Chen and Ong,[41] with 89 elements and no particular constraints on its applicability (although there is a strong bias in training to solid phases), or the above-mentioned MACE-MP0, which is trained on the same data and was shown to be effective for running stable MD simulations.[32]

The exact values of the number of elements or level of structural constraint in the categories above are somewhat arbitrary, although T-MLPs are distinct from U-MLPs in that the latter typically require scalable implicit AEF methods (see **Sec. 4**). In particular, in this section U-MLPs will be used rather loosely to indicate an MLP which has been trained on sufficiently large and diverse datasets such that it provides usefully accurate predictions on a wide range of compositions and structures for molecules and/or materials. If the training data is sufficiently large and diverse, the MLP may provide accurate predictions for the behavior of most chemically relevant elements in the periodic table.

Universal potentials are not limited to MLPs and have been developed previously in the context of PBPs. The creation of universal PBPs dates back to 1981 with the seminal work of Weiner et al.[61] Since this time, the universal force field (UFF) of Rappe et al.[62] and the Assisted Model Building with Energy Refinement (AMBER) force fields [61,63] have emerged as some of the most popular universal PBPs, where the main utility of these potentials is for modeling molecular systems (e.g., to aid drug discovery), as opposed to condensed phases. The relative utility of these universal traditional PBPs vs. U-MLPs is difficult to determine at this stage since the development of U-MLPs is still in the nascent stages. The first reported U-MLP for organic systems (representing molecules initially containing only C, H, O, N atoms) is the Accurate NeurAl networK engINe for Molecular Energies (ANAKIN-ME, ANI for short) potential from the work of Smith et al. in 2017,[12,64] which was expanded in 2020 to include S, Cl and F elements (thus covering ~ 90 % of drug-like molecules).[65] The ANI potential has similar applicability as the universal AMBER PBP for organic systems, but in MLP form. The latest iteration of this U-MLP as of this writing came in late 2023, termed the atoms-in-molecules neural network potential (AIMNet2) U-MLP.[18] This U-MLP extends the ANI potential to include up to 14 elemental species and additional energy terms related to short-range van der Waals (vdW) correction and long-range electrostatic correction, enabling higher fidelity predictions of organic molecules and macromolecules which can also include the effects of charged species and species with different valence states. In addition, the ANI-1xnr potential extended the success of the ANI U-MLP to also enable the accurate modeling of condensed phases of organic systems (comprising C, N, H, O) such as liquids, supercritical fluids, and chemical reactions.

[66] Finally, the MACE-OFF3 potential,[33] also published in late 2023, uses the MACE message-passing framework to construct a U-MLP for the 10 most-occurring elements in organic chemistry (H, C, N, O, F, P, S, Cl, Br, I). Compared to the most recent ANI potentials, MACE-OFF3 uses only short-range interactions, yet results in improved performance on a number of benchmark molecular simulation properties compared to ANI.

The first published U-MLPs intended to have broad applicability across most elements in the periodic table came nearly simultaneously in early 2022.[41,58] In just the past two years, many U-MLPs capable of representing most elements in the periodic table have been developed: (1) the 3-body Materials Graph NETwork (M3GNet) potential from Chen and Ong;[41] (2) the Crystal Hamiltonian Graph Neural Network (CHGNet) of Deng et al.;[20] (3) the unified atomistic line graph neural network-based force field (ALIGNN-FF) of Choudhary et al.;[17] (4) the tensorial message passing neural network PreFerred Potential (PFP) from the work of Takamoto et al.,[37] which is now shared as a commercial product in the Matlantis package;[67] (5) the Graph Networks for Materials Exploration (GNoME) U-MLP from Merchant et al.,[28] which is a custom-trained version of NequIP from the work of Batzner et al.[35] fit to an in-house database of roughly 80 million DFT calculations;[28] (6) the SevenNet-0 potential from Park et al.,[38] which is also based on NequIP and trained in the same Materials Project data as M3GNet but refined to provide good scaling on many processors for modeling larger systems; (7) the equivariant graph tensor network MACE-MP0, developed by Batatia et al.,[32] which was trained on the same publicly available data used by the CHGNet model, and was demonstrated to have a high degree of accuracy on three illustrative applications of dynamics of aqueous systems, heterogeneous catalysis, and metal–organic frameworks but also showed stable nanosecond-long molecular dynamics on over 30 examples with diverse chemistry; (8) the graph-based pre-trained transformer force field (GPTFF) developed by Xie et al.,[27] a GNN model with transformer blocks integrated into the model architecture; (9) MatterSim,[30] a large-scale deep learning model from researchers at Microsoft trained on actively-learned DFT data from a large custom database of roughly 17 million atomic configurations, including many non-ground state structures over a large temperature (0–5000 K) and pressure (0–1000 GPa) range; (10) the Orb model developed by Neumann et al.,[36] which achieved excellent performance on the MatBench leaderboard and offers the advantage of faster performance compared to other leading U-MLPs, where, for example, it was found Orb performed 3–6 times faster than MACE, particularly for large system sizes and if dispersion corrections were included; and, finally, (11) the EquiformerV2-OMAT24 model from Meta,[24] which trained the EquiformerV2 model[68] on a novel open source database of roughly 118 million atomic configurations, leading to, as of this writing, the best performance on the MatBench leaderboard. An overview of some example capabilities of U-MLPs is given in Fig. 3.

The above U-MLPs are made possible by advancements in previously developed GNN models to include physical information of how the bond energies of a system evolve with the positions of the constituent atoms, enabling the acquisition of forces and stresses via differentiation of this learned energy dependence. For example, the M3GNet potential is an extension of the MatErials Graph Network (MEGNet) model[69] to include 3-body interactions (note, general N-body interactions are possible, but 3-body is used for computational efficiency), explicit atomic coordinates, and the 3×3 crystal lattice matrix.[41] As another example, ALIGNN-FF extends the ALIGNN model,[70] which already incorporates many-body interactions, to also produce atomwise and gradient predictions, thus enabling calculation of the force on each atom and stress on the system.[17] In addition to advancements to underlying GNN models, U-MLPs have been made possible by the growth of large computational databases, namely those containing tens of thousands of static DFT calculations and AIMD simulations. Each cataloged DFT structure provides one energy and 3 N forces (N = number of atoms in the structure) to use for training the universal MLP. The presently available U-MLPs were all trained on various databases of DFT calculations, as summarized in Table 1. In addition, Fig. 4 shows the evolution of DFT database size used to train various U-MLPs over time. We find, on average, that the database size has increased by more than an order of magnitude each year, from roughly 2×10^5 in 2022 (M3GNet) to a present maximum of 1.18×10^8 in 2024 (EquiformerV2-OMAT24). Even one more year of following this trend would bring the community to the level of training on billion calculation databases, a demanding goal but one that would likely bring further improvements in performance.

While it is possible to train a U-MLP on only energies, Chen and Ong recommend training on energies, forces, and stresses to obtain the most physically accurate potential, and the inclusion of stresses is needed if one is interested in modeling structural phase transformations or performing molecular dynamics simulations where volume can vary (e.g., NPT ensemble).[41] These U-MLPs tend to have very good accuracy when averaged over large test data sets, evidenced by test errors in CHGNet (M3GNet) which have energy, force and stress mean absolute errors on test data of 29 (35) meV/atom, 70 (72) meV/Angstrom, and 0.308 (0.41) GPa, respectively. ALIGNN-FF, trained only on energy and forces as seen in Table 1, has energy and force mean absolute errors on test data of 86 meV/atom and 47 meV/Angstrom, respectively. Note the higher errors for ALIGNN-FF are likely due to the authors using roughly 300 k of the 4 million data points available to them for training due to hardware and speed constraints, and not a fundamental limitation of the ALIGNN-FF approach. These values are comparable to other MLPs that cover much smaller domains of chemical space.

In general, the developers of these U-MLPs (M3GNet, CHGNet, ALIGNN-FF, PFP, GNoME, MACE-MP0) all perform multiple benchmark tests on various classes of materials structures, chemistries, and prediction of resulting materials properties. While the specific tests and comparisons are too numerous to list here and also not directly comparable due to different databases used for training and testing, all of these U-MLPs are successful in accurately modeling a very large domain of materials phenomena, with typical energy, force and stress errors greatly surpassing many-body PBPs such as EAM and modified EAM and achieving comparable or slightly worse accuracy than explicit AEF approaches relying on local environment representations like MTP. Therefore, it appears possible these U-MLPs may soon be able to achieve near DFT accuracy across many different arrangements of atoms.

The CHGNet U-MLP is unique from the other U-MLPs discussed here because it additionally includes the electronic effects of valences by explicitly embedding the magnetic moments on the vector representation of each atom, thereby enabling charge-informed atomistic simulations.[20] The inclusion of such electronic effects in an MLP might be beneficial to modeling some materials phenomena that are highly correlated with charge states (i.e., transition metal bonding dictated by the ions' valence states, and phase transformations driven by charge disproportionation, discussed more below). There are different approaches to represent charge on an atom, and in CHGNet, the charge is inferred via the DFT-calculated magnetic moment, which is essentially the localized spin density that is governed by the electron orbital occupancies of a given valence. Therefore, the training data and predicted outputs of CHGNet consist of energies, forces, stresses, and magnetic moments on every atom in the system, where the addition of magnetic moments in training led to further error reductions of energy, force and stress (in the range of 1–10 %, depending on the property) compared to not including magnetic moments in training. More important than slight error reductions is the new ability to model key pieces of physics governed by specific valence states and charge transfer, which was not possible with any previously formulated MLP. To illustrate the power of this capability, Deng et al.[20] highlight the ability of CHGNet to (i) ac-

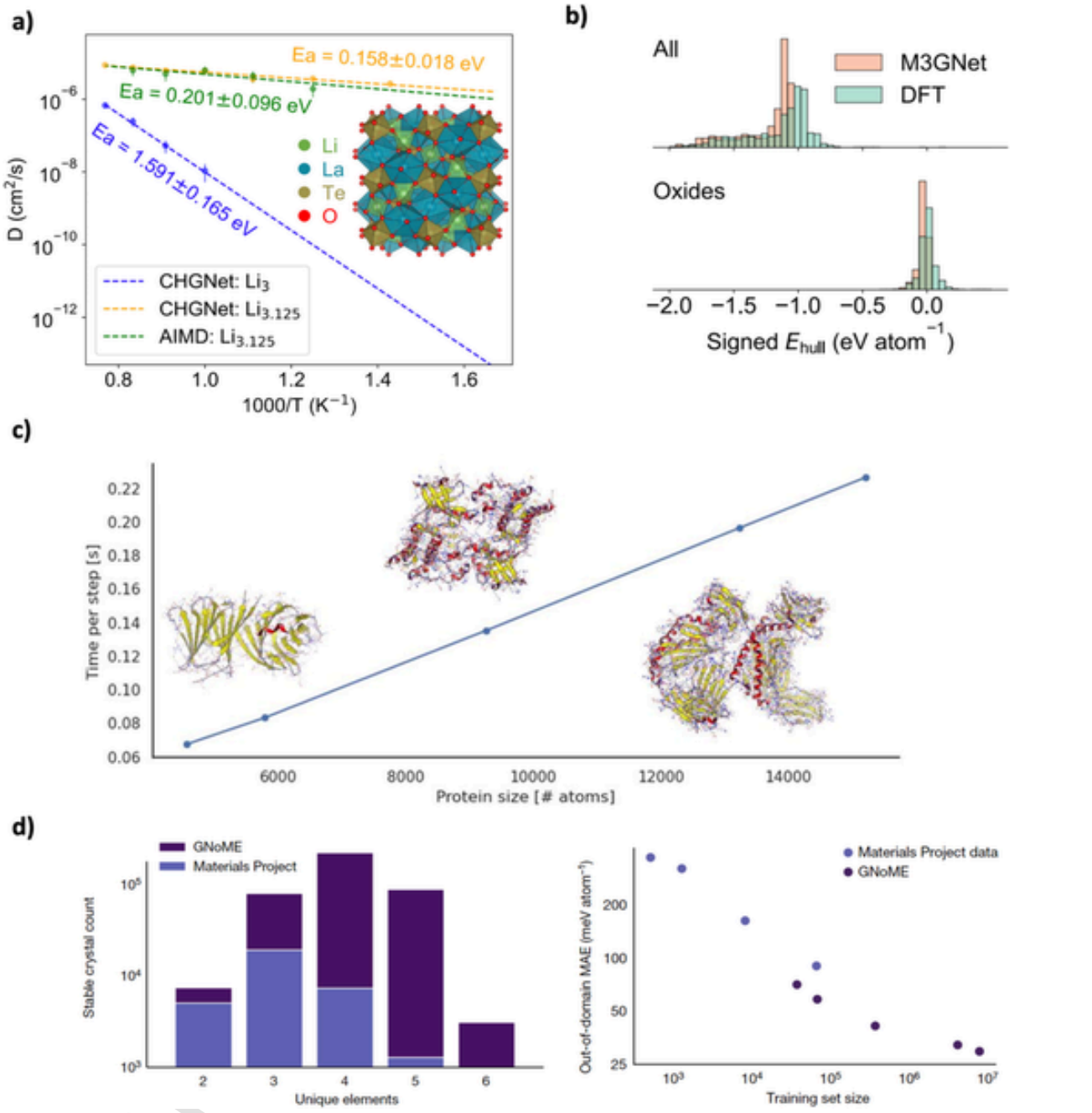


Fig. 3. Examples of applications of U-MLPs. (a) Li diffusivity in the solid electrolyte material $Li_3La_3Te_2O_{12}$ using the CHGNet potential, adapted with permission from Ref. [20]. (b) Calculated thermodynamic stabilities (as signed convex hull distance) of a set of hypothetical predicted materials using the M3GNet potential and compared to DFT calculations, adapted with permission from Ref. [41]. (c) Time to optimize the structure of large protein structures using AIMNet2, adapted with permission from Ref. [18]. (d- left) Number of newly discovered stable materials using a GNN model that only predicts the formation energy of a given crystal based on number of unique elements in the structure, (d- right) mean absolute error as a function of training set size using the same GNN, Ref. adapted from [28].

curately discriminate different valence states of transition metal with the example of V oxidation in $Na_4V_2(PO_4)_3$, (ii) enable the study of charge transfer-based dynamic information with the example of charge-coupled degradation in $LiMnO_2$ battery cathode material, where the degradation is driven by the dynamic differences of Mn^{2+} and Mn^{3+} vs. the immobile Mn^{4+} cations, and (iii) model how the electronic entropy effects in the battery cathode material Li_xFePO_4 drives the finite temperature phase stability of this material, where the inclusion of Fe valences in CHGNet correctly reproduces the qualitative miscibility gap as Li is added to Li_xFePO_4, whereas no miscibility gap is observed if the Fe valence effects are ignored. Finally, it is worth noting that while the original CHGNet model took 8.3 days to train on a single A100 GPU, the recently developed FastCHGNet includes several optimizations which results in significantly faster training, down to just 1.5 h when using 32 GPUs.[71].

There have been at least five notable, recent studies benchmarking the performance of different U-MLPs. First, work by Yu et al.[72] compared the ability of M3GNet (and the newer Pytorch-based MAT-GL im-

Table 1
Summary of data and applicability domain of U-MLPs.

U-MLP name	Training database	Number of elements represented	Training data amount	Notes
M3GNet	Materials Project	89	62,783 compounds: 187,687 energies, 16,875,138 forces, and 1,689,183 stresses	Training data taken from Materials Project dating back to its inception in 2011
CHGNet	Materials Project + Trajectory database	89	146,000 compounds: 1,580,395 energies, 49,295,660 forces, and 14,223,555 stresses	Training data taken from Materials Project GGA and GGA + U relaxation trajectory up to Sept 2022 version.
ALIGNN-FF	JARVIS-DFT	89	307,113 energies and 3,197,795 forces for 72,708 compounds	
PFP (Matlantis)	Custom	96 (previous versions were for 18 (TeaNet) and then 45 elements)	Roughly 60 million configurations	Training data is a custom in-house set performed by a collaboration of Preferred Networks, Inc. and the ENEOS Corporation
GNoME	Materials Project + Custom	94	Roughly 89 million configurations from 6 million compositions	Initial training done on Materials Project data from 2018 comprising 69,000 materials. Later fits include about 89 million configurations
MACE-MP0	Materials Project + Trajectory database	89	~150 k compounds comprising ~ 1.5 million atomic configurations	An additional dispersion correction model can be used to accurately capture dispersion physics not present in the training data
SevenNet-0	Materials Project	89	Same training data as used to build the M3GNet potential	Same training data as used to build the M3GNet potential
GPTFF	Atomly.net	Value not given in text	Roughly 2.2 million crystal structures, consisting of a total of 37.8 million energies (349 k of these are equilibrium states), 11.7 billion force vectors, and 340.2 million stresses	
MatterSim	Initial data from public databases like Materials Project, Materials Project Trajectory, and Alexandria, then customized with additional DFT calculations	89	Roughly 17 million atomic configurations	Sampling techniques include simulations with temperatures ranging from 0-5000 K and pressures from 0 to 1000 GPa
Orb	Materials Project Trajectory and Alexandria	89	Value not directly mentioned in text	Orb found to be 2–6 times faster than closest competitors (depends on system size)
EquiformerV2-OMAT24	Initial data from public databases like Materials Project, Materials Project Trajectory, and Alexandria, then customized with additional DFT calculations	89	Roughly 118 million atomic configurations	As of this writing, state-of-the-art performance on MatBench leaderboard and largest publicly-available DFT database

plementation), CHGNet, MACE-MP0, and ALIGNN-FF to predict various materials properties. Regarding the convergence behavior of cell relaxations, they found CHGNet and MACE-MP0 to be best, with M3GNet having numerous cases of providing non-converged full-volume relaxations. All models could predict formation energies roughly as well, though CHGNet had the lowest MAE at just 81 meV/atom, while all other models had MAEs that were 129 meV/atom or higher. For vibrational properties, MACE-MP0 emerged as the best, while ALIGNN-FF demonstrated some significant qualitative errors with reproducing phonon band structures. Second, work by Focassio et al.[73] compares predictions of M3GNet, CHGNet, and MACE-MP0 for predictions of bulk and surface total energies and surface energies for 73 elemental systems for which bulk and surface slab data were available in the Materials Project, where a total of 1497 surface structures were considered. As shown in Fig. 5, all three U-MLP models were able to accurately reproduce the total energies of bulk (note, on average CHGNet has the lowest prediction errors), which is sensible as these bulk structures were included in the U-MLP training data. The errors for surface energies are much more significant than for bulk, which is the result of these surfaces not being present in the training data. Surface energy prediction errors with M3GNet and CHGNet show systematic underprediction and MACE-MP0 shows multiple instances of overprediction. Focassio et al. also show that targeted MLPs like MTP and NequIP can have lower errors for predicting properties of specific systems vs. the zero-shot U-MLP predictions, improving accuracy at the cost of losing generality. The third benchmark from Deng et al.[74] shows the underprediction of energy and forces by U-MLP in a series of material modeling tasks, including surface energy, defect energy, mixing energy, phonon vibrations, ion migration barriers, etc. The observation of underpredicted properties aligns with the report by Focassio et al. The underpredicted energies and forces are attributed to a systematic softening of the U-MLP PES, where the U-MLPs are found to predict smoother energy landscapes than the real PES described by DFT. The author claimed the softening effect is driven by the biased sampling in U-MLP training dataset, where the training atomic configurations are taken from DFT ionic relaxations and are therefore close to local PES minima. The fourth benchmarking work we discuss here is from Riebesell et al., [75] who focused on the ability of U-MLPs and other GNN-based ML models (e.g., MEGNet, ALIGNN) to predict stable materials (i.e., materials with a convex hull energy within some threshold, chosen as being on or below the Materials Project training data convex hull). They tested these models on the dataset from Wang et al.,[76] which consists of unrelaxed structures of materials less well-sampled in the Materials Project and was generated by a chemical-similarity based element substitution process using structures from the Materials Project. They found that all three U-MLP models outperformed all other models, and that, in particular, MACE-MP0 performed best for discovering new stable materials, where the classification F1 scores for finding stable materials followed the order of 0.67 (MACE-MP0) > 0.61 (CHGNet) > 0.57 (M3GNet) > (everything else). The MACE-MP0 and CHGNet models had MAE values of convex hull energy of 60 meV/atom. Finally, work from Casillas-Trujillo et al. sought to evaluate the ability of M3GNet, CHGNet and MACE-MP0 to predict metallic alloy mixing thermodynamics. A striking result of their work is that none of these 3 U-MLPs were able to accurately reproduce the mixing energies of metallic binary alloys in adequate agreement with DFT results.[77] These findings

point to the need for careful benchmarking when pursuing the use of U-MLPs for a new problem of interest, and, if sufficient accuracy is not obtained, the consideration of carefully selected additional training data to fine-tune the U-MLP to obtain enhanced accuracy. To this end, recent work from Wines and Choudhary established the Computational High-Performance Infrastructure for Predictive Simulation-based Force Fields (CHIPS-FF), which is an open-source infrastructure specifically tailored for benchmarking materials properties predicted with various U-MLPs.[78] The problems identified in the above benchmarking studies suggest a benefit to a focusing on not just force and energy errors but also quantitative assessment of U-MLP errors on physically relevant properties, e.g., surface energies, defect energies, mixing behavior, elastic constants, etc. Testing of such behavior will benefit from the careful generation of sophisticated test datasets to assess U-MLP performance. There are challenges for how to do this effectively since, once a test set is established, it is tempting for the community to begin effectively fitting new potentials to minimize errors on these test sets, which may inadvertently create undesirable performance of the potential with respect to other properties. Developing and properly utilizing such test datasets is expected to play an important role in the refinement of U-MLPs. It is important to note the that the limited ability for low energy and force errors on training and test data to assure good performance in predicting important materials properties is not limited to U-MLPs and is a challenge for MLPs. We discuss this issue further in **Sec. 7.5.3**.

U-MLPs are expected to keep improving rapidly. Such improvements can come from simply refitting a potential to more data e.g., as already demonstrated by Takamoto et al.[37,58,79] Improvement can also come from expanding the underlying MLP formalism to include new pieces of physics, as was done by Anstine et al. to include vdW and electrostatic contributions to the total energy of the organic U-MLP AIMNet2,[18] and the addition of dispersion and vdW interactions to MACE-MP0 despite the potential only being trained on PBE-level DFT data.[32] A different and more subtle method of using additional data to improve a U-MLP is through fine-tuning of an existing model. Fine tuning is a process by which a large NN model that is already trained has its weights only slightly altered to match a small amount of new data, with the goal to keep the model accuracy on its original training data while increasing the model accuracy on the new data. Fine-tuning typically involves updating a small fraction of the weights, typically in layers involved in just the final steps before output. Such fine-tuning has been widely applied in other ML problems (e.g., computer vision and language models). Through this approach, U-MLPs may form the basis for more focused models that can be fine-tuned using new data comprising more specific chemical or structural families of materials or molecules. Since the weights in the U-MLPs have been pre-conditioned on comprehensive datasets, the fine-tuning process typically requires fewer data compared to fitting a new potential, and may result in lower errors than training from scratch.[28,74,80] For example, Merchant et al. found that the error of a fine-tuned U-MLP also follows a power-law as a function of its pretraining data size[28] (i.e. larger pre-training dataset sizes led to better downstream fine-tuned U-MLP's. The M3GNet, CHGNet and MACE Python packages already allow for fine-tuning, so this approach can be readily explored by users.

U-MLPs have several promising use cases. The first is that they may drastically speed up DFT calculations by providing a means to quickly relax a set of atomic positions much closer to equilibrium positions prior to running a full DFT calculation. In their work on developing M3GNet, Chen and Ong discuss how such speedup may reduce DFT calculation time for relaxing material structures by a factor of three.[41] This application is largely insensitive to inaccuracies in the U-MLP since the final output is from a full *ab initio* calculation and it is therefore extremely appealing. One could imagine it becoming standard practice and having a large impact, cutting typical *ab initio* calculation times significantly across potentially billions of future calculations. A second use case is replacing and expanding beyond AIMD. Similar to all MLPs, U-MLPs are useful for simulating large-scale, long-time dynamic phenomena inaccessible to current AIMD length and time scales. Such speed-up of DFT and MD studies has the potential for disruptive transformation of atomistic modeling, potentially impacting thousands of studies each year. A third use case is materials exploration. Different

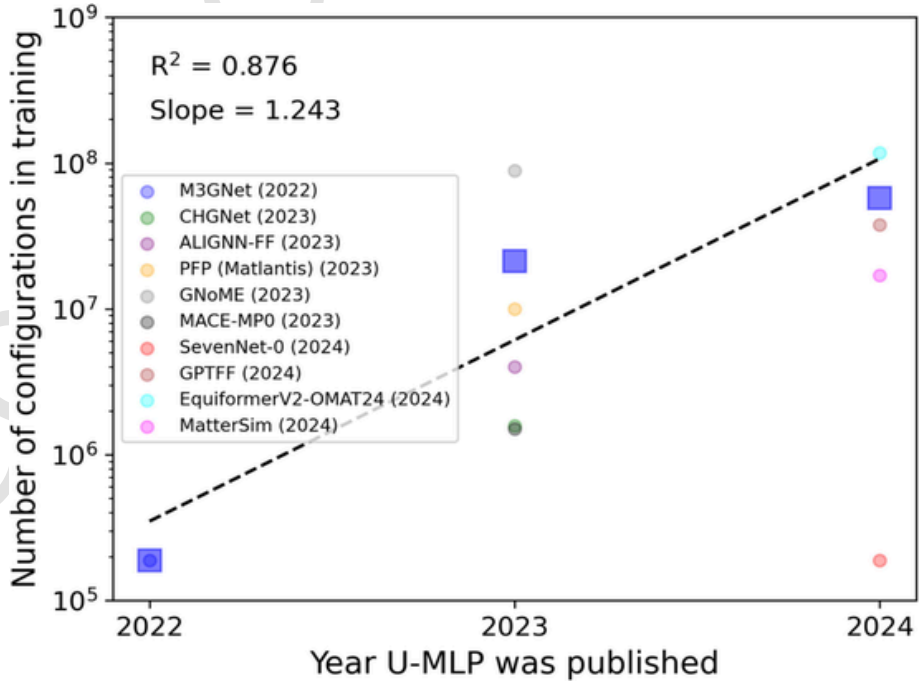


Fig. 4. Evolution of DFT database size used to train U-MLPs over time. The small circle points are values for individual U-MLPs, and the large blue squares are the average for a given year (note that SevenNet-0 was not included in the average for 2024). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

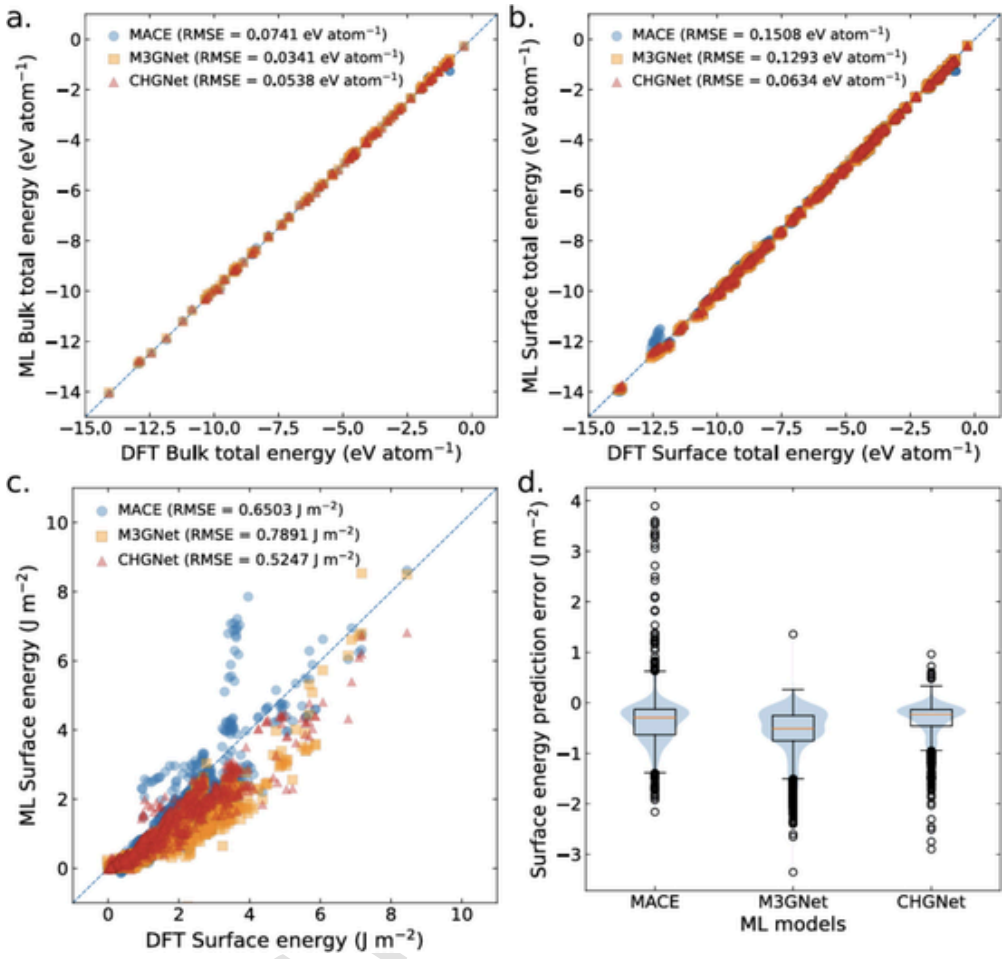


Fig. 5. Benchmarking performance of U-MLP models for predicting bulk and surface total energies and surface energies. ML vs. DFT total energies for (a) bulk and (b) surface. (c) ML predicted vs. DFT-calculated surface energies. (d) Data from (c) but plotted as a box-and-whisker plot of the ML vs. DFT residuals. Figure adapted with permission from Ref. [73].

from more targeted MLPs, U-MLPs have a much broader domain of applicability, increasing the chemical and structural complexity of systems that can be modeled with typically a small or minimal loss in accuracy. This makes U-MLPs particularly powerful for exploring many chemistries and structures, e.g., screening for a certain property like Li-ion conductivity or high elastic modulus. In particular, the lack of scaling issues with many components makes U-MLPs uniquely positioned for exploration of chemically and structurally complex multicomponent materials with, for example, >5 species. There is thus a massive opportunity to screen materials properties across the periodic table which was only possible with computationally expensive *ab initio* calculations in the past, but which could be made roughly 1000× faster for even modest-size unit cells with the aid of U-MLPs. As a demonstration of the beginnings of such an approach, Chen and Ong developed matterverse.ai, a Materials Project-like repository containing millions of hypothetical structures generated using physics-based considerations of reasonable materials structures and chemistries, and for which formation energies were subsequently calculated and screened using the M3GNet potential.[41] Similarly, Merchant et al. used a GNN model that directly predicts the formation energy of a crystal to propose 381,000 new stable (at T = 0 K) materials, expanding the number of known stable inorganic materials by nearly an order of magnitude. Given the rapid advances in generative AI, one can imagine the possibilities of combining generative inverse materials design approaches together with U-MLPs for fast materials exploration and screening, for example using tools like the Crystal Diffusion Variational Autoencoder (CDVAE) of Xie et al.[81,82], the MatterGen model of Zeni et al.[83], the Symmetry-aware Hierarchical Architecture for Flow-based Traversal (SHAFT) model of Nguyen et al.[84], a diffusion probabilistic model employing unified crystal representations of materials (UniMat) from Yang et al.,[85] or even using large language models trained to produce stable crystal structures.[86,87] Joining generative and U-MLP methods may provide a powerful new way to discover exceptional new materials that would not have been considered by way of conventional screening approaches.

Presently, the main drawbacks of U-MLPs include the same limitations as noted for more targeted MLPs (see **Sec. 9**) with the additional (and quite major) limitation that its true domain of applicability is quite uncertain. While U-MLPs are much broader in their domain than targeted MLPs, the currently available models almost certainly have many areas of major weakness that cannot be easily predicted in advance. For example, using a U-MLP to study Li diffusion in solid electrolytes might provide excellent diffusivity values for 95 % of the materials but be quite far off for 5 % of considered materials.

We discuss some strategies for the effective use of U-MLPs in their present stage of development in **Sec. 7**. U-MLPs are also generally slower than targeted MLPs, as noted in the discussion of MLP execution speed in **Sec. 6**. That said, there is an enormous advantage to a large, centralized effort around one or a few U-MLPs. These advantages include the ability to efficiently integrate state-of-the-art improvements,

e.g., adding long-range forces, speed optimizations, multi-fidelity learning, fine-tuning, uncertainty quantification, etc. It may be that the aggregation in one place of all the best practices and state-of-the-art approaches helps grow the value of U-MLPs over targeted MLPs. Overall, U-MLPs represent a very exciting advance of MLPs that will likely have a significant impact on the field of atomistic modeling. The coupled facts that a single potential (i) may soon produce energy, force, and stress values (and perhaps additional properties, such as magnetic moments) with near *ab initio*-level accuracy and order of magnitude more than *ab initio* speed, (ii) can be applied to almost every chemically relevant element in the periodic table, and (iii) can include increasingly complex physics, offers the tantalizing possibility of future U-MLPs functioning as a truly foundational model for materials modeling, in turn replacing a significant fraction of explicit quantum mechanical calculations with no need for explicit training. Fully realizing the potential of U-MLPs would allow researchers to quickly and easily explore problems that are practically inaccessible to present physics-based approaches and greatly increase the overall impact of atomic-scale materials modeling.

6. Execution (Inference) speed of MLPs

Speed for execution of an MLP is important when one is performing a large number of calculations, which might occur during long MD runs or large-scale searches of configuration and chemical spaces. Key issues to consider for timing are: the processor used for calculations (speed of CPU, GPU, or other hardware), system size, and MLP type (e.g., complexity, where increasing complexity generally corresponds to greater accuracy and slower execution). It is very difficult to quantitatively assess the speed of MLPs unless one makes a direct comparison of the same calculations with proper controls for hardware, hyperparameter settings, etc. However, there are some relevant studies available, and qualitative trends can be determined from different performance reports in the literature. We stress that the values given here should be treated very cautiously as qualitative guides and careful benchmarking for your project should be part of any extensive study where speed is an issue. A common metric for assessing performance that allows for some comparison across different numbers of atoms, processors, and steps from MD or other multi-step simulations is processor-seconds per atom per step, $\left(\frac{N_{proc}}{N_{atom}N_{step}}\right)\tau_{sim}\left[\frac{proc-s}{\frac{atom}{step}}\right]$, where N_{atom} is the number of atoms [88], N_{step} is the number of steps in the simulation (where one step of an MLP involves evaluation of the total energy and forces on all the atoms for one atomic configuration, e.g., as might occur during one MD timestep), N_{proc} is the number of processors being used), and τ_{sim} is the wall-clock time required to execute N_{step} steps. The units for each measure are given in brackets. Note that processors could be either individual cores on a multicore CPU or entire GPU processors. Typical nodes on high performance computing resources may contain dozens of cores and up to 6 or more GPUs. As long as each processor has a sufficiently large number of atoms to work with, the performance in proc-s/atom/step will be insensitive to both N_{atom} and N_{proc}. The results discussed here will mostly be approximately in this linear scaling regime. An exception to this is the $N_{proc} = 1$ special case, where simulations are run on a single core or single GPU. Performance here is usually significantly better than larger scale parallel calculations with $N_{proc} \gg 1$, where there is additional overhead of MPI network communication. For the $N_{proc} = 1$ special case, performance is given in units of simply s/atom/step. All performance results are based on a typical state-of-the-art CPU or GPU from the last few years (relative to 2023). For CPUs, these provide about 10^{11} floating point operations per second (FLOPS) and for GPUs (e.g., NVIDIA® V100 Tensor Core GPU) these are about 10^{12} FLOPS. Note that in the following discussions we will be giving approximate performance values and thus generally round to the nearest order of magnitude.

First, we consider performance for cases running on a single CPU or GPU processor under close-to-optimal conditions, with a reasonable system size (e.g., 100–1000 atoms) that can fit into memory limits on the CPU/GPU. Good scaling for parallel execution up to very large system sizes has been achieved and will be discussed more below. Timing values for a number of explicit AEF type MLPs (see **Sec. 4.3**) under different levels of complexity (i.e., basis set size) are shown in Fig. 6.[1, 60] Well-fit MLPs of the explicit AEF type range from about 10^{-5} to 10^{-3} s/atom/step (note that this is just proc-s/atom/step for one processor) depending on the number of degrees of freedom used, typically set by the number of terms that are included in the basis function expansions.

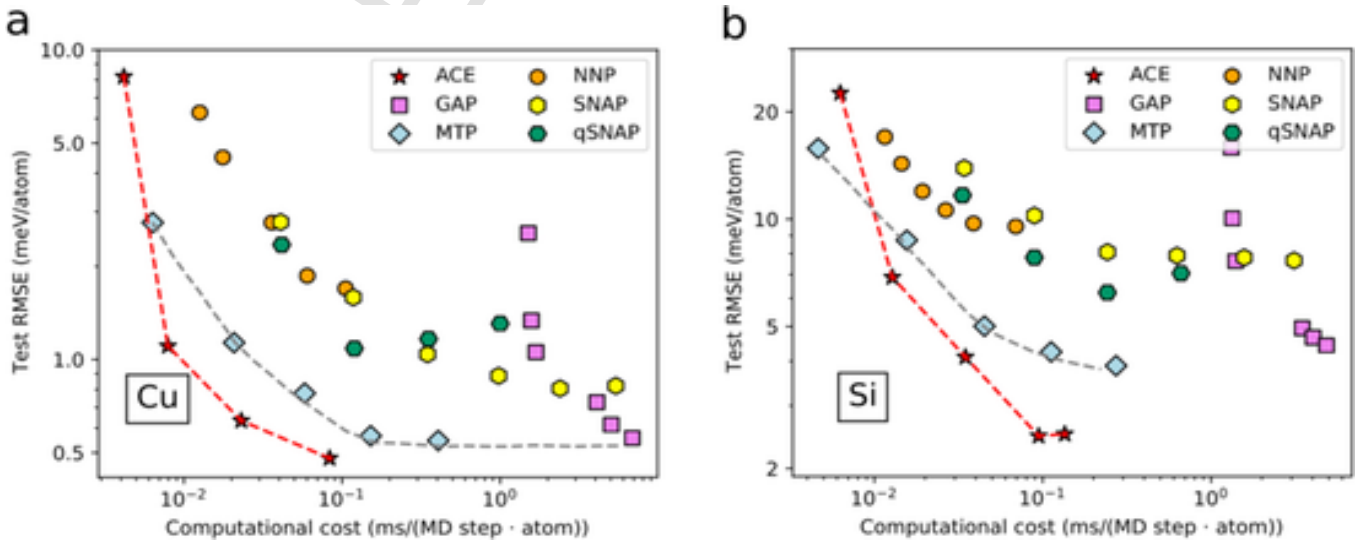


Fig. 6. Trends in computational cost (speed of the MLP) for a set of major MLPs for (a) Cu and (b) Si molecular dynamics calculations (done on one CPU with 108 atoms for 2500 steps). The varying colors correspond to different MLPs and the points for each color correspond to larger basis function sets, which generally lead to greater accuracy and larger computational cost. The abbreviations are Atomic Cluster Expansion (ACE), Gaussian Approximation Potentials (GAP), Moment Tensor Potentials (MTP), Neural Network Potential (NNP), Spectral Neighbor Analysis Potential (SNAP), and quadratic SNAP (qSNAP). Figure with data originally from Ref. reproduced with permission from Ref. [60][1].

A typical speed of the faster explicit AEF type MLPs (e.g., MTP and ACE) is about 10^{-4} s/atom/step. As a concrete example of timings, Bernstein evaluated GAP, ACE and MACE potentials for 1024-atom cells of Cu_xAl_{1-x} alloys using MD.[88] He found that ACE timings ranged from about 0.06×10^{-3} to 0.18×10^{-3} s/atom/step, while GAP was slower at 2.1×10^{-3} s/atom/step, both computed for one processor. MACE timings ranged from 0.042×10^{-3} to 0.12×10^{-3} s/atom/step on a single NVIDIA A100 GPU. For reference, timing is about 10^{3} s/atom/step for standard well-converged DFT in a state-of-the-art code for ~ 100 atom unit cells of a typical set of elements, 10^{-3} s/atom/step for a ReaxFF potential (one of the most complex physics-based traditional potentials), and 10^{-6} s/atom/step for the Lennard-Jones and EAM interatomic potentials (some of the fastest physics-based potentials).[89].

Next, we consider the timing of deep learning-based MLPs. Deep learning MLPs are generally similar to, or somewhat slower, than non-deep learning explicit AEF approaches, although it can be hard to compare as the former are often run on GPUs. Nonetheless, some results exist that give a qualitative sense of the relative speeds of deep learning MLPs under different conditions. DeepMD typically performs at about 10^{-3} proc-s/atom/step on a CPU, and was accelerated 39 times (so about 10^{-4} proc-s/atom/step) on a GPU in a direct comparison.[90] M3GNet,[41] which models a very large number of elements (89) and is what we refer to as a U-MLP (see **Sec. 5**), takes about 10^{-3} proc-s/atom/step on a single CPU to perform a structural relaxation of $K_{57}Se_{34}$, an example chosen for its large energy change during relaxation. Recent testing of the PFP U-MLP from Matlantis gave about 10^{-3} s/atom/step on a GPU for 100–1000 atom unit cells running MD in LAMMPS.[91] These values were about 5 times slower than well-converged MTP and ACE fits on identical systems run on a single CPU, and about 50–100 times slower than the same runs on a large set of CPUs.

An interesting developing area to increase MLP speed is the ultra-fast approach,[42] which uses computationally cheap spline functions to describe the atomic environments and linear regressions for energy/force predictions. The potentials resulting from the ultra-fast approach are extremely fast compared to existing MLPs at the price of limited flexibility and possibly greater errors for complex systems. For example, such potentials are about 10^{3} times faster than typical explicit AEF MLPs, with similar prediction accuracy to SNAP, GAP, and MTP on some test cases, putting them at about 10^{-6} s/atom/step and comparable to the fastest simple PBPs.

Efficient architectures and scaling up the number of CPUs and GPUs used to evaluate MLPs can lead to large speedups, which is particularly useful for the somewhat slower deep learning methods. Note that these timing values are somewhat faster than above, likely because the inclusion of more atoms is allowing for more efficient use of the processors. DeepMD achieved about 10^{-5} proc-s/atom/step with about 127 million Cu atoms, and SNAP achieved 10^{-6} proc-s/atom/step on about 20 billion C atoms, both running on 27,900 GPUs (4650 nodes on the Summit machine).[92] A deep learning MLP particularly optimized for scaling and performance is Allegro,[13] which uses a strictly local equivariant neural network and ACE-like atomic features, and while it can be executed on CPUs, it is best run on GPUs. Allegro models of water achieved about 10^{-5} proc-s/atom/step with 4, 64, 1024 GPUs and 10^{5}, 10^{6}, 10^{7} atoms, respectively.[93] Note that the choice of hyperparameters (i.e., complexity) can change this approximate timing by an order of magnitude and that this is for an optimally tuned MLP. To increase execution speed, typically a more complex model is first used to verify the fidelity of the training data and learning process, before being reduced in size, while still reproducing a target property of interest with sufficient accuracy.

As another example of an MLP particularly optimized for scale and speed, the GPU implementation of the FLARE potential,[25] based on C++ with a Python wrapper, was used to model heterogeneous catalysis of H_2/Pt(111) for 0.5 trillion atoms on 27,336 GPUs nodes, achieving 10^{-6} proc-s/atom/step.[94] However, the speed of FLARE on CPUs reduces significantly compared to GPUs, down to roughly 10^{-3} proc-s/atom/step. It is useful to note that the performance for SNAP, FLARE, and Allegro all begin to deviate significantly from linear scaling of inverse time with processors (constant B values) by around 10^{5} atoms/GPU for the Summit hardware used in these tests (NVIDIA V100-16 GB GPUs). While these timings are very impressive, it is important to realize that PBPs can also take advantage of parallelization and GPUs. For example, a GPU-accelerated classical force fields model based on the Martini potential recently achieved 6 microseconds/day on 136,000 particles ($B = 10^{-9}$ proc-s/atom/step) using six V100 GPUs.[95].

In summary, from the above-discussed timings we can learn at least two important lessons. The first lesson is that, similar to PBPs, scaling up to even hundreds of billions of atoms is possible for some MLPs. These calculations generally require multiple GPUs, which can be a challenge to access, but options are becoming increasingly available (see discussion of infrastructure for MLPs in **Sec. 8**). The second lesson is that the general trend of speed we noted on one CPU, which is that simple PBPs are fastest, followed by explicit AEF MLPs, then finally implicit AEF deep learning MLPs, largely still holds with larger-scale calculations. That said, we stress that the details of the MLP fit and optimization can matter a lot for large-scale calculations, so one should choose an optimal approach carefully if pursuing such studies.

7. MLP choices – what should i use when?

When choosing MLPs, many factors can be considered. We list a few of these factors in this section and provide some guidance on how to think about each of them. We start from basic aspects of hardware, accuracy, and speed and then progress to the details of pursuing a specific MLP.

7.1. Hardware resources

Hardware resources could be an initial deciding factor in choosing MLPs both when fitting a new potential or using a pre-trained potential. Generally, explicit AEF MLPs such as MTP, ACE, SNAP, and GAP have fewer parameters and functions than NN-based MLPs and run well on CPUs. On the other hand, NN– and GNN-based MLPs mostly rely on GPUs. Some potentials, like MTP, can presently only be run on CPUs, while ACE is faster when fit using GPUs but can be used for MD simulations on both CPUs and GPUs. NN-based MLPs are primarily created to be fit and used with GPUs, although they can be run on CPUs, with typically a 10-100× slowdown on CPU vs. GPU calculations (see discussion of MLP timings in **Sec. 6**). These trends generally suggest that if you only have access to CPUs, then explicit AEF MLPs are likely best as they will be certain to run and will typically run with reasonable speed. If you have access to GPUs, then both explicit AEF and NN-based MLPs are potentially good choices. Given the growing importance of U-MLPs and the use of GPUs in training and executing many MLPs, it is likely advisable to have access to at least one high-performing GPU if you are planning extensive use of MLPs. In addition to the discussion above, in Matlantis, which is commercially deployed as SaaS, PFP is provided via an API, allowing users to execute inference without considering the environment setting and optimization of computing devices. In practice, the inference is executed in backend GPUs or specific deep learning accelerators named MN-Core series. [96].

7.2. Speed requirements

The overall simulation time depends on the size of the system, the number of execution steps in the simulation, available hardware resources, and the computational cost of the MLP. Assuming the first three factors are fixed by the project and infrastructure needs of the user, the MLP framework determines the overall simulation time. As

previously discussed in **Sec. 6**, explicit AEF MLPs are about 10-100× faster than implicit AEF deep learning MLPs. If the computational cost is not a limitation, deep learning MLPs typically provide the highest accuracy and may be adopted. Otherwise, the user could opt for any of the explicit AEF MLPs that provide the desired accuracy.

7.3. Accuracy requirements

While the promise of MLPs is to achieve any desired property accuracy with respect to *ab initio* methods, in practice there is an accuracy limit of MLPs to keep the computational cost of the MLP reasonable given the available resources. One limiter of MLP accuracy stemmed from the insufficient description of the atomic environment in the earlier MLPs such as Behler-Parrinello NNs, GAP, and SNAP. More recent MTP and ACE formalisms introduced new methods to give a complete description of the atomic environment and used linear regression to learn the PES, enabling an increase in the accuracy of MLPs while keeping the computational cost tractable. In recent years, it has been shown that equivariant GNNs can achieve very high accuracies with a practical computational cost, where NequIP, Allegro, TeaNet, SO3krates[97] and MACE are examples of such approaches. Thus, the authors recommend that if GPUs are available, training with equivariant models such as NequIP, Allegro, TeaNet, SO3krates or MACE is likely to yield the highest accuracy. Furthermore, it has been found that higher accuracy may be obtained by fine-tuning a pre-trained potential as opposed to to training a new MLP from scratch, even for tasks that were out-of-distribution compared to the training data.[80] In a few personal experiences by the authors, we have found that for real systems, with abundant training data available (meaning we could keep running more DFT as needed until we see little improvement in the potential), the AEF methods like ACE tend to have root mean squared error (RMSE) on energies and forces that are 2–3 times those of GNN methods like MACE. If there are only CPUs at hand, the authors suggest MTP or ACE. Implementations of these various methods are likely to improve and diversify utilizing popular hardware, and therefore we expect the field to change rapidly.

7.4. Using A pre-trained potential

Depending on the type of study, one may decide to use a pre-trained potential or to fit a potential from scratch. It will likely save a lot of time if one can start from a pre-trained potential, so this is a logical first step to explore. Pre-trained potentials may be found in online repositories or by searching through scientific articles. For example, pre-trained targeted MLPs for specific systems (e.g., GAP potential for Cu) can be found on the NIST Interatomic Potentials Repository and the Open Knowledgebase of Interatomic Models (OpenKIM).[98–100] When deciding to use a pretrained potential, one must make sure that the potential is suitable for the study. Given the recent availability of U-MLPs and their ease of use across many systems, they represent an appealing option, and importing pre-trained versions of U-MLPs from their respective repositories is straightforward.[101–103] However, although U-MLPs generally have low energy and force errors compared to their *ab initio* training data, their ability to predict accurate materials properties is not ensured by these low errors (see **Sec. 5**) and they have not been thoroughly validated for accurate prediction of materials properties across most systems. It is therefore quite possible that despite some impressive successes (see **Sec. 5**) that many properties, from vacancy formation energies to melting temperatures, may be incorrectly predicted. Furthermore, U-MLPs can be slower than other MLP or PBP approaches (see **Sec. 6**), so speed requirements should be considered. However, given the rapid rise of such U-MLPs in just the past couple of years, it is likely that increased property prediction benchmarking will be available, and iterative improvements to the U-MLPs, e.g., through fine-tuning, will further aid in improving their accuracy and generalizability. For the time being, there are a few simple strategies one can use to apply U-MLPs most effectively, which we summarize here:

1. Validate the U-MLP predicted energies and forces for your system of interest. One way to ensure the accuracy of an untested MLP for a specific system and purpose is to run some relevant *ab initio* simulations for your problem and compare the *ab initio* and U-MLP energies and forces. One should be careful to choose *ab initio* settings such as functional, energy cutoff, k-point density, etc., consistent with the training data used for the MLP (e.g., choosing the right pseudopotentials and Hubbard U values for GGA + U calculations). Good agreement is strong support that the U-MLP is applicable to your system. Such a benchmark can be done with just a handful of static *ab initio* calculations on small unit cells and therefore can be quite fast. The use of benchmarks that directly relate to the property of interest, e.g., an activated state for a chemical reaction or few points on a gamma surface for stacking fault energies, are likely best.
2. Validate the U-MLP property predictions for your system. In many cases, the benchmarking described above can be easily extended to include comparing *ab initio* and U-MLP calculation of specific properties of interest, e.g., a set of phonon dispersion curves, diffusion coefficients or defect formation energies, particularly for small systems or simplified cases. Good agreement on target properties provides even greater confidence in the U-MLP than just similar energies and forces on select structures. For example, one might calculate diffusion coefficients in a small unit cell with *ab initio* and the U-MLP and, if similar results are achieved, apply the U-MLP to much larger systems or different compositions.
3. Apply U-MLPs to problems that can easily detect failures or are not overly sensitive to failures. Many applications might not suffer too much from intermittent failures of the U-MLP. For example, using a U-MLP to pre-relax other *ab initio* calculations is a very robust application tolerant to U-MLP failures since the final calculated result does not directly depend on the accuracy of the U-MLP. In addition, failures in the pre-relaxing can be easily identified and corrected by checking against the corresponding *ab initio* relaxation. As noted above, in developing the M3GNet U-MLP, Chen and Ong comment that pre-relaxing hypothetical structures with their U-MLP before performing *ab initio* calculations resulted in approximately 3× time savings compared to running *ab initio* on un-relaxed structures.[41] Another example might be using U-MLPs for an initial screening of a large set of candidate materials for a specific property, where a highly accurate calculation may not be necessary in the initial steps. Failures of the U-MLP might lead to false positives (keeping unpromising materials) or false negatives (removing promising materials) but later screening with full *ab initio* calculations can catch the false positives, and, typically, screening is often more focused on getting a few successes than ensuring no false negatives. A final example is generating physically relevant atomic configurations (which need to be calculated with *ab initio* methods later) for training a more specific potentials, a way in which U-MLPs might help accelerate the development of more targeted MLPs.

Despite the exciting potential of U-MLPs, the high levels of uncertainty in their applicability means that many practitioners presently still either fit their own potential or use a pre-trained potential that is specifically fit for the material under investigation. As a new trend different from this, some early adopter researchers have begun to perform calculations without finetuning. For example, Matlantis provides pre-trained U-MLP (PFP), with the aim of allowing users to do practical simulations without having to perform finetuning. In all cases, it still matters that the training data used for the potential is consistent with the

type of study being considered, both in terms of atomic structures, chemical states, and relevant physics. For example, (i) an MLP that is trained on pristine crystalline phases and crystals with stacking faults and vacancies may not be appropriate to conduct a study on the amorphous phases of the same material, (ii) an MLP trained on low valence transition metal states might not represent high valence states of these same metals well, or (iii) a potential trained on *ab initio* methods like the DFT-PBE functional may not be suitable for layered materials or molten salts, where vdW contributions are significant (although in this case the potential might be corrected by empirical vdW corrections). In general, for all MLPs, one should validate the energy, force, and property predictions as much as possible for a specific use case unless it very closely matches previously published or well-validated work. Many considerations related to the issues above are likely relevant for choosing an optimal pre-trained potential but, since the potential has already been developed, it is likely that the original authors have already taken these items into consideration (e.g., choosing the right potential for the hardware they ran on, etc.). Thus, one can take guidance from the earlier work about optimal use. That said, it might still be useful to have a sense of how different potentials behave related to the above issues, and in the next section we summarize the key concerns in the context of fitting a new potential.

7.5. Fitting A new potential: general workflow

7.5.1. Basic ideas

If no pre-trained potential is available, one will need to choose an MLP framework and fit a potential from scratch. In this section, we provide some strategies and guidance from hands-on experience to help new users approach choosing an MLP to fit. In addition to the above considerations when using a pre-trained potential, a few new factors become relevant when fitting your own potentials, which we discuss here.

The basic idea behind fitting MLPs is the same as in almost all regression ML problems. One defines a loss function and adjusts the parameters of the ML model, typically using some kind of matrix inversion or backpropagation, until the loss function is minimized. For MLPs, the loss function is usually a weighted sum of RMSEs on a few targets, which are usually forces on atoms and total energy, but can also include other properties such as stress tensor, virial, polarizability, etc. Typically, the most important terms are the RMSE in forces and energy and these are standard to report. It is important to realize that, although MLP fitting is similar to other ML models, it is helpful to use domain knowledge (from physics, chemistry, and materials science) to perform successful training and assessment, which we call *science-informed fitting*. Science-informed fitting is very helpful, at least at present, because the MLP fit will almost certainly not be perfect for all possible configurations of atoms, so the user is suggested to apply their domain knowledge to develop a model that is adequate for their needs.

At present, there is no agreed-upon standard or widely accepted optimal workflow for fitting an MLP. However, multiple authors have provided very helpful articles that cover the major considerations and provide excellent practical guidance.[104–107] Here, we describe the typical general workflow, and then go into some of the detailed questions and choices associated with its implementation. In addition, a standard set of procedures and software for generating or acquiring training data, fitting, comparison, and deployment of MLPs is provided in **Sec. 8**. The general approach is to generate an initial set of *ab initio* data $\{s_1\}$, consisting of atomic configurations related to your problem of interest (e.g., liquid configurations for studying a melt, different vibrational modes for studying phonons, or multiple distortions for studying molecular systems). Then, fit an initial potential to ~ 80 % of $\{s_1\}$ and test on the left-out ~ 20 % of $\{s_1\}$ to assess accuracy on energy and forces (this approach and the details below can be readily extended to other targets if they are used). In the case of training GNNs, it is common practice to train on 80 % of the data, reserving 10 % for validation (to guide the GNN training process) and 10 % for testing. If the fit quality is not adequate (e.g., the force and/or energy RMSE is too high), one develops additional data, adds it to $\{s_1\}$ to form a new data set we call $\{s_2\}$, and then performs a similar assessment. If $\{s_1\}$ is sufficiently large, then no iterations may be needed. If the system is complex and/or relatively small data sets are being added at each step, then this might take many iterations. Atomic configurations for different $\{s_i\}$ are generally determined based on user intuitions about important configurations for the application of interest (e.g., known stable compounds in the material), independent samples from MD trajectories, guidance from active learning (discussed below), or some combination of all of these, depending on the application. The required amount of data to obtain a desirable fit can vary, but for typical systems with 3–4 species, the number of total energies N_E and the number of forces N_F used in training are approximately $N_E \sim 10^3$ and $N_F \sim 10^5$. This estimate is very approximate, and model type and architecture (e.g., MTP vs. ACE, equivariant vs. invariant features, etc.) can also affect the results. In particular, for deep learning methods, the error vs. amount of training data (the learning curve) is expected to follow a power law, but the power law exponent can depend on significantly on the details of the MLP. [108].

7.5.2. Determination of test data

The first potential fitting issue we address is strategies for determining useful test data sets for validating the MLP fit. Random cross-validation (CV) or k-fold CV are both reasonable if the data is not highly correlated. However, if the data has many similar conditions, e.g., as occurs for data generated from AIMD trajectories or small perturbations to existing structures, then these random CV approaches will yield overly optimistic predictions. The predictions will be overly optimistic due to the "twin" problem, where extremely similar data is present in both the train and test sets, and the model predictions are thus indicative of data that looks just like the training data. In the case of highly correlated or otherwise similar data, one can assess the potential more robustly by comparing *ab initio* and MLP predictions from new conditions, e.g., MD at a new temperature or MD from a much later time than that used during training. An even better way to assess the MLP in such cases is to apply the MLP in expected or near-to-expected use cases and check errors on select configurations from those conditions. For example, assume you are trying to predict the diffusion of Li in a solid-state electrolyte at low or even room temperature. The bulk of the training data might be AIMD trajectories at higher temperatures so that many Li hops occur. An example of good test data would be to simulate low-temperature hopping with the MLP, extract configurations where the hopping occurs, run these with *ab initio* methods, and compare the *ab initio* and MLP energies and forces for those configurations as a test. Obviously, when possible, testing the ability of the potential to predict the properties of interest is essential. Continuing the example above, one should be sure that the *ab initio* and MLP-predicted Li diffusion match in the higher temperature conditions where the *ab initio* simulations are reliable and can be well-converged. However, extensive property testing is generally difficult as it can be challenging to have a robust ground truth, proper simulations often take a long time for the ground truth even using the MLP, and there are generally relatively few property values for comparison (e.g., one might have only 5–10 densities or diffusion coefficients as compared to many thousands of forces). This disparity makes it desirable to know as much as possible that a potential will be robust before starting significant property exploration. This robustness is generally assessed through energy and force errors and brings us to the second issue.

7.5.3. Required energy and force accuracy

The second potential fitting issue we address is what accuracy of energies and forces is needed in the test data to ensure a useful MLP, by which we mean an MLP that can be used for a wide range of simulations

and yields accurate predictions for properties of interest. At present, there is no exact answer to this question. The accuracy that can be achieved will depend on the conditions being explored and the range of elemental species and structures considered, as well as the type of MLP. For example, a simple liquid phase of just one element may yield much smaller errors, both relative and absolute, than modeling oxidation of a complex surface at high temperatures. However, there are still challenges in learning even single-element systems. For example, Owen, et al.[109] found that early transition metals have higher relative errors than late platinum- and coinage-group elements. This apparent difficulty in learning is attributed to the sharp *d*-electron density of states above and below the Fermi level, resulting in complex physics which makes the PES difficult to learn. The relative energy and force errors in their study of transition metals ranged over about a factor of 10. That said, typical values for energy and force errors are in the ranges of 1–10 meV/atom and 20–40 meV/Å, respectively, for a very good fit, although force errors of up to around 100–200 meV/Å have been reported in nominally successful MLPs.[1,109] Very accurately trained MLPs can achieve errors for energies, forces, and stress tensor components on the order of 1 meV/atom, 10 meV/Å, and 0.1 GPa, respectively, although in practice one may find somewhat higher (e.g., 2×) energy, force and stress errors which are highly system and potential dependent.

It is reasonable to assume that for a relevant and diverse set of training data, a lower RMSE on energies and forces will generally translate into more accurate property prediction. However, depending on the application, low energy, force, and stress errors may not be sufficient criteria for ensuring accurate property predictions.[110,111] In addition, an MLP trained on a large number of chemically diverse systems may exhibit energy and force RMSEs that vary widely by element or chemistry type (e.g., defects in oxides vs. elemental metals), system state (e.g., solid vs. liquid), and simulation conditions. Obviously, the MLP is at best as accurate as the *ab initio* method used to train it, so for the discussion here we will assume that the *ab initio* method yields accurate results. In the case of negligible RMSE on all atoms in all situations, it is expected that the MLP is essentially equivalent to the *ab initio* method used to train it and will ideally yield robust property prediction. However, this ideal scenario is difficult to reach in practice, due to poor predictions on outliers. RMSE values are averages over many configurations, so even MLPs with low RMSE can have outliers that have significant errors. If these outliers are important for a given property, then the prediction may not be accurate. Again, referring to the example above, an MLP trained on a large body of *ab initio* MD simulation data of a Li conducting compound may show very low RMSEs on energies and forces on all the different atom types, but still not accurately capture the activated state energy of Li during a hop (i.e., this activated state is an outlier) and therefore yield inaccurate diffusion coefficients. The result of a low RMSE but the inability of the model to capture some piece of physics is analogous to situations that often arise when developing standard ML regression models, where the model is generally reliable for interpolation tasks (test data similar to training data) but unreliable for other tasks, even when not formally extrapolating.[86].

As a concrete example of MLP extrapolation issues encountered during a study, Zhai et al.[112] demonstrated that a widely-used deep neural network potential, i.e., DeepMD, can reliably reproduce the properties of liquid bulk water but provides a less accurate description of the vapor–liquid equilibrium properties. This problem can be compounded by two potential issues: (1) The ML architecture cannot capture the essential symmetries and physics, e.g., many-body interactions; (2) The training data is not at all evenly distributed in structural or chemical space, a common issue when data is sampled from MD or biased toward widely studied compositions, leading to data imbalance issues when training robust MLPs. As discussed above in the hypothetical case of studying Li conductors, the simplest way to avoid such issues is to be sure that the training data samples as much of the relevant configuration space as possible. If one is concerned about predicting diffusion, then use training data with many activated states for hops, and if one is concerned about predicting bulk moduli, then use training data from a range of different stresses. Another way to improve predictability is to change the evaluation metrics to include force predictions on important outliers. This technique was suggested by Liu et al. when they observed that large discrepancies can still be observed in migration barriers even when defects are included in the training.[113] Considering relevant rare-event-based metrics (e.g., accuracy for diffusion hops, defects, atomic vibrations) for MLPs is important, since it is for these configurations where force errors can potentially be large.

An additional complexity in ensuring a robust potential is that small RMSE is not a guarantor of stable simulations.[111] By *stable simulations*, we mean particularly long-time (e.g., tens of nanoseconds) MD simulations.[114] There is a tendency for MLPs to become unstable during MD simulations and crash. Depending on your needs, this can make the potential useless. We hypothesize that crashing of the potential typically occurs due to the system exploring regions of configuration space where forces are not accurate and change in ways that are too fast for the MD time step to manage. This leads to errors that accumulate and eventually cause numerical instability. In other words, the numerical integration of the equations of motion being performed by the MD becomes unstable because the energies and forces are, at least during some parts of the simulation, not changing slowly on the time scale of the MD time step. Such an event is not unlikely if the potential becomes unphysical, since the MD time step, generally taken to be 1–2 fs, is tuned to be effective for a physically realistic system. This problem can be reduced by starting with progressively more varied training data. It can also be remedied by running *ab initio* calculations on configurations from the MLP simulation just before the observed instability to obtain new training data, which can stabilize the model after retraining. It is also possible to flag configurations that appear during the use of the MLP that are in some way outside the domain of the training data and running *ab initio* calculations to add these cases to the training data. The domain of the training data is typically determined using some measure of difference from the training data, e.g., active learning with D-optimality (discussed more below).[115,116] These domain-based approaches are quite effective in establishing a stable potential for MD and are widely used. Such approaches may require multiple iterations, and it is not clear *a priori* how many will be needed to achieve a stable simulation, although typically no more than 5 iterations are needed. The above discussions offer many qualitative guides for training and test data, but do not provide any concrete approach to assembling a training database, which brings us to our discussion of this third important issue.

7.5.4. Determination of training data, use of active learning

The third potential fitting issue we address is how one should choose training data. Again, this does not have a unique settled answer, but there are useful guides. The simplest approach is to use domain-specific intuition to develop a training database that is diverse, relevant, and large. This is easier than it might sound, and given the speed of modern *ab initio* methods, often quite practical. The advantage of this "intuitive structures" approach is that it is relatively easy to implement, makes good use of materials knowledge, and tends to yield a good MLP in a practical amount of time. However, the approach is almost certainly not optimal in terms of getting the best potential for the least training data, it is not readily automated, and it may not scale well to MLPs that are targeting many elements and many kinds of physics all at once. A different second approach that seeks to solve these issues is active learning, which is described next.

For the most efficient training data generation, users have a few options, and *active* learning is commonly useful. Note, by active learning we mean an iterative approach that uses the results of a collection of fits to suggest the best new training data to add for the next fit to optimize

some condition, e.g., creating an MLP with the lowest RMSE on some property. A general overview of the use of active learning to train MLPs is provided in Fig. 7. To apply active learning, models need access to uncertainty estimates during configurational sampling. When uncertainties are high, a ground truth calculation (i.e., *ab initio* calculation) is automatically invoked. The most common active learning approaches for MLPs are D-optimality,[116] Gaussian process regression,[25] querying by a committee of GNNs,[117,118] Bayesian inference force fields,[119–121] and uncertainty-driven MD simulations with bias potentials.[122,123] The active learning process typically makes use of the featurization of the atomic environment to automatically guide the search for unseen and uncorrelated atomic configurations to improve model predictability. Implementing active learning therefore requires access to the featurization used in the MLP, and is most easily applied when built into the MLP package. For instance, MLPs such as MTP, ACE, and FLARE have built-in active learning functionality in their fitting routines. It is possible to use a featurization separate from the MLP to implement one's own active learning framework. Packages such as Dscribe[124] and matminer[125] can be used to featurize the atomic configurations for developing one's own active learning or other data generation approach.

Here we give a few examples of fitting approaches used in recent studies. Attarian et al.[126] explored the intuitive structures vs. active learning based on D-optimality in a study of properties of eutectic composition FLiBe salts with an MTP potential. They found that either way of training data generation resulted in a robust potential, though the active learning approach was more efficient as it produced about the same prediction error with less than half as many training structures (600 vs. 1400 structures). Work from Vandermause et al. also compared the use of active learning vs. random sampling, and they found that active learning resulted in more efficient MLP training (i.e., lower RMSE per training data added) and an overall lower RMSE compared to random sampling.[25].

There are additional approaches for the efficient generation of training data that do not leverage active learning. For example, in a recent study using bias potentials, Kulichenko et al. merged the ideas of querying by committee and metadynamics to model the phase space of proton transfer in glycine.[122] The use of a bias potential, instead of high-temperature MD simulations, generates low and high-energy configurations, thus avoiding sampling unnecessary structural distortions. When the main purpose of active learning is to add weakly correlated or uncorrelated configurations to the training data, bias potentials may be a direct and efficient approach. As a second example, in their study of Mo, Chen et al. outlined the selection of training structures using principal component analysis, and the selection of hyperparameters using a differential evolution algorithm.[127] Their procedure, using a SNAP MLP, achieved close to DFT accuracy for elastic constants, melting point, and surface and grain boundary energies. As a third example, Vandermause et al. built a FLARE potential for vacancy and adatom diffusion in Al, where the training data was obtained on-the-fly, where select DFT calculations were performed if the GPR error bar became too large. In their MD runs, they found that the majority of the DFT calculation calls occurred near the beginning of the run, with no DFT queries occurring after 400 ps of MD time.[25].

A valuable tool for developing training data can be to use a classical PBP or a U-MLP to generate a large initial set of atomic configurations, which are then sampled intelligently to obtain DFT runs for training data. This sampling can be done with active learning, as described above. It can also be done with other approaches. For example, a clustering algorithm can be used to separate different groups of configurations based on some similar features and later a collection of configurations from each cluster is chosen to be calculated with *ab initio* methods and used as the training set. Users can take advantage of packages such as Dscribe[124] to featurize the atomic configurations and ML packages such as scikit-learn to do the clustering. More recently, enhanced sampling techniques have been utilized to accelerate the sampling of rare events and integrate that sampling with active learning procedures for the generation of training datasets for MLPs that can describe rare events.[122,123,128].

7.6. Fitting A new Potential: More specific considerations

7.6.1. Chemical complexity

As discussed in **Sec. 4.3**, a drawback of most explicit AEF MLPs as they are currently formulated is that they scale poorly with the number of species. This scaling results in a higher computational cost for systems with a higher number of species both when training and executing simulations with the potential. For example, FLARE is generally extremely fast in its execution time (see **Sec. 6**), but can scale poorly with training set size and chemical complexity. The general rule is that an update (i.e., retraining of model parameters) of the FLARE sparse Gaussian process can become prohibitively expensive when there are around N_{env} = 1,000,000 environments in its training set, where $N_{env} = O[$ (# training *ab initio* frames) × (# atoms/frame) × (# species)^2], where a frame is one set of *ab initio* calculated forces and energies (note that 1 M environments is an upper bound, where in practice FLARE users may experience slow timing and large memory requirements when using > 600 k environments).[121] The quadratic scaling with the number of species is particularly limiting for chemically complex systems since explicit AEF MLPs can handle less training data, but typi-

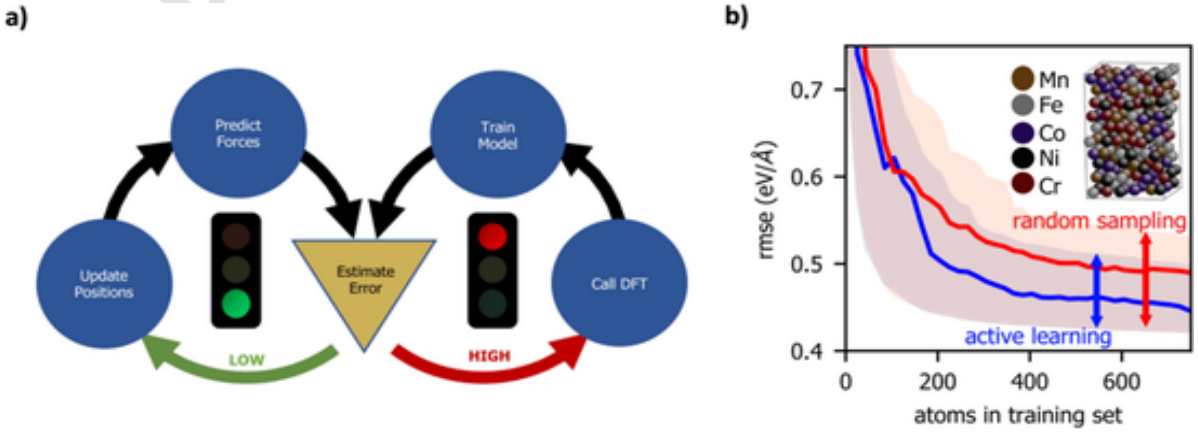


Fig. 7. (a) Overview of the use of active learning in constructing reliable MLPs. (b) Learning curve showing improved efficiency of active learning approach vs. random sampling for developing MLP of 5-component high entropy alloy. Adapted with permission from Ref. [25].

cally need sufficiently varied training data to explore the many chemical configurations. Considering the above limit for FLARE as a concrete example, offline training (where *ab initio* calculation frames have already been calculated and are available for fitting) for a system with a single species is possible for about 4000 training frames (250 atoms per frame) while for 5 species the scaling limits the user to about 150 frames, which is likely too few to fit an accurate potential. Similar issues exist for MTP, ACE, and other explicit AEF MLPs, and a brief review of the literature shows that almost all fits with these explicit AEF MLPs are to 5 or fewer chemical species. A recent attempt by the authors to fit an ACE potential to a 12 species system of chloride salts with 3500 training data using Nvidia Tesla v100-32 GB GPU failed at the very beginning and the code did not even start the training. A more in-depth discussion of scaling issues is given in **Sec. 4.3**. [29].

7.6.2. Training requirements

The difficulty of training an MLP is a key factor in choosing one that is right for your project. Key things to consider include both the amount (and potentially variety) of training data and the training time. All other things being equal (e.g., for the same chemical system and desired accuracy), the training data requirements for different potentials can be quite different. For example, molten salt FLiBe potentials were recently trained with DeepMD and MTP.[126,129] Both approaches produced excellent potentials, but the MTP fitting was achieved with less than 1 % of the amount of the DeepMD data (although it should be noted that this was not a head-to-head comparison by the same authors under identical conditions so should be taken as only a qualitative guide for the training data differences from these potentials). Early deep learning MLPs[129,130] required much more training data than explicit AEF MLPs, but this no longer seems to be true for the newer equivariant deep learning MLPs, which are much more data efficient. For example, studies with NequIP report a 1000× improvement vs. DeepMD with respect to data requirements.[131] However, even if a deep learning and explicit AEF MLP require the same amount of training data, the complexity of the former will typically cause it to train more slowly.

7.6.3. Ease of fitting (Tools and Hyperparameters)

Ease of the fitting process is also a key factor in considering which potential to use. First, it is important to have good fitting tools associated with the potential that allow for easy fitting, ideally with active learning. Most popular potentials now provide such tools, and more are being developed rapidly, so we will not say more about this requirement and just assume it is satisfied for any potentials one might consider. More fundamentally, potentials with fewer hyperparameters are significantly easier to use. This difference can be large, ranging from just one hyperparameter in MTP, which makes hyperparameter optimization trivial and fast, to many for Allegro, which can require significant experience and skill to optimize to achieve state-of-the-art results. In this regard, the authors have found that MTP is one of the easiest MLPs to fit as it only has one hyperparameter, which is called the "complexity level" parameter of MTP, and the user can start from lower levels and increase the complexity level step-by-step to achieve the desired accuracy. It should be noted that as of this writing (early 2024), MTP does not support GPU training, so with large training sets many CPU cores are required. However, this hardware limitation may be removed at any time with an update to the MTP code. Compared to MTP, the ACE potential provides much more flexibility in terms of fitting parameters for interaction between each pair, triplets, etc., of species. This flexibility creates a lot of hyperparameters, which correspond to the bond order of many-body interactions, the number of radial basis functions, and the angular resolution of the description. However, because ACE featurization allows for good physical intuition, after a few training sessions with different hyperparameters, the user gets an understanding of how to balance the hyperparameters to achieve the desired accuracy while keeping computational cost low. While this modest complexity from hyperparameter optimization may seem unimportant, it can easily increase the overall time to fit a potential by a few multiples. This is because the *ab initio* simulations and fitting efforts are largely automated, but the hyperparameter optimization is still often done somewhat sequentially and by hand. This challenge may reduce quickly as standardized hyperparameter choices emerge or more automated optimization methods become available.

7.7. Summary of considerations for choosing a potential

There are many options for possible MLPs for fitting, including those already mentioned in this paper as well as many others, and, as with many aspects in this emerging field, there is no standard consensus on the best MLPs. However, we can provide some guidance to help users navigate the options. If one needs a fast potential (e.g., simulations for tens of nanoseconds and longer) and/or one does not have access to GPUs, then explicit AEF MLPs are likely a good starting point, where we suggest starting with MTP or ACE due to their ease of fitting and high accuracy, respectively. Conversely, if one does not need a lot of speed (e.g., exploring a few thousand structural energies) and/or one has access to GPUs, then deep learning potentials are a practical option, although not necessarily required or even the best option. The requirements of accuracy, noted in **Sec. 7.3**, suggest using the more complete potentials (e.g., MTP, ACE) vs. the older forms (e.g., SNAP), due to greater potential accuracy with no obvious downsides. In particular, the work of Zuo, et al. performed a very useful comparison of different MLPs in 2020 and found that MTP was both highly accurate and very fast to execute, performing generally somewhat better than GAP, SNAP, and Behler-Parrinello NN potentials.[1] This suggests that MTP is a good potential to start with in the absence of more information. As discussed in **Sec. 4.3**, recent developments in MLP formalism have shown that essentially all of the basis functions that underlie different explicit AEF methods (e.g., ACSF, SOAP, HBFs, MTFs) are special cases of the ACE formalism.[14] This suggests ACE is a method of choice, but its flexibility comes with more hyperparameter choices, which can make it more complex for the user to navigate.

Using a pre-trained MLP avoids training time, which is typically days to months, and is therefore worth pursuing (**Sec. 7.4**). U-MLPs can be a great starting point, but need to be carefully vetted, and at this stage are likely best used in cases where some post-calculation checking is built into the project workflow. Reusing targeted MLPs can be a great solution, but it is recommended to validate at least some aspects of the MLP since it is likely being used in some ways different from those in the original studies and assessments. Finally, if you are fitting your own MLPs, then deep learning potentials are generally needed for more than ~ 5 elements, although recent methodological developments are potentially removing this constraint (**Sec. 7.6.1**). However, deep learning MLPs can require more data and take more time to train (**Sec. 7.6.2**), and may require more human time for hyperparameter optimization (**Sec. 7.6.3**).

8. MLP infrastructure

In recent years, a plethora of software packages have emerged within the dynamic landscape of MLPs, catering to both standard-scale and large-scale simulations. These packages aim to streamline the process of training, fitting, and deploying MLPs for running MD for diverse applications in chemistry and materials science. For standard-scale simulations, ideal features for packages should include ease of use, adaptability, intuitive interfaces, and flexibility in handling various data types and model architectures. On the other hand, large-scale simulations demand efficient parallelization, scalability, robustness, and high-performance computing integrations. In the following section, we explore some of the most prominent and user-friendly packages in both categories, detailing their features, strengths, and ideal use cases.

In the effort to promote adoption and advance the accessibility of MLPs, many tools and platforms have emerged. As mentioned in **Sec. 7.4**, a notable example is ColabFit Exchange, which functions as an informatics platform tailored for advanced materials and chemistry applications.[132] ColabFit Exchange contains curated, high-quality data from publications useful for fitting MLPs. As of January 2025, there are nearly 400 datasets comprising more than 230 million unique atomic arrangements. Recent work from Andolina and Saidi generated curated training datasets of 23 single-element systems and built MLPs with DeepMD, where all of the training data are hosted on ColabFit Exchange.[133,134] Furthermore, packages like the Knowledgebase of Interatomic Models-based Learning-Integrated Fitting Framework (KLIFF) have been developed for general-purpose fitting of MLPs, offering the versatility to deploy these models within simulation software like LAMMPS via OpenKIM, as well as automated model verification, testing (i.e., the automated computation of a wide range of physical properties for all archived potentials), and archiving on https://openkim.org.[91] KLIFF also incorporates uncertainty quantification, a powerful feature for assessing the reliability and confidence associated with MLP predictions. These tools exemplify some of the concerted efforts made by the community to surmount adoption barriers and propel the field of MLPs forward. Another emerging platform in the ecosystem is Garden.[135,136] Garden is designed to make ML models more accessible and deployable across different computing environments. Models are collected into domain-specific "gardens", as a collection of containerized models linked with structured data via the Materials Data Facility[137,138] or Foundry,[139] benchmarks, tests, and computing resources. Garden addresses key infrastructure challenges by containerizing models for consistent execution, facilitating model discovery and simplified deployment across local machines, cloud resources, and HPC clusters through Globus Compute integration. Finally, as discussed in **Sec. 7.4**, at present there are at least two notable examples of interatomic potential repositories, OpenKIM and the NIST Interatomic Potentials Repository, that include many PBPs but also a growing number of MLPs.

Below, we discuss a standard set of procedures and software for generating or acquiring training data, fitting, comparison, and deployment of MLPs in MD simulations. For MD simulations, LAMMPS has been a standard in the past decades in the field of materials science with a comprehensive documentation and most widely used MLPs have libraries in LAMMPS. The earlier MLP formulations such as Behler-Parrinello NNs, GAP, SNAP, and ACE have well-tested libraries (ML-HDPNN, ML-QUIP, ML-SNAP, and ML-PACE) that have become part of the LAMMPS code and is easier for users to install and use them. MTP also has a LAMMPS library, but it needs to be separately acquired from its Gitlab repository and added to LAMMPS. Recent MLPs such as DeepMD, MACE and Allegro also have LAMMPS libraries, but currently their libraries need to be downloaded from their GitHub repositories and added to LAMMPS. More streamlined integration of state-of-the-art MLPs with molecular simulation codes is ongoing. For example, the newest MLPs (e.g., NequIP, MACE) now provide native integration with JAX-MD, which is a Python library to run end-to-end differentiable MD simulations on GPUs.[140] In addition to LAMMPS and JAX-MD, another Python library frequently used for MD simulations is the Atomic Simulation Environment (ASE).[141] ASE provides numerous functionalities such as MD simulations or static energy/force calculation for each atomic configuration, that can be used for testing and comparing MLPs. Many aforementioned MLPs have specific libraries to use with ASE which are called calculators. U-MLPs such as M3GNet,[41] CHGNet,[20] MACE-MP0,[32] and EquiformerV2-OMAT24 [24] integrate seamlessly with ASE code, enabling a new user to load in a pretrained U-MLP and perform atomic relaxations or MD runs with only a few lines of python code.

As discussed in **Sec. 6** and **Sec. 7.1**, many MLPs require GPUs for efficient operation. Access to modest numbers of GPUs (e.g., 1–10) is becoming widespread in computational labs but can still be challenging to access when many are needed for large-scale studies. The Department of Energy (e.g., Summit) and National Science Foundation (e.g., ACCESS, National Artificial Intelligence Research Resource (NAIRR)) all have machines with large numbers of GPUs to which researchers can apply for resources. In addition, cloud computing resources, e.g., from Amazon Web Services (AWS) and Microsoft Azure can be leveraged to carry out large simulations with modest cost. This "pay-as-you-go" infrastructure provides users with instant access to state-of-the-art GPUs for large-scale applications. Of note is AWS, which recently launched its EC2 UltraCluster, which contains more than 4000 NVIDIA A100 GPUs. If more modest GPU computing is sufficient, users may consider using the free or paid tiers of Google Colab. The Garden framework further simplifies access to these diverse computing resources by providing standardized methods for deploying MLPs across different platforms. Through its integration with Globus Compute, Garden allows researchers to seamlessly utilize various computing resources, from local machines to DOE facilities and cloud providers.

9. Limits of standard MLPs and advanced MLPs to overcome those limits

MLPs have significantly enhanced our ability to describe PESs in various material systems. When dealing with complexities such as long-range forces, magnetism, and electronic excitation states, it is generally the case that modifications to standard MLPs are needed. However, adding more physics is more difficult than simply including more data for training MLPs. In this section, we provide an overview of the limits of MLP application within the realm of complex materials and the recent advancements to overcome these constraints.

9.1. Long-range interactions

Long-range interactions are not included in standard MLPs as they typically focus on learning local atomic descriptors for environments encompassing a radius of just 5–10 Å, becoming much slower for longer ranges. A graphical depiction of long-range interactions researchers hope to integrate into future MLPs is given in Fig. 8. It is possible that the MLPs which consider contributions only from short-range interactions may be deficient for accurately predicting some properties.[142] In cases where the importance of nonlocal physics and chemistry is fundamental in explaining properties, it becomes imperative to focus on nonlocal electrostatic and dispersion interactions, which are usually not represented by local descriptors. To overcome this challenge, several methodologies and models are employed to enhance the performance of MLPs for handling long-range interactions.

The first strategy is implicitly incorporating long-range interactions into short-range interactions, which is particularly useful for homogeneous condensed-phase systems with strong screening effects. This essentially comes down to trying to include the correct physics in the training data and hoping the long-range effects are largely screened or reasonably quantitatively renormalized into the short-range MLP. One approach is increasing the cutoff radius in standard MLPs to accommodate long-range interactions. For instance, AP-NET utilizes 8 Å cutoff atom-pair symmetry functions for evaluating monomer–monomer interaction energies.[143] A concrete example of renormalizing a naturally long-range interaction is including dispersion in DFT calculations for training data for standard short-range MLPs. It is interesting to note that for molten salts, which are ionic systems with large electrostatic interactions and significant dispersion contributions, there are many successful MLP models, demonstrating how effective this simple approach can be.[126,144–146] The ability to represent long-range electrostatics with short-range interactions can be understood as a result of screening, where local charge neutrality makes longer range interactions zero on average. The nature of this screening has been explained and studied

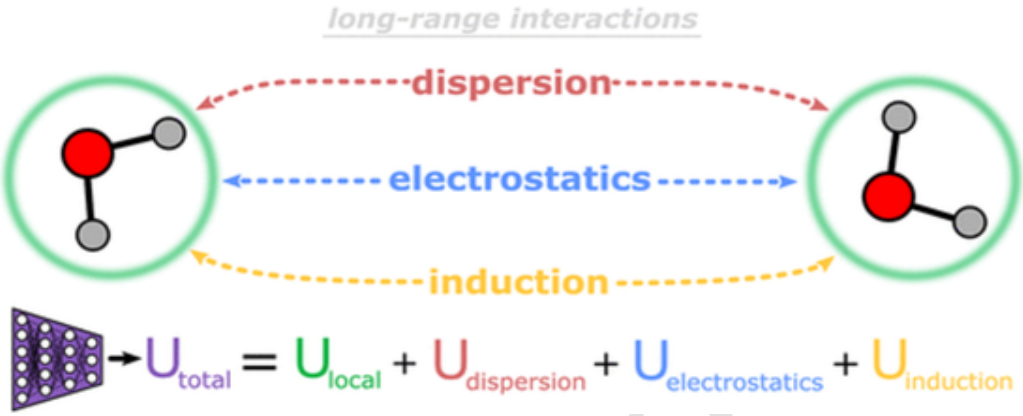


Fig. 8. Summary of the general energetic contributions composing the total potential energy (U_{total}) of a system. U_{local} refers to the short-range system energies and is typically inferred using a machine learning model trained on local features. Dispersion corrections, electrostatics, and induction are collectively referred to as the long-range interaction energy contributions. Adapted with permission from Ref. [142].

quantitatively by Ceder et al.[147] Their work points out that local charge neutrality is strongly correlated with lower-energy states, and that higher-energy states, where local charge neutrality is less robust, have electrostatic interaction that are not well-represented with short-range interactions. Thus, the success of short-range potentials for ionic systems may be in a large part due to the typical states explored in training and application data, which are often lower-energy states associated with near-equilibrium molecular dynamics simulations. These observations imply that for simulations with higher energy states, or more precisely, states without strong local charge neutrality, researchers should be very careful about using only short-range interaction and more complete treatment of long-range electrostatics may be necessary.

The second strategy is including explicit long-range interactions, such as electrostatics, using physics-based functional forms like Coulomb's law, with or without a dependency on the local atomic environment. For instance, the deep neural network potential called DeepPot utilizes a model based on (averages of the positions of) maximally localized Wannier centers to accurately calculate electrostatics.[148] A more refined version of DeepPot is the self-consistent field neural network (SCFNN), where SCFNN combines an iterative refinement approach with maximally localized Wannier centers to enhance the accuracy of electrostatics calculations, demonstrated by its ability to accurately predict the high-frequency dielectric constant of water.[149] The recently updated AIMNet2 (also mentioned in **Sec. 5**) directly includes long-range interactions into the MLP formalism, in which the DFT-D3 vdW and electrostatic corrections are explicitly included as energy terms, allowing an expanded application to neutral and charged states, as well as diverse organic compounds composed of many different chemical elements.[18,142] Also, as mentioned in **Sec. 5**, the MACE-MP0 U-MLP was trained only on PBE-level DFT calculations (which only incorporate short-range interactions) and has the ability to add on the DFT-D3 vdW interactions, which are an empirical correction on top of the PBE-level model. This correction can be done easily using the torch-dftd dispersion model implemented in PyTorch.[79] Another example of dispersion corrected-MLP is SO3LR.[150] As another example, in the global gradient-domain machine learning force field, i.e., Symmetric Gradient Domain Machine Learning (*s*GDML) approach, the descriptor of a molecular system is treated as a unified entity, bypassing the need for arbitrary partitioning of energy into atomic contributions.[151] The learned model essentially includes all interaction scales. This unique approach enables the sGDML framework to effectively capture both chemical interactions and long-range forces. However, due to the requirement of all correlations of atom–atom interactions, the global MLPs are usually limited in scaling up to large molecules.

In summary, the presence of long-range interactions has posed some challenges for MLPs, and substantial efforts have been made to address this issue. Specifically, the first strategy of renormalizing long-range interactions from training data into short-range interactions in the MLP has been extensively employed in standard MLPs, requiring no additional knowledge or extra effort. For applications where long-range interactions have minimal impacts, users are encouraged to implement this straightforward approach for the first strategy. The second strategy of explicit long-range interactions is becoming more routine for state-of-the-art MLPs. For studies that require long-range interactions or where such interactions are of interest, particularly electrostatics (e.g., ions, electrolytes), users are encouraged to employ the MLPs mentioned in the second strategy discussed above. In addition, small molecular systems, usually consisting of (at most) a few hundred atoms, where significant long-range interactions are in play, are well-suited for using global representations of the features. Utilizing a global representation of the entire system typically leads to a reduction in computational complexity compared with previous methods, thereby enhancing both the training process and the efficiency of molecular simulations.

9.2. Modeling systems off the born-oppenheimer surface

MLPs are typically a mapping of atomic positions to energy and forces, and therefore assume this mapping is unique. The natural unique PES is that of the lowest energy electron configurations for each atomic arrangement, which is the Born-Oppenheimer surface. However, it is often of interest to consider some forms of excitations. If the excitations are fixed, e.g., we ionize the system, then this is just another uniquely defined Born-Oppenheimer surface determined by some constraint and presents no fundamental challenge. One can simply train a standard MLP on data from the constrained system Born-Oppenheimer surface. However, if the excitations can move between different Born-Oppenheimer surfaces, e.g., multiple magnetic states or varying electronic excitations, then a significant change in the MLP formalism is required. Here, we discuss two areas being widely studied, namely magnetism and electronic excitations, although other types of excitations might also be of interest.

9.2.1. Magnetism

Different magnetic states of ions possess significantly different properties, and this complexity becomes critical in the context of magnetic materials. How to differentiate ions with different spin states is difficult

and lacks a unique solution in the MLP community. Incorporating spin degrees of freedom into MLPs, which are crucial for accurately representing finite temperature phenomena in magnetic materials, has remained a challenging task. In spin density functional theory (SDFT), magnetization arises from the interplay between magnetic exchange and band energy contributions,[152,153] where the energy required for electron redistribution between up and down spin channels depends on the local density of states (DOS). Iron, for example, exhibits a bimodal DOS in its body-centered crystal structure (bcc), resulting in larger magnetic moments compared to the face-centered cubic (fcc) structure with a more unimodal DOS.[154] This intricate relationship between magnetic and atomic structure necessitates the consideration of multi-atom, multi-spin interactions to capture various magnetic and atomic arrangements within a single model. Unlike methods derived from electronic structure theory that seamlessly incorporate the complexity of magnetic interactions,[154] classical PBPs require additional terms to mimic quantum exchange interactions. One common approach involves using a classical Heisenberg Hamiltonian,[155] where atomic spin operators are replaced by spin vectors, and exchange interactions are parameterized using *ab initio* calculations.[156] Many MLP approaches for magnetic systems have adopted similar strategies. For instance, Nikolov et al.[157] expanded the SNAP framework with a two-spin bi-linear Heisenberg model. Yu et al.[158] developed a neural network-based approach to describe contributions to the Heisenberg Hamiltonian based on the local magnetic environment, although this method did not account for lattice information and treated magnetic moments as unit vectors. Eckhoff and Behler[159] extended the original Behler-Parrinello[19] symmetry functions framework but the formalism was limited to collinear configurations. Novikov et al.[160] incorporated magnetic moments as additional degrees of freedom in the MTP framework, albeit also restricted to collinear moments. Domina et al.[161] extended the SNAP framework to handle arbitrary vectorial fields, demonstrating its functionality with non-collinear spin configurations. Chapman and Ma introduced a neural network correction to an embedded atom method potential augmented with a Heisenberg-Landau Hamiltonian for large-scale spin–lattice dynamics simulations.[162] Finally, as discussed in **Sec. 5**, the CHGNet U-MLP developed by Deng et al.[20] goes beyond reporting energies and forces by also predicting the magnetic moment on every atom in the system, enabling differentiation of different valence states and analysis of the underlying magnetic properties. Despite these efforts, none of the existing ML approaches for magnetic systems have achieved a transferable and quantitatively accurate description of magnetic interactions suitable for modeling magnetism in different crystal structures.

The ACE method has been expanded to accommodate vectorial or tensorial characteristics, alongside the inclusion of atomic magnetic moments and charges in addition to atomic positions, as detailed by Drautz et al.[163] This extended ACE framework offers a complete foundation for characterizing the local atomic environment. Unlike being limited to representing energies solely as a function of atomic positions and chemical species, ACE can be adapted to encompass vectorial or tensorial properties and incorporate additional degrees of freedom. This adaptability is particularly significant for magnetic materials where potential energy surfaces depend on both atomic positions and atomic magnetic moments concurrently. Notably, recent work by Rinaldi et al. introduced a non-collinear magnetic ACE parameterization specifically tailored for the prototypical magnetic element, iron.[164] The model was trained using a diverse set of collinear and noncollinear magnetic structures, computed using SDFT. Their findings demonstrate that this non-collinear magnetic ACE method not only accurately reproduces the ground state properties of various magnetic phases of iron but also captures magnetic and lattice excitations crucial for an accurate description of finite-temperature behavior and crystal defect properties.[164].

Recently, Yu et al.[165] introduced the Time-reversal Equivariant Neural Network (TENN) framework, which incorporates time-reversal symmetry into the equivariant neural network (ENN). This extension allows ENN to account for physical aspects related to time-reversal symmetry, such as the spin and velocity of atoms. Specifically, they developed TENN-e3, an expansion of the E(3) equivariant neural network, to maintain the time reversal E(3) equivariance while considering the inclusion of the spin–orbit effect in situations involving both collinear and non-collinear magnetic moments in magnetic materials. TENN-e3 can construct a spin neural network potential and the Hamiltonian for magnetic materials based on *ab initio* calculations. TENN-e3 employs Time-reversal-E(3)-equivariant convolutions to model interactions between spinor and geometric tensors. TENN-e3 excels at accurately describing the complex spin–lattice coupling while preserving time-reversal symmetry, a feature not present in existing E(3)-equivariant models. Additionally, TENN-e3 facilitates the construction of the Hamiltonian for magnetic materials with time-reversal symmetry.

In summary, TENN offers a new approach for conducting spin–lattice dynamics simulations over extended time scales and performing electronic structure calculations on large-scale magnetic materials. As an instance of TENN-e3, Spin-Allegro can help generate the spin interatomic potential.[166] On the other hand, the ACE approach for iron can be directly extended to multicomponent systems, such as technologically important magnetic alloys and carbides. While conceptually straightforward, generating precise and comprehensive DFT reference data for magnetic multicomponent materials is challenging. People can use efficient sampling techniques based on D-optimality active learning to address this challenge (see **Sec. 7.5.4**), which is expanded to include magnetic degrees of freedom. It can help reduce the number of required DFT reference calculations. Although these novel methods have been proposed, the testing has only been on a small number of systems. Therefore, further exploration and testing of such MLPs on more magnetic systems are needed to assess the general efficacy.

9.2.2. *Excited states*

At present, a well-established MLP specifically for excited systems does not exist. Nevertheless, it is crucial to emphasize that ongoing research efforts aimed at developing and enhancing MLPs are progressing rapidly, and we discuss a few recent efforts here.

Electronically excited states are central to various fields such as photochemistry and photophysics. Like magnetism, they represent an additional degree of freedom that must be added to the potential. Most MLPs are attempting to learn the PES of molecular/condensed phase systems at the ground state. It requires careful consideration to design an MLP that can learn the secondary outputs, i.e., excited-state PES, corresponding forces, and nonadiabatic and spin–orbit couplings between them.[167,168] For instance, multiple PESs and their couplings should be considered when dealing with excited states. Furthermore, the complexity and high computational expense of generating the underlying training data calculations and the associated complexity of the corresponding ML models make it more challenging to train an MLP for excited states than for the ground state. Therefore, the application of ML models for excited states is significantly more challenging than for the ground state.

Recently, Marquetand and co-workers developed SchNarc, a framework for excited-state molecular dynamics simulations.[169] SchNarc combines the surface hopping including arbitrary couplings (SHARC) approach for photodynamics, which handles states of different multiplicities, with SchNet (a message-passing deep neural network), which efficiently and accurately fits potential energies and other molecular properties. This framework overcomes current limitations of existing MLP-based MD simulations for excited states by allowing (i) phase-free training, eliminating the costly preprocessing of raw quantum chemistry data, (ii) treatment of rotationally covariant non-adiabatic couplings (NACs), which can either be trained or (iii) approximated from

only ML potentials, their gradients, and Hessians, and (iv) handling of spin–orbit couplings. They extended the model using a NN with multiple outputs to fit all non-adiabatic vectors between different states of the same spin multiplicity simultaneously,[170] which increases the accuracy of the prediction of excited-state dynamics simulations.

More recently, Zhang and co-workers applied a symmetry-adapted high-dimensional neural network to treat couplings as derivatives of NN representations.[171] In this approach, electronic friction was modeled using machine learning and applied to MD simulations of molecules at metal surfaces, thereby treating electron-nuclei coupling in a rotationally covariant manner. For the non-adiabatic coupling vectors, a similar strategy akin to force-only training for potentials, by implementing them as derivatives of virtual properties—properties not explicitly defined in quantum chemistry—constructed by a deep NN. They extended their embedded atom neural network to a universal field-induced recursively embedded atom neural network (FIREANN) by introducing pseudo atomic field vectors relative to each atom with rigorous rotational equivariance. The FIREANN is capable of predicting multiple polarization values for various response properties, making it possible to accurately capture the excited-state PESs within a single model. [172].

10. The future of MLPs

Given the rapid development and evolution of the field of MLPs, discussing the future of MLPs is quite speculative. In particular, the extraordinary pace and disruptive nature of innovations in ML suggest that all predictions related to this area are highly uncertain. With that caveat, we share a few ideas of how the field of MLPs may progress in the near future.

In the near term (~3–5 years) we see numerous areas where trends that are already well-established are likely to continue. In terms of sampling, we expect to continue to see new methods of active learning and ways to determine optimal training structures to emerge, e.g., as done recently by Fonseca et al. who used ML to sample new areas of configurational space to more optimally improve both explicit AEF MLPs based on GAP and a deep learning MLP.[173] Additionally, in complex chemical applications such as bond breaking/formation, advances in active learning can guide the selection of relevant training data, as was illustrated by Kulichenko et al.[122].

At this point there are many different databases available that are used for fitting (e.g., the Materials Project and data shared through ColabFit[132,134]), typically developed by single groups in many different ways. However, there does not seem to be a leading established approach to developing databases of pre-existing *ab initio* calculations for fitting. This problem has many facets as it involves interacting with large existing databases, integrating data from multiple levels of accuracy, and providing guidance for fitting everything from very focused potentials (e.g., just C or Si) to large U-MLPs (e.g., with 90 + elements). We expect that a few underlying approaches and key databases will eventually become standard and widely adopted for the majority of use cases.

We also expect further refinements to standard MLP methods. At present, it seems that the pace of innovation has slowed compared to what was occurring over the last 10–15 years, and it appears that the explicit AEF approaches provided by methods like ACE and the deep learning equivariant GNNs are close to optimal within our present understanding. Therefore, within the present explicit AEF and deep learning MLP framework, efforts will shift to modest changes in the formalism, with a focus on allowing more rapid and turn-key fitting and evaluation of these methods, as well as scaling the fitting to larger datasets. There is a clear need to establish standard methodological approaches to some of the known limitations of present standard MLPs, which include incorporating long-range-forces and excited states (including magnetic states). Short- and long-range forces are relatively straightforward to treat with either targeted fitting and/or semi-empirical corrections. Recent work has also shown a path for magnetic states, which can be treated by some advanced methods, such as the non-collinear magnetic ACE method and the TENN model, and they are expected to be a standard part of MLP fitting packages within the next few years. More general excited state methods are being developed and will likely become widely accessible in the next 3–5 years. That all said, the tools of deep learning keep improving, driven by enormous commercial and national priority pressures, which will likely drive rapid improvements in deep learning training, execution, accuracy, interpretation, and implementation. To help readers appreciate the rate of change and improvement in this field, we note that the modern form of the MPGNN upon which so many deep learning MLPs are based is generally attributed to work published only in 2017.[174] It therefore seems likely that there will be disruptive innovations in ML that will suggest new and possibly much more powerful MLP approaches sometime within the next 3–5 years, which may alter the focus of the field significantly.

Also in the next 3–5 years, we expect large and crucially important improvements in MLP-related infrastructure and corresponding increases in the adoption of MLPs for molecular modeling across the chemistry, materials, physics, and biology communities. Many studies using MLPs are still related to benchmarking or basic property prediction, and reuse of MLPs for complex property modeling and materials discovery and design is still limited. However, as their utility becomes better known and the MLP infrastructure develops further, we can expect much more widespread use. In terms of MLP infrastructure, code packages for fitting (e.g., MTP, ACE, Allegro, etc.) and integration with major molecular development packages (e.g., ASE, pymatgen) and simulation tools (e.g., LAMMPS) are already widely available, but can still be made more comprehensive and easier to use. Furthermore, greater integration between MLP fitting codes is likely to provide many advantages. For example, we expect there to soon be code packages that can fit multiple potentials and provide assessment of which is best for your systems and problem. Similarly, such codes and pre-fit MLPs will be housed in easily accessible and searchable repositories with an automated assessment of MLP quality, as is being developed in OpenKIM. [175] Both fitting and assessment will greatly benefit from a large set of high-quality benchmark databases. Many benchmarks already exist but were often not developed with MLP development and benchmarking in mind (e.g., the Materials Project). Applying FAIR principles to MLPs will increase the useful infrastructure and enhance their adoption. The Garden framework represents an early example of this trend, providing a FAIR-oriented platform that simplifies model publishing, discovery, and deployment across various computing resources. Such frameworks will help democratize access to MLPs and ensure reproducibility across different computing environments. Finally, we note that direct integration with DFT packages is possible (e.g., as has happened in the VASP code[176]) but that does not seem to be the direction the field is moving, likely due to the ease of connecting DFT with the MLP fitting and the challenges of maintaining all the advantages of the flexible and evolving MLP ecosystem when integrated with a DFT package. Overall, navigating the multitude of available options for MLPs is likely to be a daunting task for at least a few years to come. To facilitate decision-making on the choice of MLPs for a given application, we expect to see the emergence of recommendation systems based on the user's intended applications and case-specific problems. In this regard, we believe it is important to establish a basis for informed decision-making, i.e., comparisons that aid in evaluating the suitability of different packages.

As an external factor affecting MLPs, the continually evolving supercomputing landscape can alter the relative strengths of MLPs based on their ability to adapt. Already, the dominance of GPU-based supercomputers (9 out of the top 10 in the world) renders those MLPs equipped with GPU-acceleration favorable for scientific applications that require large-scale simulations, as a CPU-locked MLP will require hundreds of CPU cores to match the performance of even a single GPU. Such differ-

ences will be exacerbated as the computing landscape becomes more diversified. Even today, of the four fastest supercomputers, one is CPU-based, while the other three use GPUs from different vendors, whose native programming models are not interoperable. For the typical user with access to one or a small handful of computing environments, the choice of MLP will strongly be influenced by the MLP's performance, or even ability to run, on the hardware available to the user. This favors MLPs that are built on a performance portability layer that makes them largely agnostic to the underlying hardware, such as SNAP and FLARE, which use Kokkos, and many of the deep learning based MLPs using PyTorch, such as MACE and Allegro. In the future, we may see more radical changes to hardware. Very recently, the Cerebras wafer-scale AI chip was used to run MD simulations more than two orders of magnitudes faster than CPUs and GPUs.[177] While the Cerebras-based simulations used an EAM potential, the results demonstrate the promise of new hardware to drastically change the capabilities of MD simulations, and the MLPs and their implementations that best adjust accordingly will have a great advantage over the competition.

In the short- to mid-term (5–10 years), a particularly interesting area will be the development of U-MLPs. U-MLPs are somewhat analogous to the foundational models that have been so impactful in the image generation and natural language processing (NLP) community. Foundational models generally refer to large models that can achieve good performance on a wide variety of tasks, which allows them to be adapted to many specific applications (i.e., they are a foundation for many other useful more specific models). For example, Large Language Models (LLMs) in the NLP community have seen an explosion of performance and utility over the last few years, and are being integrated into hundreds of different tools and products. The generality of U-MLPs across chemistry and structure will also allow them to impact many more problems than a typical PBP or targeted MLP has done in the past, which is why they are sometimes referred to as foundational models for materials and chemistry. At present, U-MLPs are mostly useful for qualitative or semi-quantitative screening across many systems, but they are rapidly becoming quantitative tools for detailed molecular modeling of specific material properties (e.g., Li diffusion in electrolytes). Future U-MLPs may function as foundational models, enabling simulation of longer time scales (e.g., $>$ 1 ms) and modeling of totally new materials phenomena currently inaccessible with today's MLPs. It is likely U-MLPs will continue to improve rapidly, increasingly taking over the applications presently dominated by targeted MLPs. This transformation will require a few improvements, but all seem to be well underway. First, larger, more diverse, high-fidelity training data is needed. However, improved hardware, both CPU and GPU, will contribute to increasing the output of *ab initio* data for fitting. Integration of multiple databases will allow for very large training sets and potentially multifidelity training sets[80,178,179] (e.g., with DFT and coupled cluster data) to support MLPs that approach chemical accuracy (1 kcal/mol) and overcome limitations of lower fidelity DFT data (e.g., like DFT-PBE calculations). We also expect infrastructure and methodological innovations to allow for more contributions from the enormous amounts of data in the broader community, e.g., through online fine tuning or federated learning approaches. It is reasonable to expect that training data sets approaching or exceeding a billion training data points will be within reach in the next few years (we are already seeing training on $\sim$ 110x10^6 DFT configurations). Along with this data, better algorithms and faster GPUs will support more rapid training and evaluation. A final piece that needs to be developed is likely some form of distillation (transferring knowledge from a larger to a smaller model) to allow fast models for specific applications to be easily developed from slower U-MLPs. This distillation could be as simple as fitting a simpler and faster MLP to U-MLP data, but more sophisticated direct methods might be developed. All of these innovations will require improved infrastructure to have their full impact realized. In particular, the scale of data and perhaps even model size of U-MLPs will require them to be trained and likely hosted by just a few leading organizations with large resources, including perhaps government (e.g., NIST), companies (e.g., Google, Matlantis), major research groups, and relevant societies (e.g., American Chemical Society (ACS)). Such hosting should allow easy use of the models, fine tuning, and distillation for use in high performance applications. These resources could supply full compute environments, just the codes, or some combination. Similar infrastructure is available for LLMs through tools like the OpenAI and HuggingFace APIs, and these tools play an enormous role in supporting the adoption of the LLMs. Frameworks like Garden[135,136] are beginning to lay the groundwork for this future by providing infrastructure that connects models with distributed computing resources and simplifies deployment across different environments – bridging the gap between model developers and users, much like how APIs from OpenAI and HuggingFace have done for LLMs.

As discussed in **Sec. 5**, the change from targeted MLPs ($\leq$ 5 elements) to U-MLPs (40–90 + elements) is a continuum. It is possible that semi-universal (SU-MLPs) (see **Sec. 5**) for key classes of materials with intermediate numbers of elements (e.g., $\sim$20) and/or limited phases or structures, might be established, e.g., for organic molecules, polymers, steel alloys, Al alloys, halide perovskites, electronic materials, molten salts, etc. Such an approach would mimic the very successful methods of the calculation of phase diagrams (CALPHAD) community, which typically develops databases in this manner. Such an approach obviously limits compositional complexity by treating fewer species, and limits the structural complexity by treating fewer phases and structures, but could also make fitting easier by treating relatively consistent physics (e.g., mostly ionic or covalent bonding). Therefore, SU-MLPs may provide a more practical solution for many materials design problems than full U-MLPs, or at least bridge the transition from models containing a few species under limited conditions to those seeking to represent the full periodic table under all conditions.

Overall, the above trends will likely lead to a significant reduction in *ab initio* molecular dynamics simulation time, although only after the method has been used to help train many potentials. This reduction may reduce overall compute and energy requirements for molecular modeling research, but we expect a large increase in MLP modeling, which may offset any gains and likely lead to an increase in the overall utilization of simulation.

More long-term ($>$10 years), it is possible the traditional potentials (e.g., Lennard-Jones, EAM, AMBER, etc.) will be almost fully replaced by MLPs, but this is not clear. For example, the AMBER potentials for many organic systems are close to chemical accuracy and very fast, making it unclear what advantages more complex MLPs would provide. However, it is possible that MLP approaches will be integrated into even the fastest and simplest potential approaches. For example, Yu et al.[180] recently described an approach to fit pair potentials with ML and then convert them to simple Buckingham form, achieving almost optimal pair potentials from ML with no loss of speed. It is also possible that MLPs will grow to become much more like full *ab initio* simulations, providing not just a mapping of positions to energies and forces but also to band structures, magnetic moments, charge densities, and even wavefunctions, replacing huge parts of what is presently done with quantum simulations.[181] On the other hand, a complementary vision is that ML integrates with *ab initio* at a more fundamental level, e.g., advancing exchange–correlation functionals and/or massively accelerating solutions of the Schrödinger equation (and relativistic extensions). This path might speed up *ab initio* methods to the level of MLPs, effectively achieving an MLP from a very different starting point.[182] Finally, there is perhaps no scientific or engineering field changing as fast as AI and ML right now, so all researchers need to be vigilant for new ideas that can bring entirely new frameworks and capabilities to the molecular modeling community.

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.

Acknowledgments

Funding for the "Machine Learning Potentials – Status and Future (MLP-SAFE)" workshop and development of this paper was provided by the National Science Foundation through an AI Institute Planning Grant, Award Number 2020243.

KC thanks the National Institute of Standards and Technology for funding, computational, and data management resources. This work was performed with funding from the CHIPS Metrology Program, part of CHIPS for America, National Institute of Standards and Technology, U.S. Department of Commerce. Certain commercial equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identifications are not intended to imply recommendation or endorsement by NIST, nor it is intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

Data availability

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

References

[1] 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, Chem. A Eur. J. 124 (2020) 731–745, https://doi.org/10.1021/acs.jpca.9b08723.

[2] J. Behler, Four generations of high-dimensional neural network potentials, Chem. Rev. 121 (2021) 10037–10072, https://doi.org/10.1021/acs.chemrev.0c00868.

[3] V.L. Deringer, A.P. Bartók, N. Bernstein, D.M. Wilkins, M. Ceriotti, G. Csányi, Gaussian process regression for materials and molecules, Chem. Rev. 121 (2021) 10073–10141, https://doi.org/10.1021/acs.chemrev.1c00022.

[4] Y. Mishin, Machine-learning interatomic potentials for materials science, Acta Mater. 214 (2021) 116980, https://doi.org/10.1016/j.actamat.2021.116980.

[5] 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 (2021) 10142–10186, https://doi.org/10.1021/acs.chemrev.0c01111.

[6] K. Wan, J. He, X. Shi, Construction of high accuracy machine learning interatomic potential for surface/interface of nanomaterials—a review, Adv. Mater. 36 (2024), https://doi.org/10.1002/adma.202305758.

[7] F. Noé, A. Tkatchenko, K.-R. Müller, C. Clementi, Machine learning for molecular simulation, Annu. Rev. Phys. Chem. 71 (2020) 361–390, https://doi.org/10.1146/annurev-physchem-042018-052331.

[8] V.L. Deringer, M.A. Caro, G. Csányi, Machine learning interatomic potentials as emerging tools for materials science, Adv. Mater. 31 (2019), https://doi.org/10.1002/adma.201902765.

[9] D. Tang, R. Ketkaew, S. Luber, Machine learning interatomic potentials for catalysis, Chem. – A Eur. J. (2024), https://doi.org/10.1002/chem.202401148.

[10] T.W. Ko, S.P. Ong, Recent advances and outstanding challenges for machine learning interatomic potentials, Nat. Comput. Sci. 3 (2023) 998–1000, https://doi.org/10.1038/s43588-023-00561-9.

[11] T.T. Duignan, The potential of neural network potentials, ACS Phys. Chem. Au 4 (2024) 232–241, https://doi.org/10.1021/acsphyschemau.4c00004.

[12] 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 (2017) 3192–3203, https://doi.org/10.1039/C6SC05720A.

[13] A. Musaelian, S. Batzner, A. Johansson, L. Sun, C.J. Owen, M. Kornbluth, B. Kozinsky, Learning local equivariant representations for large-scale atomistic dynamics, Nat. Commun. 14 (2023) 579, https://doi.org/10.1038/s41467-023-36329-y.

[14] R. Drautz, Atomic cluster expansion for accurate and transferable interatomic potentials, PhysRevB 99 (2019), https://doi.org/10.1103/PhysRevB.99.014104.

[15] N. Artrith, A. Urban, An implementation of artificial neural-network potentials for atomistic materials simulations: performance for TiO2, Comput. Mater. Sci 114 (2016) 135–150, https://doi.org/10.1016/j.commatsci.2015.11.047.

[16] N. Artrith, A. Urban, G. Ceder, Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species, PhysRevB 96 (2017) 014112, https://doi.org/10.1103/PhysRevB.96.014112.

[17] K. Choudhary, B. DeCost, L. Major, K. Butler, J. Thiyagalingam, F. Tavazza, Unified graph neural network force-field for the periodic table: solid state applications, Digital Discovery 2 (2023) 346–355, https://doi.org/10.1039/d2dd00096b.

[18] D.M. Anstine, R. Zubatyuk, O. Isayev, AIMNet2: A Neural Network Potential to Meet your Neutral, Charged, Organic, and Elemental-Organic Needs, ChemRxiv (2023). 10.26434/chemrxiv-2023-296ch.

[19] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, PhysRevLett 98 (2007) 1–4, https://doi.org/10.1103/PhysRevLett.98.146401.

[20] B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C.J. Bartel, G. Ceder, CHGNet: Pretrained universal neural network potential for charge-informed atomistic modeling, Nat. Mach. Intell. (2023), https://doi.org/10.1038/s42256-023-00716-3.

[21] 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, E. Weinan, R. Car, L. Zhang, H. Wang, DeePMD-kit v2: A software package for deep potential models, J. Chem. Phys. 159 (2023), https://doi.org/10.1063/5.0155600.

[22] H. Wang, L. Zhang, J. Han, W.E. DeePMD-kit, A deep learning package for many-body potential energy representation and molecular dynamics, Comput Phys Commun 228 (2018) 178–184, https://doi.org/10.1016/j.cpc.2018.03.016.

[23] A. Rodriguez, C. Lin, H. Yang, M. Al-Fahdi, C. Shen, K. Choudhary, Y. Zhao, J. Hu, B. Cao, H. Zhang, M. Hu, Million-scale data integrated deep neural network for phonon properties of heuslers spanning the periodic table, NPJ Comput. Mater. 9 (2023) 20, https://doi.org/10.1038/s41524-023-00974-0.

[24] L. Barroso-Luque, M. Shuaibi, X. Fu, B. Wood, M. Dzamba, M. Gao, A. Rizvi, Open Materials 2024 (OMat24) Inorganic Materials Dataset and Models, (n.d.).

[25] J. Vandermause, S.B. Torrisi, S. Batzner, Y. Xie, L. Sun, A.M. Kolpak, B. Kozinsky, On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events, NPJ Comput. Mater. 6 (2020) 20, https://doi.org/10.1038/s41524-020-0283-z.

[26] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons, PhysRevLett 104 (2010), https://doi.org/10.1103/PhysRevLett.104.136403.

[27] F. Xie, T. Lu, S. Meng, M. Liu, GPTFF: a high-accuracy out-of-the-box universal AI force field for arbitrary inorganic materials, Sci Bull (Beijing) (2024), https://doi.org/10.1016/j.scib.2024.08.039.

[28] A. Merchant, S. Batzner, S.S. Schoenholz, M. Aykol, G. Cheon, E.D. Cubuk, Scaling deep learning for materials discovery, Nature 624 (2023) 80–85, https://doi.org/10.1038/s41586-023-06735-9.

[29] A. Bochkarev, Y. Lysogorskiy, R. Drautz, Graph atomic cluster expansion for semilocal interactions beyond equivariant message passing, PhysRevX 14 (2024) 021036, https://doi.org/10.1103/PhysRevX.14.021036.

[30] H. Yang, C. Hu, Y. Zhou, X. Liu, Y. Shi, J. Li, G. Li, C. Zeni, M. Horton, R. Pinsler, MatterSim: A Deep Learning Atomistic Model Across Elements, Temperatures and Pressures, (n.d.).

[31] I. Batatia, D. Kovacs, G. Simm, C. Ortner, G. Csanyi, MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields, in: 36th Conference on Neural Information Processing Systems (NeurIPS 2022), 2022.

[32] I. Batatia, P. Benner, Y. Chiang, A.M. Elena, D.P. Kovács, J. Riebesell, X.R. Advincula, M. Asta, M. Avaylon, W.J. Baldwin, F. Berger, N. Bernstein, A. Bhowmik, S.M. Blau, V. Cărare, J.P. Darby, S. De, F. Della Pia, V.L. Deringer, R. Elijošius, Z. El-Machachi, F. Falcioni, E. Fako, A.C. Ferrari, A. Genreith-Schriever, J. George, R.E.A. Goodall, C.P. Grey, P. Grigorev, S. Han, W. Handley, H.H. Heenen, K. Hermansson, C. Holm, J. Jaafar, S. Hofmann, K.S. Jakob, H. Jung, V. Kapil, A.D. Kaplan, N. Karimitari, J.R. Kermode, N. Kroupa, J. Kullgren, M.C. Kuner, D. Kuryla, G. Liepuoniute, J.T. Margraf, I.-B. Magdău, A. Michaelides, J.H. Moore, A.A. Naik, S.P. Niblett, S.W. Norwood, N. O'Neill, C. Ortner, K.A. Persson, K. Reuter, A.S. Rosen, L.L. Schaaf, C. Schran, B.X. Shi, E. Sivonxay, T.K. Stenczel, V. Svahn, C. Sutton, T.D. Swinburne, J. Tilly, C. van der Oord, E. Varga-Umbrich, T. Vegge, M. Vondrák, Y. Wang, W.C. Witt, F. Zills, G. Csányi, A foundation model for atomistic materials chemistry, ArXiv (2023). http://arxiv.org/abs/2401.00096.

[33] D.P. Kovács, J.H. Moore, N.J. Browning, I. Batatia, J.T. Horton, V. Kapil, W.C. Witt, I.-B. Magdău, D.J. Cole, G. Csányi, MACE-OFF23: Transferable Machine Learning Force Fields for Organic Molecules, (2023). http://arxiv.org/abs/2312.15211.

[34] A.V. Shapeev, Moment tensor potentials: A class of systematically improvable interatomic potentials, Multiscale Model. Simul. 14 (2016), https://doi.org/10.1137/15M1054183.

[35] 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, Nat. Commun. 13 (2022), https://doi.org/10.1038/s41467-022-29939-5.

[36] M. Neumann, J. Gin, B. Rhodes, S. Bennett, Z. Li, H. Choubisa, A. Hussey, Orb: A Fast, Scalable Neural Network Potential, (n.d.).

[37] S. Takamoto, D. Okanohara, Q.-J. Li, J. Li, Towards universal neural network interatomic potential, J. Materiomics 9 (2023) 447–454, https://doi.org/10.1016/j.jmat.2022.12.007.

[38] Y. Park, J. Kim, S. Hwang, S. Han, Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations, J. Chem. Theory Comput. 20 (2024) 4857–4868, https://doi.org/10.1021/acs.jctc.4c00190.

[39] K.T. Schütt, F. Arbabzadah, S. Chmiela, K.R. Müller, A. Tkatchenko, Quantum-

chemical insights from deep tensor neural networks, Nat. Commun. 8 (2017) 13890, https://doi.org/10.1038/ncomms13890.
[40] A.P. Thompson, L.P. Swiler, C.R. Trott, S.M. Foiles, G.J. Tucker, Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials, J. Comput. Phys. 285 (2015), https://doi.org/10.1016/j.jcp.2014.12.018.
[41] C. Chen, S.P. Ong, A universal graph deep learning interatomic potential for the periodic table, Nat. Comput. Sci. 2 (2022) 718–728, https://doi.org/10.1038/s43588-022-00349-3.
[42] S.R. Xie, M. Rupp, R.G. Hennig, Ultra-fast interpretable machine-learning potentials, npj Comput. Mater. 9 (2023) 162, https://doi.org/10.1038/s41524-023-01092-7.
[43] 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 Commun 271 (2022) 108171. 10.1016/j.cpc.2021.108171.
[44] M.S. Daw, M.I. Baskes, Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals, PhysRevB 29 (1984) 6443–6453, https://doi.org/10.1103/PhysRevB.29.6443.
[45] M.S. Daw, S.M. Foiles, M.I. Baskes, The embedded-atom method: a review of theory and applications, Mater. Sci. Rep. 9 (1993) 251–310, https://doi.org/10.1016/0920-2307(93)90001-U.
[46] 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 (2023) 5118. 10.21105/joss.05118.
[47] Y. Mishin, Machine-learning interatomic potentials for materials science, Acta Mater 214 (2021) 116980. 10.1016/j.actamat.2021.116980.
[48] M.S. Daw, S.M. Foiles, M.I. Baskes, The embedded-atom method: a review of theory and applications, Mater. Sci. Rep. 9 (1993), https://doi.org/10.1016/0920-2307(93)90001-U.
[49] I.-B. Magdău, D.J. Arismendi-Arrieta, H.E. Smith, C.P. Grey, K. Hermansson, G. Csányi, Machine learning force fields for molecular liquids: ethylene carbonate/ethyl methyl carbonate binary solvent, NPJ Comput. Mater. 9 (2023) 146, https://doi.org/10.1038/s41524-023-01100-w.
[50] J.D. Morrow, J.L.A. Gardner, V.L. Deringer, How to validate machine-learned interatomic potentials, J. Chem. Phys. 158 (2023), https://doi.org/10.1063/5.0139611.
[51] N. Artrith, J. Behler, High-dimensional neural network potentials for metal surfaces: A prototype study for copper, PhysRevB 85 (2012) 045439, https://doi.org/10.1103/PhysRevB.85.045439.
[52] A.M. Cooper, J. Kästner, A. Urban, N. Artrith, Efficient training of ANN potentials by including atomic forces via Taylor expansion and application to water and a transition-metal oxide, NPJ Comput. Mater. 6 (2020) 54, https://doi.org/10.1038/s41524-020-0323-8.
[53] A. Rohskopf, J. Goff, D. Sema, K. Gordiz, N.C. Nguyen, A. Henry, A.P. Thompson, M.A. Wood, Exploring model complexity in machine learned potentials for simulated properties, J. Mater. Res. 38 (2023) 5136–5150, https://doi.org/10.1557/s43578-023-01152-0.
[54] J.P. Darby, J.R. Kermode, G. Csányi, Compressing local atomic neighbourhood descriptors, NPJ Comput. Mater. 8 (2022), https://doi.org/10.1038/s41524-022-00847-y.
[55] N. Lopanitsyna, G. Fraux, M.A. Springer, S. De, M. Ceriotti, Modeling high-entropy transition metal alloys with alchemical compression, PhysRevMater 7 (2023) 45802, https://doi.org/10.1103/PhysRevMaterials.7.045802.
[56] J.P. Darby, D.P. Kovács, I. Batatia, M.A. Caro, G.L.W. Hart, C. Ortner, G. Csányi, Tensor-reduced atomic density representations, PhysRevLett 131 (2023) 028001, https://doi.org/10.1103/PhysRevLett.131.028001.
[57] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, P. Marquetand, wACSF—Weighted atom-centered symmetry functions as descriptors in machine learning potentials, J. Chem. Phys. 148 (2018), https://doi.org/10.1063/1.5019667.
[58] S. Takamoto, S. Izumi, J. Li, TeaNet: universal neural network interatomic potential inspired by iterative electronic relaxations, Comput. Mater. Sci 207 (2022), https://doi.org/10.1016/j.commatsci.2022.111280.
[59] I. Batatia, S. Batzner, D.P. Kovács, A. Musaelian, G.N.C. Simm, R. Drautz, C. Ortner, B. Kozinsky, G. Csányi, The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials, (2022). http://arxiv.org/abs/2205.06643.
[60] Y. Lysogorskiy, C. van der Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, 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 (2021) 1–12, https://doi.org/10.1038/s41524-021-00559-9.
[61] P. Weiner, P. Kollman, AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions, J Comput Chem 2 (1981).
[62] A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard III, W.M. Skiff, UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations, 1992. https://pubs.acs.org/sharingguidelines.
[63] D.A. Case, T.E. Cheatham, T. Darden, H. Gohlke, R. Luo, K.M. Merz, A. Onufriev, C. Simmerling, B. Wang, R.J. Woods, The Amber biomolecular simulation programs, J. Comput. Chem. 26 (2005) 1668–1688, https://doi.org/10.1002/jcc.20290.
[64] J.S. Smith, O. Isayev, A.E. Roitberg, ANI-1, A data set of 20 million calculated off-equilibrium conformations for organic molecules, Sci. Data 4 (2017) 170193, https://doi.org/10.1038/sdata.2017.193.
[65] C. Devereux, J.S. Smith, K.K. Huddleston, K. Barros, R. Zubatyuk, O. Isayev, A.E. Roitberg, Extending the applicability of the ANI deep learning molecular potential to sulfur and halogens, J. Chem. Theory Comput. 16 (2020) 4192–4202, https://doi.org/10.1021/acs.jctc.0c00121.
[66] S. Zhang, M.Z. Makoś, R.B. Jadrich, E. Kraka, K. Barros, B.T. Nebgen, S. Tretiak, O. Isayev, N. Lubbers, R.A. Messerly, J.S. Smith, Exploring the frontiers of condensed-phase chemistry with a general reactive machine learning potential, Nat. Chem. (2024), https://doi.org/10.1038/s41557-023-01427-3.
[67] https://matlantis.com/, (2023).
[68] Y.-L. Liao, B. Wood, A. Das, T. Smidt, EquiformerV2: improved equivariant transformer for scaling to higher-degree representations, ArXiv (2023).
[69] C. Chen, W. Ye, Y. Zuo, C. Zheng, S.P. Ong, Graph networks as a universal machine learning framework for molecules and crystals, Chem. Mater. 31 (2019) 3564–3572, https://doi.org/10.1021/acs.chemmater.9b01294.
[70] K. Choudhary, B. DeCost, Atomistic line graph neural network for improved materials property predictions, npj Comput. Mater. 7 (2021) 185, https://doi.org/10.1038/s41524-021-00650-1.
[71] Y. Zhou, S. Hu, C. Wang, L.-W. Wang, G. Tan, W. Jia, FastCHGNet: Training one Universal Interatomic Potential to 1.5 Hours with 32 GPUs, ArXiv (2024). 10.48550/arXiv.2412.20796.
[72] H. Yu, M. Giantomassi, G. Materzanini, J. Wang, G.-M. Rignanese, Systematic assessment of various universal machine-learning interatomic potentials, ArXiv (2024). http://arxiv.org/abs/2403.05729.
[73] B. Focassio, L.P.M. Freitas, G.R. Schleder, Performance Assessment of Universal Machine Learning Interatomic Potentials: Challenges and Directions for Materials' Surfaces, ArXiv (2024). http://arxiv.org/abs/2403.04217.
[74] B. Deng, Y. Choi, P. Zhong, J. Riebesell, S. Anand, Z. Li, K. Jun, K. Persson, G. Ceder, Overcoming systematic softening in universal machine learning interatomic potentials by fine-tuning, ArXiv (2024).
[75] J. Riebesell, R.E.A. Goodall, P. Benner, Y. Chiang, B. Deng, A.A. Lee, A. Jain, K.A. Persson, Matbench Discovery -- A framework to evaluate machine learning crystal stability predictions, (2023). http://arxiv.org/abs/2308.14920.
[76] H.-C. Wang, S. Botti, M.A.L. Marques, Predicting stable crystalline compounds using chemical similarity, NPJ Comput. Mater. 7 (2021) 12, https://doi.org/10.1038/s41524-020-00481-6.
[77] L. Casillas-Trujillo, A.S. Parackal, R. Armiento, B. Alling, Evaluating and improving the predictive accuracy of mixing enthalpies and volumes in disordered alloys from universal pretrained machine learning potentials, PhysRevMater 8 (2024) 113803, https://doi.org/10.1103/PhysRevMaterials.8.113803.
[78] D. Wines, K. Choudhary, CHIPS-FF: Evaluating Universal Machine Learning Force Fields for Material Properties, ArXiv (2024). 10.48550/arXiv.2412.10516.
[79] S. Takamoto, C. Shinagawa, D. Motoki, K. Nakago, W. Li, I. Kurata, T. Watanabe, Y. Yayama, H. Iriguchi, Y. Asano, T. Onodera, T. Ishii, T. Kudo, H. Ono, R. Sawada, R. Ishitani, M. Ong, T. Yamaguchi, T. Kataoka, A. Hayashi, N. Charoenphakdee, T. Ibuka, Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements, Nat. Commun. 13 (2022), https://doi.org/10.1038/s41467-022-30687-9.
[80] N. Shoghi, A. Kolluru, J. Kitchin, Z. Ulissi, L.C. Zitnick, B.M. Wood, From Molecules to Materials: Pre-training Large Generalizable Models for Atomic Property Prediction, ArXiv (2024). 10.48550/arXiv.2310.16802.
[81] T. Xie, X. Fu, O.-E. Ganea, R. Barzilay, T. Jaakkola, Crystal Diffusion Variational Autoencoder for Periodic Material Generation, ArXiv:2110.06197 (2021). http://arxiv.org/abs/2110.06197.
[82] D. Wines, T. Xie, K. Choudhary, Inverse Design of Next-generation Superconductors Using Data-driven Deep Generative Models, ArXiv:2304.08446 (2023). 10.1021/acs.jpclett.3c01260.
[83] C. Zeni, R. Pinsler, D. Zugner, A. Fowler, M. Horton, X. Fu, S. Shysheya, J. Crabbe, L. Sun, J. Smith, B. Nguyen, H. Schulz, S. Lewis, C.-W. Huang, Z. Lu, Y. Zhou, H. Yang, H. Hao, J. Li, R. Tomioka, T. Xie, MatterGen: a generative model for inorganic materials design, ArXiv:2312.03687v2 (2023).
[84] T.M. Nguyen, S.A. Tawfik, T. Tran, S. Gupta, S. Rana, S. Venkatesh, Efficient Symmetry-Aware Materials Generation via Hierarchical Generative Flow Networks, ArXiv (2024). 10.48550/arXiv.2411.04323.
[85] S. Yang, K. Cho, A. Merchant, P. Abbeel, D. Schuurmans, I. Mordatch, E.D. Cubuk, Scalable Diffusion for Materials Generation, ArXiv (2024). 10.48550/arXiv.2311.09235.
[86] N. Gruver, A. Sriram, A. Madotto, A.G. Wilson, C.L. Zitnick, Z. Ulissi, Fine-Tuned Language Models Generate Stable Inorganic Materials as Text, (2024). http://arxiv.org/abs/2402.04379.
[87] K. Choudhary, AtomGPT: atomistic generative pretrained transformer for forward and inverse materials design, J. Phys. Chem. Lett. 15 (2024) 6909–6917, https://doi.org/10.1021/acs.jpclett.4c01126.
[88] N. Bernstein, From GAP to ACE to MACE, ArXiv (2024).
[89] LAMMPS Benchmarks, (n.d.). https://www.lammps.org/bench.html#oneproc (accessed October 30, 2023).
[90] D. Lu, H. Wang, M. Chen, L. Lin, R. Car, W. E, W. Jia, L. Zhang, 86 PFLOPS Deep Potential Molecular Dynamics simulation of 100 million atoms with ab initio accuracy, Comput Phys Commun 259 (2021). 10.1016/j.cpc.2020.107624.
[91] M. Wen, Y. Afshar, R.S. Elliott, E.B. Tadmor, KLIFF: A framework to develop physics-based and machine learning interatomic potentials, Comput. Phys. Commun. 272 (2022) 108218, https://doi.org/10.1016/j.cpc.2021.108218.
[92] K. Nguyen-Cong, J.T. Willman, S.G. Moore, A.B. Belonoshko, R. Gayatri, E. Weinberg, M.A. Wood, A.P. Thompson, I.I. Oleynik, Billion atom molecular dynamics simulations of carbon at extreme conditions and experimental time and

length scales, in: International Conference for High Performance Computing, Networking, Storage and Analysis, SC, 2021. 10.1145/3458817.3487400.
[93] A. Musaelian, A. Johansson, S. Batzner, B. Kozinsky, Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size, ArXiv (2023).
[94] A. Johannson, Y. Xie, C. Owen, J. Lim, L. Sun, J. Vandermause, B. Kozinsky, Micron-scale heterogeneous catalysis with Bayesian force fields from first principles and active learning, ArXiv (2022).
[95] X. Zhang, S. Sundram, T. Oppelstrup, S.I.L. Kokkila-Schumacher, T.S. Carpenter, H.I. Ingólfsson, F.H. Streitz, F.C. Lightstone, J.N. Glosli, DdcMD: A fully GPU-accelerated molecular dynamics program for the Martini force field, J. Chem. Phys. 153 (2020), https://doi.org/10.1063/5.0014500.
[96] MN-Core, (n.d.).
[97] J.T. Frank, O.T. Unke, K.-R. Müller, S. Chmiela, A Euclidean transformer for fast and stable machine learned force fields, Nat. Commun. 15 (2024) 6539, https://doi.org/10.1038/s41467-024-50620-6.
[98] The ColabFit Exchange: Data for Advanced Materials Science, (2024). https://materials.colabfit.org/ (accessed July 4, 2024).
[99] NIST Interatomic Potentials Respository, (2024). https://www.ctcms.nist.gov/potentials/ (accessed July 4, 2024).
[100] Open Knowledgebase of Interatomic Models (OpenKIM), (2009). https://openkim.org/ (accessed May 7, 2024).
[101] M3Gnet Github repository, (2023). https://github.com/materialsvirtuallab/m3gnet (accessed May 6, 2024).
[102] CHGNet Github repository, (2024). https://github.com/CederGroupHub/chgnet (accessed May 6, 2024).
[103] MACE Github repository, (2024). https://github.com/ACEsuit/mace (accessed May 6, 2024).
[104] J.D. Morrow, J.L.A. Gardner, V.L. Deringer, How to validate machine-learned interatomic potentials, J Chem Phys 158 (2023). 10.1063/5.0139611.
[105] R.S.J.-N.W.C.L.C.W.R.V.A.C.Y.X.Y.M.Y.M.J.P.Y.S. Xiaoliang Pan, The Training of Machine Learning Potentials for Reactive Systems: A Colab Tutorial on Basic Models, ChemRxiv (2023).
[106] A.M. Tokita, J. Behler, How to train a neural network potential, J. Chem. Phys. 159 (2023), https://doi.org/10.1063/5.0160326.
[107] S. Attarian, C. Shen, D. Morgan, I. Szlufarska, Best practices for fitting machine learning interatomic potentials for molten salts: A case study using NaCl-MgCl2, Comput. Mater. Sci 246 (2025) 113409, https://doi.org/10.1016/j.commatsci.2024.113409.
[108] M. Geiger, T.E. Smidt, e3nn: Euclidean Neural Networks, ArXiv abs/2207.09453 (2022). https://api.semanticscholar.org/CorpusID:250698772.
[109] C. Owen, S. Torrisi, Y. Xie, S. Batzner, K. Bystrom, J. Coulter, A. Musaelian, L. Sun, B. Kozinsky, Complexity of many-body interactions in transition metals via machine-learned force fields from the TM23 Data Set, ArXiv (2023).
[110] R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, R. Asahi, On-the-fly active learning of interatomic potentials for large-scale atomistic simulations, J. Phys. Chem. Lett. 11 (2020), https://doi.org/10.1021/acs.jpclett.0c01061.
[111] Xiang Fu, Zhanghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, Tommi Jaakkola, Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations, ArXiv (2022). 10.48550/arXiv.2210.07237.
[112] Y. Zhai, A. Caruso, S.L. Bore, Z. Luo, F. Paesani, A "short blanket" dilemma for a state-of-the-art neural network potential for water: Reproducing experimental properties or the physics of the underlying many-body interactions?, J Chem Phys 158 (2023). 10.1063/5.0142843.
[113] Y. Liu, X. He, Y. Mo, Discrepancies and error evaluation metrics for machine learning interatomic potentials, NPJ Comput. Mater. 9 (2023) 1–13. https://api.semanticscholar.org/CorpusID:259202802.
[114] I.B. Magdău, D.J. Arismendi-Arrieta, H.E. Smith, C.P. Grey, K. Hermansson, G. Csányi, Machine learning force fields for molecular liquids: ethylene carbonate/ethyl methyl carbonate binary solvent, npj Comput. Mater. 9 (2023), https://doi.org/10.1038/s41524-023-01100-w.
[115] Y. Lysogorskiy, A. Bochkarev, M. Mrovec, R. Drautz, Active learning strategies for atomic cluster expansion models, PhysRevMater 7 (2022) 43801, https://doi.org/10.1103/PhysRevMaterials.7.043801.
[116] E.V. Podryabinkin, A.V. Shapeev, Active learning of linearly parametrized interatomic potentials, Comput. Mater. Sci 140 (2017) 171–180, https://doi.org/10.1016/j.commatsci.2017.08.031.
[117] N. Artrith, J. Behler, High-dimensional neural network potentials for metal surfaces: a prototype study for copper, Phys. Rev. B: Condens. Matter Mater. Phys. 85 (2012), https://doi.org/10.1103/PhysRevB.85.045439.
[118] J.S. Smith, B. Nebgen, N. Lubbers, O. Isayev, A.E. Roitberg, Less is more: sampling chemical space with active learning, J. Chem. Phys. 148 (2018), https://doi.org/10.1063/1.5023802.
[119] R. Jinnouchi, F. Karsai, G. Kresse, On-the-fly machine learning force field generation: application to melting points, PhysRevB 100 (2019), https://doi.org/10.1103/PhysRevB.100.014105.
[120] R. Jinnouchi, J. Lahnsteiner, F. Karsai, G. Kresse, M. Bokdam, Phase transitions of hybrid perovskites simulated by machine-learning force fields trained on the fly with bayesian inference, PhysRevLett 122 (2019), https://doi.org/10.1103/PhysRevLett.122.225701.
[121] J. Vandermause, Y. Xie, J.S. Lim, C.J. Owen, B. Kozinsky, Active learning of reactive Bayesian force fields applied to heterogeneous catalysis dynamics of H/Pt, Nat. Commun. 13 (2022), https://doi.org/10.1038/s41467-022-32294-0.
[122] 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, Nat. Comput. Sci. 3 (2023) 230–239, https://doi.org/10.1038/s43588-023-00406-5.
[123] C. van der Oord, M. Sachs, D.P. Kovács, C. Ortner, G. Csányi, Hyperactive learning for data-driven interatomic potentials, NPJ Comput. Mater. 9 (2023) 168, https://doi.org/10.1038/s41524-023-01104-6.
[124] L. Himanen, M.O.J. Jäger, E.V. Morooka, F. Federici Canova, Y.S. Ranawat, D.Z. Gao, P. Rinke, A.S. Foster, DScribe: library of descriptors for machine learning in materials science, Comput Phys Commun 247 (2020) 106949, https://doi.org/10.1016/j.cpc.2019.106949.
[125] L. Ward, A. Dunn, A. Faghaninia, N.E.R. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, K. Chard, M. Asta, K.A. Persson, G. Je, I. Foster, A. Jain, Matminer : an open source toolkit for materials data mining, Comput. Mater. Sci 152 (2018) 60–69, https://doi.org/10.1016/j.commatsci.2018.05.018.
[126] S. Attarian, D. Morgan, I. Szlufarska, Thermophysical properties of FLiBe using moment tensor potentials, J. Mol. Liq. 368 (2022) 120803, https://doi.org/10.1016/j.molliq.2022.120803.
[127] C. Chen, Z. Deng, R. Tran, H. Tang, I.H. Chu, S.P. Ong, Accurate force field for molybdenum by machine learning large materials data, PhysRevMater 1 (2017), https://doi.org/10.1103/PhysRevMaterials.1.043603.
[128] M. Yang, L. Bonati, D. Polino, M. Parrinello, Using metadynamics to build neural network potentials for reactive events: the case of urea decomposition in water, Catal. Today 387 (2022) 143–149, https://doi.org/10.1016/j.cattod.2021.03.018.
[129] A. Rodriguez, S. Lam, M. Hu, Thermodynamic and transport properties of LiF and FLiBe molten salts with deep learning potentials, ACS Appl. Mater. Interfaces 13 (2021) 55367–55379, https://doi.org/10.1021/acsami.1c17942.
[130] J. Zeng, L. Cao, M. Xu, T. Zhu, J.Z.H. Zhang, Complex reaction processes in combustion unraveled by neural network-based molecular dynamics simulation, Nat. Commun. 11 (2020) 1–9, https://doi.org/10.1038/s41467-020-19497-z.
[131] T. Loose, P. Sahrmann, T. Qu, G. Voth, Coarse-graining with equivariant neural networks: a path towards accurate and data-efficient models, ArXiv (2023).
[132] J. Vita, E. Fuemmeler, A. Gupta, G. Wolfe, A. Tao, R. Elliott, S. Martiniani, E. Tadmor, ColabFit Exchange: open-access datasets for data-driven interatomic potentials, ArXiv (2023).
[133] C.M. Andolina, W.A. Saidi, Highly transferable atomistic machine-learning potentials from curated and compact datasets across the periodic table, Digital, Discovery 2 (2023) 1070–1077, https://doi.org/10.1039/D3DD00046J.
[134] E. Fuemmeler, G. Wolfe, A. Gupta, J.A. Vita, E.B. Tadmor, S. Martiniani, Advancing the ColabFit Exchange towards a Web-scale Data Source for Machine Learning Interatomic Potentials, in: AI for Accelerated Materials Design - NeurIPS 2024, 2024.
[135] B. Blaiszik, W. Engler, K. Schmidt, Garden: A FAIR Framework for Publishing and Applying AI Models for Translational Research in Science, Engineering, Education, and Industry, (2024). https://thegardens.ai/#/home (accessed February 20, 2024).
[136] R. Jacobs, L. Schultz, A. Scourtas, K.J. Schmidt, O. Price-Skelly, W. Engler, I. Foster, B. Blaiszik, P. Voyles, D. Morgan, Machine Learning Materials Properties with Accurate Predictions, Uncertainty Estimates, Domain Guidance, and Persistent Online Accessibility, Machine Learning Science and Technology 5 (2024) 4, https://doi.org/10.1088/2632-2153/ad95db.
[137] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, The materials data facility: data services to advance materials science research, JOM 68 (2016) 2045–2052, https://doi.org/10.1007/s11837-016-2001-3.
[138] B. Blaiszik, L. Ward, M. Schwarting, J. Gaff, R. Chard, D. Pike, K. Chard, I. Foster, A data ecosystem to support machine learning in materials science, MRS Commun. 9 (2019) 1125–1133, https://doi.org/10.1557/mrc.2019.118.
[139] K. Schmidt, A. Scourtas, L. Ward, S. Wangen, M. Schwarting, I. Darling, E. Truelove, A. Ambadkar, R. Bose, Z. Katok, J. Wei, X. Li, R. Jacobs, L. Schultz, D. Kim, M. Ferris, P.M. Voyles, D. Morgan, I. Foster, B. Blaiszik, Foundry-ML - software and services to simplify access to machine learning datasets in materials science, J Open Source Softw 9 5467 (2024) 10.21105/joss.05467.
[140] S.S. Schoenholz, E.D. Cubuk, JAX, M.D. A framework for differentiable physics, Adv Neural Inf Process Syst 2020-Decem (2020). 10.1088/1742-5468/ac3ae9.
[141] A. Hjorth Larsen, J. JØrgen Mortensen, J. Blomqvist, I.E. Castelli, R. Christensen, M. Dułak, J. Friis, M.N. Groves, B. Hammer, C. Hargus, E.D. Hermes, P.C. Jennings, P. Bjerre Jensen, J. Kermode, J.R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. SchiØtz, O. Schütt, M. Strange, K.S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K.W. Jacobsen, The atomic simulation environment - A Python library for working with atoms, Journal of Physics Condensed Matter 29 (2017). 10.1088/1361-648X/aa680e.
[142] D.M. Anstine, O. Isayev, Machine learning interatomic potentials and long-range physics, Chem. A Eur. J. 127 (2023) 2417–2431, https://doi.org/10.1021/acs.jpca.2c06778.
[143] Z.L. Glick, D.P. Metcalf, A. Koutsoukas, S.A. Spronk, D.L. Cheney, C.D. Sherrill, AP-Net: An atomic-pairwise neural network for smooth and transferable interaction potentials, J. Chem. Phys. 153 (2020), https://doi.org/10.1063/5.0011521.
[144] G. Pan, J. Ding, Y. Du, D.-J. Lee, Y. Lu, A DFT accurate machine learning description of molten ZnCl2 and its mixtures: 2. Potential development and properties prediction of ZnCl2-NaCl-KCl ternary salt for CSP, Comput Mater Sci 187 (2021) 110055, https://doi.org/10.1016/j.commatsci.2020.110055.
[145] M. Bu, W. Liang, G. Lu, Molecular dynamics simulations on AlCl3-LiCl molten salt with deep learning potential, Comput. Mater. Sci 210 (2022) 111494, https://doi.org/10.1016/j.commatsci.2022.111494.

[146] R. Chahal, S. Roy, M. Brehm, S. Banerjee, V. Bryantsev, S.T. Lam, Transferable deep learning potential reveals intermediate-range ordering effects in LiF–NaF–ZrF $_4$ Molten Salt, JACS Au 2 (2022) 2693–2702, https://doi.org/10.1021/jacsau.2c00526.
[147] G. Ceder, G.D. Garbulsky, P.D. Tepesch, Convergent real-space cluster expansion for configurational disorder in ionic systems, PhysRevB 51 (1995) 11257–11261, https://doi.org/10.1103/PhysRevB.51.11257.
[148] L. Zhang, H. Wang, M.C. Muniz, A.Z. Panagiotopoulos, R. Car, E. Weinan, A deep potential model with long-range electrostatic interactions, J. Chem. Phys. 156 (2022), https://doi.org/10.1063/5.0083669.
[149] A. Gao, R.C. Remsing, Self-consistent determination of long-range electrostatics in neural network potentials, Nat. Commun. 13 (2022) 1572, https://doi.org/10.1038/s41467-022-29243-2.
[150] A. Kabylda, J.T. Frank, S.S. Dou, A. Khabibrakhmanov, L.M. Sandonas, O.T. Unke, S. Chmiela, K.-R. Muller, A. Tkatchenko, Molecular Simulations with a Pretrained Neural Network and Universal Pairwise Force Fields, (2024). 10.26434/chemrxiv-2024-bdfr0.
[151] S. Chmiela, A. Tkatchenko, H.E. Sauceda, I. Poltavsky, K.T. Schütt, K.-R. Müller, Machine learning of accurate energy-conserving molecular force fields, Sci. Adv. (2017), https://doi.org/10.1126/sciadv.1603015.
[152] U. von Barth, L. Hedin, A local exchange-correlation potential for the spin polarized case. i, Journal of Physics c: Solid State Physics 5 (1972) 1629–1642, https://doi.org/10.1088/0022-3719/5/13/012.
[153] E. Stoner, Collective electron ferronmagnetism, Proc. R. Soc. Lond. A 165 (1938) 372–414, https://doi.org/10.1098/rspa.1938.0066.
[154] R. Drautz, D.G. Pettifor, Valence-dependent analytic bond-order potential for transition metals, PhysRevB 74 (2006) 174117, https://doi.org/10.1103/PhysRevB.74.174117.
[155] W. Heisenberg, Zur Theorie des Ferromagnetismus, Z. Angew. Phys. (1928).
[156] J. Tranchida, S.J. Plimpton, P. Thibaudeau, A.P. Thompson, Massively parallel symplectic algorithm for coupled magnetic spin dynamics and molecular dynamics, J. Comput. Phys. 372 (2018) 406–425, https://doi.org/10.1016/j.jcp.2018.06.042.
[157] S. Nikolov, M.A. Wood, A. Cangi, J.-B. Maillet, M.-C. Marinica, A.P. Thompson, M.P. Desjarlais, J. Tranchida, Data-driven magneto-elastic predictions with scalable classical spin-lattice dynamics, npj Comput. Mater. 7 (2021) 153, https://doi.org/10.1038/s41524-021-00617-2.
[158] H. Yu, C. Xu, X. Li, F. Lou, L. Bellaiche, Z. Hu, X. Gong, H. Xiang, Complex spin Hamiltonian represented by an artificial neural network, PhysRevB 105 (2022) 174422, https://doi.org/10.1103/PhysRevB.105.174422.
[159] M. Eckhoff, J. Behler, High-dimensional neural network potentials for magnetic systems using spin-dependent atom-centered symmetry functions, npj Comput. Mater. 7 (2021) 170, https://doi.org/10.1038/s41524-021-00636-z.
[160] 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 (2022) 13, https://doi.org/10.1038/s41524-022-00696-9.
[161] M. Domina, M. Cobelli, S. Sanvito, Spectral neighbor representation for vector fields: Machine learning potentials including spin, PhysRevB 105 (2022) 214439, https://doi.org/10.1103/PhysRevB.105.214439.
[162] J.B.J. Chapman, P.-W. Ma, A machine-learned spin-lattice potential for dynamic simulations of defective magnetic iron, Sci. Rep. 12 (2022) 22451, https://doi.org/10.1038/s41598-022-25682-5.
[163] R. Drautz, Atomic cluster expansion of scalar, vectorial, and tensorial properties including magnetism and charge transfer, PhysRevB 102 (2020), https://doi.org/10.1103/PhysRevB.102.024104.
[164] M. Rinaldi, M. Mrovec, A. Bochkarev, Y. Lysogorskiy, R. Drautz, Non-collinear Magnetic Atomic Cluster Expansion for Iron, (2023). http://arxiv.org/abs/2305.15137.
[165] H. Yu, Y. Zhong, J. Ji, X. Gong, H. Xiang, Time-reversal equivariant neural network potential and Hamiltonian for magnetic materials, ArXiv (2022).
[166] H. Yu, Y. Zhong, L. Hong, C. Xu, W. Ren, X. Gong, H. Xiang, Spin-dependent graph neural network potential for magnetic materials, ArXiv (2023).
[167] J. Westermayr, P. Marquetand, Machine learning and excited-state molecular dynamics, Mach. Learn.: Sci. Technol. 1 (2020) 043001, https://doi.org/10.1088/2632-2153/ab9c3e.
[168] J. Westermayr, P. Marquetand, Machine learning for electronically excited states of molecules, Chem. Rev. 121 (2021) 9873–9926, https://doi.org/10.1021/acs.chemrev.0c00749.
[169] J. Westermayr, M. Gastegger, P. Marquetand, Combining SchNet and SHARC: the SchNarc machine learning approach for excited-state dynamics, J. Phys. Chem. Lett. 11 (2020) 3828–3834, https://doi.org/10.1021/acs.jpclett.0c00527.
[170] J. Westermayr, M. Gastegger, M.F.S.J. Menger, S. Mai, L. González, P. Marquetand, Machine learning enables long time scale molecular photodynamics simulations, Chem. Sci. 10 (2019) 8100–8107, https://doi.org/10.1039/C9SC01742A.
[171] Y. Zhang, R.J. Maurer, B. Jiang, Symmetry-adapted high dimensional neural network representation of electronic friction tensor of adsorbates on metals, J. Phys. Chem. C 124 (2020) 186–195, https://doi.org/10.1021/acs.jpcc.9b09965.
[172] Y. Zhang, B. Jiang, Universal machine learning for the response of atomistic systems to external fields, Nat. Commun. 14 (2023) 6424, https://doi.org/10.1038/s41467-023-42148-y.
[173] G. Fonseca, I. Poltavsky, V. Vassilev-Galindo, A. Tkatchenko, Improving molecular force fields across configurational space by combining supervised and unsupervised machine learning, J. Chem. Phys. 154 (2021), https://doi.org/10.1063/5.0035530.
[174] T. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, ArXiv, 2017.
[175] OpenKIM: Interatomic Potentials and Analytics for Molecular Simulation, (2023). https://openkim.org/ (accessed October 30, 2023).
[176] R. Jinnouchi, F. Karsai, G. Kresse, On-the-fly machine learning force field generation: application to melting points, PhysRevB 100 (2019) 014105, https://doi.org/10.1103/PhysRevB.100.014105.
[177] K. Santos, S. Moore, T. Oppelstrup, A. Sharifian, I. Sharapov, A. Thompson, D. Kalchev, D. Perez, R. Schreiber, Breaking the molecular dynamics timescale barrier using a wafer-scale system, ArXiv (2024).
[178] A. Allen, N. Lubbers, S. Matin, J. Smith, R. Messerly, S. Tretiak, K. Barros, Learning Together: towards foundational models for machine learning interatomic potentials with meta-learning, ArXiv (2023).
[179] T. Shiota, K. Ishihara, T.M. Do, T. Mori, W. Mizukamai, Taming multi-domain, -fidelity data: towards foundation models for atomistic scale simulations, ArXiv 10.48550/arXiv (2024) 2412.13088.
[180] Z. Yu, A. Annamareddy, D. Morgan, B. Wang, How close are the classical two-body potentials to ab initio calculations? Insights from linear machine learning based force matching, ArXiv (2023).
[181] 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, PhysRevB 104 (2021) 035120, https://doi.org/10.1103/PhysRevB.104.035120.
[182] J. Hermann, Z. Schätzle, F. Noé, Deep-neural-network solution of the electronic Schrödinger equation, Nat. Chem. 12 (2020) 891–897, https://doi.org/10.1038/s41557-020-0544-y.
[183] C. Shen, S. Attarian, Y. Zhang, H. Zhang, M. Asta, I. Szlufarska, D. Morgan, SuperSalt: Equivariant Neural Network Force Fields for Multicomponent Molten Salts System, ArXiv (2024), https://doi.org/10.48550/arxiv.2412.19353.