

<https://doi.org/10.1038/s41524-024-01441-0>

From electrons to phase diagrams with machine learning potentials using pyiron based automated workflows

Check for updates

Sarath Menon¹✉, Yury Lysogorskiy², Alexander L. M. Knoll^{3,4}, Niklas Leimeroth⁵, Marvin Poul⁶, Minaam Qamar², Jan Janssen⁶, Matous Mrovec², Jochen Rohrer⁵, Karsten Albe⁵, Jörg Behler^{3,4}, Ralf Drautz² & Jörg Neugebauer⁶✉

We present a comprehensive and user-friendly framework built upon the `pyiron` integrated development environment (IDE), enabling researchers to perform the entire Machine Learning Potential (MLP) development cycle consisting of (i) creating systematic DFT databases, (ii) fitting the Density Functional Theory (DFT) data to empirical potentials or MLPs, and (iii) validating the potentials in a largely automatic approach. The power and performance of this framework are demonstrated for three conceptually very different classes of interatomic potentials: an empirical potential (embedded atom method - EAM), neural networks (high-dimensional neural network potentials - HDNNP) and expansions in basis sets (atomic cluster expansion - ACE). As an advanced example for validation and application, we show the computation of a binary composition-temperature phase diagram for Al-Li, a technologically important lightweight alloy system with applications in the aerospace industry.

The advent of machine learning interatomic potentials (MLPs) is revolutionising the field of computational materials science, enabling simulations of large systems and complex material properties with *ab initio* accuracy¹⁻⁵. However, the development of these data-driven interatomic potentials is a computationally intensive task that needs automated and reliable workflows.

The life cycle of MLP development can be broadly divided into the following tasks: (i) generating a database containing reference data, (ii) fitting the model parameters to the reference data, and (iii) validating the resulting parametrization for a specified range of properties. Furthermore, it is often necessary to provide a feedback loop between the tasks via an active learning approach to ascertain transferability^{6,7}.

The initial task of setting up the reference database usually requires to perform many thousands of density functional theory (DFT) calculations for a broad range of atomic environments that span the configuration space of interest as completely as possible. Such computations can nowadays be facilitated using either general workflow frameworks⁸⁻¹¹ or tools designed specifically for a particular MLP class¹²⁻¹⁷. Nevertheless, there is still lack of standardized workflow setups, computational metaparameters and structural databases. Therefore, each research group relies mostly on their own

expertise and experience. This may not only lead to inconsistencies in the generated data (for instance, due to variations in DFT settings, such as the exchange-correlation functional, Brillouin zone sampling or plane wave cutoff)^{18,19}, but it can also strongly limit exchange of data from different sources and their collection into greater databases.

The situation remains similar when it comes to the second stage of MLP development, namely, fitting the model parameters. Many optimization algorithms and software tools are tailored to a particular class of potentials and are not easily transferable. Thus, a researcher must not only identify an appropriate type of potential that is suited for the system of interest, but often needs to learn a variety of specific software tools each of which uses its own terminology.

The final task in the development cycle is a thorough validation of the fitted parametrization. It should be stressed that simple correlations between the predictions of the potential and the reference DFT data, e.g. for energies and forces, are in most cases not sufficient and may be even misleading. It is crucial to evaluate not only fundamental physical properties, such as energy-volume curves, elastic moduli or phonon spectra, but to perform also dynamical simulations at finite temperatures that scrutinize

¹Max-Planck-Institut für Nachhaltige Materialien GmbH, 40237 Düsseldorf, Germany. ²ICAMS, Ruhr-Universität Bochum, 44801 Bochum, Germany. ³Lehrstuhl für Theoretische Chemie II, Ruhr-Universität Bochum, 44780 Bochum, Germany. ⁴Research Center Chemical Sciences and Sustainability, Research Alliance Ruhr, 44780 Bochum, Germany. ⁵Technische Universität Darmstadt, Fachbereich Material und Geowissenschaften, Fachgebiet Materialmodellierung, 64287 Darmstadt, Germany. ⁶Max-Planck-Institut für Eisenforschung GmbH, 40237 Düsseldorf, Germany. ✉e-mail: s.menon@mpie.de; neugebauer@mpie.de

spurious behaviour of the model outside of the training domain. Existing initiatives^{20–22} have mostly focused on classical interatomic potentials while automatized validations of MLPs are still rare²³. This makes it difficult for users to determine a safe application range of a particular MLP parametrization for their application. A thorough validation is also indispensable before applying the model in simulations of complex material properties, such as studies of extended defects, phase transformations, or predictions of phase diagrams.

Our aim is to demonstrate the whole MLP development cycle for three representative model potentials to elucidate the complete process to interested researchers, not on a benchmark comparison of different types of potentials. We introduce a set of standardised workflows that cover all aspects from generation of DFT data to MLP fitting and validation, as schematically illustrated in Fig. 1. As an example of an advanced application, we evaluate the phase diagrams for a prototypical binary system. The workflows presented here are reusable, reproducible, and most importantly, largely automated. While exposing all intricacies of the methods involved, we show that they significantly reduce the technical complexity. By providing the computer codes and software tools, we encourage to use this manuscript as a practitioner's guide into the field of modern MLP development as well as advanced thermodynamic applications.

As a model case, we chose the binary Al-Li system^{24,25}. Al-Li alloys are well suited for aerospace applications since they exhibit low density and high mechanical strength^{26,27}. Apart from a recent work²⁸, there is a lack of interatomic potential for this system, making it both desirable and a challenging option from the perspective of MLP development. We selected three prototypical examples of interatomic potentials: a classical central-force potential based on the embedded atom method (EAM)^{29,30}, a high-dimensional neural network potential (HDNNP)^{31,32}, and the atomic cluster expansion (ACE)^{6,33}. We use calphy³⁴ for the calculation of phase diagram, and employ pyiron¹¹ as a workflow creation and management environment to bring together various software tools. Our goal is to enable seamless

creation and validation of interatomic potentials while taking a step towards the FAIR (Findable, Accessible, Interoperable, and Reusable) data and software principles^{35,36} in the field of MLP development.

Results

pyiron as a platform for automated workflows

We employ pyiron¹¹ as a workflow environment for all stages of the MLP development and validation, as illustrated in Fig. 1. The development cycle is facilitated with pyiron by connecting the fundamental building blocks:

1. Generic and easy-to-use structure generation tools that combine standard software libraries in the field of computational materials science such as ASE³⁷, pymatgen³⁸, PyXtal³⁹, and others in a convenient and interoperable package;
2. An interoperable interface to a variety of electronic structure and atomistic simulation software packages such as VASP^{40–42} and LAMMPS⁴³;
3. A common storage format for energies, forces and stresses that can be used efficiently for hundreds of thousands of training configurations implemented in pyiron as the class TrainingContainer (see Section “TrainingContainer”);
4. A common interface to the fitting tools used in this work, namely, pacemaker^{22,44}, RuNNer^{45–47}, and atomicrex⁴⁸, implemented in pyiron as the PotentialFit class;
5. A common interface to validation workflows and tools for thermodynamic properties such as phonopy⁴⁹ and calphy³⁴.

Block (1) enables users with different backgrounds to generate easily new configurations or to import them from existing databases. The common interface in Block (2) provides a seamless switching between different quantum engines or simulation protocols to create training data with minimal changes in an existing workflow. It can also be employed to design and test a workflow using a lower-level method, which is computationally cheap, before switching to a production run using a higher-level theory. Blocks (3) and (4) provide flexibility to experiment with different MLP formalisms on the same data or selected subsets. In addition, Block (3) provides analysis and plotting routines for all types of training data

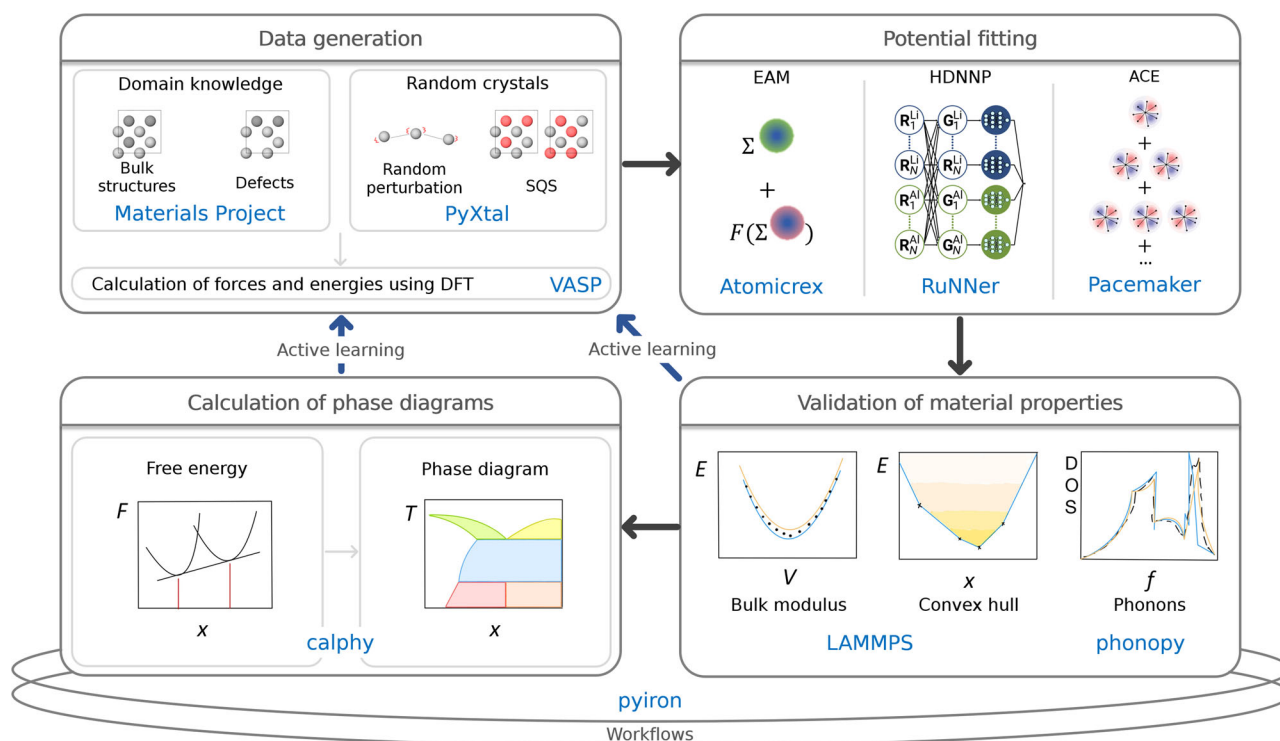


Fig. 1 | A schematic illustration of the MLP development cycle. Here, pyiron is employed as a workflow manager to combine different software tools and packages.

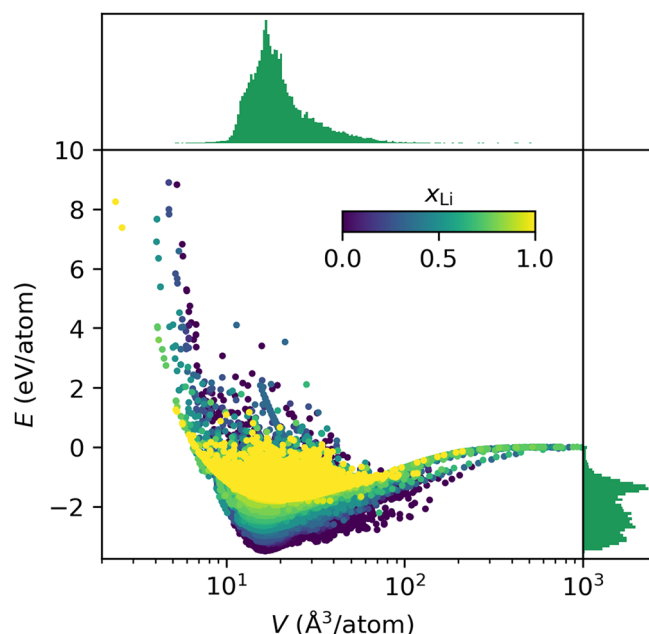


Fig. 2 | Energy distribution of the DFT reference data set as a function of the atomic volume. The fraction of lithium atoms in each structure is represented by the colour of the points.

generated in Block (2). Finally, Block (5) provides access to validation routines, helping a user to assess the quality of the fitted MLPs.

Construction of a reference DFT dataset for the Al-Li system

Selection of the reference data needed to parametrize an interatomic potential is one of the most important steps in the life cycle of MLP development. For creation of the reference DFT data we employed VASP 5.4^{40–42}, using workflows as described in Section “VASP”. We employed the projector augmented wave (PAW) method⁵⁰ and the GGA-PBE exchange correlation functional⁵¹. Several convergence tests were conducted to ensure the obtained energies and forces are highly accurate and consistent. These tests were carried out for three representative structures, namely, face-centred cubic (fcc) Al, body-centred cubic (bcc) Li, and the B32-type β -LiAl, in a range of 30 % volumetric strain around their respective equilibrium volumes. Based on these tests, the following DFT settings were used: a plane wave cutoff of 750 eV, a k-mesh spacing of 0.1 \AA^{-1} , and the Fermi smearing method with a width of 0.1 eV. With these settings we observed less than 0.5 meV/atom difference in the energies as compared to calculations performed at 800 eV plane wave cutoff and 0.05 \AA^{-1} k-mesh spacing.

There exist multiple strategies for generation of relevant atomic configurations. We find that a combination of domain knowledge, active learning, and random search can be employed effectively for the construction of a balanced training dataset. In this three-step strategy, domain knowledge is employed first to select structures based on common structural prototypes available in standard crystallographic databases⁵². Thereafter, we use active learning algorithms during validation and simulations, and random configurations obtained using a random-structure-search procedure⁵³ to augment the dataset.

The domain-knowledge step is focused on structures that are known to be important for the system of interest and to ensure that they are represented with high accuracy. In this work, we queried both elemental and binary structures with formation energies less than or equal to zero from the Materials Project database⁵². Subsequently, a series of transformations was applied to these structures and their supercells, including uniform and non-uniform deformations of the cells and random displacements of atoms (see Supplementary Note 1-2). These steps ensure that not only perfect bulk

structures but also their distortions, which are crucial to reproduce elastic and vibrational properties, are included in the training data.

We then used active learning to ensure that even during extended simulations, which are needed to compute thermodynamic properties (See Section “calphy”), potentials remain stable and accurate. Within the active learning loop, we iteratively selected structures based on high uncertainty indicators⁵⁴ derived from running molecular dynamics simulations for several Al-Li phases of interest.

Finally, we added random structures that are far from equilibrium. This step ensures a broader coverage of the configurational space. Relying on domain knowledge and active learning only may result in a short sighted training set that hampers the extrapolation capabilities of the fitted MLPs, whereas a random sampling-based approach alone might lead to the risk of missing or under-representing important phases in a material. The random structures were generated following a recent workflow⁵³ that utilizes the PyXtal³⁹ software as described in Section “PyXtal”. We generated an initial set of structures of each space group for varying numbers of atoms (Not all space groups can be generated for every composition of the cells due to Wyckoff multiplicity constraints.), which are then relaxed using DFT first allowing only the volume to vary, followed by a full relaxation of the cell shape and internal degrees of freedom. It is not necessary that these relaxations lead to highly accurate minima in the potential energy landscape, so low accuracy DFT calculations can be employed to speed up this step. These relaxed structures are then recomputed using the required precision to ensure consistency of the training dataset. This procedure, very similar to ab initio structure search methods⁵⁵, and originally developed for this purpose, has recently been applied for machine learning potentials⁵⁶. The primary advantage of this approach is that it can help to find basins of the potential energy surface without domain knowledge, while also exposing the potential to a greater variety of structural and chemical environments⁵³.

Finally, random displacements and variations of the cell shape and size were applied to the relaxed structures to obtain additional samples around the minima of the potential energy surface. Detailed parameters for the random perturbations are described in the Supplementary Note 1. The initial set of random crystals as well as structures resulting from both the minimization steps and the random perturbations are then combined and added to the training set. The complete workflow is implemented in `pyiron` and the primitives introduced in the previous section II A.

Through the combination of these three strategies — domain-knowledge, active learning, and random search — we were able to construct a robust and extensive atomic structure data set that captures a wide range of configurations. The distributions of DFT reference data over energy, volume and composition are shown in Fig. 2 and further information is provided in the Supplementary Figs. 1–3.

Training of data-driven interatomic potentials

The training of MLPs is usually carried out using dedicated computer codes that are tailored to a particular model architecture. In our case, we used `atomicrex`⁴⁸, `RuNer`^{45–47} and `pacemaker`⁴⁴ to fit the EAM, HDNNP and ACE parametrizations, respectively. Due to differences in the fitting procedures, the training data sets needed to be adjusted to the respective codes and models.

For the EAM potential, we started by fitting potentials for the single elements, following an approach outlined by Mishin et al.⁵⁷. This approach guarantees an exact fit of the lattice constant, cohesive energy and bulk modulus by constraining the fitting parameters accordingly. Parameters of functions describing Al-Li were fitted while keeping the single element parameters constant. The training data was limited to the domain knowledge subgroup of the whole set, containing 2081 structures. Because the potential has less than 100 adjustable parameters, this amount of training data is sufficient, and leads to faster parametrization routines. Furthermore, the functional form of EAMs have limited flexibility, so training to randomly

Table 1 | Energy and force RMSE (MAE) of EAM, HDNNP, and ACE with respect to DFT

Potential	Dataset Size	Energy	Force	Volume	E		F	
		Range	Range	Range	RMSE (MAE)		RMSE (MAE)	
		[eV atom ⁻¹]	[eV Å ⁻¹]	[Å ³ atom ⁻¹]	[meV atom ⁻¹]		[meV Å ⁻¹]	
EAM	2081	58	52	235	554 (82.2)	118 (89.9)	192 (116)	143 (68.7)
HDNNP	50834	3.5	20	40	10.6 (7.1)	10.2 (6.9)	64.2 (30.6)	49.2 (21.0)
ACE	51082	50	40	600	12.2 (7.5)	9.6 (6.6)	41.4 (16.9)	26.5 (11.7)

Test set metrics are given for a common test set including only structures within 1 eV/atom above the convex hull of the system. The size of the dataset and the energy, force and volume ranges of training and testing data are also summarized.

sampled structures far from equilibrium could impact the accuracy of the more important low-energy structures. In the fitting process, energies were weighted based on the distance of the formation energy to the convex hull E_D (in eV/atom) as

$$W_E = \frac{100}{(E_D + 0.2)^4}, \quad (1)$$

while a uniform weight of one was applied for forces. Further details about the fitting procedure are provided in the Methods section and in ref. 57. Details on the employed functional forms and constraints can be found in Supplementary Note 3.

Before starting the HDNNP training process, the training data set was refined by eliminating structures which were not relevant for the material properties of interest. These included structures containing isolated atoms without any bonding partners within a radius of 12 Å, structures with large positive formation energies or highly repulsive force components, and all structures with atomic volume outside the interval of 10 Å³/atom–50 Å³/atom. In total, 4915 data points were removed.

Using RUNNERASE⁴⁷, we then carried out a grid search to optimize the hyperparameters required for the HDNNP training performed with the RUNNER code^{45,46}. In particular, this includes the number, short-range cutoff radius and parameters of the atom-centred symmetry functions (ACSFs)⁵⁸ describing the atomic environments, the hyperparameters of the optimisation algorithm (Kalman filter⁵⁹), and the neural network architecture. For each trial, HDNNPs were trained on five randomly selected mini-batches of 200 data points and three random initialization seeds each. 1 ps NVT MD rapid heating runs between 300 K and 1000 K were performed to test the capabilities of each potential in a basic application. Finally, the best hyperparameters were selected based on the training accuracy and simulation stability (length of the stable trajectories, number of extrapolations) which were achieved across the 15 members of the group. The selected hyperparameters are presented in Supplementary Note 4 and Supplementary Table 3. Then, a HDNNP was trained with the selected settings on the entire training dataset, randomly separated into a training (90%) and testing set (10%). All data points have been equally weighted in the training process. The same applies to the relative weight of total energies and force components.

The ACE parameterization was carried out using the Pacemaker package⁴⁴. A cutoff for all interactions was set to 7 Å, based on the range of DFT interactions. The total dataset was randomly divided into training and testing sets with a ratio of 95% to 5%. Weights for the training structures were assigned based on their energy distance to the convex hull, following the energy-based weighting method⁴⁴.

We performed a set of parameterizations with an increasing number of basis functions (hierarchical basis extension⁴⁴) and chose basis set size equal to 1000 functions per element as a good compromise between predictive power and computational performance. This choice was based on the analysis of errors in energy, forces, formation energies, and elastic constants (see SI for more details). The final selected potential configuration, i.e. maximum radial and angular indices, dependent on the correlation order, as well as other ACE parameters are presented in Supplementary Note 5.

The final parameterization was performed from scratch in two stages. In the first stage, a higher emphasis was placed on forces ($\kappa = 0.99$), while in the second stage a more balanced distribution of energy-forces weights ($\kappa = 0.3$) was used. A strong core repulsion pair potential was added at distances below 2 Å.

Comparison of training outcomes

An overview of all training datasets as well as the achieved training accuracy for all three potentials is given in Table 1 and the correlations between predicted and reference values for energies and force components are provided in Fig. 3.

It is seen that all three fitting methods yield favourable training results despite the different train set sizes and compositions. In line with our expectations, the physically-inspired EAM requires the least amount of training data spanning over large energy, force and volume ranges, albeit at the cost of higher training errors. In contrast, the HDNNP and ACE utilize most of the available training data and reach smaller errors with respect to the DFT reference values than EAM. Due to the higher weighting of low-energy structures employed in the training of the ACE potential, there are less outliers around forces close to zero in Fig. 3c, while such a weighting has not been applied in the HDNNP training.

To facilitate the validation and to compare objectively the accuracy of all potentials, we created a single test dataset containing only structures that were not part of any training dataset. The test structures were restricted to lie within 1 eV/atom or less above the convex hull, as these represent the physically most relevant subset for the phase diagram simulations. In Table 1 and Fig. 3, we depict test set metrics for this common test dataset only. For all potentials, the test error metrics are smaller than those for the training datasets. In the case of ACE, the overall metrics are additionally biased due to the non-uniform distribution of energy-based weights.

Validation approach and strategy

Once the potentials have been parameterized with the desired accuracy, they must be extensively validated. Energy and force RMSEs of the final fit provide a first quantitative assessment of the potentials with respect to the reference data. However, it is mandatory to evaluate a broader range of fundamental material properties and to compare them to DFT reference data, and when applicable, experimental observations.

An elementary assessment of transferability is to compare energies of important bulk phases as a function of atomic volume. This data is fitted to the Murnaghan equation of state to obtain Murnaghan curves, that contain not only valuable information about the mutual stability of various phases, but also can be used for an estimation of their bulk moduli. Figure 4 shows the Murnaghan curves as predicted by all three potentials for the fcc phase of Al, bcc and fcc phases of Li, and five binary phases (AlLi, Al₃Li, Al₂Li₃, AlLi₂, and Al₄Li₅) that appear in the phase diagram²⁵.

All potentials agree well with DFT for the ground states of Al and Li, with some minor variations observed for both Li phases of the order of a few meV (Note that the ground state according to reference PBE-DFT calculations is fcc, albeit with a small energy difference compared to bcc, as seen in Fig. 4b). When considering the binary phases in Fig. 4c, a clear distinction is observed between the EAM potential and the MLPs. While the MLPs

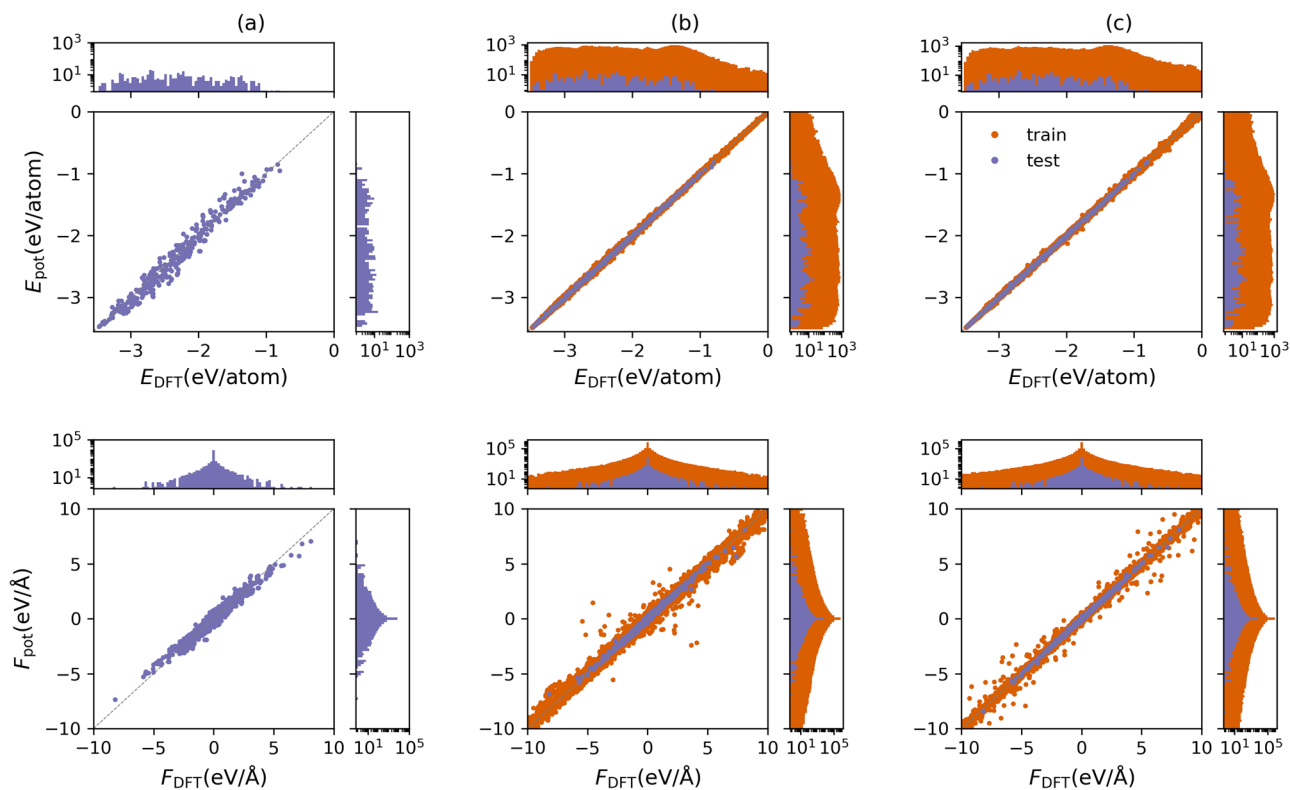


Fig. 3 | Energy and forces (E_{pot} and F_{pot}) predicted by the potentials compared to DFT data. EAM potential is shown in (a) while the HDNNP and ACE potentials are shown in (b) and (c), respectively. The corresponding values of the RMSE and MAE are given in Table 1. Note that training data (purple) is a different subset of the reference DFT dataset for each potential, while test data (orange) is based on a common test set including only structures within 1 eV/atom above the convex hull of the system.

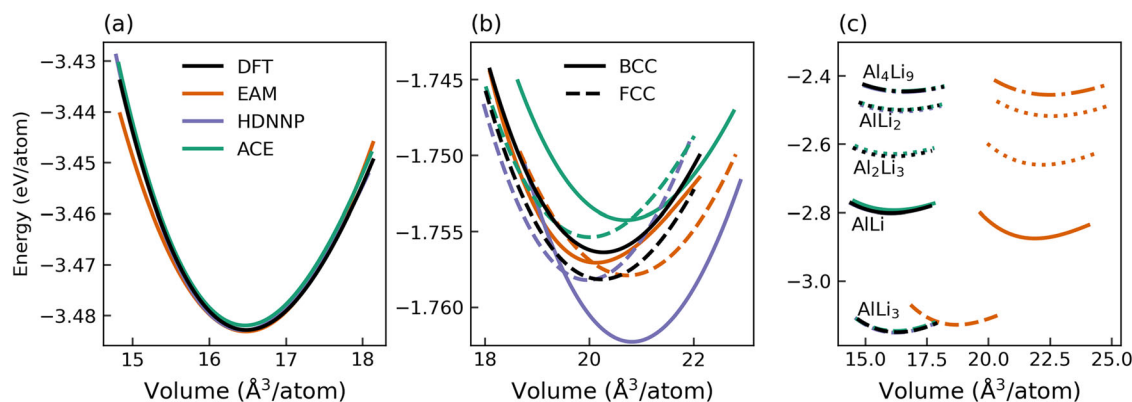


Fig. 4 | Equation of state curves. The predictions by the EAM, HDNNP, and ACE potentials for **a** pure fcc Al, **b** pure bcc and fcc Li, and **c** different AlLi compounds. The DFT reference is shown in black. In (b), as the bulk modulus of Li (11 GPa for bcc and 13 GPa for fcc) is lower than the respective value of Al (76 GPa), the range of energies is small, approximately 0.02 eV/atom, resulting in rather large visual discrepancies. For the binary compounds in (c), the predictions of the HDNNP, ACE, and DFT essentially coincide, and are thus hardly distinguishable.

predict both the atomic volumes as well as the formation energies in excellent agreement with DFT, the EAM potential shows a considerable overestimation of the atomic volumes.

The phonons predicted by the potentials are related to vibrational properties of materials and reflect the model’s behaviour for small perturbations near the equilibrium ground state structure that are relevant for an accurate reproduction of phase diagrams. Figure 5 shows the phonon densities of states for fcc Al, bcc Li, AlLi and Al₃Li. Note that we restrict ourselves to the two binary phases with $x_{\text{Li}} \leq 0.5$, as this is the region considered in the phase diagram calculations (see Section “Construction of thermodynamic phase

diagrams”). Similar to the Murnaghan curves, it is observed that HDNNP and ACE both predict the phonon DOS with good accuracy, while the EAM potential shows significant deviations for the binary phases.

As an initial step before evaluating the binary phase diagram, we evaluate and plot the formation energies of the binary phases as a function of Li content in Fig. 6. The so-called convex hull can in fact be thought of as a binary phase diagram at zero Kelvin. The convex hull plot allows the formation energies to be directly compared to DFT and our calculations show that both HDNNP and ACE predict the formation energies well with DFT accuracy while EAM predictions deviate from the reference.

Fig. 5 | Phonon density of states. Density of states as predicted by EAM, HDNNP, and ACE in comparison with the DFT reference for **a** fcc Al, **b** bcc Li, **c** AlLi, and **d** Al₃Li.

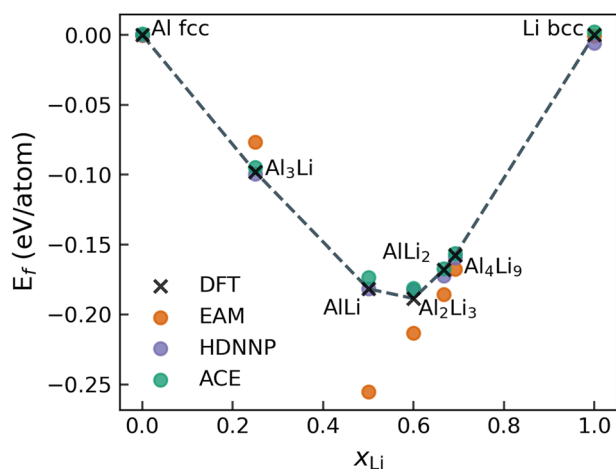
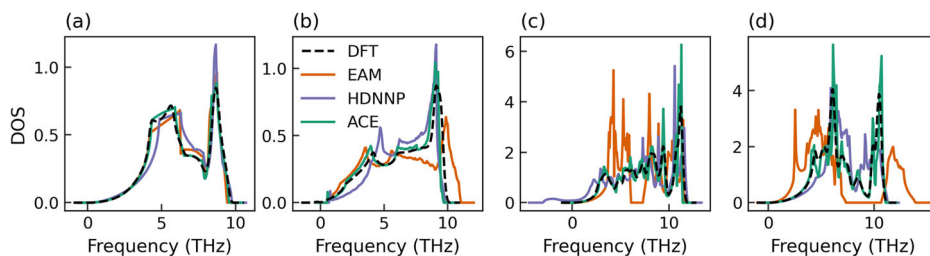


Fig. 6 | The convex hull for the Al-Li binary system as predicted by EAM, HDNNP, and ACE. The black dashed line connects the points along the DFT convex hull.

The elastic matrix elements C_{11} , C_{12} and C_{44} for the fcc Al and bcc Li ground states are computed with the fitted potentials and reported in Table 2. For fcc Al, the machine learning potentials match the DFT reference data very well while the EAM potential overestimates C_{11} and underestimates the other two elastic constants. For bcc Li, all potentials provide a good description of the elastic matrix.

Construction of thermodynamic phase diagrams

Phase diagrams provide critical information about the material system, the phases that are predicted to be stable at the given thermodynamic conditions, and the conditions at which one phase transitions to another, or two phases coexist. Phase diagrams are, therefore, a crucial and challenging test for interatomic potentials. In general, the given interatomic potential should be able to reproduce the key aspects of the phase diagram, or at least parts of it, pertaining to the expected thermodynamic region where the interatomic potential is to be employed.

The CALPHAD method⁶⁰ is perhaps the most well-known method for the calculation of phase diagrams, aided by experimental observations of thermodynamic properties of a system. From an atomistic perspective, different methods exist for the determination of phase diagrams^{61,62}. Broadly, the methods either evaluate phase stability directly, through approaches such as coexisting phase simulations^{63–65}, or indirectly, by determining the Gibbs free energy or chemical potential of the relevant phases⁶⁶. We follow the approach of calculating free energies, using the thermodynamic integration method, in which the free energy difference between a given system and a reference system is calculated^{67,68}. We combine thermodynamic integration with non-equilibrium Hamiltonian interpolation and reversible scaling to obtain the free energies efficiently (see Methods for more details). The workflow for such a calculation boils down to the code as described in Section “calphy”.

Table 2 | Elastic constants of elemental aluminium and lithium, given in GPa, as predicted by the three potentials and the DFT reference method

	DFT	EAM	HDNNP	ACE
Al-fcc				
C11	129	98	131	130
C12	52	67	67	67
C44	32	46	49	39
Li-bcc				
C11	15	15	12	13
C12	13	14	13	12
C44	11	12	12	11

For this methodology, a priori information about the relevant phases is needed, which is motivated by the currently established phase diagram^{25,69}. In order to have a set of robust, automated, and efficient workflows for the phase diagram determination, we consider only substitutional defects in the off-stoichiometric compounds, and limit ourselves to the left side of the phase diagram, until $x_{\text{Li}} = 0.5$. Therefore, we consider the fcc Al, AlLi in the bcc-like B32 lattice, and the liquid. Furthermore, the L1₂ Al₃Li appears as a metastable phase in the experimentally determined phase diagram⁶⁹, and on the convex hull determined through DFT calculations (see Fig. 6), which makes it an interesting candidate to be considered in the calculation of the phase diagram.

For pure Al, we present the pressure-temperature $P - T$ phase diagrams. In order to arrive at the $P - T$ phase diagrams, the free energies of the relevant phases are calculated as a function of temperature and pressure. To this end, we perform reversible scaling calculations which provide the free energy over a given temperature range. A pressure range of 0–40 GPa is chosen, with free energy calculations carried out at intervals of 10 GPa. The fcc and liquid phases are considered, and at each pressure, the melting temperature is obtained from the intersection of the free energy curves. A system size of approximately 7000 atoms is chosen for both phases such that any finite size effects are avoided. The same set of calculations is performed with all three potentials, the results of which are shown in Fig. 7. In general, all three potentials closely follow the predictions from experiments. Although the zero pressure melting temperature is underestimated compared to the experimental value⁷⁰, it is comparable with the melting temperature from ab initio calculations⁷¹.

For the construction of the binary phase diagram, the fcc and liquid are considered in the composition range $0 \leq x_{\text{Li}} \leq 0.5$, and B32 AlLi in $0.4 \leq x_{\text{Li}} \leq 0.5$, and Al₃Li in $0.2 \leq x_{\text{Li}} \leq 0.3$. We chose the composition ranges for AlLi and Al₃Li based on the relevant regions in the experimental phase diagram. We then ascertain that the free energies of these phases were significantly higher than the other phases outside of the selected composition range. In order to create an Al-rich fcc lattice with Li as impurity, Al atoms are randomly selected and replaced by Li, that is, we assume that Li impurities occupy substitutional lattice positions. Similarly, substitutional Al impurities are introduced in the B32 structure to create off-stoichiometric

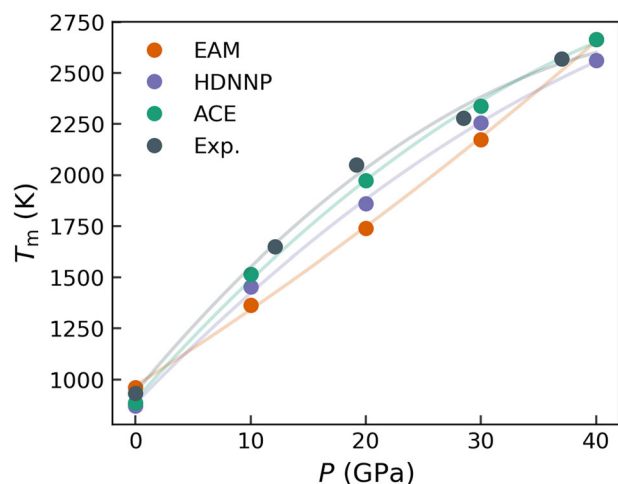


Fig. 7 | Melting curve of fcc Al up to 40 GPa. The melting temperature, T_m , at various pressures is calculated for EAM, HDNNP, and ACE. Melting temperatures determined from laser melting experiments⁹⁴ are marked in grey.

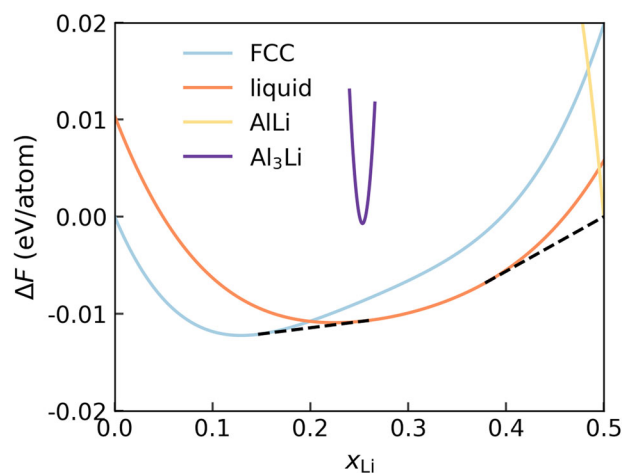


Fig. 8 | Free energy of mixing, ΔF , for fcc, liquid, AlLi, and Al₃Li at 800 K as a function of the composition of Li, calculated with the ACE potential. Substitutional impurity atoms are added in each of the phases to obtain free energy variation with composition (up to 0.5 Li). The two-phase coexisting regions are identified through common tangent constructions, indicated by black dashed lines.

compositions. No other mechanisms, such as vacancies, or interstitials are considered.

Within the selected composition range, we perform free energy calculations at composition intervals of 0.01 for all the phases. At each composition, temperatures from 600 and 1000 K are considered, and the free energy over this range is obtained in a single calculation using the reversible scaling approach. Free energy calculations are performed with timescales of 25 ps for equilibration, 50 ps for switching, and system size of roughly 7000 atoms for each phase.

Once the free energies are obtained, at each temperature the free energy for each phase with varying composition is extracted, as shown in Fig. 8. A current limitation of our workflow is that it does not include the contribution to the free energy due to configurational entropy in the solid phase, therefore the ideal mixing contributions are added to the fcc, B32, off-stoichiometric phases. In order to calculate the free energy of mixing, the end-members are chosen to be the phases with the lowest free energy at $x_{Li} = 0$ and $x_{Li} = 0.5$ at the given temperature. Finally, the convex hull is calculated

at each temperature to extract the regions of stability for each phase. Following such a construction, the regions of phase stability and coexistence can be obtained. Once again, the calculations are performed for all three potentials and the results are shown in Fig. 9a–c. The reference phase diagram calculated using the CALPHAD approach as implemented in the `pycalphad`⁷² tool, with an AllLi database from ref. 73, is shown in Fig. 9d.

The HDNNP and ACE exhibit phase diagrams (Fig. 9b, c) show excellent agreement with both the CALPHAD and the experimental phase diagrams. The main features of the phase diagram, such as the solubility of Li in the Al lattice, the eutectic point (liquid \rightarrow fcc + AlLi), general shape of the liquidus lines are all well reproduced. Both the eutectic composition and Li solubility, are close to the ranges in experimental observations. A comparison of the melting temperature of the end members, eutectic temperature, eutectic composition, and the solubility of Li in the fcc Al lattice are shown in Table 3. The phase diagram of the EAM potential, as represented in Fig. 9c, does not reproduce the characteristic features of the phase diagram, indicating the possible limitations of an empirical interatomic potential. Even though the EAM potential provides the closest estimate of the melting temperature of pure Al as compared to experiments, it predicts the stability of the competing phases incorrectly. Therefore, the applicability of the EAM potential in this study is limited to properties of the pure phases, such as the elastic constants.

The phase diagram calculated using the MLPs show some differences as compared to the Fig. 9d and the phase diagram from experiments. Both MLPs underestimate the melting temperature of Al and the eutectic temperature by approximately 6% and 12%, respectively, which is expected and consistent with the melting temperature predictions from DFT (refs. 74,75 and 76 with references therein). The prediction of a lower melting temperature by the MLPs are also evident at increasing pressure, as seen from Fig. 7.

Another major difference is the solubility of Al in the AlLi ordered phase. Experimental and CALPHAD phase diagrams show solubility, while HDNNP and ACE predictions are on the contrary. This discrepancy could be due to a limitation in the phase diagram calculation workflows rather than the interatomic potentials. The workflows for calculating phase diagrams presented here only considers substitutional defects for the off-stoichiometric compounds. However, in B32 AlLi, experimental studies propose that at lower Li concentrations, the Li atoms exhibit a vacancy mediated diffusion mechanism⁷⁷. The exclusion of vacancies in favour of substitutional defects could lead to the low solubility of Al in the AlLi phase. Furthermore, the inclusion of the different mechanisms would be needed to study regions of the phase diagram with $x_{Li} > 0.5$. Our workflows, however, can be extended beyond substitutional defects.

Although the Al₃Li phase is present on the DFT convex hull, it does not appear in the calculated phase diagram in the given temperature range, which is in agreement with previous studies⁷⁸. However, at lower temperatures, the ACE potential predicts regions of coexistence of the FCC and Al₃Li, and Al₃Li and AlLi, which disappears around 580 K (See Supplementary Note 6 and Supplementary Fig. 8).

Finally, the phase diagram workflow is limited to the ideal mixing approximation for the configurational entropy in the solid phases, which could have an impact on the calculated coexistence lines in this material system⁷⁸.

Overall, to obtain the phase diagram presented in this work for a potential, we require about 120 molecular dynamics simulations of about 150 ps each, making this approach computationally feasible even with the more expensive MLPs. Apart from the selection of relevant phases and temperature ranges, the rest of the workflow can be fully automated, allowing for the calculation of phase diagrams to be a routine task in the lifecycle of interatomic potential development. We observe that a 1 meV difference in free energy of phases could result in up to 50 K difference in the calculated transition temperature (see Supplementary Note 7 and Supplementary Fig. 9); this, in turn combined with the limitations of DFT in predicting transition temperatures indicate that a difference in transition temperatures as compared to the experimental phase diagrams is to be

Fig. 9 | Phase diagram of Al-Li up to $x_{\text{Li}} = 0.5$. The phase diagram calculated using (a) EAM, (b) HDNNP, and (c) ACE. In (d), the phase diagram calculated using the CALPHAD method is shown.

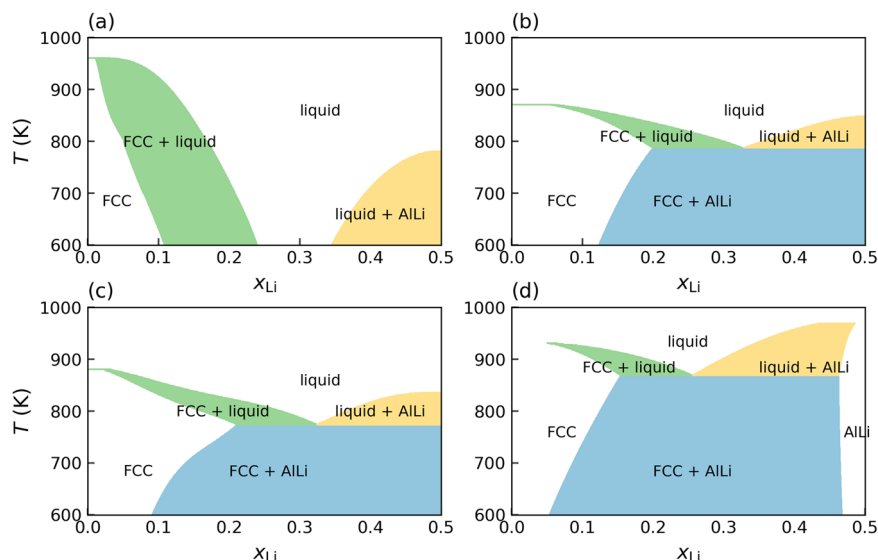


Table 3 | Comparison of the salient features of the phase diagram: the melting temperature of the end members, eutectic temperature, eutectic composition and the Li solubility in fcc Al at the eutectic temperature as predicted by the different interatomic potentials

Potential	Melting temperature (K), $x_{\text{Li}} = 0.0$	Melting temperature (K), $x_{\text{Li}} = 0.5$	Eutectic temperature (K)	Eutectic composition x_{Li}	Li solubility x_{Li}
EAM	961	782	<500	0.25–0.35	0.1–0.2
HDNNP	871	846	786	0.33	0.20
ACE	886	837	771	0.32	0.21
Exp.	933 ⁷⁰	965–991 ²⁵	871–876 ²⁵	0.234–0.300 ²⁵	0.124–0.180 ²⁵
DFT	888–972 ⁷⁵				

The melting temperature of fcc Al from DFT calculations is estimated to be 786–890 K⁷¹, depending on the chosen exchange–correlation functional (see Section “DFT Calculations”). Note that since the eutectic point does not appear in the phase diagram predicted by the EAM potential, the values are an estimation.

expected. Nevertheless, the calculated phase diagrams are highly beneficial to predict the thermodynamic conditions under which an interatomic potential is reliable, and for the interpretation of the observed phase transformation behaviour.

Discussion

In our presented framework, built upon the `pyiron` integrated development environment (IDE), establishes a comprehensive, robust, and user-friendly platform for the development of empirical and machine learning potentials. We have successfully demonstrated its versatility by running all tasks necessary in the development cycle of modern interatomic potentials, covering the creation of systematic Density Functional Theory databases, the fitting of DFT data to various interatomic potentials (EAM, HDNNP, and ACE), and the subsequent validation through a largely automated approach. The power and performance of the framework were exemplified in the computation of a binary composition–temperature phase diagram for the Al-Li alloy system, showcasing its applicability to running highly complex simulation protocols over large datasets consisting of thousands of individual atomic structures and for technologically important and complex materials systems.

The potential applications of the framework presented here are vast. Its user-friendly nature and adaptability make it an accessible and open tool for researchers in diverse fields, offering a streamlined approach to MLP development. Future efforts may focus on expanding the range of potential classes that can be incorporated, further enhancing the flexibility and

applicability of the framework to a wide range of materials science challenges requiring complex simulation protocols. Ongoing developments will seek to optimize and automate additional aspects of the MLP development cycle allowing researchers to address even more advanced materials properties needed e.g. to compute defect phase diagrams, thermoelectric behaviour, or superconductivity, thus paving the way for more efficient and reproducible research practices.

We envision our presented framework to act as a foundational platform, inviting researchers to explore and study the opportunities opened by machine learning potentials and their diverse applications. The ongoing commitment to openness, reproducibility, and automation positions our framework as a flexible and expandable basis for innovation and discovery in the quickly expanding landscape of using machine learning approaches in materials science. We therefore encourage the community to actively engage with the provided computer codes and software tools, which are openly provided via GitHub and Conda.

Methods

Workflows in materials science

The last few years showed a tremendous change in how high-performance compute clusters are used: While historically large monolithic codes allowed an up-scaling on an increasing number of cores, with the advent of machine learning a new type of computations becomes more and more important where huge numbers of small and medium-sized jobs running various codes need to be combined to get the final result. A prominent example is the

fitting and validation of machine learning potentials described in this paper. The number of DFT calculations needed to get high-quality potentials is in the range of a few ten thousands up to several hundreds of thousands individual DFT calculations. Data management for such a large number of jobs requires not only storing the input and output data but also of their status, i.e., whether they ran successfully, whether they converged, or whether they were aborted. If a jobs fail, they need to be resubmitted and it may be necessary to correct their input. For job sizes of a few 10,000 calculations, even small failure rates make a manual handling inefficient. For these types of advanced calculations automated workflow systems become almost mandatory.

In the present work, we have used `pyiron`¹¹ as a workflow platform to include all the necessary tools to create advanced machine learning potentials. `pyiron` provides features that are well-suited for these tasks: It provides an easy way to run large numbers of DFT calculations (to create the reference data set), to perform the training, and to analyse extensive sets of interatomic potential calculations (for validation). `pyiron` provides several features that make running such complex workflows efficient and intuitive for users. Its generic input and output provide an easy way to substitute one DFT code or potential/ML approach with another one. For example, to replace the DFT code the main change would be to change the job type. The generic input specifying the basis set, k-point sampling etc. remains unchanged and will be translated by `pyiron` into the code-specific format. The close integration within the Jupyter ecosystem provides interactive and easy access to all workflow components and data, and the availability of advanced job management tools provides an efficient route to upscale and run all calculations on modern supercomputer architectures.

DFT calculations

Density Functional Theory (DFT) is a quantum mechanical modelling approach that has become the de facto standard for ab initio computations of materials properties, especially for larger system sizes. This method allows for the calculation of materials properties without the need for fitting or empirical parameters, offering a rigorous and first-principles-based framework for providing the large data sets needed to fit empirical or machine learning potentials. In DFT, a pivotal approximation lies in the exchange-correlation (xc) functional. This approximation enables the reduction of the high-dimensional many-body interaction to a 3D mean field potential incorporated into the Kohn-Sham equations.

A restriction of all available xc-functionals is that they cannot be systematically improved, i.e., deviations to experiment are inherent. Common functionals, such as the PBE-GGA functional employed in this study, generally demonstrate good agreement with experimental results. However, it is important to note that deviations exist, with errors in bond lengths typically around 1%, discrepancies in elastic constants potentially reaching 10% and errors in the melting temperature in the order of 100 K.^{74,75}

While the exchange-correlation functional represents the only non-controllable approximation in DFT, there exist other parameters that can be used to systematically improve accuracy, albeit at an increased computational cost. Among these, the plane wave energy cutoff, which defines the completeness of the basis set, and the k-point sampling are particularly crucial. Achieving convergence in material properties concerning the choice of these parameters is imperative, especially when employing DFT data for training interatomic potentials. Inadequate convergence does not only lead to often non-systematic deviations from converged results but also introduces noise-like discontinuities in the energy surface due to the discrete nature of the plane wave basis set and the k-point set. For the generation of DFT datasets for potential fitting, it is therefore crucial to carefully select these convergence parameters to ensure that the amplitude of discontinuous fluctuations remains small compared to the targeted error. To be used for development of MLPs, this typically means an energy convergence to about 1 meV/atom and a force convergence to about 0.1 eV/Å.

When carefully choosing these convergence parameters, DFT is known to smoothly interpolate between similar structures. This characteristic renders DFT particularly well-suited for applications demanding a smooth

energy surface and derivatives (e.g. forces and stresses) such as developing interatomic potentials.

Embedded atom method

In EAM potentials the energy of the system is given by a pair potential V and a nonlinear function F , called embedding energy

$$E = \frac{1}{2} \sum_{ij} V(r_{ij}) + \sum_i F(\rho_i). \quad (2)$$

Here, ρ_i is given by $\rho_i = \sum_j \rho(r_{ij})$ and is called electron density. It is motivated by viewing each atom in a solid as impurity that is embedded in the host matrix and therefore subject to its electron density, leading to attractive chemical interactions. Then, V can be considered as repulsive core-core interaction²⁹. In modern EAM potentials V , ρ and F are chosen to best reproduce certain properties and do not necessarily follow the constraints resulting from this motivation, e.g. V often includes attractive terms. When freely choosing these function the EAM formalism is equivalent to the effective-medium⁷⁹ and Finnis-Sinclair⁸⁰ potentials. The potential we fitted closely follows a procedure applied by Mishin et al.⁵⁷.

High-dimensional neural network potentials

The general ansatz underlying the development of second-generation HDNNPs, introduced in 2007 by Behler and Parrinello^{31,32}, is that the total energy E_{tot} of a system can be decomposed into M environment-dependent atomic energy contributions E_i , such that

$$E_{\text{tot}} = \sum_{i=1}^M E_i(\vec{G}_i(\vec{r})). \quad (3)$$

This approach, which extended the applicability of MLPs to condensed systems containing large numbers of atoms, is based on the assumption that for many systems the atomic energy to a good approximation is a local property that depends only on the interaction of a central atom with its neighbouring atoms inside a sphere of radius r_G . The environment inside this cutoff sphere is captured by a vector \vec{G}_i of atom-centred symmetry functions, which in turn depend on the coordinates of all neighbours while maintaining the mandatory rotational, translational and permutational invariances. The functional form of these many-body descriptors is described in more detail elsewhere⁵⁸. Each entry in \vec{G}_i is passed to an input node of an element-specific dense feed-forward neural network, which provides the atomic energy as its output.

During training, the weights of all atomic neural networks are iteratively updated based on the loss gradients of both the total energies and the atomic force components in the training data set to achieve the best match to the reference values in the training set. Further information on the construction and training of HDNNPs can be found in ref. 45 and ref. 7.

Atomic cluster expansion

The atomic cluster expansion (ACE)³³ introduces basis functions that are complete in the space of atomic environments. In analogy to HDNNPs and other MLPs, the energy is represented by a sum of individual atomic energies within a cutoff sphere, for N atoms,

$$E_{\text{tot}} = \sum_i^N E_i. \quad (4)$$

The individual energies are calculated from general, abstract atomic properties (φ_i), which in ACE are expanded as

$$\varphi_i = \sum_v^{n_v} c_v B_{iv}, \quad (5)$$

where c_n are the expansion coefficients for the n_v basis functions $B_{i\nu}$. In linear ACE, E_i is written directly as

$$E_i = \varphi_i. \quad (6)$$

However, a more efficient approach is to calculate atomic energies as

$$E_i = \mathcal{F}(\varphi_i^{(1)}, \varphi_i^{(2)}, \dots, \varphi_i^{(P)}), \quad (7)$$

where \mathcal{F} can be any general non-linear function. The ACE potential used in this work employs a mildly non-linear form with two atomic properties and a square-root embedding as in the Finnis–Sinclair method

$$E_i = \varphi_i^{(1)} + \sqrt{\varphi_i^{(2)}}. \quad (8)$$

Thermodynamics

One of the most widely employed techniques to calculate free energies through atomistic simulations is thermodynamic integration^{67,68}. In this computational technique, a system of interest and a reference system with known free energy are coupled with a parameter λ . The Hamiltonian of the combined system is given by

$$H(\lambda) = \lambda H_f + (1 - \lambda) H_i \quad (9)$$

where H_i is the initial or reference system with the known free energy, and H_f is the final system, or the system of the interest. If the system of interest is in the solid state, we use a non-interacting Einstein crystal⁸¹ as the reference state, while for liquids, an Uhlenbeck–Ford model⁸² is employed. The free energy difference between the two systems can be calculated as

$$F_f - F_i = \int_{\lambda=0}^{\lambda=1} d\lambda \left\langle \frac{\partial H(\lambda)}{\partial \lambda} \right\rangle_{\lambda}. \quad (10)$$

The integration has to be performed over a discretized λ array, and therefore is computationally quite expensive, which calls for methodological improvements. In the non-equilibrium approach to thermodynamic integration⁸³, the coupling parameter λ is time-dependent, and the switching between the initial and final system is carried out in both forward and reverse directions in a single time-dependent calculation. The work done in such a switching process is calculated as,

$$W^s = \int_{t_i}^{t_f} \frac{d\lambda(t)}{dt} \frac{\partial H(\lambda)}{\partial \lambda} dt \quad (11)$$

which is related to the free energy difference ΔF between the two systems,

$$\Delta F = W^{rev} = W^s - E^d. \quad (12)$$

E^d is the energy dissipation in the switching process, which can be obtained as the difference between the forward and reverse switching. The non-equilibrium approach can be used to efficiently calculate the free energy of the system of interest at a given thermodynamic condition (P, T). Once a free energy $F(P, T)$ is known the free energy as a function of temperature over a given range from T to T_f can be obtained in a single calculation using the reversible scaling approach⁸⁴. These approaches, and associated algorithms have been discussed in more detail in ref. 34.

pyiron

`pyiron` is a workflow framework for atomistic simulation, focused on rapid prototyping and up-scaling simulation protocols. Based on an object-oriented approach, the individual components of a simulation protocol in `pyiron` are combined like building blocks. Each `pyiron` object is connected to the jupyter-based user-interface, the data storage interface which combines a

structured database (SQL) and a hierarchical file format (HDF5) as well as the resource interface to connect to computing resources and parameter databases. By implementing the potential fitting codes (`atomicrex`, `RunNER` and `pacemaker`), the simulation codes (`LAMMPS` and `VASP`) and the thermodynamics code (`calphy`) based on the same job class, the technical complexity of executing the underlying codes is greatly reduced. In addition to the three classes of interatomic potentials discussed here, `pyiron` can also be used for parametrising Moment Tensor Potentials⁸⁵, in addition to Angular Dependent Potential⁸⁶ and Tersoff⁸⁷ classical interatomic potentials.

As a first step of the simulation protocol a new project is initialized: `pr = Project("ALLi")`. The project object is represented as a folder on the file system and all calculations in this project are going to be executed in this folder. From the project object the individual job objects are created using the factoring pattern:

```
job = pr.create.job.SimulationCode('job_name')
```

The factoring pattern, which refers to using one object to create objects of different types, has two advantages: On the one hand it allows the users to use auto-completion in selecting the new object to create and on the other hand the newly created object can be already initialized with information of the object it is created from. In this case the job object receives its storage location from the project object it was created from. The individual job classes for the VASP DFT code, the different fitting codes and `calphy` and `LAMMPS` for validation are introduced below.

PyXtal

For the generation of random crystal structures we have wrapped the python code `PyXtal` in the `structuretoolkit` module distributed with `pyiron`.

```
import structuretoolkit.build.random as stkr
al_li_structures = stkr.pyxtal(
    group=[227, 194],
    species=["Al", "Li"],
    num_ions=[4, 4],
    repeat=10
)
```

would generate a list of ten structures each of the spacegroups 227 and 194 with stoichiometry Al_4Li_4 . More advanced options as document by the `PyXtal` library itself, can be passed to the function as well.

VASP

Starting with the Vienna Ab initio Simulation Package (VASP)^{40–42}, the job object is created from the project object using the factoring pattern and an atomic structure in the `Atoms` format defined by the Atomic Simulation Environment (ASE) is assigned:

```
job = pr.create.job.Vasp("job_name")
job.structure = structure
```

In addition to the atomic structure also the input parameters which determine the precision of the DFT calculation can be specified directly through the `pyiron` python interface. For this `pyiron` provides two interfaces, first the generic interface which is independent of the specific simulation code and second the code-specific interface, which allows users already experienced with a specific simulation code to directly modify specific input parameters. Using the generic interface the plane wave energy cutoff is set to 750 eV, the k-point density is set to 0.1 \AA^{-1} and the level of electronic convergence is defined as 10^{-8} eV:

```
job.set_encut(750.0)
job.set_kpoints(k_mesh_spacing=0.1)
job.set_convergence_precision(
    electronic_energy=1.0e-8,
)
job.set_occupancy_smearing(
    smearing="FermiDirac",
    width=0.2,
)
```

The advantage of using the generic interface is that the users can switch between different DFT simulation codes by only changing the create job function call `pr.create.job.Vasp()`, the rest of the commands remain the same. For expert users `pyiron` also provides the option to access the simulation code specific input directly. As an example, while the electronic smearing can be specified using the generic `set_occupancy_smearing()` function, it can also be modified based on the VASP specific input file named INCAR, which can be accessed in `pyiron` like a python dictionary:

```
job.input.incar["ISMEAR"] = -1
job.input.incar["SIGMA"] = 0.1
```

Finally, in addition to the simulation code-specific parameters the `pyiron` job object also provides the option to specify the submission to the high performance computing (HPC) queuing system:

```
job.server.queue = "gpu_queue"
job.server.cores = 4
job.server.gpus = 4
```

After the specification of the input parameters and resource assignment is completed the `pyiron` job object can be executed using the `run()` function. This triggers the internal cycle of writing the input files, submitting the calculation to the HPC for execution and once the calculation is completed parse the output files to provide the output to the `pyiron` python interface.

TrainingContainer

Following the execution of the DFT calculations, the next step is the aggregation of the outputs of these calculation to provide them to the fitting codes for the interatomic potentials. In `pyiron` this is achieved by combining two objects, the `pyirontable` object and the `TrainingContainer`. The `pyirontable` object specifies a series of functions which are applied to each job object in a given `pyiron` project, following a map-reduce pattern:

```
table = pr.create.table()
table.add.get_job_name
table.add.get_structure
table.add.get_energy_tot
table.add.get_forces
table.run()
```

The aggregated data, which is returned as a pandas DataFrame object is then stored in the `TrainingContainer` for reference in the fitting codes:

```
tr = pr.create.job.TrainingContainer("tc_job")
tr.include_dataset(table.get_dataframe())
```

Additionally, the class defines common plotting that make the creation of graphs such as Figs. 6 and 4 easier, for example,

```
tr.plot.energy_volume()
```

atomicrex

The `atomicrex` interface exposes the full functionality of the code⁴⁸ in a `pyiron` python interface, while storing relevant inputs and output necessary to reproduce fitting processes. An `atomicrex` job object can be created with

```
job = pr.create.job.Atomicrex("AtomicrexJob")
```

Currently `atomicrex` implements EAM, modified EAM³⁰, angular dependent⁴⁸, analytic bond order⁴⁹ and Tersoff⁵⁷ potentials. They can be set using

```
pot = job.factories.potentials.potential_type()
```

For potentials that allow for different functional forms like EAM potentials it is necessary to define these functions. Here the user can choose between predefined functions and own creations via a math parser.

```
morse = job.factories.functions.morse_B(
    identifier="V",
    D0=0.05,
    r0=2.5,
    beta=2.2,
```

```
S=2.4,
    delta=0.0,
    species=["Al", "Li"])
uf = ref.factories.functions.user_function(
    identifier="UserElement1Element2",
    input_variable="r")
uf.expression = "A*exp(r0-r)"
uf.derivative = "-A*exp(r0-r)"
uf.parameters.add_parameter("A", 3)
uf.parameters.add_parameter("r0", 5)
pot.pair_interactions[morse.identifier] = morse
pot.electron_densities[uf.identifier] = uf
Structures and corresponding fit properties can be directly assigned using the general TrainingContainer interface. If fine grained control over weights is required they can also be added one by one:
s = pr.create.structure.ase.bulk("Al")
job.structures.add_structure(s,
    identifier="SomeStructure",
    relative_weight=10000)
job.structures.add_scalar_fit_property(
    "atomic-energy",
    target_val=-4.0,
    relative_weight=100,
```

Nearly arbitrary parameter constraints can be added using math parser expressions:

```
job.input.parameter_constraint.add_constraint(
    identifier="SomeConstraint",
    dependent_dof="constrainedParameter",
    expression="MathparserExpression",
)
```

Finally, the user can choose between an internal LBFGS minimizer and a plethora of optimization algorithms provided via the NLOpt library⁵⁰ to fit the potential.

```
algo = job.factories.algorithms.some_algo(
    max_iter=1000)
job.input.fit_algorithm = algo
```

RunNer

Training with `RunNer` usually passes through three stages: in mode 1, the values of the atom-centred symmetry functions for the whole training dataset are calculated and stored to disk, and the data is separated into a training and a test set. mode 2 optimizes the parameters of the HDNNP in order to represent best the reference energies and forces. Finally, mode 3 is used to predict the properties of unknown configurations.

The `pyiron` job `RunNerFit` reflects these steps. Similar to the other training jobs, it is created by invoking the `create` routine of a `pyironProject` object. Every `RunNerFit` job also requires the specification of a training dataset.

```
model = proj.create.job.RunnerFit('model')
model.add_training_data(dataset)
```

In the next step, a set of atom-centred symmetry functions must be parameterized for the training dataset. `runnerase` offers the procedure `generate_symmetryfunctions` to help with this task. Afterwards, the job can be started using the `run` command:

```
sfs = generate_symmetryfunctions(dataset,
    sftype=2,
    cutoff=12.0)
model.parameters.symfunction_short += sfs
model.run()
```

After the successful termination of mode 1, mode 2 is started by reloading the first job and altering the setting `runner_mode`. This tells the underlying RuNNer code how to operate:

```
mode2 = mode1.restart('mode2')
mode2.parameters.runner_mode = 2
mode2.run()
```

The same procedure is followed to run mode 3. In a RuNNerFit, the execution of mode 3 is mandatory to complete training and obtain a full prediction of both the train and test datasets:

```
mode3 = mode2.restart('mode3')
mode3.parameters.runner_mode = 3
mode3.run()
```

In order to use the trained potential in an application with LAMMPS, one can call the `get_lammps_potential` routine which returns the required `pair_style` and `pair_coeff` commands. The HDNNP pair style is part of the LAMMPS interface provided by the `n2p2` package⁹¹:

```
mode3.get_lammps_potential()
```

pacemaker

In order to setup the pacemaker job, one needs to create the corresponding pyiron job and add the training dataset.

```
job = pr.create.job.PacemakerJob("pacemaker_
job")
job.add_training_data(dataset)
```

Parameters for ACE parameterization procedure will be initialized to their defaults. However, one always can configure all of them. For example, setting energy-force weights balance (κ from Ref. 44) as

```
job.input['fit']['loss']['kappa']=0.3
```

After that one can run the job and get the LAMMPS potential as well.

calphy

The computational approaches to obtain free energies as discussed in Section III F consists of multiple interdependent steps, and presents a complex computational workflow. In order to facilitate a user to easily calculate the free energies, and at the same time retain the ability to tune each step in the workflow as needed, we developed `calphy`³⁴, a python library for automated calculation of free energies. It uses LAMMPS as the molecular dynamics driver to perform free energy calculations in an automated manner. `calphy` when combined with `pyiron`, can leverage additional features such as interoperability with other common atomistic simulation tools, scaling to HPC systems, and job and data management.

Within `pyiron`, a non-equilibrium free calculation, for example an Al fcc lattice at 500 K and 0 pressure can be carried out by the following code:

```
pr = Project("free_energy")
job = pr.create.job.Calphy("Al_fcc_500")
job.structure = pr.create.structure.ase.bulk
("Al",
 cubic=True).repeat(4)
job.potential = "Al-atomicrex"
job.calc_free_energy(
 temperature=500,
 pressure=0,
 reference_phase="solid",
 n_equilibration_steps=25000,
 n_switching_steps=50000,
)
job.run()
```

The main inputs needed are the input structure and the interatomic potential, apart from the thermodynamic conditions at which the calculation is to be performed. For calculating the free energy of a liquid system, the only change needed is `reference_phase='liquid'`. `calphy` automatically uses a different reference system based on this command. To obtain free energies over a given temperature range, one needs to change the temperature

option: `temperature=[500, 800]`. In this case, a free energy calculation at 500 K is performed first, followed by a temperature integration up to 800 K in another calculation.

LAMMPS

Beyond the free energies calculated with `calphy` to construct the phase diagram, the LAMMPS molecular dynamics simulation code is used to validate material properties calculated with the individual machine learning potentials. In `pyiron` the workflows to calculate the material properties is defined independent of the specific simulation code, so in the first step a reference LAMMPS job is defined for the interatomic potential fitted with the `atomicrex` fitting code:

```
pr = Project('validation')
job = pr.create.job.Lammps('lmp')
job.structure = structure
job_lmp.potential = 'Al-atomicrex'
```

Following the definition of the reference job the next step is assigning this reference job to the workflow to calculate a material property, in this case the calculation of the elastic constants with the `ElasticMatrix` job:

```
elastic = pr.create.job.ElasticMatrix('elmat')
elastic.ref_job = job_lmp
elastic.run()
```

By defining the calculation of the material properties independent of the simulation code, the same validation calculation can be applied for the LAMMPS molecular dynamics simulation code to test the fitted interatomic potentials as well as the VASP DFT simulation code, to enable a direct comparison.

Software availability

The software used in this paper, `pyiron`, `pacemaker`, `RuNNer`, `atomicrex`, `calphy`, `LAMMPS`, `pycalphad`, and `PyXtal` are freely available from their respective repositories. A list of the software tools, along with their repositories and documentation is provided in the Supplementary Table 4. Exemplary workflows to illustrate the calculations mentioned in this manuscript, along with the software versions required, are available in an online repository⁹².

Data availability

The dataset used for parametrization of the interatomic potentials, along with the free energy values for the construction of the phase diagram are also made available⁹³. The LAMMPS compatible interatomic potential files are available in both⁹² and⁹³.

Received: 10 March 2024; Accepted: 12 October 2024;

Published online: 17 November 2024

References

- Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **145**, 170901 (2016).
- Deringer, V. L., Caro, M. A. & Csányi, G. Machine learning interatomic potentials as emerging tools for materials science. *Adv. Mater.* **31**, 1902765 (2019).
- Unke, O. T. et al. Machine learning force fields. *Chem. Rev.* **121**, 10142 (2021).
- Friederich, P., Häse, F., Proppe, J. & Aspuru-Guzik, A. Machine-learned potentials for next-generation matter simulations. *Nat. Mater.* **20**, 750 (2021).
- Behler, J. & Csányi, G. Machine learning potentials for extended systems - a perspective. *Eur. Phys. J. B* **94**, 142 (2021).
- Lysogorskiy, Y. et al. Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon. *Npj Comput. Mater.* **7**, 1 (2021).
- Tokita, A. M. & Behler, J. How to train a neural network potential. *J. Chem. Phys.* **159**, 121501 (2023).

8. Huber, S. P. et al. Aiida 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *Sci. Data* **7**, 300 (2020).
9. Mathew, K. et al. Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Comput. Mater. Sci* **139**, 140 (2017).
10. Gjerding, M. et al. Atomic simulation recipes: A python framework and library for automated workflows. *Comput. Mater. Sci* **199**, 110731 (2021).
11. Janssen, J. et al. pyiron: An integrated development environment for computational materials science. *Comput. Mater. Sci* **163**, 24 (2019).
12. Duff, A. I., Sakidja, R., Walker, H. C., Ewings, R. A. & Voneshen, D. Automated potential development workflow: Application to bazro3. *Comput. Phys. Commun.* **293**, 108896 (2023).
13. Zeng, J. et al. DeePMD-kit v2: A software package for deep potential models. *J. Chem. Phys.* **159**, 054801 (2023).
14. Rohskopf, A. et al. FitSNAP: Atomistic machine learning with LAMMPS. *J. Open Source Soft.* **8**, 5118 (2023).
15. Vandenhoute, S., Cools-Ceuppens, M., DeKeyser, S., Verstraelen, T. & Van Speybroeck, V. Machine learning potentials for metal-organic frameworks using an incremental learning approach. *Npj Comput. Mater.* **9**, 19 (2023).
16. Gelžinytė, E. et al. wfl Python toolkit for creating machine learning interatomic potentials and related atomistic simulation workflows. *J. Chem. Phys.* **159**, 124801 (2023).
17. Wen, M., Afshar, Y., Elliott, R. S. & Tadmor, E. B. KLIF: A framework to develop physics-based and machine learning interatomic potentials. *Comput. Phys. Commun.* **272**, 108218 (2022).
18. Kratzer, P. & Neugebauer, J. The basics of electronic structure theory for periodic systems. *Front. Chem.* **7**, 106 (2019).
19. Bosoni, E. et al. How to verify the precision of density-functional-theory implementations via reproducible and universal workflows. *Nat. Rev. Phys.* **6**, 45 (2023).
20. Becker, C. A., Tavazza, F., Trautt, Z. T. & Buarque de Macedo, R. A. Considerations for choosing and using force fields and interatomic potentials in materials science and engineering. *Curr. Opin. Solid State Mater. Sci.* **17**, 277 (2013).
21. Hale, L. M., Trautt, Z. T. & Becker, C. A. Evaluating variability with atomistic simulations: the effect of potential and calculation methodology on the modeling of lattice and elastic constants. *Modelling Simul. Mater. Sci. Eng.* **26**, 055003 (2018).
22. Lysogorskiy, Y., Hammerschmidt, T., Janssen, J., Neugebauer, J. & Drautz, R. Transferability of interatomic potentials for molybdenum and silicon. *Modelling Simul. Mater. Sci. Eng.* **27**, 025007 (2019).
23. Zuo, Y. et al. A performance and cost assessment of machine learning interatomic potentials. *J. Phys. Chem. A* **124**, 731 (2020).
24. Abd El-Aty, A. et al. Strengthening mechanisms, deformation behavior, and anisotropic mechanical properties of Al-Li alloys: A review. *J. Adv. Res.* **10**, 49 (2018).
25. Hallstedt, B. & Kim, O. Thermodynamic assessment of the Al-Li system. *Int. J. Mater. Res.* **98**, 961 (2007).
26. Gupta, R., Nayan, N., Nagasireesha, G. & Sharma, S. Development and characterization of Al-Li alloys. *Mater. Sci. Eng. A* **420**, 228 (2006).
27. Rioja, R. J. Fabrication methods to manufacture isotropic Al-Li alloys and products for space and aerospace applications. *Mater. Sci. Eng. A* **257**, 100 (1998).
28. Liu, Y. & Mo, Y. Assessing the accuracy of machine learning interatomic potentials in predicting the elemental orderings: A case study of li-al alloys. *Acta Mater.* **268**, 119742 (2024).
29. Daw, M. S. & Baskes, M. I. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* **29**, 6443 (1984).
30. Baskes, M. I. Application of the Embedded-Atom Method to Covalent Materials: A Semiempirical Potential for Silicon. *Phys. Rev. Lett.* **59**, 2666 (1987).
31. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
32. Behler, J. Four generations of high-dimensional neural network potentials. *Chem. Rev.* **121**, 10037 (2021).
33. Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104 (2019).
34. Menon, S., Lysogorskiy, Y., Rogal, J. & Drautz, R. Automated free-energy calculation from atomistic simulations. *Phys. Rev. Mater.* **5**, 103801 (2021).
35. Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
36. Chue Hong, N. P. et al. FAIR Principles for Research Software (FAIR4RS Principles), <https://doi.org/10.15497/RDA00068> (2021).
37. Hjorth Larsen, A. et al. The atomic simulation environment—a Python library for working with atoms. *J. Phys. Condens. Matter* **29**, 273002 (2017).
38. Ong, S. P. et al. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci* **68**, 314 (2013).
39. Fredericks, S., Parrish, K., Sayre, D. & Zhu, Q. PyXtal: A Python library for crystal structure generation and symmetry analysis. *Comput. Phys. Commun.* **261**, 107810 (2021).
40. Kresse, G. & Hafner, J. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B* **47**, 558 (1993).
41. Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15 (1996).
42. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169 (1996).
43. Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **117**, 1 (1995).
44. Bochkarev, A. et al. Efficient parametrization of the atomic cluster expansion. *Phys. Rev. Mater.* **6**, 013804 (2022).
45. Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quantum Chem.* **115**, 1032 (2015).
46. Behler, J. First principles neural network potentials for reactive simulations of large molecular and condensed systems. *Angew. Chem. Int. Ed.* **56**, 12828 (2017).
47. Knoll, A. & Behler, J. runnerase: An interface between the runner neural network energy representation (runner) and the atomic simulation environment (ase), <https://runner-suite.gitlab.io/runnerase/1.0.2> (2021).
48. Stukowski, A., Fransson, E., Mock, M. & Erhart, P. Atomicrex—a general purpose tool for the construction of atomic interaction models. *Model. Simul. Mat. Sci. Eng.* **25**, 055003 (2017).
49. Togo, A., Chaput, L., Tadano, T. & Tanaka, I. Implementation strategies in phonopy and phono3py. *J. Phys. Condens. Matter* **35**, 353001 (2023).
50. Kresse, G. & Joubert, D. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B* **59**, 1758 (1999).
51. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865 (1996).
52. Jain, A. et al. The Materials Project: A materials genome approach to accelerating materials innovation. *APL Mater.* **1**, 011002 (2013).
53. Poul, M., Huber, L., Bitzek, E. & Neugebauer, J. Systematic atomic structure datasets for machine learning potentials: Application to defects in magnesium. *Phys. Rev. B* **107**, 104103 (2023).
54. Lysogorskiy, Y., Bochkarev, A., Mrovec, M. & Drautz, R. Active learning strategies for atomic cluster expansion models. *Phys. Rev. Lett.* **7**, 043801 (2023).
55. Pickard, C. J. & Needs, R. Ab initio random structure searching. *J. Phys.: Condens. Matter* **23**, 053201 (2011).

56. Yanxon, H., Zagaceta, D., Wood, B. C. & Zhu, Q. Neural network potential from bispectrum components: A case study on crystalline silicon. *J. Chem. Phys.* **153**, 054118 (2020).
57. Mishin, Y., Mehl, M. J., Papaconstantopoulos, D. A., Voter, A. F. & Kress, J. D. Structural stability and lattice defects in copper: *Ab initio*, tight-binding, and embedded-atom calculations. *Phys. Rev. B* **63**, 224106 (2001).
58. Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **134**, 074106 (2011).
59. Kalman, R. E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **82**, 35 (1960).
60. Kaufman, L. & Cohen, M. The Martensitic Transformation in the Iron-Nickel System. *JOM* **8**, 1393 (1956).
61. Vega, C., Sanz, E., Abascal, J. L. F. & Noya, E. G. Determination of phase diagrams via computer simulation: methodology and applications to water, electrolytes and proteins. *J. Phys.: Condens. Matter* **20**, 153101 (2008).
62. Chew, P. Y. & Reinhardt, A. Phase diagrams-Why they matter and how to predict them. *J. Chem. Phys.* **158**, 030902 (2023).
63. Opitz, A. Molecular dynamics investigation of a free surface of liquid argon. *Phys. Lett., A* **47**, 439 (1974).
64. Ladd, A. & Woodcock, L. Triple-point coexistence properties of the Lennard-Jones system. *Chem. Phys. Lett.* **51**, 155 (1977).
65. Kranendonk, W. & Frenkel, D. Computer simulation of solid-liquid coexistence in binary hard sphere mixtures. *Mol. Phys.* **72**, 679 (1991).
66. Frenkel, D. & Smit, B. Chapter 7 - free energy calculations. In *Understanding Molecular Simulation*, 2nd ed. (eds Frenkel, D. & Smit, B.) 167–200 (Academic Press, 2002).
67. Kirkwood, J. G. Statistical Mechanics of Fluid Mixtures. *J. Chem. Phys.* **3**, 300 (1935).
68. Frenkel, D. & Smit, B. *Understanding Molecular Simulation*, 2nd ed. (Academic Press, Inc., 2001).
69. Gayle, F. W., Vander Sande, J. B. & McAlister, A. J. The Al-Li (Aluminum-Lithium) system. *Bull. Alloy Phase Diagr.* **5**, 19 (1984).
70. Lide, D. R. *CRC handbook of chemistry and physics*, Vol. 85 (CRC press, 2004).
71. Vočadlo, L. & Alfè, D. *Ab initio* melting curve of the fcc phase of aluminum. *Phys. Rev. B* **65**, 214105 (2002).
72. Otis, R. & Liu, Z.-K. pycalphad: CALPHAD-based Computational Thermodynamics in Python. *J. Open Res. Softw.* **5**, 1 (2017).
73. Wang, P., Du, Y. & Liu, S. Thermodynamic optimization of the Li-Mg and Al-Li-Mg systems. *Calphad* **35**, 523 (2011).
74. Zhu, L.-F., Grabowski, B. & Neugebauer, J. Efficient approach to compute melting properties fully from *ab initio* with application to Cu. *Phys. Rev. B* **96**, 224202 (2017).
75. Zhu, L.-F., Körmann, F., Ruban, A. V., Neugebauer, J. & Grabowski, B. Performance of the standard exchange-correlation functionals in predicting melting properties fully from first principles: Application to Al and magnetic Ni. *Phys. Rev. B* **101**, 144108 (2020).
76. Dörner, F., Sukurma, Z., Dellago, C. & Kresse, G. Melting si: Beyond density functional theory. *Phys. Rev. Lett.* **121**, 195701 (2018).
77. Kishio, K. & Brittain, J. Defect structure of β -LiAl. *J. Phys. Chem. Solids* **40**, 933 (1979).
78. Shao, W., Liu, S. & LLorca, J. First principles prediction of the Al-Li phase diagram including configurational and vibrational entropic contributions. *Comput. Mater. Sci.* **217**, 111898 (2023).
79. Jacobsen, K. W., Norskov, J. K. & Puska, M. J. Interatomic interactions in the effective-medium theory. *Phys. Rev. B* **35**, 7423 (1987).
80. Finnis, M. W. & Sinclair, J. E. A simple empirical N -body potential for transition metals. *Phil. Mag. A* **50**, 45 (1984).
81. Frenkel, D. & Ladd, A. J. C. New Monte Carlo method to compute the free energy of arbitrary solids. Application to the fcc and hcp phases of hard spheres. *J. Chem. Phys.* **81**, 3188 (1984).
82. Paula Leite, R., Freitas, R., Azevedo, R. & de Koning, M. The Uhlenbeck-Ford model: Exact virial coefficients and application as a reference system in fluid-phase free-energy calculations. *J. Chem. Phys.* **145**, 194101 (2016).
83. Watanabe, M. & Reinhardt, W. Direct dynamical calculation of entropy and free energy by adiabatic switching. *Phys. Rev. Lett.* **65**, 3301 (1990).
84. de Koning, M., Antonelli, A. & Yip, S. Optimized Free-Energy Evaluation Using a Single Reversible-Scaling Simulation. *Phys. Rev. Lett.* **83**, 3973 (1999).
85. Novikov, I. S., Gubaev, K., Podryabinkin, E. V. & Shapeev, A. V. The MLIP package: moment tensor potentials with MPI and active learning. *Mach. Learn. Sci. Technol.* **2**, 025002 (2021).
86. Mishin, Y., Mehl, M. & Papaconstantopoulos, D. Phase stability in the fe-ni system: Investigation by first-principles calculations and atomistic simulations. *Acta Mater.* **53**, 4029 (2005).
87. Tersoff, J. Empirical interatomic potential for silicon with improved elastic properties. *Phys. Rev. B* **38**, 9902 (1988).
88. Mishin, Y., Mehl, M. J. & Papaconstantopoulos, D. A. Phase stability in the Fe-Ni system: Investigation by first-principles calculations and atomistic simulations. *Acta Mater.* **53**, 4029 (2005).
89. Brenner, D. W. Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films. *Phys. Rev. B* **42**, 9458 (1990).
90. Johnson, S. G. The NLOpt nonlinear-optimization package, <https://github.com/stevengj/nlopt> (2007).
91. Singraber, A., Behler, J. & Dellago, C. Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials. *J. Chem. Theory and Comput.* **15**, 1827 (2019).
92. Menon, S. et al. Workflows for “From electrons to phase diagrams with classical and machine learning potentials: automated workflows for materials science with pyiron”. *Zenodo*, <https://doi.org/10.5281/zenodo.12073981> (2024).
93. Menon, S. et al. From electrons to phase diagrams with classical and machine learning potentials: automated workflows for materials science with pyiron, <https://doi.org/10.17617/3.VKQ3ZM> (2024).
94. Hänström, A. & Lazor, P. High pressure melting and equation of state of aluminium. *J. Alloys Compd.* **305**, 209 (2000).

Acknowledgements

The workflows, potentials, and results presented here were obtained in the framework of the POTENTIALS collaboration and scientific network “Assessment of atomistic simulations” with funding from the German Science Foundation (DFG) (grant number 405602047). Furthermore, a workshop on the subject of this manuscript at which participants could interactively execute and explore the initial versions of these workflows was held in June 2022. S. M. acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure - NFDI 38/1 - project number 460247524. J. B. acknowledges funding by the DFG (project number 405479457 as part of PAK 965/1). A. K. acknowledges funding by the Studienstiftung des Deutschen Volkes (doctoral scholarship). N. L. and J. R. acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant number 405621137. K. A. acknowledges funding from the the DFG under grant number 405621160. M. M. and R. D. acknowledge funding by the German Science Foundation (DFG), projects 405621081 and 405621217. R.D. and Y. L. acknowledge computation time by Center for Interface-Dominated High Performance Materials (ZGH) at Ruhr-Universität Bochum, Germany. J.J. and J.N. acknowledge funding by the DFG under grant number 405621217. M.P. and J.N. acknowledge funding from the DFG under grant number 405621160.

Author contributions

Conceptualization: K.A., J.B., R.D., J.N. Data curation, Formal analysis, and Investigation: S.M., Y.L., A.K., N.L., M.P., M.Q. Methodology: S.M., Y.L.,

M.P., J.B., R.D., J.N. Software: S.M., Y.L., A.K., N.L., M.P., M.Q., J.J. Validation: S.M., Y.L., A.K., N.L., M.P., M.Q. Visualization: S.M., Y.L., A.K., M.P. Writing - original draft: S.M., Y.L., A.K., N.L., M.P., M.Q., J.J., M.M., R.D., J.N. Writing - review & editing: S.M., Y.L., A.K., N.L., M.P., M.Q., J.J., M.M., J.R., K.A., J.B., R.D., J.N. These author contributions are defined according to the CRediT contributor roles taxonomy.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

J.N. is an associate editor for npj Computational Materials. The other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41524-024-01441-0>.

Correspondence and requests for materials should be addressed to Sarath Menon or Jörg Neugebauer.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024