

Enhancing the Quality and Reliability of Machine Learning Interatomic Potentials through Better Reporting Practices

Tristan Maxson[†], Ademola Soyemi[†], Benjamin W. J. Chen[‡], Tibor Szilvási^{†}*

[†]Department of Chemical and Biological Engineering, University of Alabama, Tuscaloosa, AL 35487, United States

*[‡]Institute of High Performance Computing (IHPC), Agency for Science, Technology, and Research (A*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore*

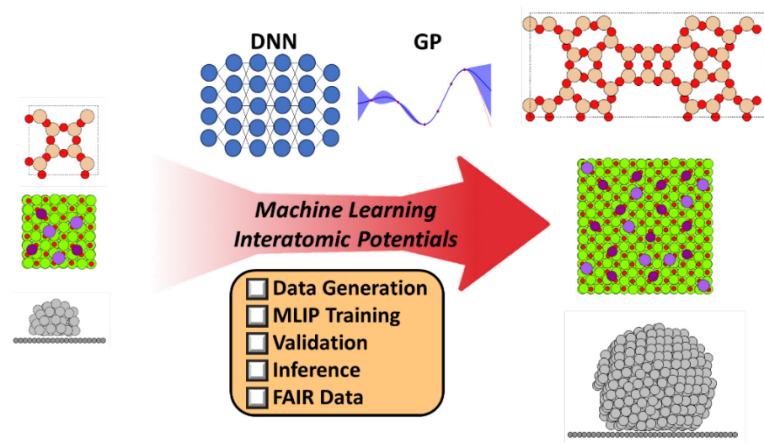
**Corresponding author. Email: tibor.szilvasi@ua.edu*

Abstract

Recent developments in machine learning interatomic potentials (MLIPs) have empowered even non-experts in machine learning to train MLIPs for accelerating materials simulations. However, the current literature lacks clear standards for documenting the use of MLIPs, which hinders the reproducibility and independent evaluation of the presented results. In this perspective, we aim to provide guidance on best practices for documenting MLIP use while walking the reader through the development and deployment of MLIPs including hardware and software requirements,

generating training data, training models, validating predictions, and MLIP inference. We also suggest useful plotting practices and analyses to validate and boost confidence in the deployed models. Finally, we provide a step-by-step checklist for practitioners to use directly before publication to standardize the information to be reported. Overall, we hope that our work will encourage reliable and reproducible use of these MLIPs, which will accelerate their ability to make a positive impact in various disciplines including materials science, chemistry, and biology, among others.

TOC Graphic



1. Introduction

Computational analysis is an indispensable tool for understanding the behavior of materials. Such analysis has led to crucial insight in various fields such as catalysis¹⁻³, bio-⁴⁻⁶, nano-⁶⁻⁸, and soft^{9, 10} materials among others, even enabling the computation-led design of materials in some cases. First-principles methods based on density functional theory (DFT) have been adopted by most communities to gain insights into materials properties due to a good balance between accuracy and computational cost. However, due to computational constraints, DFT is typically limited to simulating systems up to hundreds of atoms^{11, 12} and time scales of up to hundreds of picoseconds^{13, 14}.

Recently, machine learning interatomic potentials (MLIPs) that are trained on DFT data have emerged as a promising method to expand the time and length scale of atomistic simulations while potentially maintaining DFT-like accuracy^{15, 16}. Many ready-to-use software packages¹⁷⁻²⁸ are now available for prospective practitioners to download, install, and use within hours. The relative ease of using such packages has generated considerable interest, resulting in a rapid increase in scientific applications such as accelerating molecular dynamics^{22, 29, 30}, probing chemical reactivity³¹⁻³³, and investigating long-time- and length-scale phase transitions³⁴⁻³⁶. Some MLIPs may now be extended to provide estimates on other properties as well such as dipole moments²⁵, charge distribution³⁷, the density of states^{38, 39}, magnetic moments of atoms^{40, 41}, and the local electron density around an atom⁴².

Due to this rapid growth in the field of MLIPs, standards for documenting the use of MLIPs have yet to be developed by the community, leading to inconsistencies in reporting on the developed MLIPs. Specifically, the materials modeling community has not reached a consensus on how to ensure reproducible accurate⁴³ use of MLIPs similar to what has been achieved for DFT-based simulations^{44, 45}. As a result, widely different standards of reporting can be found in the literature ranging from merely mentioning the use of MLIPs to open-sourcing all data and scripts⁴⁶. As MLIPs are still new and evolving, such differences are understandable. Nevertheless, their reliability can vary widely depending on MLIP construction and training procedures. Reproducing results from literature may be difficult even for experienced researchers. For example, simple physical interactions such as nuclear repulsion are not always included in models and atoms may collapse together unphysically. Improper training procedures may also overfit certain properties such as energy at the cost of forces, which may lead to unphysical dynamics. Thus, considering that current MLIPs are typically not robust constructions, there is a thin line between useful and

misleading results. There is an urgent need for community to document the usage and validation of MLIPs with greater care to avoid misleading studies and over-interpreting results when there might be considerable uncertainty.

In this perspective, we aim to provide the community with guidelines for documenting the development, validation, and application of MLIPs (Figure 1) similar to previous works that establish guidelines for experimental and computational subfields⁴⁷⁻⁴⁹. Specifically, we highlight different aspects of MLIPs that should be reported: hardware and software requirements, generation of training data, training of models, validation of predictions, and inference of MLIPs. We discuss the validation and inference of MLIPs with a focus on predicting energy, force, and stress as these are currently the most common prediction types in literature. In addition, we provide a detailed checklist that MLIP users can apply to self-review manuscripts before submission. Proper documentation of such information will foster greater trust in computational analyses and increase reproducibility. Promoting direct data availability will also accelerate the standardized development of MLIPs. Ultimately, developing better standards for detailing MLIP usage will help increase the adoption of MLIPs and allow the community to reap its full benefits.

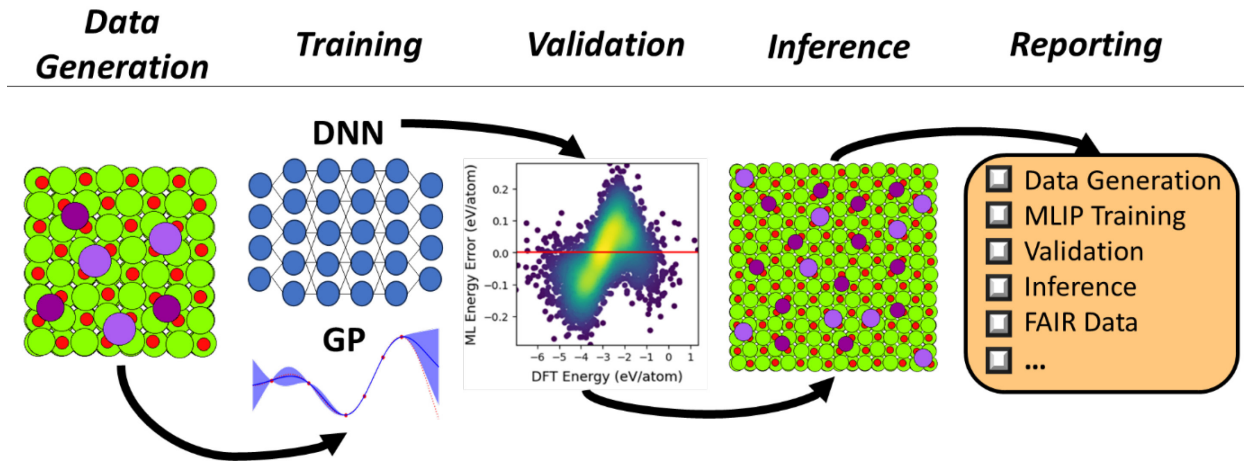


Figure 1. General workflow of data generation, training, validating, and deploying MLIPs for use in scientific research. Reporting is included as a final stage which is often overlooked in works utilizing MLIPs and is the focus of this perspective.

2. Best Practices for Reporting

2.1. Reporting Checklist

Based on the guidelines that will be discussed in this perspective, we have provided a convenient and comprehensive checklist of the process of documenting various aspects of MLIP usage, including the choice of software and hardware, training data generation, training, validation, and inference as seen in Figure 1. This checklist is provided in the Supporting Information (SI) for practitioners to go through before publication to help ensure reproducible research. The checklist also guides reviewers on what types of data should be present in submitted manuscripts, or at minimum aids in looking out for potentially missing information. An up-to-date version of the checklist may be found in the GitLab repository at <https://gitlab.com/szilvasi-group/mlip-reporting-checklist>. We encourage the community to make suggestions in the future as the development of MLIPs continues and new MLIP types become commonplace. The checklist may also be found in different formats if groups wish to improve the general checklist. We envision the checklist as a living document that will be maintained by the community to follow the standards of the field and the needs of researchers. We encourage MLIP developers to make pull requests for code-specific checklists if they wish to establish a more detailed standard for their own codes.

2.2. Software and Hardware

A range of software and hardware may be utilized by researchers. For software, we suggest reporting all aspects of the environment in which MLIPs are trained and used, such as the versions of the MLIP software package, Python, PyTorch/TensorFlow, CUDA, and the specifications of the used CPU/GPU. If an environment manager such as Conda or Python's venv is used, a versioned list of all packages installed may help track down potentially conflicting packages and allow others to reproduce the same environment.

It is also useful to note the type of hardware used as GPUs tend to be limited by memory requirements imposed by the size of the model. For example, an H100 GPU may have 80 or 188 GB while an A100 GPU may have 40 or 80 GB of memory. We suggest reporting detailed information regarding the GPU, CPU, memory, and parallelization to ensure that readers understand the viability of running the model.

2.3. Training Data

Training data for MLIPs may be generated and/or selected by a diverse range of methods. In this section, we detail various data generation strategies and their critical parameters to report. Training data typically consists of atomic configurations calculated with DFT, although any ground truth may be considered as well. The used atomistic data should be reported by providing the data source, version number, and access date that may be obtained from public repositories designed for MLIP development such as RevMD17⁵⁰, IonSolvR⁵⁰, and the Open Catalyst Project^{51, 52}. Such repositories provide either the raw calculation files or the configurations in a compact format (xyz, npy, traj) including atomic positions, energy, forces, and stresses. If only a portion of the configurations were used, the subset of data used should be stated.

Existing molecular dynamics (MD) runs from the literature that were not originally intended for machine learning can also be possible sources of training data. The reuse of literature data can help may accelerate research projects in their initial stages, highlighting how the increased data availability may have positive effects beyond reproducibility. However, when no appropriate dataset is available for the specific use case, researchers are required to generate their training data since current MLIPs cannot typically generalize from one system to another unless there is a match in composition and conditions such as temperature and pressure. If this is so, the exact methods for generating new datasets should be rigorously documented.

This is especially so as new methods for data generation can advance the field and their value should not be underestimated by the researchers generating the training data and MLIPs⁵³. While it is understandable that groups may have in-house tools that are not suitable for public sharing, the general methodology must be reported and is not restricted to the suggestions that follow in this section. Instead, we provide guidance on common methods and highlight critical parameters needed to reproduce the method. It is important to note that methods of generating training data should be viewed with equal value and disclosed as their reproducibility is not guaranteed by reporting the raw data.

When generating training data, it is helpful to start with simple and efficient methods, which usually involve enumerating different configurations of the system and evaluating them with single-point calculations. Yet, such methods can be surprisingly difficult to reproduce as critical details are often omitted. Three common methods for generating suitable atomic configurations involve random displacements^{54,55}, phonon-like displacements⁵⁶⁻⁵⁸, or combinatorial methods⁵⁹⁻⁶⁴. Random displacements, or structure rattling, perturb atomic positions to provide higher energy/force/stress structures from an optimized structure. Displaced structures provide a definition to the local PES but must be performed carefully and filtered to avoid generating unphysical configurations. Phonon displacements are similar to random displacements but are more physically motivated as they represent movements that may occur in MD simulations without directly performing them. Rattling structures or using phonons tends to be effective at exploring the local PES and it is helpful to provide the original structures before the displacement^{56, 65-67}. For these displacement techniques, the type and magnitude of displacements, starting structures, unit cell, and phonon modes, should be provided.

Combinatorial methods involve the use of an algorithm to generate structures that ideally cover the entire configuration space required. The exact algorithm and method should be reported to identify possible biases, and a representative script should be provided. For example, unoptimized structures of liquids may be generated using a code such as Packmol⁶⁸ and thus an input file for Packmol⁶⁸ would be ideal to report. In contrast, crystalline materials such as high entropy alloys would require an entirely different algorithm with knowledge of a lattice structure to randomly place elements and distort the cell to reasonable lattice constants.⁶⁹ The SI should contain an example script or pseudocode that illustrates the logical flow and allows the reader to implement it in their choice of programming language. If the cost of generating additional structures is low, there may also be value in generating a supplementary dataset without calculated properties that can be used in the future to augment the main dataset.

In contrast to single-point methods, dynamical simulations can explore the potential energy surface directly and therefore give physically meaningful configurations for a given temperature, pressure, and composition. Reporting the MD ensemble (NVE, NVT, NPT), timestep, and thermostat/barostat settings is critical as these MD parameters directly affect the system's dynamics. Directly providing the input files required for the simulation is a bonus as it resolves any doubts about what has been done. Additionally, MD-based methods such as simulated annealing^{70, 71}, contour exploration⁷², or minima/basin hopping⁷³⁻⁷⁸ may be effective for exploring a wide variety of configurations. If such methods are used, their parameters must be reported consistent with literature studies utilizing them for molecular dynamics.

It may also be helpful to point out when simulations can be performed in parallel and when equilibration is not reached. When training MLIPs, it may not be necessary to perform MD runs that are suitable for analysis since the goal of the MLIP is typically to avoid performing long MD

runs. Non-equilibrium MD should thus not be an issue when used as training data, especially as it is likely to improve the stability and robustness of the MLIP potential (stability itself is discussed in the Inference section) and biasing potentials that move the system away from equilibrium are helpful to use and report. If the bias is not directly implemented in the code being used, it is best to provide a small example of how to perform the biased MD⁷⁹.

Constraints in dynamical simulations can influence training data by intentionally biasing the system towards a given state and as such must be reported. Simple constraints such as the relationship between cell vectors or fixing atoms can be reported in the text and may be obvious in the trajectories supplied. However, it is crucial to provide unconstrained training data with all atoms having original forces restored, as calculated by the method used, even though some packages, such as ASE, zero the forces of fixed atoms by default. This is because some MLIP codes are not designed to read atomic constraints and the constrained atoms are included in the loss function with the constrained forces rather than the true forces. While some codes can mask atoms in the loss function to account for constraints, it is easier to simply store and use unconstrained data, which also ensures that the total energy and the forces in the system are fully consistent. More sophisticated constraint schemes (e.g., enhanced sampling⁸⁰⁻⁸²) must be reported as well and it is up to the author to properly describe them and/or provide an example script in the SI. For example, collective variables may be defined that control a bond distance as a function of some other property of the system to allow for a rare event to be performed that would be unlikely to occur naturally in an MD simulation.⁸³⁻⁸⁶

After generating the initial training data, it is also important to consider how relevant training points are selected. For example, using all frames of an MD simulation will result in highly correlated data. As such it is necessary to have a procedure for data selection, such as including

only every X^{th} atomic configuration. Alternatively, sampling configurations that cover a range of MD properties such as energy, force, pressure, or volume may avoid correlated structures.^{79, 87} While more data is assumed to be helpful in improving an MLIP, our experience with modern codes suggests that smaller curated datasets tend to require fewer resources to train to high accuracy. If active learning is used, it will be necessary to note down the criteria for selection of sampled configurations⁸⁸⁻⁹⁰. For example, if an uncertainty metric is utilized, the method for determining uncertainty must be described well and thresholds for data selection reported. It is common for uncertainty metrics to be used with active/iterative learning schemes, but standardization of reporting these learning schemes is non-trivial. Algorithms for active learning may also differ widely. Our suggestion is to report the detailed learning scheme and make the code public that is used if possible. Reporting the selection method and the unfiltered dataset allows for other selection methods to be tried later as they are discovered.

2.4 Training MLIPs

As model parameters may vary from package to package, detailed documentation of the model architecture, hyperparameters, and training methods is especially important for reproducibility. Although modern MLIP parameters might be confusing to those unfamiliar with specific packages, they can still be documented in a way that is generally useful and does not necessitate expert knowledge. For example, reporting of radial cutoff(s) allows for fundamental limits of the MLIP's knowledge of long-range order to be evaluated in a way that is completely agnostic to the code being used. We note, however, that while the radial cutoff determines the direct range of the MLIP, message-passing or non-local MLIPs may include contributions beyond this radial cutoff⁹¹⁻⁹³. Similarly, a model's complexity can be estimated by its degrees of freedom with the caveat that not all degrees of freedom are equal within an MLIP or between different MLIPs. Most other MLIP

parameters, however, may be crucial but difficult for non-expert readers to understand. We suggest including only the crucial parameters in the main document, especially those that have been explicitly tuned, while tabulating the rest in the SI. Another simple and efficient method for reporting is to disclose the full input file. We encourage MLIP developers to output files during training that can be directly used as input files. Such files should explicitly define all defaults to enable easy reporting in a way that is more agnostic to the specific version being used or if defaults are changed on installed package.

If changes are made to the publicly available version of the code and licensing permits, patches that are critical to training, together with their purpose, should be provided in the SI. Software improvements and implementation of alternative approaches should be viewed positively when they improve training; their value should not be underestimated even if they are simple.

While software packages may differ greatly, generic rules can be constructed for common models such as neural network (NN) potentials^{18-21, 23-25, 27, 28}. The architecture and descriptors of a NN potential must be fully described with its layers, widths, and model features. The architecture of the model limits its accuracy given a fixed set of descriptors and as such determines the final quality of the model. Learning rates and loss functions may also influence the model strongly and any changes during training (multi-phase training, learning rate schedulers, etc.) should be detailed. Recently, the introduction of equivariance to MLIPs has also been seen to improve models^{19, 94, 95}. For such codes, one should report related parameters such as the symmetries allowed or the maximum tensor ranks. Other general ML parameters such as activation function may not appear to be important but should be reported regardless as they may have unintended impact. If GPUs are used, it is important to report what numerical precision is used within the model as the usage of FP32, FP64, and TF32⁹⁶ may influence results depending on system size

and normalization of data. If CPUs are used, FP64 should always be used (or higher precision) as most modern CPUs do not perform better on FP32 and precision will be lost unnecessarily. Precision in the model is unlikely to matter except when sub-meV/atom errors are reached⁹⁷. The precision dependence can be evaluated by retraining or by reducing the model from FP64 to FP32 or TF32 in the inference stage.

Gaussian process potentials have also become common in the field especially for on-the-fly training^{22, 98, 99} due to their predictable deterministic training time and inherent uncertainty quantification¹⁰⁰. If calculated as part of the training process, the method of optimizing Gaussian process hyperparameters as well as the optimized hyperparameters for the model kernels and noise should also be reported to provide a good starting point for training similar models. It should be disclosed directly if hyperparameters are determined non-systematically. A non-systematic search of hyperparameters may produce good models but indicate that better hyperparameters might be possible to identify in future studies.

As an example of the importance of detailed reporting of the kernel parameters, Gaussian process models can be developed using 2-body, 3-body, or n -body multi-element kernels^{22, 99, 101}. The 2-, 3-, and n -body terms may have differing radial cutoff definitions between atoms; these radial cutoffs should be reported as a table such that particular interactions (e.g., C-H pairs) can be easily identified and compared with radial distribution functions. These models may also be precalculated in terms of its decomposed n -body interactions and approximated by mapping the n -body terms directly to a tabulated form to be looked up during simulation, known as a mapped Gaussian process (MGP). The model is approximated by mapping the n -body terms directly to a tabulated form to be looked up during simulation, known as a mapped Gaussian process (MGP).¹⁰² MGPs provide much faster predictions but may be less accurate if not determined on a fine enough grid.

It is therefore critical to report the exact form of the approximation (e.g., grid spacing between tabulated points, linear vs. spline interpolation, error prediction, etc.). Plotting the 2 or 3 body terms may be helpful to identify issues at extreme distances. We also note that training data for Gaussian processes come from typical DFT calculations but may be added on a per-atom basis rather than the entire configuration in some codes supporting active learning. The per-atom DFT data must be accounted for in the reporting. Training data may be better provided in the form of a list of configurations and atoms or by modifying the training dataset to include tags for atoms that are trained on if utilized. A standardized format for sparse atomic data which is supported across codes would be a useful development for the reproduction and cross-validation of codes. Any format can be used that supports tagging of atoms (extxyz, traj, npz) with an extra parameter that indicates to train on those atoms.

Hyperparameter optimization is commonly performed to search for the best models.¹⁰³ In such cases, it would be useful to note down the search space and the tabulate the tested parameters, which may save time and effort for prospective users looking to study similar systems. If online tools such as Weights & Biases are used for hyperparameter scans, the tools should also be mentioned together with the specific algorithms applied for hyperparameter optimizations¹⁰³⁻¹⁰⁵.

It is also helpful to think of MLIPs as being either feature-based or deep-learning-based. Feature-based MLIPs, such as BPNNs^{106, 107} and GAP¹⁰⁸, usually involve a transformation of the input structures into suitable features/descriptors for subsequent regression to map these features to the computed energies and forces. These features have properties such as being invariant to translations and rotations that make them conducive for training. On the other hand, deep-learning based MLIPs, learn the features on-the-fly during training. Examples of these MLIPs include NequIP²⁰ and MACE¹⁹, which use graphs to represent the input structures. Message passing neural

networks¹⁰⁹ (MPNNs) then operate on these graphs, where a series of interaction blocks can generate higher order features on-the-fly.

We note that depending on the type of MLIP trained, different analyses of the training data can be performed. For example, in feature-based MLIPs, it is simple to obtain the feature vectors and so it can be helpful to plot the distribution of the features to visualize the training space. On the other hand, for deep-learning-based MLIPs, obtaining the features can be difficult. We therefore suggest that for deep-learning-based models, the model itself—including the network architecture and weights—can be shared to ensure reproducibility and facilitate follow-up studies.

2.5. Validating MLIPs

Evaluating the accuracy of an MLIP is commonly performed by assessing errors in uncorrelated structure predictions and in statistical/thermodynamic properties relevant to physical insights that will be drawn from the MLIP. The MLIP should predict energies, forces, and stresses accurately or any predictions will be a result of coincidence or error cancellation which may harm transferability or robustness^{43, 110}. For proper evaluation of the model performance, the “test set”, used for evaluating inference errors, must include carefully sampled data that is representative of the configurations the MLIP will need to predict although computational limitations may limit the system size. If that is the case, authors should report known limitations to inform the readers. A common error in creating test sets is to include atomic configurations from intermediate MD steps between those used as training images, which will give artificially low errors. Instead, we recommend building test sets from MD runs independent of those used for training so that the model’s performance is evaluated on data that is not directly correlated with the training set. In addition, validation must not be performed using “validation set”, used for selecting models during training, as reported by MLIP codes as these errors are simply a metric to select an ideal model.

The validation set leaks knowledge of its data into the model indirectly and influences the training process. In addition, the test set and validation set meaning should be well-defined in the methods section of the paper as the literature is not consistent in their definitions.

Error metrics, such as the mean absolute error (MAE), mean signed error (MSE), mean percentage error (MPE), or root mean square errors (RMSE), allow for easy quantification of the quality of the MLIP model and can highlight where systematic errors exist for improvement. Errors must be reported for all predicted properties such as energy, force, and stress. Force error can also be reported in terms of angle and magnitude, which are more meaningful and are rotationally invariant quantities. Force errors should also be computed based on the unmodified ground truth without constraints such as fixed atoms. Contributions from fixed atoms may be removed on a per-image basis instead but removing such contributions is not directly supported by all codes. Stress errors must also be reported carefully with the ideal gas contribution of velocities to stress removed before analysis as ideal gas contributions come from velocity in the MD simulation. The stress error should also be partitioned into the normal and shear components separately before analysis as it is common for shear stresses to not be considered during NPT simulations and the normal stress solely controls the pressure dependence.

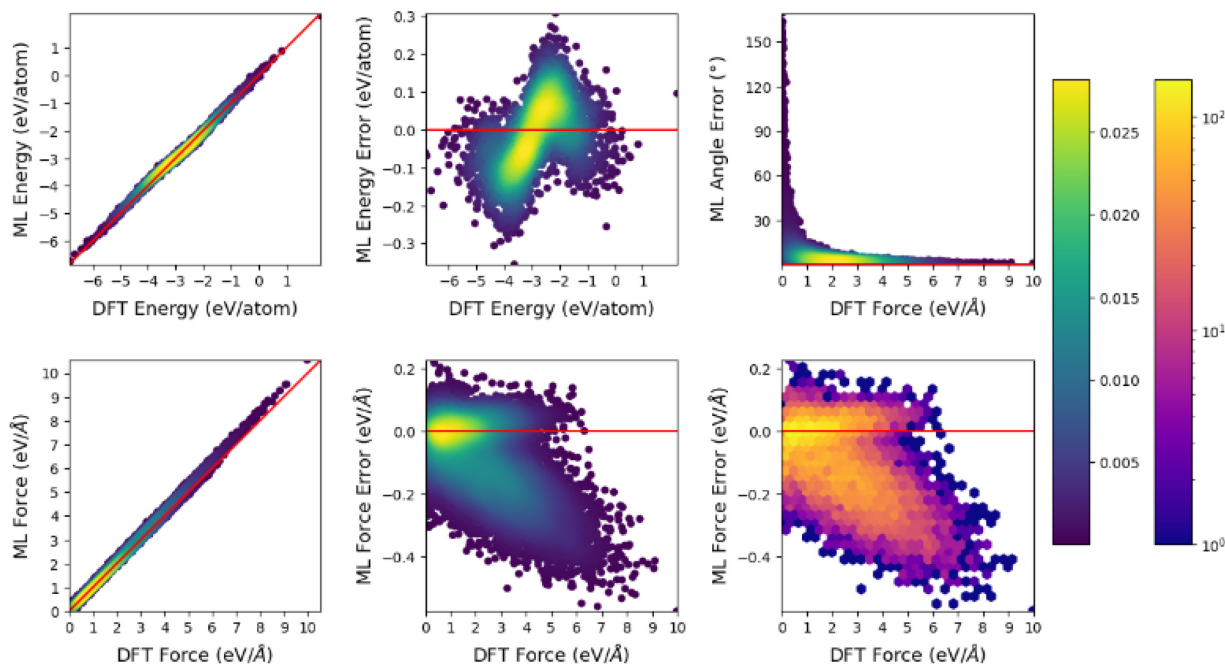


Figure 2. Improving the presentation of energy errors and force magnitude errors. Plots (a) and (d) are intended as examples of parity plots that show very little useful information, while plots (b) and (e) are the corresponding error distribution plots generated with KDE coloring highlight residual errors more effectively. The angle error of force is presented in plot (c). Plot (f) is the histogram analog of plot (e), showing the KDE plots are superior to a histogram due to distortion of data in the histogram. Data is generated as mock data from NumPy to highlight how to best plot MLIP errors and is not intended to be interpreted as a result of any simulation or training. A script to generate a similar plot is provided in the Supplementary Information. All plots are colored according to the default matplotlib color mapping (viridis), except for the hexbin plot (plasma) due to the change to a log scale.

Validation of MLIP potentials is often presented graphically as a function of system properties. For simple statistics such as energy or force errors, parity plots (Figures 2a and 2d) are typically the go-to choice as they allow the reader to easily compare predictions with benchmarks and determine if systematic errors are present. However, when parity plots are calculated over a wide range of values, the resolution of points is reduced, which may hide important discrepancies. In such cases, plotting the error distribution can help highlight specific patterns that cannot be seen on parity plots and indicate issues of the model (Figures 2b and 2e). For example, in Figure 2b there is a systematic deviation that is hidden in the parity plot of the same data. In Figure 2e, two distributions appear to be present, which may be due to atoms in different chemical environments

or accidental mixing of training sets with different ground truths. Errors may also be correlated to other properties to highlight how predictions change as a function of the ground truth or predicted properties. For example, Figure 2c demonstrates a clear relationship between the force angle error and the magnitude of the force, with low magnitudes having larger angle errors in general. We suggest also plotting errors against properties such as pressure, atomic composition, and system size to check for correlations that might indicate simulation regions of concern and possible ways to improve the model. If the MLIP provides a direct prediction of additional properties (e.g., magnetic moment, local electron density, density of states) these may also be validated at this stage.

To improve visual clarity when there is significant data overlap, the distribution of points can be colored via a kernel density estimation (KDE). This is visually clearer in comparison to a histogram, which more explicitly puts the data into bins to calculate a distribution but distorts the appearance as seen in Figure 2f. Note that the choice of bandwidth and kernel function can greatly affect the KDE, and as such these parameters should be reported as well. However, a histogram may be helpful if the data needs to be digitalized as discrete points may not be present when using a KDE in a region of high density. We also stress the importance of choosing a good color scheme when using color to encode data and suggest providing color bars and/or mentioning the color map if it has a common name. We provide an example script capable of producing both parity and error distribution plots in the SI.

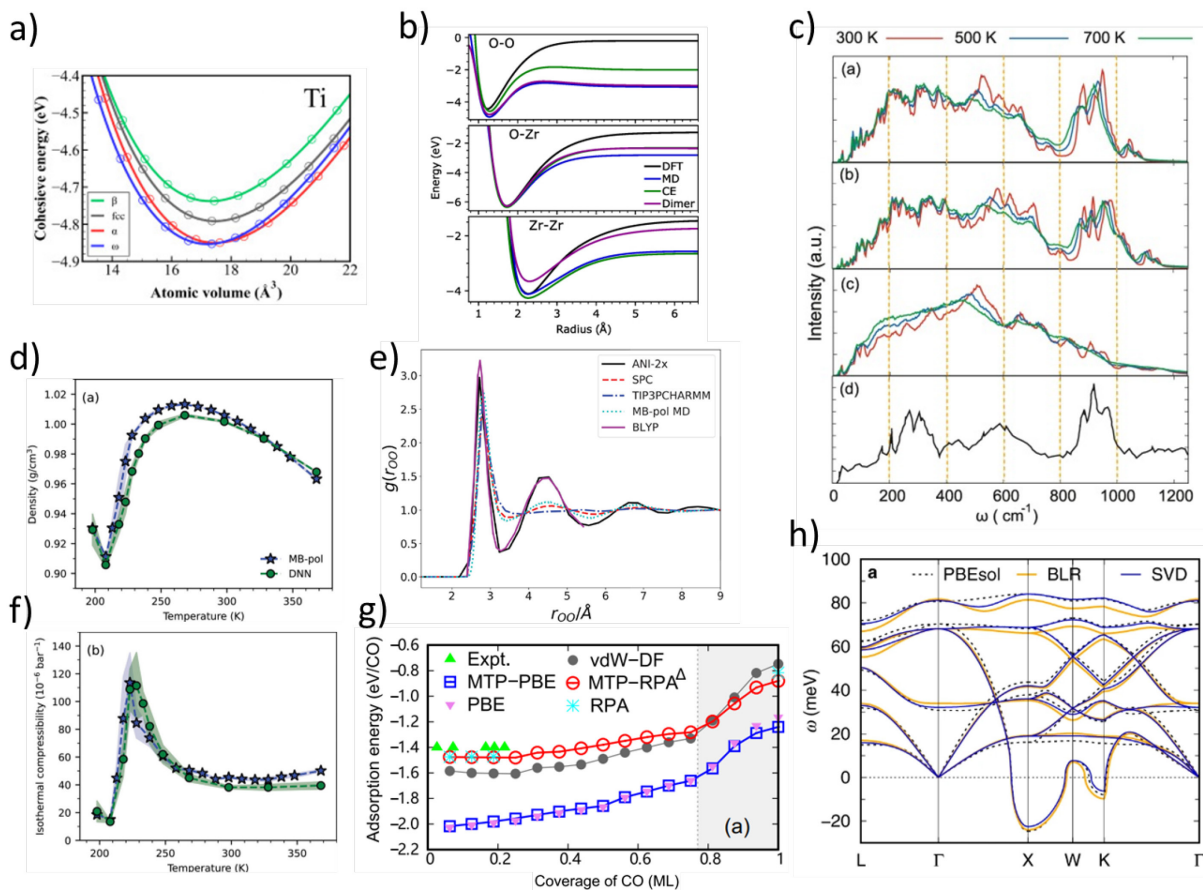


Figure 3. Example plots of (a) equation of state,¹¹¹ (b) two-body potential energy curves,¹¹² (c) vibrational density of states,¹¹³ (d) density,¹¹⁴ (e) radial distribution function,¹¹⁵ (f) isothermal compressibility,¹¹⁴ (g) coverage dependent adsorption energy,¹¹⁶ and (h) phonon band structure¹⁰¹ using MLIPs in literature studies. These highlighted plots demonstrate a wide variety of styles found in the literature for reporting MLIP predicted properties; we suggest the reader view them in the context of the original publication. (a) Adapted with permission¹¹¹. Copyright 2022 Elsevier. (b) Adapted with permission¹¹². Copyright 2022 IOP Publishing Ltd. (c) Adapted with permission¹¹³. Copyright 2022 Elsevier. (d, f) Adapted with permission¹¹⁴. Copyright 2021 AIP Publishing. (e) Adapted with permission¹¹⁵. Copyright 2021 American Chemical Society. (g) Adapted with permission¹¹⁶. Copyright 2023 American Physical Society. (h) Adapted with permission¹⁰¹. Copyright 2021 Springer Nature.

A particularly meaningful way to validate MLIPs is to check if the predictions in MD simulations match with the ground truth they are based upon. Such practice is especially useful when the validated property is directly related to the physical insights the MLIP study is intended to provide. Examples of MD-derived properties are the vibrational density of states, density, radial distribution function (RDF), and isothermal compressibility as seen in Figures 3c, 3d, 3e, and 3f, respectively. Evaluation of MD properties may be limited by equilibration time in DFT, so it may not be possible

to directly validate with MD properties. Instead, it may be possible to equilibrate using the MLIP and report if a DFT simulation started from the final MLIP configuration starts to deviate in any thermodynamic properties over a short time, which may indicate a disagreement with the MLIP prediction. Tabulating all calculated properties in the SI will also facilitate direct comparison in the future when new methods or more comprehensive DFT simulations become available. We also suggest presenting differences of properties when the MLIP and DFT results appear to be well converged, since the resolution of data in plots may make it difficult to see where agreement is better or worse. For example, in Figure 3e the radial distribution function (RDF) appears to be very similar for the MLIP and MD runs. However, the agreement is not perfect, and a difference plot of RDFs would be informative. We note that the MLIP results should not be compared with experiments: this is a common mistake during validation of the MLIP and is addressed further in the inference section of the perspective.

Relevant thermodynamic quantities related to the studied research questions, such as n -body interactions, thermal expansion coefficients, binding energies, and phonons/vibrations, should also be evaluated and reported from analyses such as many-body decomposition^{114, 117, 118} or analyses of static systems^{119, 120} to ensure the MLIP provides meaningful physical insights. In many-body decomposition, DFT is used to calculate various configurations of 2-body, 3-body, or n -body isomers, enabling the precise determination of their decomposed interaction energies. Differences between MLIP and DFT interaction energies reveal the presence of systematic errors or instances of unphysical error cancellation. Two-body interactions may also be presented as a dissociation curve as seen in Figure 3b. Optimizing structures using DFT and MLIPs and reporting the residual mean square displacements (RMSD) can provide additional information about the low-energy structures predicted by the MLIP. Properties based solely on system energy, such as the formation

energy of nanoparticles, cohesive energies of bulk structures, and binding energies of adsorbates, can be validated without running a full MD simulation at the DFT level. Other examples are an equation of state¹²¹, the binding energy of adsorbates on a surface,¹²² or the phonon band structure of a system^{58, 123} as seen in Figures 3a, 3g, and 3h, respectively. We also suggest reporting validation of thermodynamic properties both statistically (for quick comparison) and graphically (for identification of trends) when possible.

Modern MLIPs are commonly strictly local potentials with their interactions confined to their radius of interaction, which guarantees that the models are size-consistent and size-extensive. However, in large unit cells, the long-range order that forms may be longer than the MLIP radius of interaction. On the other hand, in small unit cells, there can be artificial short-range order. The orders that form can thus bias the training set towards interactions that arise in specific system sizes and the MLIP may not generalize well as a result. It is logical to validate the system as a function of system size as well when the system size is changing. However, validating the model on large systems may present unique challenges, as MLIPs are typically used when the ground truth for larger systems is too expensive to compute. Therefore, when applying MLIPs trained on small systems to larger ones, it's advisable to invest in a limited set of single-point calculations to assess the MLIP's transferability to these larger systems. Potential errors may be verified and reported by checking for discontinuous or unphysical dissociation curves of atoms or by evaluating MD properties as a function of the number of atoms.

We also suggest reporting the test results of MLIPs beyond their training set conditions as extreme conditions, such as elevated temperatures, can help identify problems with stability and help us understand how the potential will extrapolate to unknown configurations^{124, 125}. This test is particularly useful since the test accuracy of a model is not necessarily a good proxy of its

extrapolative ability^{124, 126}. Deliberately creating non-physical interactions in simulations may help identify when and why the MLIP fails during longer MD runs. For example, the O-H bond of water might break during MD runs, forming OH. If OH is not present in the training dataset, it would be good to include it to analyze how an unknown interaction will be handled in MD simulations if it occurs by chance. Similarly, performing a topology analysis in the water system would ensure that the simulation is still in a stable state given knowledge of the training dataset.

2.6 Inferring MLIPs

The most common use for MLIPs is to perform accelerated MD simulations in suitably modified classical MD software such as LAMMPS¹²⁷, GROMACS¹²⁸, or OpenMM¹²⁹. Suggestions on how to perform and report MD simulations is beyond the scope of this perspective, as they should be performed identically to similar classical simulations. We do, however, suggest providing direct input files for reproduction. It can additionally be helpful for practitioners to know the MLIP evaluation speed, preferably reported in “atoms/second”, as this is agnostic to system size and time step and may be related to the hardware reported. We also suggest reporting actual memory usage as the hardware provides a hard constraint on memory and requirements for inference may be significantly higher or lower than that for training depending on system size. Determining memory usage as a function of system size and providing this information in the SI will assist readers in understanding scaling of codes that they may be unfamiliar with.

MLIP inference is often challenging as major, hidden problems can arise due to stability issues¹³⁰. While some guidelines for MLIP stability exist, addressing this complex issue is difficult as failures may occur differently on a code-to-code basis. MLIPs can sometimes predict unrealistic behavior even if the test dataset shows low errors as ML models typically do not follow physical constraints. For example, the MLIP might incorrectly predict bonding at unphysical distances or

bonds may break unexpectedly. Such stability issues should be reported even if a solution may be out of reach. To improve stability, augmenting the MLIP with physically motivated terms such as nuclear repulsion, dispersive forces, and force constraints can help.^{91, 131-133} If augmentations are applied, their exact form must be reported. It is also important to report any known limitations with regard to the temperature, pressure, or material phase they intended to operate in as it is common for stability issues to arise whenever the potential is extrapolating beyond its original training set. The limitations may be discussed in a “Limitations” section of the manuscript (or added to SI if there are space constraints) so that readers can find all known inference issues with MLIPs. A limitations section may also be utilized to discuss strategies to keep potentials stable within known instabilities by use of adaptive timesteps or schemes to restart MD runs from slightly different random seeds.

Validation of trained MLIPs by experimental comparison is a misguided practice as the underlying training data is not necessarily correlated with experiments and may result in false confidence in the quality of the MLIP. Errors arising from the computational method or model approximations will cause deviations from experiments, which must be reproduced in the MLIP. As a result, experimental comparisons should not be used to validate the quality of MLIPs, but instead treated as emergent results to serve as predictions of the DFT simulation. As such, if the MLIP is expected to reproduce the DFT method well, any agreement with experiments likely indicates the underlying method would agree in absence of the use of MLIPs. Experimental comparisons are therefore useful for gaining physical insights into the underlying method used in data generation. However, they must be reported with caution to avoid grand claims that cannot be supported by the work when direct verification is not possible. Here, we stress to the reader that

“validation” of the MLIP by experimental results is not validation but is instead computing results from the inference stage.

2.7 Data Availability / FAIR

Data lies at the heart of machine learning. To promote software development and verify the results resulting from MLIPs, proper archival of data should be required. Data requests to authors may initially work but will be less effective as the publication and authors age. Adhering to the FAIR data principles¹³⁴⁻¹³⁶—making data findable, accessible, interoperable, and reusable—is most painless at the publication stage and must be advocated for by journals hosting MLIP publications. Public, unaltered datasets necessary for reproducing the results should be provided for publication. This should be provided in standard formats compatible with common software (Atomic Simulation Environment¹³⁷, pymatgen¹³⁸, etc.) to facilitate interoperability between codes. It is important to keep useful information when possible, such as magnetic moments, even if this is not used by the MLIP developer in their current work. This extra information may be used in future MLIPs if the dataset is reused or verify that future data generation is consistent with the old dataset if it is extended. While the SI can serve as a backup of training inputs, large datasets are better hosted on research repositories like Zenodo¹³⁹ or NOMAD¹⁴⁰. These repositories also provide DOIs for easy referencing of specific versions (which may be updated after publication) and retrieval of specific files from the dataset if properly formatted. We advise against using software-focused Git-based repositories such as GitHub and GitLab, since they may be modified less transparently, and bandwidth and size limitations may force authors to reduce the amount of data available. However, even research repositories may have limited file sizes. In that case, we suggest uploading a representative portion of the data. All training data should be reported, and it should be disclosed if it is not

Sharing trained models can also be very useful. As models are difficult to generally deploy across different computational systems without modification, we suggest providing all tools required for retraining. In some cases, the MLIP training code is not publicly available, making direct reproduction impossible. The models should still be reported as they will be useful for comparison with other MLIP training codes available at or after publication, encouraging healthy competition between development groups. Additional effort should be applied towards sharing trained models when training times are large due to the size of the dataset and retraining the model invokes an excessive cost on future practitioners preventing reproduction. An open interface online which is provided for a pre-trained model, such as that of the Open Catalyst project demo on the OC20 dataset^{51, 141, 142}, can be a helpful alternative when sharing the model is not possible. However, there must be still caution with the suggested practices as there is no way to guarantee that third-party services hosted by individual researchers will be available long term. If it is possible, providing the repository containing everything required to self-host the service helps ensure it is not lost if the original service provider is no longer functional.

3. Conclusions

Establishing clear and comprehensive reporting standards for MLIPs is critical for ensuring the reliability, reproducibility, and advancement of MLIP-accelerated simulations. This perspective highlights various aspects that may not currently be rigorously documented but are critical for increasing the usefulness and reproducibility of publications. We encourage the adoption of standardized reporting practices that are enforced at the personal, group, and journal levels before publication. A more standardized approach will also assist in peer review by ensuring that reviewers have all pertinent information on their first viewing, accelerating the publication process by reducing the need for lengthy revisions. The challenges with MLIP reporting arise largely from

the diverse reporting requirements of newer software available and the lack of long-term, field-specific standards. To tackle this, we provide a checklist in the Supporting Information encompassing the suggestions of this perspective. The dynamic and evolving nature of MLIPs and their applications requires an adaptable standard that lays the groundwork for all MLIPs to be reported similarly. We hope that the establishment of such standards may also accelerate the adoption of MLIPs in the common workflows of DFT practitioners.

Supporting Information.

The following files are available free of charge.

Checklist of parameters to report for authors to reference prior to publication and a script to serve as an example of how to plot validation data using matplotlib. (PDF)

Author Information

Corresponding Author

Tibor Szilvási - Department of Chemical and Biological Engineering, University of Alabama, Tuscaloosa, AL 35487, United States; Email: tibor.szilvasi@ua.edu

Authors

Tristan Maxson - Department of Chemical and Biological Engineering, University of Alabama, Tuscaloosa, AL 35487, United States

Ademola Soyemi - Department of Chemical and Biological Engineering, University of Alabama, Tuscaloosa, AL 35487, United States

Benjamin W. J. Chen - Institute of High Performance Computing (IHPC), Agency for Science, Technology, and Research (A*STAR), 1 Fusionopolis Way, #16–16 Connexis, Singapore 138632, Singapore

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. All authors have contributed equally.

Notes

The authors declare no competing financial interest.

Acknowledgements

T.M., A.S. and T.S. would like to acknowledge the financial support of the National Science Foundation (NSF) under grant number 2245120 and the financial support of the Department of Energy (DOE) under grant number DE-SC0024654. T.M. would like to acknowledge this material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number(s) DE-SC0023112. A.S would like to acknowledge the financial support of the University of Alabama Graduate School as a Graduate Council Fellow. B.W.J.C is grateful for support by the A*STAR SERC Central Research Fund award. T.M., A.S. and T.S. would also like to thank the University of Alabama and the Office of Information Technology for providing high-performance computing resources and support that has contributed to these research results. This work was also made possible in part by a grant of high-performance computing resources and technical support from the Alabama Supercomputer Authority. This

research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award BES-ERCAP0024218. Any opinions, findings, conclusions, and/or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the NSF or DOE.

References

1. Liu, D.-J.; Evans, J. W., Fluorine spillover for ceria- vs silica-supported palladium nanoparticles: A MD study using machine learning potentials. *The Journal of Chemical Physics* **2023**, *159* (2), 024101.
2. Zhou, C.; Ngan, H. T.; Lim, J. S.; Darbari, Z.; Lewandowski, A.; Stacchiola, D. J.; Kozinsky, B.; Sautet, P.; Boscoboinik, J. A., Dynamical Study of Adsorbate-Induced Restructuring Kinetics in Bimetallic Catalysts Using the PdAu(111) Model System. *Journal of the American Chemical Society* **2022**, *144* (33), 15132-15142.
3. Schaaf, L.; Fako, E.; De, S.; Schäfer, A.; Csányi, G., Accurate Reaction Barriers for Catalytic Pathways: An Automatic Training Protocol for Machine Learning Force Fields. *arXiv preprint arXiv:2301.09931* **2023**.
4. Noé, F.; De Fabritiis, G.; Clementi, C., Machine learning for protein folding and dynamics. *Current Opinion in Structural Biology* **2020**, *60*, 77-84.
5. Bereau, T.; DiStasio, R. A., Jr.; Tkatchenko, A.; von Lilienfeld, O. A., Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning. *The Journal of Chemical Physics* **2018**, *148* (24), 241706.
6. Gkeka, P.; Stoltz, G.; Barati Farimani, A.; Belkacemi, Z.; Ceriotti, M.; Chodera, J. D.; Dinner, A. R.; Ferguson, A. L.; Maillet, J.-B.; Minoux, H.; Peter, C.; Pietrucci, F.; Silveira, A.; Tkatchenko, A.; Trstanova, Z.; Wiewiora, R.; Lelièvre, T., Machine Learning Force Fields and Coarse-Grained Variables in Molecular Dynamics: Application to Materials and Biological Systems. *Journal of Chemical Theory and Computation* **2020**, *16* (8), 4757-4775.
7. Lai, K. C.; Liu, D.-J.; Evans, J. W., Nucleation-mediated reshaping of faceted metallic nanocrystals: Breakdown of the classical free energy picture. *The Journal of Chemical Physics* **2023**, *158* (10), 104102.
8. Lim, J. S.; Vandermause, J.; van Spronsen, M. A.; Musaelian, A.; Xie, Y.; Sun, L.; O'Connor, C. R.; Egle, T.; Molinari, N.; Florian, J.; Duanmu, K.; Madix, R. J.; Sautet, P.; Friend, C. M.; Kozinsky, B., Evolution of Metastable Structures at Bimetallic Surfaces from Microscopy and Machine-Learning Molecular Dynamics. *Journal of the American Chemical Society* **2020**, *142* (37), 15907-15916.
9. Ye, H.; Xian, W.; Li, Y., Machine Learning of Coarse-Grained Models for Organic Molecules and Polymers: Progress, Opportunities, and Challenges. *ACS Omega* **2021**, *6* (3), 1758-1772.

10. Hong, S. J.; Chun, H.; Lee, J.; Kim, B.-H.; Seo, M. H.; Kang, J.; Han, B., First-Principles-Based Machine-Learning Molecular Dynamics for Crystalline Polymers with van der Waals Interactions. *The Journal of Physical Chemistry Letters* **2021**, *12* (25), 6000-6006.
11. Ding, L.; Davidchack, R. L.; Pan, J., A molecular dynamics study of sintering between nanoparticles. *Computational Materials Science* **2009**, *45* (2), 247-256.
12. Sheavly, J. K.; Gold, J. I.; Mavrikakis, M.; Van Lehn, R. C., Molecular simulations of analyte partitioning and diffusion in liquid crystal sensors. *Molecular Systems Design & Engineering* **2020**, *5* (1), 304-316.
13. Dawson, W.; Gygi, F., Equilibration and analysis of first-principles molecular dynamics simulations of water. *The Journal of Chemical Physics* **2018**, *148* (12), 124501.
14. Buesser, B.; Gröhn, A. J.; Pratsinis, S. E., Sintering Rate and Mechanism of TiO₂ Nanoparticles by Molecular Dynamics. *The Journal of Physical Chemistry C* **2011**, *115* (22), 11030-11035.
15. Yamijala, S. S. R. K. C.; Ali, Z. A.; Wong, B. M., Acceleration vs Accuracy: Influence of Basis Set Quality on the Mechanism and Dynamics Predicted by Ab Initio Molecular Dynamics. *The Journal of Physical Chemistry C* **2019**, *123* (41), 25113-25120.
16. Saucedo, H. E.; Chmiela, S.; Poltavsky, I.; Müller, K.-R.; Tkatchenko, A., Construction of Machine Learned Force Fields with Quantum Chemical Accuracy: Applications and Chemical Insights. In *Machine Learning Meets Quantum Physics*, Schütt, K. T.; Chmiela, S.; von Lilienfeld, O. A.; Tkatchenko, A.; Tsuda, K.; Müller, K.-R., Eds. Springer International Publishing: Cham, 2020; pp 277-307.
17. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L., Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **2019**, *32*, 8026-8037.
18. Wang, H.; Zhang, L.; Han, J.; E, W., DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics. *Computer Physics Communications* **2018**, *228*, 178-184.
19. Batatia, I.; Kovacs, D. P.; Simm, G.; Ortner, C.; Csányi, G., MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems* **2022**, *35*, 11423-11436.
20. Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B., E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications* **2022**, *13* (1), 2453.
21. Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E., TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Network Potentials. *Journal of Chemical Information and Modeling* **2020**, *60* (7), 3408-3415.
22. Jinnouchi, R.; Karsai, F.; Kresse, G., On-the-fly machine learning force field generation: Application to melting points. *Physical Review B* **2019**, *100* (1), 014105.
23. Khorshidi, A.; Peterson, A. A., Amp: A modular approach to machine learning in atomistic simulations. *Computer Physics Communications* **2016**, *207*, 310-324.
24. Zeng, J.; Zhang, D.; Lu, D.; Mo, P.; Li, Z.; Chen, Y.; Rynik, M.; Huang, L. a.; Li, Z.; Shi, S.; Wang, Y.; Ye, H.; Tuo, P.; Yang, J.; Ding, Y.; Li, Y.; Tisi, D.; Zeng, Q.; Bao, H.; Xia, Y.; Huang, J.; Muraoka, K.; Wang, Y.; Chang, J.; Yuan, F.; Bore, S. L.; Cai, C.; Lin, Y.; Wang, B.; Xu, J.; Zhu, J.-X.; Luo, C.; Zhang, Y.; Goodall, R. E. A.; Liang, W.; Singh, A. K.; Yao, S.; Zhang, J.; Wentzcovitch, R.; Han, J.; Liu, J.; Jia, W.; York, D. M.; E, W.; Car, R.;

- Zhang, L.; Wang, H., DeePMD-kit v2: A software package for deep potential models. *The Journal of Chemical Physics* **2023**, *159* (5), 054801.
25. Unke, O. T.; Muwly, M., PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation* **2019**, *15* (6), 3678-3693.
 26. Shapeev, A. V., Moment tensor potentials: A class of systematically improvable interatomic potentials. *Multiscale Modeling & Simulation* **2016**, *14* (3), 1153-1173.
 27. Hu, L.; Huang, B.; Liu, F., Atomistic Mechanism Underlying the Surface Reconstruction Revealed by Artificial Neural-Network Potential. *Physical Review Letters* **2021**, *126* (17), 176101.
 28. Saucedo, H. E.; Gálvez-González, L. E.; Chmiela, S.; Paz-Borbón, L. O.; Müller, K.-R.; Tkatchenko, A., BIGDML—Towards accurate quantum machine learning force fields for materials. *Nature Communications* **2022**, *13* (1), 3733.
 29. Behler, J.; Martoňák, R.; Donadio, D.; Parrinello, M., Metadynamics Simulations of the High-Pressure Phases of Silicon Employing a High-Dimensional Neural Network Potential. *Physical Review Letters* **2008**, *100* (18), 185501.
 30. Chen, B. W. J.; Zhang, X.; Zhang, J., Accelerating explicit solvent models of heterogeneous catalysts with machine learning interatomic potentials. *Chemical Science* **2023**, *14* (31), 8338-8354.
 31. Rice, P. S.; Liu, Z.-P.; Hu, P., Hydrogen Coupling on Platinum Using Artificial Neural Network Potentials and DFT. *The Journal of Physical Chemistry Letters* **2021**, *12* (43), 10637-10645.
 32. Ang, S. J.; Wang, W.; Schwalbe-Koda, D.; Axelrod, S.; Gómez-Bombarelli, R., Active learning accelerates *ab initio* molecular dynamics on reactive energy surfaces. *Chem* **2021**, *7* (3), 738-751.
 33. Manzhos, S.; Carrington, T., Jr., Neural Network Potential Energy Surfaces for Small Molecules and Reactions. *Chemical Reviews* **2021**, *121* (16), 10187-10217.
 34. Deringer, V. L.; Bernstein, N.; Csányi, G.; Ben Mahmoud, C.; Ceriotti, M.; Wilson, M.; Drabold, D. A.; Elliott, S. R., Origins of structural and electronic transitions in disordered silicon. *Nature* **2021**, *589* (7840), 59-64.
 35. Kapil, V.; Schran, C.; Zen, A.; Chen, J.; Pickard, C. J.; Michaelides, A., The first-principles phase diagram of monolayer nanoconfined water. *Nature* **2022**, *609* (7927), 512-516.
 36. Boes, J. R.; Kitchin, J. R., Modeling Segregation on AuPd(111) Surfaces with Density Functional Theory and Monte Carlo Simulations. *The Journal of Physical Chemistry C* **2017**, *121* (6), 3479-3487.
 37. Deng, B.; Zhong, P.; Jun, K.; Riebesell, J.; Han, K.; Bartel, C. J.; Ceder, G., CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence* **2023**, *5* (9), 1031-1041.
 38. Fung, V.; Ganesh, P.; Sumpter, B. G., Physically Informed Machine Learning Prediction of Electronic Density of States. *Chemistry of Materials* **2022**, *34* (11), 4848-4855.
 39. Bang, K.; Yeo, B. C.; Kim, D.; Han, S. S.; Lee, H. M., Accelerated mapping of electronic density of states patterns of metallic nanoparticles via machine-learning. *Scientific Reports* **2021**, *11* (1), 11604.
 40. Novikov, I.; Grabowski, B.; Körmann, F.; Shapeev, A., Magnetic Moment Tensor Potentials for collinear spin-polarized materials reproduce different magnetic states of bcc Fe. *npj Computational Materials* **2022**, *8* (1), 13.

41. Chapman, J. B. J.; Ma, P.-W., A machine-learned spin-lattice potential for dynamic simulations of defective magnetic iron. *Scientific Reports* **2022**, *12* (1), 22451.
42. Grisafi, A.; Fabrizio, A.; Meyer, B.; Wilkins, D. M.; Corminboeuf, C.; Ceriotti, M., Transferable Machine-Learning Model of the Electron Density. *ACS Central Science* **2019**, *5* (1), 57-64.
43. Fu, X.; Wu, Z.; Wang, W.; Xie, T.; Keten, S.; Gomez-Bombarelli, R.; Jaakkola, T., Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *arXiv preprint arXiv:2210.07237* **2022**.
44. Mattsson, A. E.; Schultz, P. A.; Desjarlais, M. P.; Mattsson, T. R.; Leung, K., Designing meaningful density functional theory calculations in materials science—a primer. *Modelling and Simulation in Materials Science and Engineering* **2005**, *13* (1), R1.
45. Bonomi, M.; Bussi, G.; Camilloni, C.; Tribello, G. A.; Banáš, P.; Barducci, A.; Bernetti, M.; Bolhuis, P. G.; Bottaro, S.; Branduardi, D.; Capelli, R.; Carloni, P.; Ceriotti, M.; Cesari, A.; Chen, H.; Chen, W.; Colizzi, F.; De, S.; Pierre, M. d. L.; Donadio, D.; Drobot, V.; Ensing, B.; Ferguson, A. L.; Filizola, M.; Fraser, J. S.; Fu, H.; Gasparotto, P.; Gervasio, F. L.; Giberti, F.; Gil-Ley, A.; Giorgino, T.; Heller, G. T.; Hocky, G. M.; Iannuzzi, M.; Invernizzi, M.; Jelfs, K. E.; Jussupow, A.; Kirilin, E.; Laio, A.; Limongelli, V.; Lindorff-Larsen, K.; Löhr, T.; Marinelli, F.; Martin-Samos, L.; Masetti, M.; Meyer, R.; Michaelides, A.; Molteni, C.; Morishita, T.; Nava, M.; Paissoni, C.; Papaleo, E.; Parrinello, M.; Pfaendtner, J.; Piaggi, P.; Piccini, G.; Pietropaolo, A.; Pietrucci, F.; Pipolo, S.; Provati, D.; Quigley, D.; Raiteri, P.; Raniolo, S.; Rydzewski, J.; Salvalaglio, M.; Sosso, G. C.; Spiwok, V.; Šponer, J.; Swenson, D. W. H.; Tiwary, P.; Valsson, O.; Vendruscolo, M.; Voth, G., A.; White, A.; Consortium, T. P., Promoting transparency and reproducibility in enhanced molecular simulations. *Nature Methods* **2019**, *16* (8), 670-673.
46. Zuo, Y.; Chen, C.; Li, X.; Deng, Z.; Chen, Y.; Behler, J.; Csányi, G.; Shapeev, A. V.; Thompson, A. P.; Wood, M. A.; Ong, S. P., Performance and Cost Assessment of Machine Learning Interatomic Potentials. *The Journal of Physical Chemistry A* **2020**, *124* (4), 731-745.
47. Bligaard, T.; Bullock, R. M.; Campbell, C. T.; Chen, J. G.; Gates, B. C.; Gorte, R. J.; Jones, C. W.; Jones, W. D.; Kitchin, J. R.; Scott, S. L., Toward Benchmarking in Catalysis Science: Best Practices, Challenges, and Opportunities. *ACS Catalysis* **2016**, *6* (4), 2590-2602.
48. Artrith, N.; Butler, K. T.; Coudert, F.-X.; Han, S.; Isayev, O.; Jain, A.; Walsh, A., Best practices in machine learning for chemistry. *Nature Chemistry* **2021**, *13* (6), 505-508.
49. Schweitzer, N.; Rioux, R. G. R., Addressing Rigor and Reproducibility in Thermal. *Heterogeneous Catalysis* **2023**.
50. Anderson, B.; Hy, T. S.; Kondor, R., Cormorant: Covariant molecular neural networks. *Advances in neural information processing systems* **2019**, *32*, 14537–14546.
51. Tran, R.; Lan, J.; Shuaibi, M.; Wood, B. M.; Goyal, S.; Das, A.; Heras-Domingo, J.; Kolluru, A.; Rizvi, A.; Shoghi, N., The Open Catalyst 2022 (OC22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis* **2023**, *13* (5), 3066-3084.
52. Chanussot, L.; Das, A.; Goyal, S.; Lavril, T.; Shuaibi, M.; Riviere, M.; Tran, K.; Heras-Domingo, J.; Ho, C.; Hu, W., Open catalyst 2020 (OC20) dataset and community challenges. *Acs Catalysis* **2021**, *11* (10), 6059-6072.
53. Allen, C.; Bartók, A. P., Optimal data generation for machine learned interatomic potentials. *Machine Learning: Science and Technology* **2022**, *3* (4), 045031.
54. Ly, K. K.; Ceperley, D. M., Phonons of metallic hydrogen with quantum Monte Carlo. *The Journal of Chemical Physics* **2022**, *156* (4), 044108-1.

55. Ryczko, K.; Krogel, J. T.; Tamblyn, I., Machine Learning Diffusion Monte Carlo Energies. *Journal of Chemical Theory and Computation* **2022**, *18* (12), 7695-7701.
56. Eriksson, F.; Fransson, E.; Erhart, P., The Hiphive Package for the Extraction of High-Order Force Constants by Machine Learning. *Advanced Theory and Simulations* **2019**, *2* (5), 1800184.
57. Togo, A., First-principles phonon calculations with phonopy and phono3py. *Journal of the Physical Society of Japan* **2023**, *92* (1), 012001.
58. Mortazavi, B.; Novikov, I. S.; Podryabinkin, E. V.; Roche, S.; Rabczuk, T.; Shapeev, A. V.; Zhuang, X., Exploring phononic properties of two-dimensional materials using machine learning interatomic potentials. *Applied Materials Today* **2020**, *20*, 100685.
59. Vilhelmsen, L. B.; Hammer, B., A genetic algorithm for first principles global structure optimization of supported nano structures. *The Journal of Chemical Physics* **2014**, *141* (4), 044711.
60. Kwon, Y.; Kang, S.; Choi, Y.-S.; Kim, I., Evolutionary design of molecules based on deep learning and a genetic algorithm. *Scientific reports* **2021**, *11* (1), 17304.
61. Higgins, E. J.; Hasnip, P. J.; Probert, M. I., Simultaneous Prediction of the Magnetic and Crystal Structure of Materials Using a Genetic Algorithm. *Crystals* **2019**, *9* (9), 439.
62. Polishchuk, P., CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics* **2020**, *12* (1), 1-18.
63. Oselladore, E.; Ongaro, A.; Zagotto, G.; Memo, M.; Ribaud, G.; Gianoncelli, A., Combinatorial library generation, molecular docking and molecular dynamics simulations for enhancing the isoflavone scaffold in phosphodiesterase inhibition. *New Journal of Chemistry* **2020**, *44* (45), 19472-19488.
64. Okhotnikov, K.; Charpentier, T.; Cadars, S., Supercell program: a combinatorial structure-generation approach for the local-level modeling of atomic substitutions and partial occupancies in crystals. *Journal of cheminformatics* **2016**, *8*, 1-15.
65. Bisbo, M. K.; Hammer, B., Global optimization of atomic structure enhanced by machine learning. *Physical Review B* **2022**, *105* (24), 245404.
66. Erhard, L. C.; Rohrer, J.; Albe, K.; Deringer, V. L., Modelling atomic and nanoscale structure in the silicon-oxygen system through active machine learning. *arXiv preprint arXiv:2309.03587* **2023**.
67. Poul, M.; Huber, L.; Bitzek, E.; Neugebauer, J., Systematic atomic structure datasets for machine learning potentials: Application to defects in magnesium. *Physical Review B* **2023**, *107* (10), 104103.
68. Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M., PACKMOL: A package for building initial configurations for molecular dynamics simulations. *Journal of Computational Chemistry* **2009**, *30* (13), 2157-2164.
69. van de Walle, A.; Tiwary, P.; de Jong, M.; Olmsted, D. L.; Asta, M.; Dick, A.; Shin, D.; Wang, Y.; Chen, L. Q.; Liu, Z. K., Efficient stochastic generation of special quasirandom structures. *Calphad* **2013**, *42*, 13-18.
70. Chen, B. W. J.; Wang, B.; Sullivan, M. B.; Borgna, A.; Zhang, J., Unraveling the Synergistic Effect of Re and Cs Promoters on Ethylene Epoxidation over Silver Catalysts with Machine Learning-Accelerated First-Principles Simulations. *ACS Catalysis* **2022**, *12* (4), 2540-2551.
71. Lee, Y.; Timmermann, J.; Panosetti, C.; Scheurer, C.; Reuter, K., Staged Training of Machine-Learning Potentials from Small to Large Surface Unit Cells: Efficient Global Structure

- Determination of the RuO₂(100)-c(2 × 2) Reconstruction and (410) Vicinal. *The Journal of Physical Chemistry C* **2023**, *127* (35), 17599-17608.
72. Waters, M. J.; Rondinelli, J. M., Energy contour exploration with potentiostatic kinematics. *Journal of Physics: Condensed Matter* **2021**, *33* (44), 445901.
 73. Sumaria, V.; Nguyen, L.; Tao, F. F.; Sautet, P., Atomic-Scale Mechanism of Platinum Catalyst Restructuring under a Pressure of Reactant Gas. *Journal of the American Chemical Society* **2023**, *145* (1), 392-401.
 74. Khan, S. A.; Caratzoulas, S.; Vlachos, D. G., Catalyst Cluster-Induced Support Restructuring. *The Journal of Physical Chemistry C* **2023**, *127* (45), 22277-22286.
 75. Tkachenko, N. V.; Tkachenko, A. A.; Nebgen, B.; Tretiak, S.; Boldyrev, A. I., Neural network atomistic potentials for global energy minima search in carbon clusters. *Physical Chemistry Chemical Physics* **2023**, *25* (32), 21173-21182.
 76. Goedecker, S., Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. *The Journal of Chemical Physics* **2004**, *120* (21), 9911-9917.
 77. Peterson, A. A., Global Optimization of Adsorbate–Surface Structures While Preserving Molecular Identity. *Topics in Catalysis* **2014**, *57* (1), 40-53.
 78. Zhai, H.; Sautet, P.; Alexandrova, A. N., Global Optimization of Adsorbate Covered Supported Cluster Catalysts: The Case of Pt₇H₁₀CH₃ on α -Al₂O₃. *ChemCatChem* **2020**, *12* (3), 762-770.
 79. Kang, P.-L.; Shang, C.; Liu, Z.-P., Large-Scale Atomic Simulation via Machine Learning Potentials Constructed by Global Potential Energy Surface Exploration. *Accounts of Chemical Research* **2020**, *53* (10), 2119-2129.
 80. Ciccotti, G.; Ferrario, M., Blue Moon Approach to Rare Events. *Molecular Simulation* **2004**, *30* (11-12), 787-793.
 81. Torrie, G. M.; Valleau, J. P., Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics* **1977**, *23* (2), 187-199.
 82. Yang, M.; Bonati, L.; Polino, D.; Parrinello, M., Using metadynamics to build neural network potentials for reactive events: the case of urea decomposition in water. *Catalysis Today* **2022**, *387*, 143-149.
 83. Tiwary, P.; Parrinello, M., From Metadynamics to Dynamics. *Physical Review Letters* **2013**, *111* (23), 230602.
 84. Valsson, O.; Tiwary, P.; Parrinello, M., Enhancing Important Fluctuations: Rare Events and Metadynamics from a Conceptual Viewpoint. *Annual Review of Physical Chemistry* **2016**, *67* (1), 159-184.
 85. Rico, P. F. Z.; Schneider, L.; Perez-Lemus, G.; Alessandri, R.; Dasetty, S.; Menéndez, C. A.; Wu, Y.; Jin, Y.; Nguyen, T.; Parker, J., PySAGES: Flexible, advanced sampling methods accelerated with GPUs. *arXiv preprint arXiv:2301.04835* **2023**.
 86. Mendels, D.; de Pablo, J. J., Collective Variables for Free Energy Surface Tailoring: Understanding and Modifying Functionality in Systems Dominated by Rare Events. *The Journal of Physical Chemistry Letters* **2022**, *13* (12), 2830-2837.
 87. Yang, H.; Zhu, Y.; Dong, E.; Wu, Y.; Yang, J.; Zhang, W., Dual adaptive sampling and machine learning interatomic potentials for modeling materials with chemical bond hierarchy. *Physical Review B* **2021**, *104* (9), 094310.
 88. Podryabinkin, E. V.; Shapeev, A. V., Active learning of linearly parametrized interatomic potentials. *Computational Materials Science* **2017**, *140*, 171-180.

89. Zhang, L.; Lin, D.-Y.; Wang, H.; Car, R.; E, W., Active learning of uniformly accurate interatomic potentials for materials simulation. *Physical Review Materials* **2019**, 3 (2), 023804.
90. Musielewicz, J.; Wang, X.; Tian, T.; Ulissi, Z., FINETUNA: fine-tuning accelerated molecular simulations. *Machine Learning: Science and Technology* **2022**, 3 (3), 03LT01.
91. Anstine, D. M.; Isayev, O., Machine Learning Interatomic Potentials and Long-Range Physics. *The Journal of Physical Chemistry A* **2023**, 127 (11), 2417-2431.
92. Nigam, J.; Pozdnyakov, S.; Fraux, G.; Ceriotti, M., Unified theory of atom-centered representations and message-passing machine-learning schemes. *The Journal of Chemical Physics* **2022**, 156 (20), 204115.
93. Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J., General-Purpose Machine Learning Potentials Capturing Nonlocal Charge Transfer. *Accounts of Chemical Research* **2021**, 54 (4), 808-817.
94. Batzner, S.; Musaelian, A.; Kozinsky, B., Advancing molecular simulation with equivariant interatomic potentials. *Nature Reviews Physics* **2023**, 5 (8), 437-438.
95. Musaelian, A.; Batzner, S.; Johansson, A.; Sun, L.; Owen, C. J.; Kornbluth, M.; Kozinsky, B., Learning local equivariant representations for large-scale atomistic dynamics. *Nature Communications* **2023**, 14 (1), 579.
96. Fasi, M.; Higham, N. J.; Mikaitis, M.; Pranesh, S., Numerical behavior of NVIDIA tensor cores. *PeerJ Computer Science* **2021**, 7, e330.
97. Musaelian, A.; Johansson, A.; Batzner, S.; Kozinsky, B., Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size. *arXiv preprint arXiv:2304.10061* **2023**.
98. Xie, Y.; Vandermause, J.; Ramakers, S.; Protik, N. H.; Johansson, A.; Kozinsky, B., Uncertainty-aware molecular dynamics from Bayesian active learning for phase transformations and thermal transport in SiC. *npj Computational Materials* **2023**, 9 (1), 36.
99. Vandermause, J.; Torrisi, S. B.; Batzner, S.; Xie, Y.; Sun, L.; Kolpak, A. M.; Kozinsky, B., On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *npj Computational Materials* **2020**, 6 (1), 20.
100. Thomas-Mitchell, A.; Hawe, G. I.; Popelier, P., Calibration of uncertainty in the active learning of machine learning force fields. *Machine Learning: Science and Technology* **2023**.
101. Verdi, C.; Karsai, F.; Liu, P.; Jinnouchi, R.; Kresse, G., Thermal transport and phase transitions of zirconia by on-the-fly machine-learned interatomic potentials. *npj Computational Materials* **2021**, 7 (1), 156.
102. Flam-Shepherd, D.; Requeima, J.; Duvenaud, D. In *Mapping Gaussian process priors to Bayesian neural networks*, NIPS Bayesian deep learning workshop, Curran Associates Inc: 2017.
103. Yang, L.; Shami, A., On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, 415, 295-316.
104. Bergstra, J.; Yamins, D.; Cox, D. In *Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures*, International conference on machine learning, PMLR: 2013; pp 115-123.
105. Biewald, L. Experiment Tracking with Weights and Biases. <https://www.wandb.com/> (accessed 2/19/2024).
106. Boes, J. R.; Groenenboom, M. C.; Keith, J. A.; Kitchin, J. R., Neural network and ReaxFF comparison for Au properties. *International Journal of Quantum Chemistry* **2016**, 116 (13), 979-987.

107. Liu, M.; Kitchin, J. R., SingleNN: modified behler–Parrinello neural network with shared weights for atomistic simulations with transferability. *The Journal of Physical Chemistry C* **2020**, *124* (32), 17811-17818.
108. Deringer, V. L.; Csányi, G., Machine learning based interatomic potential for amorphous carbon. *Physical Review B* **2017**, *95* (9), 094203.
109. Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E., Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, Doina, P.; Yee Whye, T., Eds. PMLR: Proceedings of Machine Learning Research, 2017; Vol. 70, pp 1263--1272.
110. Unke, O. T.; Chmiela, S.; Sauceda, H. E.; Gastegger, M.; Poltavsky, I.; Schütt, K. T.; Tkatchenko, A.; Müller, K.-R., Machine Learning Force Fields. *Chemical Reviews* **2021**, *121* (16), 10142-10186.
111. Nitol, M. S.; Dickel, D. E.; Barrett, C. D., Machine learning models for predictive materials science from fundamental physics: An application to titanium and zirconium. *Acta Materialia* **2022**, *224*, 117347.
112. Waters, M. J.; Rondinelli, J. M., Benchmarking structural evolution methods for training of machine learned interatomic potentials. *Journal of Physics: Condensed Matter* **2022**, *34* (38), 385901.
113. Kobayashi, K.; Yamaguchi, A.; Okumura, M., Machine learning potentials of kaolinite based on the potential energy surfaces of GGA and meta-GGA density functional theory. *Applied Clay Science* **2022**, *228*, 106596.
114. Zhai, Y.; Caruso, A.; Bore, S. L.; Luo, Z.; Paesani, F., 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? *The Journal of Chemical Physics* **2023**, *158* (8), 084111.
115. Rosenberger, D.; Smith, J. S.; Garcia, A. E., Modeling of Peptides with Classical and Novel Machine Learning Force Fields: A Comparison. *The Journal of Physical Chemistry B* **2021**, *125* (14), 3598-3612.
116. Liu, P.; Wang, J.; Avargues, N.; Verdi, C.; Singraber, A.; Karsai, F.; Chen, X.-Q.; Kresse, G., Combining Machine Learning and Many-Body Calculations: Coverage-Dependent Adsorption of CO on Rh (111). *Physical Review Letters* **2023**, *130* (7), 078001.
117. Zhai, Y.; Caruso, A.; Bore, S. L.; Luo, Z.; Paesani, F., 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? *The Journal of Chemical Physics* **2023**, *158* (8).
118. Bull-Vulpe, E. F.; Riera, M.; Bore, S. L.; Paesani, F., Data-driven many-body potential energy functions for generic molecules: Linear alkanes as a proof-of-concept application. *Journal of Chemical Theory and Computation* **2022**, 4494–4509.
119. Zhang, Y.; Lunghi, A.; Sanvito, S., Pushing the limits of atomistic simulations towards ultra-high temperature: A machine-learning force field for ZrB₂. *Acta Materialia* **2020**, *186*, 467-474.
120. Chan, H.; Narayanan, B.; Cherukara, M. J.; Sen, F. G.; Sasikumar, K.; Gray, S. K.; Chan, M. K. Y.; Sankaranarayanan, S. K. R. S., Machine Learning Classical Interatomic Potentials for Molecular Dynamics from First-Principles Training Data. *The Journal of Physical Chemistry C* **2019**, *123* (12), 6941-6957.

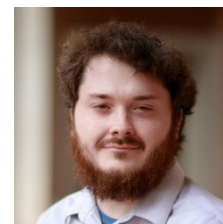
121. Veit, M.; Jain, S. K.; Bonakala, S.; Rudra, I.; Hohl, D.; Csányi, G., Equation of State of Fluid Methane from First Principles with Machine Learning Potentials. *Journal of Chemical Theory and Computation* **2019**, *15* (4), 2574-2586.
122. Broderick, K.; Lopato, E.; Wander, B.; Bernhard, S.; Kitchin, J.; Ulissi, Z., Identifying limitations in screening high-throughput photocatalytic bimetallic nanoparticles with machine-learned hydrogen adsorptions. *Applied Catalysis B: Environmental* **2023**, *320*, 121959.
123. Mortazavi, B.; Podryabinkin, E. V.; Novikov, I. S.; Rabczuk, T.; Zhuang, X.; Shapeev, A. V., Accelerating first-principles estimation of thermal conductivity by machine-learning interatomic potentials: A MTP/ShengBTE solution. *Computer Physics Communications* **2021**, *258*, 107583.
124. Vita, J. A.; Schwalbe-Koda, D., Data efficiency and extrapolation trends in neural network interatomic potentials. *Machine Learning: Science and Technology* **2023**, *4* (3), 035031.
125. Montes de Oca Zapiain, D.; Wood, M. A.; Lubbers, N.; Pereyra, C. Z.; Thompson, A. P.; Perez, D., Training data selection for accuracy and transferability of interatomic potentials. *npj Computational Materials* **2022**, *8* (1), 189.
126. Schwalbe-Koda, D.; Tan, A. R.; Gómez-Bombarelli, R., Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nature Communications* **2021**, *12* (1), 5104.
127. Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J., LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Computer Physics Communications* **2022**, *271*, 108171.
128. Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E., GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19-25.
129. Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D., OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS computational biology* **2017**, *13* (7), e1005659.
130. Fu, X.; Wu, Z.; Wang, W.; Xie, T.; Keten, S.; Gomez-Bombarelli, R.; Jaakkola, T., Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations. *arXiv e-prints* **2022**, arXiv:2210.07237.
131. Wen, M.; Afshar, Y.; Elliott, R. S.; Tadmor, E. B., KLIFF: A framework to develop physics-based and machine learning interatomic potentials. *Computer Physics Communications* **2022**, *272*, 108218.
132. Biersack, J. P.; Ziegler, J. F. In *The Stopping and Range of Ions in Solids*, Ion Implantation Techniques, Berlin, Heidelberg, 1982//; Ryssel, H.; Glawischnig, H., Eds. Springer Berlin Heidelberg: Berlin, Heidelberg, 1982; pp 122-156.
133. Chatterjee, P.; Sengul, M. Y.; Kumar, A.; MacKerell, A. D., Jr., Harnessing Deep Learning for Optimization of Lennard-Jones Parameters for the Polarizable Classical Drude Oscillator Force Field. *Journal of Chemical Theory and Computation* **2022**, *18* (4), 2388-2407.
134. Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.-W.; da Silva Santos, L. B.; Bourne, P. E., The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* **2016**, *3* (1), 1-9.

135. Stall, S.; Yarmey, L.; Cutcher-Gershenfeld, J.; Hanson, B.; Lehnert, K.; Nosek, B.; Parsons, M.; Robinson, E.; Wyborn, L., Make scientific data FAIR. *Nature* **2019**, *570* (7759), 27-29.
136. Scheffler, M.; Aeschlimann, M.; Albrecht, M.; Bereau, T.; Bungartz, H.-J.; Felser, C.; Greiner, M.; Groß, A.; Koch, C. T.; Kremer, K., FAIR data enabling new horizons for materials research. *Nature* **2022**, *604* (7907), 635-642.
137. Hjorth Larsen, A.; Jørgen Mortensen, J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Bjerre Jensen, P.; Kermode, J.; Kitchin, J. R.; Leonhard Kolsbjerg, E.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Bergmann Maronsson, J.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W., The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter* **2017**, *29* (27), 273002.
138. Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G., Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.* **2013**, *68*, 314-319.
139. Peters, I.; Kraker, P.; Lex, E.; Gumpenberger, C.; Gorraiz, J. I., Zenodo in the spotlight of traditional and new metrics. *Frontiers in research metrics and analytics* **2017**, *2*, 13.
140. Draxl, C.; Scheffler, M., The NOMAD laboratory: from data sharing to artificial intelligence. *Journal of Physics: Materials* **2019**, *2* (3), 036001.
141. Open Catalyst demo. <https://open-catalyst.metademolab.com/>.
142. Lan, J.; Palizhati, A.; Shuaibi, M.; Wood, B. M.; Wander, B.; Das, A.; Uyttendaele, M.; Zitnick, C. L.; Ulissi, Z. W., Adsorbml: Accelerating adsorption energy calculations with machine learning. *arXiv preprint arXiv:2211.16486* **2022**.

Author Biographies

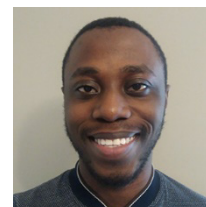
Tristan Maxson

Tristan Maxson is a Ph.D. student of chemical engineering at The University of Alabama in the Szilvási group and an active DOE Computational Science Graduate Fellow. His studies primarily concern catalysis, machine learning interatomic potentials, and code development for tools to perform atomistic simulations.



Ademola Soyemi

Ademola Soyemi earned his BEng in Chemical Engineering from Covenant University, Nigeria, in 2018 and is currently pursuing his PhD at The University of Alabama in the Szilvási group. His research has focused on the computational design of solvents, understanding catalytic reaction mechanisms, and developing machine learning interatomic potentials.



Benjamin Wei Jie Chen

Benjamin Chen is a senior research scientist at A*STAR's Institute of High Performance Computing. His research focuses on developing novel high-throughput computing methods and coupling them with machine learning to enable modelling of catalytic materials and reactions with high realism and fidelity. By doing so, he aims to increase the predictive power of computations, enhance their value and synergy with experiments, and accelerate the rational design of novel catalysts.



Tibor Szilvási

Tibor Szilvási is an assistant professor of chemical engineering at The University of Alabama. His research interests lie in computational catalysis and material design with current emphasis on understanding phenomena occurring at nanoparticle surfaces and solid-liquid interfaces. His group aims at developing robust computational workflows including machine learning interatomic potentials to gain new physical and chemical insights.

