The Journal of Chemical Physics

RESEARCH ARTICLE | OCTOBER 20 2023

ColabFit exchange: Open-access datasets for data-driven interatomic potentials

Special Collection: Software for Atomistic Machine Learning

Joshua A. Vita ; Eric G. Fuemmeler ; Amit Gupta ; Gregory P. Wolfe ; Alexander Quanming Tao ; Ryan S. Elliott ; Stefano Martiniani ; Ellad B. Tadmor

J. Chem. Phys. 159, 154802 (2023)
https://doi.org/10.1063/5.0163882

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

ColabFit exchange: Open-access datasets for data-driven interatomic potentials

Cite as: J. Chem. Phys. 159, 154802 (2023); doi: 10.1063/5.0163882
Submitted: 19 June 2023 • Accepted: 25 September 2023 •
Published Online: 20 October 2023

Joshua A. Vita,[1] Eric G. Fuemmeler,[2] Amit Gupta,[2] Gregory P. Wolfe,[3] Alexander Quanming Tao,[2] Ryan S. Elliott,[2] Stefano Martiniani,[3,4,5] and Ellad B. Tadmor[2,a)]

AFFILIATIONS

[1] Department of Materials Science and Engineering, University of Illinois Urbana-Champaign, Urbana, Illinois 61801, USA
[2] Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, USA
[3] Center for Soft Matter Research, Department of Physics, New York University, New York, New York 10012, USA
[4] Simons Center for Computational Physical Chemistry, Department of Chemistry, New York University, New York, New York 10012, USA
[5] Courant Institute of Mathematical Sciences, New York University, New York, New York 10112, USA

Note: This paper is part of the JCP Special Topic on Software for Atomistic Machine Learning.
[a)] **Author to whom correspondence should be addressed:** tadmor@umn.edu

ABSTRACT

Data-driven interatomic potentials (IPs) trained on large collections of first principles calculations are rapidly becoming essential tools in the fields of computational materials science and chemistry for performing atomic-scale simulations. Despite this, apart from a few notable exceptions, there is a distinct lack of well-organized, public datasets in common formats available for use with IP development. This deficiency precludes the research community from implementing widespread benchmarking, which is essential for gaining insight into model performance and transferability, and also limits the development of more general, or even universal, IPs. To address this issue, we introduce the ColabFit Exchange, the first database providing open access to a large collection of systematically organized datasets from multiple domains that is especially designed for IP development. The ColabFit Exchange is publicly available at https://colabfit.org, providing a web-based interface for exploring, downloading, and contributing datasets. Composed of data collected from the literature or provided by community researchers, the ColabFit Exchange currently (September 2023) consists of 139 datasets spanning nearly 70 000 unique chemistries, and is intended to continuously grow. In addition to outlining the software framework used for constructing and accessing the ColabFit Exchange, we also provide analyses of the data, quantifying the diversity of the database and proposing metrics for assessing the relative diversity of multiple datasets. Finally, we demonstrate an end-to-end IP development pipeline, utilizing datasets from the ColabFit Exchange, fitting tools from the KLIFF software package, and validation tests provided by the OpenKIM framework.

I. INTRODUCTION

Leveraging modern computing infrastructures, high-throughput pipelines for density functional theory (DFT) calculations have been able to produce results for millions of atomic configurations spanning a wide range of chemistries and applications.[1–6] These methods have led to the creation of a number of massive datasets of first principles calculations, such as the Materials Project[7] and the OpenCatalyst Project,[8,9] among others,[10–13] which have served as critical resources for materials discovery and interatomic potential (IP) development. While these repositories have proven extremely useful, there still exist opportunities for continued development and dissemination of datasets specifically tailored to fit the needs of developers of data-driven (DD) interatomic potentials (IPs). In particular, datasets intended for use with IP development typically include a variety of non-equilibrium atomic configurations or hand-selected structures depending on the target application. Furthermore, datasets intended for fitting data-driven interatomic potentials (DDIPs) are often carefully pruned and refined to enable the models to efficiently learn the physical behaviors relevant for the accurate prediction of a given material property, and to achieve stable simulations. Conversely,

existing databases of quantum mechanical (QM) calculations focus predominantly on stable equilibrium structures relevant to material discovery. Even in the case of databases that do contain portions of the data that may be suitable for use in DDIP fitting, they are rarely organized in a way that facilitates model benchmarking or targeted analysis of model behavior across chemical compound space.

In addition to the issues of content and structure of existing QM calculation databases, common methods for organizing and distributing DDIP training datasets, such as the use of personal Github repositories,[9,14–17] Figshare[18–22] or Zenodo[23–26] uploads, or other file sharing methods are inconsistent and not conducive to interpretability and interoperability of the datasets. Datasets stored in this manner often use custom formats (Extended XYZ, HDF5, VASP OUTCARs, CSV, JSON) depending upon the specific research group that generated them, and despite government insistence[27,28] typically lack metadata necessary for interpretability and reproducibility of the data (missing units, unspecified DFT settings, undocumented inconsistencies in data structure). Unfortunately, even this limited approach for sharing data is pursued by only a handful of researchers, with the vast majority of DDIP datasets being entirely inaccessible to the general public or made available through private correspondence "upon reasonable request," without always honoring such requests. The end result is a significant decrease in reproducibility of published results and the effective loss of nontrivial amounts of effort and computational time spent on data generation, inevitably hindering scientific progress.

The notion of a FAIR (findable, accessible, interoperable, and reusable) data framework reflects a growing effort in the materials and chemistry communities to address these issues and foster the open exchange of materials and chemical data.[29] A FAIR database of datasets designed for DDIP training would help to facilitate collaboration and drive innovation, but must necessarily address a few key issues in order to succeed. Specifically, it must: (1) define a consistent, efficient, and standardized method for storing the data; (2) enable the organization of the data into meaningful, well-documented groupings; and (3) provide tools for easily accessing and contributing to the database in order to promote community engagement. In this work, we outline a standard for constructing FAIR databases of first-principles calculations, and use it to construct the ColabFit Exchange, the first database of open-access DDIP training datasets. We will detail the data structure of the resulting database, summarize its content, and demonstrate the use of tools for identifying and characterizing regions of configurational and compositional space sampled by existing datasets. By serving as a centralized, standardized, and open-access hub for DDIP datasets, the ColabFit Exchange provides the community with a unique opportunity to begin performing large scale analyses of model performances and dataset qualities that were previously infeasible for most researchers.

II. STRUCTURE

In order to facilitate the construction of organized datasets, and to ensure that the underlying data is stored in an efficient manner, we develop a hierarchical data storage standard (outlined in Fig. 1) comprising seven core components that we describe in detail in this section. Each of these components is implemented in the `colabfit-tools` software package[30] following an object-oriented design scheme. In this section we will give examples of how the ColabFit Data Standard can be applied to construct a database of atomistic ground-truth datasets, as this is the primary task which the ColabFit project aims to address. It is important to note, however, that the data standard is designed to be sufficiently flexible for adaptation to many other scientific domains where data-driven approaches are of interest.

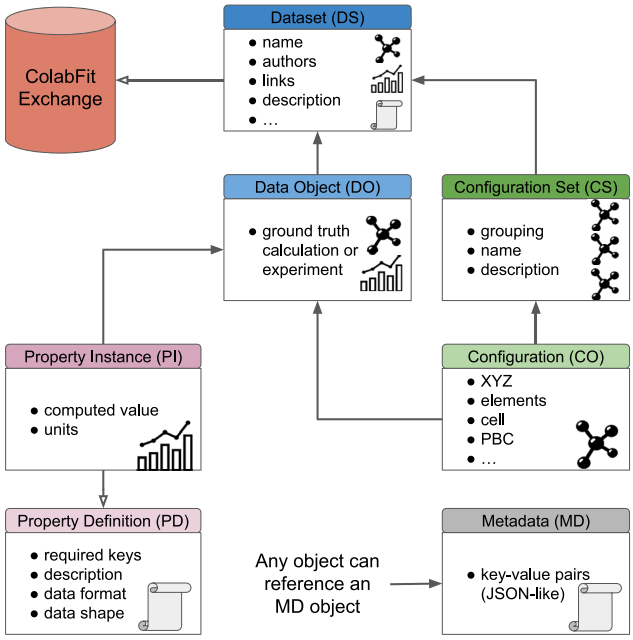

FIG. 1. A diagram of the ColabFit Data Standard, which defines the structure of the ColabFit Exchange. The standard comprises seven component types, which can be roughly grouped into three categories (with acronyms defined in the figure): primitive components (PI, CO) for storing input/output data, organizational components (DS, CS, DO) for creating meaningful groupings of lower-level components, and informational components for providing required (PD) or optional (MD) documentation of arbitrary components. Arrows between components specify relationships, e.g., a CO references CS and DO components). Open arrowheads denote many-to-one relationships, while filled arrowheads represent many-to-many relationships. For example, multiple CSs may reference multiple DSs, but each PI references only one PD.

A. Low-level components (COs and PIs)

The two fundamental building blocks of the ColabFit Data Standard are Configurations (COs) and Property Instances (PIs). Each CO stores a representation of an elementary object of interest and typically serves as input ($\mathbf{x}$) to a DD pipeline. PIs, on the other hand, store instances of property measurements associated with COs and typically serve as predictive targets ($\mathbf{y}$). For the examples outlined below, these will be atomistic configurations and target property values measured through ground-truth calculations or through experiments.

Broadly speaking, a CO subclass must define two critical functionalities: (1) it must define a list of keys whose values are used to generate a hash for comparing CO objects, and (2) it must define two

functions, one for generating a dictionary of information summarizing the contents of the CO, and another specifying how information from a *set* of COs may be aggregated into a single dictionary. These summary and aggregation functions will be called by higher-level objects to gather information about groups of COs. For example, in the case of an atomic configuration, the atom types, Cartesian coordinates of the atoms, cell vectors, and periodic boundary conditions would all be required to uniquely distinguish between two COs. A summary dictionary for an atomic configuration could include information such as the number of atoms in the cell, the chemical formula, the periodicity of the cell, or any other information deemed useful by the curators of the dataset. These traits enable the development of workflow pipelines for aggregating information about groups of configurations up to a higher-level component (see Sec. II C), which in turn aid in the construction of rich and efficiently queryable metadata.

Notably, the ColabFit Exchange currently makes the assumption that a given database is used to store only one type of CO at a time (e.g., only atomic configurations) in order to simplify the data aggregation process. This assumption may be relaxed in the future, depending upon the needs of the community. Using the ColabFit Data Standard to construct a database for data other than atomistic property predictions (e.g., property prediction for biomolecules specified by sequences whose characters span 20 naturally occurring amino acids) will typically involve writing a new CO subclass specification, with required keys matching the application of interest and custom aggregation functions.

Whereas COs store the input, PIs store the "ground-truth" output. Importantly, a PI contains a *single* computed property (and its units), such as the potential energy of the system or the atomic forces, rather than all of the properties associated with a given calculation. The decision to separate each property into its own PI allows for more efficient data storage, as it means that duplicate documents do not need to be stored in the database even in the case where two calculations have only a subset of matching properties (e.g., DFT calculations of two different single-atom primitive cells of ground-state crystals, which would both have zero forces, but will likely have different energies). Furthermore, this design choice allows PIs to be added or modified independently of the corresponding COs, which helps to simplify the process of cleaning and modifying datasets. In practice, a PI is a dictionary of key–value pairs for storing computed or measured properties and their associated units, plus some basic functionality for unit conversion and hashing. All PIs are required to point to exactly one Property Definition object in order to properly document the structure and contents of the PI (see Sec. II B for more details).

B. Informational components (PDs and MDs)

With the goal of encouraging reproducibility and ensuring that all of the data stored within a ColabFit database is well-documented,[27,28] Property Definition (PDs) and Metadata (MD) objects can be used to enforce structure in the data and provide additional information about each object.

All PIs are required to point to exactly one PD, which serves as an explicit, computer-readable definition (schema) of the contents of the PI following the Knowledgebase of Interatomic Models (KIM) Property Definition specification.[31] The most important benefit of PDs is that they improve the homogeneity of the database by ensuring that all properties of the same type are stored in the same format. PDs specify all of the keys available in the PI; for each of these keys, the PD will also specify if the key is required/optional, the data type of the corresponding value, the shape of the data (i.e., scalar, vector, tensor, . . .), if the value has units, and a brief description of the data. The KIM PD specification also supports uncertainty information for stored values, which may be included in ColabFit in the future.

A simple atomistic property example is the potential-energy PD, which has the keys "energy" (the potential energy of the system; required, float, scalar, has units), "per-atom" (if the energy has been divided by the number of atoms in the CO; required, boolean, scalar, unitless), and "reference-energy" (the value, if any, which has been subtracted from the "energy" value; optional, float, scalar, has units). As is the case with COs and PIs, by storing the PD as its own object rather than attaching the data directly to each PI, we are able to avoid duplicating data unnecessarily while still maintaining proper documentation of the PI contents.

While PDs serve as mandatory documentation of the contents of a PI, MD objects can be used to store optional additional information about objects of any type. MD objects can be any valid JSON dictionary, and are intended to be sufficiently flexible for storing data that does not fit naturally into any of the other object types. One of the most common applications of MD objects for constructing a DFT database would be to store pointers to raw input/output files (e.g., INCAR/OUTCAR files from VASP[32]) or additional information regarding simulation settings. Best practice would be to use MD objects to ensure that sufficient information is provided to reproduce any calculation in the database. In addition to improving reproducibility, proper use of MDs can also be valuable for identifying when datasets were computed using different settings or levels of theory, which can be important for transfer learning tasks[12,33] and can inform on when datasets may, or may not, be used in conjuction with each other for model training. Generally, the contents of MDs are not expected to be queryable, as available keys may vary drastically between datasets, though in some cases we found it useful to manually parse the MDs to improve the quality of common queries over COs or PIs (e.g., descriptive labels on COs, or levels of theory used for computing PIs).

C. Organizational components (DOs, CSs, and DSs)

Given that the ColabFit Data Standard is meant for constructing databases for data-driven model development, it obviously must allow for the data to be organized in meaningful and useful ways. Data Objects (DOs), Configuration Sets (CSs), and Datasets (DSs) facilitate this by defining higher-level groupings of lower-level objects.

A DO is perhaps the simplest of these groupings—it defines relationships between one or more COs with one or more PIs. Conceptually, DOs should be used to link inputs and outputs of a given calculation or measurement. For example, a DFT calculation would typically produce both an energy PI and an atomic forces PI, which could be grouped under a single DO that also points to the corresponding CO and details of the calculation in an MD. A more complex example would be a nudged elastic band calculation,[34] where it would be necessary to define a relationship between

a computed energy barrier (a PI) and multiple images interpolating between the start/end transition states (each stored as their own CO).

Another object, which we observe is particularly useful in practice for improving data interpretability, is the CS. A CS defines a grouping (and optionally, an ordering) over one or many COs, and allows a user to give a name and a description to that grouping. Generally, CSs should be used for organizing configurations into groups that will help end-users better understand the contents of the dataset. In the materials and chemical sciences, it is common for dataset developers to organize their data based on attributes such as molecule type, physical structure, or method of generation.[35–37] For example, molecular dynamics or relaxation trajectories are often grouped together by DDIP developers. Similar methods can be useful in other deep learning fields, such as with the Modified National Institute of Standards and Technology (MNIST)[38] or CIFAR-10[39] datasets where the data are naturally grouped by class. Such groupings make it easier for users of the datasets to understand the contents of the dataset, facilitate filtering, and improve interpretability of the behaviors of models trained to the data.

The highest level object (aside from a database itself) is a DS, which matches the canonical meaning of the word: a collection of data points and any associated metadata. Similar to how a CS defines a collection of COs, a DS defines a collection of DOs and CSs, and includes additional metadata such as a name, list of authors, relevant links, and a description. Notably, a DS references CSs rather than COs directly in order to ensure that any organizational structure imposed by the CSs is reflected in the DS as well. The DS serves as a complete, well-documented, and queryable representation of a collection of computed values and their corresponding inputs, and is intended to be packaged and distributed as a self-contained object to facilitate reproducibility, standardized benchmarking, and collaboration. All DSs currently in the ColabFit Exchange are assigned unique DOIs for tracking citations and can be downloaded at https://colabfit.org as extended XYZ files in a standardized format.

D. Additional technical details

Two important features of the ColabFit Data Standard are the abilities to store the data in an efficient and queryable manner, and to aggregate low-level information in order to generate information-rich, high-level metadata. While part of this functionality is achieved through careful separation of data objects into their constituent parts (PIs, COs, PDs, and MDs), it also depends upon a few other technical details discussed in this section.

First, hashing functions are used to generate unique IDs for every component in the database; these digest specified contents of each component and return a hexadecimal string. The contents of a component that are digested in order to generate the hash vary depending on the component's type: MDs directly hash their entire contents; PIs hash their computed values and units; COs hash the contents of their required keys. The hashes for higher level components (DOs, CSs, and DSs) are generated by hashing the IDs of all of their sub-components. For example, a CS's ID is a hash of the list of the IDs of all COs grouped by the CS. PDs are the only components which do not use hashes for their unique ID, but instead are given user-specified names, as there are relatively few PDs and it is important for their IDs to be human-readable. This hashing avoids the issue of duplicate entries (those whose content is identical within machine precision) when users re-upload portions of existing datasets or coincidentally generate the exact same data as another author (a relatively common occurrence in the materials and chemical sciences).

Second, aggregation pipelines were developed for building metadata for high-level objects (CSs and DSs). Although some metadata is stored on CS/DS objects directly, other information must necessarily be propagated up from the CO/PI level; for example, information such as the total number of atoms contained within a CS, the chemical formulas present, or the relative concentrations of elements. In order to enable this type of data aggregation, low-level components (COs and PIs) provide functions for returning "summaries" of their contents, which are key–value dictionaries summarizing any additional information of interest that the database authors think might be useful. The low-level components also provide functions for merging lists of metadata dictionaries into a single dictionary. Database developers may adjust the behaviors of these summary and aggregation functions depending on their needs and target applications. This aggregated metadata greatly improves the queryability and interpretability of the data, and helps to build a database that can be more easily used by model developers for drawing insights about their data.

E. Comparison to OPTIMADE

In order to simplify the process of understanding the design choices made in this work, we compare the ColabFit Data Standard outlined above to the OPTIMADE API,[40] which is a broad effort from researchers across many domains of materials science to develop interoperable databases of materials data. Although the ColabFit Exchange is not yet OPTIMADE-compliant (which is a future goal of the work), many parallels can be drawn between the components described in Fig. 1 and objects from the OPTIMADE API. Given the ubiquity within the community of the need for representing atomic configurations, it is unsurprising that the CO object described in Sec. II A contains all of the information necessary to define a Structures object in the OPTIMADE API, and could be easily made to match with some additional processing (i.e., storing various chemical formulas, or re-formatting certain fields to fit the OPTIMADE specifications). The ColabFit Standard PD and PI components roughly correspond to OPTIMADE Property Definition and Calculation objects, though the two standards begin to diverge in the specific details of these components. For example, PDs allow for specifying units, whereas the OPTIMADE Property Definition does not, and PIs are required to be associated with a PD while OPTIMADE Calculation objects are not. The largest discrepancies between the two standards arise from the higher level components described in Sec. II C, where ColabFit's need for defining groupings over objects (e.g., CSs as groups of COs, and DSs as groups of CSs and DOs) are not well-supported by the current OPTIMADE API. Although possible workarounds exist in order to represent a DO/CS/DS using existing OPTIMADE objects (e.g., with relationships), such constructions would have been inefficient and lacking in many of the desired functionalities of DOs/CS/DSs. There is, however, a current effort within the OPTIMADE community to support trajectory-like objects (groups of Structures, intended for storing

TABLE I. Counts of objects of interest in the ColabFit Exchange, excluding the data from the OpenCatalyst datasets. These values do *not* double count in the case where there exist duplicates of a given object (e.g., when an identical configuration was uploaded in multiple datasets, or an author is credited on multiple publications). Here, a "chemical system" refers to a set of unique constituent atom types.

Objects	Count
Datasets	139
Configuration sets	459
Data objects	11 185 734
Configurations	10 752 923
Atoms	512 108 838
Chemical systems	68 474
Publications	79
Authors	323

simulation trajectories) which, once fully implemented, will more easily support the needs of the ColabFit Exchange.

III. OVERVIEW

Table I provides a summary of the contents of the ColabFit Exchange, which is currently (September 2023) composed of 139 unique datasets contributed by their authors or gathered from the literature. These datasets are further broken down into 459 configuration sets, which can be readily combined, split, or grouped in order to define new datasets based on the needs of the community. In total, the ColabFit Exchange contains over 11×10^6 DOs, corresponding to $\sim 28 \times 10^6$ computed properties. Note that the OpenCatalyst datasets (which are included in the ColabFit Exchange) are not included in these summary statistics, as they are already well-documented elsewhere in the literature[8,9] and their

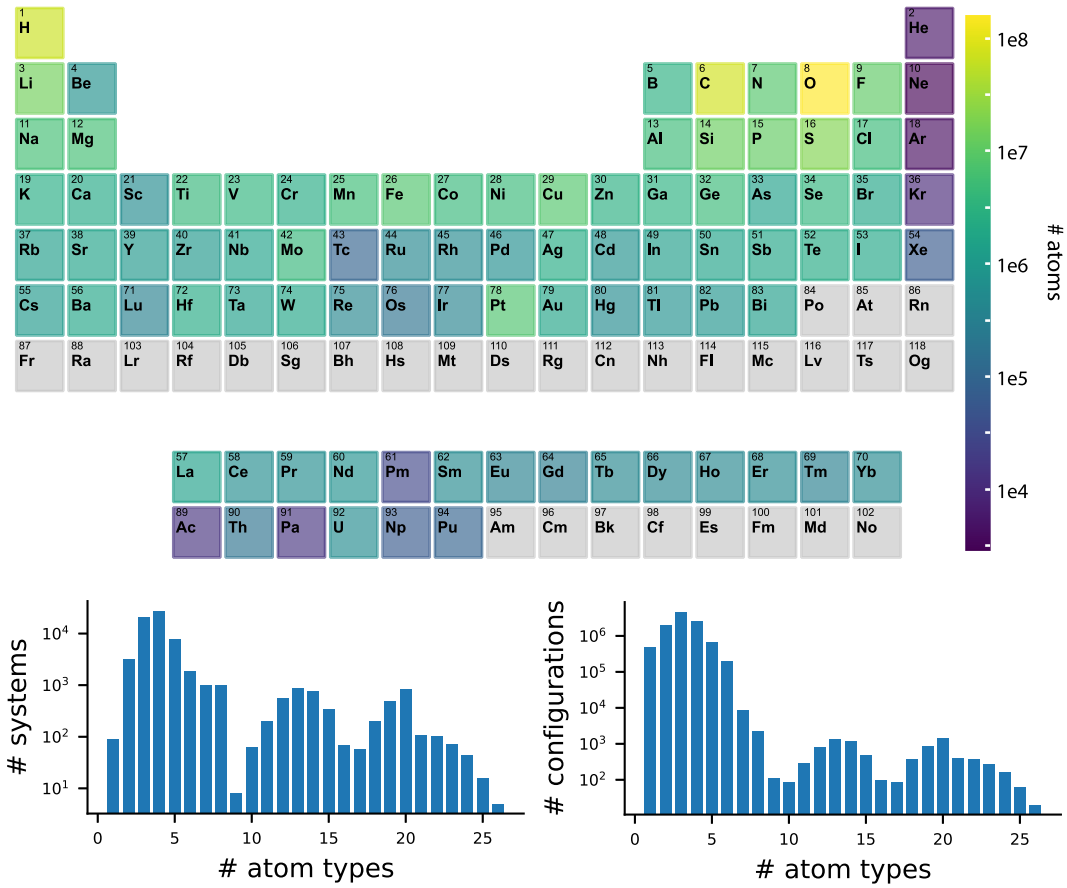

FIG. 2. Chemical composition of the ColabFit Exchange, spanning 89 of the 118 elements on the periodic table, for a total of 68 474 unique chemical systems. After excluding the OpenCatalyst data (which is not represented in this figure), the majority of the database is composed of organic molecules (C, H, and O alone make up ~60% of the data shown in this figure) due to the relative popularity and availability of molecular datasets. There is currently no data for elements with atomic numbers between 84 and 88, or greater than 94. The bottom panel shows histograms of the number of unique chemical systems (left) or configurations (right) present in the ColabFit Exchange for different numbers of atomic types (i.e., the number of unary/binary/ternary/. . . systems or configurations). The HME21 dataset[20] accounts for the majority of the data with large numbers of atom types; without HME21, all systems have fewer than ten atom types.

large sizes (~134 × 10^6 DOs for OC20) would obscure the results from the other datasets. As the ColabFit Exchange continues to grow, updated statistics summarizing its contents can be found at https://colabfit.org.

The ~11 × 10^6 atomic configurations (for a total of 512 × 10^6 atoms) spanning nearly 70 000 chemical systems can be further analyzed based on their chemical composition, as shown in Fig. 2. Here, a "chemical system" is defined as a set of unique constituent atom types, e.g., C, C–H, C–H–N, . . . , and is indicative of the types of chemistries explored within the ColabFit Exchange. Though single element datasets are the most common (see Fig. 3), 95% of the configurations in the ColabFit Exchange include at least two elements, meaning the ColabFit Exchange may be used as a starting point for the development of many multi-element models. Much of the multi-element data comes from larger datasets designed for the construction of "universal" IPs intended to model all relevant types of atomic interactions,[41–43] such as the Materials Project trajectory dataset,[43] and others from the literature.[20,42,44] By providing access to all of these datasets within a unified framework, the ColabFit Exchange will simplify the process of constructing training datasets for new chemical systems that have not yet been explicitly sampled by the datasets currently in the ColabFit Exchange.

The values in Table II provide a further breakdown of the most prevalent computed properties stored within the ColabFit Exchange that are available for supervised training. Energies are the most commonly computed property, followed by forces. Note that the energy counts in Table II are a sum over the four types of energy PIs specified by the publications associated with the datasets in the ColabFit Exchange (potential, free, atomization, and formation energy), where each energy type is given its own PD. Note, the raw number of force PIs shown in Table II does not reflect the total number of individual atomic force vectors in the ColabFit Exchange—the number of individual force vectors is much higher, approximately equaling the number of atoms in the database multiplied by the fraction of DOs that contain an atomic force PI (90%). Stresses are available for only about half of the DOs in the ColabFit Exchange, with the majority coming from the Materials Project (MP) trajectory dataset.[43] The ColabFit Exchange also includes, for subsets of the data, additional properties that are supported within the framework as their own PDs but are less relevant to DDIP development. These additional properties include indirect and direct band gaps, magnetization, atomic charges, polarizability, dipole moments, and a large collection of common molecular properties from datasets like those derived from GDB-17.[45]

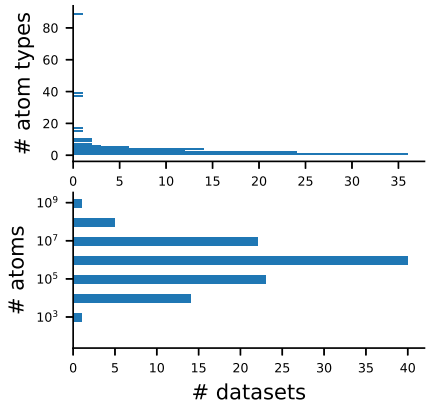

FIG. 3. Histogram showing the sizes of the datasets currently in the ColabFit Exchange. The distribution of the total number of atoms summed over all COs in a given dataset is Gaussian-like, centered about a mean of 10^6.

TABLE II. Counts of property instances in the ColabFit Exchange, excluding the data from the OpenCatalyst datasets. These values *do* double count in the case where two identical copies of a property exist (e.g., two distinct configurations were uploaded with identical potential energies) in order to accurately reflect the number of target values in the ColabFit Exchange. Though many of the datasets currently in the ColabFit Exchange contain more computed properties than the three shown here, energies, forces, and stresses are the three that are predominantly used for training DDIPs.

Property instance (PI)	Count
Energy	11 293 268
Atomic forces	10 102 772
Cauchy stress	6 729 342
Total	28 125 382

At the dataset level, Fig. 3 shows that the ColabFit Exchange has a wide range of dataset sizes, both in terms of the total number of atoms and the number of unique atom types contained within a given dataset. Though single element datasets are the most common, these datasets are typically smaller than multi-element datasets. The three datasets with greater than 20 atom types are HME-21,[20] the Materials Project trajectory dataset,[43] and the elpasolite crystal dataset.[46] The number of molecular datasets vs the number of condensed matter datasets is roughly evenly split (51 molecular, 50 condensed matter, and five mixed), though the molecular datasets usually include significantly more atomic configurations due to their smaller number of atoms per configuration.

IV. APPLICATIONS

A critical step towards improving DDIP design and efficiently constructing models for specific applications is to gain a better understanding of what regions of composition and configuration space have, or have not, been sampled by existing datasets. As the ColabFit Exchange is the first attempt at curating an exhaustive list of DDIP-fitting datasets, it provides a unique opportunity for performing this type of analysis. Towards this end, in this section we explore the use of tools for identifying and characterizing regions of overlap between two datasets. Furthermore, we demonstrate how the ColabFit Exchange can integrate with other model fitting and validation tools to create an end-to-end fitting framework.

A. Comparing atomic environments

In order to compare configurations between datasets, it is convenient to first define a method for obtaining a vector representation of the atomic environments in the configurations (which is invariant to permutations, rotations and translations). This can be done using several well-documented local "descriptors," such as the Atom-Centered Symmetry Functions (ACSF)[47] or Smooth Overlap of Atomic Positions (SOAP)[48] descriptors, among others.[49–51]

However, given the quadratic scaling of the sizes of local environment descriptors with the number of atom types, this rapidly becomes intractable when performing database-wide analyses, as is the goal here. We instead choose the descriptor to be a learned-representation, i.e., intermediate vectors generated by a pre-trained graph-based model. For this task, we chose to use the M3GNet universal potential,[42] which has been previously trained to a subset of the Materials Project relaxation trajectory dataset. The learned representation is taken from the final layer of the M3GNet model prior to the regression head, which has a size of $N_{atom} \times 64$, regardless of the number of chemical species in the atomic configuration. These $N_{atom} \times 64$ matrices are then averaged over N_{atoms} in order to produce a single length-64 vector for each atomic configuration. Unified Manifold Approximation and Projection (UMAP) visualizations of these configuration-averaged M3GNet representations are shown in Fig. 4.

B. Delaunay Component Analysis (DCA)

While visualizations like those shown in Fig. 4 are commonly used for obtaining a qualitative understanding of the contents of a dataset, and often provide advantages over methods like principal component analysis (PCA), the use of UMAP (or tSNE[52]) makes it challenging to obtain quantitative metrics since distances are not preserved between the original and embedded spaces. In order to obtain a more quantitative understanding of the relationships between datasets, we explore the recently developed Delaunay

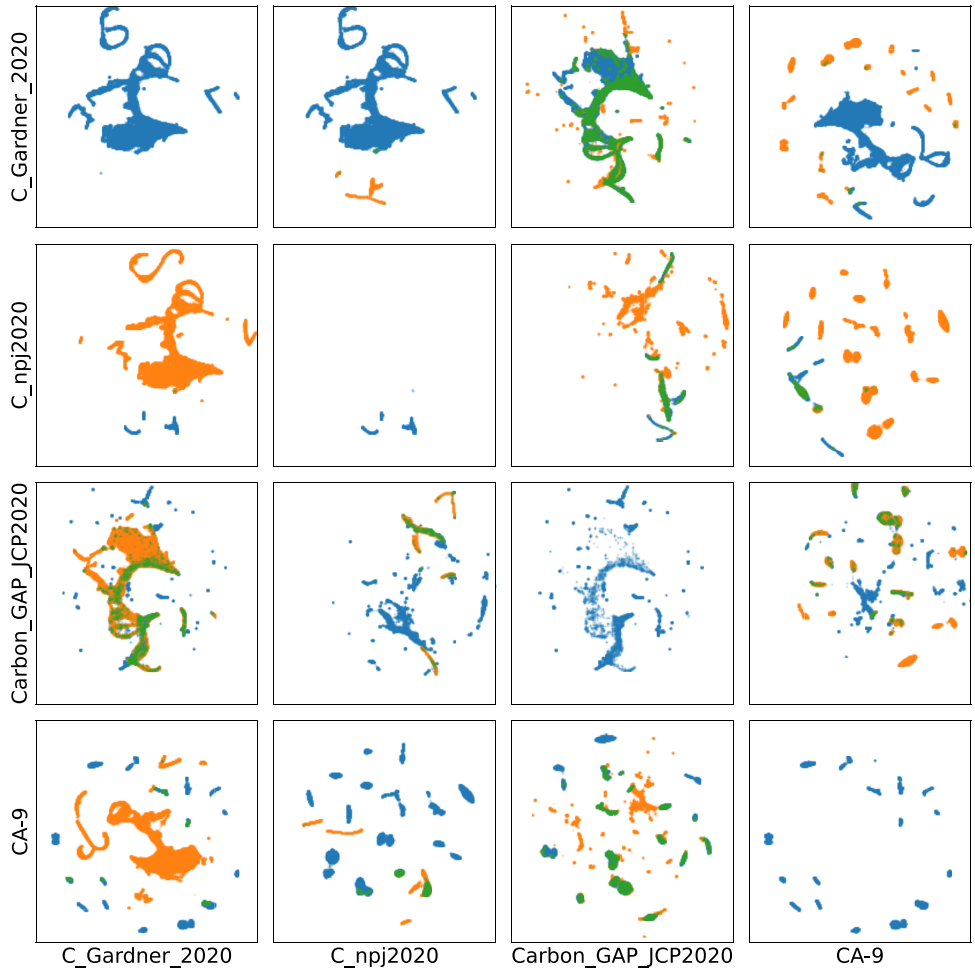

FIG. 4. Visualizations of the configurations in the C_Gardner_2022, C_npj2020, Carbon_GAP_JCP2020, and CA-9 datasets in relation to each other. Plots are generated by applying UMAP to configuration-averaged descriptors extracted from the M3GNet model, as described in Sec. IV A. Row labels denote the "reference" dataset used for DCA in Sec. IV B, which are colored blue in each panel. Column labels denote the "evaluation" dataset, and are colored orange. To help highlight regions of overlap, points from the reference dataset have been colored green if there is at least one point from the evaluation dataset within a chosen threshold value. Panels along the diagonal correspond to only the reference set, in order to help guide visual comparisons to the other panels in the same row. Note that UMAP embeddings were performed individually for each panel, including only the two datasets within that panel. This means that the embeddings may not be identical even for the same dataset across rows or down columns.

Component Analysis (DCA) technique[53] to quantify the overlap between two datasets. Originally intended for comparing between the manifolds of two learned representations of the same data, we instead apply DCA here to the separate, yet related, task of comparing two datasets under the same representation (i.e., the learned M3GNet latent vectors). Though we provide a brief summary of the DCA method here, for a more thorough explanation we refer the reader to Ref. 53. Some additional analysis of DCA as it relates to this work can be found in the supplementary material. The DCA analysis shown in this section uses the code provided by Ref. 53, which is included in the colabfit-tools package alongside a growing set of tools for dataset analysis organized under the colabfit-analyze sub-package.

The goal of DCA is to derive metrics quantifying the degree of overlap between two manifolds, where one manifold is defined by points in a "reference" dataset, and the other manifold is defined by points in an "evaluation" dataset. In this case, the manifolds exist in the 64-dimensional latent space of the M3GNet model from which we extracted the descriptors, and represent the phase spaces sampled by each dataset. DCA constructs an approximate Delaunay graph (known as the "dual graph" of a Voronoi diagram, where the circumcenters of triangles in the Delaunay graph are the vertices of the corresponding Voronoi diagram) of the manifolds, then distills the graph into connected components, i.e., robust sub-graphs, using a minimum spanning tree. Vertices in the Delaunay graph correspond to data points from the reference or evaluation datasets; edges link points which are "natural neighbors" of each other (i.e., they have adjoining Voronoi cells). Connected components are sub-graphs representing clusters in the representation space, and may be composed of a mix of vertices from both the reference and evaluation datasets. Note that DCA does not modify the representations of the configurations (descriptors) in any way, so it inherits all attributes of the M3GNet descriptor (e.g., invariance to rotations of configurations, learned embeddings of atomic types, etc.). Using the distilled components, DCA then evaluates a "consistency" (c) and "quality" (q) score for each component, defined as:

$$c(\mathcal{G}_i) = 1 - \frac{\left||\mathcal{G}_i^R|_v - |\mathcal{G}_i^E|_v\right|}{|\mathcal{G}_i|_v}$$
$$q(\mathcal{G}_i) = \begin{cases} 1 - \dfrac{(|\mathcal{G}_i^R|_\varepsilon + |\mathcal{G}_i^E|_\varepsilon)}{|\mathcal{G}_i|_\varepsilon} & \text{if } |\mathcal{G}_i|_\varepsilon \geq 1, \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathcal{G}_i$ is the Delaunay graph of component i, and $|\mathcal{G}_i^R|_v$ and $|\mathcal{G}_i^R|_\varepsilon$ denote the cardinalities of the vertex and edge sets of $\mathcal{G}_i$ restricted to dataset R, respectively. Conceptually, consistency measures how evenly represented each dataset is within a component, while quality measures how well mixed the datasets are in a component. The local metrics of consistency and quality, which are computed individually for each component, can then be used to identify "fundamental" components (those with both high consistency and high quality) in order to calculate global metrics of "precision" p and "recall" r between the two datasets, defined as:

$$p = \frac{|\mathcal{F}^E|_v}{|\mathcal{G}^E|_v} \quad \text{and} \quad r = \frac{|\mathcal{F}^R|_v}{|\mathcal{G}^R|_v}, \quad (2)$$

where $\mathcal{F}^E$ and $\mathcal{F}^R$ refers to the sub-graphs of the evaluation and reference datasets, respectively, which are contained within a fundamental component. Intuitively, the precision measures the fraction of points from the evaluation dataset which overlap with the reference dataset. Recall measures how well the reference dataset is represented by the evaluation dataset. A high precision score means that the evaluation dataset is well contained by the reference dataset; a low recall means that the reference dataset includes data which is not well-represented by the evaluation set. These definitions of precision and recall are similar to those commonly used in other deep learning tasks for quantifying the degree of overlap between two distributions, though the use of "fundamental components" is a valuable modification unique to DCA which helps apply the metrics to manifold analysis.

As a demonstration of the utility of the global metrics of precision and recall, we perform DCA using four datasets from the ColabFit Exchange which include only pure carbon data: C_Gardner_2020,[17] C_npj2020,[54] Carbon_GAP_JCP2020,[55] and CA-9.[56]

The C_Gardner_2020 dataset contains DDIP-computed molecular dynamics trajectories of a melt/quench/anneal process; C_npj2020 has a relatively narrow focus, with an emphasis on monolayer and bilayer graphene, diamond, and graphite structures; Carbon_GAP_JCP2020 contains a wide variety of carbon systems, e.g., bulk, liquid, nanotubes, fullerene, graphene, etc.; and, finally, CA-9[56] has DFT-computed molecular dynamics trajectories of nine carbon allotropes (diamond, lonsdaleite, graphene, haeckelite, SWCNT, fullerene, cumulene, carbyne, and amorphous C). We extract the configuration-averaged M3GNet representations for each dataset, as described in Sec. IV A, and use these as the representations for DCA.

The precision scores reported in Fig. 5 immediately provide quantitative insights which match our intuitions based on the UMAP visualizations in Fig. 4 and our knowledge of the physical environments sampled by each dataset. For example, the DCA-computed precision scores validate our expectations that Carbon_GAP_JCP2020 is the most diverse (highest row-average in Fig. 5) of the four datasets, and that C_npj2020 is well captured by most of the other datasets (highest column-average). The precision scores also allow us to make additional useful observations, e.g., C_Gardner_2020 is largely distinct (low precision and low recall) from both C_npj2020 and CA-9, which is supported by the minimal overlap seen Fig. 4.

These types of insights can be extremely valuable to DDIP dataset developers when designing test sets or seeking to merge existing training sets to fit a more general model. For example, when merging a new training set into an existing one, a low precision score indicates that the new data is introducing new information into the training set. Similarly, a high recall score indicates that the new data may be over-sampling regions of configurational space that are already well-represented by the existing data, therefore leading to an effective increased weighting of those regions of space on the loss function which can affect model performance and training metrics. Furthermore, precision and recall scores could help to identify more suitable test sets, where it may be desirable that the test set have low precision and high recall (e.g., to detect possible overfitting), low precision and low recall (e.g., to test model generalizability/zero-shot capacity), or any range of values in between these limits depending

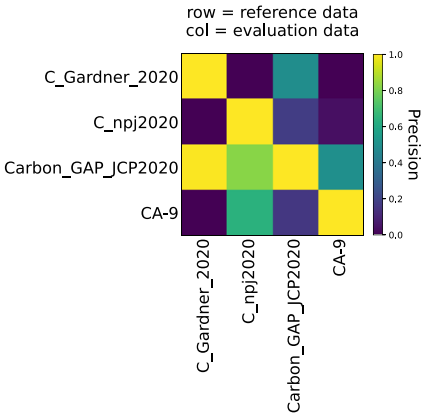

FIG. 5. Precision scores obtained by DCA comparing the four datasets from Fig. 4 to each other. A high precision score means that the evaluation dataset (column labels) is well-contained within the reference dataset (row labels). A high recall score (which corresponds to the transpose of this matrix) means that the evaluation dataset provides good sampling of all components of the reference dataset.

upon the goal of the test. Use of DCA, or related metrics, can provide a more systematic approach to dataset construction, which can help to address the known issues of high redundancy and correlation in DDIP training sets and materials data,[57–59] and will likely be essential moving forward in the field to ensure that datasets are not inhibiting the ability of researchers to properly assess model generalizability. We would like to emphasize that DCA is just *one* example of a method which could lead to better dataset design – other techniques (e.g., dataset roughness,[60] information imbalance,[61] or entropy-based metrics[62]) may be equally valuable, and should be further developed alongside the ColabFit Exchange. Importantly, because the ColabFit Exchange houses an ever-growing number of diverse datasets, it can help facilitate large-scale benchmarking and analysis of new methods (such as DCA), and provide insights across many unique datasets.

C. Example fitting workflow

In order for the ColabFit Exchange to be usable in practice, it is important that the datasets be easily accessed and interacted with by a variety of DDIP fitting frameworks.[15,63–65] While this is achievable by writing simple I/O operations for exporting datasets from the ColabFit Exchange as extended XYZ files, then re-formatting to integrate with external software, a more streamlined approach would be one which operates directly on the native ColabFit Exchange data structures and ties in with necessary simulation and validation packages. ColabFit Exchange datasets can be utilized for end-to-end DDIP development entirely within the KIM ecosystem, taking advantage of existing tools such as KLIFF[66] for model training, and OpenKIM[67–70] for model testing, archiving, and deployment. As an example of such an end-to-end workflow, we use KLIFF to train a spline-based MEAM potential[71,72] for lithium (Li) using the mlearn-Li training dataset, which has been used along with its other elemental counterparts for model benchmarking.[73,74] KLIFF supports seamless loading of ColabFit Exchange datasets, training of physics-based IPs and arbitrary machine learning DDIPs based on the PyTorch library,[75] and exporting of KIM-compliant models that can then be seamlessly deployed to a variety of molecular simulation packages that support the KIM standard including ASE,[76] DL_POLY,[77] GULP[78] and LAMMPS[79] (see Ref. 80 for a full list).

We fit the spline-based MEAM potential to energy and forces utilizing seven knots per spline and an inner and outer cutoff radius of 2.4 and 5.1 Å, respectively. The model achieved training (testing) set energy and force RMSEs of 1.55 (1.65) meV/atom and 0.049 (0.046) eV/Å, respectively. Additional material property predictions of the trained potential can be seen in Table III. The potential performs well across all computed properties, with the largest relative errors being those of surface energy predictions [0.196 for

TABLE III. Computed lattice constant (a), elastic constants (c_{ij}), bulk modulus (K), vacancy formation, migration and diffusion activation energies (E_v, E_m, E_a), and surface energies (E_s) of bcc Li using the spline-based MEAM potential and DFT. Relative errors between MEAM and DFT values are also shown. All values for the fitted MEAM potential were computed using the OpenKIM framework. DFT reference values are taken from Materials Project[7,81,82] (mp-135) except for vacancy energies which are taken from Ref. 83.

Property	MEAM	DFT	Rel. Error	E_s (Jm^{-2})	MEAM	DFT	Rel. error
a (Å)	3.44	3.44	0.000	(100)	0.466	0.462	0.009
c_{11} (GPa)	17	15	0.133	(110)	0.448	0.501	0.106
c_{12} (GPa)	13	13	0.000	(111)	0.516	0.544	0.051
c_{44} (GPa)	10	11	0.091	(210)	0.473	0.506	0.065
K (GPa)	14	14	0.000	(211)	0.505	0.538	0.061
E_v (eV)	0.455	0.481	0.054	(310)	0.473	0.497	0.048
E_m (eV)	0.055	0.042	0.309	(311)	0.494	0.527	0.063
E_a (eV)	0.510	0.523	0.025	(320)	0.603	0.504	0.196
				(321)	0.499	0.534	0.065
				(322)	0.510	0.535	0.047
				(331)	0.489	0.521	0.061
				(332)	0.592	0.524	0.130

the (320) surface]. This decreased performance of the model on surface energy predictions is not surprising, given the relatively small number of surface COs present in the training set (one CO per surface). The one exception is the vacancy migration energy, E_m, which has a higher relative error than the surface energies due its small magnitude. These results, along with results from automated verification checks on model integrity can be viewed on https://openkim.org/cite/MO_386038428339_000,[84] where the model has been archived along with >600 other curated and contributed models for a wide variety of chemical and material systems. This potential can be invoked in a portable fashion[85] within a variety of simulation platforms as explained above. We note that this example is only meant as a demonstration of how the interoperability ColabFit/KLIFF/OpenKIM leads to a streamlined fitting workflow. A potential major benefit of the ColabFit Exchange is the ability to leverage multiple datasets for DDIP development utilizing strategies such as transfer learning[86] and meta-learning.[87] However, these approaches are still very much an open scientific question, which we will seek to address in future work pertaining to the ColabFit project.

V. CONTRIBUTING

As with many open-source projects, the utility of the ColabFit Exchange will grow in proportion to the amount of engagement it receives from the research community. Contributions from the community may come in many forms. To name just a few possibilities, this could include: developing and uploading new DDIP training sets; training models to existing datasets and documenting performance metrics; improving the metadata in the database by adding labels to COs or defining new, meaningful CSs; or developing new tools (like those discussed in Sec. IV B) for characterizing dataset distributions.

Given that we foresee uploading training sets as being the most likely manner in which users will contribute to the ColabFit Exchange, we provide here some guidance on how users may best approach this task. The simplest way to contribute is through the Github repository at https://github.com/colabfit/data-lake, where instructions are provided for uploading data or requesting that the ColabFit team obtain existing data from the literature. Datasets contributed in this manner will be reviewed and parsed by the ColabFit team before submission to the database. In order to streamline the process of constructing useful and interpretable datasets, the following best practices should be followed by researchers interested in uploading their data to the ColabFit Exchange:

- DSs should be given meaningful, human-readable names. These need not be unique, since DSs are identified by their hashes, but it is useful if they are, in order to avoid confusion.
- Training/testing splits should be provided as separate DSs.
- DSs and CSs should be given concise descriptions outlining their contents. Discussions of the type of data contained within them (molecular, condensed matter, etc.) and their target applications (catalysis, radiation damage, drug discovery, benchmarking, etc.) are particularly useful.
- As much as possible, COs should be organized into conceptually meaningful CSs.
- As much as possible, COs should be given human-readable labels.
- All metadata required for reproducing a calculation (e.g., INCAR files) should be provided if possible.
- Computed properties should be adjusted to conform to existing PDs (a list of which can be found at colabfit.org). New PDs should be defined sparingly. Units must always be specified, when applicable.

Two of the most common, and challenging, issues that we struggled to overcome during the process of gathering datasets for the ColabFit Exchange were when dataset developers (1) used custom, poorly-documented storage formats for their data; or (2) did not define any conceptual groupings over their label which could be translated into CSs or CO labels. In general, we recommend the use of the Extended XYZ format as commonly used by ASE,[76] and the application of at least rudimentary labels on COs (e.g., "ground state," "liquid," "strained," etc.). For examples of well-constructed datasets, we point the reader to Refs. 88–90, whose authors we commend for publishing datasets with many desirable traits: (1) open-access, (2) well-documented storage formats, (3) good labeling of COs, and (4) clearly-defined groupings of COs.

While the Github repository is the simplest approach to contributing data, it relies upon a significant amount of effort from the ColabFit team in order to review and process the uploaded data, or to read through journal articles and contact authors to obtain access to their datasets. As an alternative, for those users who are able and willing, the colabfit-tools package provides all of the necessary code to manually parse your dataset into the data objects described in Sec. II (see https://github.com/colabfit/colabfit-tools for examples). This takes a large burden off of the ColabFit team, and can greatly accelerate the upload process.

VI. CONCLUSION

In this work we have developed a flexible and robust data standard that we applied to atomistic property data to construct the ColabFit Exchange, the first database of its kind specializing in data for data-driven interatomic potential generation typically employing machine learning techniques. At the time of writing (September 2023), the ColabFit Exchange contains 139 curated datasets and is actively being expanded, with particular emphasis on benchmarking datasets—those, which have been well tested, clearly documented, and shown to be suitable for analyzing aspects of model quality and guiding future development of reliable IPs. Along with the development of the ColabFit Exchange, we demonstrated the usefulness of DCA for identifying and characterizing overlapping regions of datasets, which can help to further guide dataset generation towards populating under-sampled regions of configurational and compositional space, thus improving the generalizability of the resultant DDIPs. Finally, we have shown how the data within the ColabFit Exchange can be utilized for end-to-end development of IPs within the KIM ecosystem, providing the benefits of seamless data retrieval, model exporting for use with major simulation software packages, and automated model verification, testing, and archiving on https://openkim.org. While our current focus is on atomistic data, specifically properties commonly applied to IP development, our framework is flexible enough to support a variety of different

data "silos," e.g., databases for meta-materials, bio-sequences, etc., which may become another application of the project in later work.

Future efforts of the ColabFit project will be to explore additional techniques for analyzing novel properties of datasets, like those described in Sec. IV B, which have been shown in some cases to correlate with generalizability and fitting errors of resultant models, and to develop metrics based on precision and recall scores for characterizing the utility of test sets. Further code development will also be done in order to expand the colabfit-tools package, with a focus on developing a Python API for accessing/contributing data, constructing datasets, and running consistency checks over contributed data (which is currently only done by hand). Perhaps most important for leveraging ColabFit's full potential will be gaining a better understanding of data interoperability and novel training strategies that can incorporate data across multiple datasets, levels of theory, and simulation parameters. As the ColabFit Exchange grows and matures, we anticipate it being an important tool for developing novel (meta-)learning strategies, which have recently been applied to atomistic datasets with promising results.[87]

We invite the community to upload data via the Github repository at https://github.com/colabfit/data-lake and will work closely with dataset developers who wish for their data (and models) to be findable, accessible, interoperable, and reusable.

SUPPLEMENTARY MATERIAL

The supplementary material includes additional details about Delaunay Component Analysis calculations, including sensitivity analysis for several hyperparameters.

ACKNOWLEDGMENTS

This research was supported through the National Science Foundation (NSF) under Grant No. OAC-2039575. S.M. acknowledges the Simons Center for Computational Physical Chemistry for financial support. This work was supported in part through the NYU IT High Performance Computing resources, services, and staff expertise. J.A.V. acknowledges the DIGI-MAT program at UIUC, which is supported by the National Science Foundation under Grant No. 1922758. The authors wish to acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the results reported in this paper.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

J.A.V. and E.G.F. contributed equally to this paper.

Joshua A. Vita: Conceptualization (equal); Data curation (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Eric G. Fuemmeler**: Data curation (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Amit Gupta**: Data curation (equal); Software (equal). **Gregory P. Wolfe**: Data curation (equal); Visualization (equal); Writing – original draft (supporting); Writing – review & editing (supporting). **Alexander Quanming Tao**: Data curation (supporting). **Ryan S. Elliott**: Conceptualization (equal); Methodology (equal). **Stefano Martiniani**: Conceptualization (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal). **Ellad B. Tadmor**: Conceptualization (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The entirety of the ColabFit Exchange can be found at https://colabfit.org. The colabfit-tools package can be found at https://github.com/colabfit/colabfit-tools.

REFERENCES

[1] A. Jain, G. Hautier, C. J. Moore, S. Ping Ong, C. C. Fischer, T. Mueller, K. A. Persson, and G. Ceder, "A high-throughput infrastructure for density functional theory calculations," Comput. Mater. Sci. **50**(8), 2295–2310 (2011).

[2] R. Armiento, B. Kozinsky, M. Fornari, and G. Ceder, "Screening for high-performance piezoelectrics using high-throughput density functional theory," Phys. Rev. B **84**(1), 014103 (2011).

[3] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials design and discovery with high-throughput density functional theory: The open quantum materials database (OQMD)," JOM **65**, 1501–1509 (2013).

[4] A. A. Emery and C. Wolverton, "High-throughput DFT calculations of formation energy, stability and oxygen vacancy formation energy of ABO_3 perovskites," Sci. Data **4**(1), 170153 (2017).

[5] A. Palizhati, W. Zhong, K. Tran, S. Back, and Z. W. Ulissi, "Toward predicting intermetallics surface properties with high-throughput DFT and convolutional neural networks," J. Chem. Inf. Model. **59**(11), 4742–4749 (2019).

[6] D. Wines, K. Choudhary, A. J. Biacchi, K. F. Garrity, and F. Tavazza, "High-throughput DFT-based discovery of next generation two-dimensional (2D) superconductors," Nano Lett. **23**(3), 969–978 (2023).

[7] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," APL Mater. **1**(1), 011002 (2013).

[8] L. Chanussot, A. Das, S. Goyal, T. Lavril, M. Shuaibi, M. Riviere, K. Tran, J. Heras-Domingo, C. Ho, W. Hu, A. Palizhati, A. Sriram, B. Wood, J. Yoon, D. Parikh, C. Lawrence Zitnick, and Z. Ulissi, "The open catalyst 2020 (OC20) dataset and community challenges," ACS Catalysis **11**(10), 6059–6072 (2021).

[9] R. Tran, J. Lan, M. Shuaibi, B. M. Wood, S. Goyal, A. Das, J. Heras-Domingo, A. Kolluru, A. Rizvi, N. Shoghi, A. Sriram, F. Therrien, J. Abed, O. Voznyy, E. H. Sargent, Z. Ulissi, and C. Lawrence Zitnick, "The open catalyst 2022 (OC22) dataset and challenges for oxide electrocatalysts," ACS Catalysis **13**(5), 3066–3084 (2023).

[10] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. H. Taylor, L. J. Nelson, G. L. Hart, S. Sanvito, M. Buongiorno-Nardelli, N. Mingo, and O. Levy, "AFLOWLIB.ORG: A distributed materials properties repository from high-throughput *ab initio* calculations," Comput. Mater. Sci. **58**, 227–235 (2012).

[11] C. Draxl and M. Scheffler, "NOMAD: The FAIR concept for big data-driven materials science," MRS Bull. **43**, 676–682 (2018).

[12]D. Jha, K. Choudhary, F. Tavazza, W. K. Liao, A. Choudhary, C. Campbell, and A. Agrawal, "Enhancing materials property prediction by leveraging computational and experimental data using deep transfer learning," Nat. Commun. **10**, 5316 (2019).
[13]K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, A. R. Hight Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, A. Davydov, J. Jiang, R. Pachter, G. Cheon, E. Reed, A. Agrawal, X. Qian, V. Sharma, H. Zhuang, S. V. Kalinin, B. G. Sumpter, G. Pilania, P. Acar, S. Mandal, K. Haule, D. Vanderbilt, K. Rabe, and F. Tavazza, "The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design," npj Comput. Mater. **6**(1), 173 (2020).
[14]F.-S. Meng, J.-P. Du, S. Shinzato, H. Mori, P. Yu, K. Matsubara, N. Ishikawa, and S. Ogata, "General-purpose neural network interatomic potential for the α-iron and hydrogen binary system: Toward atomic-scale understanding of hydrogen embrittlement," Phys. Rev. Mater. **5**(11), 113606 (2021).
[15]A. Rohskopf, C. Sievers, N. Lubbers, M. A. Cusentino, J. Goff, J. Janssen, M. McCarthy, D. M. de Oca Zapiain, S. Nikolov, K. Sargsyan, D. Sema, E. Sikorski, L. Williams, A. P. Thompson, and M. A. Wood, "FitSNAP: Atomistic machine learning with LAMMPS," J. Open Source Software **8**(84), 5118 (2023).
[16]R. Atwi, M. Bliss, M. Makeev, and N. N. Rajput, "MISPR: An open-source package for high-throughput multiscale molecular simulations," Sci. Rep. **12**(1), 15760 (2022).
[17]J. L. A. Gardner, Z. Faure Beaulieu, and V. L. Deringer, "Synthetic data enable experiments in atomistic machine learning," Dig. Discov. **2**(3), 651–662 (2023).
[18]A. S. Christensen and O. Anatole von Lilienfeld, "On the role of gradients for machine learning of molecular energies and forces," Mach. Learn. Sci. Technol. **1**(4), 045018 (2020).
[19]M. Schreiner, A. Bhowmik, T. Vegge, J. Busk, and O. Winther, "Transition1x - A dataset for building generalizable reactive machine learning potentials," Sci. Data **9**(1), 779 (2022).
[20]S. Takamoto, C. Shinagawa, D. Motoki, K. Nakago, W. Li, I. Kurata, T. Watanabe, Y. Yayama, H. Iriguchi, Y. Asano, T. Onodera, T. Ishii, T. Kudo, H. Ono, R. Sawada, R. Ishitani, M. Ong, T. Yamaguchi, T. Kataoka, A. Hayashi, N. Charoenphakdee, and T. Ibuka, "Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements," Nat. Commun. **13**(1), 2991 (2022).
[21]X. Guan, A. Das, C. J. Stein, F. Heidar-Zadeh, L. Bertels, M. Liu, M. Haghighatlari, J. Li, O. Zhang, H. Hao, I. Leven, M. Head-Gordon, and T. Head-Gordon, "A benchmark dataset for hydrogen combustion," Sci. Data **9**(1), 215 (2022).
[22]R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," Sci. Data **1**(1), 140022 (2014).
[23]Y. Lysogorskiy, C. v.d. Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hammerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, "Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon," npj Computat. Mater. **7**(1), 97 (2021).
[24]P. Ying, H. Dong, T. Liang, Z. Fan, Z. Zhong, and J. Zhang, "Atomistic insights into the mechanical anisotropy and fragility of monolayer fullerene networks using quantum mechanical calculations and machine-learning molecular dynamics simulations," Extreme Mech. Lett. **58**, 101929 (2023).
[25]A. M. Maldonado, I. Poltavsky, V. Vassilev-Galindo, A. Tkatchenko, and J. A. Keith, "Modeling molecular ensembles with gradient-domain machine learning force fields," Dig. Discov. **2**(3), 871–880 (2023).
[26]P. Wisesa, C. M. Andolina, and W. A. Saidi, "Development and validation of versatile deep atomistic potentials for metal oxides," J. Phys. Chem. Lett. **14**(2), 468–475 (2023).
[27]See https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/ostp_public_access_memo_2013.pdf for Office of Science and Technology Policy, Executive Office of the President. Increasing access to the results of federally funded scientific research (February 22, 2013).
[28]See https://www.whitehouse.gov/wp-content/uploads/2022/08/08-2022-OSTP-Public-Access-Memo.pdf for Office of Science and Technology Policy, Executive Office of the President. Ensuring free, immediate, and equitable access to federally funded research (August 25 2022).
[29]M. Scheffler, M. Aeschlimann, M. Albrecht, T. Bereau, C. Felser, M. Greiner, A. Groß, C. Koch, K. Kremer, E. Wolfgang, M. Scheidgen, C. Wöll, and C. Draxl, "FAIR data new horizons for materials research," Nature **604**, 635 (2022).
[30]See https://github.com/colabfit/colabfit-tools for ColabFit. colabfit-tools (2023).
[31]E. B. Tadmor, R. S. Elliott, and D. S. Karls, KIM Property Definition Framework, https://openkim.org/doc/schema/properties-framework/.
[32]G. Kresse and J. Hafner, "*Ab initio* molecular dynamics for liquid metals," Phys. Rev. B **47**(1), 558–561 (1993).
[33]M. L. Hutchinson, E. Antono, B. M. Gibbons, S. Paradiso, J. Ling, and B. Meredig, "Overcoming data scarcity with transfer learning," arXiv.1711.05099.
[34]H. Jónsson, G. Mills, and K. W. Jacobsen, "Nudged elastic band method for finding minimum energy paths of transitions," in *Classical and Quantum Dynamics in Condensed Phase Simulations* (World Scientific, 1998).
[35]S. Chmiela, T. Alexandre, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K. Robert Müller, "Machine learning of accurate energy-conserving molecular force fields," Sci. Adv. **3**, 5 (2017).
[36]M. Wen and E. B. Tadmor, "Hybrid neural network potential for multilayer graphene," Phys. Rev. B **100**(9), 195419 (2019).
[37]J. S. Smith, B. Nebgen, N. Mathew, J. Chen, N. Lubbers, L. Burakovsky, S. Tretiak, H. Ah Nam, T. Germann, S. Fensin, and K. Barros, "Automated discovery of a robust interatomic potential for aluminum," Nat. Commun. **12**, 1257 (2021).
[38]L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," IEEE Signal Process. Mag. **29**(6), 141–142 (2012).
[39]A. Krizhevsky, "Learning multiple layers of features from tiny images," Technical report (2009).
[40]C. W. Andersen, R. Armiento, E. Blokhin, G. J. Conduit, S. Dwaraknath, M. L. Evans, Á. Fekete, A. Gopakumar, S. Gražulis, A. Merkys, F. Mohamed, C. Oses, G. Pizzi, G.-M. Rignanese, M. Scheidgen, L. Talirz, C. Toher, D. Winston, R. Aversa, K. Choudhary, P. Colinet, S. Curtarolo, D. Di Stefano, C. Draxl, S. Er, M. Esters, M. Fornari, M. Giantomassi, M. Govoni, G. Hautier, V. Hegde, M. K. Horton, P. Huck, G. Huhs, J. Hummelshøj, A. Kariryaa, B. Kozinsky, S. Kumbhar, M. Liu, N. Marzari, A. J. Morris, A. A. Mostofi, K. A. Persson, G. Petretto, T. Purcell, F. Ricci, F. Rose, M. Scheffler, D. Speckhard, M. Uhrin, A. Vaitkus, P. Villars, D. Waroquiers, C. Wolverton, M. Wu, and X. Yang, "OPTIMADE, an API for exchanging materials data," Sci. Data **8**(1), 217 (2021).
[41]J. S. Smith, N. Ben, N. Lubbers, O. Isayev, and E. AdrianRoitberg, "Less is more: Sampling chemical space with active learning," J. Chem. Phys. **148**(24), 241733 (2018).
[42]C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," Nat. Comput. Sci. **2**(11), 718–728 (2022).
[43]B. Deng, P. Zhong, K. J. Jun, K. Han, C. J. Bartel, and G. Ceder, "CHGNet: Pretrained universal neural network potential for charge-informed atomistic modeling," Nat. Mach. Intell. **5**(9), 1031–1041 (2023)
[44]L. Komissarov and T. Verstraelen, "Zeo-1, a computational data set of zeolite structures," Sci. Data **9**(1), 61 (2022).
[45]L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17," J. Chem. Inf. Model. **52**(11), 2864–2875 (2012).
[46]F. A. Faber, L. Alexander, O. Anatole von Lilienfeld, and R. Armiento, "Machine learning energies of 2 million elpasolite (ABC_2D_6) crystals," Phys. Rev. Lett. **117**(13), September (2016).
[47]J. Behler, "Atom-centered symmetry functions for constructing high-dimensional neural network potentials," J. Chem. Phys. **134**(7), 074106 (2011).
[48]A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," Phys. Rev. B **87**, 184115 (2013).
[49]M. Rupp, A. Tkatchenko, K. R. Müller, and O. A. von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," Phys. Rev. Lett. **108**, 058301 (2012).
[50]R. Drautz, "Atomic cluster expansion for accurate and transferable interatomic potentials," Phys. Rev. B **99**, 014104 (2019).
[51]H. Huo and M. Rupp, "Unified representation of molecules and crystals for machine learning," Mach. Learn.: Sci. Tech. **3**(4), 045017 (2022).
[52]L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," J. Mach. Learn. Res. **9**(86), 2579–2605 (2008).

[53]P. Poklukar, V. Polianskii, A. Varava, F. Pokorny, and D. Kragic, "Delaunay component analysis for evaluation of data representations," arXiv.2202.06866.
[54]M. Wen and E. B. Tadmor, "Uncertainty quantification in molecular simulations with dropout neural network potentials," npj Computat. Mater. **6**(1), 124 (2020).
[55]P. Rowe, V. L. Deringer, P. Gasparotto, G. Csányi, and A. Michaelides, "An accurate and transferable machine learning potential for carbon," J. Chem. Phys. **153**(3), 034702 (2020).
[56]D. Hedman, T. Rothe, G. Johansson, F. Sandin, J. A. Larsson, and Y. Miyamoto, "Impact of training and validation data on the performance of neural network potentials: A case study on carbon using the CA-9 dataset," Carbon Trends **3**, 100027 (2021).
[57]K. Li, D. Persaud, K. Choudhary, B. DeCost, M. Greenwood, and J. Hattrick-Simpers, "On the redundancy in large material datasets: Efficient and robust learning with less data," arXiv.2304.13076.
[58]E. Heid, C. J. McGill, F. H. Vermeire, and W. H. Green, "Characterizing uncertainty in machine learning for chemistry," J. Chem. Inf. Model. **63**(13), 4012–4029 (2023).
[59]J. A. Vita and D. Schwalbe-Koda, "Data efficiency and extrapolation trends in neural network interatomic potentials," Mach. Learn.: Sci. Technol. **4**(3), 035031 (2023).
[60]M. Aldeghi, D. E. Graff, N. Frey, J. A. Morrone, E. O. Pyzer-Knapp, K. E. Jordan, and C. W. Coley, "Roughness of molecular property landscapes and its impact on modellability," J. Chem. Inf. Model. **62**(19), 4660–4671 (2022).
[61]A. Glielmo, C. Zeni, B. Cheng, G. Csányi, and A. Laio, "Ranking the information content of distance measures," PNAS Nexus **1**(2), pgac039 (2022).
[62]M. Karabin and D. Perez, "An entropy-maximization approach to automated training set generation for interatomic potentials," J. Chem. Phys. **153**(9), 094110 (2020).
[63]A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," Phys. Rev. Lett. **104**(13), 136403 (2010).
[64]A. Singraber, mpbircher, S. Reeve, D. W. H. Swenson, J. Lauret, and philippedavid, Compphysvienna/n2p2: Version 2.1.4, 2021.
[65]S. L. Batzner, M. Albert, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and E. Boris Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," Nat. Commun. **13**, 2453 (2021).
[66]M. Wen, Y. Afshar, R. S. Elliott, and E. B. Tadmor, "Kliff: A framework to develop physics-based and machine learning interatomic potentials," Comput. Phys. Commun. **272**, 108218 (2022).
[67]E. B. Tadmor, R. S. Elliott, J. P. Sethna, R. E. Miller, and C. A. Becker, "The potential of atomistic simulations and the knowledgebase of interatomic models," JOM **63**(7), 17 (2011).
[68]R. S. Elliott and E. B. Tadmor, "Knowledgebase of Interatomic Models (KIM) application programming interface (API)," https://openkim.org/kim-api (2011).
[69]E. B. Tadmor, R. S. Elliott, S. R. Phillpot, and S. B. Sinnott, "NSF cyberinfrastructures: A new paradigm for advancing materials simulation," Curr. Opin. Solid State Mater. Sci. **17**(6), 298–304 (2013).
[70]D. S. Karls, M. Bierbaum, A. A. Alemi, R. S. Elliott, J. P. Sethna, and E. B. Tadmor, "The OpenKIM processing pipeline: A cloud-based automatic material property computation engine," J. Chem. Phys. **153**, 064104 (2020).
[71]M. I. Baskes, "Modified embedded-atom potentials for cubic materials and impurities," Phys. Rev. B **46**, 2727–2742 (1992).
[72]T. J. Lenosky, B. Sadigh, E. Alonso, V. V. Bulatov, T. D. d.l. Rubia, J. Kim, A. F. Voter, and J. D. Kress, "Highly optimized empirical potential model of silicon," Modell. Simul. Mater. Sci. Eng. **8**(6), 825 (2000).
[73]Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood, and S. P. Ong, "Performance and cost assessment of machine learning interatomic potentials," J. Phys. Chem. A **124**(4), 731–745 (2020).
[74]J. A. Vita and D. R. Trinkle, "Exploring the necessary complexity of interatomic potentials," Comput. Mater. Sci. **200**, 110752 (2021).
[75]P. Adam, S. Gross, F. Massa, L. Adam, B. James, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, R. Martin, A. Tejani, S. Chilamkurthy, B. Steiner, F. Lu, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2019), Vol. 32, pp. 8024–8035.
[76]A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment—A python library for working with atoms," J. Phys.: Condens. Matter **29**(27), 273002 (2017).
[77]I. T. Todorov, W. Smith, K. Trachenko, and M. T. Dove, "DL_POLY_3: New dimensions in molecular dynamics simulations via massive parallelism," J. Mater. Chem. **16**, 1911–1918 (2006).
[78]J. D. Gale, "Gulp: A computer program for the symmetry-adapted simulation of solids," J. Chem. Soc., Faraday Trans. **93**, 629–637 (1997).
[79]A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, "LAMMPS - A flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales," Comput. Phys. Commun. **271**, 108171 (2022).
[80]See https://openkim.org/projects-using-kim/ for Software and projects using KIM.
[81]M. de Jong, W. Chen, T. Angsten, A. Jain, R. Notestine, A. Gamst, M. Sluiter, C. Krishna Ande, S. van der Zwaag, J. J. Plata, C. Toher, S. Curtarolo, G. Ceder, K. A. Persson, and M. Asta, "Charting the complete elastic properties of inorganic crystalline compounds," Sci. Data **2**, 150009 (2015).
[82]R. Tran, Z. Xu, B. Radhakrishnan, D. Winston, W. Sun, K. A. Persson, and S. P. Ong, "Surface energies of elemental crystals," Sci. Data **3**, 160080 (2016).
[83]W.-S. Ko and J. B. Jeon, "Interatomic potential that describes martensitic phase transformations in pure lithium," Comput. Mater. Sci. **129**, 202–210 (2017).
[84]E. Fuemmeler and J. Vita, MEAM spline potential for Li developed by Fuemmeler and Vita (2023) v000, OpenKIM, 2023.
[85]Y. Afshar, S. Hütter, R. E. Rudd, S. Alexander, W. W. Tipton, D. R. Trinkle, G. J. Wagner, P. Zhang, E. Alonso, M. I. Baskes, V. V. Bulatov, T. D. de la Rubia, J. Kim, J. D. Kress, B.-J. Lee, T. Lenosky, J. S. Nelson, B. Sadigh, A. F. Voter, and A. F. Wright, The modified embedded atom method (MEAM) potential v002, OpenKIM, 2023.
[86]V. Zaverkin, D. Holzmüller, L. Bonfirraro, and J. Kästner, "Transfer learning for chemically accurate interatomic neural network potentials," Phys. Chem. Chem. Phys. **25**, 5383–5396 (2023).
[87]A. E. A. Allen, N. Lubbers, S. Matin, J. Smith, R. Messerly, S. Tretiak, and K. Barros, "Learning together: Towards foundational models for machine learning interatomic potentials with meta-learning," arXiv.2307.04012.
[88]J. Byggmästar, A. Hamedani, K. Nordlund, and F. Djurabekova, "Machine-learning interatomic potential for radiation damage and defects in tungsten," Phys. Rev. B **100**(14), 144105 (2019).
[89]A. P. Bartók, K. James, N. Bernstein, and G. Csányi, "Machine learning a general-purpose interatomic potential for silicon," Phys. Rev. X **8**(4), 041048 (2018).
[90]M. A. Wood, M. A. Cusentino, B. D. Wirth, and A. P. Thompson, "Data-driven material models for atomistic simulation," Phys. Rev. B **99**(18), 184305 (2019).