Computational Materials Science 186 (2021) 110064

Contents lists available at ScienceDirect

Computational Materials Science

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

ALKEMIE: An intelligent computational platform for accelerating materials discovery and design

Guanjie Wang, Liyu Peng, Kaiqi Li, Linggang Zhu *, Jian Zhou, Naihua Miao, Zhimei Sun *

School of Materials Science and Engineering, Beihang University, Beijing 100191, China
Center for Integrated Computational Materials Engineering, International Research Institute for Multidisciplinary Science, Beihang University, Beijing 100191, China

ARTICLE INFO

Keywords:
High-throughput calculations
Machine learning
Graphical user interface
Materials informatics

ABSTRACT

Developing new materials with target properties via the traditional trial-and-error ways is cost-inefficient, and sometimes ends up with fruitlessness, therefore, simulation-driven materials design plays an important role in the past decades. Nevertheless, the advent of the era of data-driven material science requires an intelligent computational platform to accelerate the discovery and design for advanced materials. Here, we present an *open-source* computational platform named as ALKEMIE, acronyms for Artificial Learning and Knowledge Enhanced Materials Informatics Engineering, which enables easy access of data-driven techniques to broad communities. ALKEMIE is incorporated with three key components for the computational design of materials for the forthcoming data-driven sciences: data generation via high-throughput calculations, data management and data mining via machine learning models. Briefly speaking, the high-throughput calculations in ALKEMIE are implemented through the integration of automatic frameworks of model constructions, calculation performances and data analysis. And the used high-level application programming interface for the database makes the data mining through machine learning more applicable in material science. In particular, ALKEMIE is integrated with a module for the generation of machine-learned interatomic potential for large-scale molecular dynamic simulations where the dataset is obtained from high-throughput first-principles calculations. More importantly, ALKEMIE has an elaborately designed user-friendly graphical user-interface which makes the workflow and dataflow more maneuverable and transparent, facilitating its easy-to-use for scientists with broad backgrounds. Finally, the main characters of ALKEMIE are demonstrated using three computational examples.

1. Introduction

For centuries, the development of material science (old saying as metallurgy) mainly relies on the experience based on the experiment, then come the general theories such as the law of thermodynamics. The advancement of computer hardwares and softwares implementing the density functional theory and molecular dynamics, etc, leads to the computer simulation era of material science. The emergence of the big data technology facilitates the shift of the material science to the fourth paradigm, i.e., the data-driven stage [1]. The coming of the data-driven material science presents a very promising and inspiring pathway for the faster and more cost-efficient design and deployment of advanced materials compared to the traditional trial-and-error method. For the data-driven paradigm of material science, construction of relevant infrastructures is clearly needed, which has been greatly promoted by the Materials Genome Initiative (MGI) launched in 2011 [2]. In recent years, following the route presented in MGI, the development of computational tools and their applications in materials design are springing up [3–14]. A general material design platform combining the data generation, data management and data mining is very desirable for the material designers with wide backgrounds.

Extensive efforts have been put on the frameworks or codes development considering the high-throughput calculation (HTC) [15–20], material data management [21–23] and machine learning in materials science [24–30]. Materials Project [31] is a well-known pioneer for HTC in material science, and its frameworks including Pymatgen [32], FireWorks [33], Custodian [34], and Atomate [35] which enable the automatic computation are now widely used in the community. AFLOWπ [36] is another minimalist framework for high-throughput first-principles calculations, which supports the data generation, error control, curation and archiving of the data, and post-processing tools for analysis and visualization. AiiDA (automated interactive infrastructure and

* Corresponding authors at: School of Materials Science and Engineering, Beihang University, Beijing 100191, China.
E-mail addresses: lgzhu7@buaa.edu.cn (L. Zhu), zmsun@buaa.edu.cn (Z. Sun).

https://doi.org/10.1016/j.commatsci.2020.110064
Received 29 July 2020; Received in revised form 9 September 2020; Accepted 10 September 2020

database) [37] is also an *open-source* python infrastructure to help researchers with automating, managing, persisting, sharing and reproducing the complex workflows associated with modern computational science and data. In addition, there are also other codes or platforms, such as Atom simulation environment (ASE) [38], Pylada [39], Imeall [40] and MPInterfaces [41], have been developed for HTC. Meanwhile, the online material databases expand very fast in recent years, benefiting from the application of HTC. Examples of such databases are the Materials Project [31], OQMD (Open Quantum Materials Database) [42], AFLOWLIB consortium (Automatic Flow Lib) [43,44], NOMAD [45], ICSD (Inorganic experimental) [46], COD [47], CMR (ASE-database) [48] and Materials Cloud [37], etc. The rapidly increasing and easily available data in these depositories make the application of the data mining via machine leaning (ML) possible. In addition to the general ML codes such as scikit-learn [49], TensorFlow [50] and PyTorch [51], a few other codes have been developed specifically for material science with the general ML codes included as the engines. For examples, AFLOW-ML [52] is a representational-state-transfer-architecture API for machine-learning predictions of materials properties; SISSO [53], written in FORTRAN 90, is a compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates; MatMiner [54] is another platform to facilitate data-driven methods for analyzing and predicting materials properties based on scikit-learn. In addition, there are many other machine learning tools used for specific fields in material science, for example, PROphet [55], COMBO [56], Magpie [57] and JARVIS-ML [58], etc.

As far as we know, most of the MGI related codes are designed for one or two of the three components which are the generation, management and mining of the data in the data-driven materials design. Here we introduce a general computational platform ALKEMIE (**A**rtificial **L**earning and **K**nowledge **E**nhanced **M**aterials **I**nformatics **E**ngineering), which works like a normal application software while covers all of the three components of the data flow. ALKEMIE has incorporated many well-known *open-source* processing tools, computation engines at different length/time scales, etc, and more importantly, it is equipped with a very friendly graphical user interface which makes the workflow and dataflow more manageable and transparent. Moreover, ALKEMIE is greatly enhanced by a well-designed module for machine-learned interatomic potential generation, which bridges the first-principles calculations and classic molecular dynamics simulations.

This paper is organized as follows. Firstly, the main features of ALKEMIE are presented in Section 2. Next, in Section 3, we describe the design philosophy and architecture of ALKEMIE. In Section 4, the core modules in ALKEMIE are introduced in detail. Finally, in Section 5, three examples are used to demonstrate the main features of ALKEMIE: high-throughput first-principles calculation with 10^3 tasks, screening of the easily alloying elements in copper in terms of the formation energy of the alloyed system, and the generation of the machine-learned interatomic potential for Sb.

2. Main features of ALKEMIE

The main features of ALKEMIE are summarized in Fig. 1, more detailed explanations are as follows:

- **High-throughput:** ALKEMIE can manage $\geqslant 10^3$ tasks in one workflow.
- **Automation:** The entire process of HTC, from model construction, calculation to data analysis, can be run automatically by reasonably-defined default parameters without human interventions.
- **Visualization:** A user-friendly graphical user interface (GUI), based on Orange [59] and PyQt [60], is designed for ALKEMIE, allowing users to deal with tasks and data as 'seeing' them. This GUI should be very beneficial for the nonexperts or beginners.
- **Workflows:** Based on the Materials Project [31], we set up many general scientific workflows with default parameters that have sufficient accuracy. Users can define their own workflows, or use the default workflows with self-defined parameters.
- **Database:** All data in the workflows are stored in different types of databases, such as (atomic) geometry database, task database, property database and fingerprint database for machine learning, etc.
- **Machine Learning:** ALKEMIE is integrated many general machine learning tools, such as scikit-learn, PyTorch and TensorFlow, to facilitate data mining for specific material issues.
- **Plug-in mode:** ALKEMIE has various interfaces with simulation softwares at different time/length scales. And more codes will be included in future version of ALKEMIE.

3. Design philosophy and architecture of ALKEMIE

As mentioned in the Introduction section, many codes/frameworks have been built following the culture of MGI. When we start to design a general platform, python is chosen as the main programming language to make full use of its ecosystem and also because the acknowledged *open-source* MGI-related codes are mainly written in python. Our ambition in designing ALKEMIE is to expand the functions of some available codes and to build its own advantages of modules at the same time.

ALKEMIE has a client–server model: the client can be installed in a personal computer with any mainstream operating system such as Windows, Linux or MacOS; the server can be deployed in a remote

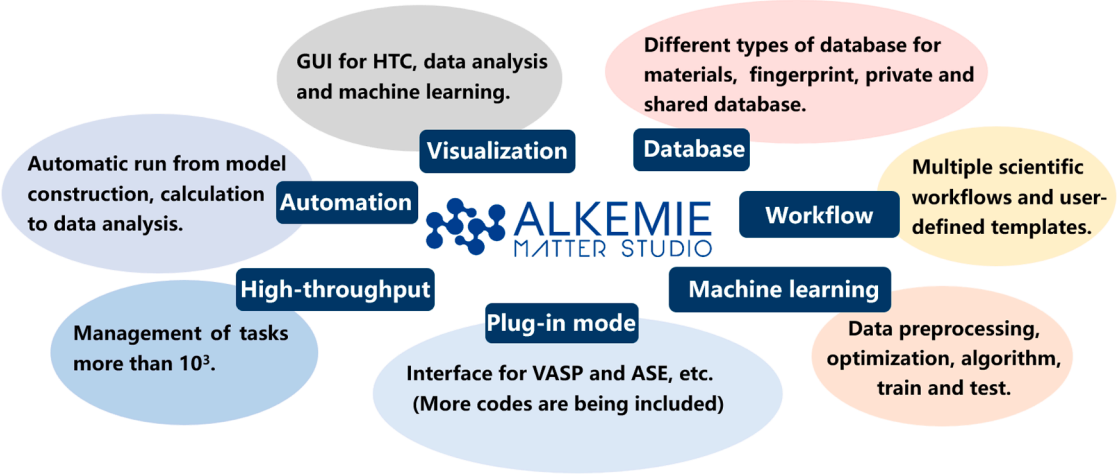

Fig. 1. The main features of ALKEMIE.

cluster. Users can switch between different clusters conveniently through the client using a switch button. At the (remote) clusters, ALKEMIE will open a daemon process and listening port, which is also the basic requirement for clients to achieve automatic access to cluster. In addition, the server of ALKEMIE can handle different resource management systems including PBS, SLURM, and LTRM, and has advanced APIs to customize other management systems.

For the data generation process, before the start of the calculation, the first thing to deal with is constructing the geometries of the material, thus in ALKEMIE we define the *Builder* module. Users can easily introduce the dopant/grain-boundary to the systems automatically, or import a batch of existed structures, both of which can be performed in a high-throughput way. The built geometry can be viewed via a *Viewer* plug-in widget, then the user can double check the structure or export it for presentation. Afterwards, users can define a specific task to compute the needed property which is called workflow. In ALKEMIE, a few commonly used workflows are pre-defined. The input files for a computation software and a specific workflow can be generated automatically with the parameter set by default or by hand. As all inputs for a computation software are ready, the jobs can be submitted to the server. We incorporate the *open-source* codes for job control/monitor, error correction as in the Materials Project, while we designed a special GUI for ALKEMIE. When the high-throughput calculations are finished, simple data analysis can be done, such as analyzing the energy, volume, and band gap distribution in the computed materials. Also, *Plotter* in ALKEMIE can handle the plot of density of states, E–V curve or band structures. For now, all these functions are fully available for the first-principles code VASP [61].

For the data management, firstly, inputs/outputs of the computation software, calculation parameters, scientific workflows, and the calculated properties, should be saved properly. Then the standardization and pre-processing of the data are needed before conducting data mining. In ALKEMIE, we choose MongoDB [62] as our core database. This database uses a simple and understandable *JSON* format file to save data, which is convenient for users to store and query data. Moreover, for researchers, data can be either shared or privatized, so we provide each user with two types of databases, i.e., a separated private database and a shared database. By default, the user creates a private database while registering a new account. And users can apply to get access to the shared database, where they may provide or get shared data.

For the data mining part, ALKEMIE integrates the current popular machine learning package Tensorflow, Pytorch and scikit-learn, and provides high-level API for users to choose the appropriate machine learning model (SVM, DNN, CNN, RNN, etc.) and machine learning convergence algorithm (Adam [63], Gradient descent [64] and so on). In the community of computational material science, molecular dynamic (MD) is a very powerful atomic-scale simulation method, covering a much longer time-scale and larger length-scale than the first-principles calculations. Nevertheless, MD simulations always lack accurate interatomic potentials to describe the interactions among atoms in materials, which is a well-known tough problem in MD community. Therefore, it is most desirable to develop a technique that can efficiently establish interatomic potentials for large-scale MD simulation. Herein, we designed a module for the generation of interatomic potentials via machine leaning methods using datasets from the accurate first-principles calculations, in which way bridges first-principles calculations and classic molecular dynamic simulations. This module has been implemented in the calculator of ALKEMIE.

More importantly, given the interdisciplinary character of the data-driven techniques as indicated above, it is very desirable to provide a user-friendly GUI for the three processes in the data flow.

The architecture of ALKEMIE is shown in Fig. 2. The overall architecture is divided into four layers, which are user layer, graphical user interface (GUI) layer, kernel layer and plug-in layer. Firstly, thanks to python's compatibility, ALKEMIE can be installed on the three mainstream operating system platforms (Linux, Windows and MacOS) through PyPi's open source repositories [65]. The second layer is the GUI layer, researchers can run all the kernel programs on the GUI layer

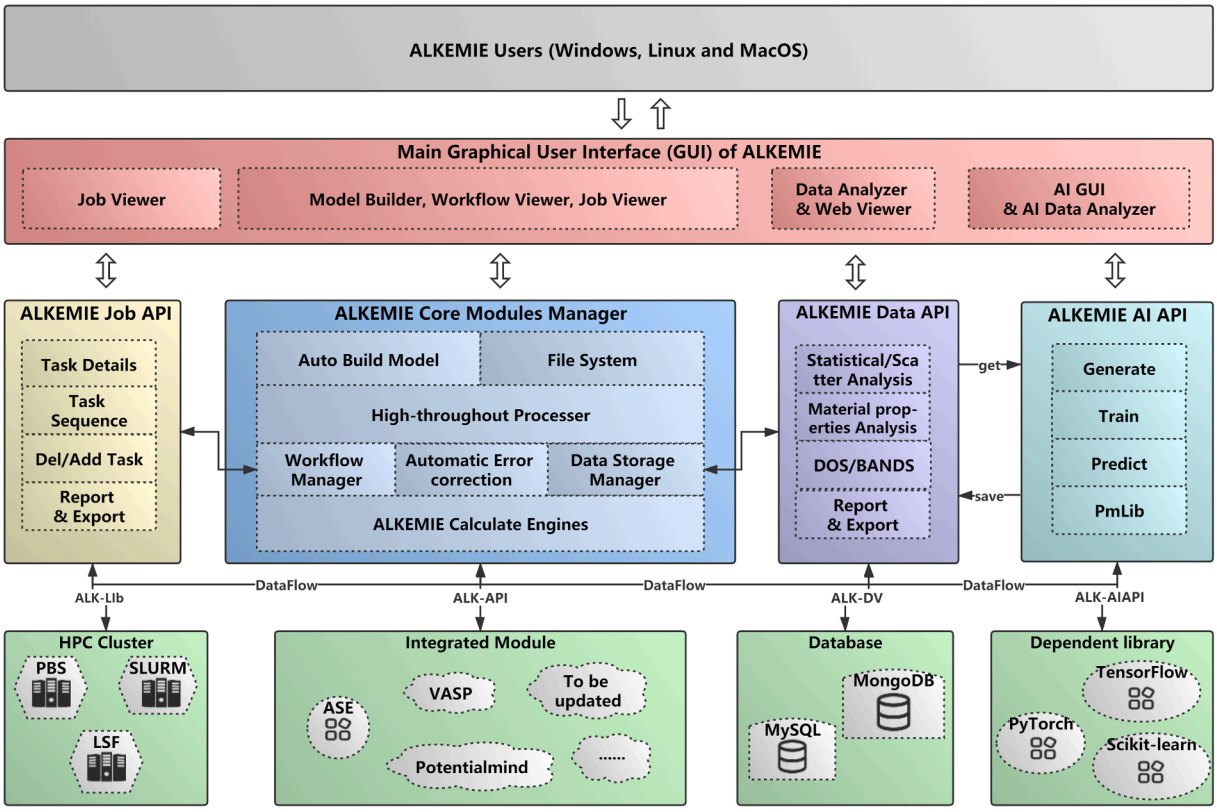

Fig. 2. The architecture diagram of ALKEMIE.

through the advanced application programming interface (API). The GUI is completely independent from the kernel, and they are connected through the specified information pipelines. This independency facilitates future developments of more visualization functions. The next layer is the kernel layer, which is divided into four parts: job management, computing engine, I/O (input and output) processing, and machine learning. Further, there are two types of API in each part: one API can be called by the GUI layer, and the other Lib-API can be used to integrate other functions. The bottom layer is a modular plug-in layer that allows users to add their own computation software, which can be called by computing engine through Lib-API. The plug-in layer can include commercial software that requires the license to use and *open-source* software that conforms to the corresponding *open-source* rules. Those functions of ALKEMIE have been tested by the python standard test library *unittest* [66].

Since ALKEMIE is designed as a platform for general use, the users can add their own code to the platform conveniently following the steps a)–c).

- **a)** Firstly, to add a new code to ALKEMIE, the users should create a new folder in the directory '*$HOME/Alkemiems/widgets*', and in this new folder, a file named '*_init_.py*' should be included which contains the code's name, icon and other basic information.
- **b)** Secondly, the users can design workflows for this code to achieve desirable functions of pre-processing, calculations or post-processing. The user-defined class must inherit from *BaseTasks*, which is a high-level API built into ALKEMIE to solve the relationship between multi subtasks. Further, it is essential to follow ALKEMIE's database architecture and organize all input/output of the computation in a *JSON* format stored in MongoDB.
- **c)** Finally, one may build the workflow widget based on our predefined *OWBaseWf* API, making the workflow visible in the main GUI of ALKEMIE. Some attributes of the class must be specified, such as the name, description, icon, input, output, workflow function defined in step (b), etc.

4. Details of the core components of ALKEMIE

In this section, we will introduce three main components of ALKEMIE, which are visualized high-throughput workflow, database, and machine learning.

4.1. Visualized high-throughput workflow in ALKEMIE

The main GUI of the ALKEMIE is shown in Fig. 3, which contains login, welcome and main working interface. Firstly, users of the ALKEMIE client will get a license for the ALKEMIE server side, to start the software and to get access to the remote cluster, as shown in Fig. 3 (a). The welcome page (Fig. 3 (b)) contains common contents such as creating a new project, loading the last saved work, and the tutorial. After the welcome page, it is the main working interface, which is divided into two parts: the left part is the software tool bar; the right part is the working panel. One example of a high-throughput workflow is also shown in Fig. 3 (c).

The working panel of the main GUI of ALKEMIE is very user-friendly and interactive, where a high-throughput work can be conveniently designed and setup. For a clearer presentation of the automatic workflows, different stages of the flow are shown with widget in different colors/shapes (as shown in Fig. 3 (c)), where the input layer is shown using purple circle, workflow layer is represented by blue hexagon, remote-job management layer is in green, while data analysis layer is in orange, etc. In the working panel, the widget can be 'clicked' to configure the corresponding functions, and users can simply 'draw' a 'line' to make different functions connected, activating the 'flow'. The line-connection between different layers will automatically start the core of one layer to run its corresponding functions and transfer the output to the next layer. Thus, using the intelligent working panel of ALKEMIE, the high-throughput work can be easily setup and managed.

Fig. 4 illustrates some details of the widgets. The widget *Builder* in Fig. 4 (a) can be used to build the grain boundary, introduce the vacancy/dopant into arbitrary site of the parent lattice, etc. In Fig. 4 (a), one carbon atom is doped at fraction coordinate (0.5, 0.5, 0.5) in the lattice of $Ge_2Sb_2Te_5$. Users can export the doping structure, or they can press *send-data* button to add this model to HTC workflow. Fig. 4 (b) shows the detail of the scientific workflow for band structure calculation, including workflow parameters configuration panel (left) and flowchart of this workflow (right). The scientific workflow can solve the nesting relationship of multi-tasks. Users can freely design their own workflows, or build the workflows based on the built-in ones, such as the *VaspBase* for VASP users. Each workflow contains three types of parameters: the first one is the software-related environmental variables, the second type is the internal relational parameters of the workflow, and the third one is the parameter needed by the software to run simulations. For some workflows using VASP code, such as the static calculation, structural optimization, as well as the calculations of

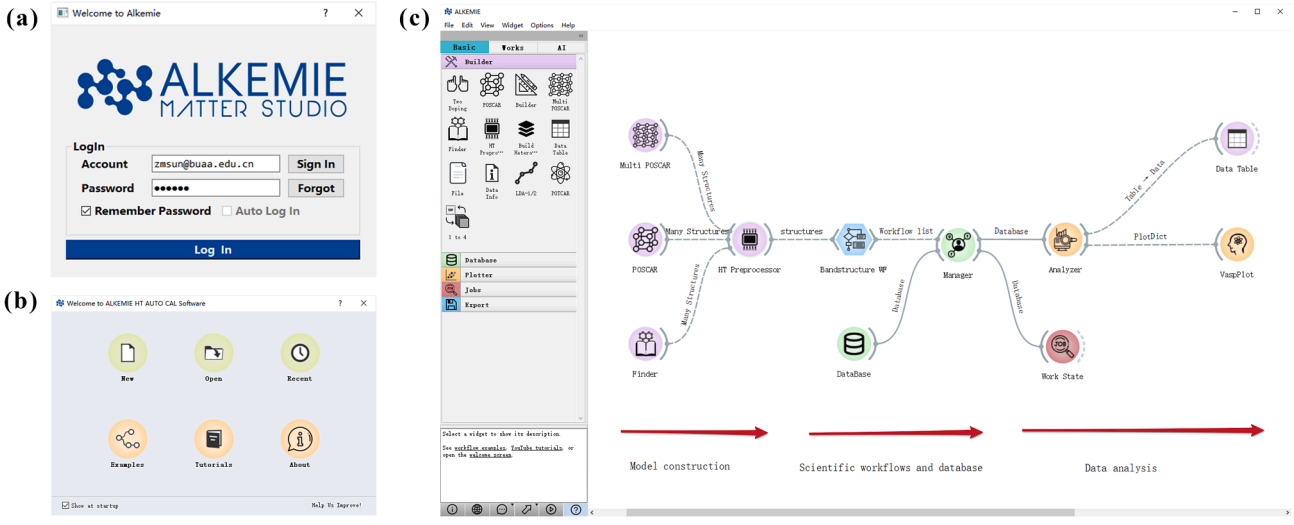

Fig. 3. The main GUI of ALKEMIE and an example of the visualized high-throughput automatic calculation. For clarity, figure (a)–(c) is shown separately as Fig. S1–S3 in the Supplemental Materials.

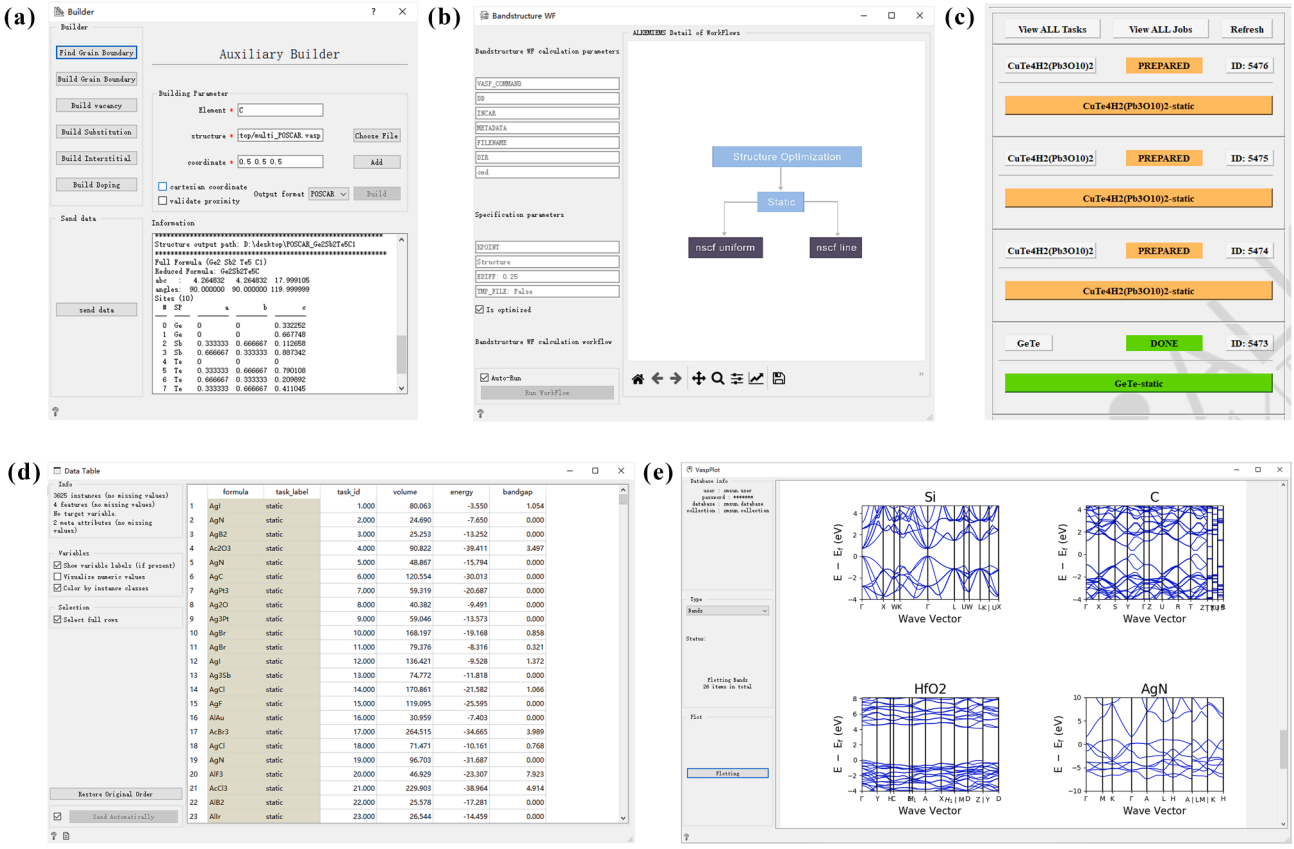

Fig. 4. The detail and parameter configuration for (a) widget *Builder*, (b) band structure workflow, (c) work state, (d) data table and (e) band structure plotter. For clarity, figure (a)–(e) is shown separately as Fig. S4–S8 in the Supplemental Materials.

density of states, band structure or elastic tensor, ALKEMIE provides some pre-tested and reliable default settings, which should be very helpful for beginners or nonexperts. More workflows with various computational softwares will be included in the future version of ALKEMIE. The *WorkState* widget is to visualize the working status of tasks in remote servers, as shown in Fig. 4 (c). This widget can display all tasks in different running states in different colors, while details of the job status will be shown after a 'click'. The *WorkState* widget integrates the well-known Firework in Materials Project, but we designed a brand new GUI for ALKEMIE.

4.2. Database in ALKEMIE

The structure of the core database of ALKEMIE (named as ALKEMIE Data-Vault or ALKEMIE-DV) is shown in Fig. 5. As one user account is created, all login information is stored in the *UserDatabase* and a private database is initialized simultaneously. All data related to high-throughput calculations are stored in this private database. However,

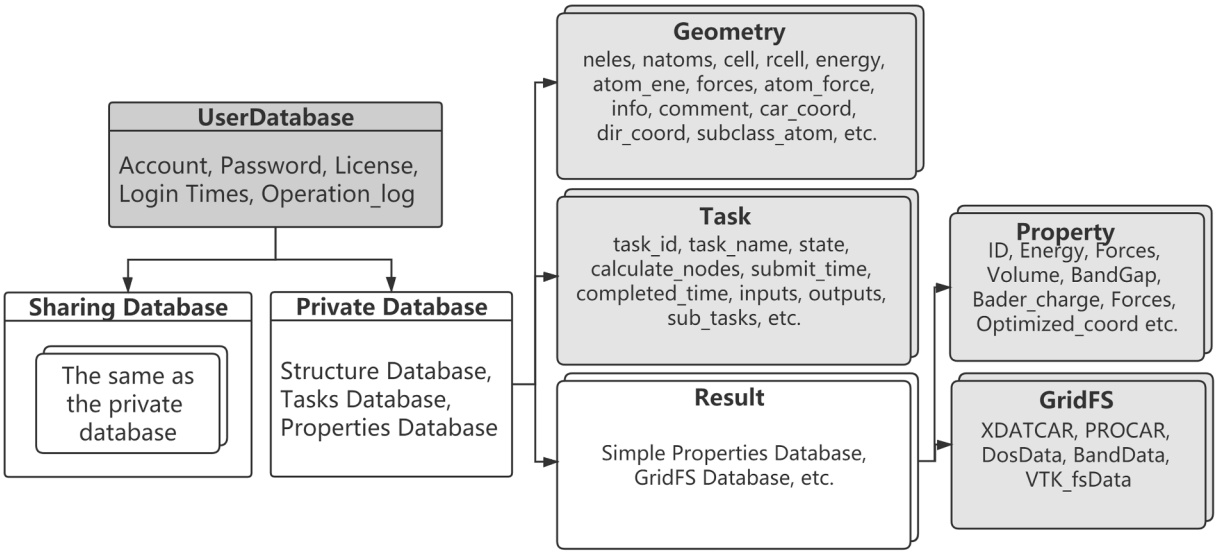

Fig. 5. The structure of the database in ALKEMIE (ALKEMIE DATA VAULT).

when the amount of data increases, and also due to different data types having different data formats, the query efficiency can not be guaranteed. To solve these problems, we divided the private database into three types of sub-databases based on the *pmatgen-db* of Materials Project[31] and inspired by the 'key values' design of AFLOWLIB database [44]. The content in each sub-database is briefly illustrated in Fig. 5.

- **Geometry Database:** This sub-database contains the material geometries involved in each computation process, including the original geometries and optimized ones. Further, the chemical environment of each atom is also analyzed and saved, prepared for data mining and machine learning.
- **Task Database:** Task database is used to store job status, the linkage between tasks, and the job running details from the hardware, such as the time cost. Overall, this database is for the real-time monitoring of the workflows.
- **Property Database:** The property database stores the calculation results in the standardized data format, ready for further data analysis or data visualization. The property database contains basic property data for quick query (such as energy, volume, band gap, etc.). Fig. 4 (d) shows the review of data in the form of *Table*. In addition, large data files are stored via *Gridfs* in MongoDB, for example, the DOS or band profile for each atom and atomic orbits.

Users can share their own data after labeling to claim the ownership, and the data label has the format: *alkemie.date.classification/user_defined_label.number*. This label has two parts divided by a slash. The first part will be automatically filled in and the field of *'alkemie'* is the data source identification, which cannot be changed. *'date'* is the instant time generated through the time module in python with the accuracy of microseconds. *'classification'* has 4 types: geometry, task, property and others, corresponding to the above three types of databases and uncategorized data. The second part is user definable, where *'user_defined_label'* is a custom label within 20 characters and *'number'* is an accumulating number index. *'date'* and *'number'* can make sure the uniqueness of the label for the shared data, *'classification'* and *'user_defined_label'* facilitate the data query. Further, this label defined in ALKEMIE can be easily expanded to the DOI format by adding DOI identification number and organization identification number at the front, and thus the database can be shared to a wide community. The data added to the shared database will be double checked by the administrator.

For the database in ALKEMIE, in addition to the fundamental storage function, we also provide a basic search engine for material properties named as *BSEngine*, and data analysis tools. For different computing softwares, users can inherit functions from *BSEngine* and develop their own search engine and data analysis method. For example, on one hand, *VASPBSEngine* can query the number of atoms, energy, volume and band gap, returning the data with *JSON* format for analysis, as shown in Fig. 4 (d). On the other hand, it can also query the data of DOS, PDOS and PBANDS, and return the data with python-dictionary format for further analysis. Fig. 4 (e) shows the output of the widget for plotting the band structure which is designed for the automatic plotting of the multiple band structures based on the data provided by *VASPBSEngine*.

4.3. Machine learning in ALKEMIE

For machine learning, the quality of the input data almost determines the accuracy of the machine learned model. In material science, the material-property data can be easily transformed into the applicable format for machine learning by means of matrix transpose and normalization. However, for the material structure information, normally specific descriptors are needed to analyze the atomic environment around a chosen atom, which thus transform the geometry structure into the high-quality initial data for machine learning. In ALKEMIE, *FingerSTP* is designed to specify the environmental descriptor for atoms of materials. So far, we have included Behler's and Chebyshev's methods as the structural fingerprints [67,68]. The information related to the atomic structure includes the atomic coordinates, as well as the force and energy corresponding to each atom. Therefore, we choose the extended XSF format to store all these atomic structure information [69]. Ultimately, with the initial structure and the chosen descriptor, users can quickly parse the structure information into binary data through a generator named *GenerateXSF*, and specify parameters to convert data into *TFdata* format or Pytorch pipeline format if needed.

After the preparation of input datasets, users can choose their needed machine learning algorithms in ALKEMIE that has incorporated many popular machining learning packages including scikit-learn, Pytorch, and Tensorflow. More importantly, in ALKEMIE, we have developed advanced high-level API for Pytorch in order to reduce the number of hyperparameters that need to be specified during the training process. If a machine learning model is trained and tested to be accurate, it can be saved and deployed to predict the material property directly in a new dataset. Inspired by the atom simulation environment (ASE), we summarize the accurate models into a *calculator-lib*, for the convenience of their re-loading. For instance, in ALKEMIE, we have trained a potential related calculator named *PmCalculator*, through which one can calculate the energy and force of each atom during large scale MD simulations.

5. Examples

In this section, we present a few case studies to illustrate the main features of ALKEMIE, including the high-throughput calculations and machine learning functions.

5.1. High-throughput static calculation workflows using VASP.

In order to demonstrate the easy-to-use, reliability and efficiency of the high-throughput calculations in ALKEMIE, 1000 binary compounds are randomly chosen from the widely used Inorganic Crystal Structure Database (ICSD) [46], and the formula as well as the basic character of these compounds can be seen in Table S1 of the Supplemental Materials. Then, we performed HT cohesive energy calculations on these compounds using VASP, and the flowchart of the workflow is shown in Fig. S9 in Supplemental Materials. By using the widgets *VASPBSEngine* and *DataAnalyzer* which have integrated the *open-source* plotting packages Matplotlib [70] and Seaborn [71], the data from the HT calculations, such as the energy, volume, and band gaps, are summarized in Fig. 6. Through this example, the reliability of the high-throughput calculations in ALKEMIE can be verified, and in practice the number of tasks that ALKEMIE can manage in HT calculations can be much more than 1000 if needed.

5.2. High-throughput screening of the easy-alloying elements in Cu

High-throughput screening of an optimal dopant or a suitable alloying element for a target function of materials is one of the most important strategies to design the new materials [72,17,73]. In our previous work, we employed the HTC workflow in ALKEMIE to get the most stable carbon configurations on the grain boundary of polycrystalline $GeSb_2Te_4$ at different doping concentrations [74]. In addition, with various HTC workflows in ALKEMIE the optimal dopants for Sb_2Te_3 phase-change memory material are found, and the excellent performance of the screened system has been confirmed by experiments [75]. Here we present a high-throughput formation energy workflow to screen the most energy favorable structure of coper alloys with different alloying element, and the flowchart of the workflow is shown in Fig. S10 in the Supplemental Materials. All the 87 elements in the periodic table except for the elements that lack of first-principles potential (marked as gray in Fig. 7 (a)) are calculated. The formation energy for alloying the element in Cu at the concentration of 0.925% is calculated using Eq. (1).

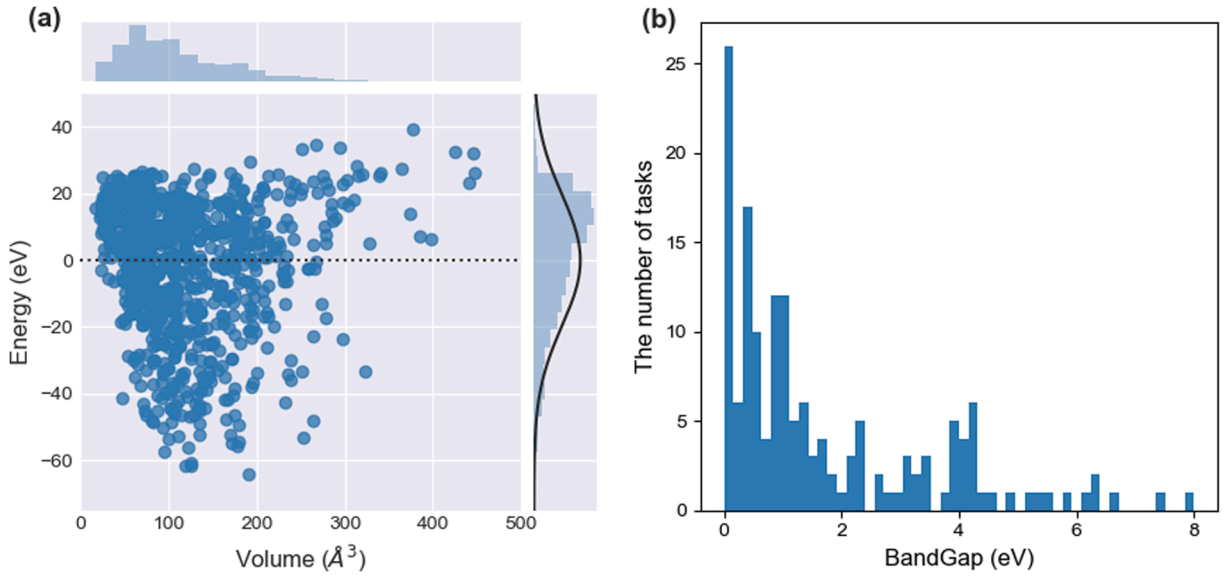

Fig. 6. The result of the workflow of HT static calculation on 1000 binary compounds employing VASP. (a) The scatter plot and distribution of the calculated energies and volumes. (b) The histogram of all no-zero band gaps of the compounds.

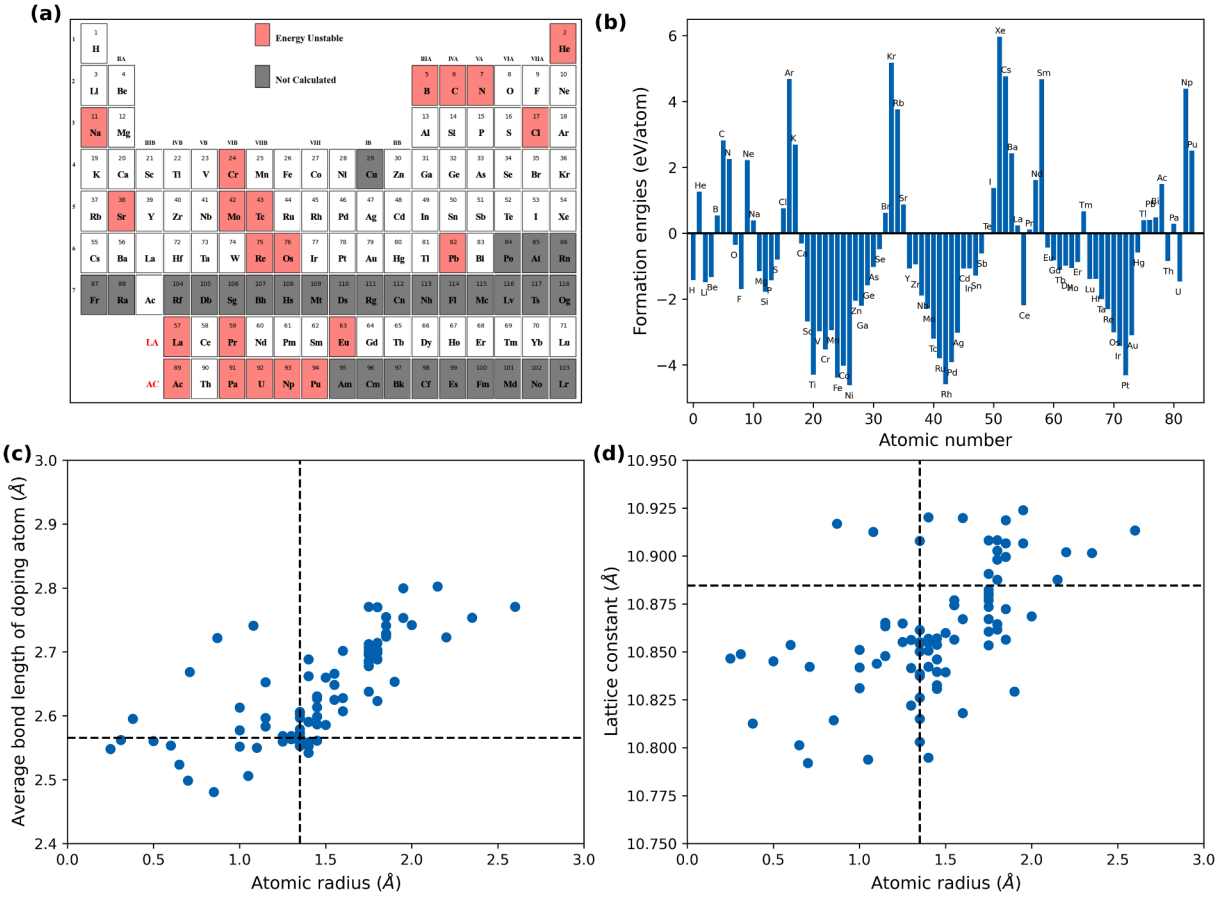

Fig. 7. (a) Illustration of the not calculated elements (gray), the energetically unstable (red) and stable elements (white) in Cu, respectively. (b) The formation energy of the 87 alloying elements. (c) The average bond length and (d) lattice constant of the alloyed compounds, and the dashed lines indicate the value of the pure Cu. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$E^f[x] = E_{tot}[x] - E_{tot}[bulk] - n\mu_c \quad (1)$$

where $E_{tot}[x]$ and $E_{tot}[bulk]$ are the total energy of the supercell with and without the alloying element, respectively. n represents the number of the alloyed atoms, and μ_c is the chemical potential of the alloying elements, which can also be calculated in a high-throughput workflow. The calculated formation energy is summarized in Fig. 7 (b), and 56 alloyed elements (marked as white in Fig. 7 (a)) have negative formation

energies, indicating an easy-soluble in Cu. Further, Fig. 7 (c) shows that overall the average bond length of X-Cu increases with the atomic radius of X (X represents the alloying element). However, a few alloying elements with small atomic radius lead to longer bond-length than Cu-Cu, the underlying mechanism should be correlated to the chemical effects and will be analyzed in detail in future study. The relationship between the lattice constant and the atomic radius is shown in Fig. 7 (d). It is worth noting that the replacement of Cu using some larger-radius atoms will shorten the lattice constant, and this anomalous lattice shrinkage should be induced by the formation of the shorter ionic bonds as a result of the electron transfer from larger-radius alloyed-elemental atom to its neighboring atom with different electronegativity, as shown in our previous work [76]. In a word, here we show that using ALKEMIE it is convenient to define the workflow to evaluate the alloying effects of different elements, which should be helpful for the defect engineering of materials.

5.3. Machine-learned potential

As materials HTC and materials database grow in size and scope, the role of machine learning methods in building predictive models becomes more significant. In our previous work, we designed a HTC for the vacancy-formation energy in the crystalline/amorphous structure, and connected these energy data to the bonding environment by using the machine learning method [77]. Nowadays, developing an accurate interatomic potential for classic molecular dynamic simulations by using machine learning algorithm combined with high-throughput first-principles calculations, attracts growing attentions of researchers. Sosso Gabriele [78], Behler Jörg [68], Mocanu Felix [79], Deringer Volker [80] and Csányi, Gábor [81] have constructed many models with different algorithms and various descriptors to develop the interatomic potential. In this example, we introduce a potential generation strategy by combining the deep neutral network (DNN) algorithm with an 'extended' Behler's symmetry function where we added two descriptors, i.e., distance scaling factor and the number of neighboring atoms, on the basis of the original function. Here, we will show the basic procedure of this potential generation, and the details will be introduced in the future. The flowchart of the workflow is shown in Fig. S11 in the Supplemental Materials.

The generation of the interatomic potential for Sb, a candidate for phase-change memory materials [82–84], is taken as the example. Firstly, a total of 22 inequivalent Sb crystalline configurations are collected from three databases (Materials Project, AFLOW-lib and ALKEMIE-DV), relaxed by the *optimize-workflow*. Then HT *ab initio* molecular dynamics (AIMD) simulation at different temperature are performed using the Sb supercells, and finally 3323 structures comprising of crystal (73.131%), amorphous (7.446%) and liquid (19.423%) states are extracted as the data set, as shown in Fig. 8 (a). The parameters of the Behler's symmetry function used here are listed in Table 1, and *GenerateXSF* in ALKEMIE converts the 3323 structures into a 3323 * 260 matrix using these symmetry functions and stores it in a file with the format that Pytorch can read directly. Afterwards, we choose the deep learning model in ALKEMIE to train the potential. The number of hidden layers is set as 7 and the number of nodes in each layer are 1000, 1000, 700, 500, 100, 50, 10, respectively. The batch size and dropout are 1280 and 0.3, respectively; the activation function used is tanh. The loss value

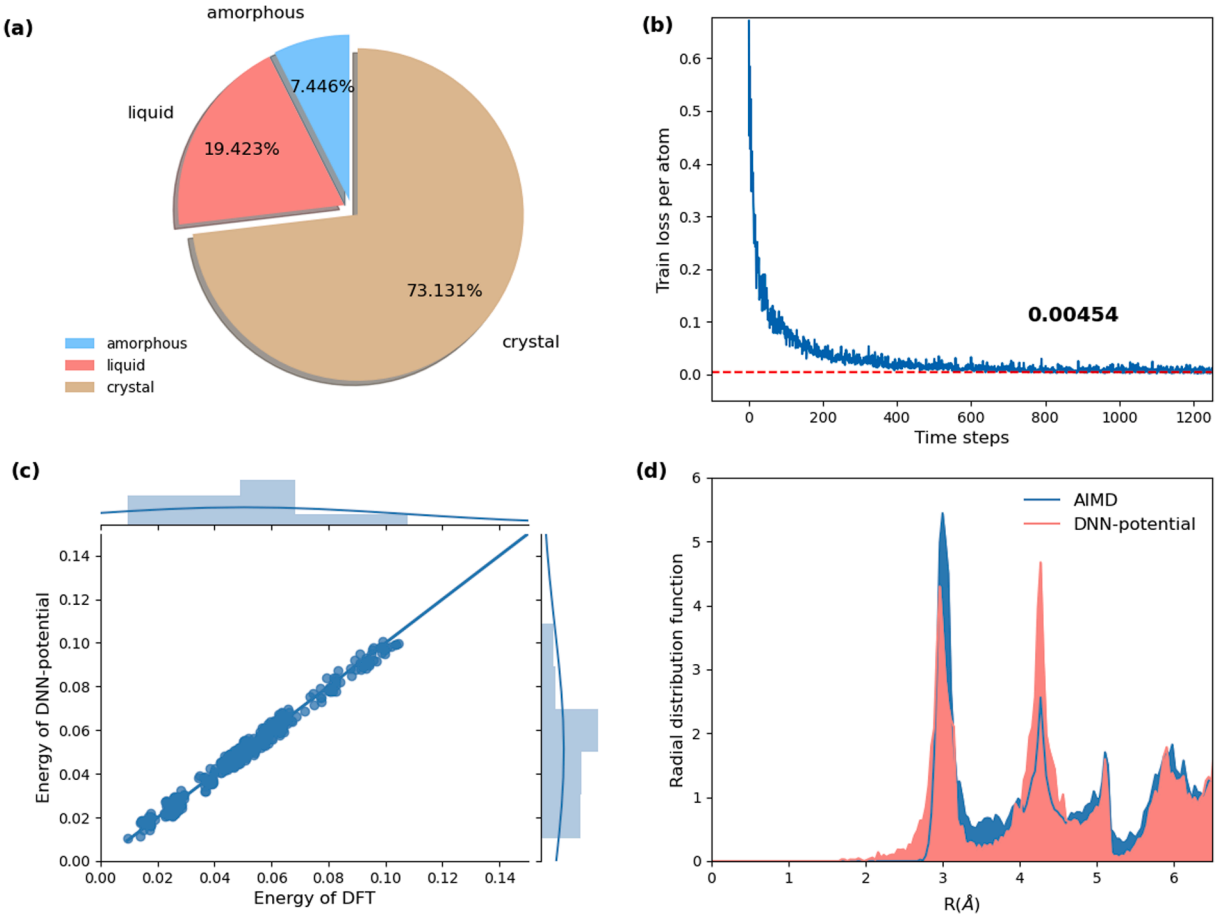

Fig. 8. (a) The constitution of the input dataset. (b) The loss values in training progress. (c) Comparison of the energy predicted by the DNN potential with the corresponding energy by DFT computation. (d) The partial pair distribution function by AIMD (blue) and DNN potential (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
The paramaters in Behler's symmtery functions used in the Sb potential fitting

Type	eta	Rs	Rc	zeta	lambd
G = 2	0.01	2	3.5	–	–
G = 2	0.1	2	3.5	–	–
G = 2	1	2	3.5	–	–
G = 2	10	2	3.5	–	–
G = 2	0.01	3	3.5	–	–
G = 2	0.1	3	3.5	–	–
G = 2	1	3	3.5	–	–
G = 2	10	3	3.5	–	–
G = 5	0.05	–	3.5	2	−1.0
G = 5	0.1	–	3.5	2	−1.0
G = 5	0.008	–	3.5	2	−1.0
G = 5	0.05	–	3.5	2	1.0
G = 5	0.1	–	3.5	2	1.0
G = 5	0.008	–	3.5	2	1.0
G = 5	0.05	–	3.5	4	−1.0
G = 5	0.1	–	3.5	4	−1.0
G = 5	0.008	–	3.5	4	−1.0
G = 5	0.05	–	3.5	4	1.0
G = 5	0.1	–	3.5	4	1.0
G = 5	0.008	–	3.5	4	1.0
G = 5	0.05	–	3.5	16	−1.0
G = 5	0.1	–	3.5	16	−1.0
G = 5	0.008	–	3.5	16	−1.0
G = 5	0.05	–	3.5	16	1.0
G = 5	0.1	–	3.5	16	1.0
G = 5	0.008	–	3.5	16	1.0

in the training is shown in Fig. 8 (b), and the mean absolute error (MAE) of energy and force is 0.0045 per atom. Fig. 8 (c) compares of the energy of various atomic structures predicted by the DNN potential with those computed by DFT, and very good agreement is clearly seen for both the energy data and their distribution. Finally, we export this DNN model (potential) to the *PmCalculator*, and run the classic molecular dynamics simulation with the ASE-MD software which is also integrated in ALKEMIE. As shown in Fig. 8 (d), the radial distribution function (RDF) obtained by the MD with DNN potential and the AIMD exhibits good agreement. The features of RDF reveal that the DNN potential accurately locates the first and second coordinate shell (corresponding to the position of the two high peaks in RDF) of the Sb atoms. However, the intensity of the peaks shows discrepant between MD and AIMD, mainly due to the displacements of the atoms caused by the less accurate forces on the atoms during MD simulations. This less-accurate force issue in DNN potential development might be solved by adjusting the hyperparameters in the machine learning model and the constitution of the input dataset, as we will show in detail in future work. Overall, in ALKEMIE, we have packaged a module that standardizes and simplifies the process of interatomic potential fitting based on DNN.

6. Conclusions

In summary, here we have presented an *open-source* computational platform ALKEMIE, which facilitates the data generation via high-throughput calculations, data management with the private/shared database, and data mining through machine learning. In ALKEMIE, we have integrated and redesigned the popular *open-source* codes and frameworks in computational materials science, while more importantly, we designed many new functions/tools to make ALKEMIE more general, easy-to-use and intelligent. The main 'keywords' of ALKEMIE include high-throughput calculation, automation, visualization, workflow, database, machine learning and plug-in mode. One of most striking features of ALKEMIE is its extremely user-friendly GUI, which makes the high-throughput calculations and workflows 'visible' and very convenient to design, manage and monitor. For data management in ALKEMIE, *BSEngine* is defined to make the searching of the calculated data efficiently in model database, task database, property database and sharing database. In addition, some smart data analysis tools are included in ALKEMIE. More importantly, for the machine learning functions in ALKEMIE, *GenerateXSF*, machine learning model API and *Pmcalculator* make the dataset preparation, model training, and model re-load more conveniently in machine learning application. Particularly, ALKEMIE is featured in the machine-learned interatomic potential development for classic molecular dynamic simulation. We demonstrated the main features of ALKEMIE by using three case studies.

To conclude, ALKEMIE will bring the data-driven techniques for materials designs to a broader community even for the beginners and nonexperts, stimulating the application of these techniques to design new material at less cost. The under-development version of ALKEMIE will be interfaced with more computational softwares at different length/time scales, and cross-scale modeling function will be designed as well.

7. Data availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Declaration of Competing Interest

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

Acknowledgements

This work is financially supported by the National Key Research and Development Program of China (Grant No.2017YFB0701700), and the National Natural Science Foundation of China (Grant No.51872017).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.commatsci.2020.110064.

References

[1] A. Agrawal, A. Choudhary, Perspective: materials informatics and big data: realization of the fourth paradigm of science in ss science, APL Materials 4 (5) (2016) 3208–3218, https://doi.org/10.1063/1.4946894.
[2] N. Science, T. Council, Materials genome initiative for global competitiveness, Executive Office of the President, National Science and Technology Council (US), 2011, https://obamawhitehouse.archives.gov/sites/default/files/microsites/ostp/materials_genome_initiative-final.pdf.
[3] G.H. Johannesson, T. Bligaard, A.V. Ruban, H.L. Skriver, K.W. Jacobsen, J. K. Norskov, Combined electronic structure and evolutionary search approach to materials design, Physical Review Letters 88 (25) (2002) 5506–5510, https://doi.org/10.1103/PhysRevLett.88.255506.
[4] W.F. Maier, K. Stowe, S. Sieg, Combinatorial and high-throughput materials science, Angewandte Chemie International Edition 46 (32) (2007) 6016–6067, https://doi.org/10.1002/anie.200603675.
[5] J. Allison, Integrated computational materials engineering: a perspective on progress and future steps, Jom 63 (4) (2011) 15–16, https://doi.org/10.1007/s11837-011-0053-y.
[6] G. Hautier, A. Jain, S.P. Ong, From the computer to the laboratory: materials discovery and design using first-principles calculations, Journal of Materials Science 47 (21) (2012) 7317–7340, https://doi.org/10.1007/s10853-012-6424-0.
[7] S. Curtarolo, G.L. Hart, M.B. Nardelli, N. Mingo, S. Sanvito, O. Levy, The high-throughput highway to computational materials design, Nature Materials 12 (3) (2013) 191–201, https://doi.org/10.1038/nmat3568.
[8] S.L. Moskowitz, The Advanced Materials Revolution: Technology and Economic Growth in the Age of Globalization, John Wiley and Sons, 2014.
[9] S.V. Kalinin, B.G. Sumpter, R.K. Archibald, Big-deep-smart data in imaging for guiding materials design, Nature Materials 14 (10) (2015) 973–980, https://doi.org/10.1038/nmat4395.
[10] C. Nyshadham, C. Oses, J.E. Hansen, I. Takeuchi, S. Curtarolo, G.L.W. Hart, A computational high-throughput search for new ternary superalloys, Acta Materialia 122 (2017) 438–447, https://doi.org/10.1016/j.actamat.2016.09.017.
[11] C.C. Fischer, K.J. Tibbetts, D. Morgan, G. Ceder, Predicting crystal structure by merging data mining with quantum mechanics, Nature Materials 5 (8) (2006) 641–646, https://doi.org/10.1038/nmat1691.

[12] G.R. Schleder, A.C. Padilha, C.M. Acosta, M. Costa, A. Fazzio, From DFT to machine learning: recent approaches to materials science–a review, Journal of Physics: Materials 2 (3) (2019) 2001–2047, https://doi.org/10.1088/2515-7639/ab084b.
[13] L.-C. Lin, A.H. Berger, R.L. Martin, J. Kim, J.A. Swisher, K. Jariwala, C.H. Rycroft, A.S. Bhown, M.W. Deem, M. Haranczyk, In silico screening of carbon-capture materials, Nature Materials 11 (7) (2012) 633–641, https://doi.org/10.1038/nmat3336.
[14] K. Yang, W. Setyawan, S. Wang, M.B. Nardelli, S. Curtarolo, A search model for topological insulators with high-throughput robustness descriptors, Nature materials 11 (7) (2012) 614–619, https://doi.org/10.1038/nmat3332.
[15] R.P. Hertzberg, A.J. Pope, High-throughput screening: new technology for the 21st century, Current Opinion in Chemical Biology 4 (4) (2000) 445–451, https://doi.org/10.1016/S1367-5931(00)00110-1.
[16] J. Greeley, T.F. Jaramillo, J. Bonde, I. Chorkendorff, J.K. Nørskov, Computational high-throughput screening of electrocatalytic materials for hydrogen evolution, Nature Materials 5 (11) (2006) 909–913, https://doi.org/10.1038/nmat1752.
[17] R. Armiento, B. Kozinsky, M. Fornari, G. Ceder, Screening for high-performance piezoelectrics using high-throughput density functional theory, Physical Review B 84 (1) (2011) 4103–4105, https://doi.org/10.1103/PhysRevB.84.014103.
[18] A. Jain, G. Hautier, C.J. Moore, S.P. Ong, C.C. Fischer, T. Mueller, K.A. Persson, G. Ceder, A high-throughput infrastructure for density functional theory calculations, Computational Materials Science 50 (8) (2011) 2295–2310, https://doi.org/10.1016/j.commatsci.2011.02.023.
[19] W. Setyawan, R.M. Gaume, S. Lam, R.S. Feigelson, S. Curtarolo, High-throughput combinatorial database of electronic band structures for inorganic scintillator materials, ACS Combinatorial Science 13 (4) (2011) 382–390, https://doi.org/10.1021/co200012w.
[20] C.J. Pickard, R. Needs, Ab initio random structure searching, Journal of Physics: Condensed Matter 23 (5) (2011) 3201–3223, https://doi.org/10.1088/0953-8984/23/5/053201.
[21] D.D. Landis, J.S. Hummelshoj, S. Nestorov, J. Greeley, M. Dulak, T. Bligaard, J. K. Norskov, K.W. Jacobsen, The computational materials repository, Computing in Science and Engineering 14 (6) (2012) 51–57, https://doi.org/10.1109/MCSE.2012.16.
[22] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, The materials data facility: Data services to advance materials science research, Jom 68 (8) (2016) 2045–2052, https://doi.org/10.1007/s11837-016-2001-3.
[23] J. Hill, A. Mannodi-Kanakkithodi, R. Ramprasad, B. Meredig, Materials data infrastructure and materials informatics, Computational Materials System Design, Springer, Cham (2018) 193–225, https://doi.org/10.1007/978-3-319-68280-8_10.
[24] S. Curtarolo, D. Morgan, K. Persson, J. Rodgers, G. Ceder, Predicting crystal structures with data mining of quantum calculations, Physical Review Letters 91 (13) (2003) 5503–5506, https://doi.org/10.1103/PhysRevLett.91.135503.
[25] C. Ortiz, O. Eriksson, M. Klintenberg, Data mining and accelerated electronic structure theory as a tool in the search for new functional materials, Computational Materials Science 44 (4) (2009) 1042–1049, https://doi.org/10.1016/j.commatsci.2008.07.016.
[26] M. Chan, G. Ceder, Efficient band gap prediction for solids, Physical Review Letters 105 (19) (2010) 6403–6406, https://doi.org/10.1103/PhysRevLett.105.196403.
[27] T. Mueller, A.G. Kusne, R. Ramprasad, Machine learning in materials science: Recent progress and emerging applications, Reviews in Computational Chemistry 29 (2016) 186–273, https://doi.org/10.1002/9781119148739.ch4.
[28] P. Raccuglia, K.C. Elbert, P.D. Adler, C. Falk, M.B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, A.J. Norquist, Machine-learning-assisted materials discovery using failed experiments, Nature 533 (7601) (2016) 73–76, https://doi.org/10.1038/nature17439.
[29] Y. Liu, T. Zhao, W. Ju, S. Shi, Materials discovery and design using machine learning, Journal of Materiomics 3 (3) (2017) 159–177, https://doi.org/10.1016/j.jmat.2017.08.002.
[30] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, C. Kim, Machine learning in materials informatics: recent applications and prospects, NPJ Computational Materials 3 (1) (2017) 1–13, https://doi.org/10.1038/s41524-017-0056-5.
[31] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, The materials project: A materials genome approach to accelerating materials innovation, APL Materials 1 (1) (2013) 1002–1013, https://doi.org/10.1063/1.4812323.
[32] S.P. Ong, W.D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K.A. Persson, G. Ceder, Python materials genomics (pymatgen): A robust, open-source python library for materials analysis, Computational Materials Science 68 (2013) 314–319, https://doi.org/10.1016/j.commatsci.2012.10.028.
[33] A. Jain, S.P. Ong, W. Chen, B. Medasani, X. Qu, M. Kocher, M. Brafman, G. Petretto, G. Rignanese, G. Hautier, Fireworks: a dynamic workflow system designed for high–throughput applications, Concurrency and Computation: Practice and Experience 27 (17) (2015) 5037–5059, https://doi.org/10.1002/cpe.3505.
[34] https://github.com/materialsproject/custodian.
[35] K. Mathew, J.H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I.-H. Chu, T. Smidt, B. Bocklund, M. Horton, Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows, Computational Materials Science 139 (2017) 140–152, https://doi.org/10.1016/j.commatsci.2017.07.030.
[36] A.R. Supka, T.E. Lyons, L. Liyanage, P. D'Amico, R.A.R. Al Orabi, S. Mahatara, P. Gopal, C. Toher, D. Ceresoli, A. Calzolari, Aflowpi: A minimalist approach to high-throughput ab initio calculations including the generation of tight-binding hamiltonians, Computational Materials Science 136 (2017) 76–84, https://doi.org/10.1016/j.commatsci.2017.03.055.
[37] G. Pizzi, A. Cepellotti, R. Sabatini, N. Marzari, B. Kozinsky, Aiida: automated interactive infrastructure and database for computational science, Computational Materials Science 111 (2016) 218–230, https://doi.org/10.1016/j.commatsci.2015.09.013.
[38] A.H. Larsen, J.J. Mortensen, J. Blomqvist, I.E. Castelli, R. Christensen, M. Dułak, J. Friis, M.N. Groves, B. Hammer, C. Hargus, The atomic simulation environment–a python library for working with atoms, Journal of Physics: Condensed Matter 29 (27) (2017) 3002–3032, https://doi.org/10.1088/1361-648X/aa680e.
[39] https://github.com/mdavezac/lada.
[40] H. Lambert, A. Fekete, J.R. Kermode, A. De Vita, Imeall: A computational framework for the calculation of the atomistic properties of grain boundaries, Computer Physics Communications 232 (2018) 256–263, https://doi.org/10.1016/j.cpc.2018.04.029.
[41] K. Mathew, A.K. Singh, J.J. Gabriel, K. Choudhary, S.B. Sinnott, A.V. Davydov, F. Tavazza, R.G. Hennig, Mpinterfaces: A materials project based python tool for high-throughput computational screening of interfacial systems, Computational Materials Science 122 (2016) 183–190, https://doi.org/10.1016/j.commatsci.2016.05.020.
[42] J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, Materials design and discovery with high-throughput density functional theory: The open quantum materials database (oqmd), Jom 65 (11) (2013) 1501–1509, https://doi.org/10.1007/s11837-013-0755-4.
[43] S. Curtarolo, W. Setyawan, G.L.W. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M.J. Mehl, H.T. Stokes, D.O. Demchenko, D. Morgan, Aflow: An automatic framework for high-throughput materials discovery, Computational Materials Science 58 (2012) 218–226, https://doi.org/10.1016/j.commatsci.2012.02.005.
[44] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R.H. Taylor, L.J. Nelson, G. L. Hart, S. Sanvito, M. Buongiorno-Nardelli, Aflowlib. org: A distributed materials properties repository from high-throughput ab initio calculations, Computational Materials Science 58 (2012) 227–235, https://doi.org/10.1016/j.commatsci.2012.02.002.
[45] C. Draxl, M. Scheffler, The NOMAD laboratory: from data sharing to artificial intelligence, Journal of Physics: Materials 2 (3) (2019) 6001–6010, https://doi.org/10.1088/2515-7639/ab13bb.
[46] A. Belsky, M. Hellenbrandt, V.L. Karen, P. Luksch, New developments in the inorganic crystal structure database (icsd): accessibility in support of materials research and design, Acta Crystallographica Section B: Structural Science 58 (3) (2002) 364–369, https://doi.org/10.1107/S0108768102006948.
[47] M. Quirós, S. Gražulis, S. Girdzijauskaité, A. Merkys, A. Vaitkus, Using smiles strings for the description of chemical connectivity in the crystallography open database, Journal of Cheminformatics 10 (1) (2018) 1–17, https://doi.org/10.1186/s13321-018-0279-6.
[48] https://cmr.fysik.dtu.dk/.
[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, Scikit-learn: Machine learning in python, The Journal of Machine Learning Research 12 (2011) 2825–2830, https://doi.org/10.1524/auto.2011.0951.
[50] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, Tensorflow: A system for large-scale machine learning, 12th Symposium on Operating Systems Design an Implementation (2016) 265–283.
[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems (2019) 8026–8037, https://arxiv.org/abs/1912.01703.
[52] E. Gossett, C. Toher, C. Oses, O. Isayev, F. Legrain, F. Rose, E. Zurek, J. Carrete, N. Mingo, A. Tropsha, S. Curtarolo, Aflow-ml: A restful api for machine-learning predictions of materials properties, Computational Materials Science 152 (2018) 134–145, https://doi.org/10.1016/j.commatsci.2018.03.075.
[53] R. Ouyang, E. Ahmetcik, C. Carbogno, M. Scheffler, L.M. Ghiringhelli, Simultaneous learning of several materials properties from incomplete databases with multi-task SISSO, Journal of Physics: Materials 2 (2) (2019) 4002–4013, https://doi.org/10.1088/2515-7639/ab077b.
[54] L. Ward, A. Dunn, A. Faghaninia, N.E. Zimmermann, S. Bajaj, Q. Wang, J. Montoya, J. Chen, K. Bystrom, M. Dylla, Matminer: An open source toolkit for materials data mining, Computational Materials Science 152 (2018) 60–69, https://doi.org/10.1016/j.commatsci.2018.05.018.
[55] B. Kolb, L.C. Lentz, A.M. Kolpak, Discovering charge density functionals and structure-property relationships with PROPhet: A general framework for coupling machine learning and first-principles methods, Scientific Reports 7 (1) (2017) 1–9, https://doi.org/10.1038/s41598-017-01251-z.
[56] T. Ueno, T.D. Rhone, Z. Hou, T. Mizoguchi, K. Tsuda, COMBO: an efficient bayesian optimization library for materials science, Materials Discovery 4 (2016) 18–21, https://doi.org/10.1016/j.md.2016.04.001.
[57] L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, A general-purpose machine learning framework for predicting properties of inorganic materials, NPJ Computational Materials 2 (2016) 8–14, https://doi.org/10.1038/npjcompumats.2016.28.
[58] K. Choudhary, B. DeCost, F. Tavazza, Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape, Physical

Review Materials 2 (8) (2018) 3801–3808, https://doi.org/10.1103/PhysRevMaterials.2.083801.
[59] https://orange.biolab.si/.
[60] https://www.riverbankcomputing.com/software/pyqt/.
[61] G. Kresse, J. Hafner, Ab initio molecular dynamics for open-shell transition metals, Physical Review B 48 (17) (1993) 115–118, https://doi.org/10.1103/PhysRevB.48.13115.
[62] https://www.mongodb.com/.
[63] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint (2014) arXiv:1412.6980, https://arxiv.org/abs/1412.6980.
[64] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint (2016) arXiv:1609.04747, https://arxiv.org/abs/1609.04747.
[65] https://pypi.org/.
[66] https://docs.python.org/3/library/unittest.html.
[67] J.C. Mason, D.C. Handscomb, Chebyshev Polynomials, CRC Press (2002), https://doi.org/10.1201/9781420036114.
[68] J. Behler, M. Parrinello, Generalized neural-network representation of high-dimensional potential-energy surfaces, Physical Review Letters 98 (14) (2007) 6401–6404, https://doi.org/10.1103/PhysRevLett.98.146401.
[69] http://www.xcrysden.org/doc/xsf.html/.
[70] J.D. Hunter, Matplotlib: A 2D graphics environment, Computing in Science and Engineering 9 (3) (2007) 90–95, https://doi.org/10.1109/MCSE.2007.55.
[71] https://seaborn.pydata.org/index.html.
[72] S. Wang, Z. Wang, W. Setyawan, N. Mingo, S. Curtarolo, Assessing the thermoelectric properties of sintered compounds via high-throughput ab-initio calculations, Physical Review X 1 (2) (2011) 1012–1019, https://doi.org/10.1103/PhysRevX.1.021012.
[73] N. Mounet, M. Gibertini, P. Schwaller, D. Campi, A. Merkys, A. Marrazzo, T. Sohier, I.E. Castelli, A. Cepellotti, G. Pizzi, Two-dimensional materials from high-throughput computational exfoliation of experimentally known compounds, Nature Nanotechnology 13 (3) (2018) 246–252, https://doi.org/10.1038/s41565-017-0035-5.
[74] G. Wang, J. Zhou, S.R. Elliott, Z. Sun, Role of carbon-rings in polycrystalline GeSb2Te4 phase-change material, Journal of Alloys and Compounds 782 (2019) 852–858, https://doi.org/10.1016/j.jallcom.2018.12.228.
[75] S. Hu, B. Liu, Z. Li, J. Zhou, Z. Sun, Identifying optimal dopants for Sb2Te3 phase-change material by high-throughput ab initio calculations with experiments, Computational Materials Science 165 (2019) 51–58, https://doi.org/10.1016/j.commatsci.2019.04.028.
[76] G. Wang, J. Zhou, Z. Sun, First principles investigation on anomalous lattice shrinkage of W alloyed rock salt GeTe, Journal of Physics and Chemistry of Solids 137 (2020) 109220–109222, https://doi.org/10.1016/j.jpcs.2019.109220.
[77] Y. Cheng, L. Zhu, G. Wang, J. Zhou, S.R. Elliott, Z. Sun, Vacancy formation energy and its connection with bonding environment in solid: A high-throughput calculation and machine learning study, Computational Materials Science 183 (2020) 109803–109811, https://doi.org/10.1016/j.commatsci.2020.109803.
[78] G.C. Sosso, G. Miceli, S. Caravati, J. Behler, M. Bernasconi, Neural network interatomic potential for the phase change material GeTe, Physical Review B 85 (17) (2012) 4103–4115, https://doi.org/10.1103/PhysRevB.85.174103.
[79] F.C. Mocanu, K. Konstantinou, T.H. Lee, N. Bernstein, V.L. Deringer, G. Csanyi, S. R. Elliott, Modeling the phase-change memory material, Ge2Sb2Te5, with a machine-learned interatomic potential, The Journal of Physical Chemistry B 122 (38) (2018) 8998–9006, https://doi.org/10.1021/acs.jpcb.8b06476.
[80] V.L. Deringer, G. Csányi, Machine learning based interatomic potential for amorphous carbon, Physical Review B 95 (9) (2017) 4203–4217, https://doi.org/10.1103/PhysRevB.95.094203.
[81] A.P. Bartók, M.C. Payne, R. Kondor, G. Csányi, Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons, Physical Review Letters 104 (13) (2010) 6403–6406, https://doi.org/10.1103/PhysRevLett.104.136403.
[82] Z. Sun, J. Zhou, A. Blomqvist, B. Johansson, R. Ahuja, Formation of large voids in the amorphous phase-change memory Ge2Sb2Te5 alloy, Physical Review Letters 102 (7) (2009) 5504–5507, https://doi.org/10.1103/PhysRevLett.102.075504.
[83] Z. Sun, J. Zhou, R. Ahuja, Structure of phase change materials for data storage, Physical Review Letters 96 (5) (2006) 5507–5510, https://doi.org/10.1103/PhysRevLett.96.055507.
[84] Z. Sun, J. Zhou, R. Ahuja, Unique melting behavior in phase-change materials for rewritable data storage, Physical Review Letters 98 (5) (2007) 5505–5508, https://doi.org/10.1103/physrevlett.98.055505.