

Machine Learning for Advanced Materials Design across Dimensions

by

Xinyu Jiang

A Dissertation

Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved October 2024 by the
Graduate Supervisory Committee:

Houlong Zhuang, Co-Chair

Qiong Nian, Co-Chair

Qijun Hong

Yang Jiao

Yongming Liu

ARIZONA STATE UNIVERSITY

December 2024

ABSTRACT

Metal/ceramic coatings and medium-entropy alloys or high-entropy alloys have received lots of attention due to their outstanding mechanical properties and resistance of corrosion compared to traditional coatings and alloys. Both materials can be designed by adding new elements from their base elements, which present great design opportunities and open up many new structural and functional applications. However, theoretical modeling and simulation by density functional theory (DFT) and classical interatomic potential in both systems are challenging due to the large number of chemical interactions between atoms, and first-principles Monte-Carlo simulation is an expensive computation. With the development of machine learning potential, especially for the descriptors that can describe complex patterns of atomic structures and can be used by neural networks, combining the emerging database of high-quality DFT calculations and efficient algorithms make the machine learning a potential way to solve the aforementioned challenges, though the design of the descriptor sometimes is painstaking, besides, the model is a black box that is hard to understand, and the training of neural network is time-consuming as well. Therefore, three studies are listed in this dissertation to study and mitigate the problem mentioned above by using machine learning. The first study uses a crystal graph convolutional neural network framework combined with reinforcement learning to efficiently find optimal structures of TiAl/TiAlN interface that have high work of adhesion. The second study uses a crystal graph convolutional neural network to efficiently investigate the energy and short-range order of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ medium entropy alloys. The last study introduces an interpretable regression-trees-based ensemble learning approach that

can efficiently predict the properties of structures like carbons with a small size of data to relieve the time-consuming problem of training.

DEDICATION

I would like to dedicate this achievement to my family for their unwavering support and love throughout my doctoral education. No effort on my part will be enough to thank you for the sacrifice you have made on my behalf. I will never succeed without their guidance, influence, and encouragement.

I would like to dedicate this achievement to my friends for their support throughout my doctoral education.

I would like to dedicate this achievement to the staff in the School for Engineering of Matter, Transport and Energy of Arizona State University for their help throughout my doctoral education.

In memory of my paternal grandfather, Xunkai Qiu, who is always in our minds, forever in our hearts.

ACKNOWLEDGMENTS

My deepest gratitude goes first and foremost to my advisors, Dr. Zhuang and Dr. Nian, for the opportunities to work on my research projects and for their guidance, support, and encouragement throughout my doctoral journey.

I would also like to sincerely thank my committee members for helping me develop my research and for providing valuable guidance and encouragement throughout my studies.

I would like to express my heartfelt gratitude to all the group members in Dr. Zhuang and Dr. Nian's labs, it is the harmonious communication environment and exchanging of different ideas that let me keep learning and progress.

Finally, I would like to thank Sol and Agave supercomputers at Arizona State University for providing computational resources. And thank the National Science Foundation for providing funding.

TABLE OF CONTENTS

	Page
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER	
1 INTRODUCTION	1
Overview.....	1
Organization.....	4
2 BACKGROUND	6
Classical Interatomic Potential and Machine Learning Potential	6
Typical Machine Learning Potentials	10
High-Dimensional Neural Network Potentials	10
Gaussian Approximation Potentials.....	19
Smooth Overlap of Atomic Positions	21
Spectral Neighbor Analysis Potential	24
Coulomb Matrix.....	27
Moment Tensor Potential.....	30
Graph Neural Networks	32
Symbolic Regression	36

Physically Informed Machine Learning Potentials.....	38
Atomic Cluster Expansion	42
Other Machine Learning Potentials	46
Machine Learning Potential Software	46
Summary	47
3 COMBINING REINFORCEMENT LEARNING WITH GRAPH CONVOLUTIONAL NEURAL NETWORKS FOR EFFICIENT DESIGN OF TiAl/TiAlN ATOMIC-SCALE INTERFACES	49
Introduction.....	49
Methodology	52
DFT Calculations	52
TiAl/TiAlN Interface Structure and Work of Adhesion.....	52
Crystal Graph Convolutional Neural Networks (CGCNN) for Work of Adhesion Prediction.....	54
Monte Carlo for TiAl/TiAlN Interface Generation	55
Genetic Algorithm (GA) for TiAl/TiAlN Interface Generation	56
Reinforcement Learning (DQN) for TiAl/TiAlN Interface Generation	58
Results and Discussion	63
Performance of CGCNN	63
Performance of DQN	66

TiAl/TiAlN Interface Generation and Work of Adhesion.....	70
Interpretability of CGCNN and RL Model.....	72
Electronic Structure and Bonding.....	76
Summary	82
4 RE-EVALUATING SHORT-RANGE ORDER IN SI-GE-SN MEDIUM ENTROPY ALLOYS USING CRYSTAL GRAPH CONVOLUTIONAL NEURAL NETWORKS AND REINFORCEMENT LEARNING CACULATIONS	
Introduction.....	84
Methodology	86
MC Sampling.....	86
DFT Calculations	86
SRO Parameter.....	87
CGCNN for Total Energy Prediction of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$	87
Reinforcement Learning Agent (A2C)	88
RL Framework for $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ Configurations Design.....	89
Results and Discussion	93
Performance of CGCNN and A2C	93
Energy Landscape and SROs of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$	98
Interpretability of CGCNN and RL Model.....	109

Summary	112
5 INTERPRETABLE ENSEMBLE LEARNING FOR MATERIALS	
PROPERTY PREDICTION WITH CLASSICAL INTERATOMIC POTENTIALS	
.....	113
Introduction.....	113
Methodology	116
Regression Trees of Ensemble Learning	116
Bagging and Boosting Methods.....	118
Data Collections.....	120
Results.....	120
Ensemble Learning Framework for Properties Prediction of Carbon Materials	120
Formation Energy of Different Carbon Structures	123
Elastic Constants of Different Carbon Structures.....	130
Interpretability of Ensemble Learning	141
Formation Energy Prediction of New Carbon Structures.....	151
Discussion.....	153
Summary	160
6 CONCLUSIONS.....	162
REFERENCES	164

LIST OF TABLES

Table	Page
1. Comparison of Machine Learning Models for Atomistic Simulations and Bulk Property Predictions.....	3
2. Performance of Deep Q-network (DQN), Genetic Algorithm (GA), and Monte Carlo (MC) Methods.....	70
3. The SRO for Each Nearest Neighbor Type of Two MC/DFT Simulations with PBE and LDA Exchange-Correlation Functionals.....	107
4. MAEs and MADs of the Partial Elastic Constants Relative to DFT Reference under AB and Tersoff	134
5. Average Feature Importance Calculated by Permutating Features.....	146
6. The PCC, MAE, Feature Importance, and Local Minimum Frequency of RF Trained by Carbon Materials under Different Formation Energy Intervals	150
7. The MAE of Features Corresponds to DFT, and Feature Importance of RF	153
8. The MAE, PCC, feature importance and local minimum frequency of COMB, Tersoff, and Vashishta	157
9. The MAE of RF Trained by Different Features Size.....	158

LIST OF FIGURES

Figure	Page
1. Flowchart of Total Energy Calculated by Classical Interatomic Potentials	8
2. Flowchart of Total Energy Calculations with Machine Learning Potentials.....	10
3. The High-Dimensional Neural Network Potential's Structure Involves Representing Each Atom's Cartesian Coordinates (R_i) in a Neural Network.....	11
4. The Architecture of the High-Dimensional Neural Network Potential (HDNNP) for Multicomponent Systems.....	15
5. The Schematic Structure of a Fourth Generation High-Dimensional Neural Network Potential (4G-HDNNP) for a Binary System.....	17
6. The Illustration of Crystal Graph Convolutional Neural Networks (CGCNN).....	34
7. Example of a Crossover Operation	37
8. Flowcharts of the Development of Atomistic Potentials	41
9. Al-Doped Ti/TiN Interface Structure with 50 mol% of Al Generated from FCC/HCP Ti/TiN Interface.....	54
10. Flow Chart of the Monte Carlo (MC) Procedure	56
11. Flow Chart of the Genetic Algorithm (GA) Procedure	58
12. Schematic of Reinforcement Learning (RL) Model, which Mainly Consists of RL Agent, Environment, and Replay Buffer	63

13. Three Types of Stacking at the Interface: FCC-Like, HCP-Like, and Node-to-Node.....	64
14. Performance of Crystal Graph Convolutional Neural Networks (CGCNN) Model	66
15. Performance of Deep Q-network (DQN) Model	69
16. The Approximate Work of Adhesion (W^{app}) Calculated by Both DFT and Crystal Graph Convolutional Neural Networks (CGCNN) Model	72
17. Visualization of Different Feature Vectors by Principal Component Analysis (PCA)	76
18. Charge Density Difference and Plane-Averaged Charge Density Difference of TiAl/TiAlN Interface	80
19. The Average Charge as a Function of the Atomic Layer at the Interface of Different Structures.....	82
20. Schematic of Reinforcement Learning (RL) Model	93
21. Performance of CGCNN Model under Different Training Dataset.....	96
22. The Evolution of Mean Unrelaxed Energy of Generated Structures by A2C and DQN Models during Training.....	98
23. The Total Energy Fluctuates of MC/CGCNN Trajectories and First-Nearest SROs Variation of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$, and Mean Energy Versus Episodes by A2C/CGCNN during Training	102

24. Variations of SRO for Each Nearest Neighbor Type and Energies in Three MC/DFT Simulations with PBE Exchange-Correlation under Different Initial Structures with 216 Atoms.....	104
25. The Variations of SRO for Each Nearest Neighbor Type in MC/DFT with PBE and LDA Exchange-Correlation Functionals, and the Variation of Energy Corresponding to PBE and LDA Exchange-Correlation Functionals	107
26. Short-Range Order Parameters α_{ij}^{ν} of SQS, Random Mixture, Low-Energy First Nearest-Neighbor S-SRO Structure Predicted by MC/CGCNN, and Low-Energy First Nearest-Neighbor S-SRO Structure Predicted by MC/DFT	109
27. PCA of the Local Environment Representation of 896 Structures Corresponding to SROs and Energy	112
28. The Schematic of a Regression Tree	117
29. Configurations of Two Regression-Tree Based Ensemble Learning Models	119
30. Ensemble Learning Framework for Properties Prediction of Carbon Materials	122
31. Formation Energy of Carbon Allotropes Calculated by 9 Classical Potentials and DFT is Used as the Reference.....	124
32. Predicted Formation Energy Versus DFT Reference of Each Carbon Structure under RF, AB, GB, XGB, VR, and GP by Using 10-fold Corss-Validation.....	127
33. MAEs and MADs of the Formation Energy Relative to DFT Reference under Different Methods.....	129

34. MAEs of Elastic Constants of Different Classical Potentials and Machine Learning Methods	132
35. Error Percentages with Respect to DFT of Different Elastic Constants	135
36. Predicted Elastic Constants under Tersoff, RandomForest (RF), and AdaBoost (AB) Versus DFT Reference	138
37. MAEs of the Formation Energy and Total Elastic Constants Relative to Corresponding DFT Reference under Different Ensemble Methods	140
38. Interpretability of Feature Importance with Different Methods	145
39. Illustration of Splitting Thresholds for Two Features	148
40. Frequency of Local Optimization Occurred in Each Feature	149
41. Performance of Different Methods for Prediction of Formation Energy of Some Carbon Structures.....	153
42. Performance of Ensemble Learning Models for Formation Energy Prediction Using Different Training Sizes	156
43. The Formation Energy of Each Si-O Structure Calculated by RF, CGCNN, ALIGNN, DFT, COMB, Tersoff, and Vashishta.....	156
44. The MAEs of Formation Energy of RF under 2-Nearest Neighbors' Imputation....	
.....	160

CHAPTER 1

INTRODUCTION

Overview

Atomic-scale computer simulations of materials play a very important role in multiscale materials modeling by providing quantitative information to mesoscale models while gaining fundamental intuition into the microscopic mechanisms [1, 2]. In computer simulations, the energy and force calculations of given atomic configurations are critical, and the most accurate calculations are performed by electronic structure methods like density functional theory (DFT) [3, 4]. However, the limitation of running time and system scales make DFT hard to deploy to large systems. Alternatively, classical or empirical interatomic potential models, which express the potential energy as a function of the atomic positions, are proposed, such as embedded atom method (EAM) [5], Tersoff [6], ReaxFF [7], and others. These potentials are often used to model large systems at large time scales efficiently, and due to their physical foundation, they can be generalized to different situations. Their accuracy, however, is limited by their empirical functions.

Metal/ceramic coatings play a pivotal role across various fields [8-10], the ceramic parts, which have high hardness, abrasion resistance, chemical stability, and wear resistance [11, 12], combined with metal parts possessed high toughness to balance ductility and hardness [13]. Besides, medium-entropy alloys (MEAs) are alloys composed of mixtures of three elements in equal or larger proportions, while alloys containing five or more principal elements are often referred to as high-entropy alloys (HEAs) [14]. These MEAs and HEAs are distinctly different from traditional metal alloys, which contain one or two main components and a small number of other

elements. Studies have shown that MEA and HEA have excellent mechanical properties and corrosion resistance compared to conventional alloys [15], as well as unusual thermal and electron transport properties [16]. By adding new elements from their base elements, both materials present great design opportunities and open up many new structural and functional applications.

However, they also present significant challenges for theoretical modeling and simulation by DFT and classical interatomic potential. The first challenge is the difficulty in constructing classical interatomic potentials, especially for MEA and HEA, due to the large number of chemical interactions between atoms. For example, the number of m -site interactions of an n -component system is

$$C(n, m) = \frac{n!}{m!(n-m)!} = \frac{n(n-1)\cdots(n-m+1)}{m(m-1)\cdots 1}. \quad (1)$$

What's more, for the first principles calculations like DFT, it is difficult to simulate both structures due to the limited number of atoms and expensive computations. For example, a million CPU core-hour first-principles thermodynamic simulations are required for a 250-atom CuZn supercell first-principles Monte-Carlo simulation (MC/DFT) [17].

With the development of machine learning, the flexibility and the capability to describe complex patterns of neural networks, combining the emerging database of high-quality DFT calculations and efficient algorithms make machine learning a potential way to solve the aforementioned challenges. Machine learning models have calculated speeds close to classical interatomic potentials while near DFT accuracy [18]. In general, the machine learning applications contain two groups, atomistic simulations of atomic scale and bulk physical property predictions of macro scale.

Table 1 shows the comparison between atomistic simulations and bulk property predictions.

Table 1. Comparison of machine learning models for atomistic simulations and bulk property predictions. Adapted from Table 1 in Ref. [19].

	Atomistic simulation	Bulk property prediction
Purpose of machine learning	Accelerating atomistic simulations with high accuracy	Predicting physical properties
Outputs	Atomic energies, forces, and stresses	Phase formation, crystal structure, elastic constants
Training data	First-principles calculations	First-principles calculations, experimental data
Training data generation	Computationally expensive (hundreds to millions of CPU hours)	Experimentally expensive (hundreds of experiments)
Models/Algorithms	Physical descriptors+ machine learning	Classical supervised machine learning (Decision trees, Gaussian process etc.)

Acronyms: CPU-central processing unit

From Table 1, the machine learning models have wide variety of use in material and have received tremendous attention and applied in materials and molecular research. In recent years lots of machine learning potentials has been published, including neural network potentials [20-23], Gaussian approximation potentials (GAP) [24], spectral neighbor analysis potentials (SNAP) [25], moment tensor potentials (MTP) [26], graph neural networks (GNN) or message passing neural networks (MPNN) [27], atomic cluster expansion (ACE) [28], and many others. Some machine learning potentials have generalized physical descriptors to contain information about local atomic environments and therefore can be used for different systems.

Even though, the design of the descriptor sometimes is time-consuming and requires the professional quality and experience of designers. Besides, the model is a black box that is hard to understand, and the training of neural network is time-consuming as well. Therefore, to mitigate the problems mentioned above, this dissertation will discuss how to use machine learning methods to accelerate the design of metal/ceramic interface and the study of short-range orders of MEA, and proposed a machine learning potential based on regression trees combined with features calculated by classical interatomic potentials for efficient training and predicting different properties of atomic structures.

Organization

In following chapters, this dissertation mainly discusses the concepts of machine learning potentials, and three studies of materials science based on machine learning potentials. CHAPTER 2 firstly provides a background for classical interatomic potentials and machine learning interatomic potentials, and then gives a more in-depth description and review of typical machine learning potentials. CHAPTER 3 uses crystal graph convolutional neural network framework as interatomic potential, combined with reinforcement learning to efficiently find optimal structures of TiAl/TiAlN interface that have high work of adhesion. CHAPTER 4 uses a similar method proposed in CHAPTER 3 to study the energy and short-range order of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ medium entropy alloys and discovers a new short-range order that is different from the short-range order of other SiGeSn compositions. CHAPTER 5 introduces a interpretable regression-trees-based ensemble learning approach which can efficiently predict the properties of structures like carbons with a small size of data to relief the problems related to time-consuming of training. The studies in

CHAPTER 3, 4, and 5 have their own introductory sections, from which the reader can learn more background about each study. And a conclusion summarizes the entire dissertation followed by CHAPTER 6.

CHAPTER 2

BACKGROUND

Classical Interatomic Potential and Machine Learning Potential

Potential energy surface, in which the energy of an electronic state is a function of the coordinates of the nuclei derived from Born-Oppenheimer approximation which treats the atomic nuclei as classical particles when determining the electronic wavefunction. The potential energy surface provides a way to perform atomic-scale calculation and get different information of systems such as energy and thermodynamical properties. Solutions of Schrödinger equation is the most accurate representation of the potential energy surface, though it is not available for most real-world systems. DFT, as an approximate solutions of Schrödinger equation based on the direct quantum mechanical treatment of the electrons, is widely used due in different systems. However, it can only be used for systems containing a few hundred atoms, and a hundred picoseconds for running ab initio molecular dynamics (AIMD). What's more, the size and time scales of many material processes which require statistical sampling of thermodynamic averages, like plastic deformation, fracture, phase transition, etc., greatly exceed the capacity of DFT calculations.

Classical interatomic potentials, an alternative approach to calculate the potential energy surface, offer a solution to extremely accelerate molecular dynamics (MD) simulations though the accuracy is reduced. Classical interatomic potential expresses the potential energy E of a system as a sum of N_{atoms} local atomic energy E_i , as shown in Eq. 2, this treatment accelerates the calculation of total energy since the linear and parallel procedure of the atomic energy. This equation only takes into account the short-range interactions, for long-range interactions like Coulomb and

dispersive interactions need extra terms. Each atomic energy is a function of the atomic distances $\mathbf{R}_i \equiv (\mathbf{r}_{i1}, \mathbf{r}_{i2}, \dots, \mathbf{r}_{in})$ between the center atom i and its neighbor n atoms (Eq. 3). This function satisfies the physical restrictions such as the invariance of the energy under permutation, rotation, and translation of the atoms. \mathbf{p}_i contains a small group of global fitting parameters $\mathbf{p} = (p_1, \dots, p_m)$ which are optimized by experimental data and DFT calculations. And the forces acting on atom i can be computed by $\mathbf{F}_i = -\partial E / \partial \mathbf{r}_i$, where \mathbf{r}_i is the position vector of atom i . Fig. 1 illustrates the flowchart of total energy calculated by classical interatomic potentials. The local atomic energy E_i of the center atom i is computed using atomic coordinates within the cutoff sphere (green region) and fixed values of the potential parameters \mathbf{p} . The local atomic energies of all atoms of the system are summed up (symbol Σ) to obtain the total energy E .

Different materials generally have different $\phi(\mathbf{R}_i, \mathbf{p}_i)$, such as EAM and modified EAM (MEAM) are designed for metallic systems [5, 29], Tersoff and Stillinger-Weber potentials for strong covalent materials such as silicon and carbon [30, 31]. Therefore, due to the different physical meanings between the materials, the classical interatomic potentials are generally incompatible with each other. Once the parameters \mathbf{p} are optimized, the parameters are fixed, and the optimization process heavily depends on the developer's knowledge which makes the development long and painful. Though $\phi(\mathbf{R}_i, \mathbf{p}_i)$ is based on the physical foundations and thus has good transferability and extrapolation.

$$E = \sum_{i=1}^{N_{\text{atoms}}} E_i \quad (2)$$

$$E_i = \phi(\mathbf{R}_i, \mathbf{p}_i) \quad (3)$$

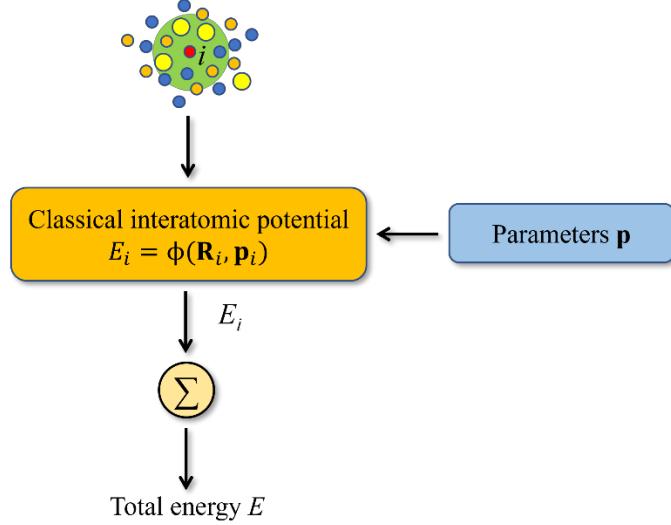


Fig. 1. Flowchart of total energy calculated by classical interatomic potentials. The local atomic energy E_i of the center atom i is computed using atomic coordinates within the cutoff sphere (green region) and fixed values of the potential parameters \mathbf{p} . Total energy E is the summation (symbol Σ) of local atomic energies of all atoms of the system. Adapted from Fig. 1 in Ref. [2].

With the massive database of quantum mechanical calculations arise and development of computer science, machine learning can predict the potential energy of the system without physical meaning by numerically interpolating between the known reference data like quantum mechanical calculations. In other words, machine learning identifies a function f with adjustable parameters \mathbf{p} that accurately predict targets from reference. Similar to classical interatomic potentials, the machine learning potentials predict the local atomic energy and sum them up to get total energy, the local atomic energy is shown in Eq. 4, except that f is a functional function of \mathbf{G}_i which is a function of the position vectors \mathbf{R}_i . \mathbf{G}_i , called descriptors or fingerprints, are smooth functions corresponding to \mathbf{R}_i and invariant under permutations of atoms and rotations and translations of the coordinate axes. Besides,

it also ensures the size of input fixed which is crucial for neural networks. Fig. 2 shows the flowchart of total energy calculations with machine learning potential. The descriptors of the center atom i is computed using atomic coordinates within the cutoff sphere (green region). Then, the descriptors are fed into trained regression model with optimized parameters to get local atomic energy. The local atomic energies of all atoms of the system are summed up (symbol Σ) to obtain the total energy E . Different descriptors have been proposed in recent years, for example, atom-centered symmetry functions (ACSF) [20], bispectrum [24], Coulomb matrix [32], and so on. And a variety of machine learning methods are used to map the descriptors onto the targets, including neural networks [20], Gaussian process regression [24], spectral neighbor analysis potentials (SNAP) [25], etc. The brief introduction and review of different machine learning interatomic potentials and their descriptors are discussed in the next chapter.

$$E_i = f[(\mathbf{G}_i(\mathbf{R}_i), \mathbf{p}_i)] \quad (4)$$

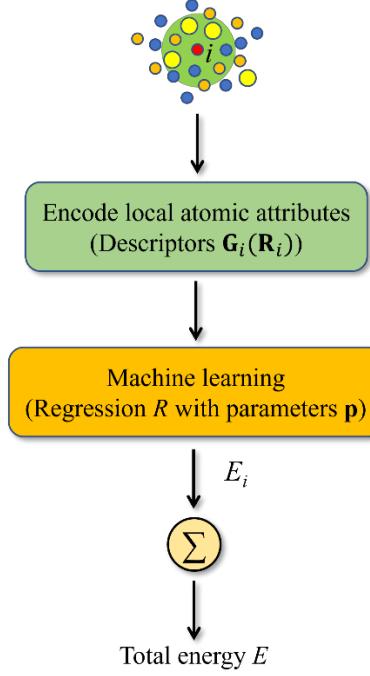


Fig. 2. Flowchart of total energy calculations with machine learning potentials. The local environment of an atom i within the cutoff sphere (green) is encoded in a set of descriptors, which are then mapped onto the local energy E_i using a regression model. The summation of the energies of other atoms of the system (symbol Σ) gives the total energy and thus a point on the PES of the system. Adapted from Fig. 3 in Ref. [2].

Typical Machine Learning Potentials

High-Dimensional Neural Network Potentials

High-dimensional neural network potentials (HDNNPs) are first proposed by Behler and Parrinello which consist of high-dimensional neural networks as the regression model and a set of descriptors [23], as in many conventional empirical potentials, the potential energy E is constructed as a sum of local atomic energies E_i of all atoms i in the system:

$$E = \sum_{i=1}^{N_{\text{atoms}}} E_i \quad (5)$$

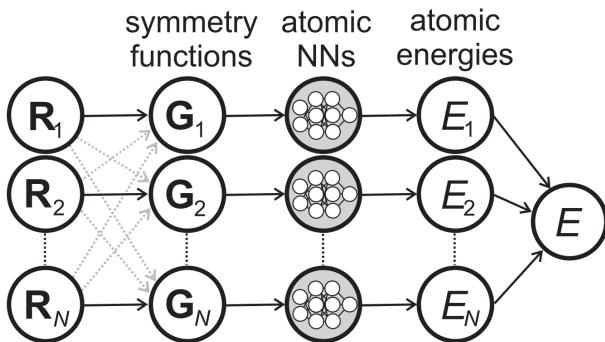


Fig. 3. The high-dimensional neural network potential's structure involves representing each atom's Cartesian coordinates (R_i) in a neural network. Initially, these coordinates undergo a transformation into a set of symmetry function values (G_i). This transformation is dependent on the Cartesian coordinates of all atoms in the local environment, as illustrated by the dotted arrows. The resulting symmetry function values serve as input vectors for atomic neural networks, which, in turn, produce the atomic energy contributions (E_i). The total energy E is the sum of all E_i . Reprinted from Behler, J., The Journal of Chemical Physics, 2011. **134**(7) [20], with the permission of AIP Publishing.

Fig. 3 shows the atom-centered symmetry functions for constructing high-dimensional neural network potentials [20]. For a system with N atoms, R_i represents the cartesian coordinates (x_i, y_i, z_i) of atom i , each atom coordinates are transferred to a set of symmetry function values G_i by considering the cartesian coordinates of all other atoms in the local environment (dotted arrows). These symmetry functions will feed into atomic neural networks as input vectors to calculate the local atomic energy E_i , and the total energy E is the sum of all E_i .

Similar to the Tersoff potential [33]. A cutoff function is applied to ensure that the value and slope of total symmetry functions decays to zero at the cutoff radius.

$$f_c(R_{ij}) = \begin{cases} 0.5 \cdot \left[\cos\left(\frac{\pi R_{ij}}{R_c}\right) + 1 \right] & \text{for } R_{ij} \leq R_c \\ 0 & \text{for } R_{ij} > R_c \end{cases} \quad (6)$$

R_{ij} is the distance between atom i and j , for the neighbor atoms j inside the cutoff radius, this function is monotonously decreasing. The cutoff function and its derivative will become zero if R_{ij} is larger than the cutoff radius R_c which means atoms exceed this the cutoff radius have no contributions to the local atomic energy. The local environment of the atoms inside R_c are described by many-body ACSF consisting of radial and angular functions.

$$G_i^{rad} = \sum_{j=1}^{N_{atom}} e^{-\eta(R_{ij}-R_s)^2} \cdot f_c(R_{ij}) \quad (7)$$

$$G_i^{ang} = 2^{1-\zeta} \sum_{j,k \neq i}^{\text{all}} (1 + \lambda \cos \theta_{ijk})^\zeta \cdot e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} \cdot f_c(R_{ij}) \cdot f_c(R_{ik}) \cdot f_c(R_{jk}) \quad (8)$$

Radial functions are sum of two-body terms and angular functions are sum of three-body terms. The Gaussians of R_{ij} can be shifted by parameter R_s and cutoff function f_c ensure a smooth decay in value and slope, these “shifted” radial functions can describe a spherical shell around the center atom i . The width of the Gaussians is controlled by the parameter η . The angular functions are summation of cosine functions of the angle $\theta_{ijk} = \arccos(\mathbf{R}_{ij} \cdot \mathbf{R}_{ik} / R_{ij} \cdot R_{ik})$, the angular resolution is controlled by the parameter ζ , and $\lambda = \pm 1$ defines the extrema position of the cosine function. Typically, a set of symmetry functions with different η , ζ , R_s , and λ are used to get descriptors. The cutoff R_c needs to be increased until all energy related interactions are included, in general, the value between 6 and 9 Å is enough. All radial and angular functions are rotationally and translationally invariant because they use internal coordinates R_{ij} and θ_{ijk} , they are invariance of permutation as well since the summation of all neighbors ignore the order of atoms.

Besides energy, the forces and stress tensor, which are important in molecular dynamics simulations or metadynamics simulations, also can be calculated by analytic gradients. The force component $F_{k,\alpha}$ acting on atom k is the negative gradients of the total energy respect to the atomic position $\alpha = (x, y, z)$. Due to the transformation of Cartesian coordinates and neural networks, the force can be expressed by the chain rule:

$$F_{k,\alpha} = -\frac{\partial E}{\partial R_{k,\alpha}} = -\sum_{i=1}^N \frac{\partial E_i}{\partial R_{k,\alpha}} = -\sum_{i=1}^N \sum_{j=1}^{M_i} \frac{\partial E_i}{\partial G_{i,j}} \frac{\partial G_{i,j}}{\partial R_{k,\alpha}}, \quad (9)$$

N is the number of atoms, and M_i is the number of symmetry functions. $\partial E_i / \partial G_{i,j}$ and $\partial G_{i,j} / \partial R_{k,\alpha}$ are defined by neural network and symmetry functions, respectively.

Similarly, the static stress σ^{static} can be calculated from the derivatives of local atomic energy.

$$\begin{aligned} \sigma_{\alpha\beta}^{\text{static,rad}} &= \sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \cdot F_{j,\beta} \\ &= -\sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \cdot \frac{\partial E}{\partial R_{j,\beta}} \\ &= -\sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \sum_{k=1}^N \frac{\partial E_k}{\partial R_{j,\beta}} \\ &= -\sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \sum_{k=1}^N \sum_{\mu=1}^{M_k} \frac{\partial E_k}{\partial G_{k,\mu}} \cdot \frac{\partial G_{k,\mu}}{\partial R_{j,\beta}} \\ &= -\sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{\mu=1}^{M_k} R_{ij,\alpha} \frac{\partial E_k}{\partial G_{k,\mu}} \cdot \frac{\partial G_{k,\mu}}{\partial R_{j,\beta}} \\ &= -\sum_{k=1}^N \sum_{\mu=1}^{M_k} \frac{\partial E_k}{\partial G_{k,\mu}} \cdot \sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \frac{\partial G_{k,\mu}}{\partial R_{j,\beta}} \end{aligned} \quad (10)$$

$$\begin{aligned} \sigma_{\alpha\beta}^{\text{static,ang}} &= \sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \cdot F_{j,\beta} + \sum_{i=1}^N \sum_{m=1}^N R_{im,\alpha} \cdot F_{m,\beta} \\ &= -\sum_{k=1}^N \sum_{\mu=1}^{M_k} \frac{\partial E_k}{\partial G_{k,\mu}} \cdot \left(\sum_{i=1}^N \sum_{j=1}^N R_{ij,\alpha} \frac{\partial G_{k,\mu}}{\partial R_{j,\beta}} + \sum_{i=1}^N \sum_{m=1}^N R_{im,\alpha} \frac{\partial G_{k,\mu}}{\partial R_{m,\beta}} \right) \end{aligned} \quad (11)$$

where

$$R_{ij,\alpha} = R_{i,\alpha} - R_{j,\alpha}. \quad (12)$$

It shows this potential can extend time and length scales of MD simulations with high accuracy, which is close to the calculations of DFT, and is 4 to 5 orders of magnitude faster than DFT calculations.

Since Eq. 5 can be written as a sum of atom pair energies E_{ij} :

$$E = \sum_i E_i = \frac{1}{2} \sum_i \sum_{j \neq i} E_{ij}. \quad (13)$$

And the E_{ij} in Tersoff potential is defined as:

$$E_{ij} = f_c(R_{ij}) \cdot [Ae^{-\lambda_1 R_{ij}} - B_{ij}e^{-\lambda_2 R_{ij}}] \quad (14)$$

where $f_c(R_{ij})$ is a cutoff function, the terms in bracket are repulsive and attractive, respectively. B_{ij} is the bond order which is environment-dependent and important for the accuracy of potential. The atom pairs provide more chemical intuitive description and related to the bond order. Therefore, the pair centered symmetry functions as shown in Eq. 15 and Eq. 16 can be used, and Jose *et al.* used methanol and bulk copper test both atom-based and pair-based potentials and they found pair-based potential is slightly higher accuracy [34].

$$G_{ij}^{\text{pair,rad}} = f_c(R_{ij}) \cdot e^{-\eta \cdot R_{ij}^2} \cdot [\sum_k f_c(R_{ik})e^{-\eta \cdot R_{ik}^2} + \sum_k f_c(R_{jk})e^{-\eta \cdot R_{jk}^2}] \quad (15)$$

$$G_{ij}^{\text{pair,ang}} = f_c(R_{ij}) \cdot e^{-\eta \cdot R_{ij}^2} \cdot 2^{1-\zeta} \sum_\alpha [(1 + \lambda \cos \alpha)^\zeta \cdot e^{-\eta \cdot (R_{ik}^2 + R_{jk}^2)} \cdot f_c(R_{ik}) \cdot f_c(R_{jk})] \quad (16)$$

Furthermore, Behler *et al.* also proposed some modified versions. For instance, for multicomponent systems, the charge transfer cannot always be ignored, so an electrostatic term of environment-dependent atomic charges is incorporated. The total energy of a system is a sum over a short-range like the Behler Parrinello descriptor (BP descriptor or ACSF) [23] and a long-range electrostatic energy part. Each term consists of a set of many-body symmetry functions and an atomic neural network (as

shown in Fig. 4). Like local environment energy E_i , the neural network output atomic charges Q_i for each local environment and the electrostatic energy E_{elec} can then be calculated utilizing standard method like an Ewald summation [35]. Based on these functions, some multicomponent systems such as zinc oxide and water dimer neural network potentials are studied, and they have the same accurate as electronic structure calculations [21, 22]. This potential is also reactive; thus, it can describe the break and formation of bonds.

$$E_{\text{tot}} = E_{\text{short}} + E_{\text{elec}} \quad (17)$$

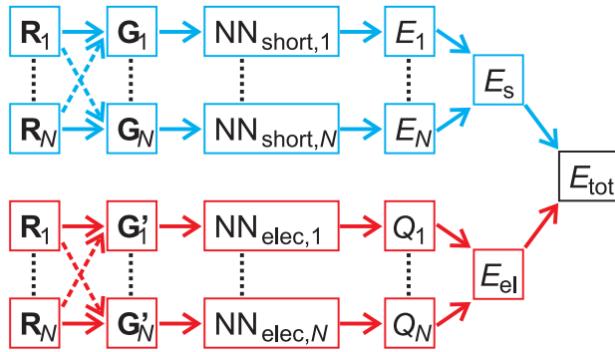


Fig. 4. The architecture of the high-dimensional neural network potential (HDNNP) for multicomponent systems involves deriving the total energy of the system by combining a short-range energy component (E_s) and a long-range electrostatic energy component (E_{el}). The short-range energy (E_s) is computed as the sum of atomic energies, and these atomic energies are influenced by the local chemical environment of the atoms. Additionally, the long-range electrostatic energy (E_{el}) is determined based on the atomic charges (Q_i), and like the short-range energies, these atomic charges also depend on the local chemical environment and are modeled using atomic neural networks. Reprinted figure with permission from Artrith, N., T. Morawietz, and J. Behler, Physical Review B—Condensed Matter and Materials Physics. **83**(15), 153101 (2011) [21]. Copyright 2024 by the American Physical Society.

Even though the long-range interactions are included in the environment-dependent term, nonlocal interactions which depend on global charge distribution are not considered. Thus, inspired by the charge equilibration neural network technique (CENT) [36-38], a fourth-generation high dimensional neural network potential (4G-HDNNP) is proposed by combining the short-range energy and long-range electrostatic energy [39], as shown in Eq. 18 and Fig. 5, where each part depends on both atomic charges $\mathbf{Q} = \{Q_i\}$ and coordinate of atoms $\mathbf{R} = \{\mathbf{R}_i\}$. For atomic charges, they are obtained indirectly from a charge equilibration method using environment-dependent atomic electronegativities $\{\chi_i\}$ predicted from atomic neural networks, which is different from the method shown in Fig. 4 to obtain the charges.

$$E_{\text{total}}(\mathbf{R}, \mathbf{Q}) = E_{\text{elec}}(\mathbf{R}, \mathbf{Q}) + E_{\text{short}}(\mathbf{R}, \mathbf{Q}) \quad (18)$$

The atomic charges are represented by the charge equilibration in which a Gaussian charge density with covalent radius σ_i of element i , atomic electronegativities χ_i , and element-specific hardness J_i are utilized, as illustrated in Eq. 19.

$$E_{\text{eq}} = E_{\text{elec}} + \sum_{i=1}^N \left(\chi_i Q_i + \frac{1}{2} J_i Q_i^2 \right) \quad (19)$$

Where the E_{elec} is the electrostatic energy of Gaussian charges,

$$E_{\text{elec}} = \sum_{i=1}^N \sum_j^N \frac{\text{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}} Q_i Q_j + \sum_{i=1}^N \frac{Q_i^2}{2\sigma_i \sqrt{\pi}} \quad (20)$$

where

$$\gamma_{ij} = \sqrt{\sigma_i^2 + \sigma_j^2}. \quad (21)$$

The atomic charges are distributed under minimal charge equilibration, which can set to zero for the derivatives of E_{eq} ,

$$\frac{\partial E_{\text{eq}}}{\partial Q_i} = 0 \Rightarrow \sum_{j=1}^N A_{ij} Q_j + \chi_i = 0 \quad (22)$$

where A_{ij} is

$$A_{ij} = \begin{cases} J_i + \frac{1}{\sigma_i \sqrt{\pi}}, & \text{if } i = j \\ \frac{\text{erf}\left(\frac{r_{ij}}{\sqrt{2}\gamma_{ij}}\right)}{r_{ij}}, & \text{if } i \neq j \end{cases} \quad (23)$$

these charges are then used to calculate the electrostatic energy and serve as nonlocal input for the short-range atomic neural networks (blue part in Fig. 5). Compared to CENT and other HDNNPs, 4G-HDNNP is suitable for variety of systems such as organic molecules, ionic clusters, and metal clusters.

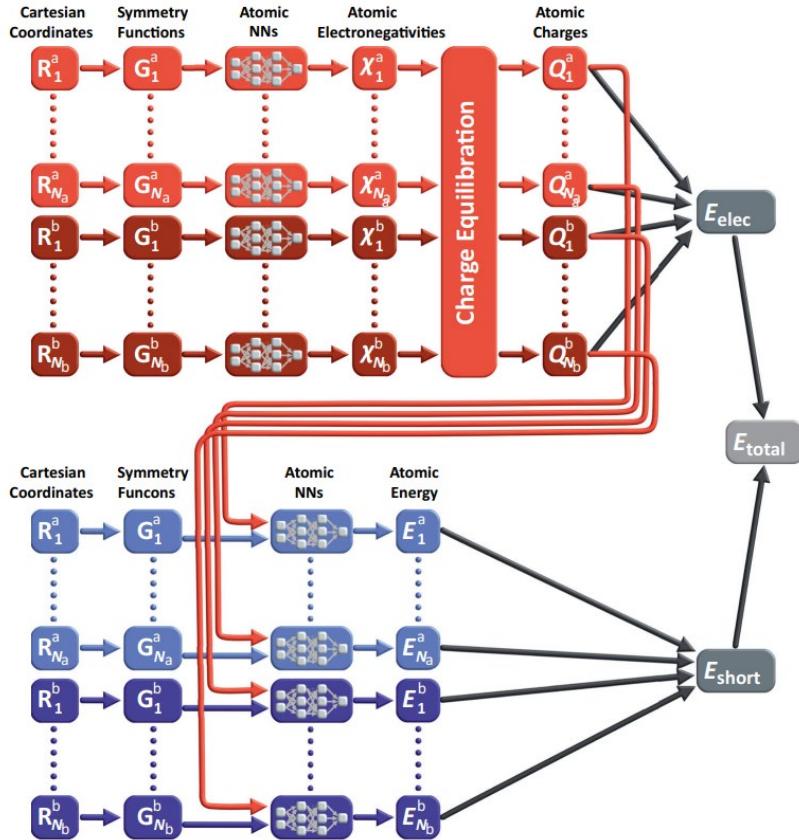


Fig. 5. The schematic structure of a Fourth Generation High-Dimensional Neural Network Potential (4G-HDNNP) for a binary system. In this system, consisting of N_a atoms of element a and N_b atoms of element b , the total energy is composed of a

short-range energy (E_{short}), which is a sum of atomic energies (E_i), and a long-range electrostatic energy (E_{elec}) computed from atomic charges (Q_i). The determination of atomic charges involves a charge equilibration method utilizing environment-dependent atomic electronegativities (χ_i) expressed by atomic neural networks (depicted in red). These charges not only contribute to the calculation of the electrostatic energy but also serve as nonlocal inputs for the short-range atomic neural networks (depicted in blue), ultimately yielding the atomic energies (E_i). The local atomic environments (G_i) are defined by atom-centered symmetry functions (ACSFs).

Reproduced under terms of the Creative Commons CC BY license [39]. Copyright 2021, The Authors, published by Springer Nature.

In general, the weight parameters in neural networks are optimized by calculating the gradients of the energy and force loss function with backpropagation [40], in most cases, the simple form of loss function as shown in Eq. 24 [41].

$$L = \frac{1}{N} \sum_{A=1}^N \left(\frac{E_{A,\text{ref}} - E_{A,\text{NN}}}{N_{\text{atom}}} \right)^2 + \gamma \frac{1}{N} \sum_{A=1}^N \left(\frac{F_{A,\text{ref}} - F_{A,\text{NN}}}{N_{\text{atom}}} \right)^2 \quad (24)$$

Where N is the number of the systems, N_{atom} is the number of atoms in the system, $F_{A,\text{NN}}$ is the gradient of energy $E_{A,\text{NN}}$, and γ is a parameter expressing the importance of forces.

HDNNPs have been proposed for different systems, e.g., single-element bulk system, multi-element bulk systems, and molecular systems [18, 42-52]. For the details of HDNNPs, different papers give a comprehensive introduction and review of HDNNPs and lists some typical applications and their corresponding references [53-55]. Recently, some research based on HDNNPs and ACSFs are studied for different applications, for examples, Hu *et al.* used modified HDNNP to reveal and understand the mechanism of the Si (111)-7×7 surface reconstruction [56], Kývala and Dellago

studied the optimized conditions of the architecture of HDNNPs under different training set size, complexity of descriptors, depth of neural network, and activation functions [57]. Gao *et al.* introduced an open-source implementation of the ANI-based potential, which is a general-purpose neural network-based interatomic potential for organic molecules used modified functions of original ACSFs [58-61], in PyTorch [62], and Abedi *et al.* predicted the equation of state of methane accurately [63].

Gaussian Approximation Potentials

Similar to neural network potentials, Gaussian Approximation Potentials (GAP) interpolates reference quantum mechanical results using high dimensional systems and Gaussian process regression based on Eq. 5 [24]. A local atomic density $\rho_i(\mathbf{r})$ respective for each atom i is firstly applied to construct the descriptor of atomic environment. In the local atomic density, δ functions at all positions of neighboring atoms j in the environment up to the cutoff and at the position of the reference atom i are utilized as

$$\rho_i(\mathbf{r}) = \delta(\mathbf{r}) + \sum_j \delta(\mathbf{r} - \mathbf{r}_{ij}) f_{\text{cut}}(|\mathbf{r}_{ij}|), \quad (25)$$

where $f_{\text{cut}}(r) = 1/2 + \cos(\pi r/r_{\text{cut}})/2$ is a cutoff function which is similar to that used in ACSF. The local atomic density is invariant to permutations of atoms in the neighborhood. By expanding it in spherical harmonics and combine a set of radial basis functions to achieve rotational invariance, however, it will lose some information of the neighborhood. In contrast, the bispectrum [64], a three-point correlation function, contains more information and can provide almost one-to-one representation of the atomic neighborhood. Furthermore, by projecting the atomic density onto a four-dimensional unit sphere, the radial dimension and all the

information from the 3D spherical are contained in the 4D spherical harmonics, which also called Wigner matices, $U_{m'm}^j$ [65]. The 4D spherical harmonics can form a complete basis and without the radial basis functions.

$$(\phi, \theta, \theta_0) = \left[\tan^{-1}\left(\frac{y}{x}\right), \cos^{-1}\left(\frac{z}{|\mathbf{r}|}\right), \frac{|\mathbf{r}|}{r_0} \right] \quad (26)$$

Like 3D spherical harmonics, the atomic density can be expanded in hyperspherical harmonics as

$$\rho = \sum_{j=0}^{\infty} \sum_{m,m'=-j}^j c_{m'm}^j U_{m'm}^j. \quad (27)$$

The coefficient $c_{m'm}^j = \langle U_{m'm}^j | \rho \rangle$ can build the bispectrum matrix.

$$B_{j_1,j_2,j} = \sum_{m'_1,m_1=-j_1}^{j_1} \sum_{m'_2,m_2=-j_2}^{j_2} \sum_{m',m=-j}^j (c_{m'm}^j)^* C_{j_1 m_1 j_2 m_2}^{jm} \times \\ C_{j_1 m'_1 j_2 m'_2}^{jm'} c_{m'_1 m_1}^{j_1} c_{m'_2 m_2}^{j_2} \quad (28)$$

Where $C_{j_1 m_1 j_2 m_2}^{jm}$ are the ordinary Clebsch-Gordan coefficients. The description of the truncated bispectrum is invariant with respect to translation, permutation, and rotation of 4D space, so does in 3D space. This descriptor can be systematically refined by using more and more spherical harmonics [66].

As nonparametric method, Gaussian process regression [67, 68], which possess flexible kernels G and ability of fitting any continuous functions, are used to interpolate the atomic energy in the truncated bispectrum space,

$$\varepsilon(\mathbf{b}) = \sum_n \alpha_n e^{-\frac{1}{2} \sum_l [(b_l - b_{n,l})/\theta_l]^2} \equiv \sum_n \alpha_n G(\mathbf{b}, \mathbf{b}_n) \quad (29)$$

Where n and l are reference configurations and bispectrum components, respectively. θ_l are hyperparameters. The coefficients α_n are computed by the covariance (measure of similarity of the reference configurations) and set of reference values \mathbf{y} ,

$$C_{nn'} = \delta^2 G(\mathbf{b}, \mathbf{b}') + \sigma^2 \mathbf{I} \quad (30)$$

$$\{\alpha_n\} \equiv \boldsymbol{\alpha} = \mathbf{C}^{-1} \mathbf{y} \quad (31)$$

where δ and σ are two hyperparameters and \mathbf{I} is the identity matrix. This potential is orders of magnitude faster than standard plane wave DFT calculations though more expensive than simple analytical potentials. In Ref. [24], the performance of both forces and energy in bulk semiconductors and iron is tested, and it is better than that of Brenner [69], Tersoff [33], and Finnis-Sinclair [70] potentials.

Smooth Overlap of Atomic Positions

Since in general, the PES fitting process is to capture the degree of similarity between the atomic environments described by target and training configurations, the key to PES fitting is not the descriptors but the similarity measure like the kernel G constructed from the descriptors (Eq. 29). Thus, Smooth Overlap of Atomic Positions (SOAP) is proposed to surpass the descriptors and directly measure the similarity between atomic environments [71].

A drawback of the atomic neighbor density defined by Dirac-delta functions (Eq. 25) is that two very close, but not identical atomic environments can lead the discontinuous similarity kernel be large dissimilar. In the SOAP, instead of the Dirac-delta functions, the atomic neighbor density function is defined as a sum of Gaussians with one centered on each neighbor i and expanded in terms of spherical harmonics functions as

$$\rho(\mathbf{r}) = \sum_i \exp(-\alpha|\mathbf{r} - \mathbf{r}_i|^2) = \sum_i \sum_{lm} c_{lm}^i(r) Y_{lm}(\hat{\mathbf{r}}), \quad (32)$$

where \mathbf{r}_i is the vector from the central atom to neighbor i . $\hat{\mathbf{r}}$ is the point on the unit sphere corresponding to the direction of the vector \mathbf{r} , so $\rho(\hat{\mathbf{r}})$ is the projection of $\rho(\mathbf{r})$ onto the unit sphere S^2 , and coefficient c is:

$$c_{lm}^i(r) \equiv 4\pi \exp[-\alpha(r^2 + r_i^2)] \iota_l(2\alpha r r_i) Y_{lm}^*(\hat{\mathbf{r}}_i) \quad (33)$$

ι_l are the modified spherical Bessel functions of the first kind. The similarity of two atomic environments is calculated by the inner product of two atomic neighbor densities ρ and ρ' ,

$$S(\rho, \rho') = \int \rho(\mathbf{r}) \rho'(\mathbf{r}) d\mathbf{r} \quad (34)$$

which satisfies the invariance of permutation, the rotationally invariant similarity kernel can be added by integrating all possible rotations of one of the environments.

$$k(\rho, \rho') = \int |S(\rho, \hat{R}\rho')|^n d\hat{R} = \int d\hat{R} \left| \int \rho(\mathbf{r}) \rho'(\hat{R}\mathbf{r}) d\mathbf{r} \right|^n \quad (35)$$

Therefore, the overlap between an atomic environment and a rotated environment is

$$\begin{aligned} S(\hat{R}) &\equiv S(\rho, \hat{R}\rho') \\ &= \int d\mathbf{r} \rho(\mathbf{r}) \rho'(\hat{R}\mathbf{r}) \\ &= \sum_{i,i'} \sum_{l,m,m'} D_{m'm''}^{l'}(\hat{R}) \int dr c_{lm}^{i*}(r) c_{l'm'}^{i'}(r) \int d\hat{\mathbf{r}} Y_{lm}^*(\hat{\mathbf{r}}) Y_{l'm''}(\hat{\mathbf{r}}) \\ &= \sum_{i,i'} \sum_{l,m,m'} I_{mm'}^l(\hat{R}) = \sum_{l,m,m'} I_{mm'}^l(\hat{R}), \end{aligned} \quad (36)$$

where

$$\tilde{I}_{mm'}^l(\alpha, r_i, r_{i'}) = 4\pi \exp\left[-\frac{\alpha(r_i^2 + r_{i'}^2)}{2}\right] \iota_l(\alpha r_i r_{i'}) Y_{lm}^*(\hat{\mathbf{r}}_i) Y_{lm}^*(\hat{\mathbf{r}}_{i'}) \quad (37)$$

and

$$I_{mm'}^l \equiv \sum_{i,i'} \tilde{I}_{mm'}^l(\alpha, r_i, r_{i'}). \quad (38)$$

So, the rotationally invariant kernel with $n = 2$ and 3 are

$$\begin{aligned} k(\rho, \rho') &= \int d\hat{R} S^*(\hat{R}) S(\hat{R}) \\ &= \sum_{l,m,m'} (I_{mm'}^l)^* I_{\mu\mu'}^l \int d\hat{R} D^*(\hat{R})_{mm'}^l D(\hat{R})_{\mu\mu'}^\lambda \\ &= \sum_{l,m,m'} (I_{mm'}^l)^* I_{mm'}^l, \end{aligned} \quad (39)$$

and

$$k(\rho, \rho') = \int d\hat{R} S(\hat{R})^3 \\ = \sum I_{m_1 m'_1}^{l_1} I_{m_2 m'_2}^{l_2} I_{mm'}^l C_{l_1 m_1 l_2 m_2}^{lm} C_{l_1 m'_1 l_2 m'_2}^{lm'}. \quad (40)$$

In experiments, raising k to some power $\zeta \geq 2$ can accentuate the sensitivity of the kernel, and a normalization by dividing by $\sqrt{k(\rho, \rho)k(\rho', \rho')}$ to ensure the overlap of any environment with itself is one. So, the general form of SOAP kernel is

$$K(\rho, \rho') = \left(\frac{k(\rho, \rho')}{\sqrt{k(\rho, \rho)k(\rho', \rho')}} \right)^\zeta, \quad (41)$$

where ζ is any positive integer.

Note the integral of coefficient $I_{mm'}^l$ is time-consuming due to it requires to be calculated for each pair of neighbors. If expand atomic neighbor density functions (Eq. 32) with radial basis function $g_n(r)$, and the radial basis functions are orthonormal, the $I_{mm'}^l$ becomes:

$$I_{mm'}^l = \sum_n c_{nlm} (c'_{nlm'})^*. \quad (42)$$

Substituting it into Eq. 39 to obtain

$$k(\rho, \rho') = \sum_{n,n',l,m,m'} c_{nlm} (c'_{nlm'})^* (c_{nlm})^* c'_{n'l'm'} \\ \equiv \sum_{n,n',l} p_{nn'l} p'_{nn'l}, \quad (43)$$

where

$$p_{nn'l} \equiv \sum_m c_{nlm} (c'_{nlm})^* \quad (44)$$

is the power spectrum, so the kernel is the dot product of the power spectrum. Similar, when $n = 3$, the kernel can be expressed as

$$k(\rho, \rho') = \sum_{l_1, l_2, l} b_{n_1 n_2 n l l_1 l_2} b'_{n_1 n_2 n l l_1 l_2}, \quad (45)$$

where

$$b_{n_1 n_2 n l_1 l_2 l} \equiv \sum c_{n_1 l_1 m_1} c_{n_2 l_2 m_2} (c_{nlm})^* C_{l_1 m_1 l_2 m_2}^{lm}, \quad (46)$$

which is the bispectrum. So SOAP is equivalent to the SO(3) power spectrum and bispectrum descriptors with Gaussian and covariance kernel. In Ref. [71], the SOAP is compared with other representations of atomic neighbor environment such as BP descriptors, angular Fourier series, SO(3) power spectrum, SO(3) bispectrum, and SO(4) bispectrum descriptors. And the SOAP exhibits outstanding faithfulness regardless of the number of neighbors. The tests on fitting models of small silicon clusters and bulk silicon crystals reveal that SOAP yields a more accurate and robust PES compared to other descriptors. GAPs have been used successfully to model different systems such as silicon [72], carbon [73], tungsten [74], iron [75], and others [76-80]. Besides, GAP are also applied to study the hydrogenation of amorphous silicon (*a*-Si:H) materials [81], and it has better performance than Tersoff [33, 82], vibrational properties of different silicon materials [83], liquid and amorphous Ge₂Sb₂Te₅ which is a chemically complex functional memory material [84, 85], interaction of lithium atoms (lithium intercalation) with carbon nanostructures in lithium ion batteries [86], and thermal conductivity of silicene [87]. Here, it is hard to list all of applications, Deringer *et al.* and Klawohn *et al.* gave an introduction of GAPs and reviewed different applications based on GAPs respectively [88, 89].

Spectral Neighbor Analysis Potential

The Spectral Neighbor Analysis Potential (SNAP), same as the GAP, uses the bispectrum of the neighbor density mapped on the 3-sphere as basis to generate descriptors [25]. The energy of the system consists of N atoms with positions \mathbf{r}^N can be expressed as the sum of a reference energy E_{ref} and a local energy E_{local}

$$E(\mathbf{r}^N) = E_{ref}(\mathbf{r}^N) + E_{local}(\mathbf{r}^N), \quad (47)$$

where reference energy contains well-established physical models, such as long-range electrostatic interactions. The local energy has the remain effects that are not considered by the reference energy, it can be decomposed into contributions of each atomic environment,

$$E_{local}(\mathbf{r}^N) = \sum_{i=1}^N E_i(\mathbf{q}_i) \quad (48)$$

where E_i is the local energy of atom i , which depends on the set of descriptors \mathbf{q}_i , in SNAP the descriptors consist of K bispectrum components $\mathbf{B}^i = \{B_1^i, \dots, B_K^i\}$. Instead of expressing the local energy term by Gaussian process kernel in GAP, the energies and forces can be accurately reproduced by linear combinations of lowest order bispectrum components:

$$E_{SNAP}^i(\mathbf{B}^i) = \beta_0^{\alpha_i} + \sum_{k=1}^K \beta_k^{\alpha_i} B_k^i = \beta_0^{\alpha_i} + \boldsymbol{\beta}^{\alpha_i} \cdot \mathbf{B}^i \quad (49)$$

where α_i is the chemical identity of atom i and β_k^α are the linear coefficients for atoms of type α . So, the total local energy can be written by the summation of bispectrum components for each atom,

$$E_{local}(\mathbf{r}^N) = E_{SNAP}(\mathbf{r}^N) = N\beta_0 + \boldsymbol{\beta} \cdot \sum_{i=1}^N \mathbf{B}^i \quad (50)$$

where $\boldsymbol{\beta}$ is the K -vector of SNAP coefficients and β_0 is the constant energy contribution for each atom. The force acting on atom j is the derivatives of the SNAP energy with respect to the position of atom j ,

$$\mathbf{F}_{SNAP}^j = -\nabla_j E_{SNAP} = -\boldsymbol{\beta} \cdot \sum_{i=1}^N \frac{\partial \mathbf{B}^i}{\partial \mathbf{r}_j} \quad (51)$$

Similarly, the stress tensor can be expressed as

$$\mathbf{W}_{SNAP} = -\sum_{j=1}^N \mathbf{r}_j \otimes \nabla_j E_{SNAP} = -\boldsymbol{\beta} \cdot \sum_{j=1}^N \mathbf{r}_j \otimes \sum_{i=1}^N \frac{\partial \mathbf{B}^i}{\partial \mathbf{r}_j}, \quad (52)$$

where \otimes is the Cartesian outer product operator.

To get the coefficients $\boldsymbol{\beta}$, a linear equation is constructed

$$\begin{bmatrix} \vdots & \vdots \\ N_s & \sum_{i=1}^{N_s} \mathbf{B}^i \\ \vdots & \vdots \\ 0 & -\sum_{i=1}^{N_s} \frac{\partial \mathbf{B}^i}{\partial r_j^\alpha} \\ \vdots & \vdots \\ 0 & -\sum_{j=1}^{N_s} r_j^\alpha \sum_{i=1}^{N_s} \frac{\partial \mathbf{B}^i}{\partial r_j^\beta} \\ \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} E_s^{qm} - E_s^{ref} \\ \vdots \\ F_{j,\alpha}^{qm} - F_{j,\alpha}^{ref} \\ \vdots \\ W_{\alpha\beta,s}^{qm} - W_{\alpha\beta,s}^{ref} \\ \vdots \end{bmatrix}. \quad (53)$$

The optimal solution $\hat{\boldsymbol{\beta}}$ for this set of equations is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \| \mathbf{w} \odot (\mathbf{A} \cdot \boldsymbol{\beta} - \mathbf{y}) \|^2, \quad (54)$$

where \odot is element-wise multiplication by the weight vector \mathbf{w} . The weight vector can ensure the regression process works well, since by giving more weight to the most important material properties, the model can mainly focus on the quantity and configurations that interest researchers.

Due to the limitation of the relationship between linear combination of bispectrum components and energy, a quadratic SNAP model with an extended embedding energy term is proposed to add many-body interactions [90], as shown in Eq. 55,

$$E_{SNAP}^i(\mathbf{B}^i) = \boldsymbol{\beta} \cdot \mathbf{B}^i + F(\rho_i) = \boldsymbol{\beta} \cdot \mathbf{B}^i + \frac{1}{2} F''(\mathbf{a} \cdot \mathbf{B}^i)^2 \quad (55)$$

where $F(\rho_i)$ represents the embedding energy of atom i into the electron density contributed by its neighboring atoms. F'' is constant value of the second derivatives at the reference density ρ_0 in Taylor expansion of the embedding energy (Eq. 56), and ρ_i is a linear function of the bispectrum components (Eq. 57).

$$F(\rho) = F_0 + (\rho - \rho_0)F' + \frac{1}{2}(\rho - \rho_0)^2 F'' + \dots \quad (56)$$

$$\rho_i = \mathbf{a} \cdot \mathbf{B}^i \quad (57)$$

In Ref. [25] and Ref. [90], the SNAPS accurately capture a broad of energetic properties in solid tantalum phases, and the quadratic SNAP has less errors than

original SNAP. Notably, SNAP accurately predicts the size and shape of the Peierls barrier governing screw dislocation motion in BCC tantalum, which is a crucial factor in describing plasticity under shear loading. And this aspect is not adequately addressed by other published potentials such as the embedded atom method (EAM) model and the angular dependent potential (ADP) [91, 92]. While the SNAP methodology demonstrates exceptional performance, a potential drawback lies in its computational cost when applied to large-scale atomistic simulations [25, 90]. Here list some studies and applications of SNAPS in recent years. Jiang *et al.* utilized explicit multielement spectral neighbor analysis (EME-SNAP), which can represent multielement systems more explicitly by separating total density neighbor density into partial densities for each element [93], to model amorphous Zirconia-doped Tantala and has better performance than Morse-Beest-Kramer-Santen (Morse-BKS) potential [94, 95]. Sikorski *et al.* used SNAP to study the microstructure and mechanical properties of dispersion-strengthening W with zirconium carbide [96]. And Wang *et al.* calculated the generalized stacking fault energies curves and Peierls stress or critical resolved shear stress in four BCC refractory metals (Mo, Nb, Ta, W), and ideal shear strengths outperform EAM [91, 97, 98]. Besides, some other systems are also studied based on SNAPS, such as lithium nitride [99], Ni-Mo binary alloys [100], 2D MoS_{2(1-x)}Se_{2x} alloys [101], etc.

Coulomb Matrix

Coulomb matrix is a nonlinear regression machine learning model, which is based on a measure of distance only in terms of both nuclear charges and atomic positions [32]. By using the same molecular information of an electronic structure calculation, the Cartesian coordinates $\{\mathbf{R}_I\}$ and nuclear charges $\{Z_I\}$, the atomization energies are

modeled. The Coulomb matrix consists of two parts, namely atomic energies and internuclear Coulomb repulsion operator,

$$M_{IJ} = \begin{cases} 0.5Z_I^{2.4} & \text{for } I = J \\ \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} & \text{for } I \neq J \end{cases} \quad (58)$$

where the diagonal elements represent polynomial fit of atomic energies to nuclear charge, and the off-diagonal elements is the Coulomb repulsion between atoms I and J . To map molecular characteristics onto atomization energies, a measurement of molecular similarity that is invariant with respect to translations, permutations, and rotations is required. Therefore, a distance between two molecules by the Euclidean norm of their diagonalized Coulomb matrices is measured.

$$d(\mathbf{M}, \mathbf{M}') = d(\boldsymbol{\epsilon}, \boldsymbol{\epsilon}') = \sqrt{\sum_I |\epsilon_I - \epsilon'_I|^2} \quad (59)$$

Where $\boldsymbol{\epsilon}$ are the eigenvalues of \mathbf{M} in order of decreasing absolute value. $\boldsymbol{\epsilon}'$ is extended by zeros for smaller system. So, the energy of a molecule \mathbf{M} is a summation of weighted Gaussians,

$$E(\mathbf{M}) = \sum_{i=1}^N \alpha_i \exp\left[-\frac{1}{2\sigma^2} d(\mathbf{M}, \mathbf{M}_i)^2\right] \quad (60)$$

where i is the index of training molecules, regression coefficients $\{\alpha_i\}$ and σ are determined during training, and $\{\alpha_i\}$ are determined by kernel ridge regression [102].

For given length scale σ and regularization parameter λ , the solution is

$$\min_{\alpha} \sum_i (E(\mathbf{M}_i) - E_i^{ref})^2 + \lambda \sum_i \alpha_i^2, \quad (61)$$

where $\{\alpha_i\}$ is given by

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{E}^{ref}, \quad (62)$$

and

$$K_{ij} = \exp\left[-\frac{d(\mathbf{M}_i, \mathbf{M}_j)^2}{2\sigma^2}\right] \quad (63)$$

is the kernel matrix of all training molecules and \mathbf{I} represent the identity matrix.

This method can take into account the stoichiometric and configurational variation and encode them uniquely; besides, the distance is continuous for small variations, and the ordered eigenvalues of the Coulomb matrix are invariant with respect to translation, permutation, and rotation. In Ref. [32], by using over 7000 small organic molecules to test, the accuracy of this machine learning model has better performance than that of semiempirical quantum chemistry method PM6 [103], and this machine learning model is more efficient. The transferability is also tested by using 1000 molecules to predict the remaining 6000 molecules, and the errors is similar to that of training mean absolute error (MAE), which means that this model has good performance for predicting unseen organic molecules and distorted equilibrium states.

A major drawback of the original Coulomb matrix is the lack of transferability, e.g., prediction of bond dissociation energy for large molecules by using trained model with small molecules. To overcome this problem, the locality similarity, in which an atom environment is constrained by using cutoff radius is applied to compare the similarities between new system and trained one [104]. In Ref. [104], the modified model has been demonstrated for prediction of quantum mechanical properties. For molecules, though the locality assumption is not necessarily true for long-range effects such as conjugated π -bond systems.

However, the Coulomb matrix itself is not permutation invariant, to address this problem, several other descriptors derived from original Coulomb matrix have been reported, such as Hansen *et al.* used a set of randomly sorted Coulomb matrix to apply invariance of permutation and richer atomic information [105]. Hansen and coworkers also proposed a more sophisticated approach called Bag of Bonds (BoB) in which

each bag contains a particular bond type's entries calculated by Coulomb matrix, then vectorize the bags by concatenating them in a specified order and sort the entries in each bag based on their magnitude [106]. Barker *et al.* modified the Coulomb matrix and sorted Coulomb matrix to build a localized coulomb representation which can be used in crystals and molecules and combine GAP to obtain a new potential called LC-GAP [107]. And Collins *et al.* used the information of the number of atom and bond for different elements, coordinate number, and bond order to define the representation, named Encoded Bonds, whose length is independent of the size of the molecule and only depends on the types of atoms in the molecule [108]. Besides, for other studies, Schrier studied the interpretability and the ability of distinguish different isomers and conformers of the Coulomb matrix representation [109]. And Xu *et al.* combined Coulomb matrix, symmetry functions, unsupervised learning and deep learning to create potentials with little human intervention and minimal redundant reference [110].

Moment Tensor Potential

Due to the completeness of the neural network descriptors is undetermined and the expanding functions of spherical harmonics basis is computationally expensive, the moment tensor potential (MTP) is proposed by Shapeev to mitigate these dilemma [26]. The main idea of the MTP is that the energy can be expressed by a linear combination of basis polynomials, similar to other interatomic potentials, the local potential energy of MTP is represented by

$$V(u) = \sum_{\alpha \in A} c_\alpha B_\alpha(u), \quad (64)$$

where A is a set of defined matrix-valued indices and c_α is the set of coefficients fitted during training process. The u is the collection of the relative coordinates $u = (u_i)_{i=1}^n$, and the $B_\alpha(u)$ is a basis function for a given α ,

$$B_\alpha(u) = \sum_{\beta} \prod_{i=1}^k M_{\alpha_{ii}, \alpha'_i}(u) \quad (65)$$

which is contraction operator of tensors $M_{\alpha_{ii}, \alpha'_i}(u)$, β is collection of multi-indices, $B_\alpha(u)$ is a k -body function, and the linear combinations of B_α are invariance for permutation and rotation. The α'_i is the sum of the off-diagonal elements of the i th row,

$$\alpha'_i = \sum_{\substack{j=1 \\ j \neq i}}^k \alpha_{i,j} \quad (66)$$

The polynomials M is the building block for the descriptor

$$M_{\mu,\nu}(u) = \sum_{i=1}^n f_{\mu,\nu}(|u_i|) u_i^{\otimes \nu}, \quad (67)$$

here, $f_{\mu,\nu}$ is the radial basis function, and it plays the same role as the "radial symmetry functions" in ACSF, which contains cutoff function such that $f_{\text{cut}}(r) = 0$ for $r > R_{\text{cut}}$. Such as the Chebyshev polynomials can be used as the radial basis function [111]. This function makes sure the smoothness with respect to the leaving and entering of the atoms in a certain cutoff. The $u_i^{\otimes \nu} := u_i \otimes \cdots \otimes u_i$ is the Kronecker product of ν copies of the vector u_i . $u_i^{\otimes \nu}$ are tensor of rank ν and it contains angular information of the neighborhood. Therefore, the $M_{\mu,\nu}(u)$ has mechanical interpretation, e.g., the expression $M_{0,0}$ denotes the count of atoms within the distance of R_{cut} , akin to the "mass" of these atoms. Similarly, $M_{0,1}$ represents the center of mass of these atoms, while $M_{0,2}$ corresponds to the tensor of second moments of inertia, and so forth. When considering $\mu > 0$, $M_{\mu,\nu}$ can be explained by weighted moments of inertia. Here, the weight assigned to the i th atom is $f_{\mu,\nu}(|u_i|)$.

To determine the coefficients c_α for energies and forces potentials, a regularized linear regression is applied to avoid overfitting and the optimized coefficients c_α are obtained by minimizing the following loss functions [112],

$$\min_c \left\| \sum_{\alpha \in A} c_\alpha B_\alpha(u) - E_{ref} \right\|^2 + \gamma \|c\|_{l_p}^2 \quad (68)$$

$$\min_c \left\| -\frac{\partial}{\partial u} \sum_{\alpha \in A} c_\alpha B_\alpha(u) - F_{ref} \right\|^2 + \gamma \|c\|_{l_p}^2 \quad (69)$$

where $\|c\|_{l_p}$ is the number of nonzero entries in c .

In Ref. [26], by comparing the computation (CPU) time and root mean square (RMS) error to GAP, it can be seen that under same accuracy, the MTP uses fewer parameters, and the computation is more than two orders of magnitude faster than that of GAP. For the same number of parameters, the accuracy and computation time are 1.5 times better and 40 times smaller than that of GAP, respectively. In recent years, lots of research based on MTP and active learning algorithm has been conducted to strength the accuracy, for example, Shapeev *et al.* calculated elastic properties of bcc-Ti from 900 K to 1700 K with the same accuracy as ab initio molecular dynamics (AIMD) simulations but more efficient than AIMD [113], Shapeev *et al.* reproduced the extended X-ray absorption fine structure (EXAFS) spectroscopy for bcc-W, bcc-Mo, fcc-Cu, and fcc-Ni with ab initio accuracy [114], Kwon and coworkers studied the diffusivity of dilute hydrogen in bcc metals (Nb, Fe, and W) with ab initio accuracy [115], Bock *et al.* predict thermal expansion, elastic constants, and vibrational properties of prototypical hard coating alloy Ti_{0.5}Al_{0.5}N [116], and Chen *et al.* built a uranium potential to accurately reproduce properties such as lattice parameters, cohesive energy, elastic, vibrational, etc. It also can predict physical properties of different uranium phases [117]. In addition, there are many studies listed in Reference not discussed here [111, 118-123].

Graph Neural Networks

Graph neural networks (GNNs) received lots of attention in recent years because they can utilize both molecules and crystals structures and have excellent performance.

Unlike other hand-crafted representations, GNNs solely based on the atom types and coordinates, make it less sensitive to the choice of atomic descriptors [124, 125]. The typical way for prediction of properties of materials based on graph networks can be treated as message-passing process [27], therefore graph networks also refer message passing neural networks (MPNNs), in which the information of each atom is transferred to its neighbors at each iteration [126]. Here, take generalized crystal graph convolutional neural networks (CGCNN) framework as an example to show how the graph network works [127]. The CGCNN framework is illustrated in Fig. 6, the central idea for CGCNN is to use a graph to represent both atomic information and bonding interactions, and the graph is fed into a convolutional neural network to iteratively update the atom feature with surrounding atoms and bonds,

$$v_i^{t+1} = v_i^t + \sum_{j,k} \sigma[(v_i^t \oplus v_j^t \oplus u_{(i,j)_k}^t) W_f^t + b_f^t] \odot g[(v_i^t \oplus v_j^t \oplus u_{(i,j)_k}^t) W_s^t + b_s^t] \quad (70)$$

where \oplus concatenate atomic feature vectors which contain the properties of atom i and j , denoted by v_i and v_j , and bond features $u_{(i,j)_k}$ which represents k th bond connecting atom i and j . W and b are convolutional weight matrix and bias of the t th layer which consider the interaction strength between neighbors. The \odot denotes element-wise multiplication to sum neighbor atoms vector up. σ and g are sigmoid function and softplus function, respectively, which introduce nonlinear coupling between layers. The convolutional layers can make the atom i learn the features of its surrounding environment, the purpose of these layers is similar to that used in convolutional neural networks and therefore, these layers are called convolutional layers [127-129]. After convolution, the vectors are fed into a pooling layer to produce a feature vector and followed by two fully connected layers to get final output.

Similar to the HDNNPs, the training is conducted by using backpropagation and gradient-based optimization algorithm to minimize the loss function L , which indicates the difference between the prediction of CGCNN function f and reference y .

$$\min_{\mathbf{W}} \sum_i L(f_i(\mathbf{W}), y_i) \quad (71)$$

Where i is the different configurations, \mathbf{W} represents the weights of neural networks in CGCNN, and f_i is the prediction property of configuration i .

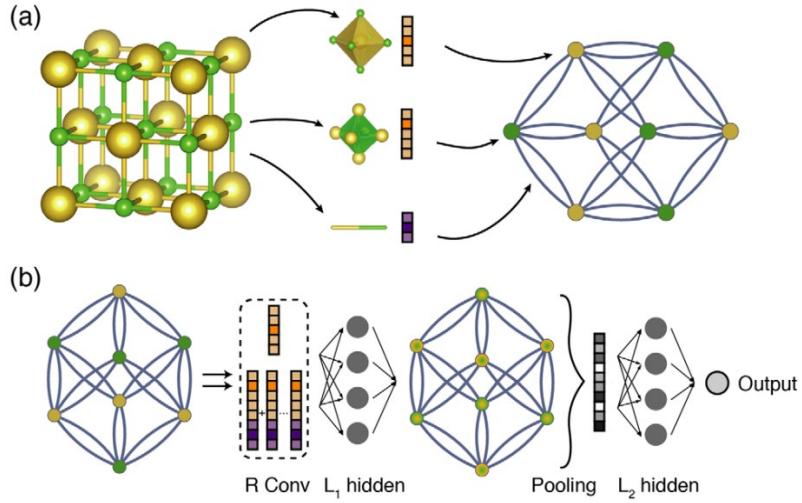


Fig. 6. The illustration of crystal graph convolutional neural networks (CCGN) involves two main components: (a) Construction of the crystal graph: Crystals are transformed into graphs where nodes represent atoms within the unit cell, and edges represent connections between atoms. Each node and edge are characterized by vectors corresponding to the atoms and bonds in the crystal, respectively. (b) Structure of the convolutional neural network on the crystal graph: The convolutional neural network is designed on top of the crystal graph. R convolutional layers and L_1 hidden layers are implemented for each node, resulting in a new graph where each node now represents the local environment of an atom. After pooling, a vector representing the entire crystal is connected to L_2 hidden layers, followed by the output layer, ultimately

providing predictions. Reprinted figure with permission from Tian and Grossman, Phys. Rev. Lett. **120**(14), 145301 (2018) [127]. Copyright 2024 by the American Physical Society.

One feature of the graph networks is that it can take into account the long-range interactions between atoms without using large cutoff radius to define neighbors [130]. During convolution process, each atom will receive its neighbors' information iteratively. For instance, the feature vector of each atom will get information of its nearest neighbors after first iteration, and each atom will obtain both neighbors and neighbors' neighbors' information in the second iteration. So, after n iterations, the feature vector of each atom will be updated by its n th-nearest neighbors. This feature is not likely to that of descriptors in which each atom only interacts with other atoms within a certain cutoff radius. What's more, the feature vectors can be easily used in multi-component systems. The MPNNs are first applied in molecular atomic [126, 128, 129, 131], and then it is also used in crystals or both [124, 127, 132, 133]. Besides, some research did further investigations, such as interpretability and effects of multi-fidelity datasets [134, 135], improvements of performance by incorporating attention mechanism or considering Voronoi neighbors, three-body correlation, and changeable edge vectors [136, 137]. However, most MPNNs only depend on pairwise distances, which lack directional information like bond angles and rotations. Therefore, equivariant MPNNs are developed to meet rotational invariance under group action and can obtain higher performance than local descriptor-based models. Many equivariant MPNNs are proposed, such as NewtonNet [27], Cormorant [138], Tensor Field Networks [139], PaiNN [140], NequIP [141], EGNN [142], Equivariant Transformers [143], and SEGNN [144]. Besides, some MPNNs models combined or

mimic physics-based classical interatomic potentials to improve the performance, for instance, ReaxFF-MPNN combined bond order of ReaxFF and MPNN to get better performance [145], and TeaNet mimic EAM potential and iterative electronic relaxation and had good performance for the first 18 elements on the periodic table [146]. Recently, in order to reduce the high time cost caused by 2-body message passing and multiple iterations, an efficient and accurate MACE architecture is introduced combining equivariant message passing and many-body messages [147]. It is impossible to list all models here, Ref. [148] and Ref. [126] gave some reviews for applications of MPNNs in molecules and chemistry.

Symbolic Regression

Symbolic regression is a hybrid method in which genetic programming is used to develop fast, accurate, and transferable many-body interatomic potentials [149]. Different from genetic algorithms which have been used to optimize parameters of interatomic potentials [150-155], genetic programming can optimize both parameters and functions. Genetic programming has been used to rediscover fundamental physical laws and explore descriptors of complex material properties such as the reconstruction of the Lennard-Jones potential and simple 3-body Stillinger-Weber potential [156-161], by optimizing simple expressions of the PES according to Darwin's natural selection rules like crossover and mutation operations from the hypothesis space of candidate functions consists of variables and mathematical operators, such as real numbers, addition, subtraction, multiplication, division, exponentiation, and summation [162]. In hypothesis space, each function is represented as a tree graph, Fig. 7 shows three functions illustrated by tree graphs and shows an example of crossover operation, in which a branch of one tree is selected

randomly and replaced with a randomly selected branch from another tree. Hernandez *et al.* also demonstrated that genetic programming could find metallic many-body potentials [149], in briefly, by following natural selection rules, genetic programming finds the candidate functions that can best fit the training data by utilizing covariance matrix adaptation evolution strategy (CMA-ES) and conjugate gradient (CG) optimizers [163, 164]. In Ref. [149], copper potentials are generated from symbolic regression, Eq. 72 shows one of them,

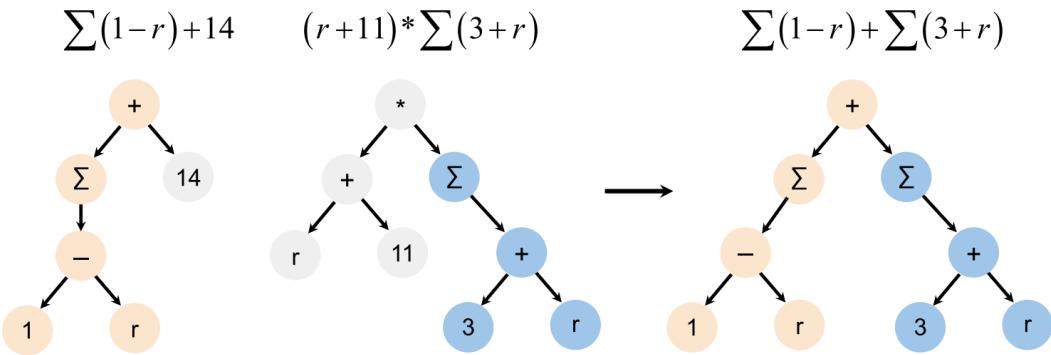


Fig. 7. Example of a crossover operation, each node that is not a leaf node represents an operation performed on its child nodes. Reproduced under the terms of the Creative Commons CC BY license [149]. Copyright 2019, The Authors, published by Springer Nature.

$$\begin{aligned}
 E_i = & 7.51 \sum_j r_{ij}^{3.98-3.93r_{ij}} f(r_{ij}) \\
 & + \left(28.01 - 0.03 \sum_j r_{ij}^{11.73-2.93r_{ij}} f(r_{ij}) \right) \left(\sum_j f(r_{ij}) \right)^{-1}
 \end{aligned} \tag{72}$$

where E_i is the local energy of atom i , r_{ij} is the distance between the atom i and its neighbor atoms j , and $f(r_{ij})$ is a smoothing function that goes to zero at the cutoff radius. This potential outperforms the known EAM-type potentials for predicting the

bcc lattice parameter, hcp–fcc formation energy difference, vacancy formation energy, dumbbell defect formation energy, and phonon frequencies [165-168]. Due to the similarity between this function and EAM model with repulsive and many-body attractive terms and the accuracy of physical properties indicates that the functions generated by this method may contain useful physical explanation. Besides, the simple expressions of these functions are less likely to overfit, both can lead to high transferability. And the symbolic regression is about 1 – 4 orders of magnitude faster than other machine-learned interatomic potentials. But symbolic regression also has some limitations. For example, the expressions in the hypothesis space must be simple and have a significant effect on potential energy, and this model cannot learn complex terms that involve bond angles [130, 149].

Physically Informed Machine Learning Potentials

For general machine learning potentials, transferability, which describes the ability of a model to correctly predict the property of an atomic configuration lying outside the training dataset, is limited. Consequently, physically informed neural networks (PINN) are proposed to improve the transferability of unknown structures [169] by combining a general physics-based interatomic potential with a neural-network regression. PINN achieve this by optimizing a set of physical-meaning parameters of a physics-based interatomic potential from local environment of atom i , and then feeding them back to improve the accuracy of the original physics-based interatomic potential. The flowchart of different types of interatomic potentials is shown in Fig. 8. A traditional potential is shown in Fig. 8a, the PES is calculated by summation of atomic energies, each atomic energy E_i for atom i is computed by n neighbors' atomic coordinates ($\mathbf{r}_{i1}, \dots, \mathbf{r}_{in}$) and potential functions with fitted

parameters $\mathbf{p} = (p_1, \dots, p_m)$ which are optimized by small size of dataset of experimental data and DFT calculations. Due to the intrinsic basic physics in the functions, the traditional potentials have good transferability. The mathematical neural network potentials (Fig. 8b), such as neural network potentials based on BP descriptors [20-22], have similar expression of total energy, except additional functions are used to map local environment of atom i into local descriptor $(G_i^1, G_i^2, \dots, G_i^k)$, then the neural network is trained by minimizing the error between the predicted total energy and the reference.

$$\varepsilon = \sum_s (E^s - E_{\text{DFT}}^s)^2 + \tau (\sum_{\epsilon\kappa} |w_{\epsilon\kappa}|^2 + \sum_{\kappa} |b_{\kappa}|^2) + \gamma \left(\sum_{\eta} |p_{\eta} - \bar{p}_{\eta}|^2 \right) \quad (73)$$

Where second term is regularization term to avoid overfitting, $w_{\epsilon\kappa}$ and b_{κ} are weights and biases of neural networks, respectively. The third term ensures the difference between the predicted parameters p_{η} and the average value \bar{p}_{η} of the database is small. τ and γ are parameters to control regularization and parameters separately.

For PINN, instead of predicting the total energy or other properties of materials, the neural network will predict the best set of physics-based potential parameters, such as analytical bond-order potential (BOP) [170, 171], depended on the similarity between trained atomic environment and current atomic configuration, as shown in Fig. 8c. And then use the physics-based interatomic potential with optimized parameters to calculate the properties of interest. The PINN potential has better transferability and interpolation than a regular neural network potential with the same number of parameters. Fig. 8d illustrates a modification of PINN in which the global parameters $\{\mathbf{b}\}$ and neural network are optimized at the same time as they are trained. And the global parameters are fixed after training to improve the efficiency of

calculation. However, Ref. [169] shows that this method is about 25% slower than conventional neural network potentials with the same number of parameters due to the time required for classical potential calculations like BOP. Similarly, Chen et al. also vectorized the energy terms in EAM and angle-dependent potential (ADP) potentials and added some physical constraints to train a neural network to optimize the parameters in the two potentials, thereby obtaining a machine learning EAM/ADP with similar accuracy to SNAP. Since the tensor form can be processed by graphic processing units (GPUs), the time required for training is significantly reduced [172].

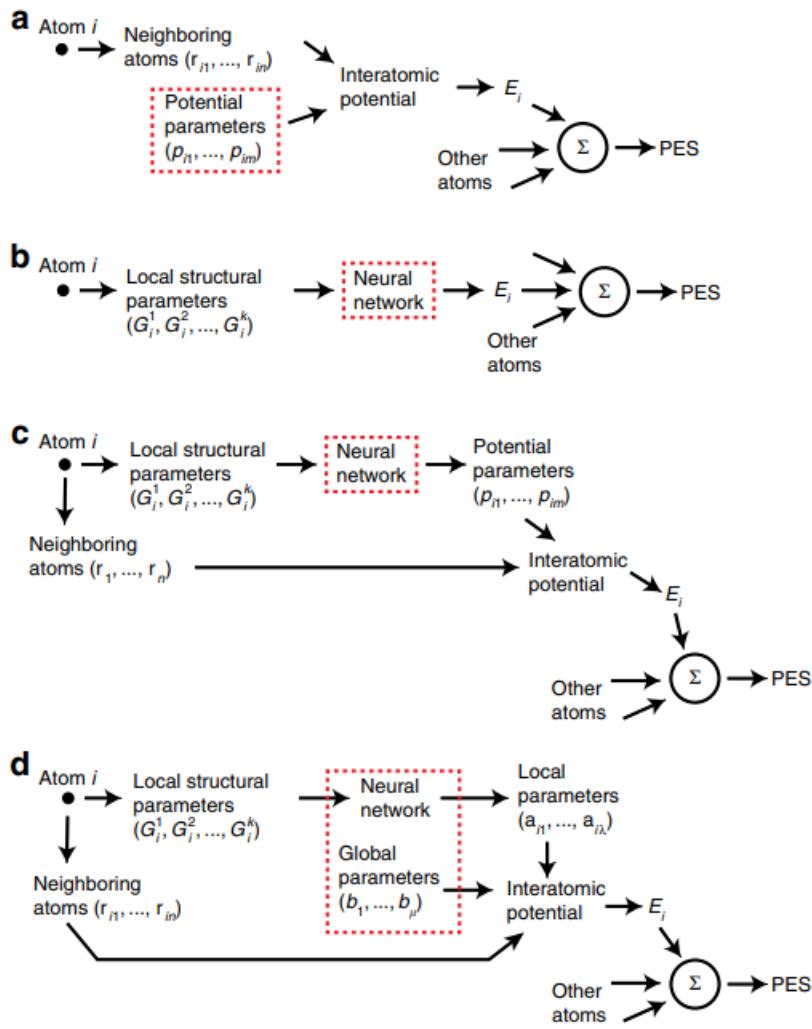


Fig. 8. Flowcharts of the development of atomistic potentials. (a) Traditional interatomic potential. (b) Mathematical neural network potential. (c) Physically informed neural network (PINN) potential with all-local parameters. (d) PINN potential with parameters divided into local and global. The dashed rectangle represents the objects requiring parameter optimization. Reproduced under the terms of the Creative Commons CC BY license [169]. Copyright 2019, The Authors, published by Springer Nature.

Atomic Cluster Expansion

Besides the descriptors mentioned above, Drautz proposed a general and complete descriptor extended from cluster expansion, called atomic cluster expansion [28]. This descriptor is based on the many-body expansion of potential energy, as shown in Eq. 74,

$$\begin{aligned}
 E_i(\boldsymbol{\sigma}) = & \sum_j \sum_v c_v^{(1)} \phi_v(\mathbf{r}_{ji}) \\
 & + \frac{1}{2} \sum_{j_1, j_2} \sum_{v_1, v_2} c_{v_1 v_2}^{(2)} \phi_{v_1}(\mathbf{r}_{j_1 i}) \phi_{v_2}(\mathbf{r}_{j_2 i}) \\
 & + \frac{1}{3!} \sum_{j_1, j_2, j_3} \sum_{v_1, v_2, v_3} c_{v_1 v_2 v_3}^{(3)} \phi_{v_1}(\mathbf{r}_{j_1 i}) \phi_{v_2}(\mathbf{r}_{j_2 i}) \phi_{v_3}(\mathbf{r}_{j_3 i}) \\
 & + \dots
 \end{aligned} \tag{74}$$

where $\boldsymbol{\sigma} = (\mathbf{r}_{1i}, \mathbf{r}_{2i}, \dots, \mathbf{r}_{Ni})$ represents the collection of $N - 1$ vectors $\mathbf{r}_{ji} = \mathbf{r}_j - \mathbf{r}_i$ from atom i to all other atoms. $c_v^{(K)}$ are expansion coefficients, and $\phi_{v_K}(\mathbf{r}_{j_K i})$ are a set of single-bond basis functions which can also be used in multicomponent systems, as shown in Eq. 75.

$$\phi_v(\mathbf{r}) = \sqrt{4\pi} R_{nl}(r) Y_l^m(\hat{\mathbf{r}}) \tag{75}$$

Where $v = (nlm)$, the radial functions R_{nl} depend on distance $r = |\mathbf{r}|$, and they can be expanded by expansion coefficients c_{nlk} and Chebyshev polynomials of the first kind T_k with scaled distance x that make smaller distance r denser.

$$R_{00}(r) = g_0 = 1 \tag{76}$$

$$R_{0l}(r) = 0 \quad \text{for } l > 0 \tag{77}$$

$$R_{nl}(r) = \sum_{k=1} c_{nlk} g_k(r) \quad \text{for } n > 0 \tag{78}$$

$$g_0 = 1 \tag{79}$$

$$g_1(r) = \left[1 + \cos\left(\frac{\pi r}{r_c}\right) \right] \tag{80}$$

$$g_k(r) = \frac{1}{4} [1 - T_{k-1}(x)] \left[1 + \cos\left(\frac{\pi r}{r_c}\right) \right] \tag{81}$$

$$x = 1 - 2 \left(\frac{e^{-\lambda(\frac{r}{r_c}-1)} - 1}{e^{\lambda} - 1} \right) \quad (82)$$

When the distance of any two atoms is more than cutoff r_c , the direct interaction between them will vanish and the atom moves outside or inside the cutoff distance will change the energy smoothly. The radial functions R_{0l} do not depend on distance but bond angles which do not contribute to local expansion, so it set to zero. and the Y_l^m are functions of the bond direction $\hat{\mathbf{r}} = \mathbf{r}/r$.

Since the summation of the many-body expansion in Eq. 74 scales as order of K , which is computationally expensive. To overcome it, reordering of summation is applied to transform a sum over products into product over sums,

$$E_i(\boldsymbol{\sigma}) = \sum_v c_v^{(1)} A_{iv} + \sum_{v_1, v_2}^{v_1 \geq v_2} c_{v_1 v_2}^{(2)} A_{iv_1} A_{iv_2} + \sum_{v_1, v_2, v_3}^{v_1 \geq v_2 \geq v_3} c_{v_1 v_2 v_3}^{(3)} A_{iv_1} A_{iv_2} A_{iv_3} + \dots \quad (83)$$

where A_{iv} is the projection of the basis functions on atomic density:

$$A_{iv} = \langle \varrho_i | \phi_v \rangle = \sum_j \phi_v(\mathbf{r}_{ij}), \quad (84)$$

and ϱ_i is atomic density,

$$\varrho_i = \sum_j \delta(\mathbf{r} - \mathbf{r}_{ji}) \quad (85)$$

by separating single-bond basis in radial and angular functions and incorporating cutoff functions to obtain a complete expansion that is invariant with respect to translation, rotation, and inversion,

$$\begin{aligned} E_i(\boldsymbol{\sigma}) &= \sum_n c_n^{(1)} B_{in}^{(1)} + \sum'_{n_1, n_2, l} c_{n_1 n_2 l}^{(2)} B_{in_1 n_2 l}^{(2)} + \sum'_{\substack{n_1, n_2, n_3 \\ l_1, l_2, l_3}} c_{n_1, n_2, n_3}^{(3)} B_{in_1, n_2, n_3}^{(3)} + \dots \\ &= \sum_{KnL} c_{nl}^{(K)} B_{inl}^{(K)} \end{aligned} \quad (86)$$

the $B^{(K)}$ is the rotational invariance products of the A_{iv} , which contains the many-body interactions. These terms can be reduced by the Wigner $3j$ symbol and the Clebsch-Gordan coefficients [173-176]. Eq. 87-89 show the $B_{in}^{(1)}$, $B_{in_1 n_2 l}^{(2)}$, and $B_{l_1, l_2, l_3}^{(3)}$, respectively.

$$B_{in}^{(1)} = A_{in00} \quad (87)$$

$$B_{in_1 n_2 l}^{(2)} = \sum_{m=-l}^l (-1)^m A_{in_1 lm} A_{in_2 l-m} \quad (88)$$

$$B_{l_1, l_2, l_3}^{(3)} = \sum_{m_1=-l_1}^{l_1} \sum_{m_2=-l_2}^{l_2} \sum_{m_3=-l_3}^{l_3} \begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \times A_{in_1 l_1 m_1} A_{in_2 l_2 m_2} A_{in_3 l_3 m_3} \quad (89)$$

The loss function expressed in Eq. 90 is operated to make the error that is roughly proportional to the energy of the reference data.

$$L = \sum_n w_n (E_n - E_n^{ref})^2 + \alpha \sum_{Kn} |c_{nl}^{(K)}| + \beta \sum_{Kn} |c_{nl}^{(K)}|^2 \quad (90)$$

Where the E_n is the predicted energy, E_n^{ref} is the reference data, the α and β are parameters of regularization. The weights w_n allow the model to use heterogeneous references whose values cover a wide range, and w_n is

$$w_n = \frac{\omega}{[E_n^{ref} - E_{min}^{ref}(1+\Delta)]^2}, \quad (91)$$

the ω is a normalization factor which ensures $\sum_n \omega_n = 1$. The E_{min}^{ref} is the smallest energy per atom in reference dataset. The parameter Δ prevent the weights of the lowest energy structures from diverging.

The atomic cluster expansion is also can be expanded to other descriptors, such as Steinhardt parameters [177], Correction electron densities [29], Moments of the density of states [178], BP symmetry functions [20], bispectrum [71, 179], SOAP [71],

SNAP [25], and MTP [26] in which some are introduced previously. They are closely related to atomic cluster expansion method and can be expanded in atomic cluster expansion. Besides, by introducing the nonlinear atomic cluster expansion to expand the local density in the Finnis-Sinclair potential for bulk structure, the nonlinear atomic cluster expansion has comparable accuracy than that of HDNNPs [28]. ACE has widely used in different systems, such as carbon [180], iron [181], binary Pt-Rh nanoparticles [182], and general framework for elements, alloys, and molecules [183], due to the high accuracy and efficiency compared to several state-of-the-art machine learning potentials [184]. The application of the ACE can beyond these, for instance, combined with flexible charge model such as extended Lagrangian (XL) Born-Oppenheimer molecular dynamics (BOMD), the ACE can represent the atomic electronegativities and short-range charge-independent parts and avoid the costly charge relaxation without any significant loss of accuracy [185]. And combine activate learning to explore new structures and improve the accuracy and transferability of ACE model [186]. ACE can also be generalized to anti-symmetric functions such as wave function and substituted ACE for deep neural networks to develop variational Monte Carlo (VMC) method for computing the ground state of the many-body Schrödinger equation [187, 188]. Dusson *et al.* provided a detailed introduction to invariant polynomial basis based on spherical harmonics expansions and demonstrated the ACE contains a complete basis and is highly efficient. Although in convergence tests, it is significantly slower than classical interatomic potentials, it is competitive with leading machine learning potentials with similar levels of parameterizations. It provides a reference for numerical analysis of symmetric

polynomials in fitting interatomic potentials or more generally molecular systems [189].

Other Machine Learning Potentials

The description of machine learning potential above is very limited, and there are many more machine learning potentials based on similar or different descriptors or regression methods in addition to the methods described above, which include but are not limited to many-body tensor representation (MBTR) [190], power-type structural descriptors (PTSD) [191], proper orthogonal descriptors (PODs) [192], DeepMD method [193], and smoothed atomic densities [194]. In general, most of them are related to the typical methods mentioned above, like inspired by Coulomb matrix, BoB, and many-body expansion, MBTR used multiple many-body information such as atom counts, distances, angles, and dihedral angles and combined with element correlation matrices to describe both molecules and crystal, and PTSD used descriptors which contain ACSF-like functions and message passing interface (MPI) to achieve linear scaling of computational cost. There are many reviews of machine learning potentials and comparisons between different methods published in recent years which can provide some overviews for readers [2, 18, 53-55, 88, 89, 105, 126, 130, 148, 184, 195-198].

Machine Learning Potential Software

Up to date, a lot of machine learning potentials are released as open-source codes, and some are available in software library such as the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [199], the Vienna Ab Initio Simulation Package (VASP) [200, 201], TensorFlow [202], PyTorch [203], or others to provide

an easy way to get started [58, 204-208]. Since Ref. [53] and Ref. [2] have mentioned some packages, a short, incomplete addition is listed here.

PACE (Performant Implementation of the Atomic Cluster Expansion) [184] implements the ACE in LAMMPS. GAP is implemented within the Quantum and Interatomic Potential (QUIP) packages [89], which has generic C and LAMMPS-specific C++ interfaces, that allows GAP models to be used in external simulation packages. ACEpotentials.jl [209] provides linear ACE models and can be used for MD simulation in LAMMPS, Atomic Simulation Environment (ASE) [210], and Molly.jl [211]. MLIP-3 (Machine-Learning Interatomic Potentials-3) [118] constructed MTP potential with active learning and can be used in LAMMPS. TorchANI [62], a neural network potential implemented in PyTorch, is a lightweight and user-friendly package due to its flexibility and ease of use.

Summary

In summary, with the improvement of computing power and the generation of DFT-based databases, machine learning methods can use more resources to train models and make more extensive predictions beyond energy and force. In addition, with the improvement and development of descriptors, more generalized, efficient and physically meaningful descriptors will make training more efficient and transferability better. Therefore, these machine learning-based potential constructions have similar accuracy to DFT and can be simulated at longer times and scales than DFT. This is undoubtedly exciting because using machine learning to build interatomic potential models has the potential to change the focus of material prediction and research, freeing it from the time-consuming and laborious construction of specific interatomic potentials and instead simulating and studying material properties on a large scale by

training machine learning potentials with universal descriptors, shortening the research cycle for new materials and material simulations in the field of materials science.

CHAPTER 3

COMBINING REINFORCEMENT LEARNING WITH GRAPH

CONVOLUTIONAL NEURAL NETWORKS FOR EFFICIENT DESIGN OF

TIAL/TIALN ATOMIC-SCALE INTERFACES

Introduction

Metal/ceramic coatings play a pivotal role across various fields, including wear-resistant materials, electronic devices, gas-turbine engines, and medical implants, by improving the structural and electrical properties of substrate materials [8-10, 212, 213]. The metal components that possess high toughness are combined with ceramic components [13], which exhibit high hardness, abrasion resistance, chemical stability, and wear resistance [11, 12], to reduce the trade-off between ductility and hardness.

Transition metal nitrides or carbide-based metal/ceramic coatings are commonly studied due to their technical feasibility [214]. Among these coatings, Ti/TiN coatings are extensively investigated in different engineering fields such as structural applications, biomedical implants, fuel cells, aircraft, and gas turbine engines owing to high hardness and toughness, particle erosion resistance, corrosion resistance, and tribocorrosion resistance [215-224]. Many studies have highlighted that by employing nano-scale multilayers, Ti/TiN multilayer coatings have higher toughness and high adhesive strength [225], higher erosion and corrosion resistance [219, 220, 222-224, 226]. Moreover, it is well known that Al can be doped into conventional binary transition metal nitride, such as TiN, to improve the properties and impact the adhesion of coating [227-231]. For example, compared to Ti/TiN coatings, TiAl/TiAlN coatings have higher hardness and elastic modulus [232], TiAl/TiAlN multilayer coatings also have good corrosion resistance and high adherence strength

[233], and nanolayered multilayered TiAl/TiAlN coatings have high solid particle erosion and corrosion resistance [216]. Consequently, understanding the impact of dopants like Al on the mechanical properties of nanoscale Ti/TiN coatings is imperative.

The quality of the interface significantly impacts mechanical properties in nanoscale multilayer coatings [12, 234]. A comprehensive understanding of the metal/ceramic interfaces is therefore necessary to design coatings with the desired strength [235]. However, shedding light on the complicated details of metal/TiN interfaces remains a formidable challenge [229, 236, 237]. To address this challenge, first-principles density functional theory (DFT) and molecular dynamics (MD) simulations emerge as powerful tools for unraveling the atomic-level behavior of metal/ceramic interfaces [238-241]. For example, the impact of Al doping on the shear strength of the Ti/TiN interface has recently been studied by DFT and MD simulations [236, 242], which show that the introduction of Al significantly increases the shear strength of the Ti/TiN interface. In these studies, a Monte Carlo (MC) approach is used to find low-energy doped configurations. However, this approach is time-consuming because it requires DFT calculations to relax each generated potential configuration, and both works studied interfaces with low Al compositions of up to 25 mol%. Since the introduction of Al to TiN has resulted in higher hardness up to 60 mol% [232, 243, 244], and the number of potential configurations increases significantly with the increase of doping atoms, it is necessary to find a new way to efficiently explore the shear strength of TiAl/TiAlN interface with higher Al compositions.

Machine learning (ML) method is an active area in material science because it has shown the potential to predict materials properties from their atomic structures or create atomic structures from desired properties [245, 246]. Among ML methods, graph-based neural networks have been successfully applied to materials property prediction and interatomic potential development due to their highly accurate prediction of DFT-calculated properties [124, 127, 132, 247]. In addition, as optimization methods, reinforcement learning (RL) and genetic algorithm (GA) are widely used in designing different materials. For example, RL has been used in two-dimensional (2D) stretchable material design, interatomic potential development, chemical reaction optimization, and molecule discovery [248-251], and GA has been applied to interatomic-potential parameters optimization, feature selection, training configuration generation, and experimental parameters optimization due to its efficiency and stability [150, 151, 252, 253]. Therefore, to efficiently obtain stable adhesive TiAl/TiAlN configurations and account for the periodic boundaries in crystals, we use crystal graph convolutional neural networks (CGCNN) as our interatomic potential to predict the work of adhesion (W_{ad}). We also combine CGCNN with different methods, such as Monte Carlo (MC), deep Q-network (DQN) [254], and genetic algorithms (GA), to identify candidates with high W_{ad} and compare their performance. In this paper, we mainly focus on 50 mol% Al dopants. Furthermore, by examining the features of different layers of CGCNN and DQN neural networks, we find that the high W_{ad} structures follow a specific pattern to dope Al atoms near the interface, and further inspection of electronic structure and bonding of TiAl/TiAlN structures with high W_{ad} and low W_{ad} , and Ti/TiN structures reveals that the atomic

size differences and electronegativity play important roles in the bonding strength at the interface.

Methodology

DFT Calculations

The Vienna *Ab initio* Simulation Package (VASP; version 5.4.4) [255-257] is used for the DFT calculations with the generalized gradient approximation (GGA) parameterized by Perdew, Burke, and Ernzerhof (PBE) [258]. The standard projector augmented wave (PAW) pseudopotentials are used for core electrons as implemented in VASP [201, 259], and plane wave basis sets with cutoff energies of 400 eV and 550 eV are utilized for non-ionic update calculations and ionic relaxation, respectively [200, 260]. A $12 \times 9 \times 1$ mesh centered at Gamma is used for the metal/ceramic systems with periodic boundary conditions [261].

TiAl/TiAlN Interface Structure and Work of Adhesion

Fig. 9 shows the geometry of the Al-doped Ti/TiN structure, which consists of a total of 56 atoms (20 for TiAl and 36 for TiAlN). This structure is based on the Ti/TiN configuration, consisting of metal (Ti) and ceramic (TiN) layers, with the preferred crystallographic orientation based on the edge-to-edge matching model [262, 263]. Here, the face-centered cubic/hexagonal closed-packed (FCC/HCP) interface orientation relationship is chosen to create a single Ti (0 0 0 1)/TiN (1 1 1) interface, aligned normally to the z direction, which has the lowest interatomic spacing misfit, so, as shown in Fig. 9, the $[\bar{1} \bar{1} 2 0]$ and $[1 \bar{1} 0]$ directions are parallel to the x direction, and the $[1 1 \bar{2}]$ and $[1 \bar{1} 0 0]$ directions are set to parallel to the y direction. The in-plane lattice constants for these two different phases extracted from the Materials Project [264] are 2.94 Å and 5.08 Å for Ti and 3.00 Å and 5.19 Å for TiN.

The mismatches in the interface are 2.0% and 2.2% in the x and y directions, respectively, indicating that the interface can be treated as a coherent interface where the Ti layer is stretched in both the x and y directions to match TiN before binding Ti and TiN layers together. To achieve strong adhesion to the Ti layer, N atoms in the TiN layer are connected to the Ti layer [12]. In this work, we focus on a 50 mol% composition of Al dopants in both layers, implying the exchange of 10 out of 20 Ti atoms in the Ti layer and 9 out of 18 Ti atoms in the TiN layer with Al atoms (see Fig. 9). A vacuum of 16-Å thickness is applied in the z direction to remove the interactions of the surface atoms between interface slab models.

When the interface is built, the W_{ad} defined as the reversible work needed to separate the interface into two free surfaces [237, 238], can be a qualitative indicator of the binding strength. A higher W_{ad} means stronger and energetically favorable binding strength [265, 266]. The W_{ad} of the interface can be given by:

$$W_{ad} = (E_{TiAl} + E_{TiAlN} - E_{TiAl/TiAlN})/A, \quad (92)$$

where E_{TiAl} and E_{TiAlN} are the relaxed energy of TiAl and TiAlN isolated systems in contact with vacuum, respectively. after full relaxation, respectively; $E_{TiAl/TiAlN}$ is the total energy of the relaxed interface system. A is the cross-sectional area of the interface slab model. When the area of the interface for all interface models is treated as a constant, an approximate W_{ad} , W^{app} (Eq. 93), which removes the divisor interface area is applied in the CGCNN model to represent the bond strength at the interface:

$$W^{app} = E_{TiAl} + E_{TiAlN} - E_{TiAl/TiAlN}. \quad (93)$$

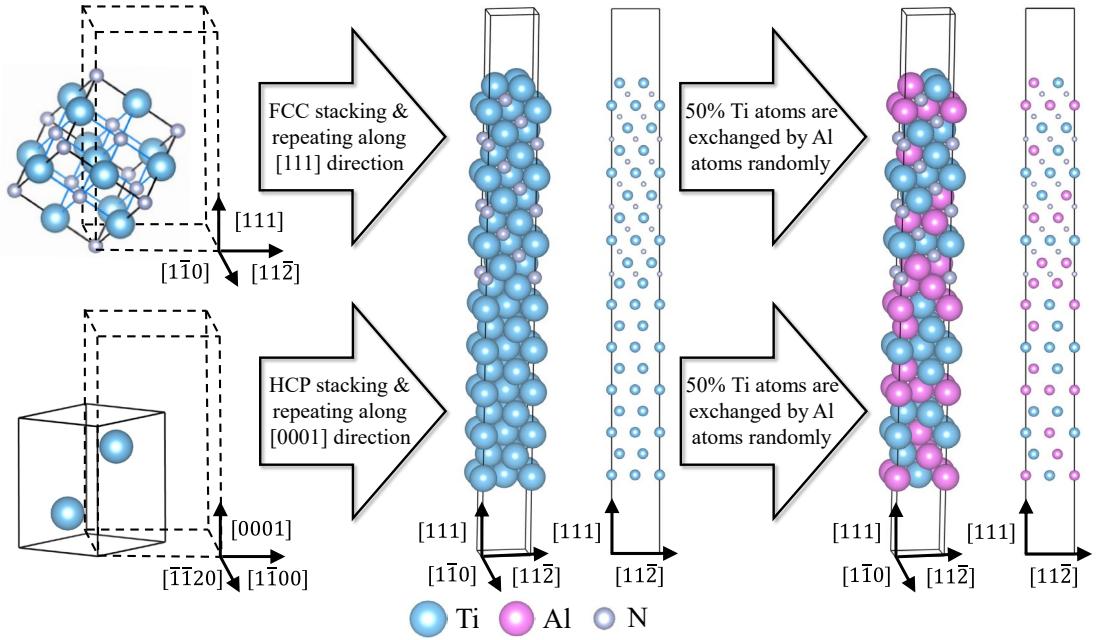


Fig. 9. Al-doped Ti/TiN interface structure with 50 mol% of Al generated from FCC/HCP Ti/TiN interface. Here, the directions of both Ti/TiN and TiAl/TiAlN interfaces are referenced by the TiN coordinate. Each system is shown by a perspective space-filling model and a 2D ball model.

Crystal Graph Convolutional Neural Networks (CGCNN) for Work of Adhesion

Prediction

For the 56-atoms Ti/TiN structure, alternating half of Ti atoms with Al atoms in both metal and ceramic parts results in $C(18, 9) \times C(20, 10) = 8,982,836,720$ possible TiAl/TiAlN combinations. Since the total elapsed time of DFT calculation of W_{ad} for each TiAl/TiAlN structure is more than one day with 64 computer cores, it is impossible to search such vast structure space to find potential structure by DFT relaxation of structures from random search. MC approach can expedite the exploration process, but it still needs time-consuming DFT calculation to relax each potential structure [236, 242]. To further enhance the efficiency, the CGCNN model instead of DFT calculation is utilized here. By randomly exchanging half Ti atoms by

Al atoms in each part of the Ti/TiN system to get TiAl/TiAlN systems, the W^{app} of generated TiAl/TiAlN structures is computed by using DFT no ionic update calculations and feeding these W^{app} and corresponding structures into CGCNN to train the model.

CGCNN is therefore leveraged as a dynamic model to predict each terminal state's W^{app} to simplify and reduce the time consumption of DFT calculations. Here, same as Eq. 70, CGCNN has 3 convolutional layers for updating the atom feature vector v_i with a nonlinear graph convolution function:

$$v_i^{t+1} = v_i^t + \sum_{j,k} \sigma[(v_i^t \oplus v_j^t \oplus u_{(i,j)_k}^t) W_f^t + b_f^t] \odot g[(v_i^t \oplus v_j^t \oplus u_{(i,j)_k}^t) W_s^t + b_s^t], \quad (94)$$

where \oplus concatenate atom, neighbor, and bond features. W and b are convolutional weight matrix and bias of the t^{th} layer which consider the interaction strength between neighbors. The \odot denotes element-wise multiplication to sum neighbor atoms vector up. σ and g are the sigmoid function and softplus function, respectively, which introduce nonlinear coupling between layers. The maximum number of considered neighbors is 12. After convolution, the vectors are fed into two fully connected layers of dimensions 64×128 and 128×1 to get the final output. Details of the method can be found in the work by Xie and Grossman [127].

Monte Carlo for TiAl/TiAlN Interface Generation

The MC procedure is similar to the Miraz et al method [242] with the workflow shown in Fig. 10. The 56-atoms Ti/TiN structure is randomly doped by Al to get an initial 50 mol% of Al-doped configuration, in order to compare with RL and GA methods, the W^{app} of each configuration is calculated by CGCNN or DFT no ionic update calculations, the details are shown in results and discussion section. A total of 55 steps ($N = 55$) is conducted, for each step, 10 trials ($T = 10$) are performed. In each

trial, two random Ti atoms in each TiAl and TiAlN part of the initial configuration are separately exchanged with two random Al atoms to create a new configuration, followed by W^{app} calculation by CGCNN or DFT no ionic update calculations. The highest W^{app} among trials and initial configuration is selected as the new initial configuration for the next step. This process is repeated until the maximum number of steps is reached. The final result is the configuration with the highest W^{app} obtained through the MC procedure.

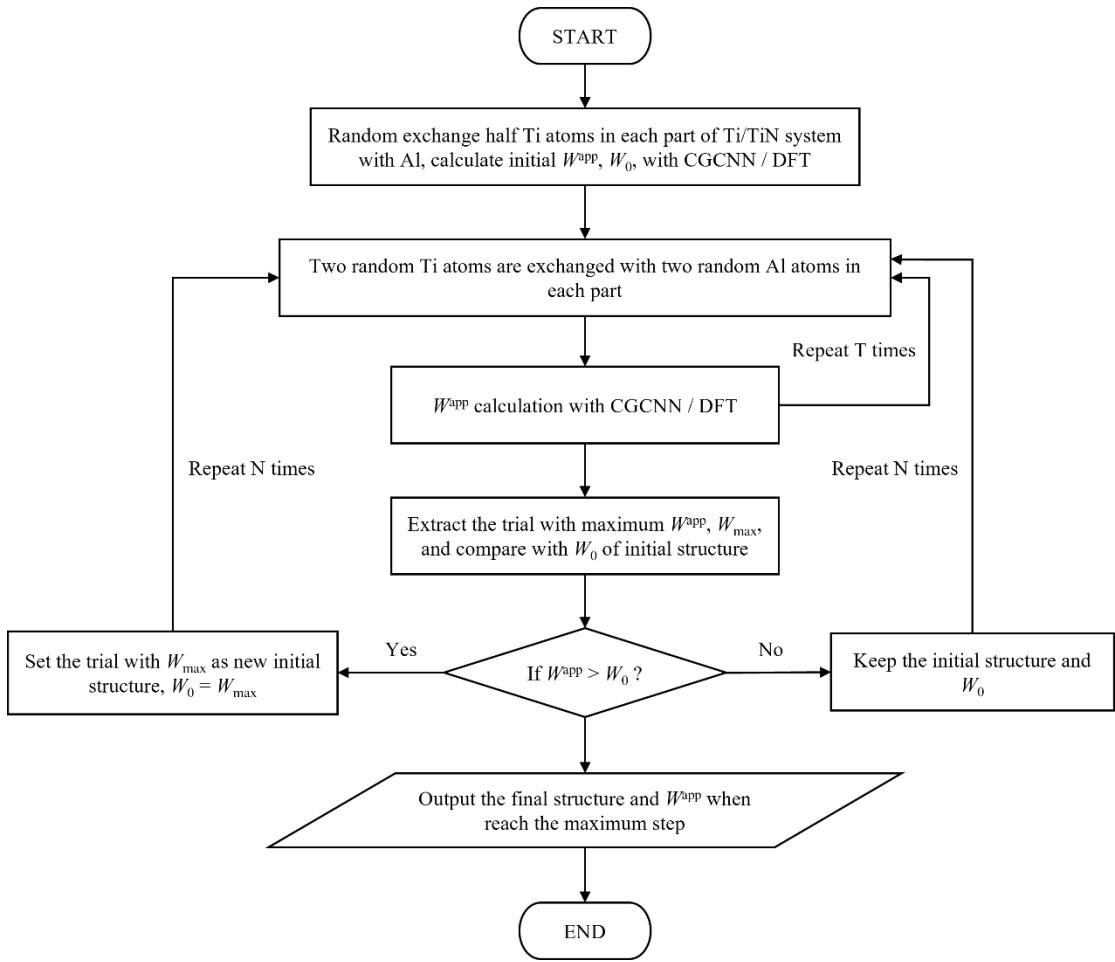


Fig. 10. Flow chart of the Monte Carlo (MC) procedure.

Genetic Algorithm (GA) for TiAl/TiAlN Interface Generation

The workflow of the GA process is outlined in Fig. 11, and its concept is inspired by the principles of natural selection [267]. In the beginning, a set of TiAl/TiAlN

structures with a size of 50 ($N = 50$) is generated randomly to form the initial population. Next, calculate the W^{app} of these structures by the CGCNN model, and elitism selection [268] is used to select the first 10 ($M = 10$) structures with the highest W^{app} among these structures, these 10 structures will directly pass over to the next generation. For the remaining offspring, the tournament selection method is applied to choose parents from the initial 50 structures. Specifically, 10 structures are selected randomly from the initial structures, and the one with the highest W^{app} is chosen as the parent. When two parents are selected, a one-point crossover is applied to generate a child, in short, there are two types of one-point crossover, the ceramic part of A combined with the metal part of B or the two are reversed, each type has 50% possibility of occurrence. After generating the child, there is a 50% probability of mutation, a swap mutation is utilized to exchange one Ti atom in each part of the child with one Al atom, as shown in Fig. 11, the dashed double arrows illustrate the atoms are exchanged. Repeat these processes 40 times ($N - M = 40$) and combine with the 10 parents from elitism selection to obtain the next generation. This new generation is treated as the new initial population and repeats the entire circle process until the number of steps reaches 9200 ($S = 9200$).

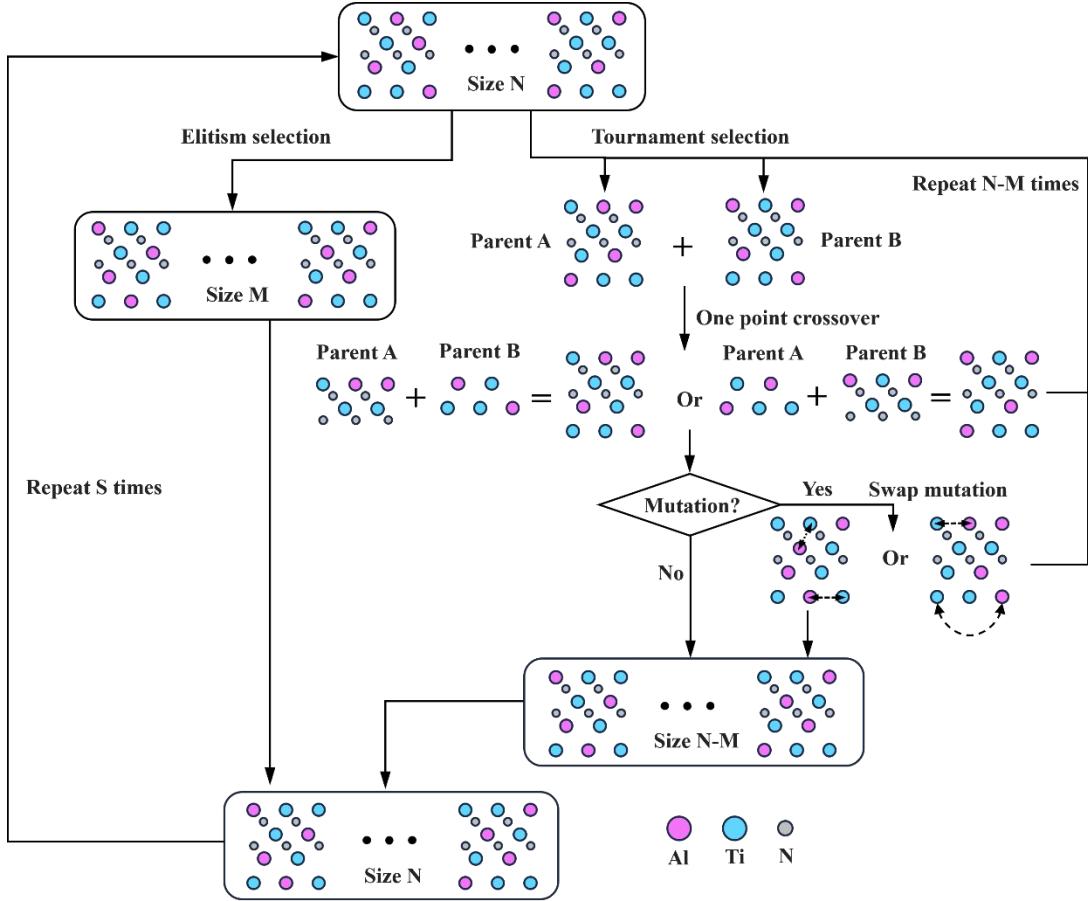


Fig. 11. Flow chart of the genetic algorithm (GA) procedure

Reinforcement Learning (DQN) for TiAl/TiAlN Interface Generation

Similar to the method used in the design of 2D stretchable kirigami materials [249], we use the RL model to find an optimal policy (action) that generates structures with high W^{app} . But we use CGCNN instead of convolutional neural network (CNN) as a dynamic model (environment) to predict each terminal state's W^{app} due to consideration of period boundary in crystal. Here, each structure has 28 atomic layers, as shown in Fig. 9, so we decompose each structure into 28 atomic layers, and each layer has two atoms and four actions, each action represents one combination of the atomic layer (Ti+Ti, Ti+Al, Al+Ti, and Al+Al atoms). The action of each layer will affect the subsequent action to make sure each part composition of Al is 50 mol%. Fig.

12 shows the schematic of our model, which contains an RL agent and a dynamic model of W^{app} as a policy evaluator. The RL agent formulates the construction of TiAl/TiAlN interface structures as a sequential decision-making process, in which the decision of the combination (action) for each atomic layer is made layer by layer. The decision process from the first atomic layer to the last atomic layer is an episode of RL in which each episode has 28 actions. During an episode at each time step t , the RL agent makes an action for the t^{th} layer from the 4 available actions, then in the next time step, the intermediate TiAl/TiAlN structure generated from the t^{th} time step will be the input state for the RL agent. At each time step, the dynamic model calculates W^{app} of each generated structure and returns the rewards to the RL agent.

The reward is defined below:

$$\text{reward}(t) = \begin{cases} 0 & t < T \text{ or } t = T \text{ and } W^{\text{app}} \leq 9.5 \\ W^{\text{app}} & t = T \text{ and } W^{\text{app}} > 9.5 \end{cases} \quad (95)$$

the reward is 0 if the structure is not in the terminal state or is in the terminal state but the final W^{app} is smaller than or equal to a certain threshold (set at 9.5 in this case). If the structure is in a terminal state and the final W^{app} is larger than a certain threshold, the reward equals W^{app} . The choice of the threshold is a critical hyperparameter that requires careful consideration. Larger thresholds can make it challenging for the RL agent to learn, while smaller thresholds may prolong the learning process. Therefore, selecting an appropriate threshold is essential for the effective training of the RL agent.

DQN is a convolutional neural network (CNN), whose input is a $2 \times 28 \times 3 \times 7$ tensor that collects the intermediate state (s_t) of the interface structure at and before time step t in one episode and outputs the state-action value $Q(s_t, a_t)$ for all the actions available in state s_t . The first dimension stores information about the atomic layers of

atoms being exchanged. The rest dimensions represent the TiAl/TiAlN structure, due to the atomic positions being arranged periodically, we convert the TiAl/TiAlN structure into tensor form, and each dimension represents the total number of positions in which the atoms can be, such as 28 is the number of z value (atomic layer) for an atom, 3 and 7 separately represent the number of x and y value for an atom, and different numbers represent different elements (e.g., 0 for none, 1 for N, 2 for Ti, and 3 for Al). The CNN network consists of two convolutional layers of dimensions $16 \times 28 \times 3 \times 7$, and $32 \times 27 \times 2 \times 6$ followed by three fully connected layers of dimensions 4160×1024 , 1024×512 , and 512×4 . $Q(s_t, a_t)$ is an estimate of the expected reward starting from time step t to the end of the episode by a given policy. As a function approximator, DQN estimates the Q-values with parameters θ , i.e., $Q(s_t, a_t, \theta) \approx Q(s_t, a_t)$. It can learn a policy (π) that maximizes the $Q(s_t, a_t, \theta)$, Eq. 96 shows the optimal Q function as the maximum return that can be obtained starting from state s_t , taking action a_t and following the optimal policy thereafter:

$$Q(s_t, a_t, \theta) = \max_{\pi} E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \dots \gamma^{T-t-1} r_N], \quad (96)$$

this function obeys the Bellman optimality equation:

$$Q(s_t, a_t, \theta) = E[r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta)], \quad (97)$$

where the γ is a discount factor which ensures the sum converges. In DQN, two neural networks, target net and policy net, with the same configurations are deployed to improve stability. Based on Bellman optimality equation (Eq. 97), the target net calculates the expected Q-value of the next state which is also called the temporal difference (TD) target (Eq. 101) with old parameters θ_{i-1} and compares it with the Q-value calculated by the policy net with updated parameters. The disparity between

these two Q-values is commonly referred to as the Temporal Difference (TD) error, a crucial metric used in the loss function to optimize both networks.

As continual learning, the DQN may abruptly lose the learned parameters from the previous task due to the weights and biases that are critical to the previous task being changed to fit the current task during learning. This phenomenon, called catastrophic forgetting [269, 270], is a particular challenge in artificial neural networks. To mitigate this, elastic weight consolidation (EWC) is utilized during the training of the RL agents. In short, EWC will constrain the learning of certain weights which are important for the previous task based on Fisher information matrix [271]. For example, imagine that there are two tasks A and B, and task A is learned first and then task B is continual learning process. The optimized parameters after training by task B can be written as,

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}_B(\theta) + \sum_i \frac{\alpha}{2} F_i (\theta_i - \theta_{A,i}^*)^2 \quad (98)$$

where the optimized parameters θ^* are calculated by the minimal loss of current task B and weights' importance of previous task A. α sets how important the old task is compared to the new task and i labels each parameter. F_i represents the i th diagonal component of Fisher information matrix F , which measures the importance of each parameter to task A, as shown in Eq. 99,

$$F_i = \frac{1}{N} \sum_{n=1}^N \nabla_{\theta} \log p(x_{A,n} | \theta_{A,i}^*) \nabla_{\theta} \log p(x_{A,n} | \theta_{A,i}^*)^T \quad (99)$$

since for most of the time the log-likelihood is intractable, here, an empirical Fisher information matrix is calculated by sampling the data from task A. Compare to L2 in which the weights are restricted by the same coefficient (Eq. 100), this method can separate contribution of each parameter and identify which parameters are more

important to task A and apply more constraint to important parameters during learning, thus, it does not hamper the learning process of task B.

$$\theta^* = \operatorname{argmin}_{\theta} \mathcal{L}_B(\theta) + \frac{\alpha}{2} (\theta - \theta_A^*)^2 \quad (100)$$

Here, Smooth L1 loss function combined with penalty which is summation of gradients of parameters is used to minimize the TD error, as shown in Eq. 102, where l_t is Smooth L1 loss function (Eq. 103), this loss function is less sensitive to outliers and prevents exploding gradients by introducing a value β , and α is scale factor of penalty P which is the EWC part. The details of performances by different loss functions are listed in Fig. 15a. During training, the experienced replay with reply memory of size 14500 is utilized for sample selection and expected Q-value calculations, the target net will be updated per 10 episodes. A decaying epsilon threshold (ϵ) for the ϵ -greedy strategy is used in order to relieve the exploration-exploitation dilemma [272], which means at each time step, the DQN will choose an action from the current state that maximizes $Q(s_t, a_t)$ with $1-\epsilon$ probability or a random action with ϵ probability.

$$Q(s_t, a_t, \theta_{i-1}) = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}, \theta_{i-1}) \quad (101)$$

$$\mathcal{L} = l_t + \alpha * P \quad (102)$$

$$l_t = \begin{cases} 0.5[Q(s_t, a_t, \theta_i) - Q(s_t, a_t, \theta_{i-1})]^2 / \beta, & \text{if } |Q(s_t, a_t, \theta_{i-1}) - Q(s_t, a_t, \theta_i)| < \beta \\ |Q(s_t, a_t, \theta_{i-1}) - Q(s_t, a_t, \theta_i)| - 0.5 * \beta, & \text{otherwise} \end{cases} \quad (103)$$

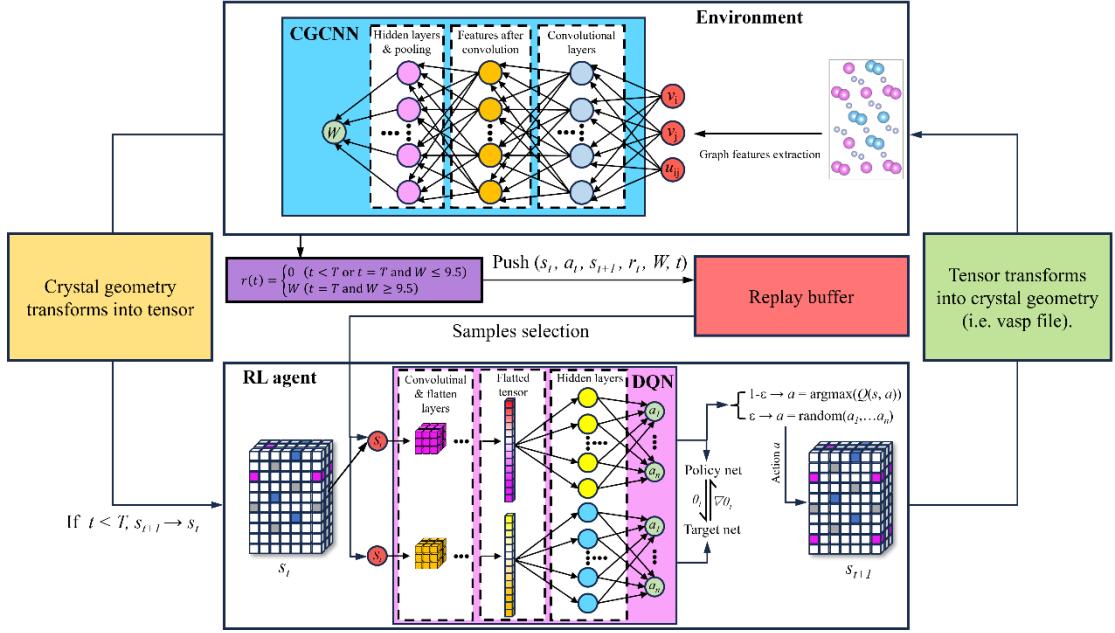


Fig. 12. Schematic of reinforcement learning (RL) model, which mainly consists of RL agent, environment, and replay buffer. Replay buffer stores generated structures information (state, action, next state, reward, work of adhesion, and step), RL agent selects samples from replay buffer to train and update the deep Q-network (DQN) model, and selects action to generate new structures s_{t+1} based on current state s_t . The environment that contains crystal graph convolutional neural networks (CGCNN) potential will predict the approximate work of adhesion (W^{app}) and calculate reward (r) based on W^{app} , then push new structure information into the replay buffer for later sampling.

Results and Discussion

Performance of CGCNN

To generate the TiAl/TiAlN structures for training the CGCNN model, the optimal initial Ti/TiN system needs to be defined for Al atoms exchange. Three initial types of interfacial stacking sequences, FCC-like stacking, HCP-like stacking, and node-to-node stacking, with 7 different distances from 0.5 Å to 3.5 Å between interfaces in

each type are considered. Fig. 13a illustrates the three types of stacking, the top row shows the 3D view of the three layers of interfaces consisting of 2 bottom layers of TiN and one top layer of Ti. The second row in Fig. 13a shows the details of these three types of stacking sequences, the stacking sequences at the interfaces (dashed line) of FCC-like, HCP-like, and node-to-node are ABC, ABA, and ABB, respectively. For each type of stacking, 7 different interlayer distances between metal and ceramic surfaces are set, and DFT calculates the total free energy of each structure to get the most stable initial Ti/TiN structure. The total free energy of each structure is shown in Fig. 13b, it is evident that the node-to-node stacking (red square) has the highest total free energy, indicating instability compared to the others. The inset in Fig. 13b shows the total free energy of both FCC-like (orange circle) and HCP-like (cyan triangle) stackings under 1 Å and 1.5 Å interlayer spacing. Since FCC-like stacking with 1 Å interlayer spacing has the lowest total free energy and small mismatch, it is chosen as the basic structure for Al doping (as shown in Fig. 9).

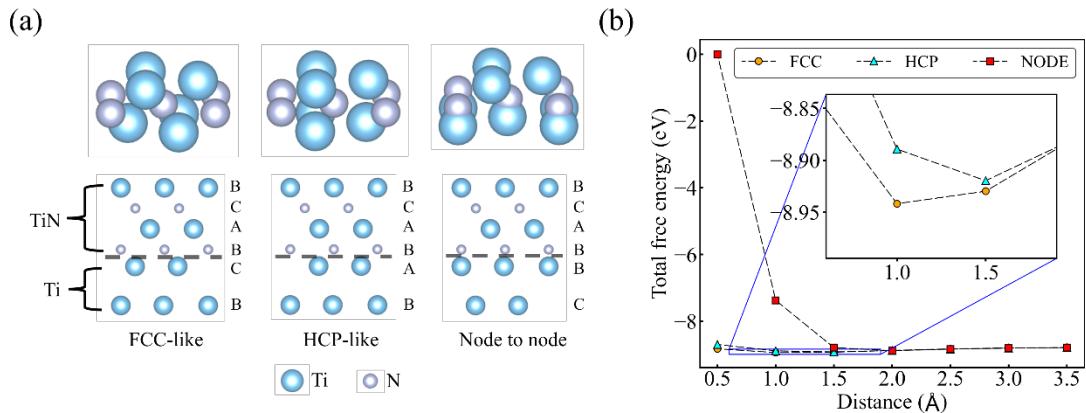


Fig. 13. (a) Three types of stacking at the interface: FCC-like, HCP-like, and node-to-node. (b) Total free energy as a function of interlayer distances between metal and ceramic surfaces, it can be seen from the inset that FCC-like stacking with 1 Å interlayer spacing has the lowest total free energy.

A total of 2000 TiAl/TiAlN structures, each with 50 mol% Al dopants in both TiAl and TiAlN systems, are randomly generated based on basic Ti/TiN structure and get their W^{app} (Eq. 93) by DFT static energy calculations as our training dataset. Fig. 14a is the histogram representing the distribution of the W^{app} in these 2000 structures, this distribution is highly sparse, where only 0.1% (2 out of 2000) structures have W^{app} higher than 11.4 eV. Therefore, it is difficult to find structures with high W_{ad} by using random search. To choose the best model, a train-validation scheme is utilized to optimize the prediction of W^{app} of TiAl/TiAlN structures. Each model is trained with part of the 2000 training data and then validated with part of the rest data, and the best-performing model in the validation set is selected. Here, the mean absolute error (MAE) of W^{app} per atom (eV/atom) under different numbers of training data is shown in Fig. 14b. 5 different sizes of training set, 160, 300, 520, 1000, and 1500, are separately used to train the CGCNN model, it is obvious that the MAE approaches to constant after 500. Furthermore, Fig. 14c shows the W^{app} of the 2000 structures calculated by DFT versus predicted values by CGCNN, the most of points are located on the black dashed line which represents the ideal fit (1:1). CGCNN model is therefore trained with 520 data points is utilized as our dynamic model to train the DQN model.

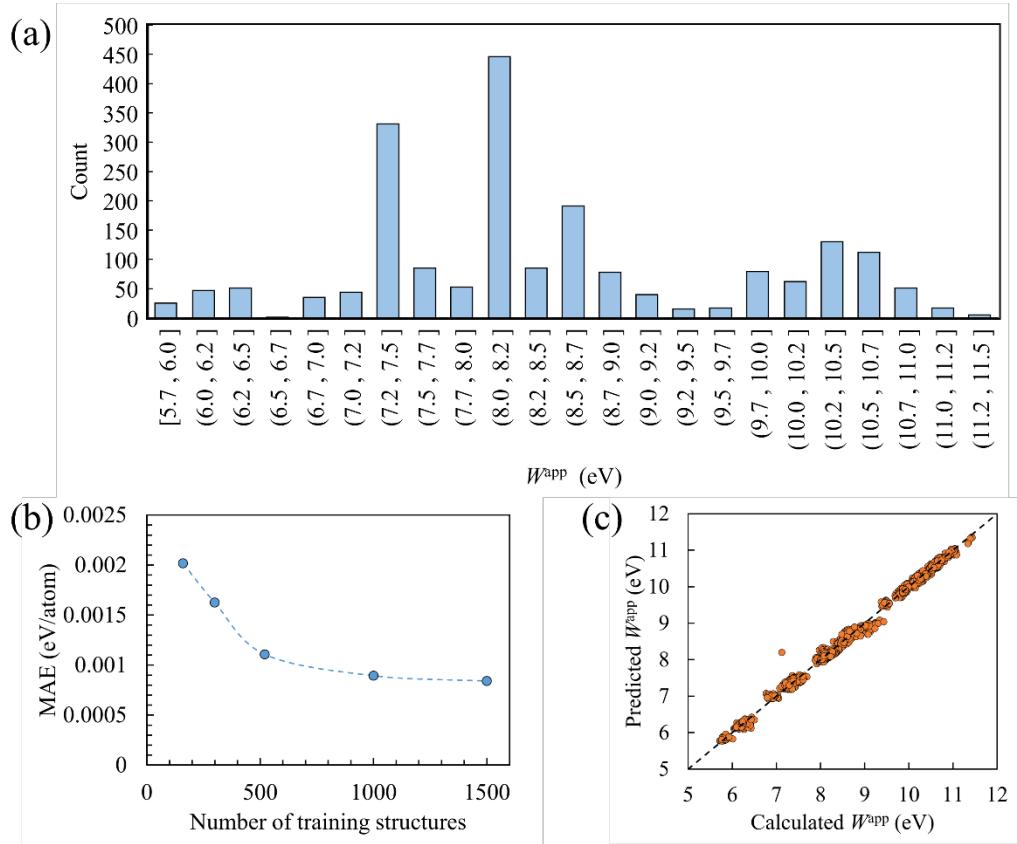


Fig. 14. Performance of crystal graph convolutional neural networks (CGCNN) model.

(a) Histogram representing the distribution of the approximate work of adhesion (W^{app}) in the training dataset. (b) Mean absolute error (MAE) as a function of training data size for predicting W^{app} . (c) The W^{app} of the 2000 structures calculated by DFT versus predicted values by CGCNN, the black dashed line represents the ideal fit.

Performance of DQN

By interacting with the CGCNN model, the DQN can update each state-action pair's Q-value based on the W^{app} (reward) of the current intermediate structure (state) predicted by the CGCNN model during training. Finally, when there is an intermediate structure, the trained DQN can select the combination (action) that can receive the highest Q-value as the next optimal action, by repeating this process until the final structures with high W^{app} are obtained. Here, a total of 2000 training episodes

are applied in DQN, and the mean W^{app} of DQN and random search are evaluated during training. The mean of W^{app} is the average W^{app} of 50 structures generated by a target net of DQN per 10 episodes. We also use a random search method to generate 50 structures per 30 episodes and calculate their W^{app} by trained CGCNN model. The mean W^{app} is shown in Fig. 15a, to compare the performance of DQN with different loss functions, three DQN models are illustrated here, all with a set threshold of 9.5 eV, updating the target net value every 10 episodes, and a batch size of 50, except for using a different loss function. It can be seen with the rise of training episodes, the DQN will learn the output of the CGCNN model (environment) and optimize the final structure to get the higher W^{app} , thus the mean W^{app} increases during the training. The random search, on the other hand, has no ability to capture the information of the CGCNN model, thus the mean W^{app} is oscillated around the highest probability of occurrence (Fig. 14a). Compared among the DQN models, the DQN model with the EWC has highest outputs, since EWC can slow down the learning process and keep the parameters' importance of previous tasks, which leads to more stable and higher outputs. On the other hand, due to catastrophic forgetting, the performance of the DQN model without EWC decreases when trained between 1000 and 1500 episodes, which makes the model less accurate in predicting training samples. After training, the DQN follows an ϵ -greedy policy to generate structures. The performance of trained DQN under different ϵ values is shown in Fig. 15b. 10 different ϵ are used to create 2000 structures in each ϵ , the total time consumption, maximal W^{app} , and the number of structures with W^{app} higher than 11 eV are extracted. With the rise of ϵ , the probability of selection of random action increased, and both max W^{app} (red square) and the number of high W^{app} structures (cyan circle) decreased, this trend is consistent

with Fig. 15a and has been explained previously. When the ϵ increases, the time consumed (orange triangle) is reduced linearly, this is due to the random search not calling the policy net to select action, which requires extra time for outputting Q-values. While the time consumption is increased when using lower ϵ (from 89s to 133s), the number of created potential structures is drastically increased (from 29 to 432) compared to the time consumption, so we use trained DQN model with 0.1 ϵ to generate structures in the following work. Fig. 15c displays the performance MC method for generating high W^{app} structure with DFT no ionic update calculations, here to consist with the training size (520) of the RL model, a total of 55 steps (550 structures) are generated by the MC method. To connect the MC method with DFT calculation, Atomic Simulation Environment (ASE) and VASP are used [210]. Two tests with different initial configurations are conducted to check if the MC method can yield similar final configurations. It can be seen that the two tests have similar results, both smaller than that predicted by DQN (black dashed line), indicating that under the same data size, DQN can find optimal structure more efficiently.

Additionally, instead of using DFT calculations to find optimal structures, like RL and GA methods, the MC method can also be used in the CGCNN model to get optimal structures, as shown in Fig. 10. A comparison of different methods is shown in Table 2. In Table 2, under the same time consumption, both the DQN with combined loss function and GA methods have the highest W^{app} prediction, which is 15.552 eV, but DQN has a higher prediction in total. On the other hand, the MC method had the lowest predictive value in both tests. What is more noteworthy is that for the number of high W^{app} structures, the MC method has few or even only one structure among the 384,000 generated structures, while DQN and GA have more

high- W^{app} structures in 70,000 and 368,050 generated structures respectively, and both have high deviation for count of high W^{app} structures. In brief, DQN and GA methods can generate more candidate structures than MC methods under the same time consumption for small structures, and GA has the largest and the most efficient generation of candidate structures in total.

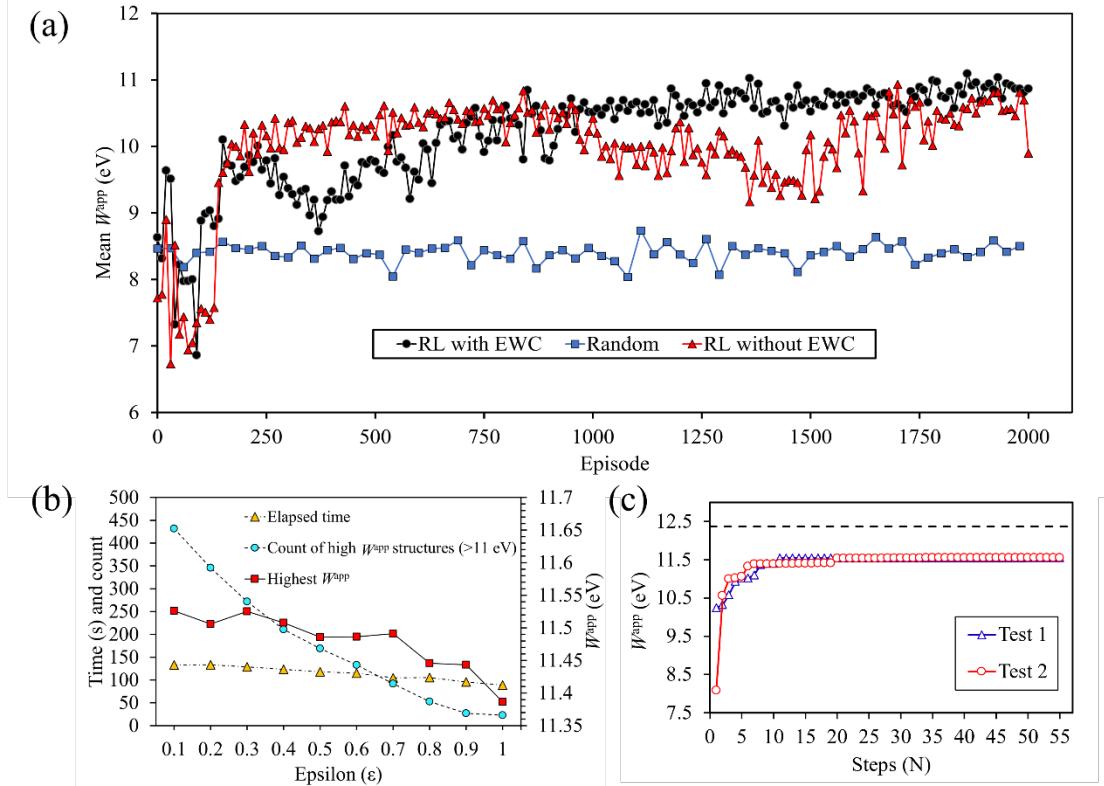


Fig. 15. Performance of deep Q-network (DQN) model. (a) The evolution of mean approximate work of adhesion (W^{app}) of generated structures by DQN models with different loss functions and random search during training. (b) The time consuming of trained DQN under different ϵ values. (c) W^{app} versus steps in the Monte Carlo (MC) method.

Table 2. Performance of deep Q-network (DQN), genetic algorithm (GA), and Monte Carlo (MC) methods.

Methods	DQN test 1	DQN test 2	GA test 1	GA test 2	MC test 1	MC test 2
Highest W^{app} (eV)	11.552	11.535	11.552	11.520	11.511	11.504
Count of high W^{app} (>11.5 eV)	95	20	337	28	4	1

TiAl/TiAlN Interface Generation and Work of Adhesion

We apply the CGCNN model-based RL agent and random search to generate 100000 and 350000 TiAl/TiAlN structures which have different W^{app} . In these structures, 60 and 12 different TiAl/TiAlN structures with W^{app} higher than 11.5 eV are generated in RL agent and random search separately. The W^{app} calculated by both DFT and CGCNN models of these 72 structures and extra 800 structures in different W^{app} ranges are plotted in Fig. 16a. In general, the predictions fluctuate above and below the DFT values, which indicates the good performance of the CGCNN model. For high W^{app} , however, the predictions are lower than that of DFT, as shown in the inset of Fig. 16a. As the W^{app} increases, the predicted value of CGCNN gradually approaches an upper limit, causing its predicted value to oscillate around 11.5 (green dotted line). This systematic error is determined by the limitation of the training set. The upper limit of W^{app} in training data is less than 11.5 (11.43 eV), which results in the model's extrapolation capability being limited to around 11.5. Even so, the existing systematic errors are small and there are no cases of excessive errors, and the overall growth trends of DFT and predicted values are similar, this indicates the graph-based features in CGCNN capture some local environment information of each atom, this enables the CGCNN model to have certain extrapolation capabilities under

similar structures. Even if the CGCNN model underestimates the actual W^{app} , but its predicted value still falls within the highest energy range in the DFT calculation. Therefore, we select structures with a predicted energy greater than 11.5 as our analysis objects. Besides, the CGCNN are trained by the structures without relaxation to improve efficiency, which means we use Eq. 93 to get the W^{app} of unrelaxed structures. This can lead to discrepancies with the relaxed value W_{ad} (Eq. 92). Therefore, to check if the W_{ad} has a similar energy distribution with W^{app} , the W_{ad} and W^{app} of the 20 structures with high W^{app} (18 for RL agent and 2 for random search) and extra 27 random structures are calculated by DFT as well. Fig. 16b plots the W_{ad} and W^{app} calculated by DFT values. We can see though the overall W_{ad} is smaller than W^{app} , they have a similar overall change trend. The cyan and green dashed lines are the W^{app} and W_{ad} of unrelaxed Ti/TiN and relaxed Ti/TiN correspondingly, some parts of W^{app} and W_{ad} of TiAl/TiAlN are higher than that of Ti/TiN, which shows the TiAl/TiAlN has stronger bonds at the interface. The structures within the black dotted box are the high W^{app} structures predicted by both RL agent and random search and their corresponding W_{ad} after relaxation, we can see the structures that have W^{app} higher than 11.5 eV also have higher W_{ad} than others, this also confirms that it is feasible for us to use the W^{app} of the structure without relaxation to train our model and predict the high W_{ad} structure.

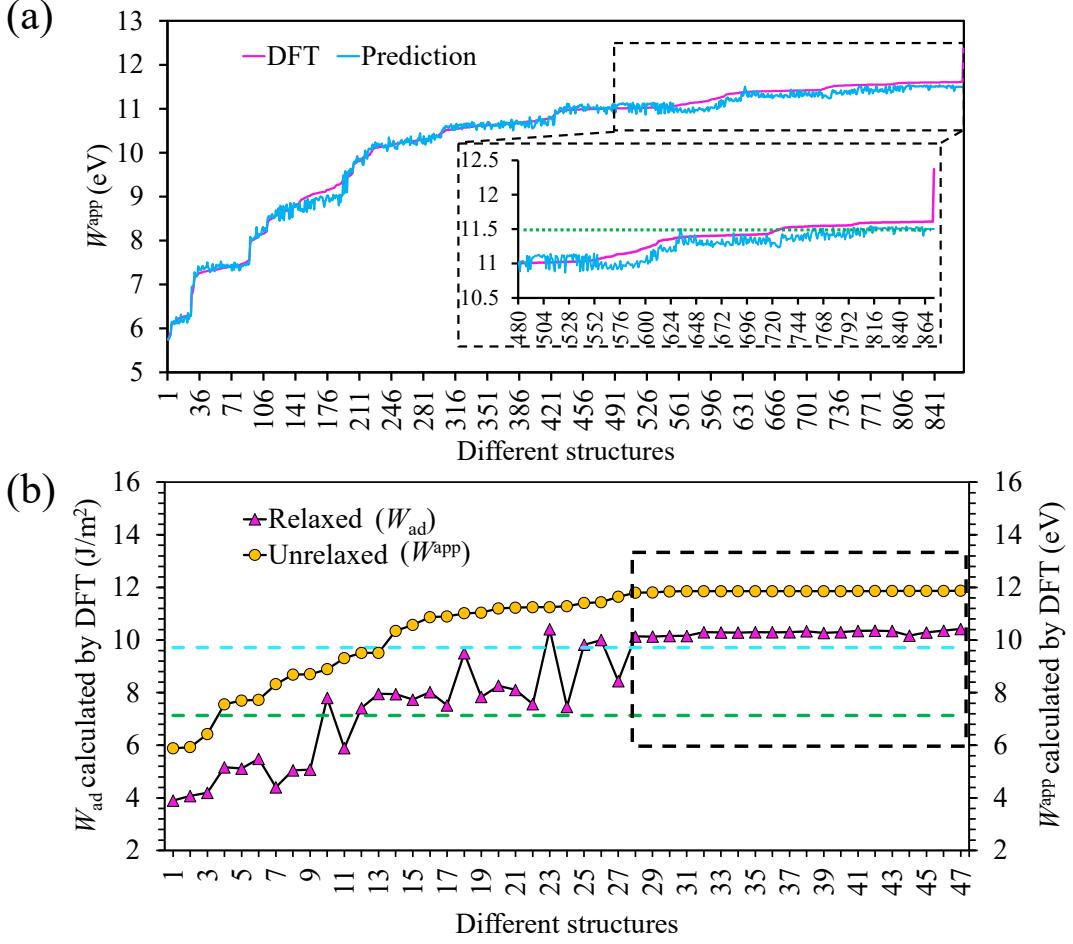


Fig. 16. (a) The approximate work of adhesion (W^{app}) calculated by both DFT and crystal graph convolutional neural networks (CGCNN) model of 72 predicted high W^{app} structures (60 for reinforcement learning (RL) agent and 12 for random search) and 800 structures in different W^{app} . (b) The work of adhesion (W_{ad}) and W^{app} of the 20 structures with high W^{app} (18 for RL agent and 2 for random search) and extra 27 random structures.

Interpretability of CGCNN and RL Model

To further investigate the structures with high W^{app} , both CGCNN and RL models are interpreted by using principal component analysis (PCA) to gain some insights. In the CGCNN model, two vectors, local environment representation and averaged total

representation, are analyzed. The representation of the local environment is the output of the convolutional layers, when the vector of input which contains the atoms and bonds information passes through t convolutional layers, the convolution operations (Eq. 94) will make the vector have the local environment information such as atomic types and geometries. Therefore, the local environment representation can describe the similarity between the local environments of any two structures. The averaged total representation is the output vector obtained by adding a pooling layer (Eq. 104) after the convolutional layer. This process is analog to classical potential's total energy, which calculates the total energy by summing the local atomic energies of all atoms in the system, here, the averaged total representation contains the contribution of each local chemical environment and shows the structure's information. For the RL model, since the policy network can output the Q-value based on the current state and next action, the Q-value of each state with different following actions can be calculated by the trained RL model and visualized by PCA.

$$v = (\sum_i v_i^t)/n \quad (104)$$

Fig. 17a and Fig. 17b show the PCA of local environment representation and averaged total representation of the CGCNN model, respectively. A total of 2072 structures consisting of train data and the 72 high W^{app} structures predicted by RL and random search methods are plotted. For local environment representation, due to each structure having 56 atoms, the local environment representation for each structure is a 56×64 matrix. So, before applying PCA, the local environment representation is flattened from a 56×64 matrix to a 1×3584 vector. The averaged total representation reduces the 56×64 matrix to 1×64 vector by element-wise addition of each local environment representation (Eq. 104). After PCA, the original 3584 and 64-

dimensional vectors are reduced to 2 dimensions. In Fig. 17a and 9b, the structures with similar W^{app} are clustered together. Except for the cluster with the highest W^{app} (since these structures come from the 72 predicted structures that are out of training data), the distribution of the rest clusters is similar to the histogram of training data (Fig. 14a), and the energy is gradually changed in one direction, all indicate that the CGCNN model embeds the physical meaning such as the geometry of the structure, atomic information, and local environment of the atom into each representation. By further inspecting the structures of both high and low W^{app} , some specific patterns for these two types of structure, especially around the interface of TiAl/TiAlN are found. As shown in Fig. 17b, two examples of interface for high W^{app} structure and low W^{app} structure are illustrated in red dashed inset and blue dashed inset, respectively. Total 10 atomic layers, where top 6 layers come from TiAlN and bottom 4 layers come from TiAl, are extracted in each structure, it can clearly see that the metal layers (5th and 7th layer from top to bottom) on both sides of the N atomic layer (6th layer from top to bottom) at the interface exhibit an inverse pattern in these two types of structures, where the element of the metal layer is Al and the ceramic layer is Ti for high W^{app} structures, on the contrary, the element of the metal layer is Ti or Ti + Al and the ceramic layer is Al for low W^{app} structures, for the other structures, their interface have the patterns other than these patterns. What's more, the AlN structure mainly appears at the interface of high W^{app} structures, and an Al layer follows the Ti layer at the interface of the TiAl part. Fig. 17c shows the 2D Q-values when the ceramic layer generation is completed, and the metal contact layer is about to be generated (as shown in the inset of Fig. 17c). Here, a 1×512 Q-value vector predicted by the middle-hidden layer for each structure are utilized, and each vector is

reduced to 2D by PCA. The Q-values larger than 3 and smaller than 1.5 are plotted, a clear separation can be seen in these Q-values, and by comparing to the structures and actions in both high and low Q-values, we can notice that for high Q-value structures, their following actions mainly generate the Ti atomic layer, and their previous non-N atomic layer is the Al layer, for low Q-value structures, they are just the opposite of the high Q-value structures, which is consistent with the two patterns found previously. Besides the interpretability of CGCNN and RL models, the similarity of each structure compared to others is also studied. To simply calculate the similarity between any two structures, the tensor form ($28 \times 3 \times 7$) of each structure is used. By calculating the dot product of the corresponding row vectors (1×7) of the structure tensor in any two structures to get the correlation of these two vectors, this process is similar to the cosine similarity calculation of two matrices, but the cosine similarity between any two structures is always 1 since any two corresponding row vectors are in the same direction, so the magnitude of each vector is ignored here in order to take into account the influence of the type of atom. Then, the element-wise summation is applied for each tensor to reduce the dimension from $28 \times 1 \times 3$ to 1×3 . Repeat these processes until each structure is compared to all structures including itself, after the flattening process, the final similarity vector of size 1×6216 for each structure is obtained, this vector refers to how similar a structure is compared to other structures. Fig. 17d is the 2D similarity of each structure, the high W^{app} structures are clustered at the top region, though other structures are more sparsely scattered across the figure due to the larger structure space. This indicates that there are more similarities between high W^{app} structures' vectors, in other words, the high W^{app} structures are similar, which also explains why the values of high W^{app} or W_{ad} have similar values

(Fig. 16b black dotted box). Based on the above analysis, it can be concluded that the RL model learns the structures and builds a method that can generate high W^{app} structures, and these structures follow a certain pattern which can lead structures to have high W^{app} .

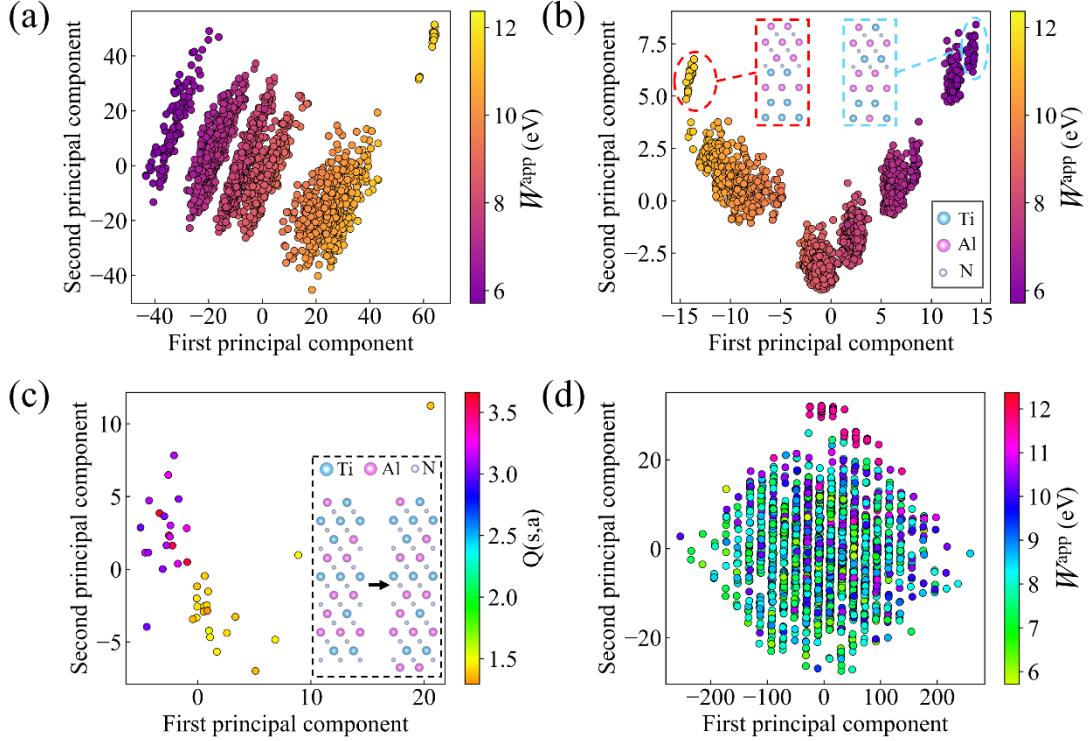


Fig. 17. Visualization of different feature vectors by principal component analysis (PCA). (a) PCA of output vectors of the convolutional layers in crystal graph convolutional neural networks (CGCNN). (b) PCA of output vectors of the pooling layer in CGCNN. (c) PCA of output vectors of middle hidden layer in deep Q-network. (d) The similarity vector of each structure compared to others is visualized by PCA.

Electronic Structure and Bonding

To understand the strengthening mechanism of the interface, we calculate the charge density of the TiAl/TiAlN and Ti/TiN by DFT. The information on bonding

and electronic structure of the interface can be obtained by charge density difference.

The charge density difference is calculated by:

$$\Delta\rho = \rho_{\text{all}} - \rho_{\text{metal}} - \rho_{\text{ceramic}}, \quad (105)$$

where ρ_{all} is the total charge density of the TiAl/TiAlN or Ti/TiN system, ρ_{metal} and ρ_{ceramic} are the charge density of the corresponding metal and ceramic parts, respectively.

The charge density difference at the interface of the TiAl/TiAlN and benchmark Ti/TiN structures are investigated here. Fig. 18a, b, and c show the 2D charge density difference on $(\bar{1}\bar{1}20)/(1\bar{1}0)$ plane of the TiAl/TiAlN interfaces with high W_{ad} (10.34 J/m^2), low W_{ad} (3.90 J/m^2), and Ti/TiN (7.14 J/m^2) interface, respectively. The top purple dashed line is the surface of ceramic, and the bottom represents the surface of metal. The red and blue areas separately represent electron accumulation and depletion. In these figures, some atoms are numbered for identification. From these figures, we can see that atom 4 at the metal-ceramic interface transfers charges to atom 3 at the ceramic interface and forms a bond. For high W_{ad} and Ti/TiN structures, this bond exhibits ionic characteristics, while for low-energy structures, this bond is more like a polar covalent bond. The different features of the bond between these two types of bonds can be explained by the bond strength since the Ti-N bond (476.1 ± 33.1) is stronger than the Al-N bond (297 ± 96) [273]. In the case of high W_{ad} structure (Fig. 18a), aluminum atom 2 transfers charge to atom 1, atom 4, and atom 8, thus forming bonds or interactions with these atoms. Furthermore, we can observe that atom 1 that gains charge also interacts with atom 4 at the metal interface. The charge distribution in the ceramic part of Ti/TiN is similar to the high W_{ad} structure, and atom 4 also interacts with atom 1. It's worth noting here that since the size of the aluminum atom

is smaller than that of the titanium atom, when the atom 2 is aluminum, i.e., for high W_{ad} structure, the distance between the upper and lower layers will be smaller compared to when the atom 2 is Ti. In Fig. 18a, b, and c, the distances between atom 1 and atom 4 are 3.61 Å, 3.77 Å, and 3.64 Å, respectively, so the interaction between these two atoms in the high W_{ad} structure is easier to form. For the low W_{ad} structure, titanium atom 2 increases the distance between atoms 1 and 4, and the accumulation of charge on atom 2 and the charge deficiency of atom 4 make the interaction among atoms 1, 2, 3, and 4 weak and difficult. In the metal part, atom 4 also interacts with the neighboring metal layers, which can be seen from the figures. For high W_{ad} and Ti/TiN structures, atom 4 interacts with atoms 5 and 7, and the interaction is stronger in the high W_{ad} structure. In addition, atom 6 in the high W_{ad} structure also has a long-range effect on atom 4. Similarly, because atom 5 and atom 7 are aluminum, the distance between atom 4 and atom 6 in the high W_{ad} structure is the shortest (4.55 Å), while the distances for the low W_{ad} structure and Ti/TiN structure are 4.85 Å and 4.66 Å, respectively. For the high W_{ad} structure, the half of 4.55 Å Ti-Ti distance is closest to the van der Waals radius of titanium element [274], so atom 6 also has a weak interaction with atom 4. Furthermore, for the high W_{ad} structure, atom 5 and atom 7 have long-range effects on atom 3 and atom 8, respectively. For the low-energy structure, due to the displacement of the metal layers, atom 4 mainly interacts with atom 7, resulting in weak interactions between the metal interface and its internal layers. Fig. 18d to f illustrates the plane-averaged charge density difference along the z-direction ([0 0 0 1]/[1 1 1] direction) calculated by VASPKIT [275]. Same as Fig. 18a to c, the region between the purple dashed lines is the interface. It can be seen that the charges transfer from the metal part (bottom dashed line) to the ceramic part (top

dashed line), and the main charge rearrangement takes place at the interface, where the accumulation and depletion peaks are also located. The peaks of accumulation (depletion) for high W_{ad} structure, low W_{ad} structure, and Ti/TiN structure are 4.27 (-3.58) e/Å, 1.94 (-2.98) e/Å, and 2.97 (-2.88) e/Å, respectively. The skewed and small peaks in the low W_{ad} structure can also confirm that the interactions among atoms 1, 2, 3, and 4 are weak. The charge accumulation on atom 2 and the charge deficiency of atom 4 leads to less accumulation around the surface of the ceramic. So, the high W_{ad} structure has the largest charge accumulation and depletion peaks, indicating the strengthening of the interfacial interaction and charge transfer compared to the other two structures.

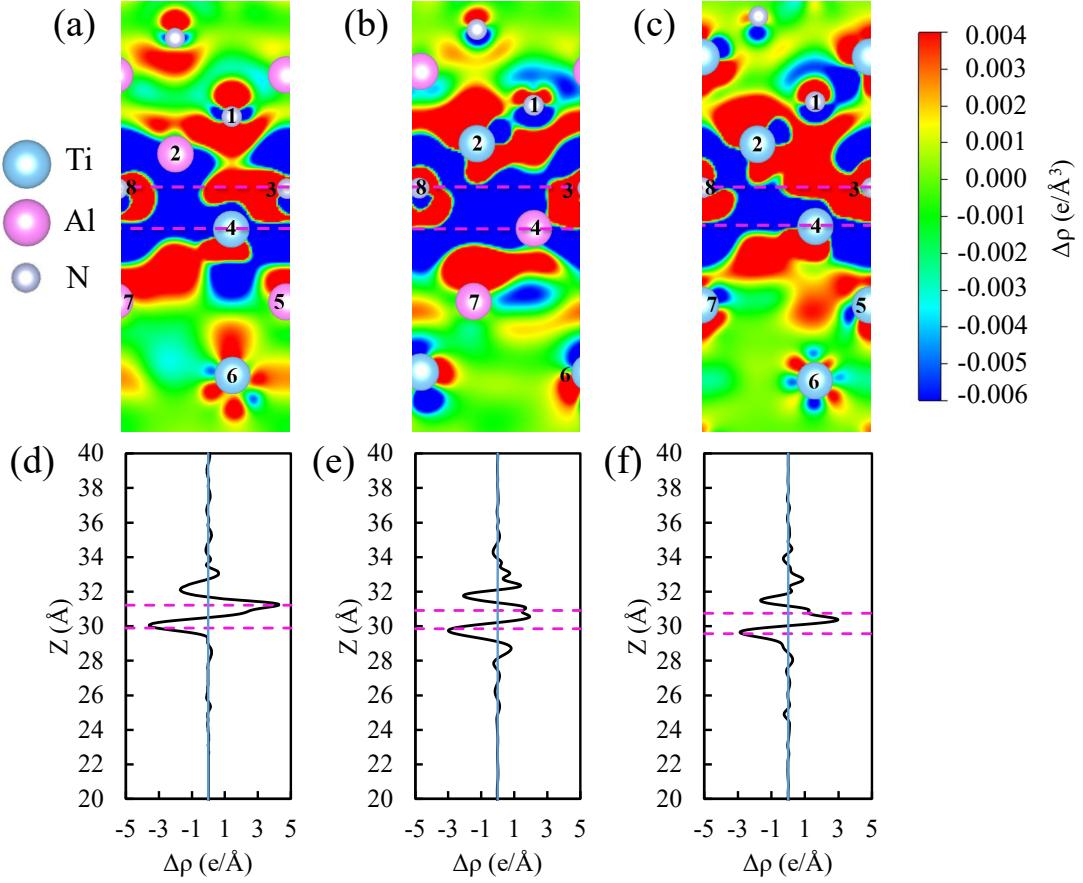


Fig. 18. (a) Charge density difference of TiAl/TiAlN interface with high work of adhesion (W_{ad}) on $(\bar{1}\bar{1}20)/(1\bar{1}0)$ plane. (b) Charge density difference of TiAl/TiAlN interface with low W_{ad} on $(\bar{1}\bar{1}20)/(1\bar{1}0)$ plane. (c) Charge density difference of Ti/TiN interface on $(\bar{1}\bar{1}20)/(1\bar{1}0)$ plane. (d) Plane-averaged charge density difference of TiAl/TiAlN interface with high W_{ad} along $[0\ 0\ 0\ 1]/[1\ 1\ 1]$ direction. (e) Plane-averaged charge density difference of TiAl/TiAlN interface with low W_{ad} along $[0\ 0\ 0\ 1]/[1\ 1\ 1]$ direction. (f) Plane-averaged charge density difference of Ti/TiN interface along $[0\ 0\ 0\ 1]/[1\ 1\ 1]$ direction.

To quantitatively compare the interface charge transfer, the Bader charge is used [276-279]. The average charge as a function of the atomic layer at the interface of different structures is plotted in Fig. 19. The left and right purple dash-dotted lines are

the surface of ceramic and metal, respectively. In the ceramic, the electronegative N atoms accumulate the amount of negative charge, and the Ti or Al atoms have a positive charge. In the metal, layer 19 has a positive charge in all structures, in layer 20, both high W_{ad} and low W_{ad} structures have a negative charge though the Ti/TiN structure possesses a slightly positive charge. The result of Ti/TiN is the same as the found of Miraz et al [242], they think that at least two layers in the Ti-phase will be influenced by the electronegative N layer, which can weak the interactions and generalized stacking fault energy (GSFE) barrier between these two layers. This phenomenon can also be observed in Fig. 18c where atoms 3 and 8 have an effect on atoms 4, 5, and 7. They also noticed that by incorporating slightly more electronegative Al atoms in layer 20 can decrease the average charge of layer 20 since Al and Ti atoms will be more attractive than Ti and Ti atoms [280]. So, it can increase the GSFE barrier. Interestingly, this dopant method is also applied by our RL model, which uses the same method to generate the high W_{ad} structure, as shown in Fig. 18a, atoms 5 and 7 are Al atoms. This method can lead to a more negative average charge in layer 20, thus it can have more attraction to layer 19, which leads to layer 19 having a more positive average charge. It can be seen that in layers 18 and 19, both high W_{ad} and low W_{ad} structures have more negative or positive average charges than that of Ti/TiN, especially for low W_{ad} , which has the largest positive average charge in layer 19. Even though, due to the lower average charge in layers 17, 20, and 21 compared to that of the high W_{ad} structure, especially for layer 17, the average charge is similar to that of the Ti/TiN structure. Therefore, the bonding strength between layer 17 and layer 18, layer 19, and layer 20 are lower than that of the high W_{ad} structure. Compared with the high W_{ad} structure and Ti/TiN structure, in the case

where the atoms of both structures in layer 19 are Ti, the high W_{ad} structure have generally more negative or positive average average charge around the interface thus high W_{ad} structure has higher bonding strength.

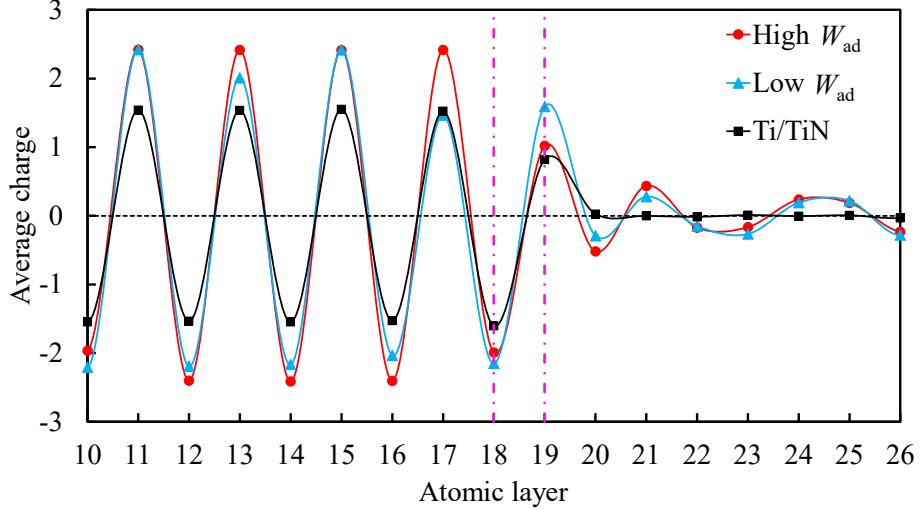


Fig. 19. The average charge as a function of the atomic layer at the interface of different structures.

Summary

We have developed a DQN model combined with CGCNN potential to identify Al-doped TiAl/TiAlN interfaces with 50 mol% Al with higher W_{ad} than that of the Ti/TiN counterpart. By comparing the performance with the random search method, the RL model is more efficient and can find the TiAl/TiAlN interface with higher W_{ad} than those identified by random search and MC method. Additionally, the visualization of features from DQN and CGCNN indicates that high W_{ad} structures follow a specific pattern of doping Al atoms near the interface, with Al and N layers alternating on the ceramic surface and a single Ti layer followed by a single Al layer on the metal surface. Further analysis of the electronic structure and bonding of W_{ad} , low W_{ad} , and Ti/TiN structures reveals that, for the TiAl/TiAlN interface with high

W_{ad} , the smaller atom size of Al can reduce the distance between the Ti layer of the metal surface and the second nearest neighbor N layer, as well as the distance between the Ti layer of the metal surface and the second nearest neighbor Ti layer in the metal. This reduction enhances interactions, such as van der Waals forces, between them, leading to stronger bonding at the interface. Bader charge analysis also demonstrates that the presence of Al atoms in the second layer of the metal part can result in a more intense interaction with the surface Ti layer, thereby increasing the bond strength at the interface. Although further research is needed on the methods used in this paper, using machine learning potentials to approximate or replace DFT calculations holds a great potential in providing a fast preliminary screening method for other optimization problems with large configuration spaces, and by combining RL, structural correlations and features can be discovered from these potential configurations.

CHAPTER 4

RE-EVALUATING SHORT-RANGE ORDER IN SI-GE-SN MEDIUM ENTROPY ALLOYS USING CRYSTAL GRAPH CONVOLUTIONAL NEURAL NETWORKS AND REINFORCEMENT LEARNING CALCULATIONS

Introduction

Medium-entropy alloys (MEAs) and high-entropy alloys (HEAs), which are generally referred to as the alloys composed of equal or relatively large proportions of three, five or more elements, respectively, have received lots of attention due to their outstanding mechanical properties such as promising combinations of tensile strength and ductility [15, 281, 282] and resistance to corrosion [283] compared to traditional alloys, and modify the thermoelectric properties [16, 284, 285]. Although the three-dimensional direct image of atomic structures of MEAs or HEAs is challenging, a number of recent studies of MEAs, especially in chromium–cobalt–nickel (CrCoNi) MEAs and silicon–germanium–tin (SiGeSn) MEAs, have shown the existence of short-range chemical order (SRO), where certain elements in a pair repel or attract each other to obtain energetically favorable configurations. For example, ab initio-based Monte Carlo (MC) simulations in which using density functional theory (DFT) calculations are applied in MEAs to show the SRO is thermodynamically favored in MEAs or HEAs and affect the mechanical, electrical, and other properties. [286-288]. Besides, in recent years, experimental observations of SRO also illustrate the existence of SRO in MEAs and HEAs and show the effect of the SRO on mechanical and physical properties [289-294].

Since the SRO is thermodynamically favored in MEAs and HEAs, the tendency of SRO formation is very strong, and the SRO is strongly related to the free energy

landscape and their configurational space. As the number of the element increases, the free energy landscape becomes more complex due to their intricate local atomic environment [288, 295]. Due to the accurate calculation of total energy by DFT relaxation after each atomic exchange, DFT-based MC simulations have been successfully applied to the SRO studies of different alloys [286-288, 296, 297]. However, the complexity of free energy landscape of MEAs and HEAs makes this simulation time-consuming. For example, for a 64-atom $\text{Si}_{0.125}\text{Ge}_{0.625}\text{Sn}_{0.25}$ MEA alloy initial random configuration, 15,000-20,000 MC steps (DFT relaxation calculations) are conducted to ensure sufficient sampling, and over 1 million DFT calculations are required for entire composition domain investigations [288], and the amount of DFT calculations is enormous for other MEAs' cells with more atoms.

Machine learning (ML) methods such as neural networks (NNs) have shown their compelling abilities in materials science for predicting material properties such as potential energy based on atomic structures or to create atomic structures with desired properties [245, 246]. For the ML methods to predict potential energy, the graph-based NNs, also called message-passing networks, have been successfully applied to interatomic potential construction due to their accuracy is comparable to that of DFT calculation [124, 127, 132]. As an optimization method, reinforcement learning (RL) has been used in different design tasks, such as development of interatomic potential parameters [298], chemical reaction optimization [250, 251], and molecular discovery [248]. In recent years, the combination of NNs and RL have illuminated a new possibility to predict and design materials efficiently. For instances, convolutional NNs and deep Q-network (DQN) are used to design two-dimensional (2D) stretchable MoS_2 nanosheet with high stretchability and more efficient than active learning [249].

And use graph-based NNs and DQN to study the adhesion of 50% Al-doped TiA/TiAlN interface in CHAPTER 3. Here, we use the similar CGCNN model illustrated in CHAPTER 3, combined with MC method to search for low-energy $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$. By comparing MC/CGCNN with MC/DFT, the CGCNN can extrapolate the energy for new $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures and can predict lower energy structures using the same training set as the number of MC/DFT simulation steps. Besides, we found that there exists a different SRO pattern from other SiGeSn compositions, which doesn't have strong repulsive interaction for the first-nearest Si-Sn bond. This found indicates the SRO of MEAs with same elements under different concentration may be different.

Methodology

MC Sampling

The Metropolis MC method [299] combined with trained CGCNN, or *ab initio* calculations are used to investigate the SRO and free energy evolution of equimolar $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ ternary. Each MC step includes a trial move of randomly selecting and swapping two different types of atoms in configuration i to create a new configuration j , then a full relaxation is applied to obtain the energy E_j of j . The acceptance probability for a new configuration j is $\min\{1, \exp(-(E_j - E_i)/k_B T)\}$, where E_i and E_j are the total energies of configuration before and after swapping, respectively, k_B is the Boltzmann constant, and T is temperature (300 K in this study).

DFT Calculations

The Vienna ab initio simulation package (VASP) is utilized for the DFT calculations [255-257]. A $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ supercell containing 216 atoms with $3 \times 3 \times 3$ diamond lattice is chosen. The Projector Augmented Wave (PAW)

pseudopotentials [201, 259] provided by VASP are used for core electrons, and the plane wave basis sets [200, 260] with a cutoff energy of 400 eV are utilized for both no ionic update calculation and fully relaxation. A $2 \times 2 \times 2$ mesh centered at Gamma is used for the $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ supercell with periodic boundary conditions [261]. Generalized gradient approximation (GGA) parameterized by Perdew, Burke, and Ernzerhof (PBE) [258] and local density approximation (LDA) [300] are separately used for exchange-correlation of electrons. The conjugate gradient algorithm is applied for structural relaxation, with the convergence criteria of 10^{-4} eV for electronic relaxation and 0.01 eV/ Å for ionic relaxations.

SRO Parameter

Here, we follow the SRO parameters defined in Ref. [288] to quantify the SRO of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ MEA. The SRO α_{ij}^ν is defined in equation 1

$$\alpha_{ij}^\nu = 1 - \frac{n_{ij}^\nu}{n_{0,ij}^\nu}, \quad (106)$$

where n_{ij}^ν is the number of pairs between atom of type i and atomic species j in the ν -th neighboring shell under the actual distribution, and $n_{0,ij}^\nu$ is the number of pairs under a random distribution. A random distribution corresponding to $\alpha_{ij}^\nu = 0$. $\alpha_{ij}^\nu > 0$ shows a tendency to decrease the number of i, j pairs, while $\alpha_{ij}^\nu < 0$ indicates a preference of i, j pairs. This method is related to Warren-Cowley SRO parameter [301].

CGCNN for Total Energy Prediction of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$

Two CGCNN models are separately trained by two training datasets in which 216-atom $3 \times 3 \times 3$ diamond lattice $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ atomic structures are randomly generated and the output is total energy, the fractional coordinates of all atoms in

these two datasets are consistent except for one with lattice constants of 16.406 Å and the other with lattice constants of 17.9 Å, which are pure Si lattice constants and approximate value of optimized lattice constants, respectively. Same as Eq. 94, each CGCNN has 3 convolutional layers for updating the atom feature vector \mathbf{v}_i with a nonlinear graph convolution function:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \sum_{j,k} \sigma[(\mathbf{v}_i^t \oplus \mathbf{v}_j^t \oplus \mathbf{u}_{(i,j)_k}^t) \mathbf{W}_f^t + \mathbf{b}_f^t] \odot g[(\mathbf{v}_i^t \oplus \mathbf{v}_j^t \oplus \mathbf{u}_{(i,j)_k}^t) \mathbf{W}_s^t + \mathbf{b}_s^t] \quad (107)$$

where \oplus concatenate central atom feature vector \mathbf{v}_i^t , neighbor atom feature vector \mathbf{v}_j^t , and bond feature vector $\mathbf{u}_{(i,j)_k}^t$ in t^{th} convolution process. \mathbf{W}^t and \mathbf{b}^t are convolution weight matrix and bias of the t^{th} convolution process which consider the interaction strength between neighbors. The \odot denotes element-wise multiplication. σ and g are activation functions such as Sigmoid function and Softplus function, which introduce nonlinear coupling between layers. The maximum number of considered neighbors is 28. After convolution, the atom features are converted to an overall feature vector by a pooling layer and then fed into two fully connected layers of dimensions 64×128 and 128×1 with Softplus to get the final output. Details of the method can be found in Xie and Grossman's study [127].

Reinforcement Learning Agent (A2C)

The input to the A2C network is a $13 \times 13 \times 13$ tensor representing intermediate $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure state s_t at time t , three convolution layers with dimensions of $16 \times 13 \times 13 \times 13$, $32 \times 12 \times 12 \times 12$, and $32 \times 5 \times 5 \times 5$, respectively, combined with pooling and flatten layers to reshape input tensor to a feature vector with 256 size. Then a LSTM cell is used to add past state's memory and encode the feature vector to another 256-sized vector [302]. And this vector is fed into actor and critic to

separately output the action a_t corresponding to current state and state value V_{t+1} . The actor contains three layers with dimensions of 256×256 , 256×128 , and 128×3 , and the critic have three layers with dimensions of 256×256 , 256×128 , and 128×1 , the hyperbolic tangent (\tanh) activation function is used for nonlinear process.

RL Framework for $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ Configurations Design

The RL model is applied to find an optimal policy (actions) that generates the 216-atom $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures with low energy. Since the DFT calculation of the total energy of each $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure is computationally demanding, treat the CGCNN model as the environment to interact with RL to predict the total energy of each 216-atom $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure. The architecture of the CGCNN is similar to what we did in CHAPTER 3 except that the maximum number of neighbors is 28 instead of 12 to contain the number of nearest neighbors of first three shells. Details of the CGCNN’s network architecture are given in the “Methods” section. The input to the RL is a $13 \times 13 \times 13$ tensor representing intermediate or final $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure, which also represents the state s_t at a certain step t . different from the definition of actions that represent combinations of the atomic layer in CHAPTER 3, we map the 216 atoms of the $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure into 216 elements of $13 \times 13 \times 13$ tensor, and each element represents one action that select one type of atom among Si, Ge, and Sn, so we decompose the process of generating a 216-atom $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure into 216 steps of placing elements in the corresponding positions of the tensor in sequence, and completing these 216 steps is called an episode. Each step t will select an action a_t to place an element at the corresponding position of the tensor according to the current state s_t . Each selection contains up to three actions to represent the type of atoms placed. Here we use four numbers to

represent the elements of the tensor, 0 represents none, 1 represents Si, 2 represents Ge, and 3 represents Sn. Therefore, the elements in the initial tensor at the start of the episode are all zeros, and the elements in the tensor will change as the step goes. Each action will change the current state s_t to the new state s_{t+1} , and the total energy of the new state s_{t+1} is calculated from the CGCNN and return it to RL as a reward. The reward is defined below:

$$\text{reward}(t) = \begin{cases} 1 & (t < T) \\ E & (t = T) \end{cases}, \quad (108)$$

the reward is 1 if the structure is not in the terminal state otherwise is the total energy predicted by the CGCNN. Although DQN performs well in dealing with problems with fewer steps, when faced with 216 steps, DQN will significantly reduce sample efficiency due to the rise of the training batch size, and it requires a lot of memory to store training data. Therefore, we use Advantage Actor Critic (A2C) as RL to handle multi-step tasks [303]. Actor and Critic are two neural networks, Actor $\pi_\theta(a|s)$ is a policy function parameterized by θ that controls how the RL agent acts under certain state s , and Critic $V_\omega(s)$ is a state value function parameterized by ω that assists the Actor to update by measuring how good the state s is. A2C not only requires less memory and training time, but also reduces variance and stabilizes learning compared to models with only policy function due to the combination of value function and advantage method. Details of the A2C's network architecture are given in the "Methods" section. Fig. 20 illustrates the schematic of the A2C agent and CGCNN model for design low-energy $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures. For a structure at step t in one episode, the current state s_t represents a $13 \times 13 \times 13$ tensor, and this tensor as the input of A2C to pass through convolutional neural network (CNN) and long short-term memory (LSTM) [302], the LSTM can store the information of the past state,

and then separately pass Actor and Critic to get policy $\pi_\theta(a_t|s_t)$ and expected reward $V_\omega(s_t)$ (state value) start from s_t . The action a_t is sampled from the policy $\pi_\theta(a_t|s_t)$ of s_t and use a_t to generate next state s_{t+1} , and calculate state value $V_\omega(s_{t+1})$ of s_{t+1} . Except for the policy and state value, s_{t+1} also converted to atomic structure and fed into CGCNN, in CGCNN, the atom and bond information of atomic structure is firstly extracted to create an undirected multigraph, which is then passed through convolution layers, pooling layer, and hidden layers to predict the total energy E and output the reward r_{t+1} of s_{t+1} based on Eq. 108. Then the loss can be calculated by:

$$Loss = \frac{\sum_t^T \log \pi_\theta(a_t|s_t) A(s_t, a_t)}{T} + \beta \frac{\sum_t^T A(s_t, a_t)^2}{T} + \varepsilon \sum_t^T \left(\sum_n^N \frac{-p(a_{n,t}|\pi_\theta) \log p(a_{n,t}|\pi_\theta)}{N} \right), \quad (109)$$

the terms of right side are loss of policy function, loss of value function, and entropy of policy function, respectively, and the β and ε are scale factors. The loss of policy function is the mean of multiplication of log of the policy distribution and advantage A in one episode with T steps, the advantage A is defined in Eq. 110, $Q_v(s_t, a_t) - V_\omega(s_t)$ shows how better taking that action a_t at a state s_t is compared to the average value of the state, it introduces a baseline $V_\omega(s_t)$ to make the gradients of policy function smaller which is more stable for training. By incorporating Bellman optimality equation to get the Temporal Difference (TD) error and reduce the number of value functions from 2 to 1, the γ is a discount factor which ensures the sum converges and r_{t+1} is the reward of state s_{t+1} .

$$A(s_t, a_t) = Q_v(s_t, a_t) - V_\omega(s_t) = r_{t+1} + \gamma V_\omega(s_{t+1}) - V_\omega(s_t) \quad (110)$$

The policy function can be optimized by taking the gradient ascent with the partial derivative of the policy parameter θ for Eq. 109:

$$\theta \leftarrow \theta + \nabla_\theta \left[\frac{\sum_t^T \log \pi_\theta(a_t|s_t) A(s_t, a_t)}{T} + \varepsilon \sum_t^T \left(\sum_n^N \frac{-p(a_{n,t}|\pi_\theta) \log p(a_{n,t}|\pi_\theta)}{N} \right) \right]. \quad (111)$$

In essence, policy gradient updates the probability distribution of policy function $\pi_\theta(a_t|s_t)$ so that actions with higher expected reward $V_\omega(s_t)$ have a higher probability value under state s_t . The entropy of the policy function which is the third term of the right side in Eq. 10.9 is added to encourage the agent to explore, the $p(a_n)$ is the probability of action a_n in policy $\pi_\theta(a_t|s_t)$, and N is the total number of actions in each state.

Similarly, the loss of value function is the mean of squared TD error, and the value function is optimized by:

$$\omega \leftarrow \omega - \beta \nabla_\omega \frac{\sum_t^T A(s_t, a_t)^2}{T}. \quad (112)$$

Therefore, the A2C can learn and update parameters from each episode. As the number of episodes increases, it will eventually learn how to select the policy in the current state to ultimately generate the structure with the lowest energy.

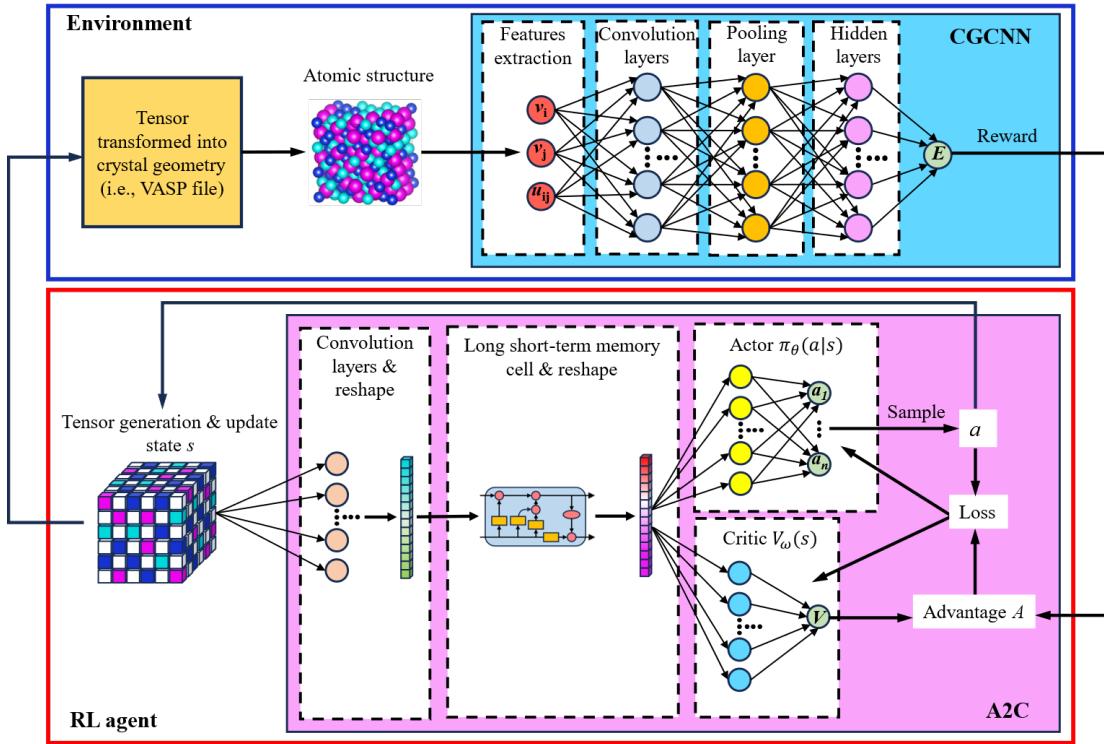


Fig. 20. Schematic of reinforcement learning (RL) model. RL agent generates structure with state s_t and selects action to generate new structure s_{t+1} based on current state s_t . The environment that contains crystal graph convolutional neural networks (CGCNN) interatomic potential predicts the total energy E of the generated structure and output the reward and update the RL networks.

Results and Discussion

Performance of CGCNN and A2C

Two different training datasets are used for two CGCNN interatomic potential training. The first model (Hereinafter referred to as model 1), similar to the CGCNN model training in TiAl/TiAlN design in CHAPTER 3, uses a total of 12,000 randomly generated $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures based on basic $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure as inputs and their corresponding total energies with no ionic update DFT calculations with PBE exchange-correlation functional as outputs, and the second one (Hereinafter referred to as model 2) uses a total of 763 randomly generated $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$

structures and their corresponding total energies by DFT fully relaxation with PBE exchange-correlation functional. The fractional coordinates of the atoms in any two structures are the same, except the lattice constants of the model 1 and the model 2 are 16.406 Å and 17.9 Å, respectively, which separately come from the pure Si lattice and approximate optimized lattice calculated by Vegard’s law [304]. Details of the CGCNN architecture are given in the “Methods”. Fig. 21a and Fig. 21b are the energy distribution histograms individually representing model 1 and model 2 training structures. Since there is no ion movement when calculating the energy of the training set of model 1, the energies are unstable, resulting in a more scattered energy distribution and higher energies in Fig. 21a. On the contrary, the energies in Fig. 21b in which the structures are fully relaxed are lower. A train-validation scheme is utilized to optimize the models. Each model is trained with part of the training data and then validated with part of the rest of the data, and the best-performing models in the validation set are selected and the test set evaluates the performance. Fig. 21c and Fig. 21d show the energy calculated by DFT versus predicted values by model 1 and model 2, respectively. The mean absolute errors (MAEs) of energy per atom (eV/atom) for both training and testing in model 1 and model 2 are 0.000402809 eV/atom and 0.001101831 eV/atom, separately. Here the energy predicted by model 1 is compared with that of the unrelaxed DFT calculation, and model 2 is compared with that of the fully relaxed DFT calculation. It can be seen that model 1 trained with energy without ionic update as output has lower error, which is reasonable because the structure has unchanged atoms and lattice constants after DFT calculation without ionic movement, which makes the only variable between input structure and output energy exist in different element arrangements. And it makes the unrelaxed structure and

corresponding energy have stronger correlation. Model 2 trained with original structure and fully relaxed energy, in addition to the change of element arrangement, the change of atomic position and lattice constant makes the variables between original structure and output energy increase, resulting in reduced correlation and greater error. However, for complex local atomic environments, this indirect simplified correspondence is necessary. Fig. 21e and Fig. 21f respectively shows the energies with fully relaxed versus the predicted energies of model 1 and model 2 of 161 structures randomly selected from the training set of model 1. Although the size of training set of model 2 is one order of magnitude smaller than that of model 1, its target is the energy after fully relaxation. So, even if most of the 161 structures are not in the training set of model 2, it can learn the local atomic environment from the relaxed energy and convolution process of CGCNN. The positive correlation between its predicted energy and the relaxed energy is larger than that of model 1, and its MAE is 0.001 eV/atom, which is similar to the result of Fig. 21d, indicating the robust prediction for new structures. It also illustrates the feasibility of using a small number of simplified atomic structures to train the machine-learned atomic potential energy. Therefore, the rest of this paper will adopt model 2 trained by fully relaxed energy as the CGCNN interatomic potential for predicting energy. To test the performance of A2C and DQN, the DQN method in is used to compare with A2C, in both methods, the model 1 as test CGCNN potential to predict the energy of each structure generated by RL agent.

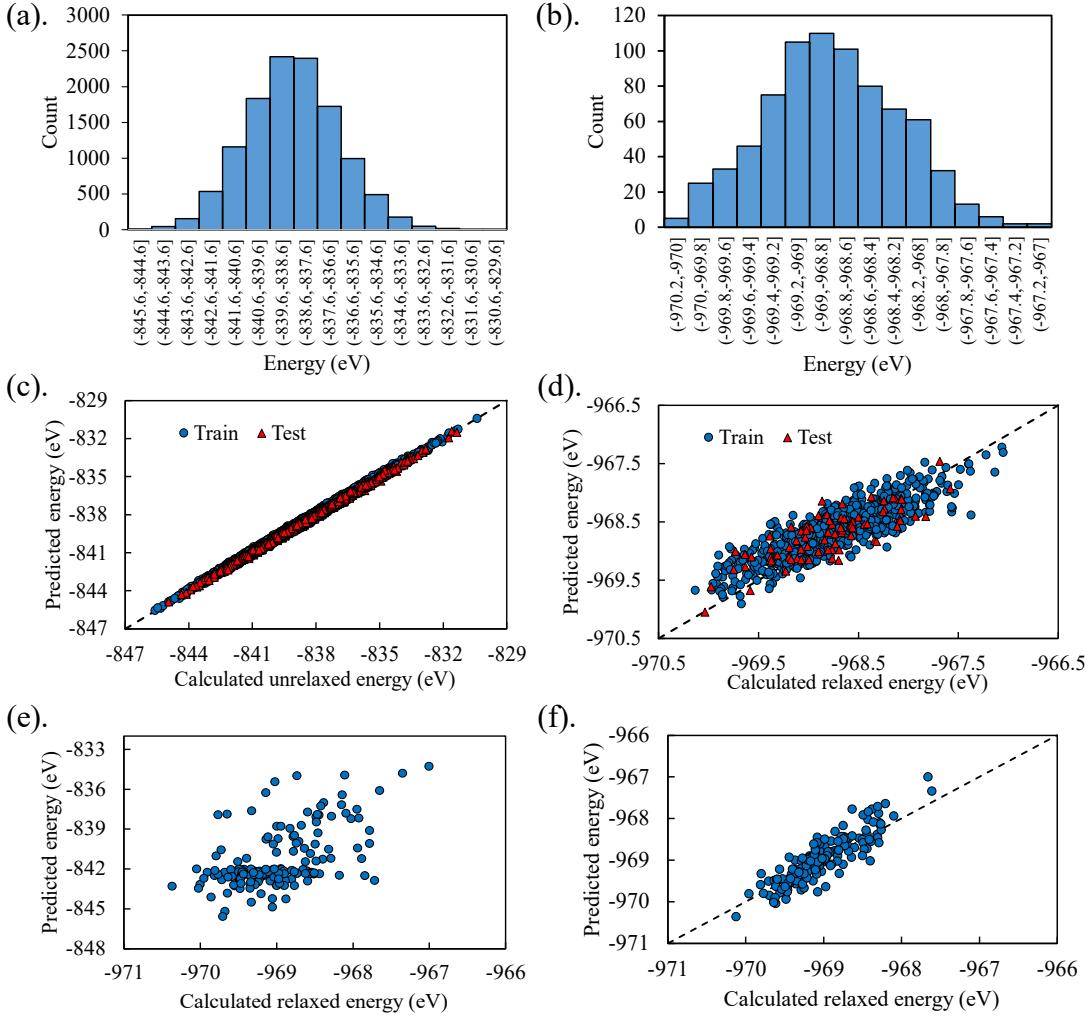


Fig. 21. Performance of CGCNN model under different training datasets. (a) Histogram representing the distribution of the energy of model 1's training set which is calculated by DFT with no ionic update of the structure. (b) The histogram shows the energy distribution of model 2's training set which is calculated by DFT with fully relaxation. (c) Prediction of model 1 trained by the energy without relaxation versus unrelaxed DFT calculation. (d) Prediction of model 2 trained by the energy with relaxation versus relaxed DFT calculation. The black dashed line in each plot represents the ideal fit

Fig. 22 shows the evolution of mean energy of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures generated by DQN and A2C with model 1 as interatomic potential. For easier comparison, the horizontal axis is the length of time consumed instead of the training episode and the black dashed line is the average energy without relaxation of training set in model 1. It can be seen that for the two models trained for five days, the value calculated by A2C is higher than that of DQN at the beginning, but as time goes by, the energy of the generated structure by A2C gradually decreases and is lower than that of DQN, while the value of DQN does not decrease but increases, even higher than the average energy (black dashed line) of 12000 random structures. In this task, since each structure has $434 \ 13 \times 13 \times 13$ tensor states that need to be stored by memory, it is also very memory-intensive for samples that only contain 100 structures in replay memory. In addition, the consumption of memory causes each episode to only use a small number of samples in the replay memory as batches to train the DQN. And since each structure has 434 states, DQN calculations are very time-consuming. In five days, DQN only trained 20,000 structures, while A2C, as an on-policy model that does not require a memory buffer, trained 200,000 structures and found the optimized structures that are much lower than any of the structures in the training set. Therefore, the rest of the article will use A2C as an RL method to study the energy and SRO of SiGeSn alloys.

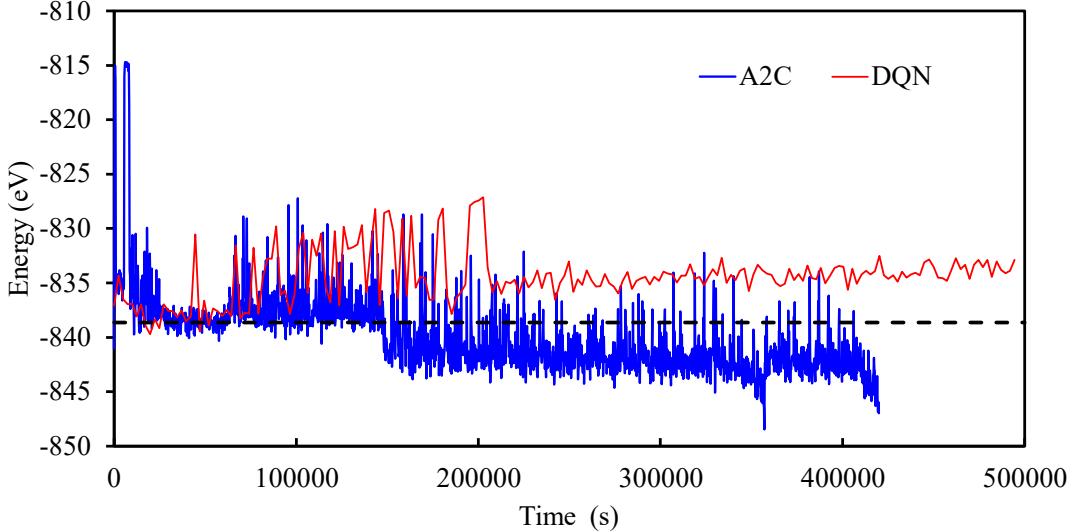


Fig. 22. The evolution of mean unrelaxed energy of generated structures by A2C and DQN models during training. The black dashed line is the average energy of 12000 random structures.

Energy Landscape and SROs of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$

The Metropolis MC method and A2C are separately combined with CGCNN to obtain the energetically favorable $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ configurations. Fig. 23a shows the energy evolved with Metropolis MC using CGCNN as potential (MC/CGCN), the initial structure is generated by randomly permute three chemical species over the sites of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure. A total of 5000 moves are conducted, and the predicted energy per 100 moves is plotted in Fig. 23a. In order to compare the predicted value and the DFT calculated value more intuitively, another line represents the corresponding energies calculated by fully relaxed by DFT is plotted as well. It can be seen that the energy of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ decreased drastically at the beginning. It usually indicates the generation of an SRO during rearranging the atomic configurations, which can form a structure with lower energy than the random structure. Intriguingly, the change trends of the two lines are similar, and both of them

gradually become stable and fluctuate within a certain energy which is well below the average energy (black dashed line) of the random $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures used for training. In addition, for structures with DFT-calculated energies lower than -972 eV, there is a systematic error between the CGCNN model and the DFT value causing an underestimation of prediction. This is mainly due to the diversity and size limitation of the training set. As shown in Fig. 21b, the lowest energy in the training set is -970.13 eV, which is much higher than the energy predicted in MC/CGCNN. Therefore, systematic errors will occur when extrapolating predictions for new structures with lower energy. Even so, the global error is less than 0.003 eV/atom. So, the similar change trends of the two lines also indicate that the CGCNN model using simplified and small training sets has a certain transferability.

To examine the evolution of SROs in $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ during MC/CGCNN simulation, six SRO parameters of first nearest neighbors in $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$, namely, Si–Si, Si–Ge, Si–Sn, Ge–Ge, Ge–Sn, and Sn–Sn, are calculated along the MC/CGCNN trajectory per 100 moves. The detail of SRO is given in the “Methods” section. Fig. 23b displays the variations of SROs for different types of nearest neighbors along the MC/CGCNN trajectory in Fig. 23a. In the beginning, all SROs change strongly, and their energies decrease rapidly. It shows the strong correlation between energy and structure. It can be seen that Sn-Sn and Ge-Sn bonds change most strongly and develop in opposite directions, with the number of Sn-Sn bonds decreasing and the number of Ge-Sn bonds increasing. As the number of steps increases, each SRO parameter deviates from the initial value to varying degrees and fluctuates within a certain range, among which Si-Ge, Ge-Ge, and Sn-Sn are positive, Si-Sn and Ge-Sn are negative, and Si-Si fluctuates around zero. To verify the

universality of this SRO distribution of low-energy structures in CGCNN potential, we used another structure, which is generated by the special quasirandom structure (SQS) [305] to make all SROs close to zero, as initial structure to run MC/CGCNN for 5000 steps, and the energy and SRO changes per 100 moves are shown in Fig. 23c and Fig. 23d, respectively. It can be seen that the results of these two simulations are similar, indicating the SRO distribution in MC/CGCNN is not a special case. As can be seen from Fig. 23b and Fig. 23d, the change trends of Si-Ge and Ge-Ge are opposite to each other, and Si-Sn and Ge-Sn also have such relationship. This opposite trend can be explained by the conservation of the total number of bonds corresponding to each element. When the number of Sn-Sn bonds is greatly reduced and finally stabilized within a certain range, the unbonded Sn atoms will bond with Si and Ge atoms and keep the total number of bonds constant, that is, when the number of Si-Sn bonds increases, the number of Ge-Sn bonds will decrease accordingly. This change can be reflected in SRO, so when the SRO of Sn-Sn bond fluctuates around 0.6, the sum of the SRO of Si-Sn and Ge-Sn will fluctuate around -0.6. Similarly, when the SRO of Ge-Sn fluctuates around -0.4, the remaining Ge atoms will bond with Si and Ge atoms and ensure that the sum of the SRO of Si-Ge and Ge-Ge fluctuates around 0.4. Fig. 23e shows the variation of the number of bonds for each type of nearest neighbors in MC/CGCNN simulation per 100 moves in Fig. 23a. The horizontal lines on the left represent the numbers of different bonds in a completely randomly arranged structure, and their values depend only on the composition of the alloy. As the MC/CGCNN simulation proceeds, the presence of SRO causes the number of different bonds to increase or decrease accordingly. Compared with Fig. 23b, the SRO and number of bonds are negatively correlated. When SRO is positive,

the number of bonds decreases, and vice versa. The degree of change in the number of bonds depends on the absolute value of SRO. The larger the absolute value, the greater the change. Fig. 23f shows the SRO polygon constructed from the averaged SRO parameters for each type of nearest neighbor calculated for all 5000 $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures obtained by 5000-step MC/CGCNN simulation starting from SQS random structure. The average SRO parameter α_{ij}^1 for the pair $i - j$ is labeled and depicted as a dot. The two hexagons separately represent a random distribution when $\alpha_{ij}^1 = 0$ and fully repulsion when $\alpha_{ij}^1 = 1$. Most types of pairs display deviations from the random distribution, and the Sn-Sn pair has the most significant deviation. Interestingly, the SRO in this work is a special SRO (S-SRO) which is different from the regular-SRO (R-SRO) and enhanced-SRO (E-SRO) reported in Ref. [288], where the Si-Ge and Si-Sn SROs are opposite to our results, and Si-Si SRO in E-SRO is also opposite.

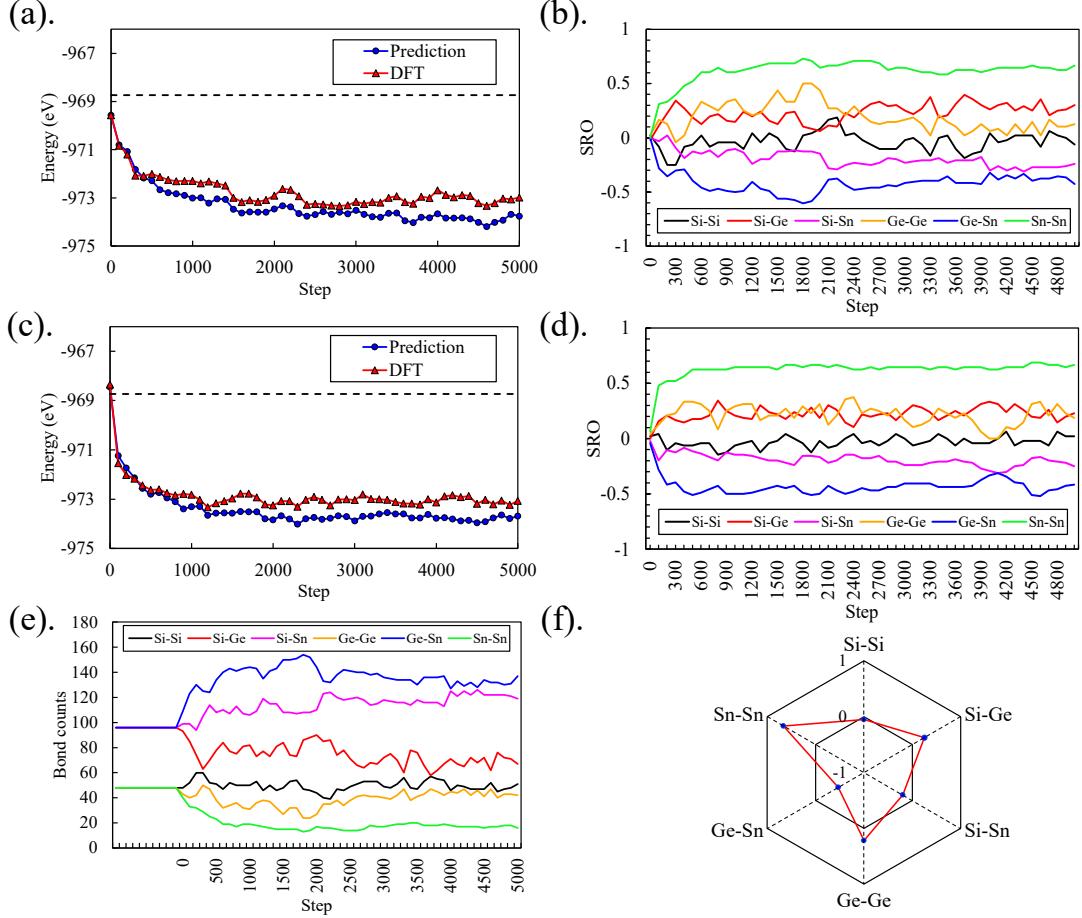


Fig. 23. The total energy fluctuates of MC/CGCNN trajectories and first-nearest SROs variation of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$, and mean energy versus episodes by A2C/CGCNN during training. (a) and (c) separately represent the energy fluctuation of MC/CGCNN trajectories with the initial structure by randomly permuting three chemical species over the sites of $\text{Si}_{0.125}\text{Ge}_{0.625}\text{Sn}_{0.25}$ structure and by SQS. (c) and (d) represent the SROs variation of MC/CGCNN trajectories in (a) and (c), respectively. (e) displays the variation of bond counts for each type of nearest neighbor. The number of each type of bond with a truly random 216-atom cell is shown as the horizontal lines, which are determined only by the overall alloy composition. (f) shows the SRO polygon constructed from the average SRO parameters α_{ij}^1 for each type of nearest neighbor of all $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structures in 5000 step MC/CGCNN.

To investigate the robustness of CGCNN and the presence of S-SRO, multiple MC/DFT simulations are conducted. Fig. 24a to Fig. 24f shows the variations of SRO for each nearest neighbor type and energies in three MC/DFT simulations with PBE exchange-correlation functional under different initial structures with 216 atoms. The initial structures in Fig. 24a, Fig. 24c, and Fig. 24e are SQS generated structure with random distribution, random mixture structure with random distribution [306], and S-SRO structure with low energy extracted from the MC/CGCNN, respectively. And their corresponding variations of energy are separately plotted in Fig. 24b, Fig. 24d, and Fig. 24f. For the random structures, similar to the MC/CGCNN simulation in Fig. 23a and Fig. 23b, the Sn-Sn and Ge-Sn SROs change intensely at the beginning, and the rest SROs will develop with the step goes and finally become to S-SRO, their energy changes also gradually decrease. Besides, in Fig. 24e and Fig. 24f, the variations of SROs and energies during simulation are small, meaning the structures with S-SRO have lower energy than others and thermodynamically favored. These results indicate the SROs predicted in MC/CGCNN are consistent with MC/DFT simulation and the S-SRO structure is prevalent in the low-energy region. What's more, by inspecting the variation of energy in each simulation, it is not difficult to see that for both MC/DFT simulations started from structures with random distribution, the lowest energy calculated in each simulation is higher than that of MC/CGCNN simulation in which the CGCNN model is trained by 610 random structures (610 DFT calculations), therefore, with the same amount of DFT calculations, the transferability of CGCNN model can predict a more energy-stable structure, thus reducing the simulation time.

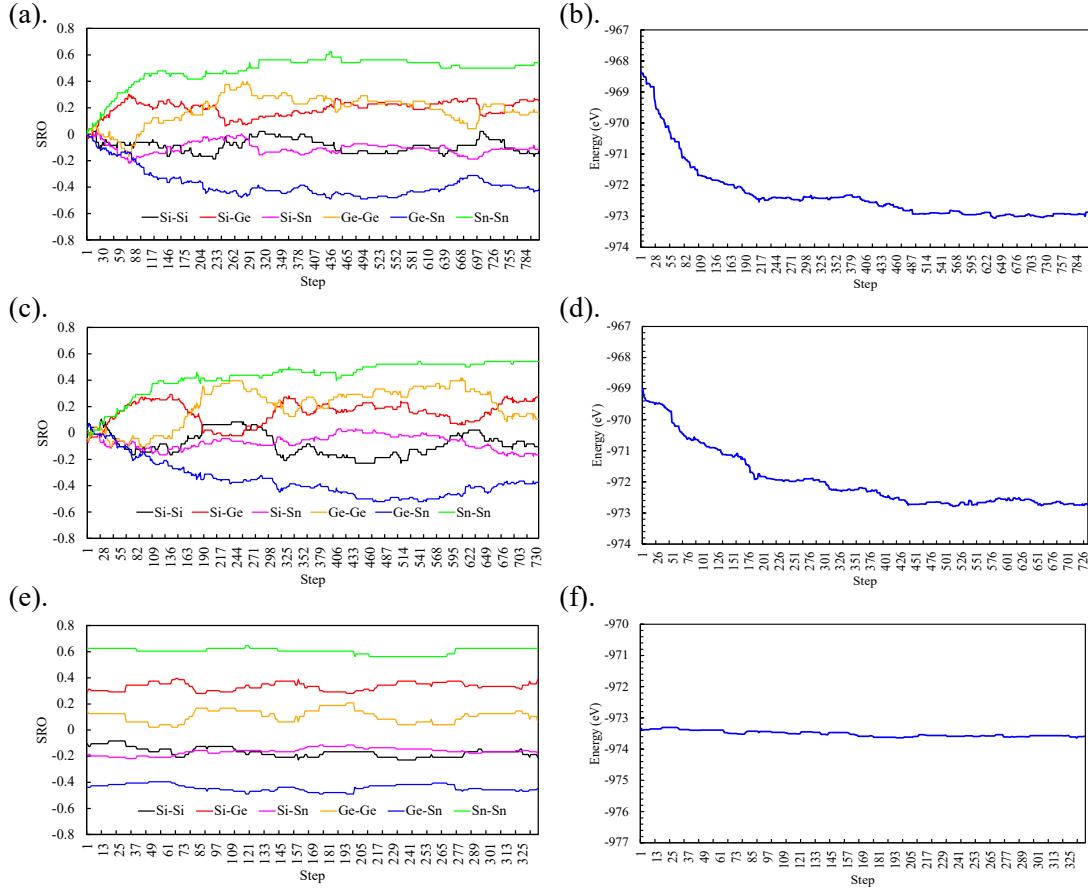


Fig. 24. Variations of SRO for each nearest neighbor type and energies in three MC/DFT simulations with PBE exchange-correlation under different initial structures with 216 atoms. (a) and (b) individually represent the SRO and energy variation during MC/DFT simulation started from SQS generated structure. (c) and (d) individually represent the SRO and energy variation during MC/DFT simulation started from random mixture structure. (e) and (f) individually represent the SRO and energy variation during MC/DFT simulation started from S-SRO structure with low energy extracted from the MC/CGCNN.

Besides, two MC/DFT simulations separately using PBE and LDA as exchange-correlation functional are performed for the same initial 64-atom $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ structure by 2078 steps to inspect the size effect and the influence under different

functionals. Fig. 25a and Fig. 25b separately show the variations of SRO for each nearest neighbor type in MC/DFT with PBE and LDA exchange-correlation functional, and their corresponding variations of energy are plotted in Fig. 25c and Fig. 25d, respectively. Table 3 shows the average SRO for each nearest neighbor type of Fig. 25a and Fig. 25b. We can see the SROs in Fig. 25a are similar to those in Fig. 24, revealing the S-SRO is energy-favored with PBE in both sizes. In Fig. 25b, the SROs are similar to S-SRO except the Si-Ge and Si-Sn are oscillating around zero, although there exist the S-SRO structures during simulation, which means the Si-Ge and Si-Sn bonds in the simulation with LDA exchange-correlation functional neither repulsive nor attractive, also indicates the different exchange-correlation functionals may result in different outcomes. Furthermore, by comparing the SROs in Table 3 to the R-SRO and E-SRO polygons of $\text{Si}_{0.125}\text{Ge}_{0.625}\text{Sn}_{0.25}$ in Ref. [288], it can clearly see that all have strong Sn-Sn repulsive interaction to protect phase separation from Sn atom cluster, which also consist with GeSn and SiSn alloys [296, 297]. Besides, both SROs in this work have negative Si-Si SRO, which are lower than R-SRO and opposite to E-SRO. For Si-Ge SROs, in Table 3, especially for simulation with PBE exchange-correlation functional, are opposite to that in R-SRO and E-SRO, and the Si-Sn SROs in Table 3 are close to zero or negative values, which are different from the positive Si-Sn SROs in R-SRO and E-SRO, indicating the attractive interaction of Si-Ge and repulsive interaction of Si-Sn in both simulations are suppressed even have the opposite effect. Since the repulsive interaction between Si and Sn and the attractive interaction between Si and Ge have been reported by other works [288, 307-309], some also showed that Sn atoms prefer to be located at the second nearest neighbor of Si atoms [307, 308]. The results seem inconsistent with others, but the

structure studied in this paper is equimolar structures, while other studies do not involve equimolar structures. Therefore, this SRO relationship may be destroyed for equimolar structures. To further verify our conjecture, we carefully inspected the relationship between different bond numbers in MC/DFT simulations of the other ten SiGeSn alloys with different contents in Ref. [288]. Interestingly, when Ge and Sn are both 25%, the number of bonds Si-Ge, and Si-Sn are equal and oscillate around the values of their corresponding initial random distributions, which indicates that the number of bonds of Si-Ge and Si-Sn keep constant during simulation, and this is similar to the SRO of simulation with LDA exchange-correlation functional, therefore, the first nearest-neighbor SROs of Si-Ge and Si-Sn, unlike other compositions, are suppressed under this component. If the content of Ge further reduces to 12.5% and the content of Sn keeps 25%, which is larger than that of Ge, when we check the variation of bond count of Si-Sn along MC/DFT simulation, we can find that the bond counts of Si-Sn and Si-Ge are individually larger and smaller than that of initial random distribution, which indicates the negative SRO of Si-Sn and positive SRO of Si-Ge, and it is consistent with the simulation with PBE exchange-correlation functional. Therefore, the similarity of the first nearest-neighbor SROs of Si-Ge and Si-Sn between this case and ours may specify a regularity between the changes in the first nearest-neighbor SRO and the changes in the composition. Especially under conditions when the content of Sn is equal to or larger than Ge, which makes the SRO of Si-Sn reduce and the SRO of Si-Ge increase and leads to the presence of S-SRO or SRO listed in Table 3. Here, no other SRO coexists in both MC/DFT simulations, which may be due to the composition being out of the region of indirect-to-direct

transition which significantly overlaps the compositions of coexistence of R-SRO and E-SRO [288, 310].

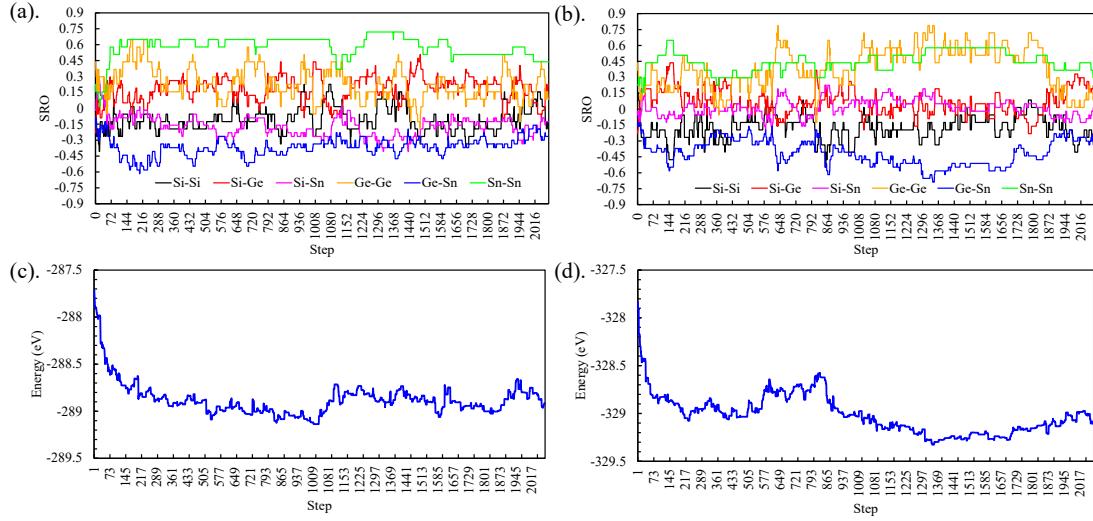


Fig. 25. The variations of SRO for each nearest neighbor type in MC/DFT with PBE (a) and LDA (b) exchange-correlation functionals, and the variation of energy corresponding to PBE (c) and LDA (d) exchange-correlation functionals.

Table 3. The SRO for each nearest neighbor type of two MC/DFT simulations with PBE and LDA exchange-correlation functionals.

SRO	Si-Si	Si-Ge	Si-Sn	Ge-Ge	Ge-Sn	Sn-Sn
MC/DFT with PBE	-0.103	0.178	-0.169	0.233	-0.365	0.580
MC/DFT with LDA	-0.174	0.069	0.012	0.390	-0.412	0.448

Besides the first nearest-neighbor SRO, SROs of second and third nearest neighbors are also plotted. Fig. 26 shows the SRO of atomic species j around an atom of type i in different neighboring shells v under different structures. Here, the SRO is calculated for the random distribution structures generated by SQS (row 1) and random mixture (row 2), the low-energy structure with S-SRO first nearest neighbor

obtained by MC/CGCNN simulation in Fig. 23a (row 3), and the lower-energy structure with S-SRO first nearest neighbor obtained by MC/DFT simulation in Fig. 24c (row 4). Clearly, the SROs of the random distribution structures generated by SQS and random mixture are close to zero, while for MC/CGCNN and MC/DFT structures, in addition to the similar S-SRO characteristics of the first nearest neighbor, their second and third nearest neighbors are also similar. For the S-SRO structure simulated by MC/DFT with lower energy, the SRO of Si at the first nearest neighbor position of the Si atom and the SRO of Ge at the second nearest-neighbor position of the Ge atom are further reduced. Like the first nearest neighbor of the E-SRO structure, the first nearest neighbor of Sn-Sn has strong repulsion, and the first nearest neighbor of Ge-Sn has a strong attraction, while the first nearest neighbors of Si-Si, Si-Ge, and Si-Sn are attractive, repulsive, and attractive respectively, which are opposite to E-SRO. By calculating the first three nearest neighbors' local configurations, for different types of central atoms in S-SRO, their first three nearest neighbors are more likely to form local bonding structures of Si-Si-Sn-Ge, Ge-Sn-Ge-Sn, and Sn-Si-Si-Sn. Although different from the Si-Ge-Sn triplet local configuration and strong Si-Sn repulsion in E-SRO, in which the Si-Ge-Sn triplet local configuration and strong Si-Sn repulsion of Si-Ge-Sn both reduce the appearance of Sn-Si-Sn, which is the major unfavorable configuration identified in SiSn [288, 297], the local bonding structure of Sn-referenced in S-SRO structure is Sn-Si-Si-Sn, which also does not tend to form the Sn-Si-Sn triplet local configuration. In addition, recent studies on SiGeSn alloys with low Sn content (less than 16 %) have also found that the Si-Si first nearest-neighbor configuration is generated such as through a Si-Si-Sn nearest neighboring configuration to reduce the lattice dilation induced by nearby Sn

atoms [311]. This is consistent with the Si-Si-Sn-Ge local bonding structure of Si-referenced and explains the rationality of the S-SRO structure.

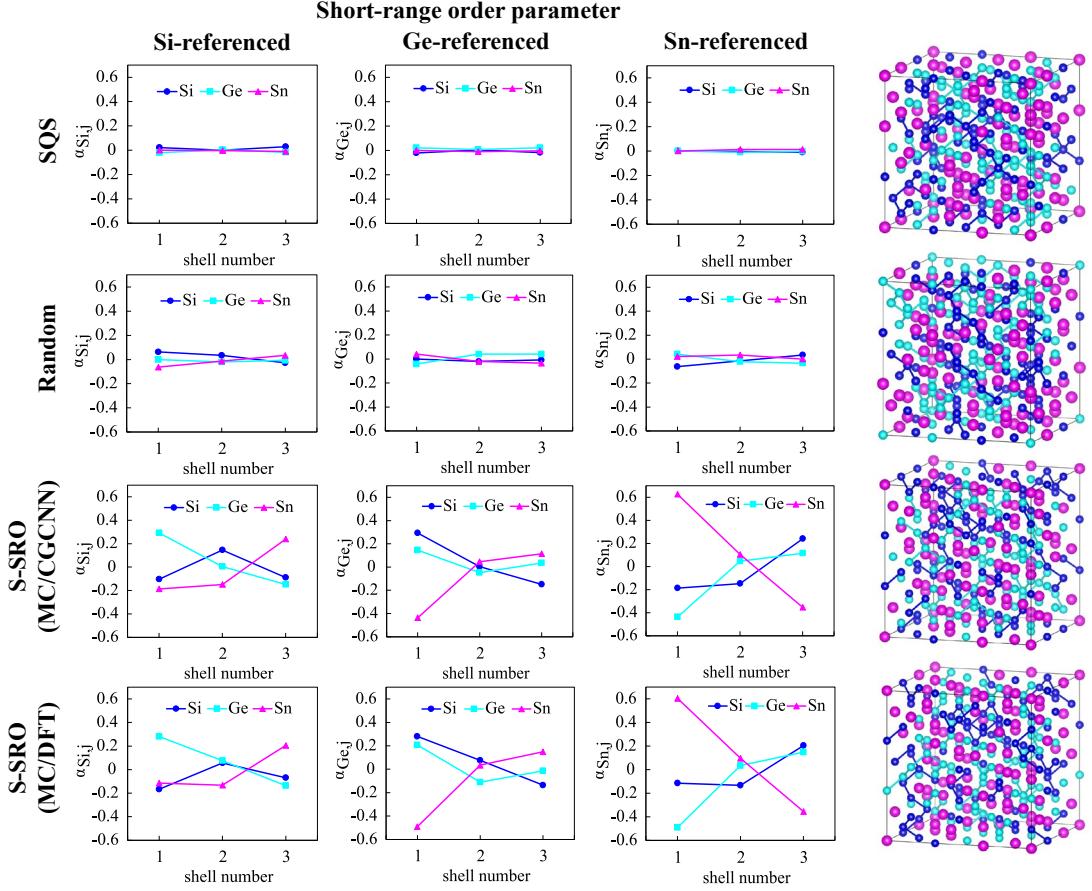


Fig. 26. Short-range order parameters α_{ij}^v of SQS (row 1), random mixture (row 2), low-energy first nearest-neighbor S-SRO structure predicted by MC/CGCNN (row 3), and low-energy first nearest-neighbor S-SRO structure predicted by MC/DFT (row 4).

Interpretability of CGCNN and RL Model

Physical interpretation of machine learning is crucial because it can provide insights into the correlations between different inputs or between inputs and outputs. In CGCNN, local environment representations are analyzed by using principal component analysis (PCA). The local environment representation for each structure is a 216×64 dimensional matrix after passing the convolutional part, which contains

the atoms and bonds information. After flattening each matrix, PCA will reduce the vector to two dimensions which have highest variations for these vectors. Fig. 27a and Fig. 27b show the PCA of the local environment representation of 896 structures containing training data, testing data, and MC/CGCNN generated structures corresponding to SROs and energy, respectively. The E-SRO-like, R-SRO-like, and the S-SRO structures are labeled in Fig. 27a. For training and testing data, because they are random structures, they have high energies and are clustered in the left region. Combining the relationship between SROs and energies in two figures, it can be seen that only a small number of the 763 randomly generated structures used for training and testing (less than 70 structures in total) have S-SRO, E-SRO-like, and R-SRO-like features and these structures are located in the relatively low energy area of the random structures. The remaining structures have other SROs and are concentrated in the high-energy area on the left side of the figure. From Fig. 27a, the structures with the same SRO are not clustered together. This is because although the overall SRO of these structures is the same, the value of each bond pair is different, that is, the structure is different. This shows that the local environment representation also has similar change under the continuous change of SRO value and energy, so the local environment representation reflects the characteristics of the structure to a certain extent. This feature is more obvious in the generated lower energy structure. The right parts of the two figures show the local environment representations of the structures with different SROs generated by MC/CGCNN. As the SRO value of each bond pair changes, the structures with S-SRO and labeled “other” structures with similar SRO have lower energy than the structures with E-SRO-like and R-SRO-like patterns, and the structures with similar bond pair SROs are clustered together and have similar

energy. The local environment representation trajectories of generated S-SRO and its similar SRO structures are different from those of E-SRO like and R-SRO like, and there is a certain gap between the two, which also shows that the local environment representation is different for different SRO structures. It is worth noting that in the trajectory of energy change, S-SRO and labeled “other” structures have similar trajectories, while E-SRO-like and R-SRO-like structures have similar trajectories. This is because S-SRO and the labeled “other” structures have similar SROs except for Si-Si SRO, and their Si-Si SRO is near zero. Similarly, in E-SRO-like and R-SRO-like structures, except for Si-Si SRO, the change trends and values of other SROs are similar, and the values of Si-Si SRO in these two are close to zero. Therefore, according to these two figures, it can be seen that the local environment representation of the CGCNN model contains structural information, and this information comes from the convolutional process in CGCNN, with each step of the convolution process, each atom passes its own information to the next neighboring atom, so that the local environment contains information about many-body interactions, and this many-body information will compare with the training set and based on their similarity to output energy.

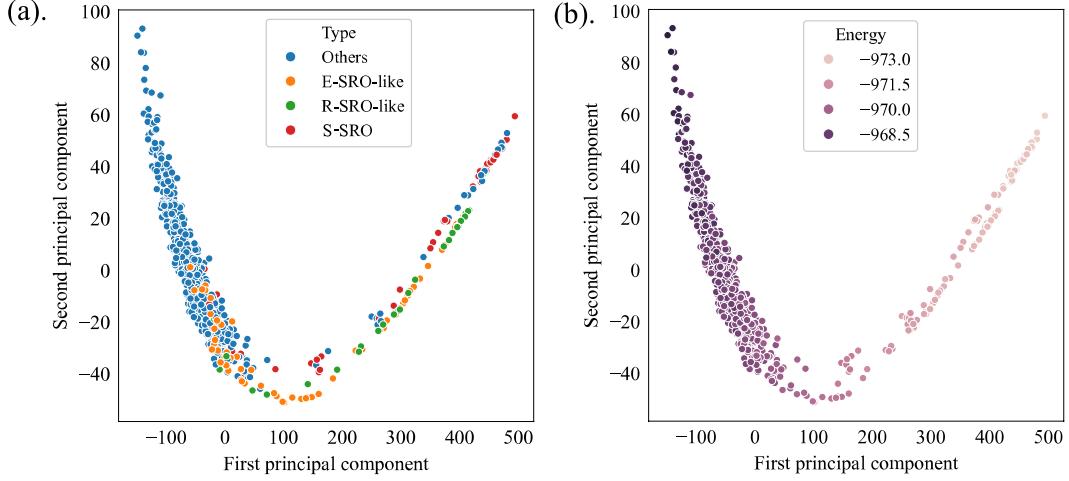


Fig. 27. PCA of the local environment representation of 896 structures containing training data, testing data, and MC/CGCNN generated structures corresponding to SROs (a) and energy (b), respectively.

Summary

In this chapter, we mainly focus on how to use CGCNN model combined with MC or RL to efficiently and accurately predict the low-energy structures of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ MEA. Compared with first-principles MC simulation, CGCNN can use the same amount of high-energy data as MC/DFT to train and obtain lower energy structures than MC/DFT. Besides, low-energy $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ displays a different SRO compared to other compositions, in this SRO, the Si-Sn repulsive interaction is suppressed, and by inspecting the first-three nearest neighbors, the low-energy structures may use a different local configuration to reduce the appearance of Sn-Si-Sn local configuration. This indicates that the SRO in SiGeSn MEA may be different under different compositions.

CHAPTER 5

INTERPRETABLE ENSEMBLE LEARNING FOR MATERIALS PROPERTY PREDICTION WITH CLASSICAL INTERATOMIC POTENTIALS

Introduction

In recent years, density functional theory (DFT) and molecular dynamics (MD) simulations have been studied and applied extensively in materials multiscale modeling [312]. For example, the calculations of energy and forces of materials across different scales have been achieved by using these simulations [313-315]. Current widely used simulation methods including Kohn-Sham density functional theory (KSDFT) [3, 4] and MD simulations with classical interatomic potentials [7, 29, 30, 316-322] have demonstrated high performance in predicting the formation energy and elastic modulus of materials. However, both methods have their own limitations. KSDFT is computationally demanding and typically restricted to systems containing only a few hundred atoms, while MD can be used in larger systems but is limited in accuracy due to the empirical nature of interatomic potentials.

To solve the limitations of KSDFT and MD, machine learning (ML) models [323, 324] such as neural network potential (NNP) [23, 325], Gaussian approximation potential (GAP) [24], spectral neighbor analysis potential (SNAP) [25, 326], and moment tensor potential (MTP) [26] have been proposed to accurately predict energy and forces of crystals and molecules. They use atomic species and nuclear coordinates to build descriptors (also called “fingerprints”), which are invariant under permutations among the same elements, and isometric transformations of rotations, as features to be fitted by a chosen regression model [23, 327]. However, these

descriptors need to be designed meticulously to satisfy the restrictions and the complex transformations thus making it difficult to explain the models [2, 127].

To get more generalized descriptors, graph networks, which represent atoms and bonds as nodes and edges, respectively, combined with convolutional neural networks have received significant attention, since convolutional neural networks can automatically find the important features compared to descriptor-based models [19]. Several graph convolutional neural networks such as generalized crystal graph convolutional neural networks (CGCNN) [127], SchNet [132], MEGNet [124] and atomistic line graph neural network (ALIGNN) [247] have been proposed. They are straightforward to be adopted and suitable for both crystals and molecules, however, these descriptors have complex configurations that contain a series of operators and hidden layers, and their fitting process is time-consuming due to the high data requirements and the regression function contains a large number of parameters to be fitted in the neural network [328].

Compared to graph-network-based potentials, symbolic regression is a faster method to build interatomic potentials by using genetic programming to find a function that accurately expresses interatomic potentials from a set of variables and mathematical operators [149, 159, 329, 330]. But symbolic regression also has some limitations. For example, the expressions in the hypothesis space must be simple and have a significant effect on potential energy, and this model cannot learn complex terms that involve bond angles.

Besides, for general ML potentials, transferability, which describes the ability of a model to correctly predict the property of an atomic configuration lying outside the training dataset, is limited. Consequently, physically informed neural networks (PINN)

are proposed to improve the transferability of unknown structures [169, 331, 332] by combining a general physics-based interatomic potential with a neural-network regression. PINN achieves this by optimizing a set of physical-meaning parameters of a physics-based interatomic potential from trained neural networks, and then feeding them back to improve the accuracy of the original physics-based interatomic potential. However, this method encounters a similar obstacle to the graph networks mentioned above, which is the time-consuming fitting process in the neural network resulting from a large size of data and numerous parameters.

As the molecular-dynamics simulation database gradually improves, such as the force-field database of NIST JARVIS which contains properties like formation energy and elastic constants calculated by different classical potentials, the force-field database can be the potential input for machine-learning model [333, 334]. In this study, we present a regression-trees-based ensemble learning approach that efficiently predicts the formation energy and elastic constants of carbon allotropes with a small size of data calculated by classical potentials. We use carbon allotropes as an example to evaluate the performance of our model since carbon is one of the fundamental elements on Earth [335] these carbon allotropes have a variety of physical properties while being applied widely in cutting and polishing tools [336], superlubricity [337], solar thermal energy storage [338], etc. Therefore, understanding of the physical properties of carbon allotropes plays a significant role in both scientific research and engineering applications. We begin by extracting the structures of carbon allotropes from the Materials Project (MP) [339], and compute their formation energy and elastic constants using MD simulations with nine different classical interatomic potentials via the Large-scale Atomic/Molecular Massively Parallel Simulator

(LAMMPS) [340]. Then use these computed properties as features and corresponding DFT references as targets to train and test four different ensemble learning models [341-344] consisting of regression trees [345]. In general, the performance of ensemble learning models is better than that of nine classical potentials, and based on feature importance, ensemble learning can find the accurate features and use them to improve the precision of prediction.

Methodology

Regression Trees of Ensemble Learning

Regression trees, a type of decision tree, are used to predict outputs consisting of numerical values instead of categorical targets. They are also the base estimators in ensemble learning (the tree structures in Fig. 29). Fig. 28 illustrates a regression tree that has seven nodes in total. The tree starts from the top node, and each node contains sorted samples and will be split into two subsets based on the criteria (thresholds) for the features until the terminal condition is reached. The blue nodes are parent nodes, they have two subsets called children. The green nodes are end nodes, representing numerical outputs that are decided by the targets. In scikit-learn, the optimized version of Classification and Regression Trees (CART) [345] is used. This algorithm determines how to divide the sorted samples by trying different thresholds and calculating the MSE at each step. In this study, the feature vectors $x_i \in R^n$ and target vector $y \in R^k$ are properties calculated by classical interatomic potentials and corresponding DFT reference, respectively. Where subscript i represents indexes of different materials, superscript n shows the number of input variables (the number of classical interatomic potentials), and superscript k is the total number of materials. We denote Q_m as the dataset at node m with N_m samples, Q_m^{left} and Q_m^{right} as the children

of Q_m , with N_m^{left} and N_m^{right} as the number of samples of these children. These children will split Q_m into two parts using a threshold. The quality of the split of node m is calculated by minimizing the weighted average of impurity.

$$G(Q_m) = \frac{N_m^{left}}{N_m} H(Q_m^{left}) + \frac{N_m^{right}}{N_m} H(Q_m^{right}) \quad (113)$$

Where H is loss function (such as MSE), for example, at node m , the MSE of its left child Q_m^{left} is given by:

$$\bar{y}_m = \frac{1}{N_m^{left}} \sum_{y \in Q_m^{left}} y \quad (114)$$

$$H(Q_m^{left}) = \frac{1}{N_m^{left}} \sum_{y \in Q_m^{left}} (y - \bar{y}_m)^2, \quad (115)$$

here, \bar{y}_m is average value of target at node of Q_m^{left} . By recursing the Q_m^{left} and Q_m^{right} , the weighted average of impurity changes as well, and selects the threshold that minimizes the impurity G as the node m . Repeat and do the same steps for each node until the terminal condition is reached, and finally, a trained regression tree is obtained.

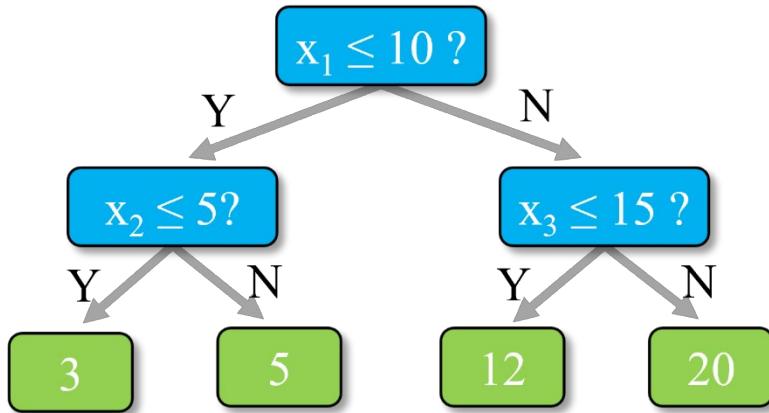


Fig. 28. The schematic of a regression tree illustrates the blue nodes (parent nodes) are split into two subsets (children) based on the threshold for the features until the terminal condition is reached. When using the tree to predict, the trained regression

tree will depend on the inputs and the thresholds to select the children, finally, one green node (output) will be decided.

Bagging and Boosting Methods

Bagging and boosting methods shown in Fig. 29 are used to achieve better performance than a single regression tree in this work. In bagging methods, several regression trees are trained independently by their own subsets in which the data can be chosen more than once. The final prediction is obtained by averaging the predictions [341] of all individual regression trees. On the contrary, the regression trees in the boosting method are generated sequentially and each regression tree has limited depth and is related to the previous one. Instead of averaging the outputs of all regression trees, the final prediction is calculated by calculating weighted median [342] of the predictions of all regression trees or summing predictions of all regression trees up [343].

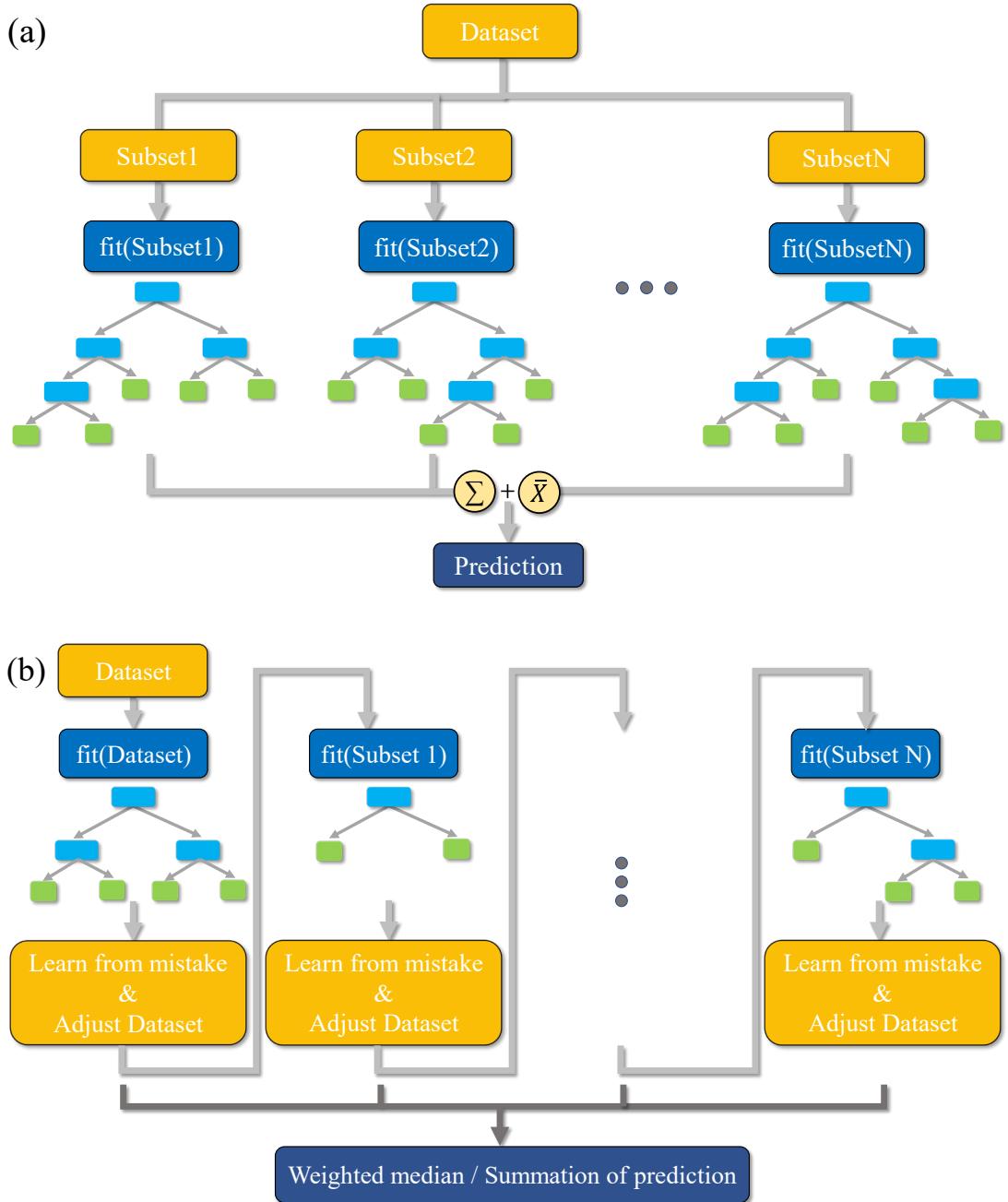


Fig. 29. Configurations of two regression-tree based ensemble learning models: bagging (a) and boosting (b). (a). In bagging methods, several regression trees are trained independently by their own subsets in which the data can be chosen more than once, and the final prediction is obtained by averaging the summation of all regression trees' predictions. (b). The boosting method is generated sequentially, and each regression tree is related to the previous one. The final prediction is weighted median

of all predictions of regression trees or summation of all predictions of regression trees.

Data Collections

For the formation energy, 58 carbon structures and 335 Si-O structures and their DFT reference are extracted from the MP database. And use the nine classical potentials available for carbon elements in LAMMPS to calculate the energy minimization of each structure to get the formation energy per atom of each structure, then the energy above hull in the Materials Project database is used and extracted the structure with a value of 0 as a reference, and the nine-potential-calculated values under this structure are used as their respective reference to obtain the energy above hull of all structures and use these values as the input feature value. For elastic constants, 20 out of 58 carbon structures' DFT reference are used due to the absence of DFT reference and removal of unstable or erroneous calculations [346]. For the features of each structure, use the same nine potentials to calculate the elastic constants at 0 K with the LAMMPS. The 21 elastic constants calculated by each potential are used as the input features.

Results

Ensemble Learning Framework for Properties Prediction of Carbon Materials

Fig. 30 illustrates the schematic of the ensemble learning framework. Firstly, carbon structures are extracted from the MP database. Then, the formation energy and elastic constants of each structure are calculated by MD with nine classical interatomic potentials, including the analytic bond-order potential (ABOP) [347], adaptive intermolecular reactive empirical bond order potential (AIREBO) [318], standard Lennard-Jones potential (LJ) [322], AIREBO-M [319] potential that replaces

the LJ term with a Morse potential in AIREBO potential [348], environment-dependent interaction potential (EDIP) [347], long-range carbon bond order potential (LCBOP) [321], modified embedded atom method (MEAM) [29], reactive force field potential (ReaxFF) [7], and Tersoff potential [349]. The training dataset is composed of these properties and corresponding DFT reference collected from the MP database, and encoded into feature vectors x_i and target vectors y_i , respectively. For the formation energy, 58 carbon structures and their DFT reference are extracted from the MP database, and for the elastic constants, 20 out of 58 carbon structures' DFT reference are used due to the absence of DFT reference and removal of unstable or erroneous calculations [350]. Next, the regression-trees-based ensemble models will be trained based on these vectors. We select regression-trees-based ensemble learning models for the following reasons. First, compared to neural networks, regression trees are white-box models which make the models and outputs easy to understand and interpret. Second, as non-linear models, regression trees have better performance than classical linear regression and neural networks methods when dealing with small-size data and highly non-linear features. Third, to mitigate locally optimal decisions of regression trees, ensemble learning combined with the predictions of several regression trees will be deployed to improve robustness over a single regression tree. Last, for multi-target problems such as the prediction of elastic constants, the ensemble learning method can learn the correlations of elastic constants and output multiple targets at once. Here, different regression-trees-based ensemble-learning methods implemented in the Scikit-Learn package [351], including bootstrap aggregation (bagging) [352] and boosting [342], are used to build simple, fast, and interpretable models. Details of the architectures and methods of regression trees and

ensemble learning are given in the “Methods” section. For the new carbon structures, we can calculate the same properties by using MD with the nine potentials and feed these calculated properties into the trained model with the smallest mean absolute error (MAE) during testing, as shown in Eq. 116, to predict the properties of new structures.

$$MAE = \frac{\sum_{i=1}^n |y_i^{pre} - y_i|}{n} \quad (116)$$

Where y^{pre} is prediction of model, y_i is value of reference, and n is the number of samples.

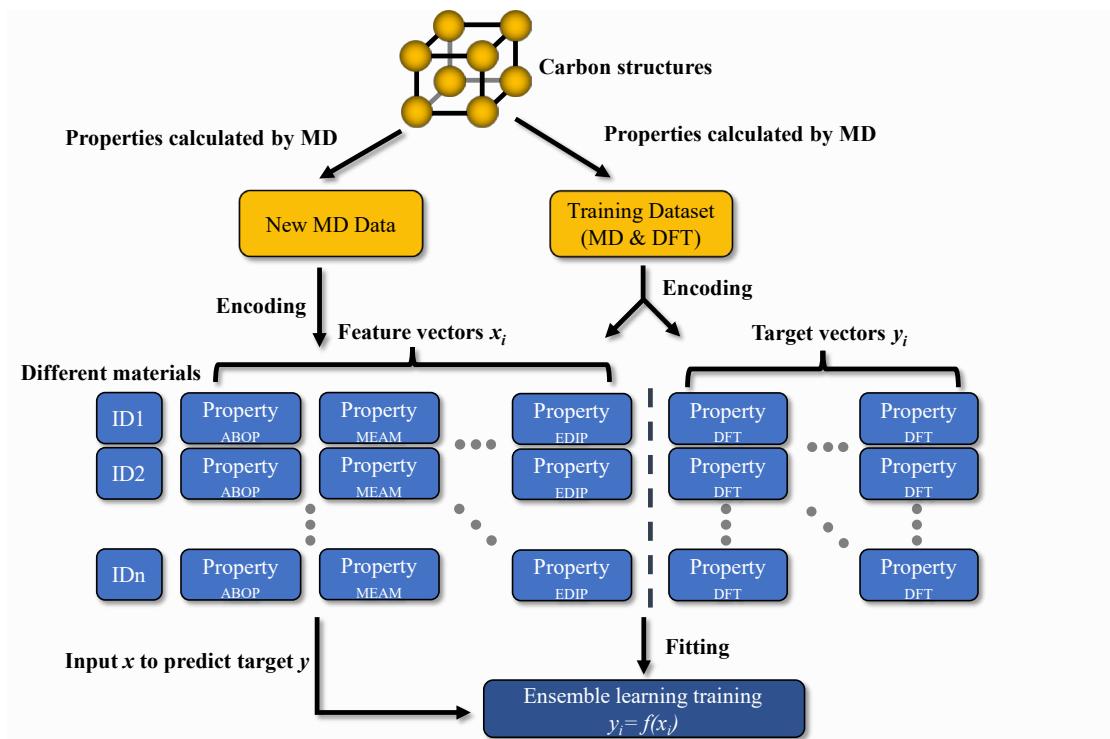


Fig. 30. Ensemble learning framework for properties prediction of carbon materials.

The properties of carbon structures are calculated by MD and use these calculated values and DFT references as input to train the ensemble learning model. To get the same properties of new carbon structures, first calculate the properties of new structures by using MD, then put these values into the best trained model to get final

predictions.

Formation Energy of Different Carbon Structures

Fig. 31 shows the formation energies of 58 carbon structures calculated by 9 different interatomic potentials. In comparison to the DFT reference, it is easy to notice that different potentials yield varying degrees of errors. LJ potential exhibits the highest error since it only considers two-body interactions. But multi-body interactions play a considerable role for solid phases like crystals. Under this condition, LJ potential omits some crucial factors hidden behind the carbon structures. MEAM can be utilized in carbon materials which have covalent interactions. MEAM potential used in this study, however, only considers the interactions among the first nearest-neighbor atoms [353], which causes high error. For the rest potentials, they are all bond-order-potentials which can handle chemical reactions better by incorporating more physical based parameters. Tersoff potential is one of the most common potentials for studying covalent-bond structures, but it has weak performance for diamond-like carbon (tetrahedral amorphous carbon) due to the interaction of short-range, which lead to a graphitic density similar to or exceeds that of diamond [354]. As shown in Fig. 31, the structures between mp-66 and mp-1008395 are diamond-like carbon, and the formation energies calculated by Tersoff are higher than those of DFT calculations. EDIP and ABOP include the effects of pi bond empirically, which are more accurate than Tersoff. Additionally, AIREBO and AIREBO-M are proposed based on the Tersoff potential, but they have more physical-based parameters such as pi bond and nonbonded interaction, resulting in lower MAEs (Fig. 33). Conversely, the values calculated by LCBOP (purple dotted line) and ReaxFF (lime dotted line) have the smallest error among the potentials since

they can handle dynamic bonding processes by considering much longer-range bond order relation. From Fig. 31 and Fig. 33, it can be observed that the accuracy of different potentials fluctuates across different structures. For instance, although LCBOP has the most accurate calculation of formation energy for 20 out of 58 structures, it may have larger errors for certain structures like mp-57002, mp-1244913, and mp-1095633, compared to ReaxFF, EDIP, and other potentials. Therefore, ensemble learning is a good choice to learn such complex input features.

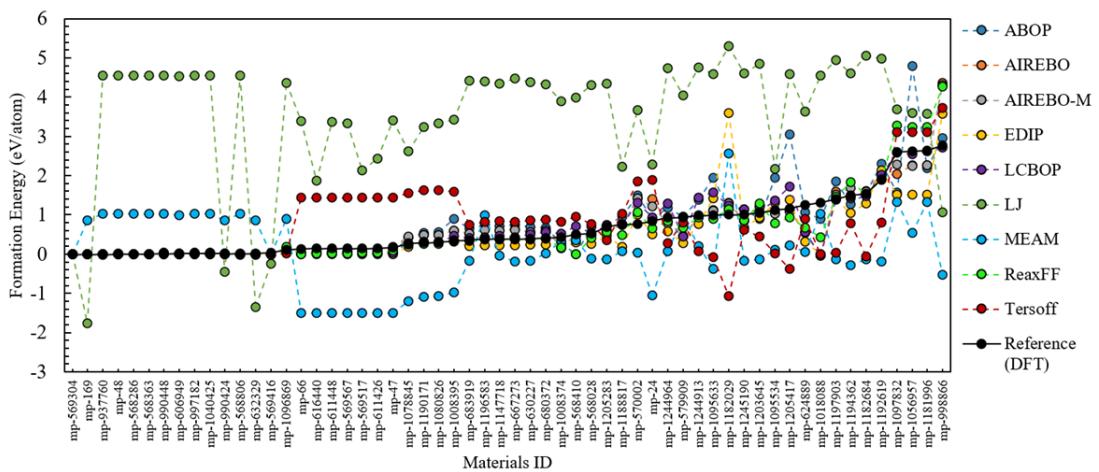


Fig. 31. Formation energy of carbon allotropes calculated by 9 classical potentials and DFT is used as the reference. The x-axis represents different carbon structures ID of MP database. Black line represents the DFT reference. The values calculated by the LJ (green dotted line) potential have the highest error, whereas the values calculated by the LCBOP (purple dotted line) potential have the lowest error.

Four different ensemble-learning methods are deployed, namely RandomForest (RF), AdaBoost (AB), GradientBoosting (GB), and XGBoost (XGB), to evaluate their performance. Because the selection of model hyperparameters such as the number of regression trees (estimators), learning rate, and depths of regression trees will occasionally cause overfitting which can influence the performance of the models, the

hyperparameters must be set manually before training to avoid this problem. To tune the ensemble-learning hyperparameters, a grid search in combination with 10-fold cross-validation is applied. Specifically, 80% of the data set is used for tuning hyperparameters, which are divided into ten equal parts to do 10-fold cross-validation. The best combination of hyperparameters, which has the smallest MSE, will be extracted as the final optimized hyperparameters. After tuning, each method with optimized hyperparameters is trained and tested by 10-fold cross-validation individually. Fig. 32 shows the prediction versus reference of different models by using 10-fold cross-validation, where the black dashed lines represent the indicators of the accuracy, the closer the data points are to the dashed lines, the more accurate the prediction. From Fig. 32a to Fig. 32d, most parts of the data points in the RF model are clustered around the dashed line, indicating that the RF has the slightest error and the average R^2 is 0.82. For the prediction of high formation energy, AB and GB models have better performance than RF and XGB models, yet it is more dispersed for the other points. However, for RF, AB, and GB models, all of them have a weak prediction for the orange point (mp-1018088 in Fig. 31) shown in Fig. 32. The reason may be due to the overall weak performance of the 9 potentials for this structure, as seen in Fig. 31. It is worth noting that except for the value calculated by LJ potential, the values calculated by the other potentials for mp-1018088 are smaller than the reference, leading to an underestimated prediction. The values of mp-624889 have a distribution similar to mp-1018088, but the performance of prediction (red points in Fig. 32) is better than that of mp-1018088, likely because of the smaller deviation in mp-624889 (-0.33 eV/atom for mp-624889 and -0.55 eV/atom for mp-1018088) that makes the prediction more accurate. Conversely, since the values of

mp-570002 calculated by the most classical potentials are larger than the reference in Fig. 31, the prediction of mp-570002 (lime points in Fig. 32) is overestimated. Based on these four fundamental ensemble-learning methods, voting regressor (VR) is also employed to mitigate the overall error. The performance of VR model is shown in Fig. 32e, here, VR combines RF, AB, and GB models conceptually and returns the average predicted values, so it is useful to balance out each model's weakness. For example, the predicted value of the VR on the orange point (Fig. 32e) is outperformed compared to AB and GB models. Gaussian process (GP) [355], as a generic supervised learning method to solve regression problems, is also evaluated here. The performance of GP is shown in Fig. 32f, where the errors at both ends are larger than general ensemble methods, may be attributed to the weak extrapolation.

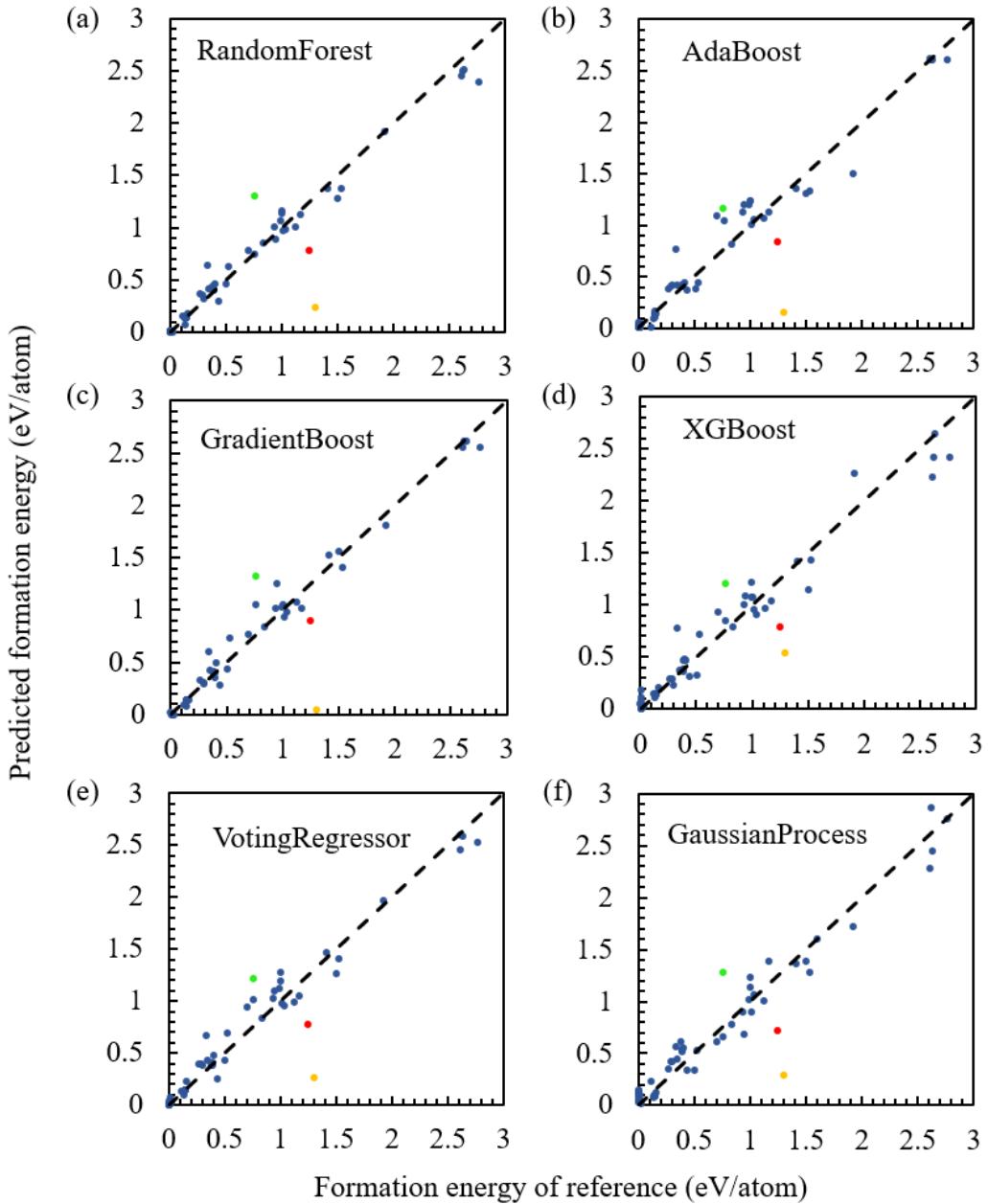


Fig. 32. Predicted formation energy versus DFT reference of each carbon structure under RF (a), AB (b), GB (c), XGB (d), VR (e), and GP (f) by using 10-fold cross-validation. The predictions of red and orange points are underestimated in each model due to the features of these two structures are generally small compared to the reference. On the contrary, the prediction of green point is overestimated since the feature of the structure are larger than reference mostly.

To further investigate the performance of these methods, twenty times 10-fold cross-validation for each method with the optimized hyperparameters are run and calculate the MAEs compared to DFT reference. Furthermore, the median absolute deviation (MAD) of each method is calculated. MAD is defined as the median of the absolute deviations from the median of the residuals, as shown in Eq. 117 and Eq. 118.

$$MAD = \text{median}(|r_i - \tilde{r}|) \quad (117)$$

Where r_i is residual between i th target's prediction and its' corresponding target. \tilde{r} represents residuals' median.

$$\tilde{r} = \text{median}(r) \quad (118)$$

MAD represents the dispersion of residuals. It is more robust than MAE since it can ignore the influence of outliers. The MAE and MAD for each method are depicted in Fig. 33. Here, the voting regressor (VR) which combined RF, AB, and GB models is utilized to mitigate the overall error by averaging the predictions. Besides, Gaussian process (GP) [355], as a generic supervised learning method to solve regression problems, is also evaluated. Overall, ensemble-learning models have better performance than that of classical interatomic potentials and GP model. Notably, all these MAEs are lower than the most accurate classical potential, LCBOP. Since the formation energy values calculated by different classical potentials have high non-linear and complex relationships, under this condition, the regression trees have better performance than classical regression methods such as GP. In the inset of Fig. 33, the formation energy of various structures predicted by RF, LCBOP, and DFT are illustrated. It can be observed that RF outperforms LCBOP in terms of overall error. However, for the structures with the highest formation energy, RF's predictions are less accurate than those by LCBOP may be due to the inherent nature of weak

extrapolation that causes the lower formation energy of mp-998866. It is worth noting that RF has weak performance for mp-1008395, mp-570002, mp-624889, and mp-1018088. The reason is the deviation of the features with respect to DFT for each structure. For mp-1018088, except for the feature calculated by LJ potential, the features calculated by the other potentials are smaller than the reference, leading to an underestimated prediction. The values of mp-624889 have a distribution similar to mp-1018088, but the deviation is smaller (-0.33 eV/atom for mp-624889 and -0.55 eV/atom for mp-1018088) that makes the prediction more accurate. Conversely, since the values of mp-1008395 and mp-570002 calculated by the most classical potentials are larger than the reference, the predictions of both are overestimated. Nevertheless, RF provides more accurate predictions in general.

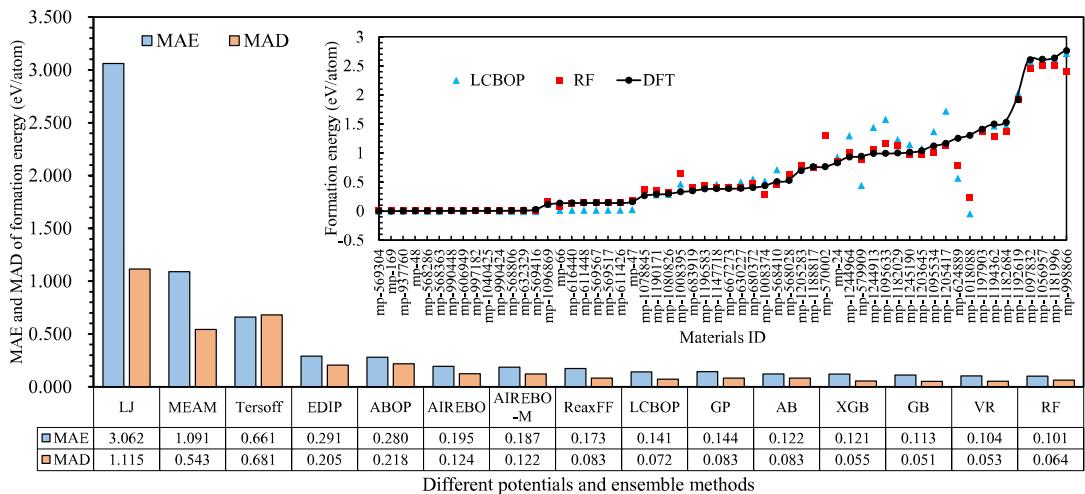


Fig. 33. MAEs and MADs of the formation energy relative to DFT reference under different methods. Overall, ensemble-learning models have better performance than that of classical interatomic potentials and GP models, and the RF has the best performance than the others. Here, the inset shows the prediction of the formation energy of carbon structures by LCBOP and RF. The black circle represents the DFT reference. The RF has a better performance than LCBOP overall. For the structures

with high formation energy, due to the inherent nature of weak extrapolation, RF is less accurate than LCBOP.

Elastic Constants of Different Carbon Structures

In order to compare the accuracy of different classical potentials in calculating the elastic constants of carbon structures, the MAE of each elastic constant for each structure with respect to the DFT reference is calculated. There are 21 elastic constants in each structure, and to facilitate comparison, Fig. 34a only shows the MAE of the first four classical potentials with the smallest error. In Fig. 34a, out of the 21 elastic constants, the Tersoff potential exhibits the smallest MAE for 15 elastic constants due to its accurate descriptions of the elastic properties of carbon structures. Since the average of MAEs of the Tersoff potential is at least an order of magnitude smaller than those of other classical potentials, in this section, The Tersoff potential is utilized as a benchmark to compare the performance of different ensemble methods. Similarly, RF, AB, GB, and XGB models are trained and tested using elastic constants calculated by the same 9 classical potentials. Grid search in combination with 5-fold cross-validation is applied to tune the hyperparameters of each model. Then the 10-fold cross-validation is conducted on the models with optimized hyperparameters to evaluate the performance of the models. Fig. 34b illustrates the MAEs of the total elastic constants of four ensemble methods conducted by twenty times 10-fold cross-validation individually. In this section, the prediction of elastic constants is a multi-target problem. However, AB, GB, and XGB don't support multi-target regression. To overcome this limitation, multi-target regressor [351] combined with all four ensemble methods is used to predict elastic constants. In brief, the multi-target regressor fits one regressor based on ensemble methods per elastic constant, it

is a simple strategy to extend the regressors which don't support multi-target problems. The MAEs of AB, RF, XGB, and GB are 30, 40, 34, and 40 GPa, respectively, which are much smaller than that of Tersoff (417 GPa). It is worth noting that the MAE of Tersoff is significantly increased by one structure (mp-1095534), as shown in the inset of Fig. 34b, that includes both sp and sp³ hybridization of carbon, making the structure more complex than others such as diamond or graphite, which only have single hybridization (sp³ or sp² hybridization). This complexity makes it difficult for Tersoff to accurately calculate. If the error associated with the mp-1095534 is removed, the MAE of Tersoff drops to 63 GPa, which is still larger than those of RF, AB, and XGB, though it is equal to the MAE of GB. Notably, different potentials behave differently for different carbon structures, hence, to get a minimal MAE of all structures from classical potentials, the smallest error among 9 classical potentials is extracted with respect to DFT reference for each structure, and then calculate the total MAE of these smallest errors. As shown in Fig. 34b, the Min represents the best performance among these classical potentials. It can be seen that AB has a smaller MAE than Min, and XGB performs similarly to Min. Therefore, ensemble learning has better performance than those of classical potentials in general.

To further assess the performance of ensemble methods compared to Tersoff, the MAEs and MADs of partial elastic constants for AB and Tersoff are presented in Table 4. The MAEs and MADs of the remaining elastic constants are excluded due to their negligible errors. In general, all 9 smallest MAEs and MADs are obtained using AB. For Tersoff with mp-1095534, the MAEs and MADs are higher than the others. Although Tersoff without mp-1095534 yields smaller MAEs and MADs than that

with mp-1095534, the MAEs and MADs of AB are still smaller, some are even over 50% lower than those of Tersoff without mp-1095534. Both tables demonstrate that the ensemble learning has better performance than Tersoff. Besides, the frequency of error percentages of the AB and Tersoff with respect to DFT is calculated as well. Fig. 35 shows the C_{11} , C_{12} , and C_{44} elastic constants error percentages distributions. It can be clearly seen that the AB has more error percentages within 20%. And especially in C_{12} , Tersoff has more frequency in large error percentages than that of AB.

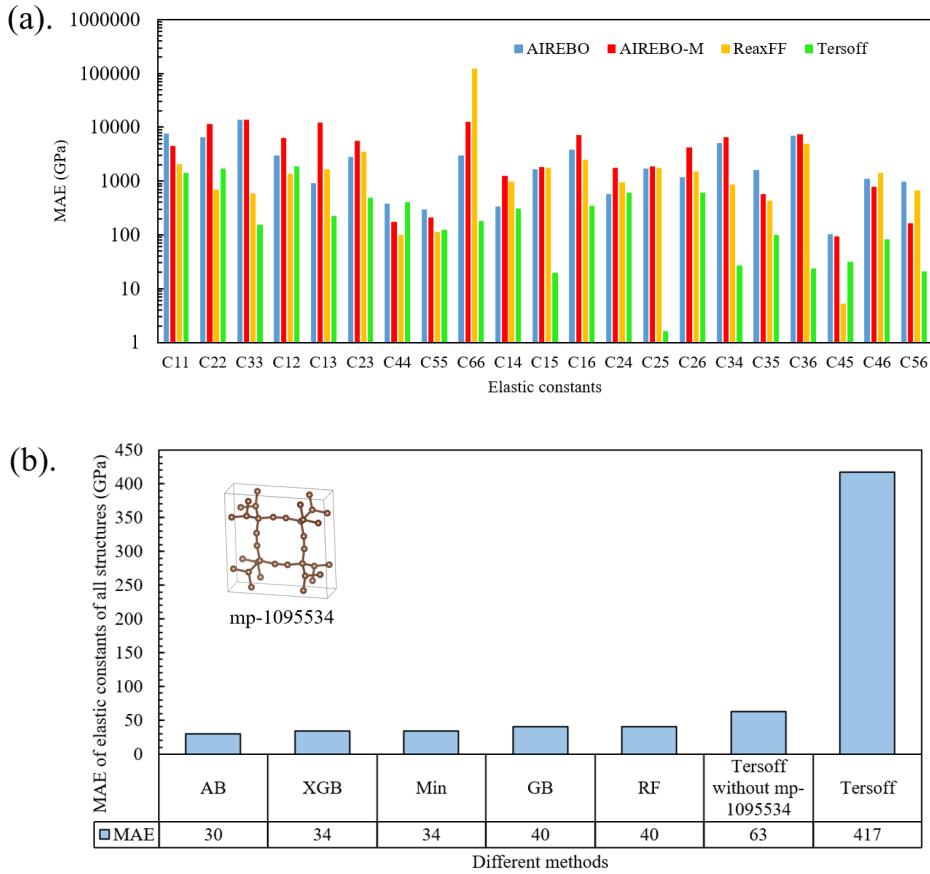


Fig. 34. MAEs of elastic constants of different classical potentials and machine learning methods. (a). MAEs of the elastic constants relative to DFT reference under the most accurate four classical interatomic potentials. 15 out of 21 smallest MAEs of elastic constants come from the Tersoff, and the average MAEs of Tersoff is at least an order of magnitude smaller than those of other classical potentials. (b). MAEs of

the total elastic constants relative to DFT reference under different methods, Min represents the best performance for all structures by using these 9 classical potentials. AB and XGB have better or similar performance than the Min, and all ensemble models are better than the Tersoff.

Table 4. MAEs and MADs of the partial elastic constants relative to DFT reference under AB and Tersoff. The smallest MAEs and MADs for each elastic constant are bolded for ease of reference.

Methods	MAE of C ₁₁ (GPa)	MAE of C ₂₂ (GPa)	MAE of C ₃₃ (GPa)	MAE of C ₁₂ (GPa)	MAE of C ₁₃ (GPa)	MAE of C ₂₃ (GPa)	MAE of C ₄₄ (GPa)	MAE of C ₅₅ (GPa)	MAE of C ₆₆ (GPa)
AdaBoost	110	135	134	34	42	47	39	50	58
Tersoff (with mp- 1095534)	1395	1732	156	1860	224	492	412	123	183
Tersoff (without mp- 1095534)	266	262	157	138	68	65	100	107	152
Methods	MAD of C ₁₁ (GPa)	MAD of C ₂₂ (GPa)	MAD of C ₃₃ (GPa)	MAD of C ₁₂ (GPa)	MAD of C ₁₃ (GPa)	MAD of C ₂₃ (GPa)	MAD of C ₄₄ (GPa)	MAD of C ₅₅ (GPa)	MAD of C ₆₆ (GPa)
AdaBoost	38	28	59	13	15	17	3	8	16
Tersoff (with mp- 1095534)	291	302	173	75	102	38	116	246	206
Tersoff (without mp- 1095534)	217	241	152	72	54	36	59	238	194

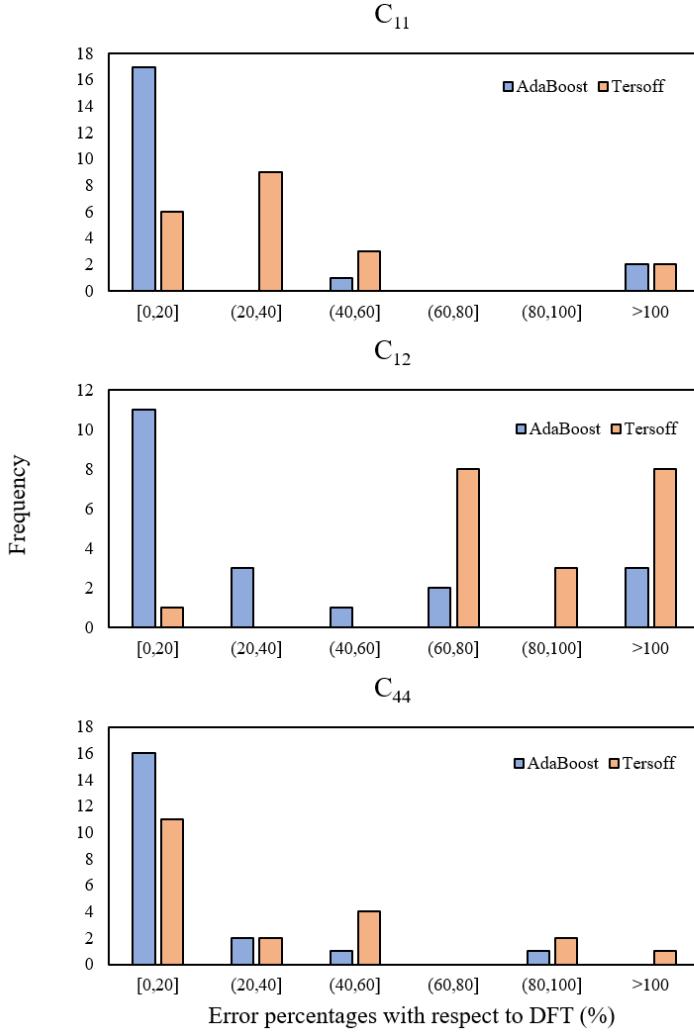
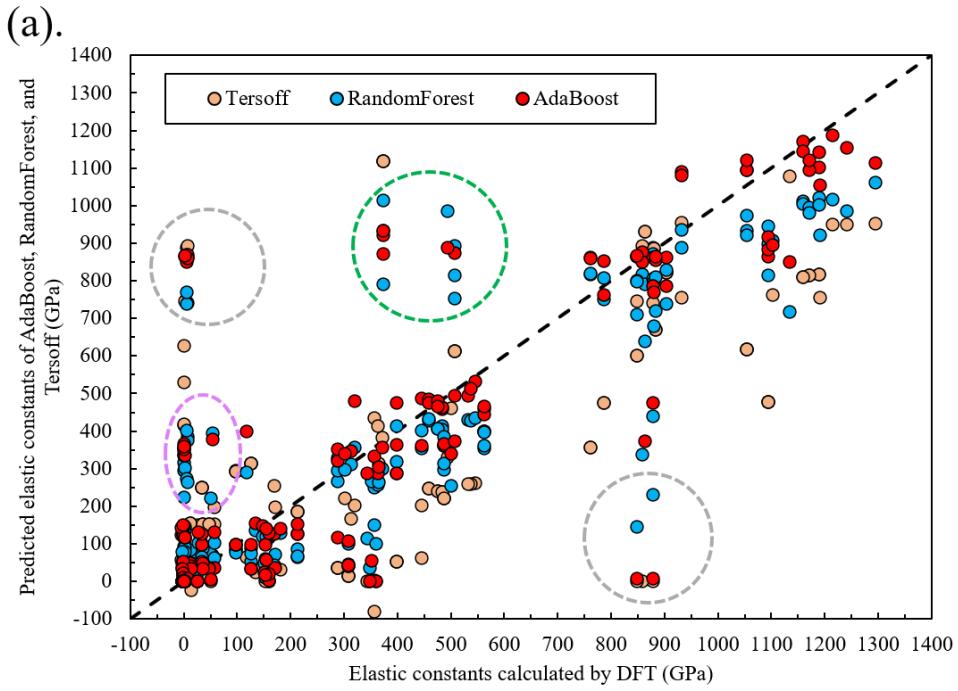


Fig. 35. Error percentages with respect to DFT of different elastic constants. The AB (AdaBoost) has higher frequencies in small error percentage regions.

Fig. 36a and Fig. 36b show the elastic constants calculated by Tersoff and predicted by AB and RF by using 10-fold cross-validation before and after input correction, respectively. As with Fig. 32, the black dashed lines are the ideal fit (1:1). To fit the plot, fifteen points of Tersoff with excessively large MAEs are removed. In Fig. 36a, the AB model displays the smallest overall MAE compared to the distribution of points at large values in both RF and Tersoff. Even so, some points have large prediction errors, as indicated by the two gray circles in Fig. 36a, where AB and RF both predict these points to be too large or too small. These points in the

gray circles correspond to elastic constants C_{11} , C_{22} or C_{33} , furthermore, upon investigating the corresponding structures, it is discovered that the structures in these circles have a commonality in which all of them are graphite-like structures. Consequently, one of the elastic constants among C_{11} , C_{22} , and C_{33} is distinct from the other two and much smaller than them, while these elastic constants are isotropic for diamond-like structures. Since the values are correlated and vary among different structures, it is necessary to consider the relationship between any two elastic constants to predict more accurately. However, the AB model cannot support multi-target regression, therefore, it can only fit one regressor individually per elastic constant. Under this way, the AB model is not able to capture the relationship among these elastic constants which leads to the inaccuracy of prediction for anisotropic structures. Unlike the AB model, the RF model can manage multi-target problems by using regression trees which can predict all targets in a single leaf. The performance of the RF with is shown in Fig. 36a. Although the overall MAE is larger than AB, the accuracy of prediction for the elastic constants in gray ovals is improved. It can be seen in Fig. 36a the values predicted by the RF (blue points) in the gray circles are much closer to the black dashed line than AB (red points). Moreover, there are only two blue points that lie in the top left gray circle which is less than the red points (seven red points lie in this area). On the contrary, more blue points fall into the purple oval which indicates less error. What needs to be explained here is before correcting the input, the MAEs of both AB and RF are close to Tersoff's without mp-1095534, which is 63 GPa. However, by comparing the dataset of MP with our DFT calculations, it is noticed that the elastic constants of graphite-like structures in gray circles are wrongly positioned. For example, the positions of C_{22} and C_{33} , C_{12} and C_{13}

are reversed. Hence, the elastic constants of the structures with wrong positions and used the are recalculated revised dataset to train and test all ensemble methods. Fig. 36b shows the prediction versus reference after correcting the dataset. It can be seen that both AB and RF have lower MAEs than Tersoff, as shown in Fig. 34b. The AB has a lower MAE than RF, possibly due to the elastic constants of similar structures in the changed dataset are more correlated with each other, and sequential regression trees like AB can reduce the bias. It is worth noting that the points in the green circles in both Fig. 36a and Fig. 36b have large errors, and all these points come from C_{11} , C_{22} or C_{33} of the structures (mp-570002, mp-998866, and mp-1008374), as shown in the inset of Fig. 36b, which have complex hybridization and limited training data lead to weak prediction. These structures have smaller values in C_{11} , C_{22} or C_{33} compared to most of the training data, which are larger than 700 GPa. Therefore, the models don't have enough fitted regressors for these structures, resulting in inaccurate predictions.



(b).

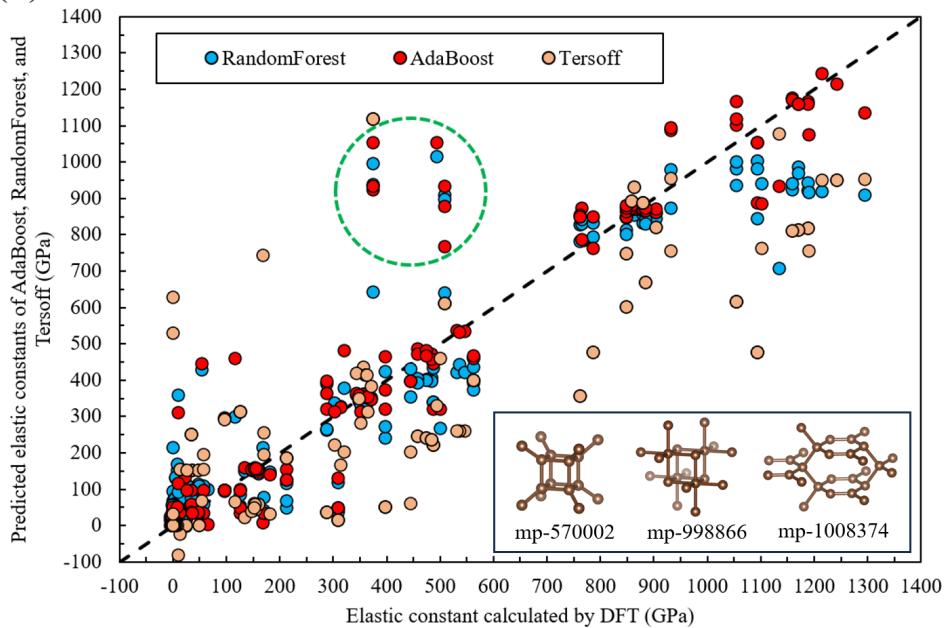


Fig. 36. Predicted elastic constants under Tersoff, RandomForest (RF), and AdaBoost (AB) versus DFT reference. (a). AB and RF are trained by uncorrected datasets. (b). AB and RF are trained by corrected datasets. In (a), there are more blue points predicted by the RF with multi-target regression trees lie into the purple oval than red points predicted by AB which fits one regressor individually per elastic constant, and in gray circles, the blue points are much closer to the black dashed line than red points, indicate multi-target regression trees can capture the correlation among elastic constants and cause more accurate prediction. (b) shows the prediction versus reference after correcting the dataset, both RF and AB models have lower MAEs than before. The points in the green circles in both (a) and (b) have large errors, and all these points come from the complicated structures which are out of training sets in both ensemble models.

Besides, we combine formation energy and elastic constants data together to train and test the same four ensemble methods. The MAEs for both properties are shown in

Fig. 37. All models perform weaker than before, with some MAEs even three times larger than when only one property is predicted. Even so, the MAEs of formation energy in RF, AB and GB models are lower than that calculated by over half of classical potentials, and the MAEs of formation energy in XGB are similar to AIREBO-M. Additionally, the MAEs of elastic constants in all models are lower than that calculated by all classical potentials, including Tersoff, though the MAE of Tersoff is lower if it removes the residuals of mp-1095534. The main reason for the increase in errors is the large size of the feature and the complexity of the correlation between features and targets will make regression trees hard to learn the relationship between features and targets correctly, the limited samples also prevent the models from learning the features of complex structures well. In our dataset, most structures are either graphitic or diamond-like. Graphite-like structures typically have anisotropic C_{11} , C_{22} , and C_{33} elastic constants, with two of them usually close to 900 GPa, and their formation energy is lower than that of diamond-like structures, which is around 0.05 eV. On the other hand, the C_{11} , C_{22} , and C_{33} are isotropic in diamond-like structures and all of them are around 1100 GPa, which are higher than those of graphitic structures, and have a higher formation energy of around 0.15 eV. Given these connections between formation energy and elastic constants, we can only use one type of property as a feature to reduce the dimension of the feature and predict both properties. For instance, when only applying the dataset of formation energy as features to train and predict both formation energy and elastic constants, we find that the MAEs are similar to those shown in Fig. 37.

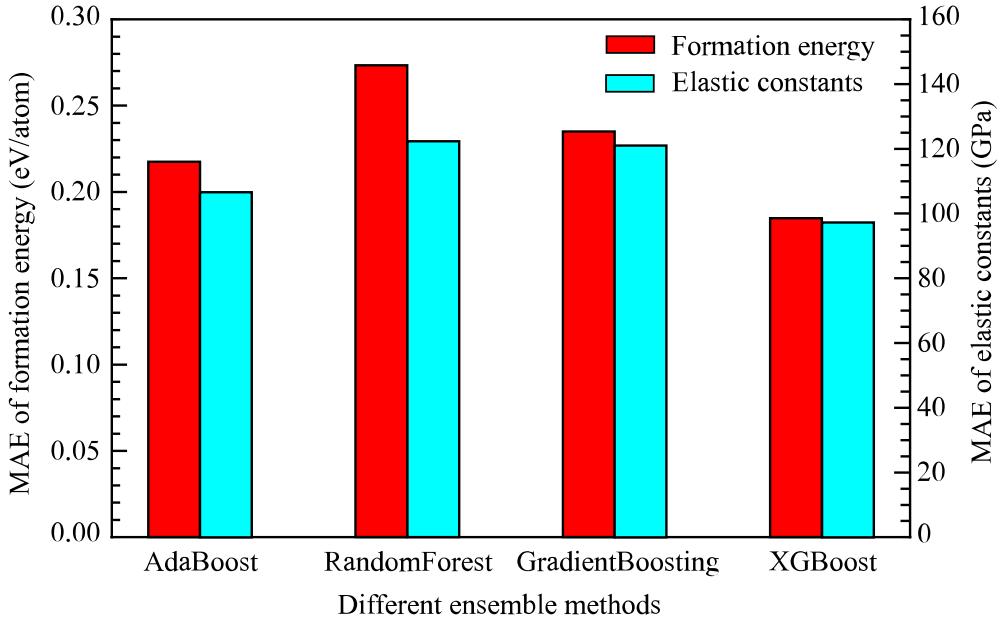


Fig. 37. MAEs of the formation energy and total elastic constants relative to corresponding DFT reference under different ensemble methods. Due to the complexity of the relationship between features and targets, all models perform weaker than when only one property is predicted. Even so, the MAEs of formation energy in RF, AB and GB models are lower than half of classical potentials. And the MAEs of elastic constants in all models are lower than that calculated by all classical potentials.

Interpretability of Ensemble Learning

To reveal the correlation or any useful information behind these features, principal component analysis (PCA) is used to decompose the high-dimensional dataset into a set of orthogonal components and project the dataset onto these components to indicate the maximum variance. Fig. 38a shows the projection of the 9-dimensional formation energy on a 2D plane, the graphite-like structures are grouped on the left of the plot, the diamond-like structures followed by fullerene-like structures are clustered on the right of graphite-like structures relative to the first principal

component. And for the others, they are located more scattered and on the right in general. This distribution is consistent with the formation energy of these structures that the graphite-like structures have the smallest formation energy followed by diamond-like and fullerene-like structures, and the complex structures have relatively higher formation energy. If combined with the second principal component, similar structures are close to each other, some of them are far away from their clusters due to their higher formation energies, all of these indicate that the feature space contains the corresponding physical meaning that is consistent with that of the target property. Fig. 38b shows the PCA of the representations after the first hidden layer in CGCNN with the same structures, likewise, the graphite-like, diamond-like, and fullerene-like structures are clustered and more compact overall, and for those points that are far away from their similar structures in Fig. 38a, they are also far away from their clusters in Fig. 38b. For the high-energy structures, three similar structures are closer to each other than the other one. Especially, one of the middle-energy structures, even though there are other structures with similar energies, is far from others in both figures, indicating the features of structures with similar energies may be different due to the different structures. Since each feature vector is composed by the calculations of the classical potentials, the correlation between these features and each feature itself for similar structures leads to the similarity of the features of similar structures, while the features of dissimilar structures differ depending on the values of each feature in them, even if the energies of these structures are similar. Therefore, different from directly using structural information as representation, these special features based on energy indirectly reflect the correlation between structures. To further demonstrate that this kind of feature vector can distinguish different structures

with similar energies, Fig. 38c shows the features' values of different structures. It can be clearly seen that among the three structures graphite-like, diamond-like, and fullerene-like, the features of the same type of structures are similar, while the features of different types of structures are different. For the outliers in these three types of structures, each outlier has different features from its own cluster, indicating the difference in their structures. Similarly, for the high-energy structure, the features of the first three are similar, while the other ones are different. In the middle-energy structures, the overall features of the second-to-last structure are different from other structures due to its high EDIP and MEAM calculated values and low Tersoff value, which causes it to be far away from other points. The second structure in middle-energy structures is similar to the fullerene-like in terms of the relative relationship between each feature, which makes it close to the fullerene-like structure in Fig. 38a and it also can be seen in Fig. 38b. This shows that these features contain the correlation between structures to a certain extent. However, the correlation between similar structures may be influenced by individual changed feature values, such as some graphite-like structures being separated from their cluster in Fig. 38a due to the underestimated calculations from LJ potential than others, it can be improved by removing the unstable features.

Besides, the criteria for node splitting in regression trees is mainly based on the loss function like mean squared error (MSE), which describes the distribution of the targets under different features, and the regression trees will identify the feature with the minimal MSE as the threshold for the split point. Since there is a correlation between the features and targets in this study, it is ideal that the feature and corresponding targets have a linear relationship. So, the accuracy of the features

varies for different structures, leading to different levels of linearity. Regression trees can evaluate the linear relationship of each feature at each split point and capture the most important feature that has minimal MSE. If a feature has relatively weak performance, it will have a nonlinear relationship between its values and targets. Thus, the targets corresponding to any two adjacent sorted values in this feature will become far apart, which causes the MSE to be larger compared to the more accurate feature. Table 5 shows the average feature importance of the regression tree fitted by formation energy, where 20 times permutation importance is employed for feature evaluation [341]. The permutation feature importance calculates the difference of error before and after permutation of the values of the features. In Table 5, ReaxFF has the most impact on the accuracy of the model, and AIREBO-M follows ReaxFF in three models. LJ and MEAM potentials have the smallest impact due to their high deviations. It should be noticed that LCBOP has a smaller MAE than ReaxFF and AIREBO-M in Fig. 33, this is because the node splitting depends on the linear relationship between features and targets instead of the difference between feature and target. Therefore, the Pearson correlation coefficient (PCC) is also used to assess the feature importance. PCC can measure linear correlation between two sets of data. The equation of PCC is as follows,

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}} \quad (119)$$

where x_i and y_i are values of the x and y variables, respectively, and \bar{x} and \bar{y} are their means. Fig. 38d is PCC of features and reference. From the last column of PCC, we can see ReaxFF has the largest positive linear correlation with the reference, and the regression tree will capture this linear correlation and use ReaxFF to split nodes, which indicates ReaxFF is the most important feature. The high positive linear

correlation also explains why ReaxFF is more important than LCBOP although LCBOP has a smaller MAE in Fig. 33. However, the LCBOP’s correlation with reference is higher than that of AIREBO-M, yet its importance is lower than that of AIREBO-M in Figure 6b. This suggests that other factors may also play a role in determining feature importance.

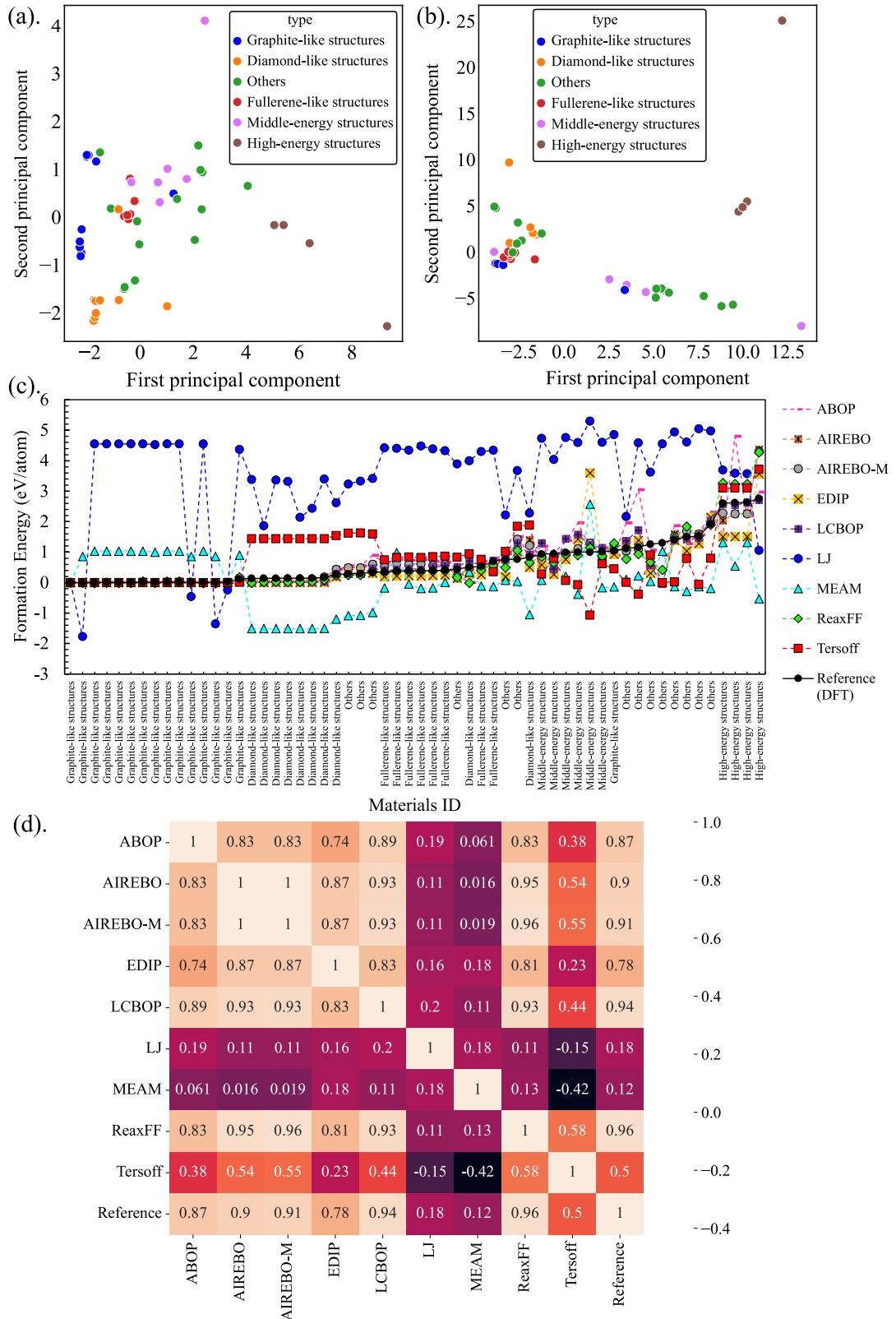


Fig. 38. Interpretability of feature importance with different methods. (a).

Visualization of the features. The original 9D vectors are reduced to 2D with PCA. Similar structures are clustered together, and the others are scattered due to their different structure from each other. The distribution of the features for all structures on the first principal component is also similar to the distribution of the formation energy for the same structures. (b). PCA of representations of the same structures in CGCNN has a similar distribution to (a). (c). The feature values corresponding to each structure. It can be seen that similar structures have similar feature values. (d). PCC of features and reference. Same as feature importance, ReaxFF has the largest PCC with the reference. The LCBOP, however, has a larger PCC than AIREBO-M, which means other factors may also play a role in determining feature importance.

Table 5. Average feature importance calculated by permutating features. A higher value indicates a more important feature and the largest importance for each method is bolded for ease of reference. The ReaxFF in all methods is the most important features for the accuracy of regression trees.

Feature	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff
AB	0.013	0.005	0.019	0.004	0.009	0.003	0.001	0.568	0.004
RF	0.042	0.059	0.370	0.025	0.038	0.008	0.004	0.900	0.036
GB	0.034	0.038	0.053	0.008	0.033	0.013	0.001	0.847	0.022
XGB	0.140	0.076	0.051	0.020	0.054	0.002	0.001	0.196	0.010

Except the factors mentioned before, the loss function indicates that local minimal error also impacts on the choice of features for splitting. The regression tree algorithm is inherently greedy (Eq. 113 to 115), aiming to find the feature with the local minimal error as splitting feature as long as the sorted target values of the samples are

close to each other. Fig. 39 shows two different accurate features, blue points are the ideal feature, and orange points are the less accurate feature, but both have the same splitting threshold (black dashed line) and by calculating each subsets impurity, it will be found that they have the same impurities, 5 and 2 (the green and red dashed lines) in each subset. So, the feature which can lead local minimal impurity can also be selected as splitting feature, it is necessary to consider about the local minimal error of each feature. To quantify the level of local minimal error of each feature, a way to describe the frequency of its occurrence is proposed. Fig. 40a illustrates the process of computing the local minimal level of each feature. In the beginning, initialize the local minimal level of each feature to 0, and split each feature's vector and target's vector into small parts, each part of the target consists of adjacent sorted target values and each part of the feature value is the corresponding target values after sorting according to the MD-calculated values. For each part of the feature, we find its minimum and maximum values and compare them to each part of the target. If both minimum and maximum values exist in the target's part, which indicates a local minimal error and adds 1 to this feature's local minimal level. Fig. 40b shows the local minimal level of each feature, where ReaxFF has the largest local minimal level, followed by AIREBO-M. Thus, the local minimal level may explain a certain degree of the feature importance of ReaxFF and AIREBO-M in Table 5. In addition, the PCC, MAE, feature importance, and local minimum frequency of RF trained by carbon materials' formation energy are analyzed to illustrate that under different structures, the ensemble learning likely to use the more accurate potential as criteria to output. As shown in Table 6, the highest PCC, lowest MAE, largest Importance, and largest frequency are bolded. It can be seen that under different energy intervals, the accuracy

of each potential is different, the PCC, MAE, feature importance, and local minimum frequency of each feature in ensemble models are positively correlated to each other in general. The ensemble learning splits the node based on these indicators and normally uses the more accurate potentials' properties as criteria to output, although the ReaxFF has the largest PCC and importance overall, ensemble learning can generally utilize the more accurate feature as criteria for predicting under the corresponding structures, for the structures with high formation energy, LCBOP is not the most important feature may be due to the lack of training data in which only 5 samples, the importance of LCBOP is still the second large. this characteristic is based on the local minimum algorithm which can capture relatively accurate features for splitting the tree's node.

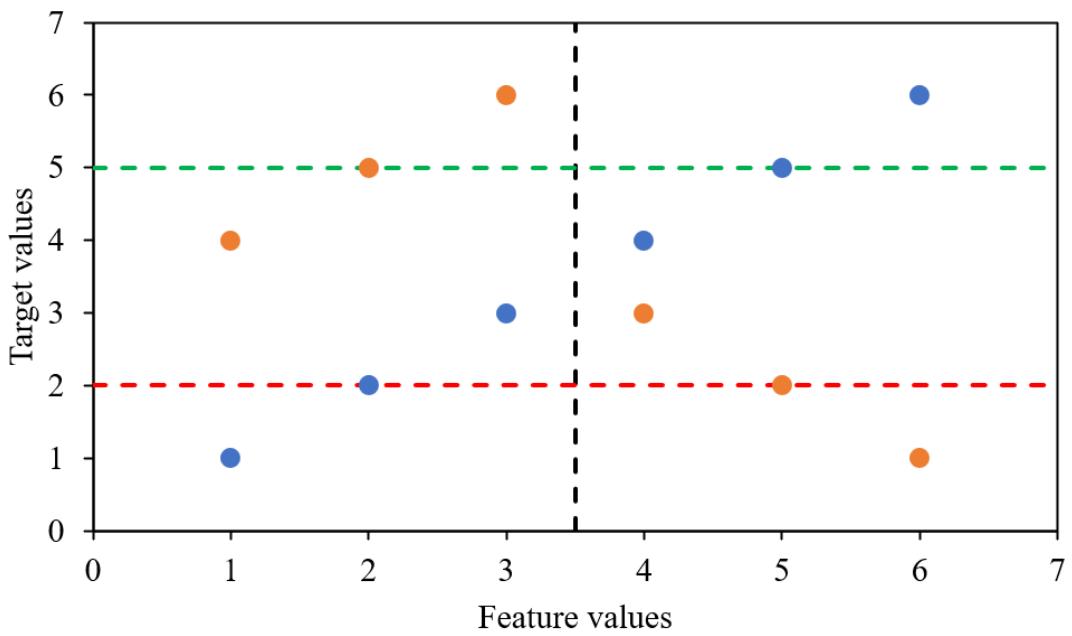


Fig. 39. Illustration of splitting thresholds for two features, they have the same splitting threshold because they have the same local minimal impurity.

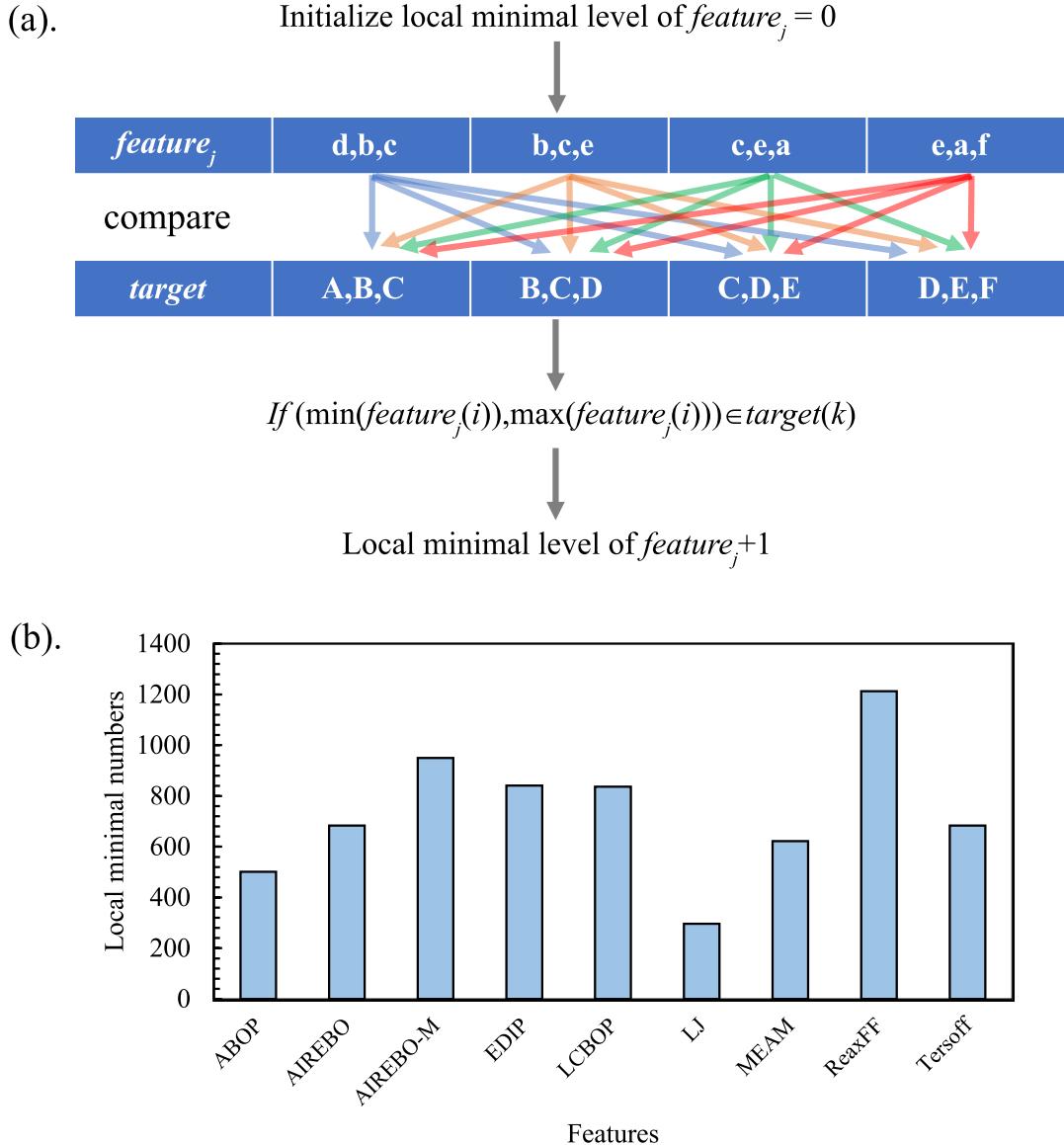


Fig. 40. Frequency of local optimization occurred in each feature. (a). Flowchart of calculating local minimal level of $feature_j$. In the beginning, initialize the local minimal level of each feature to 0 and split each feature's vector and target's vector into small parts, each part of the target consists of adjacent sorted target values (such as A-C represent three similar formation energy) in the figure, and each part of the feature contains the MD-calculated values of structures corresponding to the target vector (such as a-c). Then compare each feature's part to each target's part to check if

both minimum and maximum values of each feature part also exist in target's part, if so, which means it has a local minimal error and add 1 to this feature's local minimal level. (b). Local minimum level of different features. ReaxFF has the largest local minimal level, followed by AIREBO-M, which consists of the feature importance.

Table 6. The PCC, MAE, feature importance, and local minimum frequency of RF trained by carbon materials under different formation energy intervals. The smallest MAE, largest PCC, largest feature importance, and highest local minimum frequency are bolded for ease of reference.

0-0.507 eV/atom	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff
PCC	0.956	0.973	0.972	0.988	0.990	0.273	-0.014	0.981	0.334
MAE	0.141	0.089	0.086	0.081	0.057	3.245	0.972	0.063	0.529
Importance	0.080	0.047	0.063	0.064	0.149	0.031	0.019	0.002	0.046
Frequency	921	945	1297	1227	1432	241	856	1107	409
0.507-1.528 eV/atom	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff
PCC	0.727	0.636	0.650	0.482	0.671	0.626	0.112	0.788	-0.557
MAE	0.379	0.278	0.275	0.514	0.313	3.130	1.606	0.237	0.885
Importance	0.161	0.064	0.139	0.010	0.070	0.021	0.032	0.199	0.062
Frequency	87	373	349	172	235	39	1	582	0
1.528-2.763 eV/atom	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff
PCC	0.877	0.928	0.939	0.883	0.999	0.233	0.521	0.989	0.952
MAE	0.852	0.602	0.533	0.872	0.054	1.552	2.013	0.684	0.707
Importance	0.001	0.001	0.047	0.000	0.124	0.001	-0.014	0.124	0.227
Frequency	4	29	28	15	31	0	4	29	21

In summary, the MAE, PCC, feature importance, and local minimum frequency of each feature in ensemble models are correlated, and positively correlated to each other in general. The feature with the highest importance also generally owns the smallest MAE, highest PCC, and highest local minimum frequency, indicating the relationship between the feature and target, and the algorithm of the decision tree are correlated to each other and will decide the feature importance of decision trees. So, based on the linear correlation between features and targets and the local greedy characteristic of algorithms, ensemble learning can capture relatively accurate features calculated by the nine classical potentials under the corresponding structure for splitting the node. More accurate features can be used to improve the ensemble model's performance systematically.

Formation Energy Prediction of New Carbon Structures

To further evaluate the performance, the trained formation energy RF model is employed to predict the formation energy of new carbon structures. We extract all structures of both silicon carbide and silicon from the MP database and compared them with carbon structures by using the similarity method [356]. This method evaluates the dissimilarity of any two structures by calculating the statistical difference in local coordination environment of all sites in both structures. Out of 76 silicon carbide and silicon structures, 10 are extracted as new structures based on the similarity method. We replace the silicon element with the carbon element in these 10 structures to get new carbon structures and calculate their formation energy with the nine classical potentials as features. These features are then input into the pre-trained RF model to predict formation energy. Fig. 41 illustrates the formation energy of each new structure calculated by RF, CGCNN, ALIGNN, DFT, and 3 most accurate

classical potentials. The MAEs of CGCNN, ALIGNN, and RF are 0.376 eV/atom, 0.446 eV/atom, and 0.850 eV/atom, respectively, while the minimum MAE in classical potentials is 1.402 eV/atom for AIREBO. The CGCNN and ALIGNN models use interatomic structural information as input, which enables them to have certain transferability when facing structures outside the training set. Different from these two models, the RF model based on the feature values instead of atomic structures, although RF performs well in interpolation, its extrapolation ability is limited for new structures with energies higher than those in the training set. This can be seen from the fact that the high energy prediction value approaches a certain value in Fig. 41. Since only 4 carbon structures in the training dataset have formation energy larger than 2 eV/atom, and the highest energy is 2.7 eV/atom, its maximum prediction value is around 2.7 eV/atom, and it cannot make reasonable predictions for structures higher than 2.7 eV/atom. In addition, for new structures whose energies are within the range of the training set, RF's predictions depend on the accuracy of the features. In other words, for the lowest energy structure in the graph, since all features are higher than those calculated by DFT, RF will over-predict like the classical potential. Therefore, the performance of RF depends on the diversity of the training set and the transferability of the features. To further inspect the relationship between features and model, the MAE of features corresponding to DFT, and feature importance of RF are calculated, as shown in Table 7. Interestingly, the feature importance changes with the accuracy of the features for the new structure. The AIREBO-M has the smallest MAE and has the largest importance, but the ReaxFF has a large MAE and its importance is low. This may indicate that the RF can filter

out the features with large errors according to the trained feature values, so as to split the trees by the features within a reasonable range.

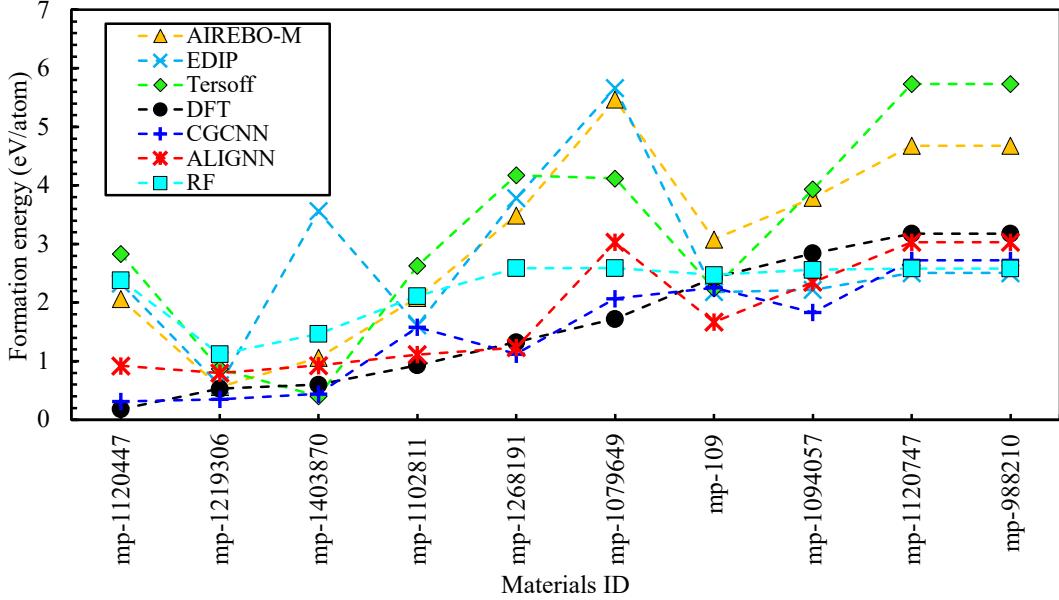


Fig. 41. Performance of different methods for prediction of formation energy of some carbon structures. The x axis represents the ID of silicon carbide and silicon structures in MP database corresponding to the carbon structures.

Table 7. The MAE of features corresponds to DFT, and feature importance of RF.

	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff
MAE	2.393	3.253	1.402	1.446	1.934	4.137	48.292	5.012	1.651
Importance	0.001	0.073	0.143	0.005	0.091	0.000	-0.03	0.000	0.025

Discussion

There are some limitations and opportunities for improvement. First, the limited size of the training dataset may restrict the performance of the models. This constraint is acknowledged in Fig. 36b and Fig. 41, and it could be mitigated by including more training samples to extend learning space or to make the interpolate prediction

smoother. Due to the limited carbon structures in MP, the Si-O binary systems are used to test the performance of ensemble learning in larger datasets and evaluate the influence of training data size. Here, 335 Si-O structures are extracted from MP and three classical potentials, COMB [357], Tersoff [358], and Vashishta [359], calculate their formation energies. Fig. 42 shows the performance of different models under different k -fold cross-validation, as the k increases, the training data becomes more and the error decreases. Finally, the error tends to be stable when a certain amount of training data is reached, indicating that under-fitting will lead to prediction errors when there is insufficient training data set. When the number of training sets reaches a certain level, the very similar training data won't help the model to improve the performance, since the model has learned enough feature information based on the previous training set to predict new structures. Although for the extrapolate new structures is limited, such as in Fig. 41, the overall errors in all ensemble models for 10-fold cross-validation (0.132 eV/atom, 0.143 eV/atom, 0.140 eV/atom, and 0.141 eV/atom for RF, AB, GB, and XGB, respectively) are smaller than that of three potentials (0.240 eV/atom, 0.156 eV/atom, and 0.147 eV/atom for COMB, Tersoff, and Vashishta, respectively). Fig. 43 illustrates the formation energy of each Si-O structure calculated by RF, CGCNN, ALIGNN, DFT, and 3 classical potentials in a logarithmic scale, negative values in CGCNN and ALIGNN are taken as absolute values for plotting. For low-energy structures (the first 250 structures), compared with DFT calculations, the CGCNN, ALIGNN, COMB, and Tersoff have higher deviations than those of Vashishta and RF in general. The COMB and ALIGNN have higher predictions than the DFT, and the Tersoff and CGCNN are vice versa. For high-energy structures (the remaining 85 structures), however, Vashishta's predictions are

generally lower than DFT values, and the other two classical potentials are more accurate than Vashishta's. The RF and ALIGNN have smaller deviations than the CGCNN. The overall MAEs of RF, ALIGNN, CGCNN, Vashishta, Tersoff, and COMB are 0.134 eV/atom, 0.106 eV/atom, 0.146 eV/atom, 0.148 eV/atom, 0.158 eV/atom, and 0.240 eV/atom, respectively. Briefly, the ML-based models have lower MAEs than that of classical potentials and the RF has the lowest overall error besides the ALIGNN. Even though the ALIGNN has a larger deviation in the low-energy region than the RF on a logarithmic scale, the energy difference between DFT and ALIGNN is small on a linear scale, and the RF has some points that have relatively large deviations in both low-energy and high-energy regions, the ALIGNN, however, has the lowest MAE for predictions of high-energy structures. To interpret the RF, the PCC, MAE, local minimum frequency and feature importance between different potentials in the low-energy and high-energy regions are calculated as well. As shown in the Table 8, for the low-energy region, Vashishta has the smallest MAE, the largest PCC, the largest feature importance, and largest local minimum frequency, while Tersoff is in the high-energy region. These results are generally consistent with those in Table 6, also indicating that ensemble learning can find more accurate potential energy calculations under the corresponding structure as features for prediction.

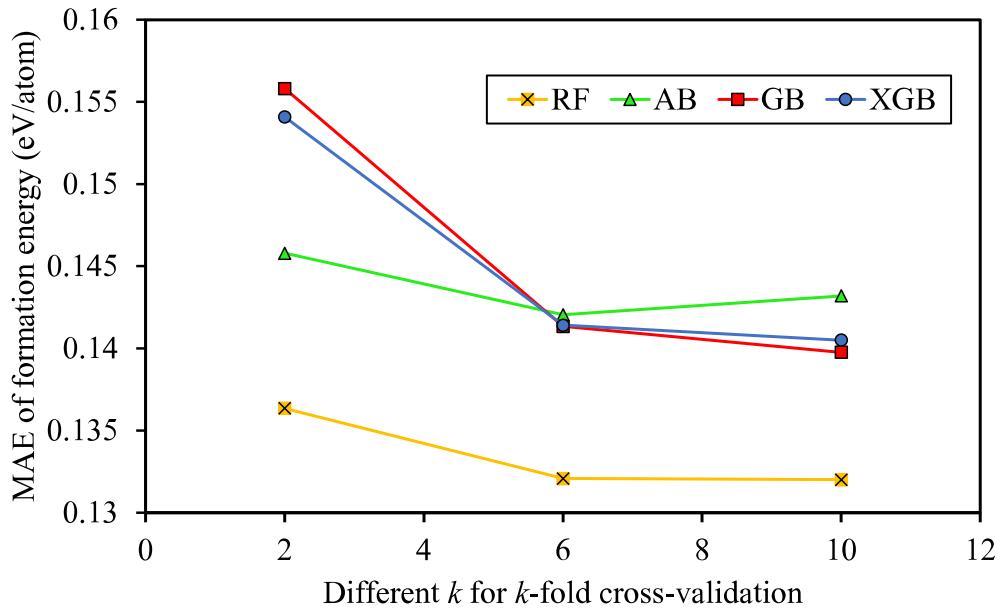


Fig. 42. Performance of ensemble learning models for formation energy prediction using different training sizes.

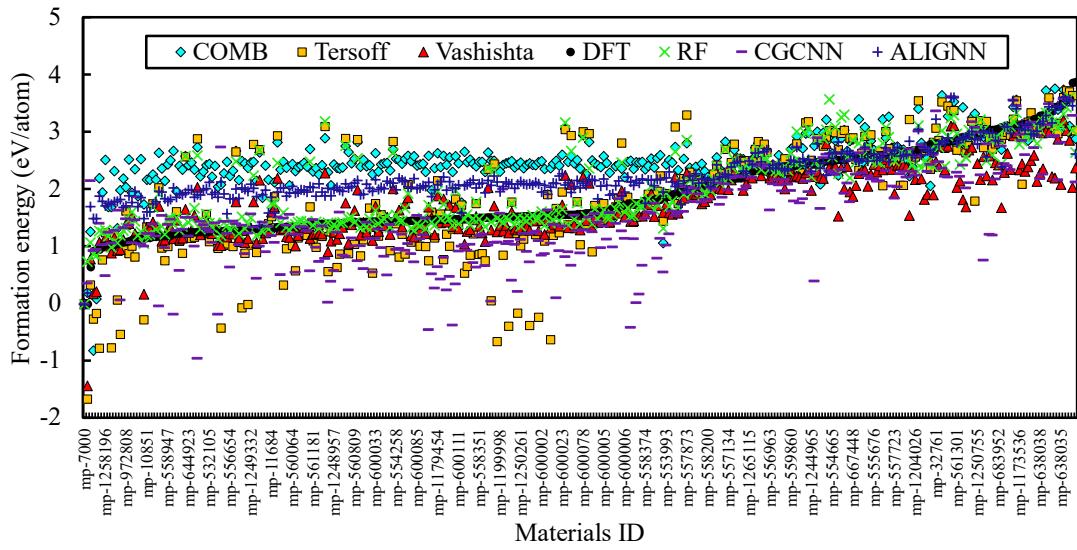


Fig. 43. The formation energy of each Si-O structure calculated by RF, CGCNN, ALIGNNN, DFT, COMB, Tersoff, and Vashishta. Among the classical potentials, Vashishta and Tersoff are the most accurate potentials for energy prediction of low-energy structures and high-energy structures, respectively. And the ALIGNNN has the

best performance among all methods.

Table 8. The MAE, PCC, feature importance and local minimum frequency of COMB, Tersoff, and Vashishta. The smallest MAE, largest PCC, largest feature importance, and highest local minimum frequency are bolded for ease of reference.

Low-energy structures	COMB	Tersoff	Vashishta
MAE	0.143	0.061	0.015
PCC	0.350	0.420	0.810
Importance	1.499	3.375	7.242
Frequency	356	486	530
High-energy structures	COMB	Tersoff	Vashishta
MAE	0.501	0.424	0.518
PCC	0.310	0.540	0.240
Importance	0.369	0.613	0.261
Frequency	229	397	218

Second, the performance of the regression trees is related to the accuracy of the features and a linear correlation between features and targets. So, more accurate classical interatomic potentials may be used as features to improve performance. In addition, more features make ensemble methods more complex and harder to interpret the feature importance, and complex correlations between features and targets also may lead to regression trees being unstable which affects the performance of ensemble learning, like the overall performance of the two properties prediction model (Fig. 37) is worse than single property prediction model (Fig. 33 and Fig. 36), so to get better performance and interpretability, single property prediction model and

appropriate feature size need to be considered. Table 9 shows the performance of RF for predicting the formation energy of carbon materials with different feature sizes. Compared with the MAE of MD simulation with a single classical potential (Fig. 33), the RF characterized by one feature calculated with a single potential performs better, but not as well as the RF characterized by all potentials. Besides, except for the RF with low-precision features (using ABOP, LJ, MEAM, Tersoff, and EDIP as features), the RFs with high-precision features (using AIREBO, AIREBO-M, LCBOP, and ReaxFF as features) and using only accurate LCBOP and ReaxFF as features perform better than the RF with only a single feature. In particular, the RF performs best when only the highest-precision potentials LCBOP and ReaxFF are used as features, and these results also show that when the number of features increases, especially when inaccurate feature values are added, the accuracy of the model will decrease due to the more complex feature relationship, on the contrary, when only the accurate features are used, the correlation between features and targets are more linear, make the regression tree easier to find the intrinsic correlation between feature and target.

Table 9. The MAE of RF trained by different features size.

	ABOP	AIREBO	AIREBO-M	EDIP	LCBOP	LJ	MEAM	ReaxFF	Tersoff	Low-precision features	High-precision feature	LCBOP+ReaxFF	All
MAE of RF	0.229	0.181	0.154	0.247	0.129	0.341	0.401	0.104	0.236	0.167	0.102	0.088	0.101

Except for the discussions above, under a certain feature size, the input feature composed of physical properties calculated by different classical interatomic potentials is not convenient to get since it needs to calculate each new structure's physical properties by these potentials. Inspired by the imputation of missing input values, this dilemma can be relieved by utilizing imputation methods to infer the

missing values from the known part of the data. Here, we use k-Nearest Neighbors (KNN) approach [360] to impute the missing features in the input. KNN based on Euclidean distance metric to learn the correlation between features and find the nearest neighbors of the missing values among the samples that have values for the features, and the missing values are imputed using average weighted values from the nearest neighbors. Fig. 44 shows under the 10-fold cross-validation of the formation energy dataset, the performance of the RF model combined with 2-nearest neighbors' imputation when only one or two features use MD calculation. It can be clearly seen that when the more accurate features are calculated and other features are imputed, the accuracy of the model is higher. This is because the more accurate features are more important in the model, so calculating these features instead of imputing them will reduce the deviations of these features, thereby improving the prediction stability of the model. It can also be found from Fig. 44 that if more feature values are calculated as input, the accuracy of the model will be higher, such as when ReaxFF and AIREBO-M are used as input, the MAE is smaller than others, and the accuracy of the model is similar to that of the GB full-input model (Fig. 33). So, it is feasible to reduce the workload of obtaining the input part through the imputation of missing data though it will increase the error to a certain extent. Finally, it is worth mentioning that there are also some questions not discussed in this paper, such as the feasibility of the ensemble learning method in MD simulation and structure optimization problems, and these questions need further research to determine whether ensemble learning can do these calculations.

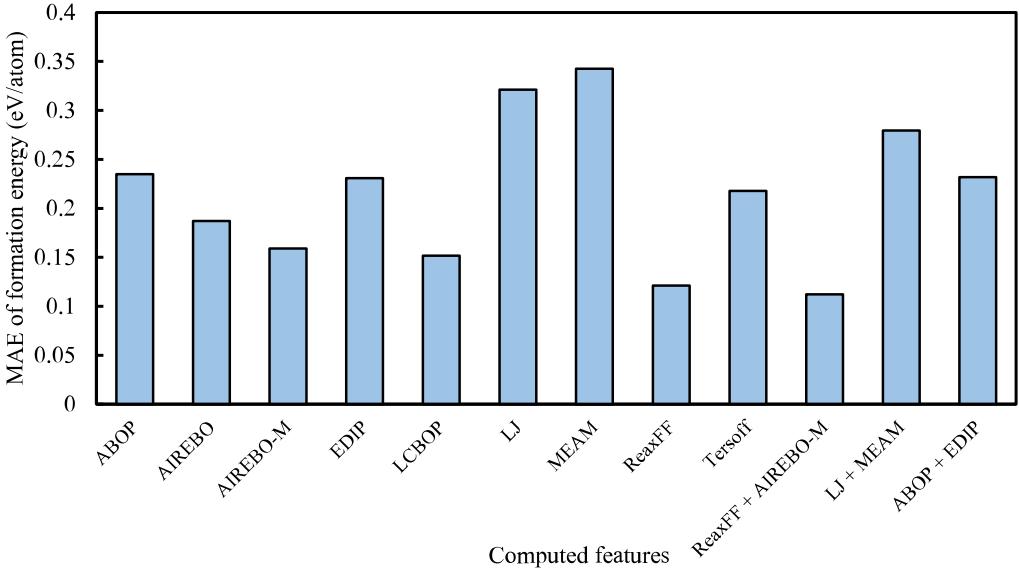


Fig. 44. The MAEs of formation energy of RF under 2-nearest neighbors' imputation.

The x-axis represents different conditions, where the computed features are obtained by using MD simulations, and all other features in each condition come from 2-nearest neighbors' imputation. The RF model has better performance when more features or more accurate features are calculated as input instead of imputation.

Summary

In summary, we explore the possibility of prediction for the physical properties of a small size of carbon allotropes based on ensemble learning. The formation energy and elastic constants of carbon structures, as examples, can be predicted by using this kind of method. In general, the ensemble methods have better performance than the classical interatomic potentials we used in this work. Although at some points, the prediction is not accurate due to a lack of training data, the high dimensionality of features, and the local greedy characteristic of the algorithm, making the model difficult to learn the relationship between features and targets correctly. The principal component analysis shows the input which consists of the values calculated by

different classical interatomic potentials, has a similar distribution with the corresponding target physical property. What's more, the Pearson correlation coefficient illustrates the linear correlation between input and output, and the regression trees can capture the relatively accurate feature as criteria for splitting the point in regression trees by evaluating the feature importance.

CHAPTER 6

CONCLUSIONS

This dissertation introduces three studies in different materials based on machine learning. The first two of them use the crystal graph convolutional neural networks as potential, one to predict the work of adhesion of 50% Al-doped TiAl/TiAlN interface and another one to predict the energy of $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ medium-entropy alloys in, and combined with reinforcement learning or Monte Carlo methods to efficiently search for the optimal TiAl/TiAlN interface with good adhesion and the $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ with low energy. The last research proposed a new machine learning potential method that is based on ensemble learning, combined with the features calculated by different classical interatomic potentials to efficiently train and predict properties of structures. The first two research projects demonstrate that graph convolutional neural networks can be used in different advanced materials' designs to approximate or replace DFT calculations and hold great potential in providing a fast preliminary screening method for optimization problems with large configuration spaces, and by combining RL, structural correlations and features can be discovered from these potential configurations. What's more, the electronic structure and bonding analysis of the TiAl/TiAlN interface with high work of adhesion shows that the atomic structure at the TiAl/TiAlN interface follows a certain pattern, and this pattern is similar to the TiAl/TiAlN interface with lower Al-doped, indicating a similar strengthening mechanism may commonly exist in Al-doped Ti/TiN structures. The short-range order analysis of low-energy $\text{Si}_{0.33}\text{Ge}_{0.33}\text{Sn}_{0.33}$ also indicates the existence of different SRO compared to other compositions in which the repulsive interaction of the Si-Sn bond is suppressed. The last research from a different perspective to

generate machine learning potential without complex descriptors or neural networks, by using ensemble learning consists of regression trees to learn the correlations between features composed by different classical interatomic potentials' calculations of the structures and output their properties based on the accuracy of each feature. This method generally can capture the most accurate classical potential's feature under different conditions as the most important feature to predict. Therefore, these studies demonstrate the feasibility of machine learning in fundamental research on different materials and their efficiency compared with DFT calculations. Meanwhile, the results and findings obtained from the study of interfaces and medium-entropy alloys may provide some reference for similar studies in the future.

REFERENCES

1. Van Der Giessen, E., et al., *Roadmap on multiscale materials modeling*. Modelling and Simulation in Materials Science and Engineering, 2020. **28**(4): p. 043001.
2. Mishin, Y., *Machine-learning interatomic potentials for materials science*. Acta Materialia, 2021. **214**: p. 116980.
3. Kohn, W. and L.J. Sham, *Self-consistent equations including exchange and correlation effects*. Physical review, 1965. **140**(4A): p. A1133.
4. Hohenberg, P. and W. Kohn, *Inhomogeneous electron gas*. Physical review, 1964. **136**(3B): p. B864.
5. Daw, M.S. and M.I. Baskes, *Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals*. Physical Review B, 1984. **29**(12): p. 6443.
6. Kınacı, A., et al., *Thermal conductivity of BN-C nanostructures*. Physical Review B—Condensed Matter and Materials Physics, 2012. **86**(11): p. 115410.
7. Chenoweth, K., A.C. Van Duin, and W.A. Goddard, *ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation*. The Journal of Physical Chemistry A, 2008. **112**(5): p. 1040-1053.
8. Bian, H., et al., *Microstructure evolution and mechanical properties of titanium/alumina brazed joints for medical implants*. Metals, 2019. **9**(6): p. 644.
9. Lee, H.C., et al., *Highly conductive flexible metal–ceramic nanolaminate electrode for high-performance soft electronics*. ACS applied materials & interfaces, 2018. **11**(2): p. 2211-2217.
10. Leyland, A. and A. Matthews, *On the significance of the H/E ratio in wear control: a nanocomposite coating approach to optimised tribological behaviour*. Wear, 2000. **246**(1-2): p. 1-11.
11. Si Abdelkader, H., H. Faraoun, and C. Esling, *Effects of rhenium alloying on adhesion of Mo/HfC and Mo/ZrC interfaces: A first-principles study*. Journal of Applied Physics, 2011. **110**(4).
12. Yang, W., et al., *Deformation mechanisms in Ti/TiN multilayer under compressive loading*. Acta Materialia, 2017. **122**: p. 99-108.
13. Salehinia, I., et al., *Molecular dynamics simulations of plastic deformation in Nb/NbC multilayers*. International Journal of Plasticity, 2014. **59**: p. 119-132.

14. Zhang, Y., et al., *Microstructures and properties of high-entropy alloys*. Progress in materials science, 2014. **61**: p. 1-93.
15. Miracle, D.B. and O.N. Senkov, *A critical review of high entropy alloys and related concepts*. Acta materialia, 2017. **122**: p. 448-511.
16. Mu, S., et al., *Electronic transport and phonon properties of maximally disordered alloys: From binaries to high-entropy alloys*. Journal of Materials Research, 2018. **33**(19): p. 2857-2880.
17. Khan, S.N. and M. Eisenbach, *Density-functional Monte-Carlo simulation of CuZn order-disorder transition*. Physical Review B, 2016. **93**(2): p. 024203.
18. Zuo, Y., et al., *Performance and cost assessment of machine learning interatomic potentials*. The Journal of Physical Chemistry A, 2020. **124**(4): p. 731-745.
19. Liu, X., J. Zhang, and Z. Pei, *Machine learning for high-entropy alloys: progress, challenges and opportunities*. Progress in Materials Science, 2022: p. 101018.
20. Behler, J., *Atom-centered symmetry functions for constructing high-dimensional neural network potentials*. The Journal of chemical physics, 2011. **134**(7).
21. Artrith, N., T. Morawietz, and J. Behler, *High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide*. Physical Review B—Condensed Matter and Materials Physics, 2011. **83**(15): p. 153101.
22. Morawietz, T., V. Sharma, and J. Behler, *A neural network potential-energy surface for the water dimer based on environment-dependent atomic energies and charges*. The Journal of chemical physics, 2012. **136**(6).
23. Behler, J. and M. Parrinello, *Generalized neural-network representation of high-dimensional potential-energy surfaces*. Physical review letters, 2007. **98**(14): p. 146401.
24. Bartók, A.P., et al., *Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons*. Physical review letters, 2010. **104**(13): p. 136403.
25. Thompson, A.P., et al., *Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials*. Journal of Computational Physics, 2015. **285**: p. 316-330.
26. Shapeev, A.V., *Moment tensor potentials: A class of systematically improvable interatomic potentials*. Multiscale Modeling & Simulation, 2016. **14**(3): p. 1153-1173.

27. Haghightlari, M., et al., *NewtonNet: a Newtonian message passing network for deep learning of interatomic potentials and forces*. Digital Discovery, 2022. **1**(3): p. 333-343.
28. Drautz, R., *Atomic cluster expansion for accurate and transferable interatomic potentials*. Physical Review B, 2019. **99**(1): p. 014104.
29. Baskes, M.I., *Modified embedded-atom potentials for cubic materials and impurities*. Physical review B, 1992. **46**(5): p. 2727.
30. Tersoff, J., *New empirical approach for the structure and energy of covalent systems*. Physical review B, 1988. **37**(12): p. 6991.
31. Stillinger, F.H. and T.A. Weber, *Computer simulation of local order in condensed phases of silicon*. Physical review B, 1985. **31**(8): p. 5262.
32. Rupp, M., et al., *Fast and accurate modeling of molecular atomization energies with machine learning*. Physical review letters, 2012. **108**(5): p. 058301.
33. Tersoff, J., *Modeling solid-state chemistry: Interatomic potentials for multicomponent systems*. Physical review B, 1989. **39**(8): p. 5566.
34. Jose, K., N. Artrith, and J. Behler, *Construction of high-dimensional neural network potentials using environment-dependent atom pairs*. The Journal of chemical physics, 2012. **136**(19).
35. Darden, T., D. York, and L. Pedersen, *Particle mesh Ewald: An $N \cdot \log(N)$ method for Ewald sums in large systems*. The Journal of chemical physics, 1993. **98**(12): p. 10089-10092.
36. Ghasemi, S.A., et al., *Interatomic potentials for ionic systems with density functional accuracy based on charge densities obtained by a neural network*. Physical review B, 2015. **92**(4): p. 045131.
37. Faraji, S., et al., *High accuracy and transferability of a neural network potential through charge equilibration for calcium fluoride*. Physical Review B, 2017. **95**(10): p. 104105.
38. Faraji, S., et al., *Surface reconstructions and premelting of the (100) CaF₂ surface*. Physical Chemistry Chemical Physics, 2019. **21**(29): p. 16270-16281.
39. Ko, T.W., et al., *A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer*. Nature communications, 2021. **12**(1): p. 398.

40. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. nature, 1986. **323**(6088): p. 533-536.
41. Yao, K., et al., *The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics*. Chemical science, 2018. **9**(8): p. 2261-2269.
42. Yoo, D., et al., *Atomic energy mapping of neural network potential*. Physical Review Materials, 2019. **3**(9): p. 093802.
43. Marques, M.R., et al., *Neural network force fields for simple metals and semiconductors: construction and application to the calculation of phonons and melting temperatures*. Physical Chemistry Chemical Physics, 2019. **21**(12): p. 6506-6516.
44. Eckhoff, M. and J.r. Behler, *From molecular fragments to the bulk: development of a neural network potential for MOF-5*. Journal of chemical theory and computation, 2019. **15**(6): p. 3793-3809.
45. Eckhoff, M., et al., *Closing the gap between theory and experiment for lithium manganese oxide spinels using a high-dimensional neural network potential*. Physical Review B, 2020. **102**(17): p. 174102.
46. Eckhoff, M., et al., *Predicting oxidation and spin states by high-dimensional neural networks: Applications to lithium manganese oxide spinels*. The Journal of Chemical Physics, 2020. **153**(16).
47. Marchand, D., et al., *Machine learning for metallurgy I. A neural-network potential for Al-Cu*. Physical review materials, 2020. **4**(10): p. 103601.
48. Liu, M. and J.R. Kitchin, *SingleNN: modified behler–Parrinello neural network with shared weights for atomistic simulations with transferability*. The Journal of Physical Chemistry C, 2020. **124**(32): p. 17811-17818.
49. Mangold, C., et al., *Transferability of neural network potentials for varying stoichiometry: Phonons and thermal conductivity of Mn_xGey compounds*. Journal of Applied Physics, 2020. **127**(24).
50. Gastegger, M., et al., *wACSF—Weighted atom-centered symmetry functions as descriptors in machine learning potentials*. The Journal of chemical physics, 2018. **148**(24).
51. Schran, C., J.r. Behler, and D. Marx, *Automated fitting of neural network potentials at coupled cluster accuracy: Protonated water clusters as testing ground*. Journal of chemical theory and computation, 2019. **16**(1): p. 88-99.

52. Litman, Y., J. Behler, and M. Rossi, *Temperature dependence of the vibrational spectrum of porphycene: a qualitative failure of classical-nuclei molecular dynamics*. Faraday Discussions, 2020. **221**: p. 526-546.
53. Behler, J., *Four generations of high-dimensional neural network potentials*. Chemical Reviews, 2021. **121**(16): p. 10037-10072.
54. Kocer, E., T.W. Ko, and J. Behler, *Neural network potentials: A concise overview of methods*. Annual review of physical chemistry, 2022. **73**(1): p. 163-186.
55. Käser, S., et al., *Neural network potentials for chemistry: concepts, applications and prospects*. Digital Discovery, 2023. **2**(1): p. 28-58.
56. Hu, L., B. Huang, and F. Liu, *Atomistic Mechanism Underlying the Si (111)-(7×7) Surface Reconstruction Revealed by Artificial Neural-Network Potential*. Physical Review Letters, 2021. **126**(17): p. 176101.
57. Kývala, L. and C. Dellago, *Optimizing the architecture of Behler–Parrinello neural network potentials*. The Journal of Chemical Physics, 2023. **159**(9).
58. Smith, J.S., O. Isayev, and A.E. Roitberg, *ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost*. Chemical science, 2017. **8**(4): p. 3192-3203.
59. Smith, J.S., et al., *Less is more: Sampling chemical space with active learning*. The Journal of chemical physics, 2018. **148**(24).
60. Smith, J.S., et al., *Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning*. Nature communications, 2019. **10**(1): p. 2903.
61. Devereux, C., et al., *Extending the applicability of the ANI deep learning molecular potential to sulfur and halogens*. Journal of Chemical Theory and Computation, 2020. **16**(7): p. 4192-4202.
62. Gao, X., et al., *TorchANI: a free and open source PyTorch-based deep learning implementation of the ANI neural network potentials*. Journal of chemical information and modeling, 2020. **60**(7): p. 3408-3415.
63. Abedi, M., J.r. Behler, and C.F. Goldsmith, *High-Dimensional Neural Network Potentials for Accurate Prediction of Equation of State: A Case Study of Methane*. Journal of Chemical Theory and Computation, 2023. **19**(21): p. 7825-7832.
64. Dianat, S.A. and R.M. Rao, *Fast algorithms for phase and magnitude reconstruction from bispectra*. Optical Engineering, 1990. **29**(5): p. 504-512.

65. Varshalovich, D.A., A.N. Moskalev, and V.K.m. Khersonskii, *Quantum theory of angular momentum*. 1988: World Scientific.
66. Behler, J., *Perspective: Machine learning potentials for atomistic simulations*. The Journal of chemical physics, 2016. **145**(17).
67. Williams, C.K. and C.E. Rasmussen, *Gaussian processes for machine learning*. Vol. 2. 2006: MIT press Cambridge, MA.
68. MacKay, D.J., *Information theory, inference and learning algorithms*. 2003: Cambridge university press.
69. Brenner, D.W., et al., *A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons*. Journal of Physics: Condensed Matter, 2002. **14**(4): p. 783.
70. Finnis, M. and J. Sinclair, *A simple empirical N-body potential for transition metals*. Philosophical Magazine A, 1984. **50**(1): p. 45-55.
71. Bartók, A.P., R. Kondor, and G. Csányi, *On representing chemical environments*. Physical Review B—Condensed Matter and Materials Physics, 2013. **87**(18): p. 184115.
72. Deringer, V.L., et al., *Origins of structural and electronic transitions in disordered silicon*. Nature, 2021. **589**(7840): p. 59-64.
73. Deringer, V.L. and G. Csányi, *Machine learning based interatomic potential for amorphous carbon*. Physical Review B, 2017. **95**(9): p. 094203.
74. Szlachta, W.J., A.P. Bartók, and G. Csányi, *Accuracy and transferability of Gaussian approximation potential models for tungsten*. Physical Review B, 2014. **90**(10): p. 104108.
75. Dragoni, D., et al., *Achieving DFT accuracy with a machine-learning interatomic potential: Thermomechanics and defects in bcc ferromagnetic iron*. Physical Review Materials, 2018. **2**(1): p. 013808.
76. Deringer, V.L., M.A. Caro, and G. Csányi, *A general-purpose machine-learning force field for bulk and nanostructured phosphorus*. Nature communications, 2020. **11**(1): p. 5461.
77. Bartók, A.P., et al., *Machine-learning approach for one-and two-body corrections to density functional theory: Applications to molecular and condensed water*. Physical Review B—Condensed Matter and Materials Physics, 2013. **88**(5): p. 054104.
78. Kloppenburg, J., et al., *Reassignment of magic numbers for icosahedral Au clusters: 310, 564, 928 and 1426*. Nanoscale, 2022. **14**(25): p. 9053-9060.

79. Kloppenburg, J., et al., *A general-purpose machine learning Pt interatomic potential for an accurate description of bulk, surfaces, and nanoparticles*. The Journal of chemical physics, 2023. **158**(13).
80. Liu, Y.-B., et al., *Machine learning interatomic potential developed for molecular simulations on thermal properties of β -Ga₂O₃*. The Journal of Chemical Physics, 2020. **153**(14).
81. Unruh, D., et al., *Gaussian approximation potential for amorphous Si: H*. Physical Review Materials, 2022. **6**(6): p. 065603.
82. de Brito Mota, F., J. Justo, and A. Fazzio, *Hydrogen role on the properties of amorphous silicon nitride*. Journal of applied physics, 1999. **86**(4): p. 1843-1847.
83. George, J., et al., *Combining phonon accuracy with high transferability in Gaussian approximation potential models*. The Journal of Chemical Physics, 2020. **153**(4).
84. Mocanu, F.C., et al., *Modeling the phase-change memory material, Ge₂Sb₂Te₅, with a machine-learned interatomic potential*. The Journal of Physical Chemistry B, 2018. **122**(38): p. 8998-9006.
85. Mocanu, F., K. Konstantinou, and S. Elliott, *Quench-rate and size-dependent behaviour in glassy Ge₂Sb₂Te₅ models simulated with a machine-learned Gaussian approximation potential*. Journal of Physics D: Applied Physics, 2020. **53**(24): p. 244002.
86. Fujikake, S., et al., *Gaussian approximation potential modeling of lithium intercalation in carbon nanostructures*. The Journal of chemical physics, 2018. **148**(24).
87. Zhang, C. and Q. Sun, *Gaussian approximation potential for studying the thermal conductivity of silicene*. Journal of Applied Physics, 2019. **126**(10).
88. Deringer, V.L., et al., *Gaussian process regression for materials and molecules*. Chemical Reviews, 2021. **121**(16): p. 10073-10141.
89. Klawohn, S., et al., *Gaussian approximation potentials: Theory, software implementation and application examples*. The Journal of Chemical Physics, 2023. **159**(17).
90. Wood, M.A. and A.P. Thompson, *Extending the accuracy of the SNAP interatomic potential form*. The Journal of chemical physics, 2018. **148**(24).

91. Zhou, X., R. Johnson, and H. Wadley, *Misfit-energy-increasing dislocations in vapor-deposited CoFe/NiFe multilayers*. Physical Review B, 2004. **69**(14): p. 144113.
92. Mishin, Y. and A. Lozovoi, *Angular-dependent interatomic potential for tantalum*. Acta materialia, 2006. **54**(19): p. 5013-5026.
93. Cusentino, M.A., M.A. Wood, and A.P. Thompson, *Explicit multielement extension of the spectral neighbor analysis potential for chemically complex systems*. The Journal of Physical Chemistry A, 2020. **124**(26): p. 5456-5464.
94. Jiang, J., et al., *Amorphous Zirconia-doped Tantala modeling and simulations using explicit multi-element spectral neighbor analysis machine learning potentials (EME-SNAP)*. Physical Review Materials, 2023. **7**(4): p. 045602.
95. Trinastic, J., et al., *Unified interatomic potential and energy barrier distributions for amorphous oxides*. The Journal of chemical physics, 2013. **139**(15).
96. Sikorski, E., et al., *Machine learned interatomic potential for dispersion strengthened plasma facing components*. The Journal of Chemical Physics, 2023. **158**(11).
97. Wang, X., et al., *Generalized stacking fault energies and Peierls stresses in refractory body-centered cubic metals from machine learning-based interatomic potentials*. Computational Materials Science, 2021. **192**: p. 110364.
98. Lin, D.-Y., et al., *An n-body potential for a Zr–Nb system based on the embedded-atom method*. Journal of Physics: Condensed Matter, 2013. **25**(10): p. 105404.
99. Deng, Z., et al., *An electrostatic spectral neighbor analysis potential for lithium nitride*. npj computational materials, 2019. **5**(1): p. 75.
100. Li, X.-G., et al., *Quantum-accurate spectral neighbor analysis potential models for Ni-Mo binary alloys and fcc metals*. Physical Review B, 2018. **98**(9): p. 094104.
101. Gu, X. and C. Zhao, *Thermal conductivity of single-layer MoS₂ (1-x) Se_{2x} alloys from molecular dynamics simulations with a machine-learning-based interatomic potential*. Computational Materials Science, 2019. **165**: p. 74-81.
102. Hastie, T., et al., *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. 2009: Springer.

103. Stewart, J.J., *Optimization of parameters for semiempirical methods V: Modification of NDDO approximations and application to 70 elements*. Journal of Molecular modeling, 2007. **13**: p. 1173-1213.
104. Rupp, M., R. Ramakrishnan, and O.A. Von Lilienfeld, *Machine learning for quantum mechanical properties of atoms in molecules*. The Journal of Physical Chemistry Letters, 2015. **6**(16): p. 3309-3313.
105. Hansen, K., et al., *Assessment and validation of machine learning methods for predicting molecular atomization energies*. Journal of chemical theory and computation, 2013. **9**(8): p. 3404-3419.
106. Hansen, K., et al., *Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space*. The journal of physical chemistry letters, 2015. **6**(12): p. 2326-2331.
107. Barker, J., et al., *Localized Coulomb descriptors for the Gaussian approximation potential*. arXiv preprint arXiv:1611.05126, 2016.
108. Collins, C.R., et al., *Constant size descriptors for accurate machine learning models of molecular properties*. The Journal of Chemical Physics, 2018. **148**(24).
109. Schrier, J., *Can one hear the shape of a molecule (from its Coulomb matrix eigenvalues)*? Journal of Chemical Information and Modeling, 2020. **60**(8): p. 3804-3811.
110. Xu, M., T. Zhu, and J.Z. Zhang, *Automated construction of neural network potential energy surface: The enhanced self-organizing incremental neural network deep potential method*. Journal of Chemical Information and Modeling, 2021. **61**(11): p. 5425-5437.
111. Novikov, I.S., et al., *The MLIP package: moment tensor potentials with MPI and active learning*. Machine Learning: Science and Technology, 2020. **2**(2): p. 025002.
112. Baştanlar, Y. and M. Özysal, *Introduction to machine learning*. miRNomics: MicroRNA biology and computational analysis, 2014: p. 105-128.
113. Shapeev, A.V., et al., *Elinvar effect in β -Ti simulated by on-the-fly trained moment tensor potential*. New Journal of Physics, 2020. **22**(11): p. 113005.
114. Shapeev, A.V., D. Bocharov, and A. Kuzmin, *Validation of moment tensor potentials for fcc and bcc metals using EXAFS spectra*. Computational Materials Science, 2022. **210**: p. 111028.

115. Kwon, H., et al., *Accurate description of hydrogen diffusivity in bcc metals using machine-learning moment tensor potentials and path-integral methods*. Acta Materialia, 2023. **247**: p. 118739.
116. Bock, F., F. Tasnadi, and I. Abrikosov, *Active learning with moment tensor potentials to predict material properties: Ti0. 5Al0. 5N at elevated temperature*. Journal of Vacuum Science & Technology A, 2024. **42**(1).
117. Chen, H., et al., *Development of a machine-learning interatomic potential for uranium under the moment tensor potential framework*. Computational Materials Science, 2023. **229**: p. 112376.
118. Podryabinkin, E., et al., *MLIP-3: Active learning on atomic environments with moment tensor potentials*. The Journal of Chemical Physics, 2023. **159**(8).
119. Novikov, I., et al., *Magnetic Moment Tensor Potentials for collinear spin-polarized materials reproduce different magnetic states of bcc Fe*. npj Computational Materials, 2022. **8**(1): p. 13.
120. Novikov, I.S., A.V. Shapeev, and Y.V. Suleimanov, *Ring polymer molecular dynamics and active learning of moment tensor potential for gas-phase barrierless reactions: Application to S+ H₂*. The Journal of chemical physics, 2019. **151**(22).
121. Attarian, S., D. Morgan, and I. Szlufarska, *Thermophysical properties of FLiBe using moment tensor potentials*. Journal of Molecular Liquids, 2022. **368**: p. 120803.
122. Novoselov, I., et al., *Moment tensor potentials as a promising tool to study diffusion processes*. Computational Materials Science, 2019. **164**: p. 46-56.
123. Luo, Y., et al., *A set of moment tensor potentials for zirconium with increasing complexity*. Journal of Chemical Theory and Computation, 2023. **19**(19): p. 6848-6856.
124. Chen, C., et al., *Graph networks as a universal machine learning framework for molecules and crystals*. Chemistry of Materials, 2019. **31**(9): p. 3564-3572.
125. Gasteiger, J., J. Groß, and S. Günnemann, *Directional message passing for molecular graphs*. arXiv preprint arXiv:2003.03123, 2020.
126. Gilmer, J., et al. *Neural message passing for quantum chemistry*. in *International conference on machine learning*. 2017. PMLR.
127. Xie, T. and J.C. Grossman, *Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties*. Physical review letters, 2018. **120**(14): p. 145301.

128. Duvenaud, D.K., et al., *Convolutional networks on graphs for learning molecular fingerprints*. Advances in neural information processing systems, 2015. **28**.
129. Kearnes, S., et al., *Molecular graph convolutions: moving beyond fingerprints*. Journal of computer-aided molecular design, 2016. **30**: p. 595-608.
130. Mueller, T., A. Hernandez, and C. Wang, *Machine learning for interatomic potential models*. The Journal of chemical physics, 2020. **152**(5).
131. Schütt, K.T., et al., *Quantum-chemical insights from deep tensor neural networks*. Nature communications, 2017. **8**(1): p. 13890.
132. Schütt, K.T., et al., *Schnet—a deep learning architecture for molecules and materials*. The Journal of Chemical Physics, 2018. **148**(24).
133. Jørgensen, P.B., K.W. Jacobsen, and M.N. Schmidt, *Neural message passing with edge updates for predicting properties of molecules and materials*. arXiv preprint arXiv:1806.03146, 2018.
134. Xie, T. and J.C. Grossman, *Hierarchical visualization of materials space with graph convolutional neural networks*. The Journal of chemical physics, 2018. **149**(17).
135. Chen, C., et al., *Learning properties of ordered and disordered materials from multi-fidelity data*. Nature Computational Science, 2021. **1**(1): p. 46-53.
136. Louis, S.-Y., et al., *Graph convolutional neural networks with global attention for improved materials property prediction*. Physical Chemistry Chemical Physics, 2020. **22**(32): p. 18141-18148.
137. Park, C.W. and C. Wolverton, *Developing an improved crystal graph convolutional neural network framework for accelerated materials discovery*. Physical Review Materials, 2020. **4**(6): p. 063801.
138. Anderson, B., T.S. Hy, and R. Kondor, *Cormorant: Covariant molecular neural networks*. Advances in neural information processing systems, 2019. **32**.
139. Thomas, N., et al., *Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds*. arXiv preprint arXiv:1802.08219, 2018.
140. Schütt, K., O. Unke, and M. Gastegger. *Equivariant message passing for the prediction of tensorial properties and molecular spectra*. in *International Conference on Machine Learning*. 2021. PMLR.

141. Batzner, S., et al., *E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials*. Nature communications, 2022. **13**(1): p. 2453.
142. Satorras, V.G., E. Hoogeboom, and M. Welling. *E (n) equivariant graph neural networks*. in *International conference on machine learning*. 2021. PMLR.
143. Thölke, P. and G. De Fabritiis, *Torchmd-net: equivariant transformers for neural network based molecular potentials*. arXiv preprint arXiv:2202.02541, 2022.
144. Brandstetter, J., et al., *Geometric and physical quantities improve e (3) equivariant message passing*. arXiv preprint arXiv:2110.02905, 2021.
145. Xue, L.-Y., et al., *ReaxFF-MPNN machine learning potential: a combination of reactive force field and message passing neural networks*. Physical chemistry chemical physics, 2021. **23**(35): p. 19457-19464.
146. Takamoto, S., S. Izumi, and J. Li, *TeaNet: Universal neural network interatomic potential inspired by iterative electronic relaxations*. Computational Materials Science, 2022. **207**: p. 111280.
147. Batatia, I., et al., *MACE: Higher order equivariant message passing neural networks for fast and accurate force fields*. Advances in Neural Information Processing Systems, 2022. **35**: p. 11423-11436.
148. Tang, M., B. Li, and H. Chen, *Application of message passing neural networks for molecular property prediction*. Current Opinion in Structural Biology, 2023. **81**: p. 102616.
149. Hernandez, A., et al., *Fast, accurate, and transferable many-body interatomic potentials by symbolic regression*. npj Computational Materials, 2019. **5**(1): p. 112.
150. Chan, H., et al., *Machine learning classical interatomic potentials for molecular dynamics from first-principles training data*. The Journal of Physical Chemistry C, 2019. **123**(12): p. 6941-6957.
151. Li, Y., et al., *Machine learning force field parameters from ab initio data*. Journal of chemical theory and computation, 2017. **13**(9): p. 4492-4503.
152. Pahari, P. and S. Chaturvedi, *Determination of best-fit potential parameters for a reactive force field using a genetic algorithm*. Journal of molecular modeling, 2012. **18**: p. 1049-1061.

153. Larsson, H.R., A.C. Van Duin, and B. Hartke, *Global optimization of parameters in the reactive force field ReaxFF for SiOH*. Journal of computational chemistry, 2013. **34**(25): p. 2178-2189.
154. Sen, F., et al., *Towards accurate prediction of catalytic activity in IrO₂ nanoclusters via first principles-based variable charge force field*. Journal of Materials Chemistry A, 2015. **3**(37): p. 18970-18982.
155. Cherukara, M.J., et al., *Ab initio-based bond order potential to investigate low thermal conductivity of stanene nanostructures*. The journal of physical chemistry letters, 2016. **7**(19): p. 3752-3759.
156. Schmidt, M. and H. Lipson, *Distilling free-form natural laws from experimental data*. science, 2009. **324**(5923): p. 81-85.
157. Mueller, T., E. Johlin, and J.C. Grossman, *Origins of hole traps in hydrogenated nanocrystalline and amorphous silicon revealed through machine learning*. Physical Review B, 2014. **89**(11): p. 115202.
158. Yuan, F. and T. Mueller, *Identifying models of dielectric breakdown strength from high-throughput data via genetic programming*. Scientific reports, 2017. **7**(1): p. 17594.
159. Slepoy, A., M. Peters, and A. Thompson, *Searching for globally optimal functional forms for interatomic potentials using genetic programming with parallel tempering*. Journal of computational chemistry, 2007. **28**(15): p. 2465-2471.
160. Brown, W.M., A.P. Thompson, and P.A. Schultz, *Efficient hybrid evolutionary optimization of interatomic potential models*. The Journal of chemical physics, 2010. **132**(2).
161. Kenoufi, A. and K. Kholmurodov, *Symbolic regression of interatomic potentials via genetic programming*. Biol. Chem. Res, 2015. **2**: p. 1-10.
162. Koza, J.R., *Genetic programming as a means for programming computers by natural selection*. Statistics and computing, 1994. **4**: p. 87-112.
163. Hansen, N. and A. Ostermeier. *Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation*. in *Proceedings of IEEE international conference on evolutionary computation*. 1996. IEEE.
164. Polak, E. and G. Ribiere, *Note sur la convergence de méthodes de directions conjuguées*. Revue française d'informatique et de recherche opérationnelle. Série rouge, 1969. **3**(16): p. 35-43.

165. Sutton, A. and J. Chen, *Long-range finnis–sinclair potentials*. Philosophical Magazine Letters, 1990. **61**(3): p. 139-146.
166. Mishin, Y., et al., *Structural stability and lattice defects in copper: Ab initio, tight-binding, and embedded-atom calculations*. Physical Review B, 2001. **63**(22): p. 224106.
167. Mendelev, M., et al., *Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu*. Philosophical Magazine, 2008. **88**(12): p. 1723-1750.
168. Onat, B. and S. Durukanoğlu, *An optimized interatomic potential for Cu–Ni alloys with the embedded-atom method*. Journal of Physics: Condensed Matter, 2013. **26**(3): p. 035404.
169. Pun, G.P., et al., *Physically informed artificial neural networks for atomistic modeling of materials*. Nature communications, 2019. **10**(1): p. 2339.
170. Gillespie, B., et al., *Bond-order potential for silicon*. Physical Review B—Condensed Matter and Materials Physics, 2007. **75**(15): p. 155207.
171. Drautz, R., et al., *Analytic bond-order potentials for modelling the growth of semiconductor thin films*. Progress in materials science, 2007. **52**(2-3): p. 196-229.
172. Chen, X., et al., *Machine learning enhanced empirical potentials for metals and alloys*. Computer Physics Communications, 2021. **269**: p. 108132.
173. Böhm, A., *Quantum mechanics: foundations and applications*. 2013: Springer Science & Business Media.
174. Yutsis, A., L. Levinson, and V. Vanagas, *Mathematical Apparatus of the Theory of Angular Momentum*, Israel Program for Sci. Transl. Ltd., Jerusalem, 1962.
175. Van Der Waerden, B.L., *Die gruppentheoretische Methode in der Quantenmechanik*. 1932.
176. Brink, D. and G. Sactchler, *Angular momentum*, Claren. 1968, don Press, Oxford.
177. Steinhardt, P.J., D.R. Nelson, and M. Ronchetti, *Bond-orientational order in liquids and glasses*. Physical Review B, 1983. **28**(2): p. 784.
178. Pettifor, D. and I. Oleinik, *Bounded analytic bond-order potentials for σ and π bonds*. Physical Review Letters, 2000. **84**(18): p. 4124.

179. Kondor, R., *A novel set of rotationally and translationally invariant features for images based on the non-commutative bispectrum*. arXiv preprint cs/0701127, 2007.
180. Qamar, M., et al., *Atomic cluster expansion for quantum-accurate large-scale simulations of carbon*. Journal of Chemical Theory and Computation, 2023. **19**(15): p. 5151-5167.
181. Rinaldi, M., et al., *Non-collinear magnetic atomic cluster expansion for iron*. npj Computational Materials, 2024. **10**(1): p. 12.
182. Liang, Y., et al., *Atomic cluster expansion for Pt–Rh catalysts: From ab initio to the simulation of nanoclusters in few steps*. Journal of Materials Research, 2023. **38**(24): p. 5125-5135.
183. Bochkarev, A., et al., *Efficient parametrization of the atomic cluster expansion*. Physical Review Materials, 2022. **6**(1): p. 013804.
184. Lysogorskiy, Y., et al., *Performant implementation of the atomic cluster expansion (PACE) and application to copper and silicon*. npj computational materials, 2021. **7**(1): p. 97.
185. Goff, J., et al., *Shadow molecular dynamics and atomic cluster expansions for flexible charge models*. Journal of Chemical Theory and Computation, 2023. **19**(13): p. 4255-4272.
186. Lysogorskiy, Y., et al., *Active learning strategies for atomic cluster expansion models*. Physical Review Materials, 2023. **7**(4): p. 043801.
187. Drautz, R. and C. Ortner, *Atomic cluster expansion and wave function representations*. arXiv preprint arXiv:2206.11375, 2022.
188. Zhou, D., et al., *A multilevel method for many-electron Schrödinger equations based on the atomic cluster expansion*. SIAM Journal on Scientific Computing, 2024. **46**(1): p. A105-A129.
189. Dusson, G., et al., *Atomic cluster expansion: Completeness, efficiency and stability*. Journal of Computational Physics, 2022. **454**: p. 110946.
190. Huo, H. and M. Rupp, *Unified representation of molecules and crystals for machine learning*. Machine Learning: Science and Technology, 2022. **3**(4): p. 045017.
191. Shang, C., S.D. Huang, and Z.P. Liu, *Massively parallelization strategy for material simulation using high - dimensional neural network potential*. Journal of Computational Chemistry, 2019. **40**(10): p. 1091-1096.

192. Nguyen, N.C. and A. Rohskopf, *Proper orthogonal descriptors for efficient and accurate interatomic potentials*. Journal of Computational Physics, 2023. **480**: p. 112030.
193. Wang, H., et al., *DeePMD-kit: A deep learning package for many-body potential energy representation and molecular dynamics*. Computer Physics Communications, 2018. **228**: p. 178-184.
194. Willatt, M.J., F. Musil, and M. Ceriotti, *Atom-density representations for machine learning*. The Journal of chemical physics, 2019. **150**(15).
195. Nguyen, T.T., et al., *Comparison of permutationally invariant polynomials, neural networks, and Gaussian approximation potentials in representing water interactions through many-body expansions*. The Journal of chemical physics, 2018. **148**(24).
196. Musil, F., et al., *Physics-inspired structural representations for molecules and materials*. Chemical Reviews, 2021. **121**(16): p. 9759-9815.
197. Yu, W., et al., *Machine - learning - based interatomic potentials for advanced manufacturing*. International Journal of Mechanical System Dynamics, 2021. **1**(2): p. 159-172.
198. Al Mamun, A., et al., *Comparing interatomic potentials in calculating basic structural parameters and Peierls stress in tungsten-based random binary alloys*. Physica Scripta, 2023. **98**(10): p. 105923.
199. Thompson, A.P., et al., *LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales*. Computer Physics Communications, 2022. **271**: p. 108171.
200. Kresse, G. and J. Furthmüller, *Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set*. Computational materials science, 1996. **6**(1): p. 15-50.
201. Kresse, G. and D. Joubert, *From ultrasoft pseudopotentials to the projector augmented-wave method*. Physical review b, 1999. **59**(3): p. 1758.
202. Abadi, M. and A.A.B.P. TensorFlow. *Large-scale machine learning on heterogeneous distributed systems*. in *Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI'16)(Savannah, GA, USA*. 2016.
203. Paszke, A., et al., *Pytorch: An imperative style, high-performance deep learning library*. Advances in neural information processing systems, 2019. **32**.

204. Singraber, A., et al., *Parallel multistream training of high-dimensional neural network potentials*. Journal of chemical theory and computation, 2019. **15**(5): p. 3075-3092.
205. Artrith, N. and A. Urban, *An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO₂*. Computational Materials Science, 2016. **114**: p. 135-150.
206. Shao, Y., et al., *PiNN: A python library for building atomic neural networks of molecules and materials*. Journal of chemical information and modeling, 2020. **60**(3): p. 1184-1193.
207. Khorshidi, A. and A.A. Peterson, *Amp: A modular approach to machine learning in atomistic simulations*. Computer Physics Communications, 2016. **207**: p. 310-324.
208. Huang, S.D., et al., *LASP: Fast global potential energy surface exploration*. Wiley Interdisciplinary Reviews: Computational Molecular Science, 2019. **9**(6): p. e1415.
209. Witt, W.C., et al., *ACEpotentials.jl: A Julia implementation of the atomic cluster expansion*. The Journal of Chemical Physics, 2023. **159**(16).
210. Larsen, A.H., et al., *The atomic simulation environment—a Python library for working with atoms*. Journal of Physics: Condensed Matter, 2017. **29**(27): p. 273002.
211. Greener, J.G., *Differentiable simulation to develop molecular dynamics force fields for disordered proteins*. Chemical Science, 2024. **15**(13): p. 4897-4909.
212. Moya, J.S., S. Lopez-Esteban, and C. Pecharromán, *The challenge of ceramic/metal microcomposites and nanocomposites*. Progress in materials science, 2007. **52**(7): p. 1017-1090.
213. Padture, N.P., M. Gell, and E.H. Jordan, *Thermal barrier coatings for gas-turbine engine applications*. Science, 2002. **296**(5566): p. 280-284.
214. Chowdhury, S., et al., *Role of interlayer thickness on interdiffusion in Ti/TiN multilayers*. Applied Surface Science, 2021. **564**: p. 150430.
215. Bonu, V., et al., *Nanolayered multilayer Ti/TiN coatings: Role of bi-layer thickness and annealing on solid particle erosion behaviour at elevated temperature*. Surface and Coatings Technology, 2019. **357**: p. 204-211.
216. Bonu, V., et al., *Solid particle erosion and corrosion resistance performance of nanolayered multilayered Ti/TiN and TiAl/TiAlN coatings deposited on Ti6Al4V substrates*. Surface and Coatings Technology, 2020. **387**: p. 125531.

217. Ghasemi, S., A. Shanaghi, and P.K. Chu, *Nano mechanical and wear properties of multi-layer Ti/TiN coatings deposited on Al 7075 by high-vacuum magnetron sputtering*. Thin Solid Films, 2017. **638**: p. 96-104.
218. Ipaz, L., et al., *Improvement of mechanical and tribological properties in steel surfaces by using titanium–aluminum/titanium–aluminum nitride multilayered system*. Applied Surface Science, 2012. **258**(8): p. 3805-3814.
219. Jannat, S., et al., *Preparation and performance of nanometric Ti/TiN multi-layer physical vapor deposited coating on 316L stainless steel as bipolar plate for proton exchange membrane fuel cells*. Journal of Power Sources, 2019. **435**: p. 226818.
220. Krella, A.K., *Cavitation erosion resistance of Ti/TiN multilayer coatings*. Surface and Coatings Technology, 2013. **228**: p. 115-123.
221. Naghibi, S., K. Raeissi, and M. Fathi, *Corrosion and tribocorrosion behavior of Ti/TiN PVD coating on 316L stainless steel substrate in Ringer's solution*. Materials Chemistry and Physics, 2014. **148**(3): p. 614-623.
222. Shukla, K., et al., *Structural, mechanical and corrosion resistance properties of Ti/TiN bilayers deposited by magnetron sputtering on AISI 316L*. Surface and Coatings Technology, 2017. **324**: p. 167-174.
223. Zhao, C., et al., *Structure and tribocorrosion behavior of Ti/TiN multilayer coatings in simulated body fluid by arc ion plating*. Surface and Coatings Technology, 2020. **403**: p. 126399.
224. Zhou, D., et al., *Microstructure, hardness and corrosion behaviour of Ti/TiN multilayer coatings produced by plasma activated EB-PVD*. Surface and Coatings Technology, 2014. **258**: p. 102-107.
225. Jiang, C., et al., *Influence of titanium interlayer thickness distribution on mechanical properties of Ti/TiN multilayer coatings*. Thin Solid Films, 2017. **632**: p. 97-105.
226. Bonu, V., et al., *Ultra-thin multilayered erosion resistant Ti/TiN coatings with stress absorbing layers*. Applied Surface Science, 2019. **478**: p. 872-881.
227. Chawla, V., R. Jayaganthan, and R. Chandra, *A study of structural and mechanical properties of sputter deposited nanocomposite Ti–Si–N thin films*. Surface and Coatings Technology, 2010. **204**(9-10): p. 1582-1589.
228. Hoerling, A., et al., *Thermal stability, microstructure and mechanical properties of $Ti_{1-x}Zr_xN$ thin films*. Thin Solid Films, 2008. **516**(18): p. 6421-6431.

229. Jarvis, E. and E. Carter, *An atomic perspective of a doped metal-oxide interface*. The Journal of Physical Chemistry B, 2002. **106**(33): p. 7995-8004.
230. Sun, T., et al., *The mechanical and electronic properties of Al/TiC interfaces alloyed by Mg, Zn, Cu, Fe and Ti: First-principles study*. Physica Scripta, 2015. **90**(3): p. 035701.
231. Xiong, H.-h., et al., *Effects of alloying elements X (X= Zr, V, Cr, Mn, Mo, W, Nb, Y) on ferrite/TiC heterogeneous nucleation interface: first-principles study*. Journal of Iron and Steel Research International, 2017. **24**(3): p. 328-334.
232. Colombo, D.A., et al., *Mechanical and tribological behavior of Ti/TiN and TiAl/TiAlN coated austempered ductile iron*. Thin Solid Films, 2018. **647**: p. 19-25.
233. Xu, C., et al., *Surface protection of a V-4Cr-4Ti alloy through a multilayered TiAl/TiAlN composite coating*. Vacuum, 2023. **207**: p. 111595.
234. Sant, C., et al., *Interface effect on tribological properties of titanium–titanium nitride nanolaminated structures*. Surface and Coatings Technology, 2000. **127**(2-3): p. 167-173.
235. Spaepen, F., *Interfaces and stresses in thin films*. Acta materialia, 2000. **48**(1): p. 31-42.
236. Dhariwal, N., et al., *Strengthening the Ti/TiN interface against shear failure with Al dopants: A molecular dynamics study*. Applied Surface Science, 2023. **613**: p. 156024.
237. Miraz, A.S.M., et al., *Computational observation of the strengthening of Cu/TiN metal/ceramic interfaces by sub-nanometer interlayers and dopants*. Applied Surface Science, 2021. **554**: p. 149562.
238. Rao, L., et al., *Interface relationship between TiN and Ti substrate by first-principles calculation*. Computational Materials Science, 2018. **155**: p. 36-47.
239. Salehinia, I., et al., *Plastic deformation of metal/ceramic nanolayered composites*. Jom, 2014. **66**: p. 2078-2085.
240. Siegel, D.J., L.G. Hector Jr, and J.B. Adams, *First-principles study of metal–carbide/nitride adhesion: Al/VC vs. Al/VN*. Acta materialia, 2002. **50**(3): p. 619-631.
241. Xu, W., et al., *First-principles calculation of Mg/MgO interfacial free energies*. Journal of Alloys and Compounds, 2015. **650**: p. 228-238.

242. Miraz, A.S.M., et al., *Improvement of Ti/TiN interfacial shear strength by doping—A first principles density functional theory study*. Applied Surface Science, 2020. **517**: p. 146185.
243. Al-Bukhaiti, M., et al., *Tribological and mechanical properties of Ti/TiAlN/TiAlCN nanoscale multilayer PVD coatings deposited on AISI H11 hot work tool steel*. Applied surface science, 2014. **318**: p. 180-190.
244. Chen, L., et al., *Thermal stability and oxidation resistance of Ti-Al-N coatings*. Surface and Coatings Technology, 2012. **206**(11-12): p. 2954-2960.
245. Butler, K.T., et al., *Machine learning for molecular and materials science*. Nature, 2018. **559**(7715): p. 547-555.
246. Ramprasad, R., et al., *Machine learning in materials informatics: recent applications and prospects*. npj Computational Materials, 2017. **3**(1): p. 54.
247. Choudhary, K. and B. DeCost, *Atomistic line graph neural network for improved materials property predictions*. npj Computational Materials, 2021. **7**(1): p. 185.
248. Popova, M., O. Isayev, and A. Tropsha, *Deep reinforcement learning for de novo drug design*. Science advances, 2018. **4**(7): p. eaap7885.
249. Rajak, P., et al., *Autonomous reinforcement learning agent for stretchable kirigami design of 2D materials*. npj Computational Materials, 2021. **7**(1): p. 102.
250. Segler, M.H., M. Preuss, and M.P. Waller, *Planning chemical syntheses with deep neural networks and symbolic AI*. Nature, 2018. **555**(7698): p. 604-610.
251. Zhou, Z., X. Li, and R.N. Zare, *Optimizing chemical reactions with deep reinforcement learning*. ACS central science, 2017. **3**(12): p. 1337-1344.
252. Li, S., et al., *Hardness prediction of high entropy alloys with machine learning and material descriptors selection by improved genetic algorithm*. Computational Materials Science, 2022. **205**: p. 111185.
253. Raju, K.V.M.K., et al., *Optimization of cutting conditions for surface roughness in CNC end milling*. International journal of precision engineering and manufacturing, 2011. **12**: p. 383-391.
254. Guo, J., *Human-level control through deep reinforcement learning*. 2016.
255. Hafner, J., *Ab - initio simulations of materials using VASP: Density - functional theory and beyond*. Journal of computational chemistry, 2008. **29**(13): p. 2044-2078.

256. Kresse, G. and J. Hafner, *Ab initio molecular-dynamics simulation of the liquid-metal–amorphous-semiconductor transition in germanium*. Physical Review B, 1994. **49**(20): p. 14251.
257. Kresse, G. and J. Hafner, *Ab initio molecular dynamics for liquid metals*. Physical review B, 1993. **47**(1): p. 558.
258. Perdew, J.P., K. Burke, and M. Ernzerhof, *Generalized gradient approximation made simple*. Physical review letters, 1996. **77**(18): p. 3865.
259. Blöchl, P.E., *Projector augmented-wave method*. Physical review B, 1994. **50**(24): p. 17953.
260. Kresse, G. and J. Furthmüller, *Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set*. Physical review B, 1996. **54**(16): p. 11169.
261. Monkhorst, H.J. and J.D. Pack, *Special points for Brillouin-zone integrations*. Physical review B, 1976. **13**(12): p. 5188.
262. Zhang, M.-X., et al., *Crystallography of the simple HCP/FCC system*. Metallurgical and Materials Transactions A, 2008. **39**(5): p. 1077-1086.
263. Zhang, M.-X. and P. Kelly, *Edge-to-edge matching and its applications: Part II. Application to Mg–Al, Mg–Y and Mg–Mn alloys*. Acta materialia, 2005. **53**(4): p. 1085-1096.
264. Jain, A., et al., *The materials project: a materials genome approach to accelerating materials innovation*. APL Mater 1: 011002. 2013.
265. Ma, L., et al., *First-principles investigation of Sn₉Zn (0 0 0 1)/a-Al₂O₃ (0 0 0 1) interfacial adhesion*. Applied Surface Science, 2018. **435**: p. 863-869.
266. Miraz, A.S.M., et al., *Computational study of metal/ceramic interfacial adhesion and barriers to shear displacement*. Computational Materials Science, 2019. **168**: p. 104-115.
267. Holland, J.H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1992: MIT press.
268. Eiben, A., et al., *Fitness, Selection, and Population Management*. Introduction to Evolutionary Computing, 2015: p. 79-98.
269. McCloskey, M. and N.J. Cohen, *Catastrophic interference in connectionist networks: The sequential learning problem*, in *Psychology of learning and motivation*. 1989, Elsevier. p. 109-165.

270. Ratcliff, R., *Connectionist models of recognition memory: constraints imposed by learning and forgetting functions*. Psychological review, 1990. **97**(2): p. 285.
271. Kirkpatrick, J., et al., *Overcoming catastrophic forgetting in neural networks*. Proceedings of the national academy of sciences, 2017. **114**(13): p. 3521-3526.
272. Thrun, S. and M.L. Littman, *Reinforcement learning: an introduction*. AI Magazine, 2000. **21**(1): p. 103-103.
273. Liao, Y., *Practical electron microscopy and database*. An Online Book, 2006.
274. Batsanov, S.S., *Van der Waals radii of elements*. Inorganic materials, 2001. **37**(9): p. 871-885.
275. Wang, V., et al., *VASPKIT: A user-friendly interface facilitating high-throughput computing and analysis using VASP code*. Computer Physics Communications, 2021. **267**: p. 108033.
276. Henkelman, G., A. Arnaldsson, and H. Jónsson, *A fast and robust algorithm for Bader decomposition of charge density*. Computational Materials Science, 2006. **36**(3): p. 354-360.
277. Sanville, E., et al., *Improved grid - based algorithm for Bader charge allocation*. Journal of computational chemistry, 2007. **28**(5): p. 899-908.
278. Tang, W., E. Sanville, and G. Henkelman, *A grid-based Bader analysis algorithm without lattice bias*. Journal of Physics: Condensed Matter, 2009. **21**(8): p. 084204.
279. Yu, M. and D.R. Trinkle, *Accurate and efficient algorithm for Bader charge integration*. The Journal of chemical physics, 2011. **134**(6).
280. Allred, A.L., *Electronegativity values from thermochemical data*. Journal of inorganic and nuclear chemistry, 1961. **17**(3-4): p. 215-221.
281. Chang, H., et al., *Novel Si-added CrCoNi medium entropy alloys achieving the breakthrough of strength-ductility trade-off*. Materials & Design, 2021. **197**: p. 109202.
282. Shi, P., et al., *Enhanced strength–ductility synergy in ultrafine-grained eutectic high-entropy alloys by inheriting microstructural lamellae*. Nature communications, 2019. **10**(1): p. 489.
283. Koga, G.Y., et al., *Corrosion resistant and tough multi-principal element Cr-Co-Ni alloys*. Journal of Alloys and Compounds, 2021. **884**: p. 161107.

284. Roychowdhury, S., et al., *Enhanced atomic ordering leads to high thermoelectric performance in AgSbTe₂*. Science, 2021. **371**(6530): p. 722-727.
285. Jiang, B., et al., *High-entropy-stabilized chalcogenides with high thermoelectric performance*. Science, 2021. **371**(6531): p. 830-834.
286. Ding, J., et al., *Tunable stacking fault energies by tailoring local chemical order in CrCoNi medium-entropy alloys*. Proceedings of the National Academy of Sciences, 2018. **115**(36): p. 8919-8924.
287. Tamm, A., et al., *Atomic-scale properties of Ni-based FCC ternary, and quaternary alloys*. Acta Materialia, 2015. **99**: p. 307-312.
288. Jin, X., S. Chen, and T. Li, *Coexistence of two types of short-range order in Si–Ge–Sn medium-entropy alloys*. Communications Materials, 2022. **3**(1): p. 66.
289. Zhang, R., et al., *Short-range order and its impact on the CrCoNi medium-entropy alloy*. Nature, 2020. **581**(7808): p. 283-287.
290. Chen, X., et al., *Direct observation of chemical short-range order in a medium-entropy alloy*. Nature, 2021. **592**(7856): p. 712-716.
291. Lei, Z., et al., *Enhanced strength and ductility in a high-entropy alloy via ordered oxygen complexes*. Nature, 2018. **563**(7732): p. 546-550.
292. Ding, Q., et al., *Tuning element distribution, structure and properties by composition in high-entropy alloys*. Nature, 2019. **574**(7777): p. 223-227.
293. Guo, L., et al., *Short-range ordering induced serrated flow in a carbon contained FeCoCrNiMn high entropy alloy*. Micron, 2019. **126**: p. 102739.
294. Seol, J.B., et al., *Short-range order strengthening in boron-doped high-entropy alloys for cryogenic applications*. Acta Materialia, 2020. **194**: p. 366-377.
295. Wu, Y., et al., *Short-range ordering and its effects on mechanical properties of high-entropy alloys*. Journal of Materials Science & Technology, 2021. **62**: p. 214-220.
296. Cao, B., et al., *Short-range order in GeSn alloy*. ACS Applied Materials & Interfaces, 2020. **12**(51): p. 57245-57253.
297. Jin, X., S. Chen, and T. Li, *Short-range order in SiSn alloy enriched by second-nearest-neighbor repulsion*. Physical Review Materials, 2021. **5**(10): p. 104606.

298. Manna, S., et al., *Learning in continuous action space for developing high dimensional potential energy models*. Nature communications, 2022. **13**(1): p. 368.
299. Metropolis, N., et al., *Equation of state calculations by fast computing machines*. The journal of chemical physics, 1953. **21**(6): p. 1087-1092.
300. Ceperley, D.M. and B.J. Alder, *Ground state of the electron gas by a stochastic method*. Physical review letters, 1980. **45**(7): p. 566.
301. Cowley, J.M., *An approximate theory of order in alloys*. Physical Review, 1950. **77**(5): p. 669.
302. Hochreiter, S., *Long Short-term Memory*. Neural Computation MIT-Press, 1997.
303. Mnih, V., *Asynchronous Methods for Deep Reinforcement Learning*. arXiv preprint arXiv:1602.01783, 2016.
304. He, Q. and Y. Yang, *On lattice distortion in high entropy alloys*. Frontiers in Materials, 2018. **5**: p. 42.
305. Van de Walle, A., et al., *Efficient stochastic generation of special quasirandom structures*. Calphad, 2013. **42**: p. 13-18.
306. Zunger, A., et al., *Special quasirandom structures*. Physical review letters, 1990. **65**(3): p. 353.
307. Shimura, Y., et al., *EXAFS study of local structure contributing to Sn stability in Si_yGe_{1-y}-zSn_z*. Materials Science in Semiconductor Processing, 2017. **70**: p. 133-138.
308. Matsutani, R., K. Sueoka, and E. Kamiyama, *First principles analysis of atomic configurations of group IV elements in Ge crystal for solar cells*. physica status solidi (c), 2014. **11**(11 - 12): p. 1718-1721.
309. Mukherjee, S., et al., *Short-range atomic ordering in nonequilibrium silicon-germanium-tin semiconductors*. Physical Review B, 2017. **95**(16): p. 161402.
310. Moutanabbir, O., et al., *Monolithic infrared silicon photonics: The rise of (Si) GeSn semiconductors*. Applied Physics Letters, 2021. **118**(11).
311. Jifeng, L., et al., *Atomic short-range order in SiGeSn alloys*. 2024.
312. Hafner, J., *Atomic-scale computational materials science*. Acta Materialia, 2000. **48**(1): p. 71-92.

313. Vakhrushev, A.V., *Introductory Chapter: Molecular Dynamics: Basic Tool of Nanotechnology Simulations for “Production 4.0” Revolution*, in *Molecular Dynamics*. 2018, IntechOpen.
314. Hermann, A. and D. Kurzydłowski, *First-principles prediction of structures and properties in crystals*. 2019, MDPI. p. 463.
315. Sholl, D.S. and J.A. Steckel, *Density functional theory: a practical introduction*. 2022: John Wiley & Sons.
316. Pettifor, D. and I. Oleinik, *Analytic bond-order potential for open and close-packed phases*. Physical Review B, 2002. **65**(17): p. 172103.
317. Murdick, D., et al., *Analytic bond-order potential for the gallium arsenide system*. Physical Review B, 2006. **73**(4): p. 045206.
318. Stuart, S.J., A.B. Tutein, and J.A. Harrison, *A reactive potential for hydrocarbons with intermolecular interactions*. The Journal of chemical physics, 2000. **112**(14): p. 6472-6486.
319. O’Connor, T.C., J. Andzelm, and M.O. Robbins, *AIREBO-M: A reactive model for hydrocarbons at extreme pressures*. The Journal of chemical physics, 2015. **142**(2): p. 024903.
320. Justo, J.F., et al., *Interatomic potential for silicon defects and disordered phases*. Physical review B, 1998. **58**(5): p. 2539.
321. Los, J. and A. Fasolino, *Intrinsic long-range bond-order potential for carbon: Performance in Monte Carlo simulations of graphitization*. Physical Review B, 2003. **68**(2): p. 024107.
322. Lennard-Jones, J.E., *Cohesion*. Proceedings of the Physical Society, 1931. **43**(5): p. 461.
323. Vasudevan, R.K., et al., *Materials science in the artificial intelligence age: high-throughput library generation, machine learning, and a pathway from correlations to the underpinning physics*. MRS Communications, 2019. **9**(3): p. 821-838.
324. Choudhary, K., et al., *Recent advances and applications of deep learning methods in materials science*. npj Computational Materials, 2022. **8**(1): p. 59.
325. Behler, J., *Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations*. Physical Chemistry Chemical Physics, 2011. **13**(40): p. 17930-17955.

326. Wood, M.A. and A.P. Thompson, *Extending the accuracy of the SNAP interatomic potential form*. The Journal of chemical physics, 2018. **148**(24): p. 241721.
327. Mueller, T., A. Hernandez, and C. Wang, *Machine learning for interatomic potential models*. The Journal of chemical physics, 2020. **152**(5): p. 050902.
328. Fung, V., et al., *Benchmarking graph neural networks for materials chemistry*. npj Computational Materials, 2021. **7**(1): p. 84.
329. Brown, W.M., A.P. Thompson, and P.A. Schultz, *Efficient hybrid evolutionary optimization of interatomic potential models*. The Journal of chemical physics, 2010. **132**(2): p. 024108.
330. Khusenov, M. and K.T. Kholmurodov. *ON MOLECULAR DYNAMICS MODELING OF THE DNA–CNT INTERACTION PROCESSES: THE DIAGNOSTIC APPLICATION AND SIMULATION ASPECTS*. in *7th RUSSIAN-JAPANESE INTERNATIONAL WORKSHOP MSSMBS'14 “Molecular*. 2014.
331. Pun, G.P., et al., *Development of a general-purpose machine-learning interatomic potential for aluminum by the physically informed neural network method*. Physical Review Materials, 2020. **4**(11): p. 113807.
332. Lin, Y.-S., G.P.P. Pun, and Y. Mishin, *Development of a physically-informed neural network interatomic potential for tantalum*. Computational Materials Science, 2022. **205**: p. 111180.
333. Choudhary, K., et al., *Evaluation and comparison of classical interatomic potentials through a user-friendly interactive web-interface*. Scientific data, 2017. **4**(1): p. 1-12.
334. Choudhary, K., et al., *High-throughput assessment of vacancy formation and surface energies of materials using classical force-fields*. Journal of Physics: Condensed Matter, 2018. **30**(39): p. 395901.
335. Andreas, H., *The era of carbon allotropes*. Nature materials, 2010. **9**: p. 868.
336. Irifune, T., Kurio A. Sakamoto S. Inoue T. Sumiya H. Nature, 2003. **421**: p. 599-600.
337. Chen, X. and J. Li, *Superlubricity of carbon nanostructures*. Carbon, 2020. **158**: p. 1-23.
338. Badenhorst, H., *A review of the application of carbon materials in solar thermal energy storage*. Solar Energy, 2019. **192**: p. 35-68.

339. Jain, A., et al., *Commentary: The Materials Project: A materials genome approach to accelerating materials innovation*. APL materials, 2013. **1**(1): p. 011002.
340. Plimpton, S., *Fast parallel algorithms for short-range molecular dynamics*. Journal of computational physics, 1995. **117**(1): p. 1-19.
341. Breiman, L., *Random forests*. Machine learning, 2001. **45**: p. 5-32.
342. Drucker, H. *Improving regressors using boosting techniques*. in *Icml*. 1997. Citeseer.
343. Friedman, J.H., *Stochastic gradient boosting*. Computational statistics & data analysis, 2002. **38**(4): p. 367-378.
344. Chen, T. and C. Guestrin. *Xgboost: A scalable tree boosting system*. in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.
345. Breiman, L., et al., *Classification and regression trees (Wadsworth, Belmont, CA)*. ISBN-13, 1984: p. 978-0412048418.
346. Jiang, C., D. Morgan, and I. Szlufarska, *Carbon tri-interstitial defect: A model for the D II center*. Physical Review B, 2012. **86**(14): p. 144118.
347. Zhou, X., D.K. Ward, and M.E. Foster, *An analytical bond - order potential for carbon*. Journal of computational chemistry, 2015. **36**(23): p. 1719-1735.
348. Morse, P.M., *Diatomique molecules according to the wave mechanics. II. Vibrational levels*. Physical review, 1929. **34**(1): p. 57.
349. Kinaci, A., et al., *Thermal conductivity of BN-C nanostructures*. Physical Review B, 2012. **86**(11): p. 115410.
350. De Jong, M., et al., *Charting the complete elastic properties of inorganic crystalline compounds*. Scientific data, 2015. **2**(1): p. 1-13.
351. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python*. the Journal of machine Learning research, 2011. **12**: p. 2825-2830.
352. Breiman, L., *Bagging predictors*. Machine learning, 1996. **24**: p. 123-140.
353. Lee, B.-J., et al., *The modified embedded-atom method interatomic potentials and recent progress in atomistic simulations*. Calphad, 2010. **34**(4): p. 510-522.
354. Ma, T., et al., *Effect of impact angle and substrate roughness on growth of diamondlike carbon films*. Journal of applied physics, 2007. **101**(1).

355. Rasmussen, C.E. and H. Nickisch, *Gaussian processes for machine learning (GPML) toolbox*. The Journal of Machine Learning Research, 2010. **11**: p. 3011-3015.
356. Pan, H., et al., *Benchmarking coordination number prediction algorithms on inorganic crystal structures*. Inorganic chemistry, 2021. **60**(3): p. 1590-1603.
357. Shan, T.-R., et al., *Charge-optimized many-body potential for the hafnium/hafnium oxide system*. Physical Review B, 2010. **81**(12): p. 125328.
358. Mumetoh, S., *Interatomic potential for Si-O systems using Tersoff parameterization Comp. Mat. Sci*, 2007. **39**: p. 334-339.
359. Broughton, J.Q., et al., *Direct atomistic simulation of quartz crystal oscillators: Bulk properties and nanoscale devices*. Physical Review B, 1997. **56**(2): p. 611.
360. Troyanskaya, O., et al., *Missing value estimation methods for DNA microarrays*. Bioinformatics, 2001. **17**(6): p. 520-525.

ProQuest Number: 31636819

INFORMATION TO ALL USERS

The quality and completeness of this reproduction is dependent on the quality
and completeness of the copy made available to ProQuest.



Distributed by

ProQuest LLC a part of Clarivate (2024).

Copyright of the Dissertation is held by the Author unless otherwise noted.

This work is protected against unauthorized copying under Title 17,
United States Code and other applicable copyright laws.

This work may be used in accordance with the terms of the Creative Commons license
or other rights statement, as indicated in the copyright statement or in the metadata
associated with this work. Unless otherwise specified in the copyright statement
or the metadata, all rights are reserved by the copyright holder.

ProQuest LLC
789 East Eisenhower Parkway
Ann Arbor, MI 48108 USA