공학박사 학위논문

Expanding the application range of neural network potential by uniform training and uncertainty estimation

균일한 학습과 불확실성 추정에 의한 인공 신경망 퍼텐셜 적용범위의 확장

2021년 2월
Feb. 2021

서울대학교 대학원
재료공학부
정 원 석
JEONG WONSEOK

Expanding the application range of neural network potential by uniform training and uncertainty estimation

균일한 학습과 불확실성 추정에 의한 인공 신경망 퍼텐셜 적용범위의 확장

지도교수 한 승 우

이 논문을 공학박사 학위논문으로 제출함

2021년 2월

서울대학교 대학원

재료공학부

정 원 석

정원석의 공학박사 학위 논문을 인준함

2021년 2월

위 원 장: 이 명 규 M.G.Lee

부위원장: 한 승 우 Seungwu Han

위 원: 최 인 석 최인석

위 원: 장 현 주

위 원: 한 상 수 Han Sang Soo

Abstract

The machine learning approach to develop interatomic potential attracts considerable attention because they can achieve simulation accuracy comparable to the reference first-principles calculations, with a much lower calculation cost. Of the many choices of machine learning potentials, high-dimensional neural network potential is highly anticipated due to its successful demonstrations in a wide range of materials, including metals, oxides, semiconductors, and molecular reactions. Despite its success and attractiveness, machine learning potentials are often regarded as a black-box method, and efforts to understand the basic foundation are lacking. The difficulty in understanding machine learning potentials stems from the difference in traits that distinguish it from the traditional classical potentials. Due to the fundamental differences, machine learning potentials present unique challenges that must be overcome to achieve high accuracy when used.

In this dissertation, we address three significant challenges of machine learning potentials and, with neural network potential, in particular, suggest the solution to the challenges. First, we discuss an unbalanced training problem coming from a biased distribution of training points. We provide various examples of biased sampling and how it undermines the accuracy of the simulation. Using the Gaussian density function that quantifies the sparsity of training points, we propose a weighting scheme to solve the unbalanced training problem. Next, we focus on the establishment of prediction uncertainty indicators. Because machine learning potentials do not have physics-based functions like conventional classical potentials, their reliability can be questionable. Therefore, a prediction uncertainty indicator is essential for machine learning potentials. The uncertainty indicator should have atomic-level resolution to identify the exact local atomic environment lacking in the training set. To this end, we propose a replica ensemble method that can ensure the atomic-level resolution of uncertainty estimation by excluding uncertainties arising from atomic energy mapping. We demonstrate this method to run molecular dynamics simulations of the Ni-Si interface reaction. Finally, we touch on one of the grand challenges in machine learning potential application, which is applying machine learning potentials to search for new crystal structures without any preceding information other than the chemical composition. In usual practices, machine learning potentials are first trained over structures derived from known structures. However, such information is not available at the outset in exploring unknown crystals. As we will address that machine learning potentials can effectively map the atomic energies from the reference total energy, we find that they can sample diverse local orders that can appear in crystals from a training set composed only of disordered structures. We prove this analogy on four different multinary crystal systems over experimental phases as well as low-energy crystal structures. By addressing and overcoming the inherent challenges in machine learning potentials, this dissertation will extend the application range of the machine learning potentials.

keywords: machine learning potential, neural network potential, molecular dynamics

student number: 2015-20862

Contents

Abstract i

Contents iii

List of Tables vi

List of Figures vii

1 Introdunction 1

1.1 Overview of machine learning potential 1

1.2 Goal of the dissertation 4

1.3 Organization of the dissertation 5

2 Theoretical background 6

2.1 Density functional theory 6

2.1.1 Born-Oppenheimer approximation 6

2.1.2 Hohenberg-Kohn theorem 8

2.1.3 Kohn-Sham equation 9

2.1.4 Exchange-correlation energy 12

2.2 Classical potentials 13

2.3 Machine learning potentials 17

2.3.1 Models 19

2.3.2 Descriptors 27

2.3.3 The training techniques for NNP 34
2.3.4 Atomic energy mapping of machine learning potentials 37

3 Unbalanced training problem 41

3.1 Introduction 41
3.2 Examples of sampling bias 42
3.3 GDF weighting scheme 45
3.4 Summary 56

4 Prediction uncertainty quantification 57

4.1 Introduction 57
4.2 Replica ensemble 59
4.3 Application: Ni silicidation simulation 61
4.4 Training set and validation of reference NNP 63
4.5 Large-scale Ni silicidation simulation 70
4.6 Replica NNP training 74
4.7 Uncertainty quantification on Ni silicidation simulation 77
4.8 Schottky barrier height estimation 82
4.9 Summary 84

5 Crystal structure prediction by machine learning potentials 85

5.1 Introduction 85
5.2 Test materials 88
5.3 Construction of training set 90
5.4 Crystal structure prediction by genetic algorithm 91
5.5 Comparison of NNP and DFT energies 92
5.6 Summary 96

6 Conclusion 97

Bibliography 99

Abstract (In Korean) 109

Acknowlegement 112

List of Tables

3.1 Formation energies of vacancy clusters in eV calculated by DFT, NNP-c, and NNP-GDF. 52

4.1 Summary of Ni–Si interface reaction reference structures and root-mean-square errors (RMSEs) for the validation set 64

List of Figures

2.1 The schematic description of the self-consistency method for the Kohn-Sham equations. 11

2.2 The citation number of seminal machine learning potential papers per year. 18

2.3 Schematic figure of a feed-forward neural network. 21

2.4 Schematic figure of a high-dimensional neural network potential . . . 23

2.5 The shapes of cutoff function and radial symmetry functions 29

2.6 The shapes of angular symmetry functions 31

3.1 An example of sampling bias in Si bulk with a vacancy. 44

3.2 Example of GDF weighting scheme applied to Si bulk with a vacancy. 47

3.3 An example of sampling bias in Si bulk with an interstitial. 48

3.4 An example of sampling bias in potential energy surface around Si interstitial. 49

3.5 Comparison of the MD trajectories obtained by conventional NNP and NNP with GDF weighting 50

3.6 An example of sampling bias in Pd(111) surface with one oxygen adsorbate. 53

3.7 Schematic diagram of training uncertainty 55

4.1 Schematic diagram of replica NNP. 60

4.2 The training set structures of Ni crystal, Ni(001) surface slab, Si crystal, and Si(001) surface slab 65
4.3 The training set structures of Ni silicides. 66
4.4 The training set structures of Ni–Si interfaces. 67
4.5 Comparison of the DFT and NNP total energies per atom along MD trajectories for a small Ni–Si interface structure. 69
4.6 Model structure of an 8.5-nm-thick Ni/Si(001) interface slab. 71
4.7 Snapshots and concentration profiles of Ni and Si with respect to z from the large-scale NNP MD trajectory. 73
4.8 The training set and validation set RMSE of atomic energy during the replica NNP training iterations. 76
4.9 Snapshots from the Ni-silicidation MD trajectory with the color scale by σ_{at} 78
4.10 Sapshots and concentration profiles of Ni and Si with respect to z from the large-scale r-NNP MD trajectory 81
4.11 Model structure of the Ni_xSi_y/Si interface slab for the Schottky barrier height estimation. 83

5.1 Unit cells of the stable phase of Ba_2AgSi_3, Mg_2SiO_4, $LiAlCl_4$, and $InTe_2O_5F$ 89
5.2 Correlation between DFT and NNP energies for USPEX metastable structures. 93
5.3 The distribution of **G** vectors in Ba_2AgSi_3 projected onto the first two principal component axes. 95

Chapter 1

Introdunction

1.1 Overview of machine learning potential

Machine learning approaches to develop interatomic potential attract considerable attention as they are anticipated to overcome the inherent shortcoming of the classical potential and density functional theory (DFT). Namely, classical potentials suffer from limited accuracy and difficulty in development, and DFT demands a substantial computational cost. On the other hand, by learning structure-energy relationships directly from first-principle calculations through very flexible functional forms, machine learning potentials can achieve the simulation accuracy comparable to the reference calculations, but at a much lower computational cost scaling linearly with the number of atoms.

To date, several types of machine learning potential have been suggested, including spectral neighbor analysis potential (SNAP),[1] moment tensor potential (MTP),[2] gradient-domain machine-learning (GDML),[3] deep tensor network (DTNN),[4] and accurate neural network engine for molecular energies (AN1).[5] Among the machine-learning potentials, two types are the most popular in material researches: Gaussian-approximation potential (GAP)[6] and high-dimensional neural network potential (NNP).[7] In particular, the high-dimensional neural network potential suggested by Behler and

Parrinello is attracting broad interests with applications demonstrated over various materials. Some of the most advanced simulations carried successfully by NNP include metastable structures search of Pt-H nanocluster,[8] CO_2 reduction active site search on Ni-Ga bimetallic facets,[9] and crystallization simulations of amorphous GeTe,[10] to name a few.

Despite their success and attraction, machine learning potentials are often considered black-box methods, and there is a significant lack of effort to understand their fundamental domain. The difficulty of understanding the functioning of machine learning potentials comes from its distinct characteristics-difference from conventional classical potentials. That is to say, the machine-learning potentials do not have prefixed functional forms, and they try to mimic the structure-energy relationships of a given training set. Also, unlike conventional classical potentials, which are based on the approximation of chemical bonds, machine learning potentials assume atomic energies given according to the local atomic environments. Owing to the philosophical differences at fundamental level, machine learning potentials poise unique challenges to overcome for reliable accuracy upon use.

One of the most severe problems is the unbalanced training of machine learning potentials, which originates from the highly biased distribution of training points. From various examples, we found that unbalanced training significantly undermines the accuracy and reliability of machine learning potentials. Moreover, as typical first-principle calculations only provide the given system's total energy rather than the individual energy of atoms, the unbalanced training problem challenges us to produce a specialized solution. Such problems are less acute with traditional classical potentials for which the developer can explicitly adjust the underlying functions.

Another of the most urgent challenges of machine learning potential is establishing simulation uncertainty indicators with atomic-level resolution. As machine-learning potentials lack the physics-based functional forms, their prediction uncertainty grows rapidly as input features deviate from the training domain. In other words, the accuracy of the energy prediction degrades unacceptably if given local atomic configurations are substantially different from those in the training set. If this happens during molecular dynamics (MD) simulations, the computational results may not be entirely meaningful even if the simulation terminates without any severe failures, such as diverging energies. Therefore unlike in conventional classical potentials where principle-based, hard-coded functions can safeguard atomic configurations against unintended structures, monitoring uncertainty levels during atomic simulations when utilizing machine learning potentials is essential. The uncertainty quantification should be in atomic-level resolution to secure the information of which exact local atomic environment among the large-scale simulation cell is absent in the training set in order that the information can be utilized in the refinement of the potential. Although some methods suggested estimating the prediction uncertainty in machine learning potentials, such as posterior predictive variances in GAP and ensemble methods in NNP, those methods can not guarantee the identification of problematic configurations with clear atomic-resolution.

Outside of MD simulations, another significant application of machine learning potentials is fast energy evaluation in crystal structure prediction (CSP). Prediction of the stable crystal structure for multinary (ternary or higher) compounds demands fast and accurate evaluation of free energies in exploring the vast configurational space. Therefore, machine learning potentials are poised to meet the requirement for evaluating energies in multinary CSP. However, there is a distinctive challenge when machine learning potentials are applied to search for new crystal structures. The heart of the machine learning potentials is the gathering of a proper training set. However, such information is not available at the outset in exploring unknown crystals, and one should construct the potentials out of nothing. Therefore, there is an acute need for the strategy to build a training set that can encompass the diverse local orders that can appear in crystals without any prior knowledge other than the chemical composition.

1.2 Goal of the dissertation

The main goal of this dissertation is to address three major challenges of machine learning potential and, with neural network potential in particular, suggest the solution to the challenge. To adequately address the uniqueness of the challenges, we also provide an overview of density functional theory, classical interatomic potentials, and machine learning potentials. From the overview, a hint can be found on why machine learning potentials inherently possess the challenges mentioned above.

The first challenge to be addressed is the unbalanced training problem caused by sampling bias. We define the Gaussian density function to quantify the sampling bias, and by using the function, we suggest a weighting scheme to remedy the unbalanced training problem. Next, we turn the topic to prediction uncertainty quantification with atomic-level resolution. We discuss the importance of uncertainty indicator and suggest a "replica" ensemble method that can secure the atomic-level resolution by excluding the uncertainty coming from atomic energy mapping. In addition, we demonstrate these methods to molecuar dynamics simulation of Ni–Si interface reaction. Finally, we provide a strategy to train a machine learning potential for crystal structure space exploration without any preceding information on material structures except for the chemical composition. The key strategy is to train a machine learning potential over disordered structures such as liquid and melt-quenched amorphous phases.

Through the discussion, we present a perspective on the unique foundation of the machine learning potentials differentiated from conventional classical potentials and contribute to expanding the application range of machine learning potentials in general.

1.3 Organization of the dissertation

This dissertation consists of six chapters. Chapter 1 is the introduction, in which we give an overview of machine learning potentials, the goal and the organization of this dissertation. Chapter 2 provides the theoretical background on density functional theory, classical interatomic potentials, and machine learning potentials. In Chapter 3, we discuss the unbalanced training problem of machine learning potentials caused by sampling bias. From the discussion, we derive a method to quantify the sampling bias. Then, a weighting scheme is suggested as the solution to the unbalanced training problem. With examples of defective silicon systems, we demonstrate the proposed scheme. In Chapter 4, we address the necessity of a prediction uncertainty indicator for machine learning simulations. Next, we present an efficient and atomic-resolution uncertainty indicator for NNP based on a replica ensemble. The demonstration of the replica ensemble-based indicator is carried on highly reactive MD simulation of Ni–Si interface reaction. Chapter 5 provides a strategy to build a training set for crystal structure prediction of multinary systems. As no information other than chemical formula is available at the outset in exploring unknown crystals, we suggest building a training set out of disordered structures and demonstrate its effectiveness for four different multinary systems. Finally, in Chapter 6, we summarize and conclude each study.

Chapter 2

Theoretical background

2.1 Density functional theory

2.1.1 Born-Oppenheimer approximation

To calculate the properties of materials at a quantum mechanical level without any empirical parameters, one should solve the Schrödinger equation; the fundamental equation that describes the state of atoms. For systems with multiple nuclei and electrons, the (time-independent) Schrödinger equation is described as:

$$H\Psi = E\Psi, \tag{2.1}$$

and the Hamiltonian operator *H* is:

$$\begin{aligned} H = &-\frac{\hbar^2}{2m_e}\sum_i \nabla_i^2 - \sum_I \frac{\hbar^2}{2M_I}\nabla_I^2 \\ &+ \frac{1}{2}\sum_{i\neq I}\frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_{I\neq J}\frac{Z_I Z_J e^2}{|\mathbf{r}_I - \mathbf{r}_J|} + \sum_{i,I}\frac{Z_I e^2}{|\mathbf{r}_i - \mathbf{r}_I|}, \end{aligned} \tag{2.2}$$

where *i*, *j* is the electron index, *I*, *J* is the atom index, *Z* is the charge, and *M* is the mass of nucleus. In Eq. (2.2), the first two terms correspond to the kinetic energy of electrons and nuclei, and the next three terms represent electron-electron, nucleus-nucleus, electron-nucleus interactions, respectively.

As the electron mass is negligible compared to the nucleus mass, the position of the nuclei can be considered fixed. Thus the kinetic energy of the nuclei can be ignored, and the nuclei merely act as the source of external potential on the electrons. This approximation is called the Born-Oppenheimer approximation,[11] and H becomes:

$$H = -\frac{\hbar^2}{2m_e}\sum_i \nabla_i^2 + \sum_i V_{\text{ext}}(\mathbf{r}_i) + \frac{1}{2}\sum_{i \neq I} \frac{e^2}{|\mathbf{r}_i - \mathbf{r}_j|} + \frac{1}{2}\sum_{I \neq J} \frac{Z_I Z_J e^2}{|\mathbf{r}_I - \mathbf{r}_J|}, \tag{2.3}$$

where V_{ext} is the external potential. For practical materials systems (i.e. many-body systems), solving Eq. (2.3) is not feasible. Therefore, several well-working approximation methodologies have been developed. One of the most successful approximation is density functional theory (DFT).

2.1.2 Hohenberg-Kohn theorem

Density functional theory is based on the Hohenberg-Kohn theorem.[12] Hohenberg-Kohn theorem states that 1. for a system of electrons in an external potential $V_{\text{ext}}(\mathbf{r})$, the total energy E is given exactly as a functional of the electron density $\rho(\mathbf{r})$, and 2. the electron density that minimizes the energy functional $E[\rho]$ is the ground state density ρ_0, with which other ground-state properties are also given. With the Hohenberg-Kohn theorem, the energy functional $E[\rho]$ can be expressed as:

$$E[\rho] = \int V_{\text{ext}}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + F[\rho], \tag{2.4}$$

with

$$F[\rho] = T[\rho] + V_{\text{ee}}[\rho] \tag{2.5}$$

where $T[\rho]$ is the kinetic energy and $V_{\text{ee}}[\rho]$ is the electron-electron interaction energy. While Hohenberg-Kohn theorem states that $E[\rho]$ exists, it does not provide the actual form of the functional. Thus further development is required to solve the many-body Hamiltonian Eq. (2.3)

2.1.3 Kohn-Sham equation

To bypass the many-body problem, Kohn and Sham assumed that the electron density $\rho(\mathbf{r})$ of N electrons can be expressed as a sum of one-electron orbital ϕ_i: [13]

$$\rho(\mathbf{r}) = \sum_{i=1}^{N} |\phi_i(\mathbf{r})|^2. \tag{2.6}$$

The one-electron orbital $\phi_i(\mathbf{r})$ is also known as the Kohn-Sham orbital. With this assumption, the multi-electron problem becomes the set of one-electron problems. Thus the Kohn-Sham equation is given as:

$$\left(-\frac{\hbar^2}{2m_e}\nabla_i^2 + V_{\text{eff}}(\mathbf{r})\right)\phi_i(\mathbf{r}) = \epsilon_i\phi_i(\mathbf{r}), \tag{2.7}$$

where $V_{\text{eff}}(\mathbf{r})$ is the effective potential:

$$V_{\text{eff}}(\mathbf{r}) = V_{\text{ext}}(\mathbf{r}) + V_{\text{Hartree}}(\mathbf{r}) + V_{\text{xc}}(\mathbf{r}). \tag{2.8}$$

In Eq. (2.8) $V_{\text{Hartree}}(\mathbf{r})$ corresponds to the Hartree energy which indicates the Coulomb interaction energy of electron with itsef:

$$V_{\text{Hartree}}[\rho] = \frac{\mathrm{e}^2}{2}\int d\mathbf{r}d\mathbf{r}'\frac{\rho\mathbf{r}\rho\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|}. \tag{2.9}$$

where $V_{\text{XC}}(\mathbf{r})$ corresponds to the exchange-correlation energy, which includes all the corrections from many-body interactions between electrons. The exact form of the exchange-correlation energy is unknown, and some form of approximation must be introduced. The approximated form of the exchange-correlation energy is discussed in the next subsection.

From the previous discussions, one can see that there is a self-consistency problem in solving the Kohn-Sham equations. That is to say, to solve the Kohn-Sham equations, external potentials need to be defined. To define the external potentials, electron density must be given. To find the electron density, the Kohn-Sham orbital must be given. To know the Kohn-Sham orbital, the Kohn-Sham equations must be solved. To break this problem, an iterative method is used. In the method, the first step is to guess the initial electron density. Then the iterative step shown in Fig. 2.1 is carried until the difference between the input electron density and the calculated electron density from the Kohn-Sham equation is below certain threshold.

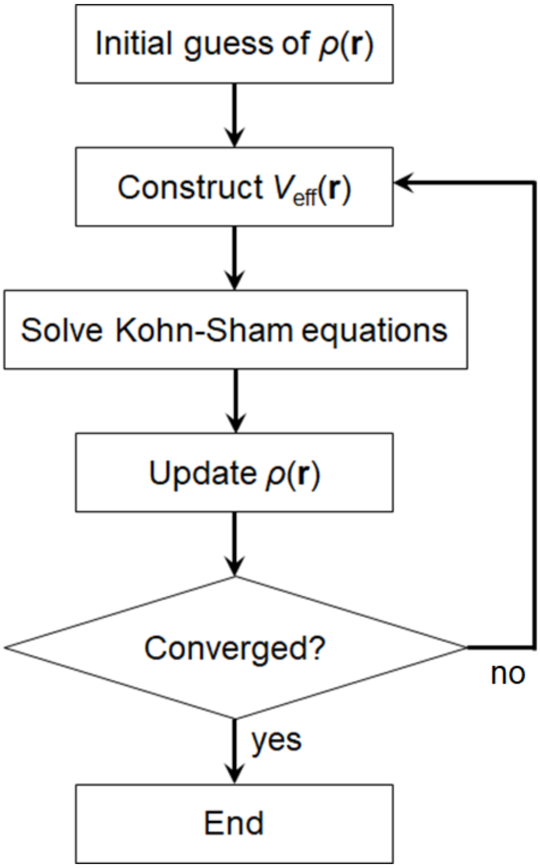
Initial guess of $\rho(\mathbf{r})$
Construct $V_{\text{eff}}(\mathbf{r})$
Solve Kohn-Sham equations
Update $\rho(\mathbf{r})$
Converged?
no
yes
End

Fig. 2.1: The schematic description of the self-consistency method for solving the Kohn-Sham equations.

2.1.4 Exchange-correlation energy

Finally, we discuss the exchange-correlation energy E_{XC} introduced in Eq. (2.8). The simplest form of approximated E_{XC} is the local density approximation (LDA),[13] which is built under the assumption that the E_{XC} of a many-electron system depends only on the local electron density. Thus the E_{XC} under LDA is expressed as:

$$E_{\mathrm{XC}}^{\mathrm{LDA}} = \int d\mathbf{r}\rho(\mathbf{r})\epsilon_{\mathrm{XC}}[\rho(\mathbf{r})]. \tag{2.10}$$

LDA gives reasonable results for systems with slowly varying charge density. However, for systems with electronic structures that deviate significantly from uniform electron gas, LDA typically provides a large error. To add more information to LDA, generalized gradient approximation (GGA) E_{XC} incorporates the gradient of the electron density in addition to the local density:[14, 15]

$$E_{\mathrm{XC}}^{\mathrm{LDA}} = \int d\mathbf{r}\rho(\mathbf{r})\epsilon_{\mathrm{XC}}[\rho(\mathbf{r}), \nabla\rho(\mathbf{r})]. \tag{2.11}$$

GGA functionals generally provide more accurate results compared to LDA. However, both LDA and GGA functionals have a representative error of severe underestimation in bandgap prediction. The error is caused by the omission of electron self-interaction energy and derivative discontinuity at the integer number of electrons. To correctly address the bandgap of insulators, beyond-DFT methods such as GW approximation[16] or hybrid functionals should be adopted.[17] However, we do not provide detailed discussions on those methods as they are out of the scope of this dissertation.

2.2 Classical potentials

Although DFT calculations can predict materials properties with high accuracy, the practical usage of DFT is often hindered by its demanding computational cost. Expressly, as the computational cost of DFT scales as $O(N_{\mathrm{e}}^3)$, the model size is generally limited to less than 1000 atoms for typical DFT calculations (N_{e} is the number of electrons in the system). In this respect, classical interatomic potentials have a significant advantage. Classical potentials approximate the atomic interactions using physically driven functions, and by doing so, can omit the costly computation processes from quantum mechanics. The computational burden of classical potentials typically scales between $O(N_{\mathrm{at}})$ and $O(N_{\mathrm{at}}^2)$, where N_{at} is the number of the atoms in the system.[18] Therefore, they can handle the model size as large as several million atoms with modest computational cost. Consequently, they are beneficial for modeling atomic systems requiring large supercell such as surface, interface, dislocation, grain boundary, stacking fault, etc.

One of the most well known interatomic potentials is the Lennard-Jones potential[19]:

$$V_{\mathrm{LJ}}(r_{ij}) = 4\epsilon\Big[\Big(\frac{\sigma}{r_{ij}}\Big)^{12} - \Big(\frac{\sigma}{r_{ij}}\Big)^{6}\Big], \tag{2.12}$$

where σ and ϵ are the parameters to set the length and the energy scale, respectively. In this section, *i*, *j*, and *k* represent the atomic indices and *r* represents the distance between two atoms. The functional terms of Lennard-Jones potential are pretty straightforward for interpretation; the first term in Eq. (2.12) corresponds to the attraction and the second term describes the repulsion. Lennard-Jones potential is very simple, however it is suitable for describing noble gases such He, Ne, Ar, Kr, and Xe for they have very low chemical reactivity.

When the atomic systems of interest have stronger chemical bondings than noble gases, the functional form to describe such bonding should contain corresponding physics and chemistry. For example, for modeling metallic systems, embedded-atom-model (EAM) potential is often the favorable choice[20]:

$$E = \sum_i F_i \Big[\sum_{j \neq i} \rho_{ij}(r_{ij}) \Big] + \frac{1}{2} \sum_{i=1} \sum_{j=1}{}' V_{ij}(r_{ij}), \tag{2.13}$$

where ρ is a function approximating the electron density, *V* is a pair potential, and F_i is an embedding function representing the energy required to insert atom *i* into a specific position in the electron cloud. The idea of EAM potential is to mimic cores (nuclei and core electrons) and surrounding delocalized electrons in metallic systems.

In contrast, for covalent materials (e.g. silicon or carbon) where the bondings are strongly directional and localized, the interatomic potential must reflect such directionality. Therefore favorable interatomic potentials for covalent systems often have an explicit angular term in their functional forms. One example of such potential is Stillinger-Weber potential[21]:

$$E = \sum_i \sum_{j>i} V_2(r_{ij}) + \sum_i \sum_{j \neq i} \sum_{k>j} V_3(r_{ij}, r_{ik}, \theta_{ijk}), \tag{2.14}$$

where the two-body term V_2 is expressed as:

$$V_2(r_{ij}) = A_{ij}\epsilon_{ij} \Big[B_{ij} \Big(\frac{\sigma_{ij}}{r_{ij}} \Big)^{p_{ij}} - \Big(\frac{\sigma_{ij}}{r_{ij}} \Big)^{q_{ij}} \Big] exp\Big(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \Big), \tag{2.15}$$

and the three-body term V_3 is:

$$\begin{aligned} V_3(r_{ij}, r_{ik}, \theta_{ijk}) = {} & \lambda_{ijk}\epsilon_{ijk}[\cos\theta_{ijk} - \cos\theta_{0,ijk}]^2 \\ & \times \exp\Big(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}} \Big) \exp\Big(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}} \Big). \end{aligned} \tag{2.16}$$

A, *B*, ϵ, *p*, *q*, *a*, σ, and γ are the potential parameters. While Stillinger-Weber potential can provide accurate results for idealized systems to which the potential is fitted, it may critically lack the transferability to other atomic systems since the underlying functional form lacks the flexibility. Bond-order potentials such as Tersoff potentials can provide a more flexible and adaptable description of covalent systems. The general functional form of the bond-order potentials is expressed as[22]:

$$E = \frac{1}{2}\sum_i \sum_j^{Z_i} [qV_R(r_{ij}) + bV_A(r_{ij})], \tag{2.17}$$

where Z is the number of neighboring atoms, q is a parameter depends on the local electronic density, V_{R} and $V_{\mathrm{A}}$are the repulsive and attractive interactions, respectively. b is the bond order parameter that controls the strength of the bond according to the bond character (i.e., single, double, or triple bond). An important variation of the bond-order potential is reactive force-field (ReaxFF).[23] ReaxFF is particularly useful since it well-describes the bond breaking and forming (i.e. chemical reactions) during the simulations. This is addressed by building general functions, including terms describing two- three- and four-body interaction energy, over- and under- coordination energy, van der Waals energy, Coulomb energy, and conjugated bond energy. Therefore ReaxFF inherently holds a large number of fitting parameters.

The classical potential parameters are typically determined such that the properties given by the potential corresponds to that of reference data. The reference can be either experiment or more accurate calculations. Some of the frequently used properties include the cohesive energy, lattice parameters, elastic constants, phonon modes, defect formation energy, surface energy, etc.

While the aforementioned classical potentials are based on strong physics and have had much success in atomic modeling for a long time, they possess some severe limitations. First, the material systems in practice often do not have distinct bonding character but rather have mixed bonding types. Therefore, it is exceedingly difficult to choose the proper functional form and sometimes considered practically impossible in such cases. Second, the parameter fitting is highly nontrivial, as the potential functions are non-linear and highly-coupled. Therefore, the parameter fitting procedure itself can be very demanding, especially for complex classical potentials and/or multispecies systems. Third, as it is very hard to develop general potential working for diverse systems, the usage of most of the classical potentials are limited to the target system they are fitted to. In other words, classical potentials lack transferability compared to first-principles calculations. Finally, even when the parameters are carefully optimized, the accuracy of the classical potentials can still be limited as they adopt major approximations. The most significant approximation is the existence of the explicit chemical bonding function itself. For real systems, the concept of chemical bonding is vaguely defined within the quantum mechanical level, especially for inorganic solids.

2.3 Machine learning potentials

DFT calculations and physics-based classical interatomic potentials have the respective strengths shortcomings: DFT demands a substantial computational cost, and classical potentials have limited accuracy and difficulty in development. In this regard, machine learning potentials are highly anticipated to achieve simulation accuracy comparable to the reference first-principles calculations, with a much lower calculation cost. Through the years, several types of machine learning potentials have been suggested varying by descriptors and models. Among the various potentials, high-dimensional neural network potential (HDNNP; NNP)[7] and Gaussian approximation potential (GAP)[6] are the two most appreciated and utilized models to this date. NNP and GAP are suggested in 2007 and 2010, respectively, and have been applied to a wide range of materials since. Figure 2.2 shows the citation number of seminal machine learning potential papers per year.

Even though machine learning potentials are often considered a subclass of conventional classical potentials, there is a fundamental difference. That is, unlike traditional classical potentials, machine learning potentials does not assume the chemical bondings. Instead, machine learning potentials map atomic energies from the given total energy of reference quantum mechanical calculations. This difference causes unique characteristics of machine learning potentials. In the following subsection, we discuss the atomic energy mapping of machine learning potentials in detail. But before that, we first give a theoretical background for NNP and GAP from the perspective of models and descriptors. Moreover, we provide some of the training techniques for NNP, which is used throughout Chapter 3, 4, and 5.

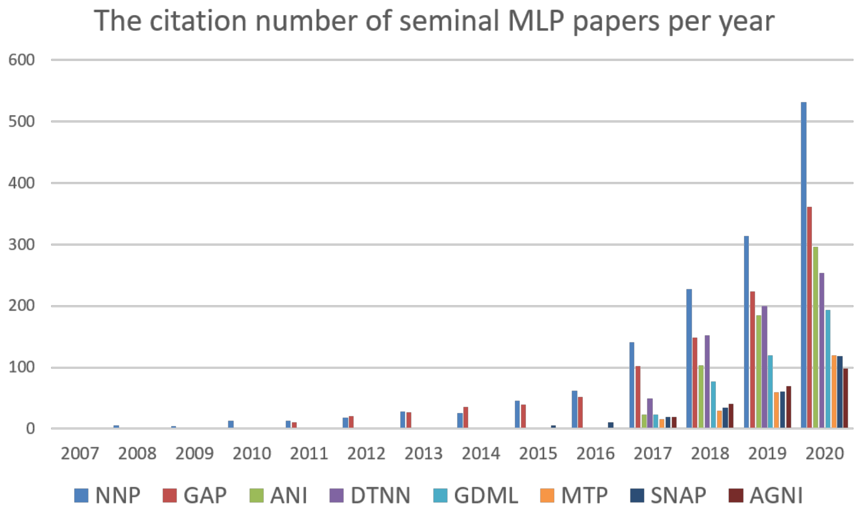

Fig. 2.2: The citation number of seminal machine learning potential papers per year. The machine learning potentials include high-dimensional neural network potential (NNP),[7] Gaussian approximation potential (GAP),[6] accurate neural network engine for molecular energies (ANI),[5] deep tensor neural network (DTNN),[4] gradient-domain machine-learning (GDML),[3] moment tensor potential (MTP),[2] spectral neighbor analysis potential (SNAP),[1] and AGNI force field.[24]

2.3.1 Models

Machine learning potentials utilize flexible machine learning models to describe the structure-property relation of a given system. The models of machine learning potentials can be categorized into four broad classes; linear models (e.g., SNAP, MTP),[1, 2] artificial neural network-based models (e.g., NNP, ANI),[5, 7] kernel-based models (e.g., GAP, GDML),[3, 6] and graph network-based models (e.g., DTNN).[4] Artificial neural network-based potentials and kernel-based potentials have the most extended history and have been frequently applied to advanced simulations of a wide range of material systems.

Artificial neural networks are known to be universal function approximators when the number of weights (parameters) is sufficiently large.[25] This indicates that they have enough flexibility to be used for general material systems. The parameters of neural networks are optimized by a training procedure called backpropagation.[26] The training of artificial neural networks typically requires an iterative process and thus long computation time. However, many useful techniques have been developed that can save the training cost. As neural networks take vectors as the input, the information material structures must be transformed into an input vector before it is fed to the neural network.

On the other hand, kernel-based models predict the properties by the linear combination of the kernel functions. The kernel functions measure the similarity between the training set structures and the predicted structures. The model parameters can be computed by linear algebra; thus, the training of the model can be done in an instant. However, because the similarity must be measured for every training points whenever the model evaluation is carried, the computational cost for property prediction grows with the data set size. It contrasts to the neural network-based model, where the model evaluation time is independent of the training set size once the model is trained.[27]

In the subsequent subsections, we describe high-dimensional neural network potential and Gaussian approximation potential, which represent the neural network-based and kernel-based models.

High-dimensional neural network

Artificial neural network-based potentials typically adopt a simple feed forward neural network to describe the structure-property relation. Feed forward neural network is given as:

$$x_k^{i+1} = \sigma(\sum_j^{N^i} x_j^i w_{jk}^i + b_k^i), \tag{2.18}$$

where x_j^i is the jth node of the ith layer, w_{jk}^i is a weight parameter connecting the jth node of the ith layer and the kth node of the $(i+1)$th layer, and b^i is a bias for the ith layer. The first layer or the input layer takes the structural information given as an input vector (descriptor). Before the node values of ith layer propagate to $(i+1)$th layer, an activation function σ is applied. The activation function is preferred to be a monotonically increasing nonlinear function such that it can provide nonlinearity to the neural network model. Typically used activation functions include sigmoid, hyperbolic tangent, and rectifying linear unit (ReLU).[28] In this dissertation, we mainly adopt the sigmoid function (Eq. (2.19)) as the activation function:

$$\sigma(x) = \frac{1}{1+e^{-x}}. \tag{2.19}$$

The activation function is omitted (in other words, the identity function is the activation function) between the last hidden layer and the output layer to achieve an unbounded output value. Schematic description of a feed forward neural network is shown in Fig. 2.3.

When building neural network models to describe structural-energy relationships, the simplest method is to use atomic coordinates as input and total energy as output.[29–31] However, such atomic coordinate-total energy models have serious drawbacks. The most severe disadvantage is that when the model is optimized for a specific structure,

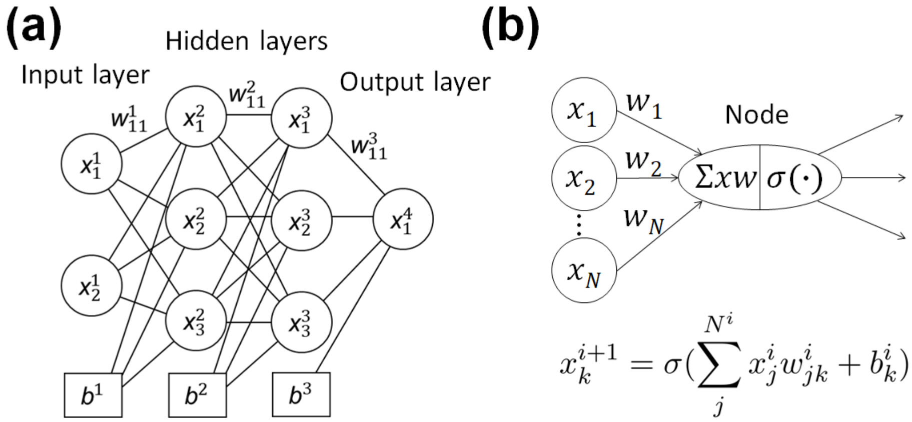

Fig. 2.3: (a) Schematic descriptionof a feed-forward neural network. (b) Schematic description of a hidden node.

it can only evaluate the energy of the structures with the same number of atoms as the trained structure. This is because the model has a fixed input node size.

To this end, Behler and Parrinello introduced a high-dimensional neural network (HDNN) model.[7] The main idea of the HDNN model is to take total energy (E_{tot}) as the sum of atomic energies (E_{at}):

$$E_{\text{tot}} = \sum_{i=1}^{N_{\text{at}}} E_{\text{at}}(\mathbf{G}_i), \tag{2.20}$$

where N_{at} is the number of atoms. The neural network of Eq. (2.18) evaluate the atomic energy rather than the total energy. Thus the input vector $\mathbf{G}_i$ describe the local environment of each atom in the system. Figure 2.4 shows the schematic description of a high-dimensional neural network potential model. The atomic forces are given by Eq. (2.20):

$$F_{i,\alpha} = -\frac{\partial E_{\text{tot}}}{\partial R_{i,\alpha}} = -\sum_{j=1}^{N_{\text{at},j}} \sum_{s=1}^{N_s} \frac{\partial E_{\text{at},j}}{\partial G_{j,s}} \frac{\partial G_{j,s}}{\partial R_{i,\alpha}}, \tag{2.21}$$

where α indicates the x, y, or z component, $R_{i,\alpha}$ is the coordinate of ith atom, $G_{j,s}$ is the sth component of $\mathbf{G}_j$, N_s is the dimension of $\mathbf{G}$, and $N_{\text{at},j}$ is the number of atoms around atom j.

Before use, the weights of neural networks get optimized by the process called training. The training process is carried by minimizing the difference between the reference property and the model prediction for given structures. Commonly used reference properties include total energy, atomic force, and viral stress. The difference of reference property and the model prediction is represented by a function named loss function (Γ). The typical loss function is represented as:

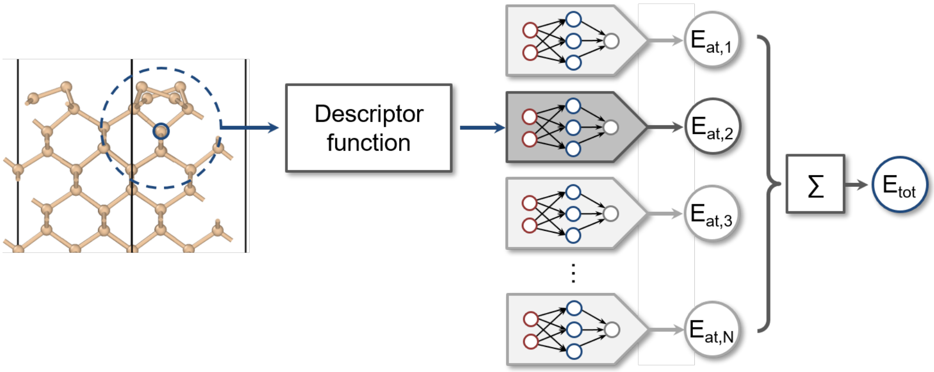

Fig. 2.4: Schematic figure of a high-dimensional neural network potential.

$$\begin{aligned}\Gamma = &\frac{1}{N_{\mathrm{str}}}\sum_{i=1}^{N_{\mathrm{str}}}\Big(\frac{E_i^{\mathrm{DFT}} - E_i^{\mathrm{NNP}}}{N_{\mathrm{at},i}}\Big)^2 \\ &+ \frac{\mu}{3\sum_{i=1}^{N_{\mathrm{str}}} N_{\mathrm{at},i}}\sum_{i=1}^{N_{\mathrm{str}}}\sum_{j=1}^{N_{\mathrm{at},i}} |\mathbf{F}_{ij}^{\mathrm{DFT}} - \mathbf{F}_{ij}^{\mathrm{NNP}}|^2 \\ &+ \frac{\nu}{6N_{\mathrm{str}}}\sum_{i=1}^{N_{\mathrm{str}}}\Big(\frac{\mathbf{S}_i^{\mathrm{DFT}} - \mathbf{S}_i^{\mathrm{NNP}}}{N_{\mathrm{at},i}}\Big)^2,\end{aligned} \tag{2.22}$$

where N_{str} is the total number of structures in the training set, $N_{\mathrm{at},i}$ is the number of atoms in the ith structure, E_i^{DFT} (E_i^{NNP}), $\mathbf{F}_{ij}^{\mathrm{DFT}}$ ($\mathbf{F}_{ij}^{\mathrm{NNP}}$), and $\mathbf{S}_i^{\mathrm{DFT}}$ ($\mathbf{S}_i^{\mathrm{NNP}}$) are the total energy of the ith structure, atomic force of the jth atom in the ith structure, and the virial stress of the ith structure from DFT (NNP), respectively. μ and ν are the scaling coefficient which control the relative training scale between energy, force, and stress loss function term. Before training, the weights and bias are randomly initialized following a certain distribution. The favorable choice of weight initialization is a normal distribution with 0 mean and small standard deviation. When the standard deviation is too small, meaning $\sigma_w \to 0$ the weight optimization by backpropagation algorithm does not work. On the other hand, when σ_w is too large, the training process easily gets stuck in local minima, and the training quality degrades. The training process is updating the weights and bias iteratively such that the loss function gets minimized. One example of a training algorithm is the simple gradient descent method[32]:

$$w_{n+1} = w_n - \alpha\frac{\partial\Gamma}{\partial w_n}, \tag{2.23}$$

where α is a hyperparameter call learning rate. When the learning rate is too small, the training process takes excessive iteration steps. In contrast, when the learning rate is too large, the loss function value may not converge properly, leading to a poorly-trained neural network model. As the gradient descent algorithm generally requires a large number of iteration steps and training time, more efficient training algorithms have been developed, such as momentum optimizer,[33] Adam optimizer,[34] and

Adagrad,[35] to name a few. Also, some of the well-known optimizing algorithms such as Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS),[36] Levenberg-Marquardt,[37, 38] and Kalman filter algorithms[39] have been applied to the training of neural network potentials.[40–42]

Gaussian process regression

Gaussian approximation potential (GAP) is a widely-used machine learning potential based on the Gaussian process regression model. [43] Like high-dimensional neural network, GAP starts with representing the total energy as the sum of atomic energies. Then the atomic energy is given as a linear combination of basis functions:

$$E_{\mathrm{at},i} = \sum_h w_h \phi_h(\mathbf{d}_i), \tag{2.24}$$

where ϕ_h is hth basis function, w_h is the corresponding coefficient, and $\mathbf{d}_i$ is a descriptor vector of ith atom, respectively. The prior distribution of the coefficients w_h follows the multivariate Gaussian distribution with 0 mean and variance of σ_w^2. GAP evaluate the atomic energy by measuring the similarity, which is given by the covariance:

$$\langle E_{\mathrm{at},i} E_{\mathrm{at},j} \rangle = \sigma_w^2 \sum_h \phi_h(\mathbf{d}_i)\phi_h(\mathbf{d}_j). \tag{2.25}$$

The covariance is expressed as a kernel function C as it is computationally much simpler:

$$C(\mathbf{d}_i, \mathbf{d}_j) = \sum_h \phi_h(\mathbf{d}_i)\phi_h(\mathbf{d}_j). \tag{2.26}$$

Gaussian commonly uses smooth overlap of atomic positions (SOAP) as the kernel function, which is explained in Section 2.3.2. When we say t_{N+1} is the property to be predicted (atomic energy in this case) and $\mathbf{t}_N$ is the collection of the properties from the training set, the prior probability of $\mathbf{t}_N$ is expressed as:

$$P(\mathbf{t}_N) \propto \exp\left(-\frac{1}{2}\mathbf{t}_N^T \mathbf{C}^{-1}\mathbf{t}_N\right), \tag{2.27}$$

where $\mathbf{C}$ is $\left\langle \mathbf{t}_N \mathbf{t}_N^T \right\rangle$. According to Bayes' theorem, the probability distribution of the t_{N+1} under training set $\mathbf{t}_N$ is given as:

$$P(t_{N+1}|\mathbf{t}_N) = \frac{P(\mathbf{t}_N|t_{N+1})P(t_{N+1})}{P(\mathbf{t}_N)}. \tag{2.28}$$

From Eq. (2.27) and Eq. (2.28), the posterior distribution is given by:

$$P(t_{N+1}|\mathbf{t}_N) \propto \exp\left(-\frac{1}{2}[\mathbf{t}_N t_{N+1}]^T \mathbf{C}_{N+1}^{-1}[\mathbf{t}_N t_{N+1}]^T\right), \tag{2.29}$$

$$C_{N+1} \equiv \begin{bmatrix} \begin{bmatrix} \\ \quad \mathbf{C}_N \quad \\ \\ \end{bmatrix} & \begin{bmatrix} \\ \mathbf{k} \\ \\ \end{bmatrix} \\ \begin{bmatrix} \quad \mathbf{k}^T \quad \end{bmatrix} & \begin{bmatrix} \kappa \end{bmatrix} \end{bmatrix} \tag{2.30}$$

The prediction value t_{N+1} can be evaluated from the mean of the posterior distribution which can be derived by brute-force inversion of $\mathbf{C}_{N+1}$. Then we find:

$$\hat{t}_{N+1} = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N. \tag{2.31}$$

Also, standard deviation of the posterior distribution is used as the prediction uncertainty:

$$\sigma_{N+1}^2 = \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}. \tag{2.32}$$

2.3.2 Descriptors

Descriptors are transformed values of atomic structure information for use in machine learning potentials. Several descriptors, including the Coulomb matrix,[44] bag of bonds,[45] bispectrum of neighbor density,[6], partial radial distribution functions,[46] and crystal graph convolutional neural networks(CGCNN),[47] have been suggested over the years. As the energy of an atomic structure is invariant to translation, rotation, and permutation of atoms with the same species, it is desirable for descriptors to be also invariant to such operations. Atom-centered symmetry function (ACSF)[48] and smooth-overlap of atomic positions (SOAP)[49] are two descriptors that satisfy such symmetry operation invariance. They are demonstrated to be very powerful when used with high-dimensional neural network potential and Gaussian approximation potential, respectively.

Atom-centered symmetry function

Atom-centered symmetry functions are the most popular choice of descriptor vectors for high-dimensional neural network potential.[48] The symmetry functions are consist of three types: cutoff function, radial functions, and angular functions. The cutoff function is adopted to ensure the continuity of the function value and its derivatives at the cutoff radius. The cutoff function has a smoothly decaying shape so that the influence of a neighboring atom can decay as the distance between the atoms increases (See Fig. 2.5 (a)):

$$f_c(R_{ij}) = \begin{cases} \frac{1}{2}\cos\left(\pi\frac{R_{ij}}{R_c}\right) + \frac{1}{2} & (R_{ij} \leq R_c) \\ 0 & (R_{ij} > R_c) \end{cases}, \tag{2.33}$$

where R_{ij} is the are distances between center atom i and neighboring atom j, and R_c is the cutoff distance. The cutoff distance R_c should be large enough such that all the relevant influences of neighboring atoms can be counted. However, when R_c is too large, the computational cost grows rapidly.

Radial functions and angular functions describe the many-body characteristics of the atomic environments. Radial functions are constructed as the sums of two-body terms, and angular functions have the additional three-body terms to the radial functions. Three types of radial symmetry functions are expressed as:

$$G_i^1 = \sum_j f_c(R_{ij}), \tag{2.34}$$

$$G_i^2 = \sum_j e^{-\eta(R_s - R_{ij})^2} \cdot f_c(R_{ij}), \tag{2.35}$$

$$G_i^3 = \sum_j \cos\left(\kappa R_{ij}\right) \cdot f_c(R_{ij}), \tag{2.36}$$

η, R_s, and κ are the radial symmetry function parameters which determine the shape of functions. η determines the decaying rate, R_s determines the center of the Gaussian, and κ determines the frequency of the cosine function. (See Fig. 2.5) In practice, G_i^2 function is mostly adopted, because G_i^1 function is too simple and G_i^3 function can give negative value. A negative value for symmetry function indicates that a presence of certain atom contributes negatively to other atoms, which is unlikely in real material systems.

Two tyeps of angular symmetry functions are:

$$\begin{aligned} G_i^4 =& 2^{1-\zeta} \sum_{j,k\neq j} (1 + \lambda \mathrm{cos}\theta_{ijk})^{\zeta} \cdot e^{-\eta(R_{ij}^2 + R_{ik}^2 + R_{jk}^2)} \\ & \cdot f_c(R_{ij}) \cdot f_c(R_{ik}) \cdot f_c(R_{jk}), \end{aligned} \tag{2.37}$$

$$\begin{aligned} G_i^5 =& 2^{1-\zeta} \sum_{j,k\neq j} (1 + \lambda \mathrm{cos}\theta_{ijk})^{\zeta} \cdot e^{-\eta(R_{ij}^2 + R_{ik}^2)} \\ & \cdot f_c(R_{ij}) \cdot f_c(R_{ik}), \end{aligned} \tag{2.38}$$

Again, ζ and λ are the angular symmetry function parameter which determine the shape of the functions. The difference between G^4 function and G^5 function is the

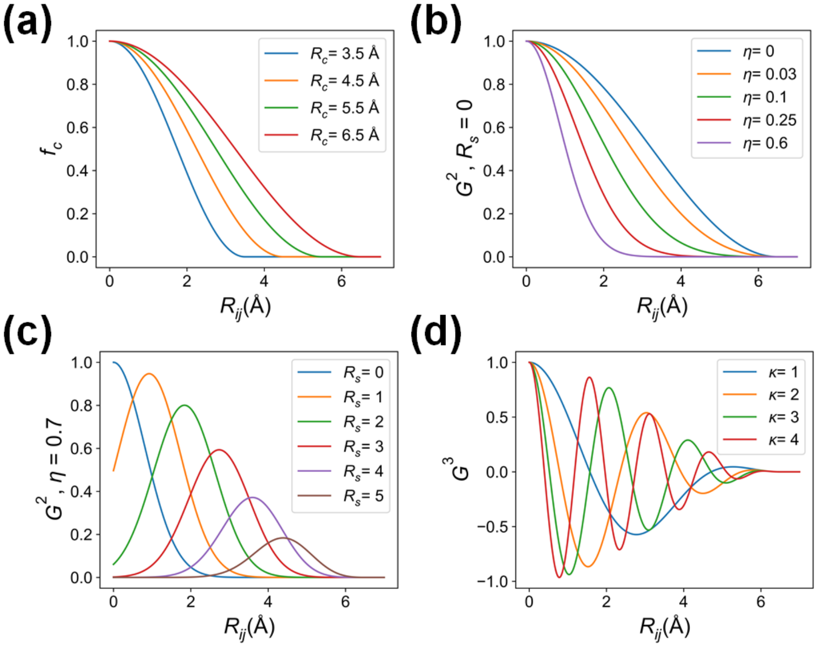

Fig. 2.5: (a) The shape of cutoff function. (b) The shape of G^2 symmetry function with respect to η (c) The shape of G^2 symmetry function with respect to R_s (d) The shape of G^3 symmetry function.

additional $\mathrm{e}^{-\eta R_{jk}^2}$ and $f_c(R_{jk})$ term. The additional terms decreases as R_{jk} increases, and in turn θ_{ijk} increases. Therefore G_i^4 function disregard the contribution of wide angle θ_{ijk} values. ζ determines the width of the Gaussian function and λ determines the peak position. (See Fig. 2.6)

When building symmetry function input vector, each element of the vector is calculated by different form of symmetry functions, varying the function parameters η, R_s, κ, ζ, and λ. With the increase of input vector size and the diversity of parameters set, the structural resolution of the descriptor vector increases. However, if the size of the input vector is too large, the computational cost becomes excessive. It is especially the case for angular symmetry functions because they require estimations of distance and angle of all combinations of neighbor atoms. Also, the increase of the input size leads to an increase in the number of neural network weights, which may cause overfitting problems or training process hindrance. Therefore, the optimum choice of symmetry function vector size and parameter set must be decided according to the training quality, computational cost, etc. [50, 51]

For multi-element systems, only a particular combination of elements is considered to evaluate the specific symmetry function. For instance, for element A in the A-B system, radial symmetry function values are measured for either AA or AB interaction for an individual symmetry function. Likewise, a particular angular symmetry function value is measured for either AAA, AAB, or ABB interactions. Thus, atoms with different species contribute differently even when they are placed in the same position. One side effect is that the size of the input vector increase with quadratic scale as the number of elements increases. As mentioned, an oversized input vector causes overfitting problems or training process hindrance. Therefore some types of modified symmetry functions have been suggested to keep the input vector size independent of the number of elements in the system. [52]

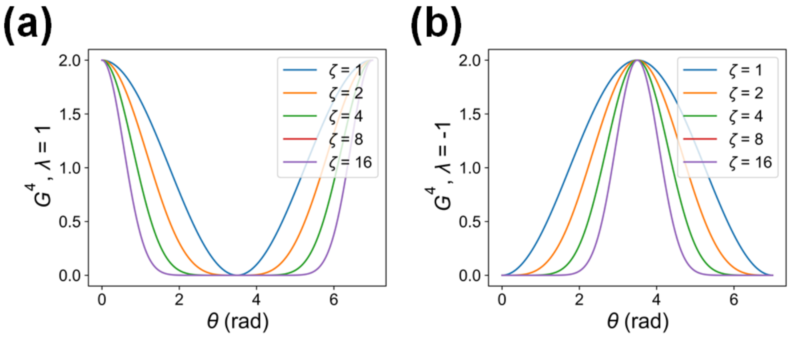

Fig. 2.6: (a) The shape of G^4 symmetry function when λ is 1. (b) The shape of G^4 symmetry function when λ is -1

Smooth overlap of atomic positions (SOAP)

When Gaussian approximation potential was first demonstrated, the bispectrum of the neighbor density method was adopted as the descriptor.[6] However, because the neighbor density employs δ-function, slight deviations of positions cause strong numerical changes. To solve this problem, smooth overlap of atomic position (SOAP) scheme has been suggested.[49] Instead of δ-function, SOAP representation employs smoothly decay Gaussian functions centered at the positions of each atoms. Thus the atomic neighbor density function is given as:

$$\rho(\mathbf{r}) = \sum_i \exp(-\alpha|\mathbf{r} - \mathbf{r_i}|^\mathbf{2}), \tag{2.39}$$

where $\mathbf{r}_i$ is a vector from center atom to neighbor atom i. In contrast to the symmetry function input vector, where each symmetry function obtains the individual element, and a set of function values obtains full input vector, Eq. (2.39) characterizes the full atomic environment at once. Therefore Eq. (2.39) cannot achieve rotational invariance. Thus when building a SOAP kernel for comparing two atomic environments, they are integrated over all possible rotations $\hat{R}$:

$$k(\rho, \rho') = \int \mathrm{d}\hat{R} \left| \int \mathrm{d}\mathbf{r} \rho(\mathbf{r}) \rho(\hat{R}\mathbf{r}') \right|^\mathbf{n}, \tag{2.40}$$

n is usually chosen as 2 since the rotational information is lost for n = 1 due to the interchangeability of the integrals. Then, the Gaussian density is expanded using a set of normalized radial basis functions and spherical harmonics for computational efficiency:

$$k(\rho, \rho') = \sum_{blm} c_{blm} g_b(|\mathbf{r}|) Y_l m(\hat{\mathbf{r}}), \tag{2.41}$$

where b, l, m, is the coefficients and g_b is normalized radial basis function. Finally, SOAP kernel function is normalized as:

$$C(\rho, \rho') = \left(\frac{k(\rho, \rho')}{\sqrt{k(\rho, \rho) k(\rho', \rho')}} \right)^{\zeta}, \tag{2.42}$$

where ζ is a hyperparameter which can be any positive integer.

2.3.3 The training techniques for NNP

This subsection introduces some of the training techniques for NNP that can improve training speed, quality, and potential stability. The first technique to be introduced is the scaling of the input descriptor vector. As explained in Section 2.3.1, the neural network weights are randomly initialized with certain distribution before training. The mean of the distribution is generally set to be 0; thus, preprocessing the range of the descriptor vector to correspond to the mean of weight distribution helps the training process by a large margin. Another meaning of the input scaling is to match the order of descriptor vector elements. That is to say, it is beneficial for each element of the descriptor vector to have a similar order of range since the machine learning model would mostly focus on the largely varying elements, even when all elements contribute in describing the atomic environments with a similar degree. A simple normalization process of scaling the descriptor vector into the range of [-1, 1] is known to be effective. In addition, scaling the descriptor vector components to have 0 mean and unit variance or scaling with reference to the uniform gas density has been demonstrated to be useful.[53]

Another essential technique is principal component analysis transformation of the descriptor vector. As the symmetry function components consist of function values from a few functions with varying function parameters, they tend to have a very high correlation with each other. The high correlation within the input vector components is known to be disadvantageous to the training speed and quality. Hence, decorrelating the input vectors using principal component analysis (PCA) and whitening is very beneficial to the training procedure. PCA is a process to compute the principal component out of a given data set, and the principal components are the direction within the vector space that best fit the data while being orthogonal to other principal components.[54] Because the principal components are linearly uncorrelated, when the input vectors are transformed into principal components and whitened before fed to the neural network model, the training convergence speed and quality improve drastically.

The PCA transformation of the descriptor vector is carried as follows: First, the covariance matrix must be calculated to extract the principal components from the input vector set. When the symmetry function vector set of training data is given as N_{at} by N_s matrix $\mathbf{X}$ (where N_{at} is the number of atoms in the training set, and N_s is the dimension of the symmetry function vector.), the covariance matrix Σ^{XX} is computed and diagonalized as:

$$\Sigma^{XX} = \mathbf{X}^T\mathbf{X} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T. \tag{2.43}$$

When $\mathbf{Z}$ is $\mathbf{XW}$, the PCA is given as:

$$\Sigma^{ZZ} = \mathbf{Z}\mathbf{X}^T\mathbf{Z} = \mathbf{W}^T\mathbf{W}\mathbf{\Lambda}\mathbf{W}^T\mathbf{W} = \mathbf{\Lambda}. \tag{2.44}$$

Then, each principal component is scaled to have unit variance by a process called whitening.

$$z_{(i)}^{\text{whiten}} = \frac{z_{(i)}}{\sqrt{\text{Var}_{(i)}}}, \tag{2.45}$$

where $z_{(i)}$ indicates the ith principal component. Furthermore, it is beneficial to add a small positive quantity ϵ to the variance to prevent numerical divergence:

$$z_{(i)}^{\text{whiten}} = \frac{z_{(i)}}{\sqrt{\text{Var}_{(i)} + \epsilon}}. \tag{2.46}$$

When ϵ is too large, overfitting becomes more likely to occur. On the other hand, when ϵ is too small, the whitening becomes less effective as $\epsilon \to \infty$ corresponds to no whitening. Thus the proper value of ϵ must be decided through tests.

The next issue to be addressed is overfitting and the prevention technique of overfitting. Overfitting is a type of error that occurs when a machine learning model is too closely fitted to training set data that its prediction ability to general data points is seriously degraded. Overfitting typically arises when the size of the training set is too small compared to the model capacity (the number of weights), or the optimization procedure is carried with an excessive amount of steps and a small learning rate. Therefore, it is highly recommended to collect a sufficient amount of training data before training. Also, to monitor whether training steps are excessive, dividing the whole data set into a training set and validation set is very common. A validation set is a set of data points that are randomly selected and intentionally left out of the training procedure so that its error can be monitored during the training. If the validation set error increases while the training set error decreases, it is a clear sign of overfitting, and the training procedure should be terminated at the point.

Another way to prevent the overfitting problem is to control the model capacity. There are two popular algorithms to avoid excessive model capacity. The first method is the regularization technique.[55] The regularization technique adds a penalty term to the loss function to prevent the divergence of weight values. L_1 and L_2 norm is the most favorable penalty terms.

$$\Gamma_{\mathrm{L1}} = \Gamma + \lambda ||w||_1, \tag{2.47}$$

$$\Gamma_{\mathrm{L2}} = \Gamma + \lambda ||w||_2^2, \tag{2.48}$$

where λ is the scaling coefficient. λ should be selected carefully such that the overfitting can be prevented while the training quality remains acceptable.

The second method to suppress excessive model capacity is dropout technique.[56] When dropout method is applied, randomly selected nodes and weights are turned off during each iteration of training procedure. The dropout method introduce noise into the weight update and thus prevent overfitting. Mathemetically, a dropout neural network is equivalent to Bayesian network, which means that the dropout neural network can be considered as averaged of neural network ensemble.[57]

2.3.4 Atomic energy mapping of machine learning potentials

As explained in Section 2.3.1, most machine learning potentials are based on the assumption that the DFT total energy ($E_{\text{tot}}^{\text{DFT}}$) can be represented as a sum of the atomic energies (E_{at}) that depends on the local environment within a certain cutoff radius (R_{c}):

$$E_{\text{tot}}^{\text{DFT}} = \sum_i E_{\text{at}}(\mathcal{R}_i; R_{\text{c}}) \ , \tag{2.49}$$

where $\mathcal{R}_i$ is the collection of relative position vectors of atoms lying within cutoff radius R_{c} from the ith atom. Thus it is important to understand that the atomic energy is determined only by local atomic arrangement within DFT level: In other words, the atomic energy is transferable within DFT level. Only when such condition is established, the transferability of machine learning potentials would be secured.

By the semilocal density approximation, $E_{\text{tot}}^{\text{DFT}}$ can be expressed as a sum of kinetic energy (E_{kin}), exchange-correlation energy (E_{XC}), and Coulomb energy (E_{Coul}):

$$\begin{aligned} E_{\text{tot}}^{\text{DFT}} &= E_{\text{kin}} + E_{\text{XC}} + E_{\text{Coul}} \\ &= -\frac{1}{2}\int \nabla_{\mathbf{r}}^2 \rho(\mathbf{r},\mathbf{r}')|_{\mathbf{r}=\mathbf{r}'} d\mathbf{r}' + \int \rho(\mathbf{r})\varepsilon_{\text{XC}}(\rho(\mathbf{r}), \nabla\rho(\mathbf{r}))d\mathbf{r} \\ &+ \frac{1}{2}\int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}d\mathbf{r}' - \sum_i \int \frac{q_i \rho(\mathbf{r})}{|\mathbf{r}-\mathbf{r}_i|} d\mathbf{r} + \sum_{i>j} \frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j|} \ , \end{aligned} \tag{2.50}$$

where $\rho(\mathbf{r},\mathbf{r}')$ is the one-electron density matrix and $\rho(\mathbf{r})$ is the electron density. When we assume $O(N)$ methods, in particular, the divide-and-conquer (DAC) approach,[58, 59] each energy term can be partitioned into atomic contributions defined locally around each atomic site. In the following, we show that atomic energies depend only on nearby atoms such that it is transferable to other systems as long as local environments are maintained.

As the first step, the given system is partitioned into atomic cells with the volume V_i, without gaps or overlapping. Then $\rho_i(\mathbf{r})$ can be defined as $\rho_i(\mathbf{r}) = \rho(\mathbf{r})[\mathbf{r} \in V_i]$

where [...] is the Iverson bracket whose value is 1 when the logical proposition in the bracket is true and 0 when is false. Thus the atomic contribution of exchange-correlation energy becomes:

$$E_{\mathrm{XC},i} = \int \rho_i(\mathbf{r})\varepsilon_{\mathrm{XC}}(\rho_i(\mathbf{r}), \nabla\rho_i(\mathbf{r}))d\mathbf{r}. \tag{2.51}$$

The nearsightedness principles assumes that only nearby atoms influence the charge density at a certain point if the local chemical potential of electrons is fixed.[60, 61] Therefore, $\rho_i(\mathbf{r})$, and hence $E_{\mathrm{XC},i}$ is indeed affected only by atomic arrangements within a certain cutoff (R_{c}^1) from $\mathbf{r}_i$.

For next step we develop the atomic Coulomb energy by defining the total charge density in V_i: $\rho_{\mathrm{tot},i}(\mathbf{r}) = q_i\delta(\mathbf{r}-\mathbf{r}_i) - \rho_i(\mathbf{r})$ and substituting the term into Coulomb energy equation:

$$\begin{aligned} E_{\mathrm{Coul},i} = {} & \frac{1}{2}\sum_{j\neq i}\int \frac{\rho_{\mathrm{tot},i}(\mathbf{r})\rho_{\mathrm{tot},j}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' \\ & + \frac{1}{2}\int \frac{\rho_i(\mathbf{r})\rho_i(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}d\mathbf{r}' - \int \frac{q_i\rho_i(\mathbf{r})}{|\mathbf{r}-\mathbf{r}_i|}d\mathbf{r}. \end{aligned} \tag{2.52}$$

In Eq. (2.52), the second and third term can be locally defined with the finite cutoff as they describe the energy within the same atomic cell i. However, the first term is long-ranged, as it represents the energy between electrons in ith and jth atomic cells. For condensed phases that effectively screen electrostatic interactions (which is quite often), the first term can be ignored beyond a certain cutoff (R_{c}^2). Thus, the electrostatic interaction between $\rho_{\mathrm{tot},i}$ and $\rho_{\mathrm{tot},j}$ can be omitted if $|\mathbf{r}_j - \mathbf{r}_i| > R_{\mathrm{c}}^2$. Since $\rho_i(\mathbf{r})$ and $\rho_{\mathrm{tot},i}(\mathbf{r})$ are influenced by atoms within R_{c}^1, $E_{\mathrm{Coul},i}$ depends on atoms inside $R_{\mathrm{c}}^1 + R_{\mathrm{c}}^2$ (neglecting the volume of V_i). We note that aforementioned scheme may not accurately describe systems with high ionic characters. For such systems, some implementations of NNP have been suggested that explicitly describe the long-range Coulomb potential, separately from short-ranged atomic energies.[62–64] In the following discussions, we assume systems with well-screened electrostatic interactions.

Lastly, we discuss the locality of E_{kin}. It is well-known that the one-electron density matrix $\rho(\mathbf{r}, \mathbf{r}')$ decays exponentially with $|\mathbf{r} - \mathbf{r}'|$ in insulators and metals at finite temperatures.[61, 65] Thus $\rho(\mathbf{r}, \mathbf{r}')$ can be ignored when $|\mathbf{r} - \mathbf{r}'| > R_{\text{c}}^3$, where R_{c}^3 is certain threshold. That is to say, $\rho(\mathbf{r}, \mathbf{r}')$ depends only on the atomic configurations within a cutoff distance (R_{c}^4) (where $R_{\text{c}}^4 > R_{\text{c}}^3$). With the projected density matrix $\rho_{ij}(\mathbf{r}, \mathbf{r}') = \rho(\mathbf{r}, \mathbf{r}')[\mathbf{r} \in V_i][\mathbf{r}' \in V_j]$, the atomic density matrix $\rho_{\text{at},i}(\mathbf{r}, \mathbf{r}')$ is:

$$\rho_{\text{at},i}(\mathbf{r}, \mathbf{r}') = \rho_{ii}(\mathbf{r}, \mathbf{r}') + \frac{1}{2} \sum_{j \neq i}^{|\mathbf{r}_j - \mathbf{r}_i| < R_{\text{c}}^3} \rho_{ij}(\mathbf{r}, \mathbf{r}'). \tag{2.53}$$

As $\rho(\mathbf{r}, \mathbf{r}') = \sum_i \rho_{\text{at},i}(\mathbf{r}, \mathbf{r}')$, the atomic kinetic energy can be expressed as:

$$E_{\text{kin},i} = -\frac{1}{2} \int \nabla_{\mathbf{r}}^2 \rho_{\text{at},i}(\mathbf{r}, \mathbf{r}')|_{\mathbf{r}=\mathbf{r}'} d\mathbf{r}'. \tag{2.54}$$

Combining the above analyses, the atomic energy of the ith atom formally derives from the DFT calculations:

$$E_{\text{at},i} = E_{\text{kin},i} + E_{\text{XC},i} + E_{\text{Coul},i}. \tag{2.55}$$

By evaluating $E_{\text{at},i}$ under various environments, the atomic energy in principle can be obtained as a continuous function of the local environment:

$$E_{\text{at},i} \longrightarrow E_{\text{at}}^{\text{DFT}}(\mathcal{R}; R_{\text{c}}), \tag{2.56}$$

where $R_{\text{c}} = \max(R_{\text{c}}^1 + R_{\text{c}}^2, R_{\text{c}}^4)$. The atomic energy is not unique and depends on the way to partition atomic cells.

The above discussions imply that machine learning potential aims to identify underlying $E_{\text{at}}^{\text{DFT}}$ when total energies are given. This perspective differs from the conventional belief that machine learning potentials merely interpolate given total energies.[66, 67] Thus, it is safe to say that the accuracy of machine learning potentials relies on how close $E_{\text{at}}^{\text{NN}}$ is to the reference $E_{\text{at}}^{\text{DFT}}$ over the configurational space spanned by the given training set. However, since $E_{\text{at}}^{\text{NN}}$ is fitted to the total energies, rather than directly to $E_{\text{at}}^{\text{DFT}}$, the training procedure does not guarantee sufficient accuracies in $E_{\text{at}}^{\text{NN}}$. That is to say, $E_{\mathrm{at}}^{\mathrm{NN}}$ can reproduce total energies in the training set precisely but deviate significantly from $E_{\mathrm{at}}^{\mathrm{DFT}}$. Indeed, machine learning potentials are vulnerable to such 'ad hoc' energy mapping when the training set is not collected with care.[68] For example, when the training set only consists of structures with fixed volume, fixed composition, or a fixed number of atoms, the atomic energy error may cancel each other, and ad hoc mapping may occur. In multi-component systems, ad hoc mapping can occur when the training set only includes structures with a single stoichiometry, as the atomic energy offset becomes completely arbitrary in such a situation. Ad hoc mapping can undermine the transferability and cause serious instability of machine learning potentials. Therefore, it is recommended to collect training set structures with care and monitor the atomic energy mapping during the training process using invariant points in the descriptor space.

Chapter 3

Unbalanced training problem

3.1 Introduction

While the machine learning potentials are getting popular, the weakness and strength of the machine learning potentials are not fully understood at this moment, mainly because of their black-box nature. In this Chapter, we raise a critical issue in training neural network potentials (NNPs), i.e., the distribution of symmetry function vector $\{\mathbf{G}\}$ in the training set is highly inhomogeneous and biased. This results in unbalanced training, which significantly undermines the accuracy and reliability of NNP. To address this issue, we provide various examples of sampling bias and how it degrades the quality of NNP. Then we propose an effective method that equalizes the learning level over $\{\mathbf{G}\}$ in the training set. The method is based on the Gaussian density function(GDF), which quantifies the sparsity of training points. Various examples confirm that GDF weighting significantly improves the reliability and transferability of NNPs compared to the conventional training method.

3.2 Examples of sampling bias

In this section, we first demonstrate the inhomogeneous distribution of symmetry function vector with an example of crystalline Si with one vacancy (See Fig. 3.1 (a)). The training set was generated by performing density functional theory molecular dynamics (DFT-MD) simulations on the 63 atom supercell for 16 ps with the temperature varying from 500 to 1300 K and sampling the snapshots every 20 fs. In total, 50, 400 points were sampled in the feature space. Fig. 3.1 (b) shows the frequency of $\{\mathbf{G}\}$ on the G^2_X–G^4_Y plane with log scale, where G^2_X and G^4_Y correspond to certain radial component and angular component from symmetry function input vector, respectively. (G^2 and G^4 denote G^2 and G^4 symmetry function explained in Section 2.3.2). The distributions of four-fold coordinated bulk Si atoms and three fold-coordinated defective Si atoms neighboring the vacancy site (red atoms in Fig. 3.1 (a)) are displayed separately. It is shown in Fig. 3.1 (b) that the distribution is highly biased and concentrated within a narrow range of $\mathbf{G}$. Namely, most $\mathbf{G}$ vectors belong to bulk Si atoms. This is because training structures contain far more bulk atoms than defective ones (59 versus 4). In addition, the data is concentrated around the equilibrium point, which is a result of the Boltzmann distribution. Because the NNP learns on $\mathbf{G}$ in the training set, the inhomogeneous distribution enforces the NNP to be optimized toward specific configurations (bulk and equilibrium), sacrificing accuracy for underrepresented configurations (defect or off-equilibrium). In order to investigate this quantitatively, we define a Gaussian density function (GDF; $\rho(\mathbf{G})$) defined for an arbitrary $\mathbf{G}$ in the symmetry-function space as follows:

$$\rho(\mathbf{G}) = \frac{1}{N_{\text{tot}}} \sum_{i=1}^{N_{\text{str}}} \sum_{j=1}^{N_{\text{at}.i}} \exp\Big(- \frac{1}{2\sigma^2} \frac{|\mathbf{G} - \mathbf{G}_{ij}|^2}{N_s} \Big), \tag{3.1}$$

where σ is the Gaussian width, N_s is the dimension of the symmetry function vector, N_{str} is the number of structures in the training set, $N_{\text{at}.i}$ is the number of atoms in the training set structure i, $\mathbf{G}_{ij}$ is the $\mathbf{G}$ vector of jth atom in the ith structure and N_{tot}

indicates the total number of atoms in the entire training set. Eq. (3.1) implies that the value of $\rho(\mathbf{G})$ ranges between 0 and 1; values close to 0 (or 1) mean that training points are scarce (or abundant) around the specific $\mathbf{G}$.

Fig. 3.1 (c) shows the calculated energy barrier of vacancy migration by the NNP trained with the conventional energy and atomic force loss function in Eq. (2.22) (NNP-c hereafter). The nudged elastic band(NEB) method is used for barrier calculation and reference DFT result is also plotted for comparison.[69] Even though trajectories of the vacancy migration are included in the training set (the vacancy migrates several times at temperatures higher than 1000 K within the present simulation time), the NNP overestimates the energy barrier by 50 meV, which will significantly underestimate the diffusion coefficient of the vacancy. It is also seen that the force error increases for atoms with small GDF values and the root mean square error (RMSE) of defective Si is 0.27 eV/Å, much larger than 0.18 eV/Å for the bulk atoms. These force errors are for the training set and hence evidence the unbalanced training between bulk atoms and undersampled defective atoms. We stress that the inhomogeneous distribution of $\mathbf{G}_{ij}$ persists for any reasonable choice of training sets. For instance, nonbulk structures such as surfaces and defects are underrepresented in general because they should be modeled together with bulk atoms that outnumber the atoms under the interested environment. In addition, breaking and forming of bonds, critical in chemical reactions, are rare events and occur only a few times during long-time MD simulations.

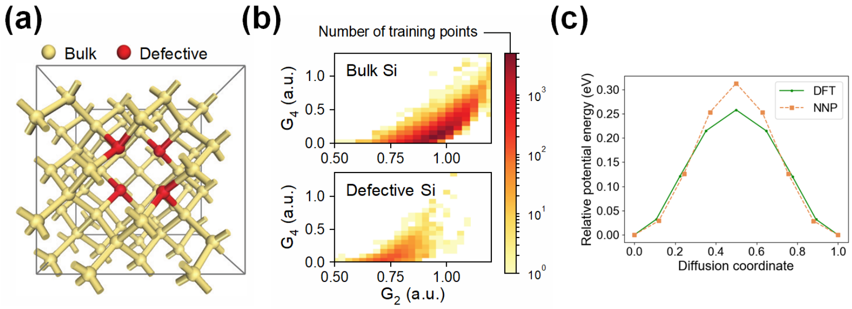

Fig. 3.1: (a) Atomic structures of Si bulk with a vacancy. The defective atoms surrounding the vacancy are marked in red. (b) Distribution of training points in the **G** space. G^2_X and G^4_Y indicate a certain radial component and an angular component from symmetry function vector, respectively, that are selected out of 26-dimensional coordinates of **G**. The number of training points is enumerated on the 20 × 20 mesh and color-coded on the log scale. (c)Vacancy migration barrier calculated by the NEB method. The energy of the Si vacancy at equilibrium is set to zero.[70]

3.3 GDF weighting scheme

In order to alleviate the unbalanced training, it is necessary to enhance the influence of undersampled local environments.We first note that GDF defined in Eq. (3.1) can detect the scarcity of training points around a specific $\mathbf{G}$ point. In addition, while the DFT energy per se cannot be split into local atomic energies, the DFT forces are obtained for individual atoms. Exploiting these two facts, we modify the loss function as follows:

$$\begin{aligned}\Gamma = \frac{1}{N_{\text{str}}}\sum_{i=1}^{N_{\text{str}}}\Big(\frac{E_i^{\text{DFT}}-E_i^{\text{NNP}}}{N_i}\Big)^2 + \\ \frac{\mu}{3\sum_{i=1}^{N_{\text{str}}} N_{\text{at},i}}\sum_{i=1}^{N_{\text{str}}}\sum_{j=1}^{N_{\text{at},i}}\Theta(\rho^{-1}(\mathbf{G}_{ij}))|\mathbf{F}_{ij}^{\text{DFT}}-\mathbf{F}_{ij}^{\text{NNP}}|^2.\end{aligned} \tag{3.2}$$

By choosing Θ in Eq. (3.2) as a monotonically increasing function, one can magnify the influence of $\mathbf{G}_{ij}$'s with small GDF values on the loss function. Among various choices of Θ, we select a modified sigmoid function as it produced the best results in various cases:

$$\Theta(x) = \frac{Ax}{1+\mathrm{e}^{-b(x-c)}}, \tag{3.3}$$

where A is a normalizing constant that makes the average of Θ to be 1 and b and c are parameters that are fine-tuned for balanced training

We train a NNP with GDF weighting scheme on the same training set explained in section 3.2. (NNP-GDF hereafter) In Fig. 3.1 (a), it is shown that the migration barrier by NNP-GDF agrees with the DFT result within 3 meV. Also, the RMSEs of defective and bulk atoms are 0.19 and 0.18 eV/Å for NNP-GDF, respectively, which are more even than those in NNP-c. Fig. 3.1 (b) shows the remnant force error for each atom ($\Delta\ F_{ij} = |\mathbf{F}_{ij}^{\text{DFT}} - \mathbf{F}_{ij}^{\text{NNP}}|$). For visual clarity, $\Delta\ F_j$'s are interval-averaged with respect to the GDF (see circles). It is seen the NNP-GDF gives more uniform errors over the whole range of GDFs and the average force errors at $\mathbf{G}$'s with GDF $< 10^{-3}$ are less than 0.3 eV/Å, which is disntictive from the NNP-c result where the force error increases for atoms with small GDF values. These force errors are for the training set and hence evidence the unbalanced training between bulk atoms and undersampled defective atoms has been remedied by GDF weighting. It is also noticeable that the average force error slightly increases for GDF $> 10^{-1}$. This supports that the GDF weighting effectively increases (decreases) the influence of underrepresented (overrepresented) $\mathbf{G}$ points. (The force errors on the validation set also confirm a similar effect of GDF weighting.)

We also confirm benefits of the GDF weighting with the example of Si interstitials; the training set is generated by carrying out DFT-MD simulations with the 65 atom supercell including one interstitial atom at temperatures from 500 to 1300 K. Like in the vacancy example, force errors for interstitials are much larger than those for bulk atoms in NNP-c, but errors become more even when GDF weighting is applied. (See Fig. 3.3) To further check the accuracy of the NNP, we scan the potential energy surface (PES) of the Si interstitial around the equilibrium point (see Fig. 3.4). The PES is obtained by displacing the interstitial atom on a spherical surface with a radius of 0.6 Å while other atoms are fixed. Comparing PESs from DFT, NNP-c, and NNP-GDF (Fig. 3.3 2(b)–(d), respectively), we can see that NNP-GDF gives a PES closer to the DFT result than NNP-c.

Furthermore, the GDF weighting improves the stability of NNP-MD simulations. For instance, MD simulations with a Si interstitial using NNP-c trained in the above failed repeatedly within 1 ns for temperatures above 1200 K and resulted in unphysical structures. This can be understood as follows: the configurations with large vibrational amplitudes occur during MD at high temperatures but they are underrepresented due to the Boltzmann factor. This leads to large and unpredictable force errors. In contrast, MD with NNP-GDF run stably for a much longer time. (see Fig. 3.5)

As another test, we calculate formation energies of vacancy clusters when the train-

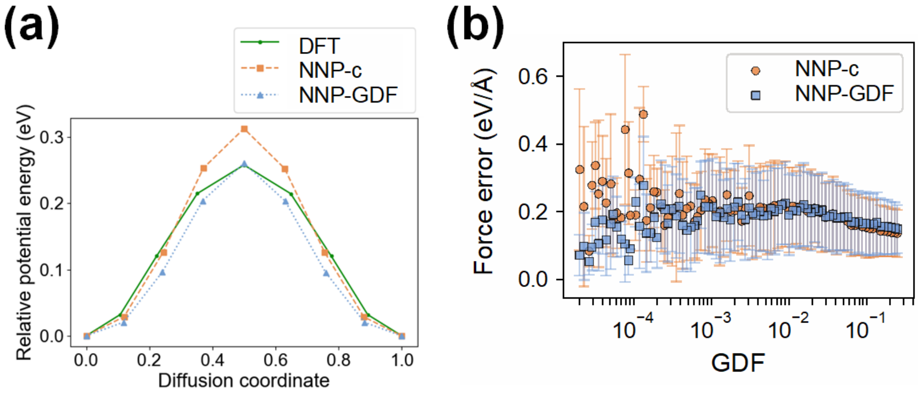

Fig. 3.2: (a) Vacancy migration barrier calculated by DFT, NNP-c, and NNP-GDF (b) GDF value versus force error for each training point. The data are interval-averaged along the GDF, and error bars are standard deviations.[70]

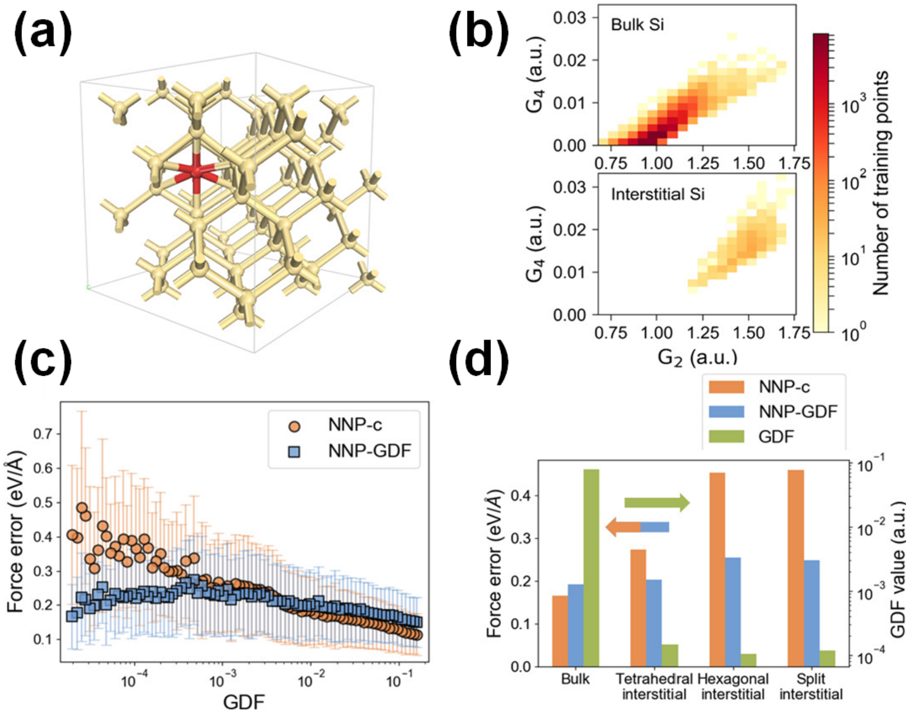

Fig. 3.3: The NNP training for crystalline Si with one interstitial. (a) The atomic structures used in training. The interstitial atom is marked in red. (b) The distribution of $52,000$ training points in the $\mathbf{G}$ space. The number of training points are enumerated on the 20× 20 mesh and color-coded in the log scale. (c) The GDF value versus force error for each training point. The results with NNP trained with the conventional method (NNP-c) are compared with those with the GDF weighting (NNP-GDF). The data are interval-averaged along GDF and error bars represent the standard deviation. (d) The root mean square force error and GDF values by the Si interstitial character. The force error is compared between NNP-c and NNP-GDF.[70]

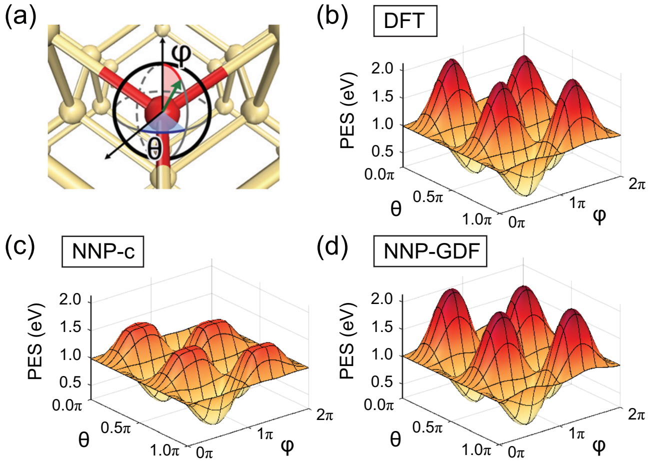

Fig. 3.4: PES of the tetrahedral Si interstitial around the equilibrium point. (a) Angular coordinates of the displaced interstitial. The radial distance from the equilibrium point is 0.6 Å. (b)–(d) PESs as a function of angular coordinates calculated by DFT, NNP-c, and NNP-GDF, respectively. The energy is referenced to the value at equilibrium for each method.[70]

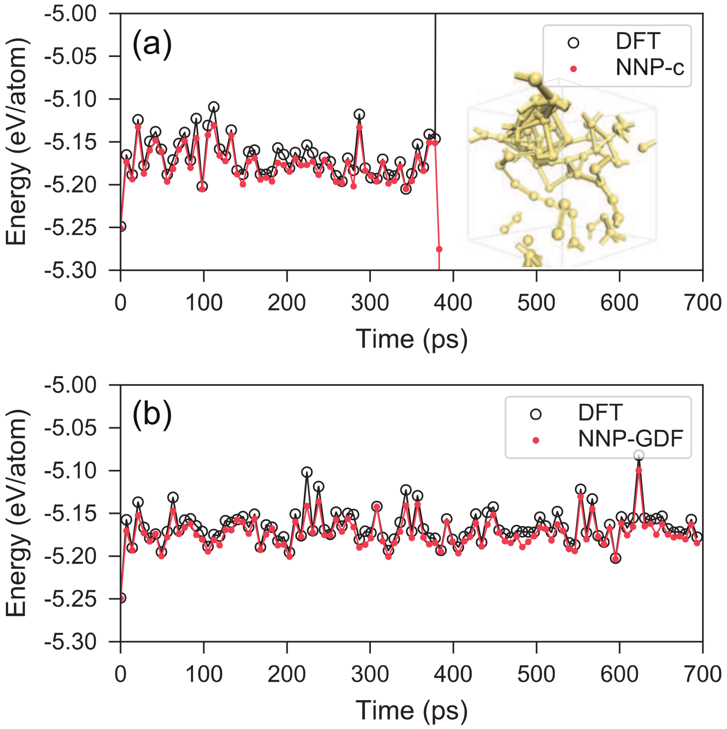

Fig. 3.5: Comparison of the MD trajectories obtained by conventional NNP (NNP-c) and NNP with GDF weighting (NNP-GDF). (a) The trajectory from NNP-c. (b) The trajectory from NNP-GDF. In both figures, black line and unfilled dot indicate the DFT results and the red line and solid dot indicates the NNP results. Inset figure of (a) describes the snapshot of the trajectory (a) (after 400 ps).[70]

ing set includes only mono-vacancy and perfect bulk structures. This will probe the transferability of the NNP to configurations that are slightly different from those in the training set. We first train NNP-c and NNP-GDF with the training set consisting of fcc Si structures that are prefect or include one vacancy. In detail, snapshots are sampled from DFT-MD simulations with the temperature ramped from 500 to 1300 K and crystals with various deformations. When calculating the vacancy cluster formation energy, $2\times2\times2$ supercells are used for mono- and tri-vacancies, while $3\times3\times3$ supercells are used for the penta-vacancy. The vacancy clusters are generated by removing Si atoms connected in the most compact way. We then calculate the formation energy (E_{form}) as E_{form} = E_{tot}(defect) $-$ N $\times$ E_{Si}, where E_{tot}(defect) is the total energy of the N-atom supercell with vacancies and E_{Si} is the total energy per atom of the perfect fcc Si. The results are summarized in Table 3.1. Even for the mono-vacancy, NNP-GDF gives a smaller error than NNP-c even though mono-vacancy structures are included in the training set, implying that NNP-GDF learns on the vacancy property better than NNP-c. More importantly, Table 3.1 shows that the prediction error for larger vacancies increases for NNP-c (up to 12.2%) while those for NNP-GDF remain similar to that of the mono-vacancy. This demonstrates that NNP-GDF has better transferability than NNP-c. The enhanced transferability will allow for simplification of the training set.

Clearly, the advantages of GDF weighting extend to other systems. For example, we carried out similar analysis on the Pd(111) surface with an oxygen adsorbate, which is a key step in various catalytic reactions and reconfirmed the merits of NNP-GDF.[71, 72] The training results again confirm that the GDF weighting improves the NNP quailty. (See Fig. 3.6)

Fig. 3.7 schematically depicts the main idea of the present method using the diagram of energy versus configuration coordinates. The training points indicated by circles are concentrated near the energy minimum or equilibrium point. In the conventional training (Fig. 3.7 (a)), training and prediction uncertainties (shaded region)

Table 3.1: Formation energies of vacancy clusters in eV calculated by DFT, NNP-c, and NNP-GDF.

	Mono-vacancy	Tri-vacancy	Penta-vacancy
DFT	3.59	7.14	9.87
NNP-c	3.34 (−6.8%)	7.83 (9.6%)	11.08 (12.2%)
NNP-GDF	3.45 (−4.0%)	7.09 (−0.6%)	10.19 (3.2%)

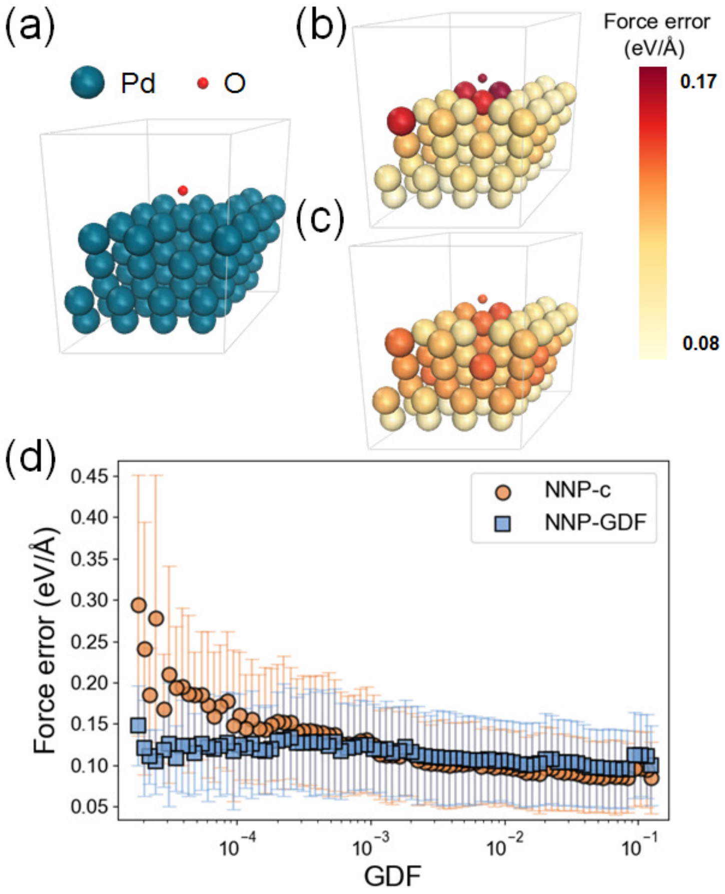

Fig. 3.6: (a) The structures of Pd(111) surface with one oxygen adsorbate. The big (small) spheres represent the Pd (O) atoms. (b) The force error given by NNP-c is averaged in the atom-wise manner and color-coded on each atom. It is seen that the force error is higher for O and neighboring Pd atoms. (c) A similar figure with NNP-GDF. The force error more even than in (b). (d) The GDF value versus force error for each training point. The data are interval-averaged along GDF and error bars represent the standard deviation.[70]

increase rapidly for underrepresented points. Through GDF weighting (Fig. 3.7 (b)), the uncertainty level becomes more even throughout the training range regardless of the sampling density. This also implies that the GDF weighting will effectively improve the transferability of NNP because new configurations lie outside of the training set (see stars in Fig. 3.7) and NNP-GDF would give a prediction error smaller than NNP-c. This schematic idea can be confirmed with actual systems by estimating the prediction uncertainty from multiple NNPs trained on the same data.

For a fixed size of neutral network, it is unavoidable to sacrifice the accuracy of the PES in some part in order to improve the accuracy in another part. This means that the GDF weighting can undermine the accuracy of physical properties at the equilibrium point with high GDF values (the bottom region in Fig. 3.7. For instance, Fig. 3.2 (b) and Fig. 3.3(c) show that the force error slightly increases for $\mathbf{G}$'s with high GDF values.

The data imbalance has been widely discussed within the machine learning community.[73] Suggested solutions are adding more features to the training set that has been undersampled (sometimes synthetic samples[74]) or leaving out some of the features that have been oversampled. However, it is not feasible to apply these methods to the present problem because the NNP predicts the atomic energy while it is trained through total energies that are the sum of atomic energies. This poses a unique challenge in training the NNP, which motivated us to suggest the present method based on the GDF.

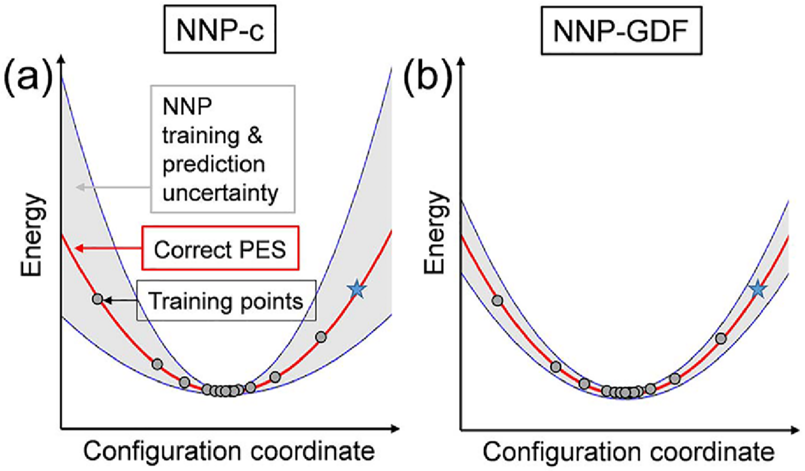

Fig. 3.7: Schematic diagram of training uncertainty with conventional NNP (NNP-c, (a)) and NNP with GDF weighting scheme (NNP-GDF, (b)). The gray circle indicates the known training points and the red line is the target function. The shade is the training uncertainty.[70]

3.4 Summary

In this chapter, we provided various examples of biased sampling and how it undermines the accuracy of the simulation. The examples include Si vacancy migration energy, the PES around Si interstitial defect, and MD stability of Si bulk with an interstitial. First, we defined the Gaussian density function to quantify the sparsity of training points and verified that the distribution of the training set points in the $\mathbf{G}$-space is highly biased. As the DFT energy per se cannot be split into local atomic energies, but the DFT forces are obtained for individual atoms, we modified the loss function such that the additional weighting term in the force loss function can enhance the influence of undersampled local environments. The results of the examples suggest that the GDF weighting scheme can effectively remedy the unbalanced training problem. In a bigger picture, the GDF weighting contributes to establishing a close correspondence between what one wants the NNP to learn and what the NNP actually learns through the training procedure, which is at the heart of every machine learning potential. Finally, we note that inhomogeneous feature sampling is a general issue for any machine learning potentials that adopt local feature vectors as input, the present method can be equally applied to other types of machine learning potentials.

Chapter 4

Prediction uncertainty quantification

4.1 Introduction

It is well-known that the prediction uncertainty of a machine learning model overgrows as input features deviate from the training domain. Machine learning potentials are no exception; the accuracy of the energy prediction of machine learning potentials degrades unacceptably if local atomic configurations are substantially different from those in the training set. If this happens during molecular dynamics (MD) simulations, computational results may not be fully meaningful even if the simulation terminates without any drastic failures such as diverging energies. Therefore, it is critical to monitor uncertainty levels during MD simulations when utilizing machine learning potentials. To note, such problems are less acute with traditional classical potentials because principle-based, hard-coded functions can safeguard atomic configurations against unintended structures.

While Gaussian approximation potential (GAP) can automatically estimate the prediction uncertainty using posterior predictive variances,[75] no such formula exists with neural network potential(NNP). In the machine-learning community, the prediction uncertainty of neural network models is often assessed by employing a model ensemble formed by varying the training data or network structure.[76] The output variances within the ensemble are used as an indicator of uncertainty. The ensemble method was also applied to NNP.[77–79] However, these ensemble methods require training multiple NNPs, which will impede the potential development for big training data. Also, as atomic energy mapping under given total energies is not unique, a certain degree of randomness is introduced in atomic energies each time the training of NNP in the ensemble is carried. (See 2.3.4 for more detailed discussions.) This should obscure the atomic-scale resolution of the reliability indicator. Thus it is difficult to localize atoms with high uncertainties among 10^3-10^5 atoms in the simulation box, making it hard to refine NNPs by augmenting the training set.

In this chapter, we present an efficient and atomic-resolution uncertainty indicator for NNP that is based on a "replica" ensemble. The NNPs in the replica ensemble differ in the network structure and initial weights, and they are trained directly over the atomic energies of the reference NNP that drives MD simulations. The training time for replica NNPs is much shorter than for the reference NNP, and the standard deviation within the ensemble plays as the atomic-resolution uncertainty indicator during MD simulations. We apply this method in simulating silicidation processes of Ni contacts in semiconductor devices and demonstrate that the method can reveal regions with high uncertainty at the atomic scale. By analyzing the problematic part, we can improve NNP with a repairing data set, thereby obtaining a reliable simulation.

4.2 Replica ensemble

In this section, we explain the concept of the replica ensemble as an uncertainty indicator. We first obtain a reference NNP in a standard way by training it over first-principles results based on the density functional theory (DFT). This reference NNP is used in evaluating potential energy surfaces for MD simulations. Similar to the previous literatures,[77–79] we employ an ensemble approach to gauge uncertainties in E_{tot} and E_{at}. The main difference in the present approach is that NNPs in the ensemble (called replica NNPs hereafter) directly learn atomic energies $\{E_{\mathrm{at}}(\mathbf{G}); \mathbf{G} \in \text{training set}\}$ output by the trained reference NNP (see Fig. 4.1 (a)). To calibrate numeric scales in uncertainty, we control the initial range of randomized NN weights and also diversify the network size of replica NNPs. The standard deviation in the atomic energy ($\sigma_{\mathrm{at}}(\mathbf{G})$) is calculated among replica NNPs and used as prediction uncertainty at $\mathbf{G}$. During the MD simulation, E_{at} of each atom is calculated by the reference NNP, and it is associated with the uncertainty value of σ_{at} from replica NNPs.

There are two benefits in employing such double-tier NNP: first, by training only over atomic energies, excluding derivatives such as forces and stresses, the computational cost of training replica NNPs becomes negligible compared to that for training the reference NNP over the whole set of properties. Second, replica NNPs by construction is destined to produce the same atomic energies for local configurations in the training set, but their inferences disagree for atoms with environments outside the training set. This leads to an atomic resolution in the uncertainty estimation by replica NNPs. As mentioned in the introduction, in the ensemble NNPs trained over total energies, such atomic resolution is not always guaranteed because of freedom in partitioning total energies into atomic contributions.

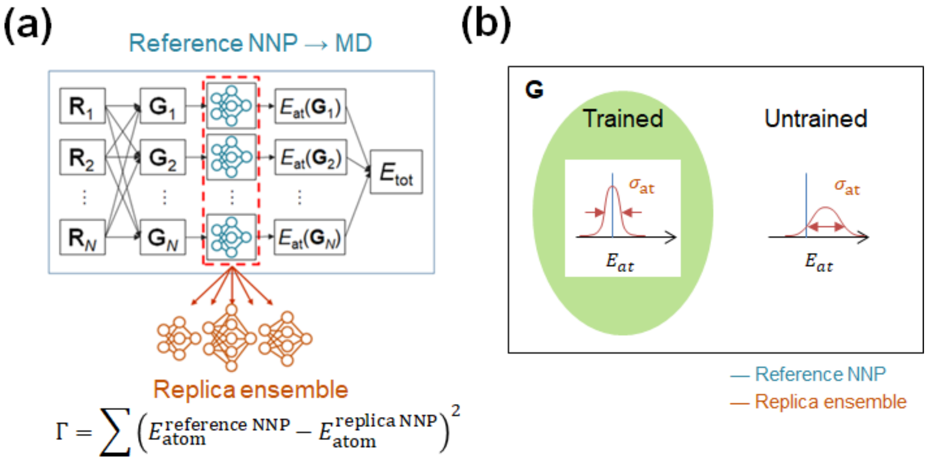

Fig. 4.1: (a) Schematic description of reference NNP and the replica NNP ensemble. Reference NNP drives MD simulations, and the replica NNP ensemble measures the uncertainty level of atomic configurations by the standard deviation (σ_{at}) in atomic energies. (b) Schematic description of how σ_{at} can be used as the uncertainty indicator of atomistic simulations. When the MD trajectory of a certain atom deviates from the trained area in the **G** space, (i.e, the grean area) the uncertainty in the replica ensemble becomes larger. In contrast, the uncertainty of atoms within the trained area remains small.[80]

4.3 Application: Ni silicidation simulation

As an example to apply the replica ensemble as an uncertainty indicator, we choose the Ni-silicidation process, one of the major processes in the fabrication of semiconductor devices. The metal-silicidation process reduces the contact resistance at gate-source/drain contacts by lowering Schottky barriers. The process itself is relatively simple; the metal layer is deposited on the Si substrate at room temperatures and annealed at 350–450 °C, which yields high-quality silicides with a controlled thickness.[81] Recently, the metal-silicidation process receives renewed attention in sub-10 nm device architectures because the mechanism of silicide formation changes when the deposited Ni thickness is below 4 nm.[82, 83] Furthermore, additive elements such as Pt and Co are known to control the silicide phase and redistribute dopants in the Si channel, but its microscopic understanding at the atomic scale is incomplete, which may hinder further optimization of the silicidation process in highly scaled devices.

In Ref. [84], the formation of Ni silicide on the Si(001) substrate was studied by first-principles MD simulations, focusing on the role of Pt as a stabilizer of the NiSi phase. However, the sheer computational costs of DFT calculations severely limited the system size and simulation time. Furthermore, to accelerate interface reactions, the simulations were executed at an unrealistically high temperature of 2000 K, which may affect the simulation results significantly in view of the temperature dependence of the process.[82, 85] These limitations of DFT calculations could be mitigated by employing NNP. However, the development of NNP aiming at the silicidation process is challenging because the system concurrently involves typical covalent (Si) and metallic (Ni) bonds, along with complicated mixed bonding ($\mathrm{Ni}_x\mathrm{Si}_y$). Furthermore, the bonding nature continuously changes during reactive interdiffusion of Ni and Si atoms, posing challenges in selecting a training set that can properly sample the configurational space explored during MD simulations. The complexity of the training set also underscores the importance of monitoring MD simulations using an uncertainty indicator.

4.4 Training set and validation of reference NNP

Table 4.1 summarizes the training set for the reference NNP. Besides static distortions of crystals, the training set includes DFT MD snapshots of crystals, liquids, amorphous structures, and surface slabs. (Fig. 4.2 and Fig. 4.3) Crystal structures are annealed at relatively high temperatures up to 2000 K because the highly exothermic reaction can significantly increase local temperatures at the interface. We also consider MD simulations of Ni–Si interfaces at various temperatures (300–1500 K) to train NNP over interface reactions directly. (Fig. 4.4) The training set contains 4,944 structures with 198,694 and 259,388 training points for Ni and Si, respectively.

From the total set in Table 4.1, 70% is randomly selected for training, and the remaining 30% is used in model validation. The architecture of the reference NNP is 70-70-70-1 for both Ni and Si, with 20,022 parameters in total. We train the model by fitting to total energies and atomic forces from the DFT calculations. We also employ a GDF weighting scheme aiming at uniform training of reference atomic structures. (See Section 3 for detailed discussion.) We transform symmetry function vectors by PCA without truncating dimensions. After the transformation, variances of all components are normalized (whitened). We also apply an L2 regularization to avoid overfitting. (See Section 2.3.3 for detailed explanation.)

Using the trained NNP, we compare NNP and DFT on key properties such as the equation of states for crystals, density and energy of amorphous structures, and radial distribution function of liquid and amorphous structures. The results show that all the properties agree well between NNP and DFT. (Not shown here) As a more direct test, we also perform Ni silicidation for a relatively small Ni–Si interface structure consisting of 60 Ni atoms and 120 Si atoms (see the top left figure in Fig. 4.5). This is the same structure as Interface(2) in Table 4.1, but the test simulation lasts for 20 ps, longer than 5 ps in the training set. The blue line in Fig. 4.5 shows total energies by NNP for the simulation time, and top figures are structures at selected instances. For

Table 4.1: Summary of Ni–Si interface reaction reference structures and root-mean-square errors (RMSEs) for the validation set

structure type	number of training points	temputre (K)	$\mathrm{RMSE}_{\mathrm{energy}}$	$\mathrm{RMSE}_{\mathrm{energy}}$
Ni	3704	0, 500–1500	3.6	0.16
Ni(001)	8000	1000	2.1	0.16
Si	7408	0, 500–1500	3.0	0.16
Si(001)	15360	1000	4.4	0.38
δ-Ni_2Si	18240	0, 1000–2000	3.9	0.28
l-Ni_2Si	24000	2000	5.1	0.45
a-Ni_2Si	9600	800	3.8	0.28
Ni_2Si ($l\rightarrow a$)	14400	2000–300	3.1	0.28
NiSi	12160	0, 1000–2000	7.1	0.45
l-NiSi	16000	2000	7.0	0.41
a-NiSi	6400	800	7.4	0.43
NiSi ($l \rightarrow a$)	9600	2000-300	6.8	0.41
α-$NiSi_2$	18240	0, 1000–2000	4.8	0.33
l-$NiSi_2$	24000	2000	5.2	0.30
a-$NiSi_2$	9600	800	8.3	0.50
$NiSi_2$ ($l \rightarrow a$)	14400	2000–300	5.5	0.49
interface (1)	108410	300, 1000, 1300, 1500	4.8	0.40
interface (2)	92160	1000, 1300	2.5	0.30
interface (3)	46400	1000	3.0	0.28
total	458082		5.1	0.34

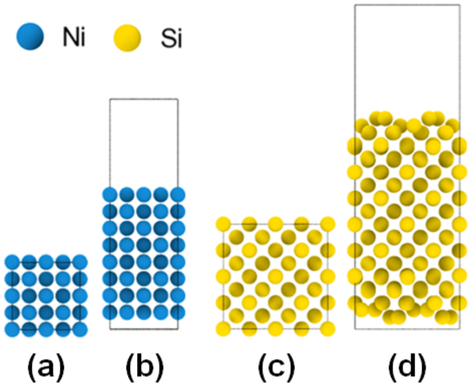

Fig. 4.2: The training set structures. (a) Ni crystal. (b) Ni(001) surface slab. (c) Si crystal. (d) Si(001) surface slab.[80]

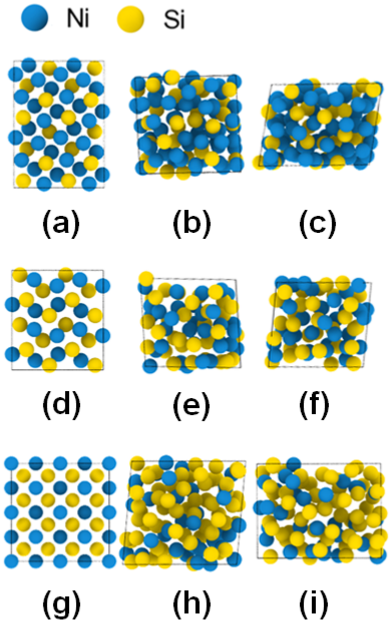

Fig. 4.3: The training set structures. (a) δ-Ni_2Si crystal. (b) Liquid Ni_2Si. (c) Amorphous Ni_2Si. (d) NiSi crystal. (e) Liquid NiSi. (f) Amorphous NiSi. (g) α-$NiSi_2$ crystal. (h) Liquid $NiSi_2$. (i) Amorphous $NiSi_2$.[80]

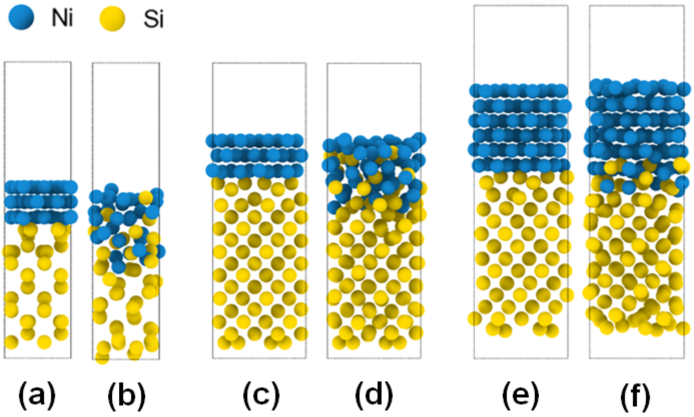

Fig. 4.4: The training set structures. (a) Interface(1) at 0 K. (b) Interface(1) after 4.4 ps of 1300 K MD simulation. (c) Interface(2) at 0 K. (d) Interface(2) after 2 ps of 1300 K MD simulation. (e) Interface(3) at 0 K. (f) Interface(3) after 2 ps of 1000 K MD simulation.[80]

comparison, DFT energies are calculated every 50 fs along the NNP-MD trajectory (see the green line). The overall agreements between NNP and DFT are good, and DFT well reproduces fine features in the total energy. However, the energy difference linearly increases with time (see the red line). This is attributed to a small discrepancy in the reaction energy (Ni + Si $\rightarrow$ NiSi) between DFT and NNP.

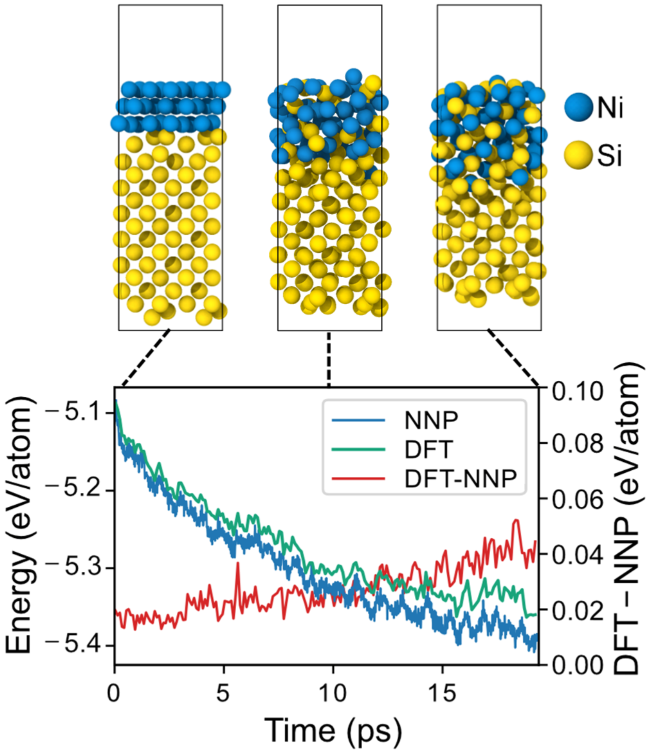

Fig. 4.5: Comparison of the DFT and NNP total energies per atom along MD trajectories for a small Ni–Si interface structure with 60 Ni atoms and 120 Si atoms. The blue line represents the energy from 1000 K NNP-MD with a time step of 1 fs. The green line represents the DFT energy of the NNP-MD trajectory sampled with a 50 fs interval. The red line with the right-hand *y* scale represents differences between DFT and NNP energies.[80]

4.5 Large-scale Ni silicidation simulation

For a large-scale simulation of the silicidation process using NNP, we construct a Ni/Si(001) interface slab of $\sim$8.5 nm thickness as shown in Fig. 4.6, which includes 3,008 Si and 2,080 Ni atoms. The Ni layer is 2 nm thick in the simulation, while Ni films with thickness 1–10 nm are deposited in experiments. The lattice parameters in the *xy* plane are fixed to those of crystalline Si at theoretical equilibrium. In Fig. 4.6, the bottom two layers are fixed, and the above three layers are heated to 1000 K with the NVT condition. The rest of the atoms are simulated in the NVE condition. This allows for simulating the transport of the heat generated during interfacial reactions and dissipated into the bottom layer through thermal conduction.

In the experiment, Ni is deposited at room temperatures. When we carry out the simulation at this temperature, only a few Ni and Si atoms react at the interface during 600 ps. Therefore, we start the high-temperature annealing right after the initial relaxation of the structure in Fig. 4.6. Furthermore, at the experimental annealing temperatures of 350–450 °C, the reaction almost stops after four Ni layers ($\sim$0.7 nm) are silicidated. To accelerate the dynamics and observe meaningful reactions within the simulation time, we increase the temperature to 726 °C or 1000 K.

Fig. 4.7 displays characteristic instances during the MD simulation carried out up to 1 ns. At the very early stage (t $<$ 0.2 ns), vigorous interfacial reactions are observed, which originates from the highly exothermic reaction of Ni + Si $\rightarrow$ NiSi (1.0 eV per formula unit). As a result, amorphous Ni$_x$Si$_y$ layers grow up to a thickness of $\sim$2 nm (see Fig. 4.7 (b)). For 0.2 $<$ t $<$ 0.5 ns, the concentration gradient of Ni decreases along the z-axis, and so does the driving force for the reactive diffusion. Consequently, the silicidation process gradually slows down. In Fig. 4.7 (c), the concentration profile within the Ni$_x$Si$_y$ layer is approximately linear with the atomic fraction of Si varying continuously from 0.55 to 0.3, implying that there is no phase separation along the z-direction. At around 0.6 ns, the L1$_2$ phase of NiSi nucleates and grows from the in-

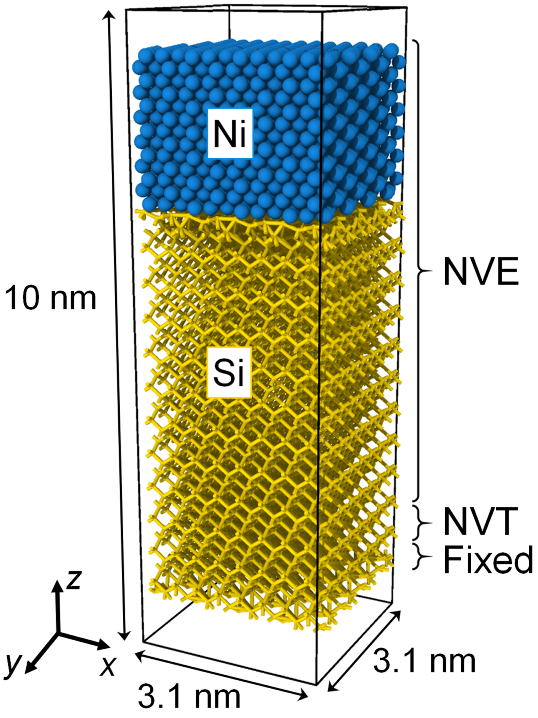

Fig. 4.6: Model structure of an 8.5-nm-thick Ni/Si(001) interface slab. The two bottom layers are fixed during the MD simulations. The next three layers are heated to 1000 K using the NVT ensemble. Other parts are simulated using the NVE ensemble.[80]

terface as shown in Fig. 4.7 (d). Once the crystalline phase with the thickness of $\sim$2 nm forms at the interface, the diffusion is virtually stopped, and no further interdiffusion is observed until 1 ns. As a result, the system arrives at a steady-state distribution comprising three distinct layers (a-Ni_2Si/$L1_2$-NiSi/Si).

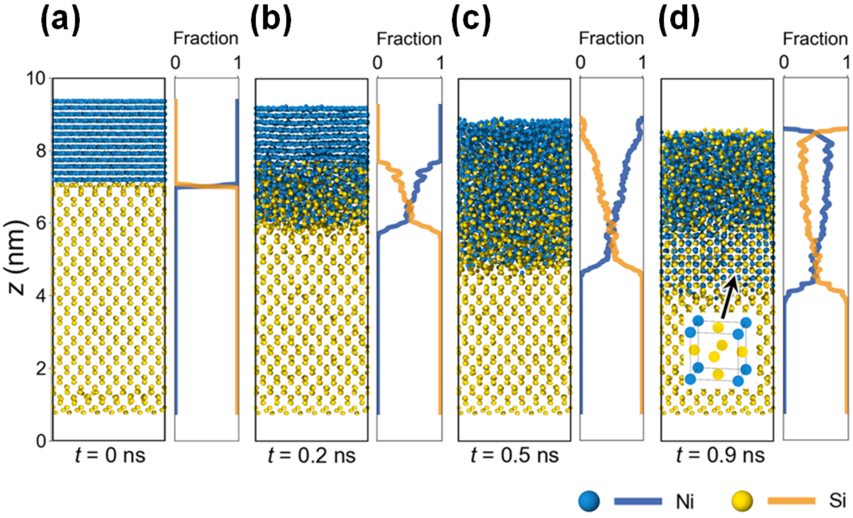

Fig. 4.7: Snapshots and concentration profiles of Ni and Si with respect to z from the large-scale NNP MD trajectory. (a) 0, (b) 0.2, (c) 0.5, and (d) 0.9 ns. The inset in (d) indicates that the $L1_2$ phase formed at the interface.[80]

4.6 Replica NNP training

Next, we check the soundness of the simulation using the replica ensemble. We train a replica ensemble consisting of five NNPs independently trained over atomic energies of the reference NNP at $\mathbf{G}$ values in the training set. To obtain output variances that are large enough for uncertainty quantification, NNPs in the ensemble should be decorrelated. This can be achieved by diversifying model parameters and training conditions such as initial weights, training/validation set selection from a data set (for example, bootstrap aggregating), network design such as the layer size and depth, and training hyperparameters. We find that varying the weight parameters and network size achieves a sufficient range of prediction variation among these methods. In detail, the initial weights are chosen from a normal Gaussian distribution with a zero mean and a standard deviation of σ_w. When σ_w is too small, the replica NNPs tend to converge, and energy variances of the ensemble become much smaller than actual errors relative to DFT results. On the other hand, if σ_w is too large, then NNPs easily fall into local minima, degrading training. To strike a balance, we choose σ_w = 3.0; when σ_w exceeds this value, some replica NNPs produce RMSEs larger than 100 meV, which is a signature of poor training quality. In addition to tuning σ_w, we also diversify the network structure (70-30-30-1, 70-50-50-1, 70-70-70-1, 70-100-100-1, and 70-120-120-1) and select a random 70% of atomic energies by the reference NNP as a training set, independently for each replica NNP. In training replica NNPs, we use the same hyperparameters as for the reference NNP. Fig. 4.8 shows the learning curves of the replica NNPs, and it is seen that resulting RMSEs range from 30 to 70 meV (RMSE of the ensemble average is 16 meV). Notably, the standard deviation among replica NNPs ($\sigma_{at}(\mathbf{G})$) is 50 meV for $\mathbf{G}$ in the training set, confirming that the prediction uncertainty is low for the well-learned local configurations. Note that RMSEs of replica NNPs are an order of magnitude larger than RMSEs for typical NNPs, usually a few meV/atom. This is because direct learning of the atomic energy is far more constrained than learning total energies that are sums of atomic energies. In addition, σ_w is larger than for regular training, causing the training result to fall into local minima.

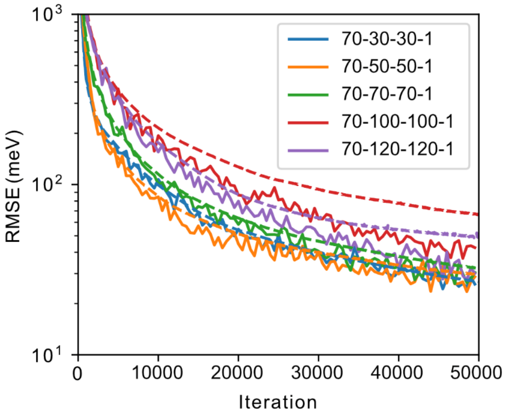

Fig. 4.8: The training set (solid lines) and validation set (dashed lines) root mean square error (RMSE) of atomic energy during the replica NNP training iterations.[80]

4.7 Uncertainty quantification on Ni silicidation simulation

4.9 (a) shows atom-by-atom coloring of σ_{at} for instances selected in Fig. 4.7. The frequency histogram of σ_{at} for the whole simulation is shown in Fig. 4.9 (b), and it is seen that σ_{at} is mostly less than 50 meV, which is the average σ_{at} for training points, and thus can be used as a uncertainty threshold. As such, MD simulations appear to be reasonable for most configurations as confirmed by dominantly light colors in Fig. 4.9 (a). Nevertheless, the uncertainty indicator also detects concerning values of σ_{at} $\geq$ 0.1 eV for a number of regions near Ni/NiSi and NiSi/Si interfaces as indicated by blue (Ni) and red (Si) atoms. Most notably, σ_{at} is the largest for the $\mathrm{L1}_2$ phase. In fact, the $\mathrm{L1}_2$ phase was not considered in the training set, and DFT calculations show that the$\mathrm{L1}_2$-NiSi phase is unstable against the most stable MnP-type NiSi by 0.21 eV/atom. However, the present NNP infers $\mathrm{L1}_2$ to be more stable than the MnP type by 0.08 eV/atom. This means that the nucleation of the $\mathrm{L1}_2$ phase is an artifact triggered by a subtle deviation from the training set.

The atomic resolution of σ_{at} allows for localizing simulation defects temporally as well as spatially. At ~0.5 ns, a cluster of atoms near the NiSi/Si interface begins to develop high uncertainties (see the round box in Fig. 4.9 (a)), which later nucleates into the $\mathrm{L1}_2$ phase. Also, some atoms near the NiSi/Ni interface and Si atoms at the Ni surface exhibit notable uncertainties. These errors should follow from deficiencies of the interface models in the training set (see Table 4.1). By close inspections, we find that $\mathrm{Ni}_x\mathrm{Si}_y$ layer in the training set is rather Ni-rich compared to the boxed region in Fig. 4.9 (a). This is because the simulation time is too short to permit sufficient interdiffusion. Furthermore, there are only three Ni layers in Interface(1–2) models, which are too thin to properly sample Ni/NiSi interfaces.

Inspired by the above analysis, we augment the training set with additional structures from DFT calculations. We first add MD trajectories of a model with five Ni layers (similar to Interface(3)) at the higher temperature of 1300 K and a longer sim-

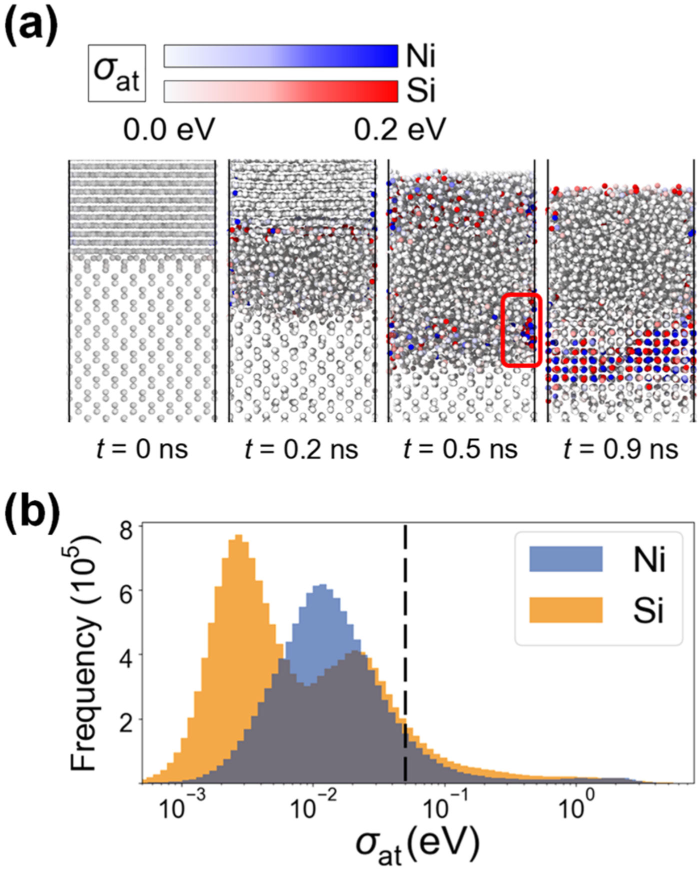

Fig. 4.9: (a) Snapshots from the Ni-silicidation MD trajectory at 0, 0.2, 0.5, and 0.9 ns with the color scale by σ_{at}. (Parts of the bottom Si layers are removed.) (b) Histogram of σ_{at} collected over the 1 ns simulation.[80]

ulation time of 40 ps, thereby sampling interfacial compositions similar to NNP-MD simulations. Furthermore, we add annealing trajectories at 1000–2000 K for crystals such as γ-$Ni_{31}Si_{12}$, β1-Ni_3Si, β2-Ni_3Si, ϵ-Ni_3Si_2, θ-Ni_2Si, and NiSi (ϵ-FeSi type) to calibrate formation energies of Ni_xSi_y mixed layers with fine stoichiometric variations. We also consider a surface model of amorphous NiSi because Si atoms with high uncertainties are found at the surface region (see Fig. 4.9 (a) for t = 0.9 ns). These supplementary structures add 258,030 points to the final training set, which is then used for training a refined NNP (r-NNP). The RMSEs of r-NNP are 5.62 meV/atom (energy) and 0.23 eV/Å (force) for the validation set. The computational results on test structures are very similar between r-NNP and the original NNP.

Fig. 4.10 (a–d) show the sequence of silicidation process simulated by r-NNP up to 0.9 ns (total simulation time is 3.6 ns). The model and simulation conditions are the same as in the previous simulation and the prediction uncertainties by σ_{at} in Figs. 4.10 (e) and (f) are evaluated similarly to Fig. 4.9. In contrast with Fig. 4.9 (b), almost all the atoms exhibit $\sigma_{\mathrm{at}} < 0.1$ eV throughout the whole simulation except for brief fluctuations above the threshold, indicating that the supplementary training structures enabled reliable and stable simulations. Overall, the simulation proceeds similarly to Fig. 4.9 up to $\sim$0.5 ns, namely, vigorous initial reactions until 0.2 ns followed by sluggish interdiffusion. The amorphous Ni_xSi_y layer is thicker than the previous simulation by $\sim$1.5 nm. Unlike the previous simulation, the incubation of any crystal phase is absent with r-NNP, which contributes to the formation of thick interfacial layers (see the dashed box in Fig. 4.10 (c)) rather than the abrupt interface in Fig. 4.7 (c). After 0.6 ns, the thickness of the Ni_xSi_y layer remains nearly constant, and atoms diffuse mainly within the amorphous layer. Further simulation up to 3.6 ns only changes the stoichiometry within the amorphous layer close to 1:1 (not shown). In experiments, epitaxial $NiSi_{2-\delta}$(δ = 0.5–0.6) phases appear at high annealing temperatures.[86] Energy differences between crystal and amorphous phases are similar between r-NNP and DFT, so amorphous NiSi will eventually crystalize in the simulation. However, the incubation time for crystallization is much longer than the feasible simulation time, so we could not observe the formation of crystalline phases within 3.6 ns.

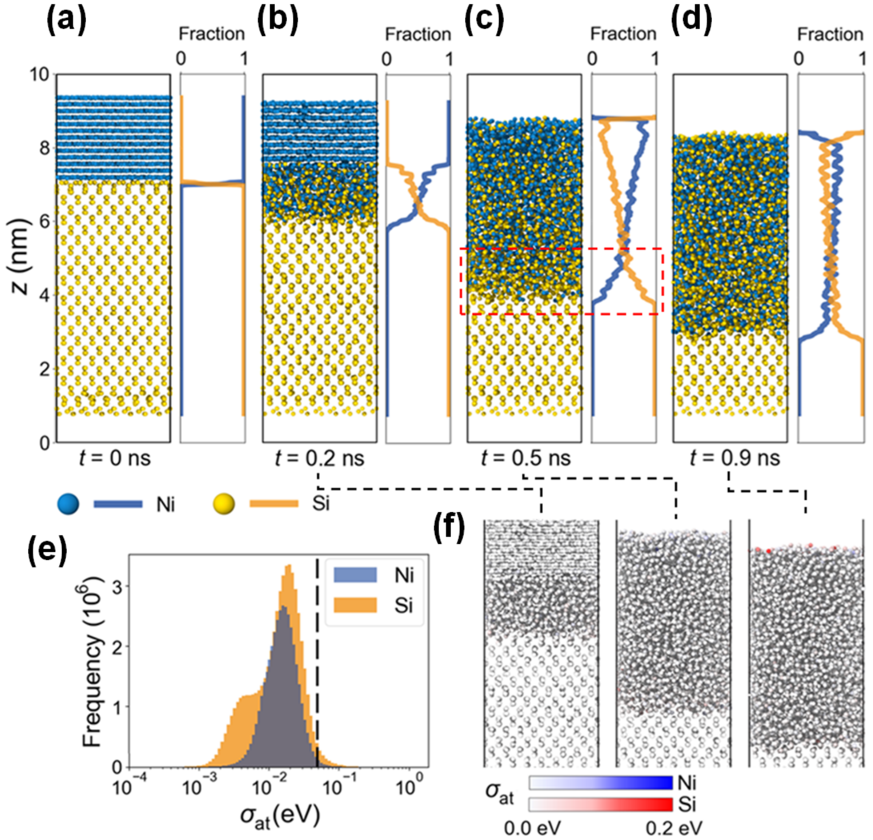

Fig. 4.10: Snapshots and concentration profiles of Ni and Si with respect to z from the large-scale r-NNP MD trajectory at (a) 0, (b) 0.2, (c) 0.5, and (d) 0.9 ns. (e) Histogram of σat of the 3.6 ns r-NNP simulation trajectory atoms. (f) Snapshots from the large-scale Ni silicidation r-NNP MD trajectory at 0.2, 0.5, and 0.9 ns. The color intensity is scaled by σ_{at}. (Part of the bottom Si layers are removed.)[80]

4.8 Schottky barrier height estimation

For further analysis on the electronic property, we theoretically estimate the Schottky barrier height (SBH, Φ_b) for the as-obtained $\mathrm{Ni}_x\mathrm{Si}_y$/Si interface. The $\mathrm{Ni}_x\mathrm{Si}_y$/Si interface structure is obtained from the r-NNP MD simulation of the model structure after 3.6 ns of 1000 K MD. Before the DFT calculation, atomic positions are relaxed with r-NNP at 0 K. Then, the interface region of the whole structure is sliced to reduce the number of atoms below 1000. Dangling bonds of the bulk-terminated Si surface are passivated with H atoms. As a result, the interface structure consists of 7 layers of Si and 6.2-Å thick $\mathrm{Ni}_x\mathrm{Si}_y$ layer (see Fig. 4.11). To address the finite-size effect, the crystalline Si is also computed independently. The SBHs for n- and p-type Si are obtained as follows:

$$\Phi_{b,n} = E_C - E_F - (\langle V \rangle_{\mathrm{Si}}^{\mathrm{bulk}} - \langle V \rangle_{\mathrm{Si}}^{\mathrm{interface}}) \tag{4.1}$$

$$\Phi_{b,n} = E_F - E_V + (\langle V \rangle_{\mathrm{Si}}^{\mathrm{bulk}} - \langle V \rangle_{\mathrm{Si}}^{\mathrm{interface}}) \tag{4.2}$$

where E_C and E_V denote the conduction and valence band edges obtained from the bulk Si structure, and E_F indicates the Fermi energy obtained from the $\mathrm{Ni}_x\mathrm{Si}_y$/Si interface structure. In Eqs. (4.1) and (4.2), $\langle V \rangle_{\mathrm{Si}}^{\mathrm{interface}}$ refers to the average electrostatic potential of bulk Si part in $\mathrm{Ni}_x\mathrm{Si}_y$/Si interface system, while $\langle V \rangle_{\mathrm{Si}}^{\mathrm{bulk}}$ is the corresponding value for the crystalline Si.[87] In addition, to compensate the band-gap underestimation in semilocal functionals, the band edge energies are corrected with the hybrid functional method (HSE06).[17, 88] As a result, the interface model gives SBH of 0.83 and 0.21 eV for $\Phi_{b,n}$ and $\Phi_{b,p}$, respectively, which is in reasonable agreement with experimental values of 0.7 and 0.35 eV for NiSi/Si interfaces.

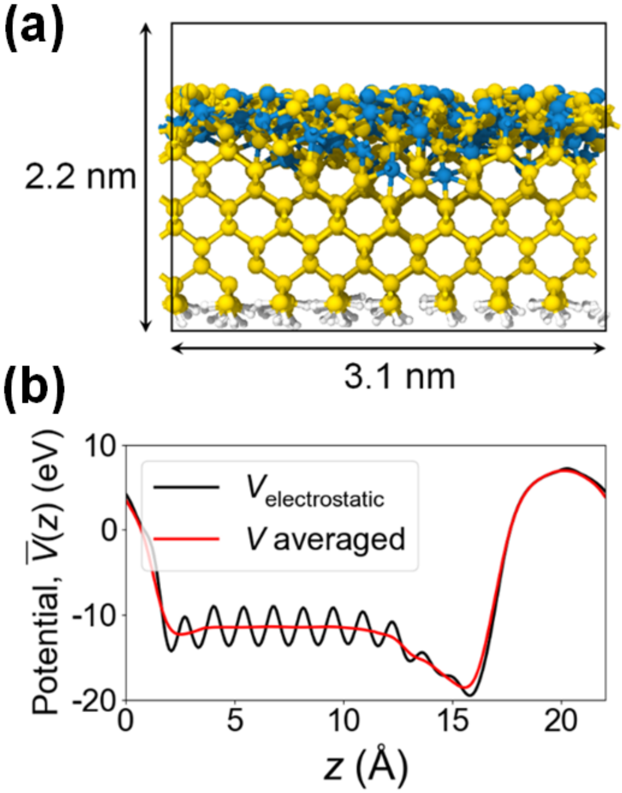

Fig. 4.11: (a) Model structure of the Ni_xSi_y/Si interface slab for the Schottky barrier height estimation. The dangling bonds of bulk-terminated Si surface are passivated with H atoms. (b) The black line represents the electrostatic potential of the Ni_xSi_y/Si interface slab averaged to the z-axis. The red line represents the macroscopic average of the potential with a sampling window set to the layer spacing of the Si(001) slab.[80]

4.9 Summary

In summary, we proposed the replica ensemble that enables efficient and atomic-resolution uncertainty estimation in energy prediction by NNPs. The replica NNPs are trained directly over the atomic energies of the reference NNP that calculates potential energy surface during MD simulations. By excluding energy derivatives from the loss function, the construction of the replica ensemble takes much less time than for conventional NNPs. The usefulness of the suggested indicator was demonstrated by simulating the Ni silicidation process. The replica ensemble was able to trace highly uncertain atomic configurations at interfaces during the MD simulation, which later evolved into the formation of the unphysical $L1_2$ phase. The uncertainty analysis revealed structures that were deficient in the training set. With the augmented training set, a refined NNP was developed, and reliable Ni silicidation simulation was carried out until 3.6 ns.

Chapter 5

Crystal structure prediction by machine learning potentials

5.1 Introduction

Crystal structure prediction (CSP) for given chemical composition is among the grand challenges in condensed matter physics.[89] The goal of CSP is to identify atomic arrangements in space that produce the lowest free energy under given thermodynamic conditions. Mathematically, this is equivalent to the global optimization in a high-dimensional space, to which there is no general solution.[90] Nevertheless, various heuristic methods have been developed for navigating the gigantic configurational space efficiently and intelligently such as random structure sampling,[91, 92] simulated annealing,[93] particle-swarm optimization,[94, 95] minima hopping,[96] basin hopping,[97] metadynamics,[98] and evolutionary algorithm.[99] In evaluating the objective function or free energy, the method of choice is the first-principles calculations based on the density functional theory (DFT). The non-empirical nature of DFT allows for exploring the energy landscape with little restrictions yet achieving high precisions. The DFT-based CSP has been successfully applied to identifying structures of organic crystals, superconducting materials, and inorganic crystals under extreme conditions.[100–103] One intriguing field to apply CSP is the structure prediction of ternary or higher (simply multinary hereafter) inorganic crystals at ambient conditions. Its importance arises from the low throughput of crystal synthesis: while structures of most unary and binary compounds were thoroughly investigated by X-ray crystallography, only about 16% and 0.6% have been revealed within the ternary and quaternary spaces, respectively. [104]

There have been several theoretical attempts to identify the stable structures of unknown ternary systems.[102, 103, 105–110] However, the previous works often limited the configurational space by referring to available prototypes or fixing the number of atoms in the unit cell. The main reason for the restrictive searching is the sheer computational cost of DFT calculations. To predict the equilibrium structure of multinary crystals from exhaustive searching, an exponential increase of possible atomic arrangements demands evaluating the objective function far more efficiently than DFT. The classical potential is computationally cheap, which allowed for predicting stable structures of even quinary compounds.[111] However, scarcity and low fidelity of classical potentials prohibit wide use as an objective function. Therefore, machine learning potential is poised to meet the requirement for evaluating energies in multinary CSP.

The development style of machine learning potential invokes a distinct challenge when applying machine learning potential to CSP for unknown multinary compounds. That is to say, machine learning potential infers total energies after learning on DFT results for reference structures. In usual practices, machine learning potentials are first trained over structures derived from known crystals. However, such information is not available at the outset in CSP for unknown compounds, and one should construct machine learning potentials out of 'nothing'.

Motivated by the above discussions, we propose a way to construct machine learning potential as a hi-fidelity surrogate model of DFT, mainly targeting to predict the most stable structure of multinary inorganic crystals at ambient conditions. The key strategy is to train a machine learning potential over disordered structures such as liquid and melt-quenched amorphous phases. With only compositional information, the molecular dynamics (MD) simulation on liquids can self-start from a random distribution and quickly equilibrates at sufficiently high temperatures (well above melting points), which is then cooled to amorphous structures. Thus, it is feasible to build the whole training set without preceding knowledge of the crystal structure. Furthermore, short-range orders in the amorphous phase resemble those in the crystalline phase (for example, consider amorphous Si and SiO_2), and local fluctuations in liquid and amorphous phases also sample diverse local orders that can appear in crystals. Therefore, it is anticipated that the trained machine learning potential provides correct energies for stable as well as metastable phases, thus properly ranking energies of various structures emerging from search algorithms.

5.2 Test materials

To test the concept, we choose four compounds (three ternary and one quaternary) from Inorganic Crystal Structure Database (ICSD):[112] Ba_2AgSi_3 (*Fddd*), Mg_2SiO_4 (*Pnma*), $LiAlCl_4$ ($Pmn2_1$) and $InTe_2O_5F$ ($C222_1$). (See Fig. 5.1.) These four materials encompass diverse structural motifs such as layers, intercalations, and shared polyhedra. In addition, they are employed in a variety of applications owing to their mechanical, electrical, and optical properties: Ba_2AgSi_3 is a member of the Ba-Ag-Si system, which is anticipated for potential high-T_c superconductors,[113] and has a layered structure formed by Si_6.[114] Mg_2SiO_4 (also known as forsterite) features high fracture toughness and is actively studied as bioceramic implants.[115] It has a shared-polyhedra structure with Mg and Si occupying octahedral and tetrahedral sites, respectively. There are also three additional structures of Mg_2SiO_4 available in ICSD with the space group of *Ibmm*, $Fd\bar{3}m$, and $Cmc2_1$. $LiAlCl_4$ is an archetypal halide solid-state Li-ion conductor.[116] There are two structures of $LiAlCl_4$ in ICSD with the space group of $P2_1/c$ and $Pmn2_1$. Among them, we find that the $Pmn2_1$ structure has a slightly lower DFT energy at 0 K. $LiAlCl_4$ has a relatively simple structure where both Li and Al ions are tetrahedrally coordinated by Cl ions. Lastly, $InTe_2O_5F$ is anticipated for nonlinear optical applications owing to a non-centrosymmetric structure.[117] The In ions occupy octahedral sites surrounded by four O and two F ions.

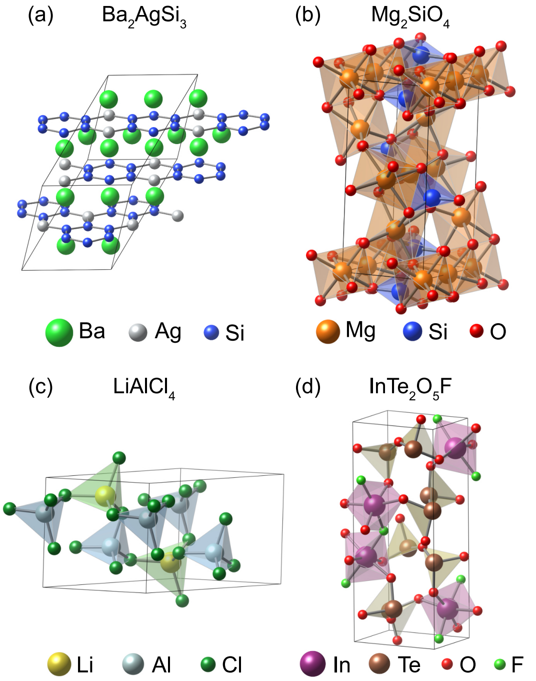

Fig. 5.1: Unit cells of the stable phase of (a) Ba_2AgSi_3, (b) Mg_2SiO_4, (c) $LiAlCl_4$, and (d) $InTe_2O_5F$.[118]

5.3 Construction of training set

To generate training structures for NNP to be used in CSP, we carry out first-principles molecular dynamics (FPMD) simulations on the melt-quench-annealing process for each material. We first determine the melting temperature (T_{m}) and simulation volume as follows: the initial structure is prepared by randomly distributing $\sim$100 atoms for the given stoichiometry, which is then superheated at 4000 K for 5 ps. Next, we perform FPMD simulations by lowering the temperature gradually and select an ad hoc T_{m} as the lowest temperature at which the mean square displacement of atoms linearly increases with time. The determined T_{m}'s are 1500, 3500, 1500, and 2000 K for $\mathrm{Ba_2AgSi_3}$, $\mathrm{Mg_2SiO_4}$, $\mathrm{LiAlCl_4}$, and $\mathrm{InTe_2O_5F}$, respectively. (The experimental T_{m}'s are 2174 K [119] and 419 K [120] for $\mathrm{Mg_2SiO_4}$ and $\mathrm{LiAlCl_4}$, respectively.) The cell volume is then adjusted such that the average hydrostatic pressure is equal to zero. Using the obtained T_{m} and cell parameters, we generate liquid-phase trajectories for 20 ps in the NVT condition. Subsequently, the liquid is quenched with a cooling rate of 100 K/ps from T_{m} to 300 K, and then annealed at 500 K for 15 ps to sample amorphous structures. The training set samples the whole melt-quench-annealing trajectory every 20 fs, and consists of 2,400–3,200 snapshots.

5.4 Crystal structure prediction by genetic algorithm

To sample metastable structures for each material, CSP is performed by an evolutionary algorithm implemented in the Universal Structure Predictor: Evolutionary Xtallography (USPEX) package,[121] while DFT calculations carry out the energy evaluation. We fix the number of formula units (Z) to that of the stable structure (4 for every material). We set the population size to 20–60, which increases with the number of atoms in the unit cell. Initial structures are generated by either random symmetric [122] or topological structure generators.[123] The succeeding structures are produced by random generators and evolutionary operators, including heredity, permutation, soft mutation, and lattice mutation. The ratio of variation operators in USPEX is set automatically, encouraging the operators to produce more diverse structures in the low-energy spectrum.[123] The generated structures are fully relaxed (both atomic positions and lattice vectors) until atomic forces and total stress is less than 0.1 eV/Å and 20 kbar, respectively, or the number of relaxation steps reaches 400. Also, we turn on the antiseed option, which prevents the evolution from being trapped in local minima by adding repulsive Gaussian potentials for sampled structures.[124] We collect metastable structures generated during the whole evolution and use them in benchmarking NNPs. For an accurate evaluation of energies, the metastable structures are further relaxed by DFT calculations until atomic forces and stresses are less than 0.02 eV/Å and 4 kbar.

5.5 Comparison of NNP and DFT energies

To judge the suitability of the developed NNPs for multinary CSP, we compare NNP and DFT energies for metastable structures collected by USPEX and experimental structures in ICSD. During 10–20 generations, USPEX garnered on average $\sim$274 metastable structures with energies relative to that of the stable phase (ΔE_{tot}) less than 500 meV/atom. The lowest ΔE_{tot}'s are 46.5, 28.2, 1.9, and 33.2 meV/atom for $\mathrm{Ba_2AgSi_3}$, $\mathrm{Mg_2SiO_4}$, $\mathrm{LiAlCl_4}$, and $\mathrm{InTe_2O_5F}$, respectively. (The package could not identify experimental structures.) We note that many metastable structures share similar atomic configurations, such that slight shifts of a few atoms relax one metastable structure to another.

Using NNPs trained over disordered structures, we evaluate energies for metastable structures without further relaxations and compare them with DFT energies in Fig. 5.2. It is seen that the NNP and DFT energies are highly correlated, and Pearson coefficients among the structures with $\Delta E^{\mathrm{DFT}}_{\mathrm{tot}} < 200$ meV/atom are 0.769, 0.864, 0.977, and 0.962 for $\mathrm{Ba_2AgSi_3}$, $\mathrm{Mg_2SiO_4}$, $\mathrm{LiAlCl_4}$, and $\mathrm{InTe_2O_5F}$, respectively. This is striking because none of the metastable structures were explicitly included in the training set. Thus, it is confirmed that the training set could sample the structural motifs appearing in low-energy metastable phases. It is also intriguing in Fig. 5.2 that the NNPs consistently predict the most stable experimental structure to be more stable than any theoretical structures (see red squares). This is the case even if the structures are relaxed using the NNPs. Therefore, by combining the NNPs and search algorithms, one should be able to identify the stable phase in principle without calling for additional DFT calculations.

Each disordered structure in the training set consists of various local configurations similar to structural motifs in metastable phases. The strong energy correlations in Fig. 5.2 suggest that the machine-learning procedure successfully delineated local energies without ad hoc energy mapping.[68] In Fig. 5.2, it is also seen that the errors

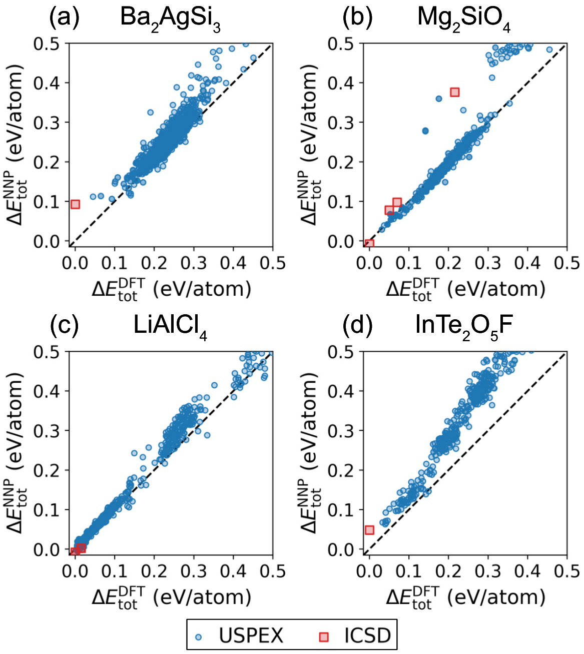

Fig. 5.2: Correlation between DFT and NNP energies. Structures are fixed to metastable structures from USPEX (blue circles) or experimental structures from ICSD that are relaxed by DFT (red squares). In both $\Delta E_{\mathrm{tot}}^{\mathrm{NNP}}$ and $\Delta E_{\mathrm{tot}}^{\mathrm{DFT}}$, the reference energy is the DFT energy of the stable phase. The experimental structures are plotted as red squares. (a) Ba_2AgSi_3, (b) Mg_2SiO_4, (c) $LiAlCl_4$, and (d) $InTe_2O_5F$.[118]

of NNP prediction increase with ΔE_{tot}. This is because structural features in high-energy phases were not sufficiently sampled in the liquid and amorphous structures. (For reference, ΔE_{tot} of the amorphous structure is 60–180 meV/atom.)

In Fig. 5.2 (a), we note that the energy scale is not well resolved for low-energy structures of Ba_2AgSi_3. For instance, ΔE_{tot} for the third-lowest metastable structure is 93 meV/atom in DFT but it is only 14 meV/atom in NNP. This is attributed to deficiencies in the training set: in the disordered phases, we find that the hexagonal Si ring in the crystalline structure is absent and linear Si chains embedded with Ag atoms are prevalent. Since the low-energy structures mainly differ in the connection topology of Si chains, the energy prediction in this region becomes rather inaccurate. This sampling problem is caused by the high cooling rate of 100 K/ps, which may not provide enough time to establish medium-range orders such as hexagonal rings. However, this can be remedied by coupling metastable structures obtained from the NNP-CSP algorithm to the training set.[118]

In Fig. 5.3, using the principal component analysis, we examine element-by-element distributions of **G** vectors in the training set and (meta)stable structures of Ba_2AgSi_3. It is seen that most **G** points from USPEX-generated and experimental structures lie within those from the training set. (Analysis on other principal axes show a similar relationship.) Other materials also exhibit **G**-point distributions similar to Ba_2AgSi_3. (The only exception is Al in $LiAlCl_4$, where Al atoms from high-energy metastable structures are distinct from the training set.) Therefore, we explicitly confirm that the local motifs of USPEX-generated structures and ICSD structures are well-included in a region spanned by the training set.

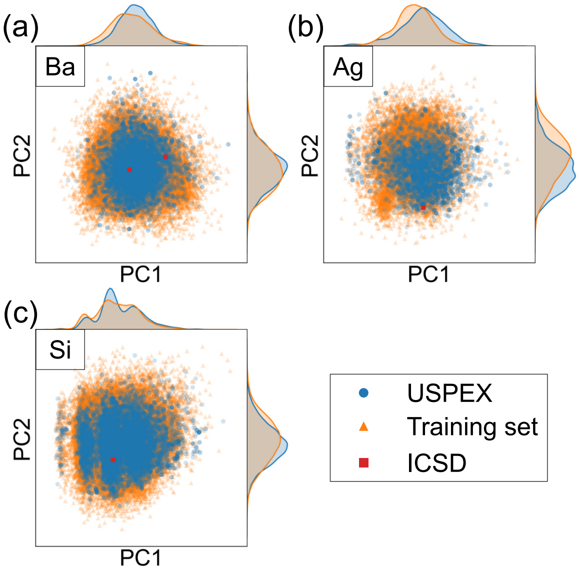

Fig. 5.3: The distribution of **G** vectors in Ba_2AgSi_3 projected onto the first two principal component axes (PC1 and PC2). The distribution of (a) Ba atoms, (b) Ag atoms, and (c) Si atoms. The projected density on each axis is plotted on the top and side.[118]

5.6 Summary

In summary, we proposed a way to train NNPs using disordered structures sampled from liquid-quench-annealing MD trajectories. From the strong correlations of NNP and DFT energies among diverse metastable structures, it was confirmed that the NNPs can properly rank energies of structures that emerge from search algorithms. As a future work, more NNP-oriented and efficient CSP algorithm should be developed to be used with NNP. In conclusion, by proposing a way to develop machine-learning potentials for CSP, this work will pave the way to identifying unexplored multinary phases efficiently.

Chapter 6

Conclusion

We addressed three significant challenges of machine learning potentials and, with the example of neural network potential, suggested the solution to the challenges. First, we provided various examples of biased sampling and how it undermines the accuracy of the simulation. The examples include Si vacancy migration energy, the PES around Si interstitial defect, and MD stability of Si bulk with an interstitial. First, we defined the Gaussian density function to quantify the sparsity of training points and verified that the distribution of the training set points in the $\mathbf{G}$-space is highly biased. As the DFT energy per se cannot be split into local atomic energies, but the DFT forces are obtained for individual atoms, we modified the loss function such that the additional weighting term in the force loss function can enhance the influence of undersampled local environments. The results of the examples suggest that the GDF weighting scheme can effectively remedy the unbalanced training problem. Next, we proposed the replica ensemble that enables efficient and atomic-resolution uncertainty estimation in energy prediction by NNPs. The replica NNPs are trained directly over the atomic energies of the reference NNP that calculates potential energy surface during MD simulations. By excluding energy derivatives from the loss function, the construction of the replica ensemble takes much less time than for conventional NNPs. The usefulness of the suggested indicator was demonstrated by simulating the Ni silicidation process. The uncertainty analysis revealed structures that were deficient in the training set. Finally, we proposed a way to train NNPs using disordered structures sampled from liquid-quench-annealing MD trajectories. From the strong correlations of NNP and DFT energies among diverse metastable structures, it was confirmed that the NNPs could properly rank energies of structures that emerge from search algorithms. Therefore it is expected that the proposed way will pave the way to identifying unexplored multinary phases efficiently. In conclusion, by addressing the critical challenges and their solutions, we believe that this dissertation will extend the application range of the machine learning potentials.

Bibliography

[1] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, Journal of Computational Physics **285**, 316 (2015).

[2] A. V. Shapeev, Multiscale Modeling & Simulation **14**, 1153 (2016).

[3] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, Science advances **3**, e1603015 (2017).

[4] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nature communications **8**, 1 (2017).

[5] J. S. Smith, O. Isayev, and A. E. Roitberg, Chemical science **8**, 3192 (2017).

[6] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Physical Review Letters **104**, 136403 (2010), ISSN 0031-9007.

[7] J. Behler and M. Parrinello, Physical review letters **98**, 146401 (2007).

[8] G. Sun and P. Sautet, Journal of the American Chemical Society **140**, 2812 (2018).

[9] Z. W. Ulissi, M. T. Tang, J. Xiao, X. Liu, D. A. Torelli, M. Karamad, K. Cummins, C. Hahn, N. S. Lewis, T. F. Jaramillo, et al., ACS Catalysis **7**, 6600 (2017).

[10] D. Lee, K. Lee, D. Yoo, W. Jeong, and S. Han, Computational Materials Science **181**, 109725 (2020).

[11] M. Born and R. Oppenheimer, Annalen der physik **389**, 457 (1927).

[12] P. Hohenberg and W. Kohn, Physical review **136**, B864 (1964).

[13] W. Kohn and L. J. Sham, Physical review **140**, A1133 (1965).

[14] J. P. Perdew and W. Yue, Physical review B **33**, 8800 (1986).

[15] A. D. Becke, Physical review A **38**, 3098 (1988).

[16] L. Hedin, Physical Review **139**, A796 (1965).

[17] J. Heyd, G. E. Scuseria, and M. Ernzerhof, The Journal of chemical physics **118**, 8207 (2003).

[18] E. B. Tadmor and R. E. Miller, *Modeling materials: continuum, atomistic and multiscale techniques* (Cambridge University Press, 2011).

[19] J. E. Lennard-Jones, Proceedings of the Physical Society (1926-1948) **43**, 461 (1931).

[20] M. S. Daw and M. I. Baskes, Physical Review B **29**, 6443 (1984).

[21] F. H. Stillinger and T. A. Weber, Physical review B **31**, 5262 (1985).

[22] J. Tersoff, Physical review B **37**, 6991 (1988).

[23] A. C. Van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, The Journal of Physical Chemistry A **105**, 9396 (2001).

[24] V. Botu, R. Batra, J. Chapman, and R. Ramprasad, The Journal of Physical Chemistry C **121**, 511 (2017).

[25] P. Auer, H. Burgsteiner, and W. Maass, Neural networks **21**, 786 (2008).

[26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, nature **323**, 533 (1986).

[27] A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington Jr, and S. Manzhos, The Journal of chemical physics **148**, 241702 (2018).

[28] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, Nature **405**, 947 (2000).

[29] T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, The Journal of chemical physics **103**, 4129 (1995).

[30] H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson, The Journal of Physical Chemistry A **102**, 4596 (1998).

[31] S. Lorenz, A. Groß, and M. Scheffler, Chemical Physics Letters **395**, 210 (2004).

[32] H. B. Curry, Quarterly of Applied Mathematics **2**, 258 (1944).

[33] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, in *International conference on machine learning* (PMLR, 2013), pp. 1139–1147.

[34] D. P. Kingma and J. Ba, arXiv preprint arXiv:1412.6980 (2014).

[35] J. Duchi, E. Hazan, and Y. Singer, Journal of machine learning research **12** (2011).

[36] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, SIAM Journal on scientific computing **16**, 1190 (1995).

[37] K. Levenberg, Quarterly of applied mathematics **2**, 164 (1944).

[38] D. Mardquardt, J. Soc. Ind. Appl. Math **11**, 431 (1963).

[39] G. V. Puskorius and L. A. Feldkamp, in *IJCNN-91-Seattle International Joint Conference on Neural Networks* (IEEE, 1991), vol. 1, pp. 771–777.

[40] N. Artrith and A. Urban, Computational Materials Science **114**, 135 (2016).

[41] M. Gastegger and P. Marquetand, Journal of chemical theory and computation **11**, 2187 (2015).

[42] K. Lee, D. Yoo, W. Jeong, and S. Han, Computer Physics Communications **242**, 95 (2019).

[43] C. E. Rasmussen, in *Summer School on Machine Learning* (Springer, 2003), pp. 63–71.

[44] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, Physical review letters **108**, 058301 (2012).

[45] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. Von Lilienfeld, K.-R. Mu?ller, and A. Tkatchenko, The journal of physical chemistry letters **6**, 2326 (2015).

[46] K. T. Schütt, H. Glawe, F. Brockherde, A. Sanna, K.-R. Müller, and E. K. Gross, Physical Review B **89**, 205118 (2014).

[47] T. Xie and J. C. Grossman, Physical review letters **120**, 145301 (2018).

[48] J. Behler, The Journal of chemical physics **134**, 074106 (2011).

[49] A. P. Bartók, R. Kondor, and G. Csányi, Physical Review B **87**, 184115 (2013).

[50] W. Li and Y. Ando, Physical Chemistry Chemical Physics **20**, 30006 (2018).

[51] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, The Journal of chemical physics **148**, 241730 (2018).

[52] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, The Journal of chemical physics **148**, 241709 (2018).

[53] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, Journal of Chemical Physics **148**, 241730 (2018).

[54] K. Pearson, The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 559 (1901).

[55] S. J. Nowlan and G. E. Hinton, Neural computation **4**, 473 (1992).

[56] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Journal of Machine Learning Research **15**, 1929 (2014).

[57] M. Wen and E. B. Tadmor, npj Computational Materials **6**, 1 (2020).

[58] W. Yang, Physical review letters **66**, 1438 (1991).

[59] W. Yang and T.-S. Lee, The Journal of chemical physics **103**, 5674 (1995).

[60] W. Kohn, Physical Review Letters **76**, 3168 (1996).

[61] E. Prodan and W. Kohn, Proceedings of the National Academy of Sciences **102**, 11635 (2005).

[62] N. Artrith, T. Morawietz, and J. Behler, Physical Review B **83**, 153101 (2011).

[63] T. Morawietz, V. Sharma, and J. Behler, The Journal of chemical physics **136**, 064103 (2012).

[64] T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, Nature communications **12**, 1 (2021).

[65] S. Goedecker, Reviews of Modern Physics **71**, 1085 (1999).

[66] A. Khorshidi and A. A. Peterson, Computer Physics Communications **207**, 310 (2016).

[67] J. Behler, Angewandte Chemie International Edition **56**, 12828 (2017).

[68] D. Yoo, K. Lee, W. Jeong, D. Lee, S. Watanabe, and S. Han, Physical Review Materials **3**, 093802 (2019).

[69] D. Sheppard, P. Xiao, W. Chemelewski, D. D. Johnson, and G. Henkelman, The Journal of chemical physics **136**, 074103 (2012).

[70] W. Jeong, K. Lee, D. Yoo, D. Lee, and S. Han, The Journal of Physical Chemistry C **122**, 22790 (2018).

[71] J. R. Boes and J. R. Kitchin, Molecular Simulation **43**, 346 (2017).

[72] Y.-C. Lu, H. A. Gasteiger, and Y. Shao-Horn, Journal of the American Chemical Society **133**, 19048 (2011).

[73] H. He and E. A. Garcia, IEEE Transactions on knowledge and data engineering **21**, 1263 (2009).

[74] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, Journal of artificial intelligence research **16**, 321 (2002).

[75] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, npj Computational Materials **6**, 1 (2020).

[76] T. G. Dietterich, in *International workshop on multiple classifier systems* (Springer, 2000), pp. 1–15.

[77] A. A. Peterson, R. Christensen, and A. Khorshidi, Physical Chemistry Chemical Physics **19**, 10978 (2017).

[78] M. Gastegger, J. Behler, and P. Marquetand, Chemical science **8**, 6924 (2017).

[79] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, The Journal of chemical physics **148**, 241733 (2018).

[80] W. Jeong, D. Yoo, K. Lee, J. Jung, and S. Han, The Journal of Physical Chemistry Letters **11**, 6090 (2020).

[81] C. Lavoie, F. dHeurle, C. Detavernier, and C. Cabral Jr, Microelectronic Engineering **70**, 144 (2003).

[82] Z. Zhang, S.-L. Zhang, B. Yang, Y. Zhu, S. M. Rossnagel, S. Gaudet, A. J. Kellock, J. Jordan-Sweet, and C. Lavoie, Applied Physics Letters **96**, 071915 (2010).

[83] F. Geenen, K. van Stiphout, A. Nanakoudis, S. Bals, A. Vantomme, J. Jordan-Sweet, C. Lavoie, and C. Detavernier, Journal of Applied Physics **123**, 075303 (2018).

[84] M. Christensen, V. Eyert, C. Freeman, E. Wimmer, A. Jain, J. Blatchford, D. Riley, and J. Shaw, Journal of Applied Physics **114**, 033533 (2013).

[85] J. Foggiato, W. S. Yoo, M. Ouaknine, T. Murakami, and T. Fukada, Materials Science and Engineering: B **114**, 56 (2004).

[86] J. Luo, Z. Qiu, C. Zha, Z. Zhang, D. Wu, J. Lu, J. Åkerman, M. Östling, L. Hultman, and S.-L. Zhang, Applied Physics Letters **96**, 031911 (2010).

[87] B. Shan and K. Cho, Physical Review B **70**, 233405 (2004).

[88] S. Park, B. Lee, S. H. Jeon, and S. Han, Current Applied Physics **11**, S337 (2011).

[89] J. Maddox, Nature **335**, 201 (1988).

[90] D. H. Wolpert and W. G. Macready, IEEE transactions on evolutionary computation **1**, 67 (1997).

[91] C. J. Pickard and R. Needs, Physical review letters **97**, 045504 (2006).

[92] C. J. Pickard and R. Needs, Journal of Physics: Condensed Matter **23**, 053201 (2011).

[93] J. Pannetier, J. Bassas-Alsina, J. Rodriguez-Carvajal, and V. Caignaert, Nature **346**, 343 (1990).

[94] Y. Wang, J. Lv, L. Zhu, and Y. Ma, Physical Review B **82**, 094116 (2010).

[95] S. T. Call, D. Y. Zubarev, and A. I. Boldyrev, Journal of computational chemistry **28**, 1177 (2007).

[96] S. Goedecker, The Journal of chemical physics **120**, 9911 (2004).

[97] D. J. Wales and J. P. Doye, The Journal of Physical Chemistry A **101**, 5111 (1997).

[98] R. Martoňák, A. Laio, and M. Parrinello, Physical review letters **90**, 075503 (2003).

[99] A. R. Oganov and C. W. Glass, The Journal of chemical physics **124**, 244704 (2006).

[100] Q. Zhu, A. R. Oganov, C. W. Glass, and H. T. Stokes, Acta Crystallographica Section B: Structural Science **68**, 215 (2012).

[101] D. Duan, X. Huang, F. Tian, D. Li, H. Yu, Y. Liu, Y. Ma, B. Liu, and T. Cui, Physical Review B **91**, 180502 (2015).

[102] A. R. Oganov, C. W. Glass, and S. Ono, Earth and Planetary Science Letters **241**, 95 (2006).

[103] A. R. Oganov, S. Ono, Y. Ma, C. W. Glass, and A. Garcia, Earth and Planetary Science Letters **273**, 38 (2008).

[104] A. R. Oganov, C. J. Pickard, Q. Zhu, and R. J. Needs, Nature Reviews Materials **4**, 331 (2019).

[105] M. Amsler, L. Ward, V. I. Hegde, M. G. Goesten, X. Yi, and C. Wolverton, Physical Review Materials **3**, 035404 (2019).

[106] Z. Shao, D. Duan, Y. Ma, H. Yu, H. Song, H. Xie, D. Li, F. Tian, B. Liu, and T. Cui, npj Computational Materials **5**, 1 (2019).

[107] G. Hautier, C. C. Fischer, A. Jain, T. Mueller, and G. Ceder, Chemistry of Materials **22**, 3762 (2010).

[108] A. O. Oliynyk, L. A. Adutwum, B. W. Rudyk, H. Pisavadia, S. Lotfi, V. Hlukhyy, J. J. Harynuk, A. Mar, and J. Brgoch, Journal of the American Chemical Society **139**, 17870 (2017).

[109] M. Rahm, R. Hoffmann, and N. Ashcroft, Journal of the American Chemical Society **139**, 8740 (2017).

[110] J. Wang, T.-N. Ye, Y. Gong, J. Wu, N. Miao, T. Tada, and H. Hosono, Nature communications **10**, 1 (2019).

[111] C. Collins, M. Dyer, M. Pitcher, G. Whitehead, M. Zanella, P. Mandal, J. Claridge, G. Darling, and M. Rosseinsky, Nature **546**, 280 (2017).

[112] G. Bergerhoff, R. Hundt, R. Sievers, and I. Brown, Journal of chemical information and computer sciences **23**, 66 (1983).

[113] R. F. Herrmann, K. Tanigaki, S. Kuroshima, and H. Suematsu, Chemical physics letters **283**, 29 (1998).

[114] R. C. Gil, W. Carrillo-Cabrera, M. Schultheiss, K. Peters, H. G. v. Schnering, and Y. Grin, Zeitschrift für anorganische und allgemeine Chemie **625**, 285 (1999).

[115] S. Ni, L. Chou, and J. Chang, Ceramics International **33**, 83 (2007).

[116] W. Weppner and R. Huggins, Physics Letters A **58**, 245 (1976).

[117] N. J. Boukharrata, J.-R. Duclere, J.-P. Laval, and P. Thomas, Acta Crystallographica Section C: Crystal Structure Communications **69**, 460 (2013).

[118] C. Hong, J. M. Choi, W. Jeong, S. Kang, S. Ju, K. Lee, J. Jung, Y. Youn, and S. Han, Physical Review B **102**, 224104 (2020).

[119] P. Richet, F. Leclerc, and L. Benoist, Geophysical Research Letters **20**, 1675 (1993).

[120] W. Weppner and R. Huggins, Journal of the Electrochemical Society **124**, 35 (1977).

[121] C. W. Glass, A. R. Oganov, and N. Hansen, Computer physics communications **175**, 713 (2006).

[122] A. R. Oganov, Y. Ma, A. O. Lyakhov, M. Valle, and C. Gatti, Reviews in Mineralogy and Geochemistry **71**, 271 (2010).

[123] P. V. Bushlanov, V. A. Blatov, and A. R. Oganov, Computer Physics Communications **236**, 1 (2019).

[124] A. O. Lyakhov, A. R. Oganov, H. T. Stokes, and Q. Zhu, Computer Physics Communications **184**, 1172 (2013).

초 록

현재 나노 스케일 소재 연구에서 활발히 사용되는 제일원리계산(First-principles calculations)은 과다한 컴퓨팅 자원을 요구하여 모델링 사이즈가 수 nm를 넘기 힘들다는 한계가 있다. 반대로 고전 원자간 퍼텐셜(Classical interatomic potential)의 경우 정해진 함수 형태를 통해서 원자 간 결합을 기술하여 계산 속도가 매우 빠르지만, 정확도가 제한적이고 복잡한 화학 결합을 기술하는 함수 형태를 개발하는 것이 매우 어렵다는 단점이 있다.

기계학습 모델을 통해 제일원리계산을 학습하여 주어진 시스템의 퍼텐셜 에너지 표면(Potential energy surface)을 기술하는, 일명 기계학습 퍼텐셜은 앞서 언급한 두 방법론의 장점을 조합하여 높은 정확도의 계산을 훨씬 적은 계산 비용으로 얻을 수 있는 방법론으로 많은 관심을 받고 있다. 그 중 특히 인공신경망 모델을 기반으로 하는 고차원 인공신경망 퍼텐셜(High-dimensional neural network potential, NNP)은 금속, 산화물, 반도체, 분자 간 반응 등 광범위한 물질계에 성공적으로 응용된 사례가 이미 보고되었으며 점차 그 응용 분야를 넓혀가는 모습이다.

하지만 여전히 연구자들 사이에 기계학습 퍼텐셜에 대한 근본적인 이해가 많이 부족한데 상황인데, 그에 따라 흔한 저지르는 오류는 기계학습 퍼텐셜을 고전적인 원자간 퍼텐셜의 일종으로 생각하여 그 둘을 비슷한 방식으로 해석하려는 것이다. 그러나 고전적인 원자간 퍼텐셜과 기계학습 퍼텐셜 간에는 더욱 근본적인 차이가 존재한다. 그것은 바로 고전적인 퍼텐셜이 원자 간의 화학 결합을 명시적인 함수 형태로 근사하는 반면 기계학습 퍼텐셜은 이러한 근사 없이 단지 학습 세트(Training set)로 주어지는 제일원리계산의 전체 에너지(Total energy)를 각 원자 에너지(Atomic energy)의 합으로 맵핑(mapping)한다는 것이다. 이러한 특수성 때문에 기계학습 퍼텐셜은 고전적 원자간 퍼텐셜과는 상당히 다른 방식으로 작동하게 되며 따라서 시뮬레이션 중 오류를 유발하는 요인 또한 새로이 이해하고 그에 맞는 해결책을 제시하여야 한다.

본 논문에서는 비교적 복잡한 시스템을 다루는 기계학습 퍼텐셜을 개발할 때

쉽게 마주할 수 있는 내재적 어려움 중 세 가지 주요한 주제를 다루고 그 문제들에 대한 해결책을 제시한다. 본 논문에서 논의하는 첫 번째 주제는 학습 데이터의 편중된 분포에 의한 불균형적인 학습의 문제이다. 이 문제는 특히 공공(Vacancy), 침입(Interstitial) 등 결함(Defect)이 포함된 원자 구조를 다룰 때 크게 두드러지는데 그것은 쉽게 말해 결함이 포함된 원자 구조를 기계학습 퍼텐셜로 학습할 경우 학습 에러가 모든 원자에 고르게 분포하지 않고 적게 샘플 되는 결함 원자에 편중된다는 것이다. 이러한 에러의 편중은 많은 경우 시뮬레이션 정확성을 크게 하락시키며 결함 에너지 (Defect energy) 등의 물리량 예측에 있어서 큰 오류를 유발한다. 우리는 훈련 포인트의 밀도를 정량화하는 가우스 밀도 함수를 새롭게 제시하고 이 함수를 통한 가중치 부여 방법론을 적용함으로써 불균형적 학습 문제를 해결할 수 있었다.

다음으로 다루는 주제는 예측 불확실성의 정량화 문제이다. 고전적인 원자 간 퍼텐셜이 물리적 기반을 가진 함수를 근간으로 하는 것에 반해 기계학습 퍼텐셜은 그러한 물리적 근간이 부족하므로 예측 결과의 높은 신뢰성을 보장하기 어렵다. 따라서 기계학습 방법론의 신뢰성 확보를 위해서는 예측 불확실성의 정량화가 필요하다. 이에 우리는 복제 인공신경망 앙상블(Replica NNP ensemble)을 통한 예측 불확실성의 정량화 방법을 제시한다. 복제 인공신경망 앙상블 방법은 인공신경망 퍼텐셜이 물질 시스템의 기저에 깔린 원자 단위의 에너지를 학습할 때 필연적으로 발생하는 불확실성을 배제함으로써 원자 단위에서 예측 불확실성을 정확하게 정량화할 수 있다는 강력한 장점을 지닌다. 우리는 이 방법론을 니켈-실리콘 고체 계면반응 모사를 위한 인공신경망 퍼텐셜 개발 과정에 적용하였고 해당 방법론의 효용성을 확인할 수 있었다.

마지막으로, 우리는 기계학습 퍼텐셜을 사용하여 화학식 이외의 어떠한 선행 정보도 없이 안정한 결정 구조를 찾을 때 발생하는 어려움에 대해 논의한다. 일반적으로 기계학습 퍼텐셜을 개발할 때는 목표로 하는 시뮬레이션 중 나타날 수 있는 국소적인 원자 구조를 예측하여 학습 세트를 구축하게 된다. 그러나 기계학습 퍼텐셜을 미지의 결정 구조 탐색에 적용한다면 구조 탐색 중 어떠한 결정 구조가 나타날지 예측할 수 없다는 문제가 발생한다. 이에 따라 우리는 액상과 비정질 구조로만 학습 세트를 구성하여 결정 구조 내에 나타날 수 있는 다양한 국지적 원자 환경을

샘플링하는 방법을 제안한다. 이 방법론을 세 가지의 3성분계(Ternary system)와 한 가지 4성분계(Quaternary system) 물질의 안정한 결정구조 탐색에 적용했을 때 비록 기계학습 퍼텐셜이 액상과 비정질 구조만을 학습하였음에도 결정 구조 예측 알고리즘 중 나타나는 수많은 다양한 결정 구조들의 에너지를 제일원리계산에 비하여 정확히 예측한다는 것을 확인하였다.

본 연구에서는 기계학습 퍼텐셜이 가진 특징적인 문제 중 세 가지 주요한 주제를 다루고 해결책을 제시하였다. 우리는 해당 논의가 기계학습 퍼텐셜의 적용 범위 확장에 기여할 것으로 기대한다.

주요어: 기계 학습 퍼텐셜, 인공 신경망 퍼텐셜, 분자 동역

학번: 2015-20862