Downloaded from orbit.dtu.dk on: May 03, 2024

DTU Library

Investigating oxygen reduction at gold-water interface with machine learning potentials

Yang, Xin

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Yang, X. (2023). *Investigating oxygen reduction at gold-water interface with machine learning potentials*. Technical University of Denmark.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

XIN YANG

Investigating oxygen reduction at gold-water interface with machine learning potentials

PhD Thesis

September 2023

Department of Energy Conversion and Storage
Technical University of Denmark

Investigating oxygen reduction at gold-water interface with machine learning potentials

Author:
Xin Yang
xinyang@dtu.dk

Supervisor: Assoc. Prof. Heine Anton Hansen
Section for Atomic Scale Materials Modelling (ASM)
Department of Energy Conversion and Storage
Technical University of Denmark
E-mail: heih@dtu.dk

Co-supervisor: Assist. Prof. Arghya Bhowmik
Section for Autonomous Materials Discovery (AMD)
Department of Energy Conversion and Storage
Technical University of Denmark
E-mail: arbh@dtu.dk

Co-supervisor: Prof. Tejs Vegge
Section for Autonomous Materials Discovery (AMD)
Department of Energy Conversion and Storage
Technical University of Denmark
E-mail: teve@dtu.dk

DTU Energy
Department of Energy Conversion and Storage
Technical University of Denmark
Anker Engelunds Vej
Building 301
2800 Kongens Lyngby
Denmark
info@energy.dtu.dk
www.energy.dtu.dk

Preface

This thesis is submitted in candidacy for a Doctor of Philosophy (PhD) degree from the Technical University of Denmark (DTU). The work has been carried out between October 2020 and September 2023 at the Section for Atomic Scale Materials Modelling (ASM) at the Department of Energy Conversion and Storage. The studies have been supervised by Associate Professor Heine Anton Hansen, Assistant Professor Arghya Bhowmik, and Professor Tejs Vegge. Part of the work was carried out during a 2-month external stay at Technical University of Berlin hosted by Professor Klaus-Robert Müller. The PhD project was funded by the Carlsberg Foundation Young Researcher Fellowship (Grant No. CF19-0304).

Kongens Lyngby, September 2023

Xin Yang

Acknowledgements

Upon the completion of this thesis, I would like to express my gratitude to a lot of people whose guidance, support, and encouragement have been instrumental in the completion of this thesis.

First and foremost, my deepest gratitude goes to my supervisors, **Heine Anton Hansen**, **Arghya Bhowmik**, and **Tejs Vegge**, for their unwavering support, invaluable guidance, constructive discussions, and endless patience throughout my PhD study. Their trust and understanding enabled me to overcome the challenges of this project; without them, this thesis would not have been possible.

I would also like to thank Dr. Klaus-Robert Müller and Dr. Michael Gastegger for hosting my research stay in Technical University of Berlin. graciously hosting my research stay at the Technical University of Berlin. My gratitude extends to Stefaan Hessmann and Jonas Lederer for their academic support and invaluable assistance during my stay in the group.

I am also grateful for the current and former colleagues at ASM and AMD. Many thanks to Martin Hoffmann Petersen, Sam Walton Norwood, Yogeshwaran Krishnan, Smobin Vincent, and Renata Sechi for their collaborative discussions and collective efforts in developing the active learning workflow. I am also grateful to Peter Bjørn Jørgensen and Jonas Busk, whose expertise in graph neural networks enriched my knowledge base. A special thank goes to Changzhi Ai, who is not only a brilliant colleague but also a cherished friend. My gratitude also extends to Shuang Han, Xueping Qin, Armando Antonio Morin Martinez, Juan Maria García Lastra, Xiaotong Zhang, Sukanya Sinha, Kai Zheng, Karina Ulvskov Frederiksen, Jinhyun Chang, Bjarke Arnskjær Hastrup, Tipaporn Patniboon, Pernille Dalsgaard Pedersen, Tina Høiberg Jensen, and countless others who contributed to my PhD journey. Working alongside such brilliant minds has truly been an honor.

Finally, my heartfelt gratitude goes to my parents. Their unwavering love, support, and understanding have been my guiding light throughout this journey.

Abstract

Fuel cell devices are considered as an ideal solution for the transition of a sustainable future, with their performance significantly influenced by catalysts that reduce the overpotential of the oxygen reduction reaction (ORR). A prerequisite for optimizing ORR catalysts is an in-depth understanding towards the reaction mechanisms in an atomistic level, which is often achieved by density functional theory (DFT) calculations. However, the expensive computational cost of DFT has significantly limited its length-scale. Machine-learned interatomic potentials (MLIPs) have emerged as powerful tools in the domain of atomistic simulations due to their exceptional computational efficiency and ab-initio level accuracy. At the heart of creating superior MLIPs for specific applications lies the imperative for high-quality data. Yet, acquiring high-quality data for vast chemical spaces remains challenging, often requiring costly ab-initio simulations. This thesis focuses on creating a robust framework to accelerate the generation of MLIPs and leverage it to gain insights into ORR mechanisms on gold surfaces.

Firstly, we designed an autonomous active learning workflow `CURATOR` for training high-fidelity graph neural network potentials for atomistic simulations. With the well-designed batch active learning algorithms, it can efficiently acquires high-quality data for optimizing model improvement during retraining. By integrating advanced neural networks with reliable uncertainty quantification techniques, `CURATOR` ensures accurate and efficient data acquisition, reducing human efforts and computational costs for MLIP construction. Additionally, it includes trustworthy and efficient uncertainty estimation techniques. By integrating different key components, this workflow is able to autonomously manage the complex tasks for generating MLIPs.

Subsequently, we investigated the ORR at confined Au(100)-water interface by using MLIPs-accelerated metadynamics. Combining MLIPs with enhanced sampling techniques allowed our simulations to achieve time-scales beyond the reach of conventional DFT. This framework vividly showcased the full ORR reaction process, pinpointing an associative ORR mechanism on Au(100) with a low reaction barrier, aligning with experimental results. This framework shed the light on modeling complex chemical reactions under complex ambient conditions.

Having verified the predictive power of the developed framework, we extended our research systems to other primary facets of gold, using a larger simulation box to better capture the reaction dynamics. Our simulations simulations revealed the notable presence of $^*H_2O_2$ on Au(110) and Au(111). This observation is in good agreement with experimental results and shed the light on optimizing the performance of gold-based ORR catalysts.

Resumé

Brændselscelleenheder betragtes som en ideel løsning for overgangen til en bæredygtig fremtid, og deres ydeevne påvirkes væsentligt af katalysatorer, der reducerer overpotentialet for iltreduktionsreaktionen (ORR). En forudsætning for at optimere ORR-katalysatorer er en dybdegående forståelse af reaktionsmekanismerne på et atomistisk niveau, hvilket ofte opnås ved beregninger baseret på densitetsfunktionsteori (DFT). Dog har DFT's høje beregningsomkostninger betydeligt begrænset dens skala. Maskinlærte interatomiske potentialer (MLIP'er) er fremkommet som kraftfulde værktøjer inden for atomistiske simuleringer på grund af deres enestående beregningseffektivitet og ab-initio nøjagtighed. Kernen i at skabe overlegne MLIP'er til specifikke anvendelser er behovet for data af høj kvalitet. Dog forbliver indhentning af data af høj kvalitet for store kemiske rum en udfordring, ofte krævende dyre ab-initio simuleringer. Denne afhandling fokuserer på at skabe et robust framework for at fremskynde genereringen af MLIP'er og udnytte det til at få indsigt i ORR-mekanismer på guldoverflader.

Først designede vi en autonom aktiv læringsarbejdsproces `CURATOR` til træning af højfidelitets grafneuralnetværkspotentialer for atomistiske simuleringer. Med de veludformede batchaktive læringsalgoritmer kan den effektivt indhente data af høj kvalitet for at optimere modelforbedring under genoplæring. Ved at integrere avancerede neurale netværk med pålidelige usikkerhedskvantificeringsteknikker sikrer `CURATOR` nøjagtig og effektiv dataindsamling, hvilket reducerer menneskelige bestræbelser og beregningsomkostninger for MLIP-konstruktion. Desuden inkluderer den pålidelige og effektive usikkerhedsestimatingsteknikker. Ved at integrere forskellige nøglekomponenter kan denne arbejdsproces autonomt håndtere de komplekse opgaver ved at generere MLIP'er.

Derefter undersøgte vi ORR ved det begrænsede Au(100)-vandinterface ved hjælp af MLIP'er-accelereret metadynamik. Kombinationen af MLIP'er med forbedrede prøvetagningsteknikker gjorde, at vores simuleringer kunne opnå tidsskalaer ud over konventionel DFT's rækkevidde. Dette framework viste tydeligt den fulde ORR-reaktionsproces, idet det identificerede en associeret ORR-mekanisme på Au(100) med en lav reaktionsbarriere, hvilket stemmer overens med eksperimentelle resultater. Dette framework kastede lys over modellering af komplekse kemiske reaktioner under komplekse omgivelsesbetingelser.

Efter at have verificeret det udviklede frameworks forudsigelseskraft udvidede vi vores forskningssystemer til andre primære facetter af guld og brugte en større simuleringsboks for bedre at fange reaktionsdynamikken. Vores simuleringer afslørede den bemærkelsesværdige tilstedeværelse af $*H_2O_2$ på Au(110) og Au(111). Denne observation stemmer godt overens med eksperimentelle resultater og kastede lys over optimering af ydeevnen for guldbaserede ORR-katalysatorer.

List of publications

Publications included in this thesis:

- **Paper I**

  Batch Active Learning at the Core: Building Robust Machine Learning Potentials for Atomistic Simulations

  Xin Yang, Changzhi Ai, Sam Walton Norwood, Martin Hoffmann Petersen,Renata Sechi, Yogeshwaran Krishnan, Smobin Vincent, Jonas Busk, François Raymond J Cornet, Ole Winther, Juan Maria García Lastra, Tejs Vegge, Heine Anton Hansen, and Arghya Bhowmik

  To be submitted

- **Paper II**

  Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces

  Xin Yang, Arghya Bhowmik, Tejs Vegge and Heine Anton Hansen

  Chemical Science, 2023, **14**, 3913-3922

- **Paper III**

  A comprehensive study of facet-dependent oxygen reduction dynamics on gold surfaces using metadynamics and graph neural networks

  Xin Yang, Arghya Bhowmik, Tejs Vegge, and Heine Anton Hansen

  To be submitted

Publications out of the scope of this thesis:

- **Paper IV**

  Oxidation and de-alloying of PtMn particle models: a computational investigation

  Thantip Roongcharoen, Xin Yang, Shuang Han, Luca Sementa, Tejs Vegge, Heine Anton Hansen and Alessandro Fortunelli

  Faraday Discussion, 2023, **242**, 174-192

- **Paper V**

  Graph neural network-accelerated multitasking genetic algorithm for optimizing PdxTi1-xHy surface under various CO2 reduction reaction conditions

  Changzhi Ai, Shuang Han, Xin Yang, Tejs Vegge and Heine Anton Hansen

  Preprint: https://doi.org/10.26434/chemrxiv-2023-0kjx4

Contents

Preface i

Acknowledgements iii

Abstract v

Resumé vii

List of publications ix

Contents xiii

1 Introduction 1

1.1 Oxygen reduction reaction at solid-liquid interface . . . 1

1.2 Revolutionizing molecular dynamics with machine learning potentials . . . 4

1.3 Active learning of machine learning interatomic potential . . . 6

1.4 Outline of thesis . . . 8

2 Theory and Methods 9

2.1 Density functional theory . . . 9

2.1.1 Schrödinger equation . . . 9

2.1.2 Born-Oppenheimer approximation . . . 10

2.1.3 Hohenberg-Kohn theorems . . . 11

2.1.4 Kohn-Sham equations . . . 11

2.1.5 Exchange-correlation functionals . . . 13

2.2 Molecular dynamics . . . 15

2.2.1 Practical MD simulation: integrator, ensemble, and thermostats . . . 15

2.2.2 Potential energy function . . . 16

2.3 Machine learning potential . . . 16

2.3.1 Descriptor-based machine learning potentials . . . 17

2.3.2 Message-passing neural networks . . . 18

2.3.3 Training machine learning potentials in practice . . . 21

2.4 Enhanced sampling technique 22
2.4.1 Collective variables 22
2.4.2 Enhanced sampling methods 23

3 Automated active learning workflow **27**
3.1 Introduction 27
3.2 Neural network potential library 29
3.2.1 Model training and evaluation 29
3.2.2 Efficient gradient calculation 30
3.3 Batch active learning 33
3.3.1 Feature engineering 34
3.3.2 Batch mode selection 39
3.4 Uncertainty-aware simulation 45
3.5 Automated workflow 50
3.6 Conclusions 52

4 Oxygen reduction at confined Au(100)-water interface **53**
4.1 Introduction 53
4.2 Computational methodology 55
4.2.1 Active learning framework 55
4.2.2 Training details 57
4.2.3 AIMD and single point DFT calculations 59
4.2.4 Production MD simulations 60
4.2.5 Metadynamics simulations 60
4.3 Results and discussions 62
4.3.1 Validation of models 62
4.3.2 Full metadynamics simulation of ORR 65
4.4 Conclusions 68

5 Facet-dependent ORR on Au surfaces **69**
5.1 Introduction 69
5.2 Computational methods 70
5.2.1 Generation of neural network potentials 70
5.2.2 DFT calculations 71
5.2.3 Production MD simulation 72
5.2.4 Metadynamics simulation 73
5.3 Results and discussion 73

5.3.1 Regular MD simulations 73
5.3.2 Metadynamics with single oxygen molecule 77
5.3.3 Metadynamics with two oxygen molecules 80
5.4 Conclusion and outlooks 83

6 Conclusions and Outlooks **85**

Bibliography **87**

Appendix A **103**

Appendix B **111**

Appendix C **123**

Paper I **137**

Paper II **182**

Paper III **193**

CHAPTER 1

Introduction

As the world is facing the pressing challenges of climate change, overexploiting natural resources, and growing energy demands, the pursuit of a sustainable future has never been more paramount. At the heart of this vision is the transition from non-renewable, pollutant-heavy energy sources to cleaner, more sustainable alternatives. Fuel cell devices can convert the chemical energy of fuels (typically hydrogen, methanol, etc.) to clean electrical energy and represent one of the pivotal technologies in the realm of clean energy solutions.[1, 2] The oxygen reduction reaction (ORR), which takes place at the cathode of fuel cells, is central to many fuel cell devices and is of significant importance in electrocatalysis.[3, 4, 5] However, the absence of an efficient and economical ORR catalyst remains a bottleneck in the large-scale commercialization of fuel cells. Consequently, the pursuit of novel and optimized ORR catalysts has consistently motivated researchers in academia and industry globally.

1.1 Oxygen reduction reaction at solid-liquid interface

A fundamental step to searching for and optimizing efficient ORR catalysts is an in-depth understanding of the underlying reaction mechanisms. Here, density functional theory (DFT) calculations have played a significant role in rationalizing the trends in ORR catalytic activity across various materials.[6, 7, 8, 9] In the ORR process, molecular oxygen undergoes electrochemical reduction with four protons and electrons to form water, producing an electrical potential capable of energizing various electronic devices. Depending on the chosen catalysts, the oxygen reduction can proceed in both two- and four-electron

oxygen reduction pathways as demonstrated in eqs. 1.1–1.12.

$$\text{Two electron} : O_2 + 2(H^+ + e^-) \rightarrow H_2O_2 \tag{1.1}$$

$$O_2 + H^+ + e^- \rightarrow {}^*OOH \tag{1.2}$$

$${}^*OOH + H^+ + e^- \rightarrow H_2O_2 \tag{1.3}$$

$$\text{Four electron} : O_2 + 4(H^+ + e^-) \rightarrow 2H_2O \tag{1.4}$$

$$\text{Associative} : O_2 + * \rightarrow {}^*O_2 \tag{1.5}$$

$$O_2 + H^+ + e^- \rightarrow {}^*OOH \tag{1.6}$$

$${}^*OOH + H^+ + e^- \rightarrow {}^*O + H_2O \tag{1.7}$$

$${}^*O + H^+ + e^- \rightarrow {}^*OH \tag{1.8}$$

$${}^*OH + H^+ + e^- \rightarrow H_2O + * \tag{1.9}$$

$$\text{Dissociative} : O_2 + 2* \rightarrow 2{}^*O \tag{1.10}$$

$$2{}^*O + 2H^+ + 2e^- \rightarrow 2{}^*OH \tag{1.11}$$

$$2{}^*OH + 2H^+ + 2e^- \rightarrow 2H_2O + 2* \tag{1.12}$$

The two-electron pathway (eqs. 1.1–1.3) results in a partial reduction, yielding hydrogen peroxide (H_2O_2) as the final product, and involving only OOH* as the reaction intermediate. In contrast, the complete reduction of oxygen to water can typically proceed in either an associative or dissociative mechanism, depending on whether the O_2 molecule splits before undergoing reduction. The associative mechanism (1.5–1.9) involves three intermediates, namely *OOH, *O, and *OH, while the dissociative mechanism involves only *O and *OH. The reaction activity of ORR can be evaluated by calculating the adsorption energies of these key reaction intermediates on catalyst surfaces via DFT calculations, where the effects of pH and electric field are modelled by computational hydrogen electrode method.[8, 9, 10, 7, 3] Moreover, the energetics of key reaction intermediates exhibit linear scaling relationships, which result in a volcano curve when plotting catalytic activity against and key adsorption energies as demonstrated in Figure 1.1. These theoretical approaches have greatly enriched our comprehension of the complex catalytic reaction mechanisms and aided in the discovery of efficient and cost-effective electrocatalysts.

However, many current studies predominantly relied on these adsorption energy calculations tend to oversimplify the operating conditions of catalysts and the intricate kinetic processes involved. For example, the effects of electrolytes are often simplified by modelling liquid water at the electrolyte–electrode interface as static water layers,[11, 12, 13, 14] implicitly representing them via dielectric continuum models,[15, 16, 17, 18] or even absolutely ignoring the effect of solvents.[19, 20, 21, 22] Furthermore, when calcu-

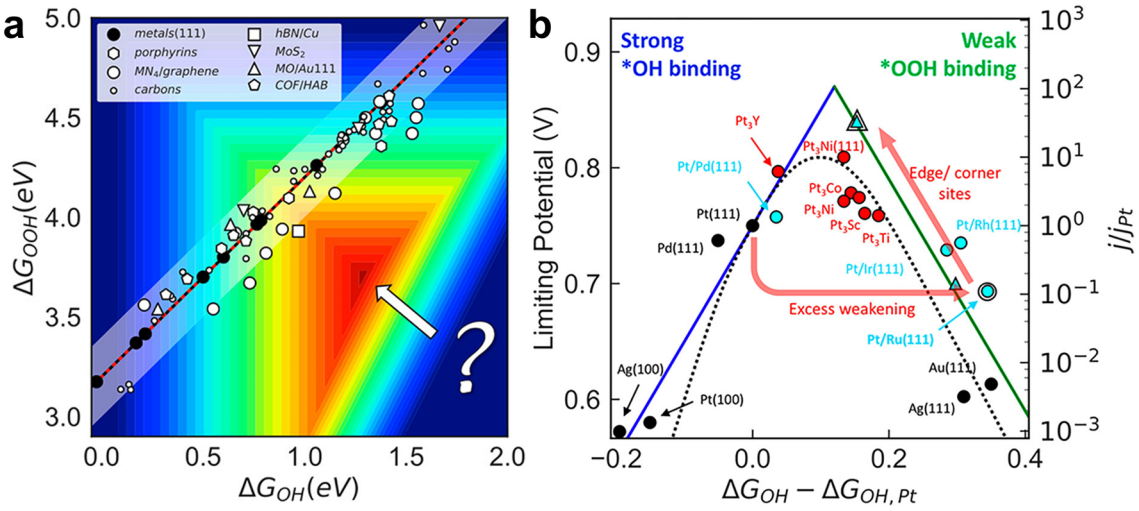

Figure 1.1: (a) Scaling relationship between ORR reaction intermediates. (b) Comparison of the limiting potential volcano plot (solid line) based on adsorption energetics and the ORR kinetic volcano (dashed line) based on microkinetic modeling. Figure reprinted from [3] with permission.

lating charge transfer barriers, traditional simulations are often conducted under constant charge conditions. This results in notable potential fluctuations throughout the elementary charge-transfer reaction steps, making it challenging to directly compare with realistic systems that operate at a consistent potential.[23, 24, 25] The absence of these crucial factors may lead to inaccurate evaluations of catalyst activity trends as compared to experiments.

A notable case is the oxygen reduction reaction taking place on gold in alkaline electrolytes.[26, 27] Using the computational hydrogen electrode method, the ORR activity of Au is predicted to be poor with the overpotential of 1.05 V vs. the reversible hydrogen electrode. This prediction contrasts sharply with the experimentally observed remarkable ORR activity of Au(100) in alkaline conditions.[27] Moreover, it is widely recognized that the ORR activity on gold surfaces are facet-dependent. Specifically, in alkaline electrolytes, the ORR on Au(100) proceeds in a full four-electron transfer mechanism. In contrast, other gold facets, such as Au(111) and Au(110), are governed by the partial two-electron transfer mechanism.[28, 29, 26] Despite the use of new techniques and persistent efforts devoted by researchers, the reason why ORR activity is exceptional and facet-dependent on gold remains elusive. The shortcomings observed in traditional DFT calculations suggest the need for a more refined approach, especially for the explicit modelling of intricate operating conditions of electrocatalysts. Such models would offer deeper insights into the ORR process on various gold facets, potentially unraveling the mysteries of their distinct behaviors and enhancing our overall understanding of electrocatalysis.

1.2 Revolutionizing molecular dynamics with machine learning potentials

By sampling the mobile solvent molecules at solid-liquid interface, ab-initio molecular dynamics (AIMD) is capable of providing direct atomistic level insights into the nature of catalytic sites and the reaction mechanisms. This method has become the gold standard for simulations of solid-liquid interfaces and has been broadly applied for studying electrocatalytic reactions.[30, 31, 32, 33, 34] There are some attempts to use AIMD in studying the ORR at solid-liquid interfaces. Cheng *et al.* utilized AIMD coupled with reactive metadynamics simulations to delve into the ORR mechanism on Pt(111), finding the predicted reaction barrier consistent with experimental results.[35] Similarly, Ikeshoji *et al.* incorporated an electric bias potential into constrained AIMD simulations, enabling a comprehensive analysis of the electrochemical ORR on Pt(111) with the derived reaction energy profiles and activation energies.[36] Furthermore, Kristoffersen *et al.* systematically investigated the liquid water–Pt(111) interface to understand catalytic reactions in fuel cells, discovering distinct hydroxyl structures and energetics in dynamic liquid environments.[37] Despite the depth of understanding achieved through these studies, they are to some extent constrained by the limited equilibration and sampling time scales of AIMD, often confined to a few to a hundred picoseconds because of its prohibitive computational cost. Furthermore, the system sizes are typically limited to of a few hundred atoms, given that the computational cost of DFT scales cubically with the number of electrons. Such constraints could potentially compromise the reliability of these findings.

In recent years, machine learning interatomic potentials (MLIPs) have emerged as a promising approach to speed up MD simulations by several orders of magnitude whilst retaining the accuracy comparable to AIMD, which enables us to considerably extend the time scale and length scale of MD simulations without compromising accuracy.[39, 38] Figure 1.2 illustrates the essential components required to create an MLIP for a specific chemical system. These include a reference database containing structures and their corresponding quantum-mechanical data (which the potential will be fitted to), a method to mathematically depict the atomic structure for the input of ML algorithms, and lastly, the regression process which can be achieved by neural networks or Gaussin process regression. MLIPs have been successfully applied for molecular dynamics across diverse chemical systems, including solid-liquid interfaces,[40, 41, 42, 43] carbon-based materials[44, 45, 46], and oxides.[47, 48] Yet, these are just glimpses of their potential applicability. In MLIPs, the total energy of chemical systems are typically decomposed into contributions from individual atoms, allowing the computational cost to scale linearly with the number of atoms,

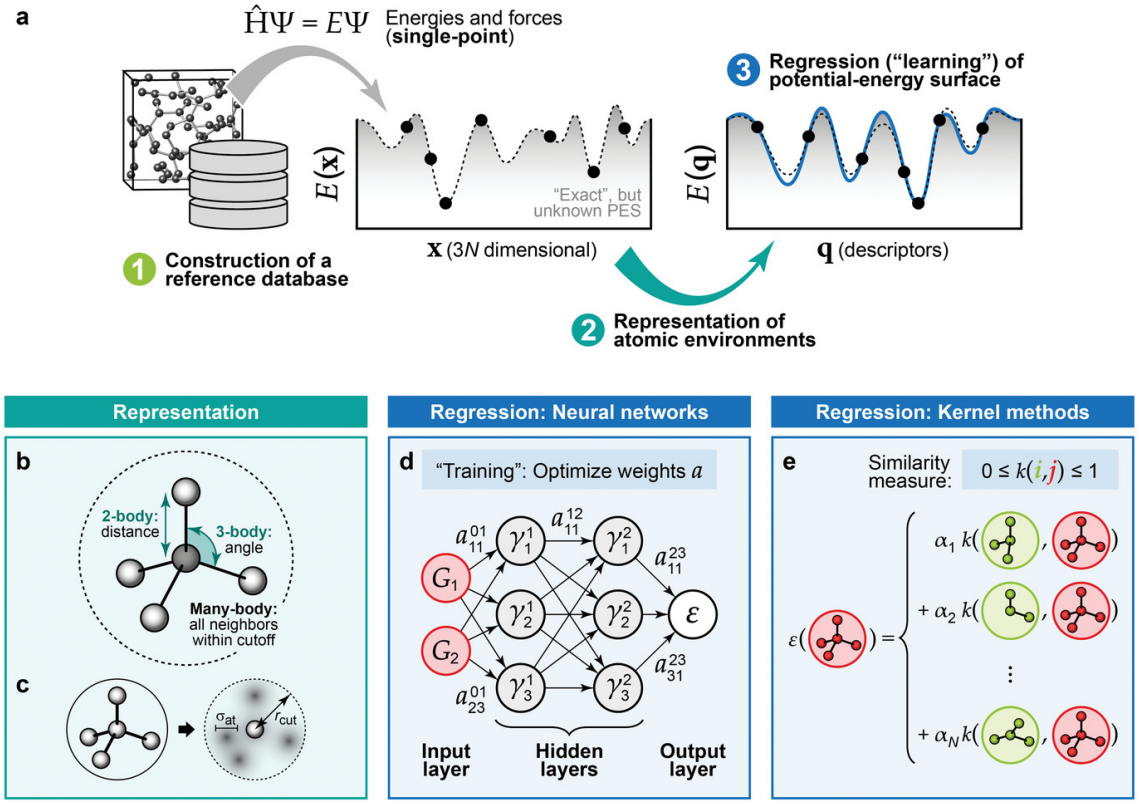

Figure 1.2: Schematic overview of how ML-based interatomic potentials are constructed. Figure reprinted from [38] with permission.

thus facilitating large-scale atomistic modeling. For MLIPs to be both data-efficient and accurate, they must uphold the invariances and equivariances inherent in physical systems, specifically spatial transformations like rotation, translation, reflection, and permutation of identical atoms. This desiderata discourages the use of simple atomic coordinates as a structural representation. Behler and Parrinello pioneered the high-dimensional neural network potential (NNP) framework,[49, 50] in which local atomic environment are represented by atom-centered symmetry functions (ACSFs) to fit atomic energies, ensuring the model remains invariant under translation and rotation. ACSFs laid the foundation for a range of MLIP designs that use predetermined rules to convert the atomic local environments into input vectors for regression. This encompasses several variants of the Behler-Parrinello neural network, including ANI,[51, 52] TensorMol,[53] and SimpleNN,[54] as well as kernel-based models like sGDML[55] and GAP.[56]

A significant drawback of these MLIPs is their reliance on extensive testing and the expertise of professionals in physics/chemistry for manually crafting features through parameter selection. The efficiency of these models is significantly determined by the choice of descriptors. Moreover, when characterizing multi-element systems, they often necessitate

a more extensive descriptor set due to the absence of atomic type specifics. To address these challenges, end-to-end NNPs have been developed, enabling direct learning of the relationship between nuclear charges, Cartesian coordinates of atoms, and atomic features entirely within the model.[57, 58]. End-to-end NNPs predominantly leveraging the message passing neural network (MPNN) architecture,[57, 58] where the atomic structures are viewed as undirected graphs with atoms as nodes and bonds as edges. Through these MPNNs, geometric information including radial distance and angles is gathered iteratively for feature refinement and then channeled into feed-forward neural networks to predict chemical properties. Examples of this approach include DTNN[59], PhysNet[60], SchNet[61, 62], and DimeNet[63]. Utilizing scalar representations, these models maintain invariance to rotations and translations. Nevertheless, some crucial chemical properties like forces and dipole moments require rotational equivariance, and solely relying on invariant features can potentially compromise model accuracy. A promising resolution is adopting advanced feature representations such as vectors and tensors, combined with rotationally equivariant message and update functions, to maintain rotational equivariance and prevent the loss of critical directional properties.[64, 65] For instance, Batzer *et al.* presented the Nequip architecture, leveraging spherical harmonics for relative position vector encoding and Clebsch-Gordon coefficients for advanced tensor products[66]. This method critically boosts model precision and data efficiency. By incorporating more than just two-body interactions, as evidenced by Batatia *et al.*[67], model performance can be further improved.

1.3 Active learning of machine learning interatomic potential

While the development of state-of-the-art machine learning models has significantly revolutionized molecular dynamics simulations, it is increasingly apparent that the cornerstone of realistic applications of these methods is high-quality data. However, acquiring data across expansive chemical spaces remains a complex and often costly process, heavily depending on expensive ab-initio simulations. Active learning stands out as the ideal solution. As illustrated in Figure 1.3, it encompasses procedures including efficiently generating high-quality data by picking representative structures from pool datasets, annotating them with DFT calculations, and feeding them back for model refinement.

Two major challenges persist in active learning for MLIPs. First, pre-trained MLIPs might misbehave in undersampled spaces, producing nonphysical structures unsuitable for labeling. It is imperative that such MLIPs be uncertainty-aware so that they can be able

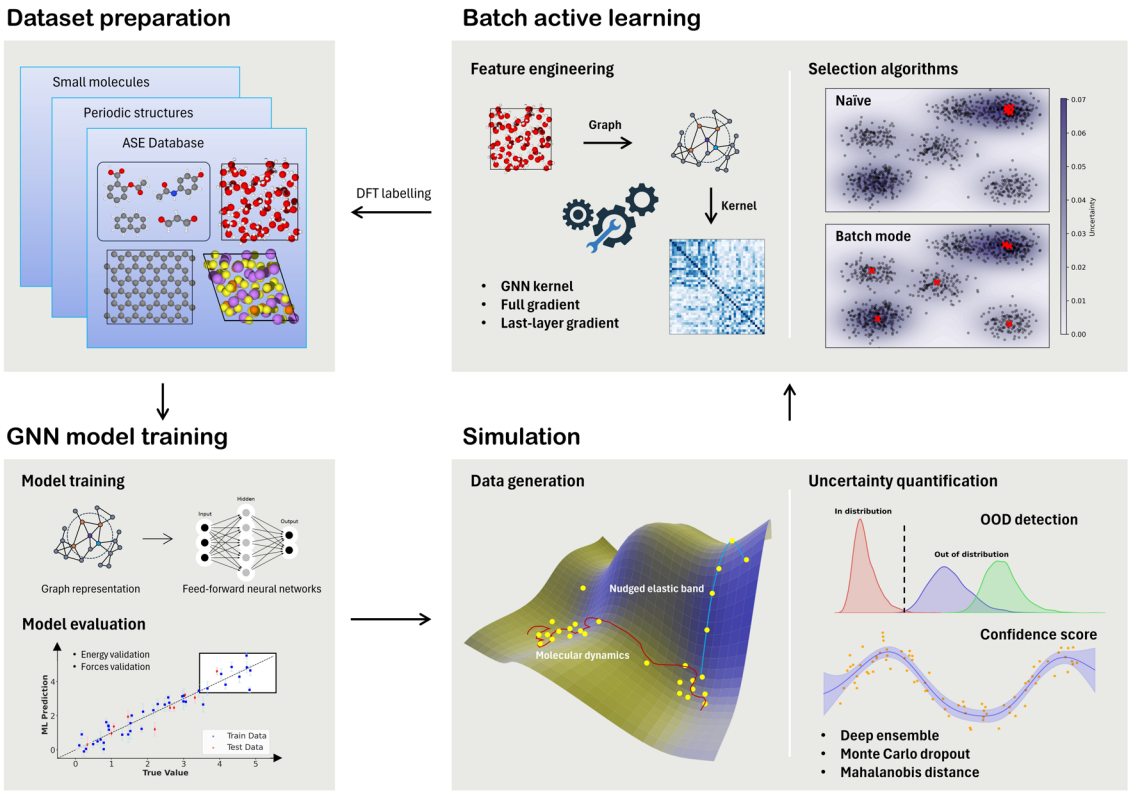

Figure 1.3: Schematic diagram of active learning procedures

to halt simulations when predictions lack confidence. Various uncertainty estimation (UE) methods have been devised to address this issue, with Gaussian process models offering inherent straightforward uncertainty estimations, and alternatives like deep ensembles and Monte Carlo dropout frequently employed where multiple predictions are made to assess model uncertainty. [68, 69, 70, 71, 72, 73, 74, 75, 76] The second challenge is the strategic selection of structures from MLIP simulations.The prevalent strategy is Query-by-Committee (QBC), where data points are assigned for labeling based on uncertainty or model disagreement.[74, 77, 78]. However, while QBC ensures single data points are valuable, it does not ensure informativeness of the entire batch. Addressing this, numerous algorithms have been devised for batch active learning that leverage uncertainties and diversities of pool data sets.[79, 80, 81, 82]

Despite advancements in models, uncertainty measures, and active learning, a seamless workflow integrating these components is yet to emerge. The data quality often depends on the efforts of materials scientists and chemists, who might not always have a background in machine learning. Bridging the expertise gap requires a user-friendly, autonomous workflow for high-quality MLIPs development. In light of these challenges, our efforts are targeted at creating an intuitive, autonomous workflow for the development of high-fidelity

MLIPs. By doing so, we aim to shape the future of MLIPs, guiding them towards a more robust and autonomous framework.

1.4 Outline of thesis

This thesis include six chapters. The remaining chapters are structured as follows:

- **Chapter 2 - Theory and Methods**
  The computational methods and relating theory used in this thesis are introduced in this chapter, including density functional theory, machine learning potential, molecular dynamics, and enhanced sampling techniques.

- **Chapter 3 - Automated active learning workflow**
  This chapter provides a detailed overview of the various components within the active learning workflow. We discuss the features employed to measure similarities among atomic structures, highlight effective batch active learning algorithms for accelerating data acquisition, and delve into the uncertainty estimation methods that ensure the reliability of simulations.

- **Chapter 4 - Oxygen reduction at confined Au(100)-water interface**
  This chapter delves into the ORR on the Au(100)-water interface, explored using MLIPs-accelerated metadynamics. We provide a detailed analysis of the active learning processes used for simulating complex reactions, and thoroughly evaluate the performance of the trained MLIPs.

- **Chapter 5 - Facet-dependent ORR on Au surfaces**
  Based on Chapter 4, we extend our research framework to more complex systems. In this chapter, we investigated the ORR dynamics on Au(100), Au(110), and Au(111) surfaces, utilizing larger simulation boxes. We provide a detailed analysis of the reaction pathways for each system and examine the impact of co-adsorbed species on the ORR dynamics.

- **Chapter 6 - Conclusions and Outlooks**
  This chapter presents the primary conclusions drawn from this thesis and offers perspectives on potential extensions and future work building upon this research.

CHAPTER 2

Theory and Methods

This chapter introduces various computational theories and methods in the thesis, including density functional theory, molecular dynamics, machine learning potential, and enhanced sampling techniques.

2.1 Density functional theory

Density Functional Theory (DFT) is a quantum mechanical modeling method used in physics and chemistry to investigate the electronic structure of many-body systems. It is based on the principles of quantum mechanics and is used to analyze the properties of electronic systems. This section predominantly draws upon a selection of distinguished textbooks that provide comprehensive introductions to the fundamental theories and prospective applications of DFT[83, 84, 85]. These works offer extensive insights and are highly recommended for readers seeking a deeper understanding and more detailed information on the theory.

2.1.1 Schrödinger equation

The foundation of DFT is the time-independent Schrödinger equation,[85] which is represented as:

$$\hat{H}\Psi(\vec{r},\vec{R}) = E\Psi(\vec{r},\vec{R}) \tag{2.1}$$

where $\hat{H}$ is the Hamiltonian operator, representing the total energy operator of the system, $\Psi(\vec{r},\vec{R})$ is the wavefunction of the system, providing the probabilities of finding particles in various locations, and E is the total energy eigenvalue of the system, with $\vec{r}$ and $\vec{R}$ being the sets of electronic and nuclear coordinates, respectively.

The Hamiltonian operator, $\hat{H}$, in the Schrödinger equation is composed of kinetic and potential energy operators for both the electrons and the nuclei in the system. It can be

expressed as:

$$\hat{H} = \underbrace{-\sum_{I=1}^{N} \frac{\hbar^2}{2M_I} \nabla_I^2}_{\hat{T}_n} + \underbrace{-\sum_{i=1}^{n} \frac{\hbar^2}{2m_e} \nabla_i^2}_{\hat{T}_e} + \underbrace{\sum_{I=1}^{N} \sum_{J>I}^{N} \frac{Z_I Z_J e^2}{4\pi\varepsilon_0 |R_I - R_J|}}_{\hat{V}_{nn}} + \underbrace{\sum_{i=1}^{n} \sum_{j>i}^{n} \frac{e^2}{4\pi\varepsilon_0 |r_i - r_j|}}_{\hat{V}_{ee}} + \underbrace{-\sum_{I=1}^{N} \sum_{i=1}^{n} \frac{Z_I e^2}{4\pi\varepsilon_0 |R_I - r_i|}}_{\hat{V}_{ne}} \tag{2.2}$$

In this representation, $\hat{T}_n$ and $\hat{T}_e$ denote the kinetic energy operators for the nuclei and the electrons, respectively, serving to quantify the kinetic energies associated with their respective motions. Similarly, $\hat{V}_{nn}$ and$\hat{V}_{ee}$ are symbolize the potential energy operators responsible for nuclear-nuclear and electron-electron repulsions, respectively, indicating the energies due to the interactions between like charges. Lastly, $\hat{V}_{ne}$ represents the potential energy operator for nuclear-electron attraction, quantifying the interaction energy stemming from the electrostatic attraction between the positively charged nuclei and the negatively charged electrons within the system.

2.1.2 Born-Oppenheimer approximation

The Schrödinger equation, in its full form, describes the behavior of both electrons and nuclei in a molecule. However, due to the vast difference in their masses, electrons move much more rapidly than nuclei. The Born-Oppenheimer approximation emphasizes on this disparity by treating the nuclei as essentially stationary during electronic motion.[86] This simplification allows for the separation of the total molecular wavefunction into electronic and nuclear components, making the equation more manageable.

$$\Psi(\vec{r}, \vec{R}) \approx \psi(\vec{r}; \vec{R}) \chi(\vec{R}) \tag{2.3}$$

where $\psi(\vec{r}; \vec{R})$ is the electronic wavefunction, $\chi(\vec{R})$ is the nuclear wavefunction, and $\vec{r}$ and $\vec{R}$ are the coordinates of electrons and nuclei, respectively. With this decomposition, the electronic energy for a given nuclear configuration $\vec{R}$ can be expressed as:

$$E(\vec{R}) = \langle \psi(\vec{r}; \vec{R}) | \hat{H} | \psi(\vec{r}; \vec{R}) \rangle \tag{2.4}$$

In essence, the Born-Oppenheimer approximation decouples the electronic and nuclear degrees of freedom. This separation is pivotal for DFT and the Hohenberg-Kohn theorems, as it allows for the focus to be primarily on the electronic structure, which predominantly determines many chemical properties. By reducing the problem to only the electronic degrees of freedom, the Hohenberg-Kohn theorems can then establish the foundation for DFT, where the electron density alone is sufficient to determine the properties of chemical systems.

2.1.3 Hohenberg-Kohn theorems

The Schrödinger equation becomes a many-body problem when dealing with systems containing more than one particle (e.g., electrons and nuclei in atoms and molecules). The interactions between every pair of particles need to be considered, leading to a combinatorial explosion in complexity as the number of particles increases. The Hohenberg-Kohn theorems lay the theoretical groundwork for reducing the complexity of many-electron problems.[87] They allows the use of electron density as the basic variable to describe the many-electron problem, rather than wavefunctions, simplifying the computational effort needed to solve the Schrödinger equation for many-electron systems. The Hohenberg-Kohn theorems stated as follows:

Theorem 1: *The external potential, and hence the energy, of a system is uniquely determined by the electron density.*

Theorem 2: *The exact ground-state density gives the minimum total energy of a system and can be obtained variationally.*

The first Hohenberg-Kohn theorem states that the ground state properties of a many-electron system are uniquely determined by the ground state electron density, $n(\vec{r})$. This implies that the electron density contains all the information needed to calculate the ground state properties of the system, including the total energy.

$$E_0 = \min_{n \to \Psi_0} \left\{ E[n] + \int V_{\text{ext}}(\vec{r}) n(\vec{r}) d\vec{r} \right\} \tag{2.5}$$

where E_0 is the ground state energy, $n(\vec{r})$ is the ground state electron density, and $V_{\text{ext}}(\vec{r})$ is the external potential.

The second Hohenberg-Kohn theorem provides the variational principle for the energy functional, stating that the true ground state electron density minimizes the energy functional, and any other density will lead to a higher energy. Mathematically, if $n(\vec{r})$ is the true ground state density, then for any $n'(\vec{r})$ that is different from $n(\vec{r})$:

$$E[n'(\vec{r})] > E[n(\vec{r})] \tag{2.6}$$

The simplification in Hohenberg-Kohn Theorems is crucial for making the study of the electronic structure of complex systems computationally tractable, enabling the practical application of quantum mechanical principles to study a wide range of materials and molecules.

2.1.4 Kohn-Sham equations

While the Hohenberg-Kohn theorems lay the theoretical foundation for expressing the energy of a many-electron system in terms of its electron density, they do not offer a

practical method to solve for this density. The Kohn-Sham equations address this by introducing a system of non-interacting electrons that generate the same ground state electron density as the real interacting system.[88] For a system of non-interacting electrons with the same ground state electron density as the real system, the Kohn-Sham equations are given by:

$$\left[-\frac{\hbar^2}{2m_e}\nabla^2 + V_{\text{eff}}(\vec{r})\right]\psi_i(\vec{r}) = \varepsilon_i\psi_i(\vec{r}) \tag{2.7}$$

where $\psi_i(\vec{r})$ is the Kohn-Sham orbital for the i-th electron, ε_i is the eigenvalue associated with the i-th Kohn-Sham orbital, with the highest occupied eigenvalue being the Fermi energy, and $V_{\text{eff}}(\vec{r})$ is the effective potential experienced by the electrons.

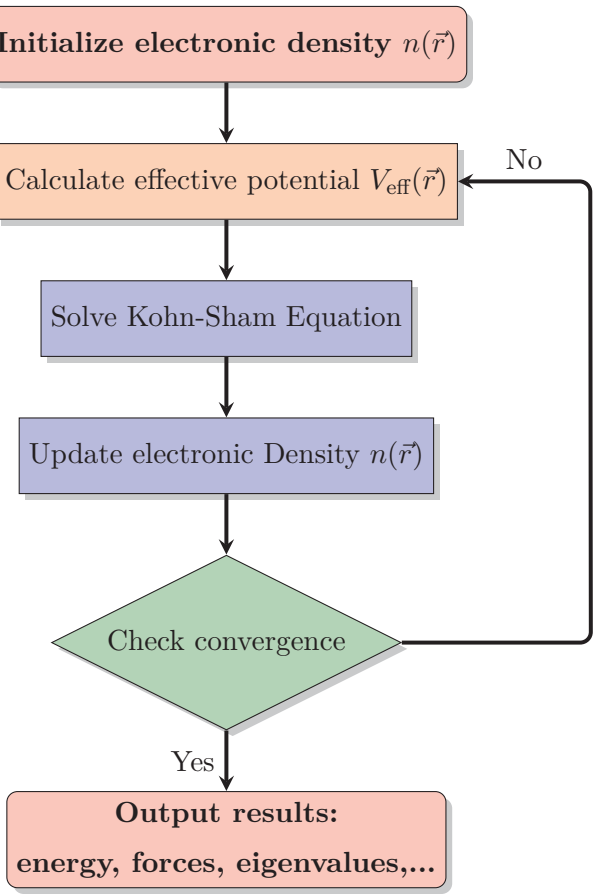

Figure 2.1: Iterative process for solving Kohn-Sham equations

In the framework of Kohn-Sham equations, the total energy functional can be decomposed into several terms, kinetic energy of non-interacting electrons, classical electron-electron interaction energy (Hartree term), external potential energy, and the exchange-correlation energy, which can be written as follows:

$$E[n] = T_s[n] + E_{\text{Hartree}}[n] + E_{\text{xc}}[n] + E_{\text{ext}}[n] \tag{2.8}$$

where n is the electronic density. The computation of the total energy functional is

crucial to construct the effective potential $V_{\text{eff}}(\vec{r})$ in the Kohn-Sham equations, which can be expressed as a sum of three potentials:

$$\begin{aligned} V_{\text{eff}}(\vec{r}) &= V_{\text{ext}}(\vec{r}) + V_{\text{Hartree}}(\vec{r}) + V_{\text{xc}}(\vec{r}) \\ &= V_{\text{ext}}(\vec{r}) + \int \frac{n(\vec{r}')}{|\vec{r} - \vec{r}'|} d\vec{r}' + \frac{\delta E_{\text{xc}}[n]}{\delta n(\vec{r})} \end{aligned} \tag{2.9}$$

In this representation, external potential $V_{\text{ext}}(\vec{r})$ represents the potential due to the external (usually nuclear) charge distribution. It is the classical electrostatic interaction between the electrons and the nuclei. Hartree potential $V_{\text{Hartree}}(\vec{r})$ represents the classical electrostatic repulsion between electrons. It is given by the convolution of the electron density $n(\vec{r})$ with the Coulomb kernel. The last part, exchange-correlation potential $V_{\text{xc}}(\vec{r})$, represents the quantum mechanical effects of electron exchange and correlation. It is the functional derivative of the exchange-correlation energy $E_{xc}[n]$ with respect to the electron density.

With the constructed $V_{\text{eff}}(\vec{r})$ and predefined basis sets, we can solve the Kohn-Sham equations to obtain the Kohn-Sham orbitals. And the electronic density $n(\vec{r})$ can be updated with the occupied Kohn-Sham orbitals, which can be mathematically expressed as:

$$n(\vec{r}) = \sum_{i}^{occ} |\psi_i(\vec{r})|^2 \tag{2.10}$$

As demonstrated in Figure 2.1, solving the Kohn-Sham equations involves self-consistently determining the effective potential, the Kohn-Sham orbitals, and the electron density until convergence is achieved. The converged electronic density, Kohn-Sham orbitals, and eigenvalues then can then be used to calculate the properties of interest.

2.1.5 Exchange-correlation functionals

The exchange-correlation functional is a pivotal component in DFT, representing the quantum mechanical effects of electron exchange and correlation within the Kohn-Sham framework. It is the only missing map in the Kohn-Sham equations and is crucial for accurately describing electron-electron interactions. Different approximations to this functional lead to different levels of accuracy and computational expense.

Local density approximation (LDA) is the simplest approximation and is based on the uniform electron gas model.[88, 89] It considers only the local electron density at each point in space, making it computationally efficient but often less accurate for systems with rapidly varying densities. The exchange-correlation energy functional per particle $\varepsilon_{xc}^{LDA}(n)$ in LDA is solely a function of the local electron density $n(\vec{r})$.

$$E_{\text{xc}}^{\text{LDA}}[n] = \int n(\vec{r}) \varepsilon_{\text{xc}}^{\text{LDA}}(n(\vec{r})) d\vec{r} \tag{2.11}$$

The exchange energy part in LDA is usually given by:

$$E_{\mathrm{x}}^{\mathrm{LDA}}[n] = -\frac{3}{4}\left(\frac{3}{\pi}\right)^{1/3}\int n(\vec{r})^{4/3}d\vec{r} \tag{2.12}$$

While the correlation part $E_{\mathrm{c}}^{\mathrm{LDA}}$ can be obtained from parametrization of quantum Monte Carlo calculations of the uniform electron gas, and it is usually represented in terms of the electron density $n(\vec{r})$ using specific parametric forms or interpolation schemes.

Generalized gradient approximation (GGA) is a more sophisticated approximation that includes the gradient of the electron density in addition to the local density itself.[90] This allows GGA to account for spatial variations in the electron density, improving the accuracy for systems with non-uniform densities. The exchange-correlation energy in GGA is given by a functional that depends on both the local electron density $n(\vec{r})$ and its gradient $\nabla n(\vec{r})$.

$$E_{\mathrm{xc}}^{\mathrm{GGA}}[n] = \int f(n(\vec{r}), \nabla n(\vec{r}))d\vec{r} \tag{2.13}$$

The choice of the exchange-correlation functional is crucial as it significantly impacts the accuracy and reliability of DFT calculations. While LDA is computationally less demanding and suitable for simple systems with uniform electron densities, GGA provides improved accuracy for a wider range of systems, especially those with varying electron densities, at the cost of increased computational effort. Classical GGA functionals such as Perdew-Burke-Ernzerhof (PBE),[90] Becke-Perdew 86 (BP86),[91] Becke-Lee-Yang-Parr (BLYP),[92, 91] and Perdew-Wang 91 (PW91)[93] are widely recognized and employed due to their diverse applicability and reliable performance across various systems and properties. PBE is renowned for its versatility, BP86 is notable for predicting molecular geometries and thermochemistry, BLYP is favored in molecular system studies, and PW91 is valued for its extensive applicability. The choice of a functional necessitates a balanced consideration of its accuracy, transferability, computational demand, and its aptness for the specific research focus, ensuring the relevance and dependability of the DFT studies conducted.

In the scope of this thesis, all DFT calculations are carried out using the Vienna Ab initio Simulation Package (VASP)[94, 95, 96, 97] and the Atomic Simulation Environment (ASE)[98]. The effects of exchange and correlation are approximated by using the PBE functional with D3 van der Waals correlation.[90, 99] The wave functions are expanded in a plane waves basis set using the projector augmented wave (PAW) method.[100]

2.2 Molecular dynamics

Molecular Dynamics (MD) is typically used to study the time-dependent behavior of atomic and molecular systems. By simulating the motion of atoms and molecules, MD provides atomistic insights into the dynamic evolution of systems, revealing detailed information about molecular vibrations, conformational changes, and interactions between molecules, which are crucial for understanding various physical, chemical, and biological phenomena.[101]

MD simulations are grounded in classical mechanics, primarily utilizing Newton's second law of motion to describe the motion of particles in a system:

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \vec{F}_i \tag{2.14}$$

where m_i is the mass, $\vec{r}_i$ is the atomic position, and $\vec{F}_i$ is the force of i^{th} particle. The force is derived from the gradient of the potential energy U of the system with respect to the particle's position:

$$\vec{F}_i = -\nabla_i U(\vec{r}_1, \vec{r}_2, ..., \vec{r}_N) \tag{2.15}$$

2.2.1 Practical MD simulation: integrator, ensemble, and thermostats

In a practical MD simulation, the system is firstly initialized with a set of atomic coordinates, and assigned initial velocities typically drawn from a Maxwell-Boltzmann distribution at a specified temperature. Then the forces acting on each atom are computed using a potential energy function. The equations of motion are integrated over discrete time steps using a numerical integrator like the Verlet algorithm,[102, 103] where the positions and velocities of the atoms are updated using the following equation:

$$\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \frac{\vec{F}_i}{m_i}\Delta t^2 \tag{2.16}$$

However, the Verlet algorithm typically operates in the Microcanonical (NVE) ensemble, where the number of particles (N), volume (V), and energy (E) are conserved, leading to potential fluctuations in temperature. To perform simulations in the Canonical (NVT) ensemble or the Isothermal–Isobaric (NPT) ensemble, where the temperature or both temperature and pressure are constrained, thermostats are essential. Thermostats, like the Langevin and Nosé-Hoover, adjust the velocities of particles to regulate the system's temperature, ensuring the kinetic energy corresponds to the desired temperature. The Langevin thermostat models the interaction of the system with an external heat bath using a stochastic approach, introducing both damping and random forces, and is particularly useful for simulating systems in contact with a solvent.[104, 105] In contrast, The Nosé-Hoover thermostat is deterministic and introduces an additional degree of freedom to the

system to control the temperature, making it suitable for simulating isolated systems.[106, 107]

2.2.2 Potential energy function

The potential energy is crucial in MD simulations as it defines the interactions between atoms in the system and solely determines the forces acting on them. The quality of a MD calculation depends largely on the employed The potential energy function, which can be derived in multiple ways. In classical MD, empirical or semi-empirical force fields are typically used to describe the potential energy of the system. A common form of the potential energy function U quantifies both bonded (e.g., covalent bonds) and non-bonded (e.g., van der Waals and electrostatic) interactions between atoms. In comparison, in ab-initio MD (AIMD), the potential energy and the forces are computed by solving Kohn-Sham equations as illustrated in Figure 2.1.[108] AIMD allows for the study of systems and phenomena where classical force fields are not applicable or accurate enough, providing insights into electronic structure and enabling the simulation of chemical reactions with the ability to account for electronic polarization effect. Although Ab Initio Molecular Dynamics (AIMD) is precise, its high computational cost limits its application to large-scale and long-scale simulations. Typically, AIMD studies are restricted to several tens of picoseconds, often insufficient to equilibrate the system and sample properties accurately, compromising the ergodicity and reliability of the simulations.

2.3 Machine learning potential

Machine Learning Interatomic Potentials (MLIPs) represent a novel approach in the field of molecular dynamics, aiming to bridge the accuracy of quantum mechanical calculations and the efficiency of classical potential models. MLIPs leverage machine learning algorithms to learn the relationship between atomic configurations and their corresponding potential energy, forces, and other properties directly from quantum chemistry data. This approach allows for the construction of highly accurate potential energy surfaces, enabling the simulation of large systems and long timescales with a precision that is comparable to ab-initio methods but with significantly reduced computational costs. MLIPs have been increasingly utilized to study a wide range of materials and molecular systems, showing great promise in advancing our understanding of complex molecular phenomena and material properties.

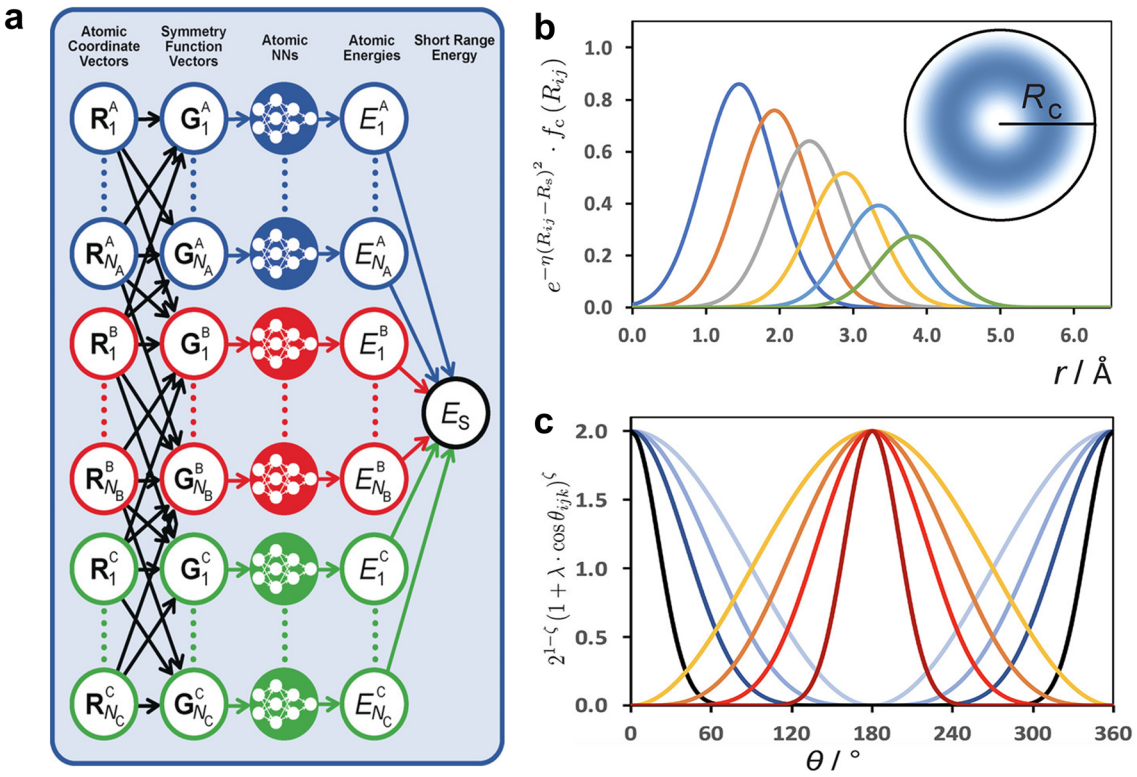

Figure 2.2: (a) Architechture of HDNNPs for a ternary system. (b) Radial ACSFs with varied R_S parameters and fixed η as a function of interatomic distance. (c) Angular ACSFs with λ=1 (blue to black lines) and λ=1 (orange to brown lines) and varied ζ as a function of θ_{ijk}. Figure taken from [42, 109] and modified with permission.

2.3.1 Descriptor-based machine learning potentials

In the initial stage, machine learning potentials predominantly relied on the handcrafted descriptors to characterize atomic environments. Behler and Parrinello[49, 50] firstly introduced the high-dimensional neural network potential (HDNNP) in which the local atomic environments are described by atom-centered symmetry functions (ACSFs). As illustrated in Figure 2.2a, the HDNNP architecture takes the geometrical information of individual atoms within a chemical system and translates it into ACSFs. These ACSFs are subsequently processed through a feed-forward neural network to derive atomic energies. ACSFs typically consist of radial and angular symmetry functions, which can be expressed as:

$$G_i^{rad} = \sum_{j \neq i} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}) \tag{2.17}$$

$$G_i^{ang} = 2^{1-\zeta} \sum_{j \neq i} \sum_{k \neq i,j} (1 + \lambda \cos\theta_{ijk})^{\zeta} e^{-\eta(R_{ij}^2+R_{ik}^2+R_{jk}^2)} f_c(R_{ij}) f_c(R_{ik}) f_c(R_{jk}) \tag{2.18}$$

where j and k are the neighboring atoms of the central atom i, r_{ij}, r_{ik} and r_{jk} are the pairwise interatomic distances, θ_{ijk} is the angle between atom i,j, and k, and the cutoff

function f_c of pairwise distance is defined as

$$f_c(R) = \begin{cases} 0.5 \cdot \left(1 + \cos\left(\pi \frac{R}{R_c}\right)\right) & \text{if } R < R_c \\ 0 & \text{if } R \geq R_c \end{cases} \tag{2.19}$$

where R_c is the cutoff radius to limit the considered atomic interactions within a certain distance. The cutoff function should be smooth and differentiable to avoid discontinuities in the potential energy surface, which is crucial for computing forces. As depicted in Figure 2.2 b and c, the atomic environments are characterized through a combination of symmetry function values, determined by the hyperparameters η, λ, ζ, and R_S. Selecting the optimal set of these hyperparameters is often nontrivial; it demands rigorous testing and deep physical/chemical insights from experts. Their choice is critical to the accuracy of trained models. Besides, to discern between various bond types within the system, distinct parameters should be allocated for each bond type. This can necessitate a large set of descriptors to accurately describe multi-element systems, leading to substantial computational overhead and compromising the performance of the models for complex chemical systems.

Despite of these limitations, the innovative methodology developed by Behler and Parrinello has paved the way for the development of more sophisticated and generalized machine learning potentials, lead to the emergence of numerous descriptor-based machine learning potential designs.

2.3.2 Message-passing neural networks

In the pursuit of overcoming challenges associated with traditional approaches, end-to-end neural network potentials have come to the forefront. These advanced models are adept at learning the mapping from nuclear charges and Cartesian coordinates of atomic structures directly to atomic features, all within the model itself. The inspiration for most of these end-to-end NNPs can be traced back to the architecture of graph neural networks (GNN),[57] with a particular emphasis on message-passing neural networks (MPNNs).[58]

In the realm of MPNNs, atomic structures are interpreted as undirected graphs, with atoms represented as nodes and atomic bonds as the connecting edges between them. The model operates by gathering geometric information, such as radial distances and angles, from neighboring nodes within a defined cutoff radius. This information is processed through a message layer, which computes the features of a specific atom. Subsequently, these features undergo refinement in an update layer. This iterative message-passing mechanism continually refines the node features, which are then channeled into a feed-forward neural network to predict the desired properties of the chemical systems.

Prominent models utilizing this approach include DTNN,[59] PhysNet,[60] SchNet,[61, 62] and DimeNet,[63] among others. These models operate on interatomic distances and employ scalar feature representations, ensuring the invariance of model output and atomic features to rotations and translations. This can be described by the following equation:

$$f(x) = f(D_X[g]x) \tag{2.20}$$

where $D_X[g]$ is the transformation in the input space.[64, 66] In the context of atomistic simulations, $D[g]$ corresponds to the group of rotations, reflections, and translations in the 3D space. However, it is crucial to note that many vital chemical properties, such as forces and dipole moments, are equivariant to rotations of the chemical systems, which fulfills

$$D_Y[g]f(x) = f(D_X[g]x) \tag{2.21}$$

where $D_Y[g]$ is the transformation operations in the output space. Relying solely on rotationally-invariant features could lead to the loss of information on these directional properties, potentially compromising the performance of the model. To mitigate this, advanced feature representations like vectors and tensors are employed, coupled with rotationally equivariant message and update functions. Among multiple state-of-the-art equivariant MPNNs, the *polarizable atom interaction neural network* [110] (PaiNN) architecture appears to be a suitable model for driving molecular dynamics as it provides outstanding model accuracy especially on force predictions and exhibits much faster inference speed in comparison with other GNN models.[67, 66] Figure 2.3 demonstrates the architecture of PaiNN with the detailed descriptions for message and update function blocks. The atomic features are comprised of two components: the invariant representation s_i and the equivariant representation $\vec{v}_i$. While the invariant representation ensures rotational invariance in predictions by emphasizing rotationally invariant inputs, the equivariant representation captures essential directional information essential for spatial and relational precision in predictions.

PaiNN receives nuclear charges $Z_i \in \mathbb{N}$ and positions $r_i \in \mathbb{R}^3$ for each atom i as inputs. The invariant atom representations are initialized to learned embeddings of the atom type, denoted as $s_i^0 = aZ_i \in \mathbb{R}^{F\times 1}$, where F represents the number of features, which is a predefined hyperparameter and kept constant throughout the network. The equivariant representations are initially set to $\vec{v}_i^0 = \vec{0} \in \mathbb{R}^{F\times 3}$, given that there is no directional information available initially. The architecture employs a residual structure of interchanging message and update blocks, which results in coupled scalar and vectorial representations. For the scalar message function, PaiNN adopts continuous-filter convolutions, represented

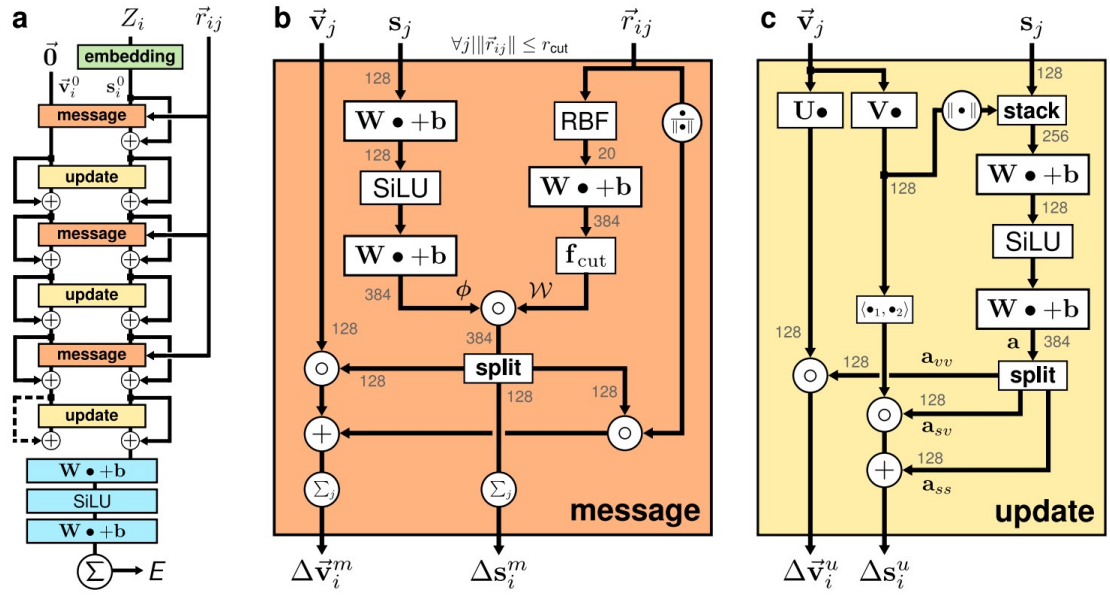

Figure 2.3: (a) The full architecture of PaiNN. (b) The message function block. (c) The update function block. Figure reprinted from [110] with permission.

as:

$$\Delta s_i^m = (\phi_s(s) * W_s)_i = \sum_j \phi_s(s_j) \circ W_s(\|\vec{r}_{ij}\|) \tag{2.22}$$

Here, φ consists of atomwise layers, and the rotationally-invariant filters W_s are linear combinations of radial basis functions. The equivariant message function also utilizes continuous-filter convolutions, formulated as:

$$\Delta v_i^m = \sum_j v_j \circ \phi_{vv}(s_j) \circ W_{vv}(\|\vec{r}_{ij}\|) + \sum_j \phi_{vs}(s_j) \circ W'_{vs}(\|\vec{r}_{ij}\|) \frac{\vec{r}_{ij}}{\|\vec{r}_{ij}\|} \tag{2.23}$$

After the feature-wise message blocks, the update blocks are applied atomwise across features. The scalar update function has a residual given by:

$$\Delta s_i^u = a_{ss}(s_i, \|\vec{v}_i\|) + a_{sv}(s_i, \|\vec{v}_i\|)\langle \vec{u}_i, \vec{v}_i \rangle \tag{2.24}$$

Here, a scaling function computed by a shared network $a(s_i, \| \vec{v}_i \|)$ is used as nonlinearity. The norm of a linear combination of features is also used to obtain the scaling, thus, coupling the scalar representations with contracted equivariant features. In a second term, the scalar product of two linear combinations of equivariant features is used. The residual for the equivariant features is defined as:

$$\Delta \vec{v}_i^u = a_{vv}(s_i, \|\vec{v}_i\|)\vec{u}_i \tag{2.25}$$

This is a nonlinear scaling of linearly combined equivariant features. The rotationally equivariant message passing and computational efficiency of PaiNN make it a powerful tool for accurate and fast predictions in MD simulations.

2.3.3 Training machine learning potentials in practice

In the preceding discussions, we have meticulously introduced the intricate architectures of multiple MLIPs, albeit without delving deeply into practical training of these MLIPs. The training of MLIPs typically involves the meticulous crafting of loss functions, the preparation of dataset, extensive tests of hyperparameters, and comprehensive model evaluation to ensure the robustness and reliability of the models.

In the sections above, we elucidated how atomic energies within the specified chemical systems can be derived from diverse designs of MLIPs. Summing up these atomic energies gives the total energy of a model system, which can be written as:

$$E = \sum_{i \in N} E_i \tag{2.26}$$

Concurrently, atomic forces are generally derived from the negative gradients of the atomic energy with respect to atomic coordinates:

$$\vec{F_i} = -\nabla_i E \tag{2.27}$$

This methodology adheres to the principle of energy conservation, a pivotal constraint for enhancing the stability of Molecular Dynamics (MD) simulations.[111] However, some models opt for a direct approach, predicting forces as part of their outputs instead of deriving them as gradients of the total potential energy.[112] This implementation could possibly lead to collapse of MD simulations over long time.

After obtaining the energy and force predictions, the model can be trained using a loss function based on a weighted sum of energy and a force loss terms:

$$\mathcal{L} = \frac{1-\lambda}{N} \sum_{i=1}^{N} (E_i - \hat{E}_i)^2 + \frac{1-\lambda}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k=1}^{3} (F_i^{jk} - \hat{F_i^{jk}})^2 \tag{2.28}$$

where λ represents the trade off between force loss and energy loss.

In this thesis, all simulations using MLIPs employed the PaiNN architecture. All the training and simulations with MLIPs were done using the `CURATOR` code, developed during this Ph.D. project. More details about training and evaluation are in the respective chapters of this thesis.

It is important to note that the foundation for the successful deployment of MLIPs is laid by high-quality ab-initio datasets. In order to collect high-value atomic structures for model training in an effective manner, we designed several batch active learning algorithms for atomistic data and built a fully automated workflow to reduce the time and computational cost for training MLIPs for chemical systems of interest. A more in-depth exploration of the workflow is available in chapter 3 of this thesis.

2.4 Enhanced sampling technique

Enhanced sampling methods are advanced simulation techniques designed to explore the configurational space of molecular systems more efficiently than conventional MD simulations, especially for systems with rugged energy landscapes.[113, 114, 115] It is particularly useful for studying systems where the timescale of the events of interest is much longer than what can be feasibly reached with standard MD simulations. These methods are designed to overcome the limitations of traditional MD by facilitating the system to cross energy barriers and escape from local minima, enabling the exploration of a wider range of configurational space and providing insights into rare events and transitions between metastable states.

2.4.1 Collective variables

Collective variables (CVs) are typically macroscopic observables that describe the relevant degrees of freedom of the chemical system, which can be represented as functions of the atomic positions: $s = s(\vec{r})$. They are designed to capture the essential features of the system's configuration and are used to probe the free energy landscape of molecular systems. In enhanced sampling methods, CVs are manipulated by adding a bias potential $V(s)$, to drive the system through different states, enabling the exploration of regions in configurational space that are not readily accessible by conventional MD simulations. The potential energy function of the system can then be expressed as:

$$U_{biased}(\vec{r}) = U(\vec{r}) + V(s(\vec{r})) \tag{2.29}$$

The choice of CVs is crucial as it determines the efficiency and success of enhanced sampling simulations. They should be chosen to represent the slow, large-amplitude motions of the system and to distinguish between different states of interest. Common choices for CVs include distances between atoms or groups of atoms, angles, dihedrals, and root-mean-square deviations (RMSD) from a reference structure. The selection often relies on prior knowledge of the system and the specific reacion process under investigation.

In the scope of this thesis, it suffices to mention that the CVs for modelling oxygen reduction are selected as path collective variables,[116] which is capable of describe the progress of a system along a predefined reaction path. Given a reference path defined by a series of configurations $\{\mathbf{R_0}, \mathbf{R_1}, \ldots, \mathbf{R_N}\}$,the progress along the path s for a given

configuration $\mathbf{R}$) is defined as:

$$s = \frac{\sum_{i=1}^{N} i e^{-\lambda d(\mathbf{R},\mathbf{R}_i)}}{\sum_{i=1}^{N} e^{-\lambda d(\mathbf{R},\mathbf{R}_i)}} \tag{2.30}$$

where $d(\mathbf{R}, \mathbf{R}_i)$ is a distance metric between the configuration $\mathbf{R}$) and the reference configuration $\mathbf{R}_i$). Various metrics can be employed for this purpose. A common choice is the root-mean-square deviation (RMSD) between the two configurations. Alternatively, one can define collective variables (CVs) to characterize the reference structures and then compute the distance based on the differences in these CV values. Using the primary path collective variable s alone is not enough to describe a reaction path as it does not provide information about how far the system deviates from this path in the orthogonal directions. The complementary variable z fills this gap by quantifying the perpendicular distance from the current configuration to the reference path, which can be written as:

$$z = -\frac{1}{\lambda} \ln \left[\sum_{i=1}^{N} e^{-\lambda d(\mathbf{R},\mathbf{R}_i)} \right] \tag{2.31}$$

In this thesis, the distance between configurations are described by the coordination numbers $C_{\mathrm{O_2-O}}$ and $C_{\mathrm{O_2-H}}$, which are the number of neighboring oxygen atoms and hydrogen atoms surrounding the O_2 molecules, respectively.

2.4.2 Enhanced sampling methods

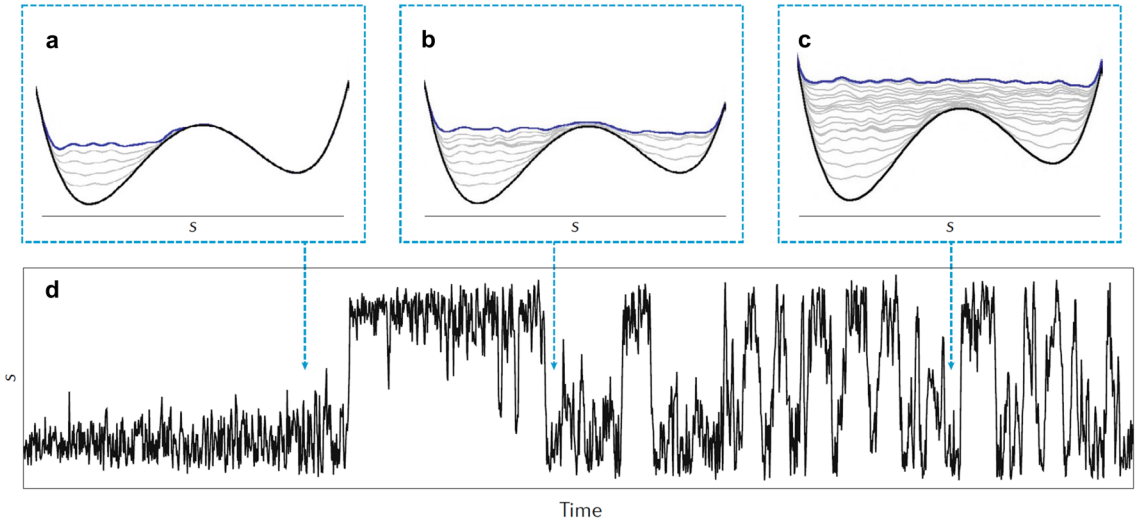

Figure 2.4: (a-c) The free energy landscape described by CV s and the metadynamics bias potential at three different times marked by arrows in (d). (d) The CV s as a function of time in a metadynamics simulation. Figure reprinted from [115] with permission.

Several enhanced sampling methods have been developed to address the challenges associated with sampling in MD simulations. Replica exchange molecular dynamics [117] (REMD) is a method where multiple replicas of the system are simulated in parallel at different temperatures, and exchanges between replicas are attempted periodically to enhance the sampling of configurational space. One of the primary drawbacks of REMD is the significant computational cost as running multiple replicas of the system at different temperatures requires substantial computational resources. Moreover, REMD is primarily effective for systems with well-defined energy landscapes and clear temperature-dependent behavior. For systems with rugged energy landscapes or those that do not exhibit significant temperature-dependent variations, REMD may be inferior compared to other sampling methods. Umbrella Sampling is one such method that uses a biasing potential to force the system to sample different regions of phase space, and the results from different simulations are then combined to obtain the overall properties of the system.[118, 119] A notable limitation of this approach is the prerequisite to predefine the bias potential prior to the simulation. This often requires iterative analyses and tests on the model system to ensure the incorporation of the bias potential is effective.

Metadynamics is another method,[113, 114, 120] which adds a history-dependent biasing potential to smooth out the free energy landscape, enabling the system to escape from local minima and explore the configurational space more efficiently. As demonstrated in Figure 2.4, as the simulation progresses, the biasing potential fills the free energy basins associated with the CVs, making it easier for the system to cross energy barriers. The biasing potential $V(\mathbf{s}, t)$ is constructed as a sum of Gaussian functions centered at the values of the CVs visited during the simulation:

$$V(\mathbf{s}, t) = \sum_{\tau=\Delta t}^{t} W(\tau) \exp\left(-\frac{||\mathbf{s} - \mathbf{s}(\tau)||^2}{2\delta^2}\right) \tag{2.32}$$

where $\mathbf{s}$ is the vector of CVs, $W(\tau)$ is the height of the Gaussian added at time τ, and δ is the width of the Gaussian. The sum runs over the simulation time, with Gaussians added every Δt time steps.

In the original metadynamics, the height of the Gaussians remains constant, which can lead to the biasing potential diverging over time. The well-tempered variant introduces a time-dependent Gaussian height to ensure convergence:

$$W(\tau) = W_0 \exp\left(-\frac{V(\mathbf{s}(\tau), \tau)}{k_B T \Delta T}\right) \tag{2.33}$$

where W_0 is the initial Gaussian height, k_B is the Boltzmann constant, T is the temperature of the system, and ΔT is a biasing temperature parameter. The free energy $A(s)$ as

a function of the CVs can be reconstructed from the biasing potential:

$$A(\mathbf{s}) = -\lim_{t\to\infty} V(\mathbf{s}, t) \tag{2.34}$$

Metadynamics allows for the efficient exploration of the free energy landscape, especially for systems with high energy barriers. With the well-tempered variant, the convergence of the free energy profile can be ensured. However, it should be noted that the success of metadynamics largely depends on the choice of appropriate CVs that can distinguish between different states of the system.

In the scope of this thesis, all metadynamics simulations are performed with the well-tempered variant, and are propagated by Langevin dynamics in ASE.[98] The calculation of collective variables and bias potential of metadynamics is achieved by PLUMED[121, 122, 117] that interfaced to ASE.

CHAPTER 3

Automated active learning workflow

This chapter is based on paper I – "Batch Active Learning at the Core: Building Robust ML Potentials for Atomistic Simulations". The paper is also included in this thesis together with the corresponding supplementary information in Appendix A.

3.1 Introduction

In this chapter, we introduce `CURATOR`, an autonomous active learning workflow devised for the construction of high-fidelity graph neural network potentials. This workflow seamlessly integrates cutting-edge equivariant MPNNs—specifically PaiNN,[110] NequIP,[66] and MACE,[67] targeting accurate predictions for specific properties within chemical systems. To ensure the robustness of simulations driven by the trained MLIPs, our approach incorporates a variety of uncertainty quantification techniques. We have incorporated efficient active learning strategies that can efficiently identify the most informative batches of structures from production simulations, adaptively enhancing model reliability and expanding their applicability across a broader chemical space. These strategies exploit both the model uncertainties and diversity of the candidate atomic configurations, improving the efficiency of batch mode data acquisition. Through rigorous testing across diverse chemical systems—ranging from simple molecules to intricate periodic molten glass structures, we demonstrate that the algorithms can remarkably enhance data acquisition efficiency. This, in turn, substantially reduces the time required to train high-quality MLIPs. In order to further accelerate the simulation speed of these MLIPs, we have developed an efficient gradient computation method that calculates forces and stress based on the energy derivative with respect to relative position vectors. Lastly, by integrating the entire workflow with myqueue,[123] we have achieved full automation in the task scheduling on

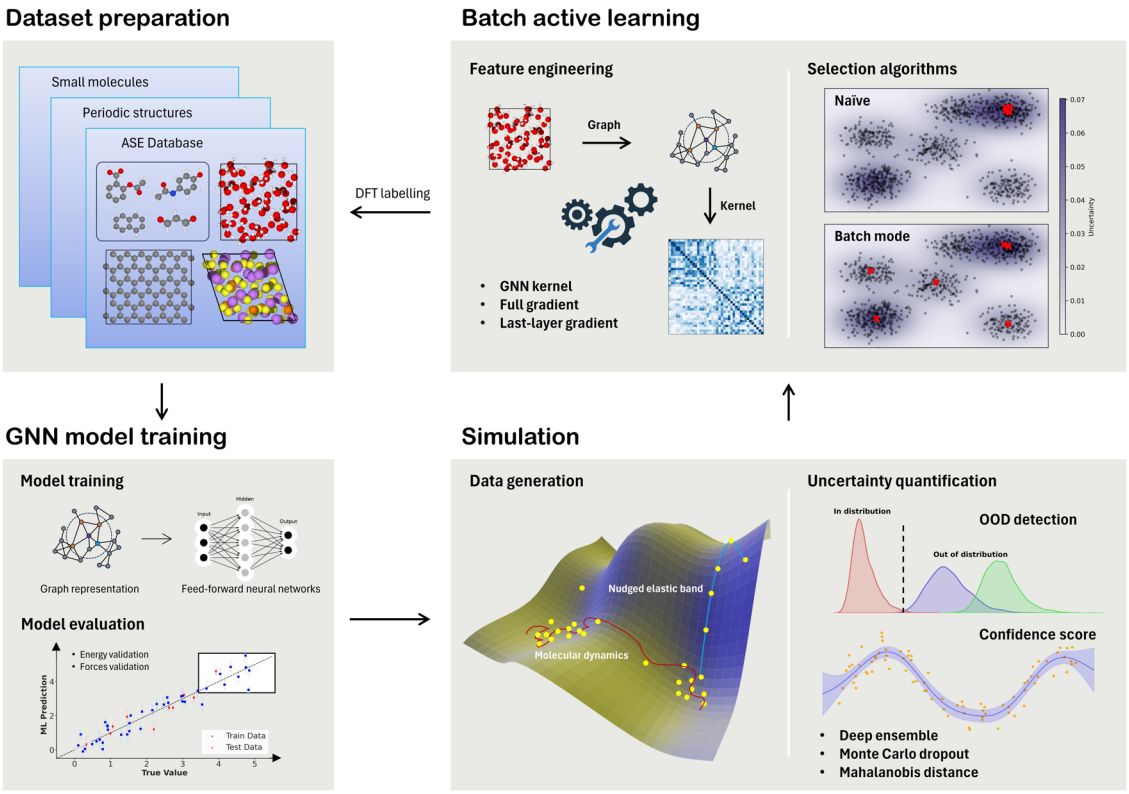

Figure 3.1: Schematic diagram of active learning workflow

modern computer clusters for various job types within the framework.

Figure 3.1 outlines the various procedures for fitting MLIPs. Initially, users must provide a small dataset comprising atomic configurations derived from DFT calculations. This dataset could originate from diverse calculations, such as a short MD trajectory, configurations from structural optimizations, or nudged elastic band calculations, among others. This initial dataset serves as the foundation for training the GNN model. Within this process, atomic configurations are mapped into graph representations, which are then modeled using feed-forward neural networks. Training stops when there is no improvement in validation error over a specified number of steps. The resulting models can then be used to generate data via methods like molecular dynamics, Monte Carlo simulations, or other user-specified applications. Using our reliable uncertainty toolbox, simulations are guaranteed to stay within the application domain of the trained models; if not, the simulations are immediately stopped. The much improved computational efficiency of MLIPs allows for the fast generation of numerous candidate structures. Batch active learning algorithms are then used to identify the most informative batches for refining the model among these candidates. This involves feature engineering for maximizing the information of individual candidate structures and minimizing the overall memory usage

for storing the information, and effective algorithms that exploit the model uncertainties and data diversity. The chosen data points are subsequently labelled via DFT single-point calculations and incorporated into the initial DFT dataset for model refinement. Such a process will be iteratively performed until the derived GNN models are reliable, accurate, and stable enough for designated simulations. In the subsequent sections, we delve into the specifics of various procedures within the workflow. The remainder of this chapter is structured as follows:

1. **Machine learning interatomic potentials:** We present several state-of-the-art Message Passing Neural Networks (MPNNs) utilized in our workflow, summarizing the trade-offs between model accuracy and speed. We also introduce an efficient method for gradient computation.

2. **Batch active learning:** This section introduces the active learning methods employed in our workflow, detailing the features and crucial transformations used for representing atomic structures. The efficacy of various active learning strategies is demonstrated through several selected benchmark systems.

3. **Uncertainty Estimation Methods:** We outline the uncertainty estimation methods integrated within the workflow and assess their performance against several critical criteria relevant to practical applications.

4. **Autonomous workflow:** Here, we illustrate how the aforementioned components are seamlessly incorporated into our active learning workflow.

5. **Conclusion:** Finally, we conclude with our remarks, summarizing the key findings and implications of our work.

3.2 Neural network potential library

3.2.1 Model training and evaluation

The workflow integrates a series of cutting-edge equivariant message-passing neural networks, specifically PaiNN,[110] NequIP,[66] and MACE.[67] Within these models, each atom is linked to features that encompass tensors of various orders, ranging from scalars and vectors to even more complex higher-order tensors. Leveraging these high-order features guarantees the rotational equivariance of the model, enhancing the accuracy of predictions related to directional attributes, such as dipole moments and forces. Adopting the notations from ref. [66], the feature vectors $V_{acm}^{(l,p)}$ can be indexed by the rotation order l and the parity notation p. Here, the term "rotation order" refers to a non-negative integer l=0,1,2,... and the parity p can either be 1 or -1. Together, they label the O(3)

irreducible representations of atomic features. Among the discussed models, PaiNN exclusively employs l=1 vectors and ignores parity transformation. This approach simplifies the dimensionality of the features, enabling PaiNN to achieve faster training and inference speeds without necessarily compromising on accuracy. In contrast, NequIP typically utilizes l=2 vectors and takes into account parity transformation. While this improves model accuracy, it demands greater computational resources. In both architectures, usually more than three message-passing layers are required to achieve desired accuracy levels. MACE differentiates itself by incorporating higher body-order interactions in its message functions, allowing for only two message-passing iterations to attain high accuracy. This design potentially optimizes computational efficiency while maintaining excellent model performance. For more detailed information on the models, please refer to the respective publications.

Figure 3.2 illustrates the trade-off between model accuracy, quantified by the mean absolute error (MAE) of forces, and inference speed. The models are trained using the aspirin molecule data from the MD17 dataset, comprising 2,000 training, 1,000 validation, and 5,000 independent test data points. More details about the error metrics and inference speed for each model can be found in Table A.1. The inference time for each model is evaluated on a diamond structure with 1000 atoms using an NVIDIA V100 GPU. This structure has an average of 86 neighbors per atom with a cutoff radius of 5.0 Å. It is important to highlight that the inference time generally remains consistent regardless of the number of atoms in the system, until all the GPU threads are occupied. As anticipated, the error diminishes with an increase in the number of message-passing layers and node feature size. However, beyond a certain threshold, increasing the number of layers and node feature size yields only marginal improvements in accuracy. It is clearly seen that both NequIP and MACE have shown outstanding model accuracy, while at a cost of much heavier computation as shown in Figure 3.2c and d. Therefore, it is recommended to use cheap PaiNN model for collecting training data points with active learning while to use more accurate models like MACE or NequIP for final production simulations.

3.2.2 Efficient gradient calculation

In the original implementations of GNN model, the total potential energy of the chemical system is calculated by aggregating individual atomic energies. Concurrently, atomic forces are generally derived from the negative gradients of the atomic energy with respect

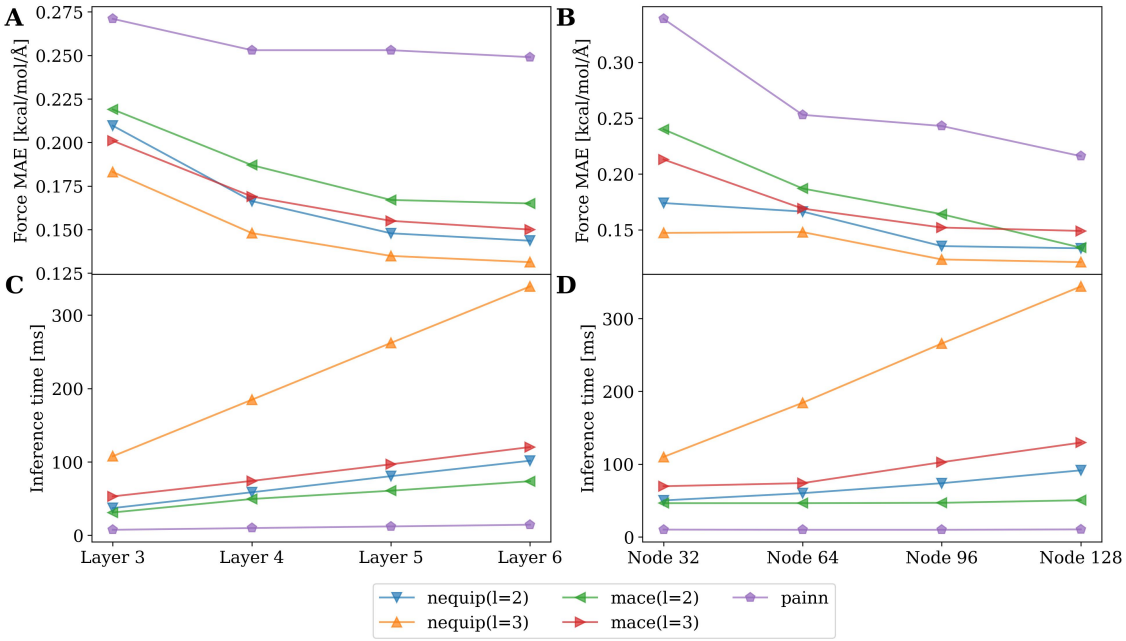

Figure 3.2: (a) and (b) Model accuracy (force MAE) of employed models plotted against the number of message-passing layers and the number of node features, respectively. (c) and (d) Inference time of employed models plotted against the number of message-passing layers and the number of node features, respectively.

to atomic coordinates:

$$E = \sum_{i \in N} E_i \tag{3.1}$$

$$\vec{F}_i = -\nabla_i E \tag{3.2}$$

This approach respects the energy conservation constraint, which is important for improving the stability of simulations such as molecular dynamics.[111] Notice that the models never directly use the coordinates of atoms $\vec{r_i}$ to determine atomic energies. Instead, they rely solely on the relative position vector $\vec{r_{ij}} = \vec{r_j} - \vec{r_i}$ and its length $\parallel \vec{r_{ij}} \parallel$ in the message layers, which are typically obtained via neighbor-list algorithms from various codes like ASE,[98] ASAP3,[124] MatScipy,[125] or NNPOps.[126] Therefore, the atomic energy is exclusively a function of $\vec{r_{ij}}$:

$$E_i = E_i(\{\vec{r_{ij}}\}_{i \neq j}) \tag{3.3}$$

The automatic differentiation feature in PyTorch,[127] which is typically the backend of most GNN models, enables the convenient computation of negative gradients of total potential energy with respect to the model inputs, i.e. relative position vectors. Yet, the derivative $\partial \vec{r_{ij}} / \partial \vec{r_i}$ remains a missing map for force calculations. For non-periodic

systems, this derivative is straightforward to compute, whereas periodic systems require consideration of cell displacements, adding extra computational overhead during data preprocessing. In contrast, our implementation calculates forces as follows:

$$\begin{aligned}\vec{F_i} &\equiv -\frac{\partial E}{\partial \vec{r_i}} \equiv -\sum_i \frac{\partial E_i}{\partial \vec{r_i}} \\ &= -\sum_{j\neq i}\left(\frac{\partial E_j}{\partial \vec{r_i}}\right) - \frac{\partial E_i}{\partial \vec{r_i}} \\ &= -\sum_{j\neq i}\left(\sum_{k\neq j}\frac{\partial E_j}{\partial \vec{r_{jk}}}\frac{\partial \vec{r_{jk}}}{\partial \vec{r_i}} + \frac{\partial E_i}{\partial \vec{r_{ij}}}\frac{\partial \vec{r_{ij}}}{\partial \vec{r_i}}\right) \\ &= -\sum_{j\neq i}\left(\frac{\partial E_i}{\partial \vec{r_{ij}}} - \frac{\partial E_j}{\partial \vec{r_{ji}}}\right)\end{aligned} \tag{3.4}$$

In this way, the forces can be computed with only $-\partial E/\partial \vec{r_{ij}}$ that can be directly obtained with automatic differentiation. This bypasses the need to compute cell displacements and re-calculate relative position vectors, streamlining the process. Additionally, by using the neighbor list of individual atoms, we can independently determine the total forces for each atom, which offers significant potential for massively parallel implementation.

Moreover, this method can also notably reduce the effort required to compute the stress of the chemical system by using an explicit analytical expression for virial tensors. Typically, the stress tensors of a periodic system can be calculated with the first-order derivative of the total energy E with respect to small strains.[128] This method requires applying a symmetrical, infinitesimal strain deformation to the periodic system prior to the model prediction. Following this, the gradients of the total energy related to the strain tensors must be calculated. This process doubles the computational burden of gradient calculations, which represent the most significant computational expense in model prediction. In our implementation, it is worth noting that the computed force is pairwise and adheres to Newton's third law:

$$\vec{F_{ij}} = -\vec{F_{ji}} = -\frac{\partial E_i}{\partial \vec{r_{ij}}} + \frac{\partial E_j}{\partial \vec{r_{ji}}} \tag{3.5}$$

where $\vec{F_{ij}}$ is the force exerted by atom j on atom i. The viral tensors can then be calculated by:

$$W = \sum_i W_i = -\frac{1}{2}\sum_i \sum_{j\neq i} r_{ij} \otimes \vec{F_{ij}} \tag{3.6}$$

This method employs an explicit expression for computing the virial stress tensors, eliminating the need for computationally intensive gradient calculations associated with stress tensors.

3.3 Batch active learning

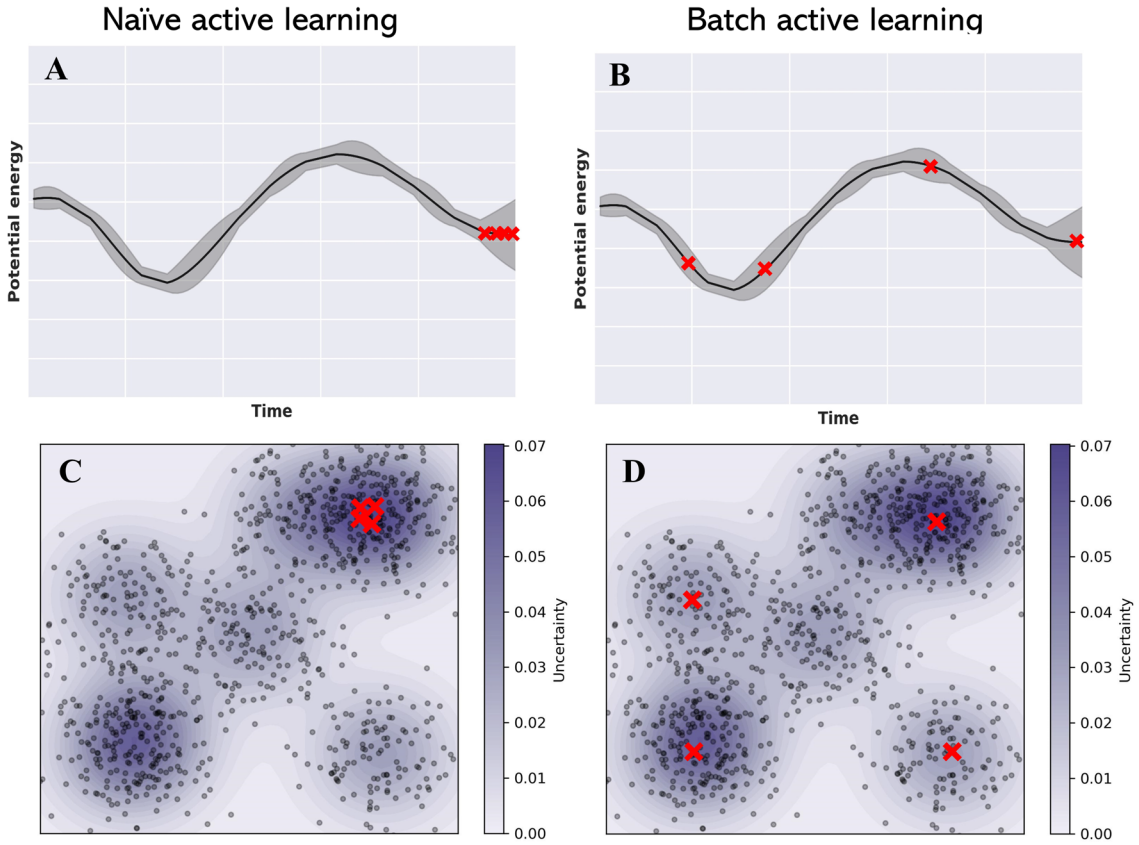

Figure 3.3: Schematic illustrations of active learning strategies: (a) naive active learning and (b) batch active learning methods for selecting data points from an MD simulation trajectory; (c) naive active learning and (b) batch active learning methods for selections in a two-dimensional space.

Active learning operates in two primary modes. In naive active learning, the algorithm continuously selects and labels the single most informative sample, updating the model after each instance. If utilizing this strategy to select a batch instead of an individual sample, multiple informative but similar samples could be selected, potentially making the labeling of these samples redundant. This becomes especially critical in specific simulations. For example, in MLMD simulations, the extrapolative structures selected by naive active learning often present in a short time interval right before the simulation ends, as illustrated in Figure 3.3a and c. Even though each of these structures is individually informative, their similarity can lead to only marginal improvements during subsequent model retraining due to the overlap in information. On the other hand, batch active learning is designed to choose and annotate sets of samples simultaneously, prioritizing both uncertainty and diversity within the batch. Optimal batch active learning methods aim to choose samples that have high uncertainties while minimizing information redundancy,

as illustrated in Figure 3.3b and d. Achieving this involves measuring atomic configuration similarities and strategically excluding similar structures from a batch. This process takes into account the features used to describe atomic structures and efficient selection algorithms, details of which will be discussed in the subsequent sections.

3.3.1 Feature engineering

Before exploring active learning selection, we must first extract features from our trained models for the candidate structures. Additionally, it is essential to understand the kernel matrix used in active learning. The following content is based on the framework from Ref.[82], where further details are provided. For a sequence of atoms taken from these structures, represented as $\mathcal{X} = (\boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{n \times d}$, the corresponding feature matrix can be defined as:

$$\Phi(\boldsymbol{\mathcal{X}})^T = \begin{pmatrix} \phi(\boldsymbol{x_1})^T \\ \vdots \\ \phi(\boldsymbol{x_n})^T \end{pmatrix} \in \mathbb{R}^{n \times d_{feat}} \tag{3.7}$$

In this context, $\phi(x_i)$ represents the feature map for an individual atom derived from the model. The local environments of two atoms i and j then can be compared by using the similarity kernel $k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle \phi(\boldsymbol{x_i}), \phi(\boldsymbol{x_j}) \rangle$. Expanding on this, we can compute the covariance matrix $k(\mathcal{X}, \mathcal{X}) = (k(\boldsymbol{x_i}, \boldsymbol{x_j}))_{i,j} \in \mathbb{R}^{n \times n}$ that encompasses all pairwise similarity within the feature matrix. There are various ways to construct the feature map and the kernel matrix. Besides, in order to make these kernels being more suitable to be applied for a selection method, some kernel transformation methods are often needed. In the following, we will introduce several kernels and transformation methods used in this study.

GNN kernel: The most intuitive approach for obtaining feature maps is leveraging the scalar node features derived from the outputs of the message-passing layers within MPNNs. This corresponds to the feature map ϕ_{gnn} and the graph neural network kernel k_{gnn}. Although evaluating this kernel through model prediction is generally fast and convenient, it solely contains the information necessary for computing the potential energy of the chemical system, ignoring the gradients of the systems. This could potentially limit its ability to accurately describe the atomic environment, consequently compromising the effectiveness of batch selection methods.

Full gradient kernel: Compared to GNN kernel, the full gradient kernel takes use of all gradients from the model to construct the feature map, which can be expressed as:

$$\phi_{grad}(\boldsymbol{x}) := \nabla_\theta f_{\theta_T}(\boldsymbol{x}) \tag{3.8}$$

where θ_T is the parameter vector of the trained model. The intuition of this method is that the magnitude of the gradients implies the required adjustments of the parameters in different dimensions, and thus can be used to evaluate the distance between different samples. Besides, it also indicates the gap between predictions and correct values, enabling it to be a potential indicator of model uncertainty.[129]

The number of parameters in deep learning models can often be large, therefore it is intractable to get the gradients of all these parameters and to use the ultra-high-dimensional features for selection. Fortunately, the feature map ϕ_{grad} can be simplified by using the product structure of NNs, which can significantly reduce the runtime and memory usage for kernel evaluation.[82]

$$\boldsymbol{z}_i^{(l+1)} = \tilde{\boldsymbol{W}}^{(l+1)}\boldsymbol{x}_i^{(l)},\ \tilde{\boldsymbol{W}}^{(l+1)} := (\boldsymbol{W}^{(l+1)}\ \boldsymbol{b}^{(l+1)}) \in \mathbb{R}^{d_{l+1}\times(d_l+1)},\ \tilde{\boldsymbol{x}}_i^{(l)} = \begin{pmatrix} \boldsymbol{x}_i^{(l)} \\ 1 \end{pmatrix} \in \mathbb{R}^{d_l+1} \tag{3.9}$$

$$\phi_{grad}(\boldsymbol{x}_i^{(0)}) = \left(\frac{d\boldsymbol{z}^{(L)}}{d\tilde{\boldsymbol{W}}^{(1)}}, \ldots, \frac{d\boldsymbol{z}^{(L)}}{d\tilde{\boldsymbol{W}}^{(L)}}\right) = \left(\frac{d\boldsymbol{z}_i^{(L)}}{d\boldsymbol{z}_i^{(1)}}(\tilde{\boldsymbol{x}}_i^{(0)})^T, \ldots, \frac{d\boldsymbol{z}_i^{(L)}}{d\boldsymbol{z}_i^{(L)}}(\tilde{\boldsymbol{x}}_i^{(L-1)})^T\right) \tag{3.10}$$

Given that ϕ_{grad} encompasses gradient contributions across different layers, sometimes it is required to balance the magnitudes of the gradients in different layers via parameter initialization [130] or normalize the gradients post hoc. While the aforementioned derivations were initially intended for fully connected neural networks (NNs), they can also be extended for application to the FFNN component within MPNNs. This adaptation is precisely how the full gradient kernel was employed in the context of this study.

Last layer kernel: The dimensionality of a full gradient feature map can often be too large. A simple approximation to this is only using the gradients of parameters in the last layer of NNs as the feature map ϕ_{ll}.[131] From equation 3.10, it is evident that ϕ_{ll} is just the input of the last layer.

Average transformation: Note that the number of atoms in the pool dataset can be vast, often ranging from several millions to billions. Direct pairwise comparisons pose significant challenges in terms of memory consumption and computational efficiency. Thus, merging the local feature maps of atoms to generate a global similarity measurement for structures is a more practical approach. When comparing two structures, a straightforward method is to use the average kernel. It is important to clarify that in this context, the term "average kernel" is somewhat misleading. It encompasses both the mean feature map of a group of atoms relative to a structure and the cumulative sum of feature maps. Mathematically,

this can be represented as:

$$\phi(\boldsymbol{S_i}) = \phi_{\rightarrow avg}(\boldsymbol{x}) = \sum_{n=1}^{N_{atoms}} \phi(x_n) \tag{3.11}$$

where S_i denotes a structure and $\phi_{\rightarrow avg}(\boldsymbol{x})$ denotes the transformation for feature maps. This notation will also be used for other transformations hereafter. Although this method can lead to some information loss, its small computational cost can greatly accelerate the selection and minimize memory consumption. More accurate methods like regularized entropy match (REMatch) can also be used to construct the global similarity kernel.[132]

Diagonal kernels: Diagonal kernels correspond to the metrics that are used for naive active learning. These metrics can individually indicate the informativeness of selected samples while capturing no correlation between them. There are multiple ways to select these metrics. When the labels (i.e., material properties like energy and forces) of samples are known, the absolute error (AE) between true values and predictions can serve as a suitable indicator for the informativeness of individual samples. The absolute error of energy and forces are considered in this study, which can be expressed as:

$$\triangle E(S) = |E^{pred} - E^{true}| \tag{3.12}$$

$$\triangle F(S) = \frac{1}{3N_{atoms}} \sum_{i=1}^{N_{atoms}} \sum_{j=1}^{3} |\vec{F}_{ij}^{pred} - \vec{F}_{ij}^{true}| \tag{3.13}$$

These two kernels will be referred to as AE(E) and AE(F) hereafter. When the labels of samples are unknown, we can then use some sampling-based UE methods to evaluate the disagreements between different predictions that obtained from different models or Monte-Carlo dropout, thus obtaining the uncertainty. There are multiple ways to calculate the disagreements, here we simply use the standard deviation of different predictions, which can be expressed as:

$$\sigma_E(S) = \sqrt{\sum_{n=1}^{N_{pred}} (E^{pred} - E^{true})^2} \tag{3.14}$$

$$\sigma_F(S) = \sqrt{\frac{1}{3N_{atoms}N_{pred}} \sum_{n=1}^{N_{pred}} \sum_{i=1}^{N_{atoms}} \sum_{j=1}^{3} (\vec{F}_{ij}^{pred} - \vec{F}_{ij}^{true})^2} \tag{3.15}$$

These two kernels will be referred to as QBC(E) and QBC(F) hereafter.

Random projections: Although the last-layer kernel can approximate the full gradient kernel to some extent, the information loss due to the discarded gradients can be large, undermining its ability to describe atomic environments. Random projections, also known

as sketching, can be used to approximate a high-dimensional feature by a lower-dimensional feature.

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \frac{1}{\sqrt{p}} U\phi(x) \in \mathbb{R}^p \tag{3.16}$$

where $U \in \mathbb{R}^{p \times d_{feat}}$ is a random matrix with entries drawn from a standard normal distribution. In the case of feature map $\phi_{grad\to avg}(\boldsymbol{x})$, the following approximations are employed to simplify the sum and product of feature maps $\phi(\boldsymbol{x}) := (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))^T$ and $\phi(\boldsymbol{x}) := \phi(\boldsymbol{x}_1) \otimes \phi(\boldsymbol{x}_1)$:

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \phi_{1\to rp(p)}(\boldsymbol{x}_1) + \phi_{2\to rp(p)}(\boldsymbol{x}_1) \tag{3.17}$$

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \phi_{1\to rp(p)}(\boldsymbol{x}_1) \otimes \phi_{2\to rp(p)}(\boldsymbol{x}_1) \tag{3.18}$$

In this way, the full gradient feature map can be conveniently transformed into features with p dimensionality.

Gaussian process transformation: Gaussian process posterior transformation is derived from a Bayesian linear regression model with respect to feature $\phi(\boldsymbol{x})$, where the atomwise property y_i can be modeled by $y_i = \boldsymbol{w}^T\phi(\boldsymbol{x}) + \varepsilon$.[133] After observing the training data $\mathcal{D}_{train}$ with inputs $\mathcal{X}_{train}$, it is well known that the posterior covariance $k(\boldsymbol{x}, \boldsymbol{x}^*|\mathcal{X}_{train})$ can be obtained by:

$$k(\boldsymbol{x}, \boldsymbol{x}'|\mathcal{X}_{train}) = k(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \mathcal{X}_{train})(k(\mathcal{X}_{train}, \mathcal{X}_{train}) + \sigma^2\boldsymbol{I})^{-1}k(\mathcal{X}_{train}, \boldsymbol{x}') \tag{3.19}$$

$$= \phi(\boldsymbol{x})^T(I - \Phi(\mathcal{X}_{train})(\Phi(\mathcal{X}_{train})^T\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-1})\phi(\boldsymbol{x}') \tag{3.20}$$

Using the matrix inversion lemma (also known as the Woodbury matrix identity)[133] we can get:

$$k(\boldsymbol{x}, \boldsymbol{x}'|\mathcal{X}_{train}) = \sigma^2\phi(\boldsymbol{x})^T(\Phi(\mathcal{X}_{train})^T(\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-1}\phi(\boldsymbol{x}'), \tag{3.21}$$

which leads to an explicit feature map:

$$\phi_{\to gp}(\boldsymbol{x}) = \sigma(\Phi(\mathcal{X}_{train})^T(\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-\frac{1}{2}}\phi(\boldsymbol{x}). \tag{3.22}$$

This feature map can then be used for measuring the similarity between structures. The idea of this transformation can be seen as approximating the feed-forward NN in MPNNs as a Bayesian NN, providing a fast and robust way to evaluate model uncertainty. We will demonstrate in the subsequent section that this operation is indeed equivalent to using Mahalanobis distance for out-of-distribution detection.[134]

On the basis of the above kernels and transformations for atomic features, we come up with 6 different combinations, namely $\phi_{gnn\to avg}(x)$, $\phi_{grad\to rp\to avg}$, $\phi_{ll\to avg}$, $\phi_{grad\to rp\to avg\to GP}$,

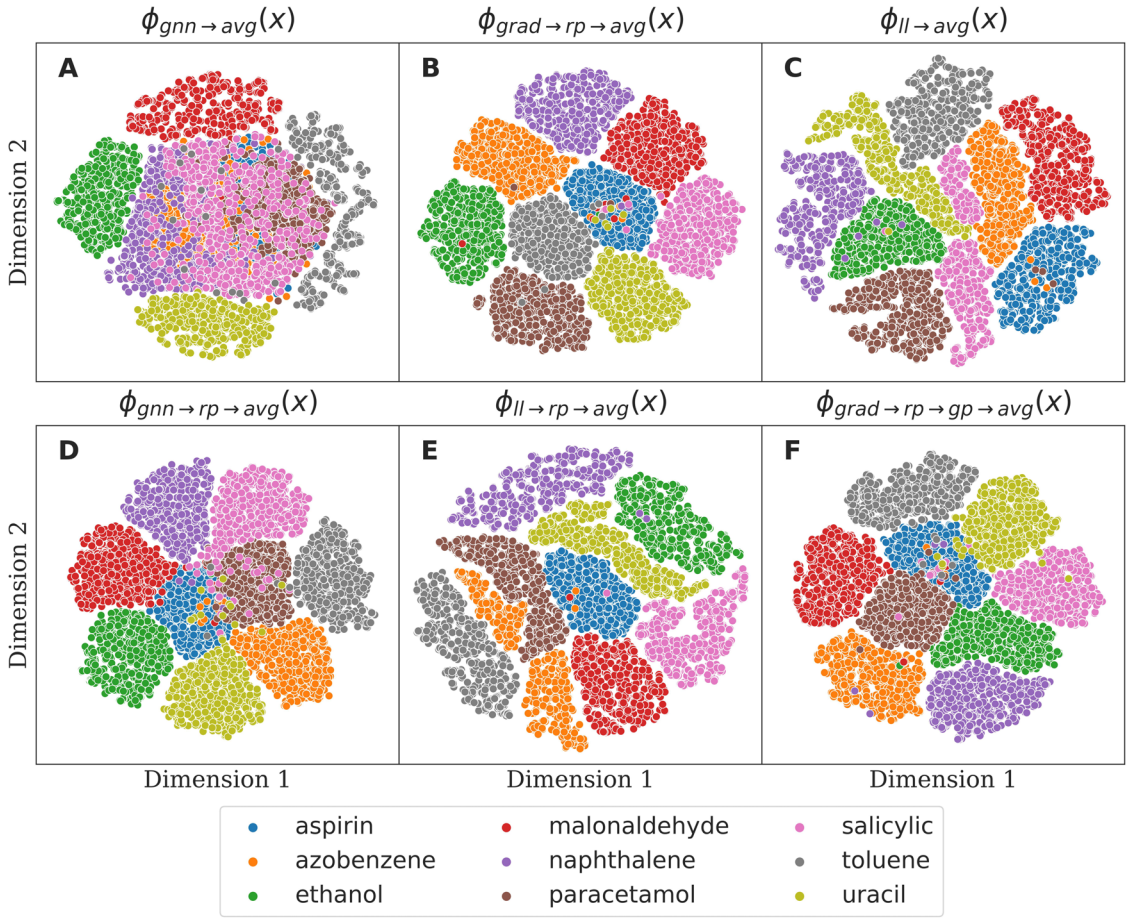

Figure 3.4: t-SNE plot of the applied kernels and transformations on the features derived from MD17 dataset.

$\phi_{ll \to rp \to avg}$, $\phi_{grad \to rp \to avg \to GP}$. A suitable tool to evaluate their ability to accurately represent atomic structures and differentiate similar structures is t-SNE (t-Distributed Stochastic Neighbor Embedding)[135], which embeds the high-dimensional data points for visualization in a low-dimensional space by using probability distributions. Figure 3.4 depicts the distribution of various molecules from the MD17 dataset, visualized using t-SNE and the kernels mentioned above. From Figure 3.4a, it is evident that solely relying on the GNN kernel $\phi_{gnn \to avg}(x)$ is not enough to capture the structural differences between these molecules. Introducing random projection transformation to the GNN kernel offers a moderate improvement, as seen in Figure 3.4d, but distinguishing between different structures remains challenging. In stark contrast, both the full gradient kernel and the last gradient kernel demonstrate superior capability in capturing the structural characteristics of these molecules, evidenced by the distinct separations in Figures 3.4b, c, and e. We also evaluated the impact of Gaussian process transformation, illustrated in Figure 3.4f. Regrettably, no obvious improvement is observed by using this transformation as it is

mainly targeted for better uncertainty evaluation.

3.3.2 Batch mode selection

With the defined kernels above, we can then use some selection methods to select data points from the pool data set. Based on the above results that only $\phi_{grad\to rp\to avg}$ and $\phi_{ll\to avg}$ well represented the distribution of atomic structures, only these two kernels are employed for selection. In the following, we will briefly introduce several selection methods used in this study. '[82, 136] Here we use $\mathcal{X}_{pool}$, $\mathcal{X}_{sel}$, $\mathcal{X}_{batch}$ to denote the pool dataset, selected data points, and the batch to be selected, respectively.

Random: Random selection will serve as a baseline for the selection methods and will be denoted as Random. The batch data points $\mathcal{X}_{batch}$ will randomly draw from an uniform distribution

$$\text{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) \sim \mathcal{U}(\mathcal{X}_{pool}) \tag{3.23}$$

The selection continues until N_{batch} number of data points have been collected.

Naive active learning: If using $k(\boldsymbol{x}, \boldsymbol{x})$ as the uncertainty of data point $\boldsymbol{x}$, naive active learning can be conceptualized as selecting data points corresponding to the maximum diagonal elements of $k(\mathcal{X}_{pool}, \mathcal{X}_{pool})$. This method encompasses all QBC methods that employ uncertainty as their selection criterion and will be termed as MaxDiag hereafter. The selection strategy can be expressed as

$$\text{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{\boldsymbol{x}\in\mathcal{X}_{pool}}{\text{argmax}}\ k(\boldsymbol{x}, \boldsymbol{x}). \tag{3.24}$$

This method only considered the informativeness of individual data points while ignoring their similarities, which can lead to similar or even identical data points in $\mathcal{X}_{batch}$.

Greedy determinant maximization: Compared to MaxDiag, the determinant maximization approach, referred to as MaxDet, curates an optimal batch $\mathcal{X}_{batch}$ by maximizing the determinant of $k(\mathcal{X}_{sel} \cup X_{batch}, \mathcal{X}_{sel} \cup X_{batch})$. This can be formalized as

$$\text{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{\boldsymbol{x}\in\mathcal{X}_{pool}}{\text{argmax}}\ \det(k(\mathcal{X}_{sel} \cup \{x\}, \mathcal{X}_{sel} \cup \{x\}) + \sigma^2 I) \tag{3.25}$$

This method accounts for the correlation among selected points, effectively ensuring uncertainty and data diversity within the chosen batches. Calculating the determinants of batches with diverse data points is usually intractable. To alleviate computational complexity, the greedy algorithm utilizing partial pivoted matrix-free Cholesky decomposition[137] is employed. Notably, this approach aligns with the D-optimal design principles previously applied in active learning for machine learning interatomic potentials.[138, 139]

Largest cluster maximum distance: Largest cluster maximum distance (LCMD) is a clustering method that aims to categorize data points from the pool set $\mathcal{X}_{pool}$ by assigning them to predefined cluster centers in $\mathcal{X}_{sel}$. Initially, every point $\boldsymbol{x}$ in $\mathcal{X}_{pool}$ is assigned to its nearest cluster center from $\mathcal{X}_{sel}$, with distances typically computed using metrics like Euclidean:

$$c(\boldsymbol{x}) := \underset{\tilde{\boldsymbol{x}} \in \mathcal{X}_{sel}}{\operatorname{argmax}} \, d_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \tag{3.26}$$

The size of these clusters is determined by the sum of the distances of each member to its cluster center:

$$s(\tilde{\boldsymbol{x}}) := \sum_{c(\boldsymbol{x})=\tilde{\boldsymbol{x}}} d_k(\boldsymbol{x}, \tilde{\boldsymbol{x}})^2 \tag{3.27}$$

Following this, the point in the largest cluster that is at the maximum distance from its center is chosen as the next cluster center.

$$\text{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{s(c(\boldsymbol{x}))=\max \, s(\tilde{\boldsymbol{x}})}{\operatorname{argmax}} \, d_k(\boldsymbol{x}, c(\boldsymbol{x})) \tag{3.28}$$

This iterative process continues until the desired number of cluster centers matches the batch size. This method emphasizes both the representativeness and diversity of data points, thus making the batch mode selection effective.

We conducted experiments on multiple datasets to assess the effectiveness of our selection methods and kernels. These datasets include: AIMD simulations of small molecules from the MD17 dataset (non-periodic) [111], an AIMD trajectory of bulk lithium thiophosphate, $Li_{6.75}P_3S_{11}$ (periodic) [140], and an AIMD trajectory of amorphous lithium phosphate, $Li_4P_2O_7$ (periodic) [66]. With a wide range of base kernels, kernel transformations, and selection modes available to us, the potential combinations were vast. This would necessitate an exhaustive number of benchmark tests, making the task unfeasibly complex. To reduce the complexity of benchmark tests and maintain clarity in our analysis, we strategically limited our focus on a select few combinations that appeared most promising. we restricted the kernels to be the full gradient and the last layer gradient, as they demonstrated a superior capability in accurately representing different structures. Both kernels are transformed by random projections with a dimensionality of 500, which can be expressed as LL(RP) and GRAP(RP). Consequently, our tests were streamlined to the following combinations: Random, MaxDiag+$\{$AE(E), AE(F), QBC(F)$\}$, MaxDet+$\{$LL(RP), GRAD(RP)$\}$, and LCMD+$\{$LL(RP), GRAD(RP)$\}$. Furthermore, all active learning tests are conducted with PaiNN model because it demonstrated superior training efficiency without compromising too much accuracy, while we anticipate that other GNN models might yield similar results.

MD17 active learning tests: We first tested these batch active learning strategies on the MD17 dataset, which comprised of MD trajectories of small molecules. The primary objective is to assess the effectiveness of various batch active learning strategies, utilizing a minimal number of data points while maximizing accuracy. The MD trajectories contain a number of frames ranging from approximately 100,000 to 1,000,000. Most of these frames are similar, making them highly suitable for active learning tests. For each molecule, a subset of 1,000 samples will be reserved as a validation dataset for early stopping, and an additional 5,000 samples will be used for an independent test of the model that exhibits the smallest validation loss in each training. The models are trained on a combined loss of energies and forces, with the energy and force weights being 0.05 and 0.95 respectively. The training began with an initial training data set of 100 samples, drawn randomly from the remaining pool dataset. Throughout each test, the training dataset is increased by a batch size of 100 until a total of 1,000 samples have been collected. The training stops when the validation loss does not improve over 150 times of validation checks. Performance was measured using multiple error metrics, including energy-based and force-based metrics like mean absolute error (MAE), root mean square error (RMSE), and maximum error (MAXE). Particularly, force error metrics were highlighted due to their pivotal role in atomic simulations such as MD, NEB, and structural optimization, where energy typically serves merely as an observer. Moreover, it is typically more challenging to achieve satisfactory force predictions.

The learning curves for the salicylic acid molecule, with a batch size of 100, are illustrated in Figure 3.5. Observations from other molecules mirrored these findings, as seen in Figure A.1 through Figure A.8. Notably, the LCMD+GRAD(RP) combination consistently yielded the smallest force errors, with MAE, RMSE, and MAXE values being 0.182, 0.286, and 0.868 kcal/mol/Å, respectively. This is in stark contrast to the baseline method Random, which exhibited force MAE, RMSE, and MAXE values of 0.289, 0.740, and 2.860 kcal/mol/Å. Remarkably, the LCMD+GRAD(RP) combination achieved similar force accuracy to the Random method but used only half the data points (500 configurations), recording 0.332, 0.563, and 1.768 kcal/mol/Å for force MAE, RMSE, and MAXE, respectively. As anticipated, some naive active learning strategies, notably MaxDiag+AE(E) and MaxDiag+AE(F), distinctly underperformed compared to Random, highlighting the crucial importance of utilizing refined batch active learning methods. It is worth noting that the force MAXE learning curve of LCMD+GRAD(GP) is notably stable. This metric is often associated with the stability of MD simulations, as large force errors can lead to the rapid collapse of a simulation within a short time interval. Consequently, we expect that this approach will considerably enhance the stability of the simulations. Compared to

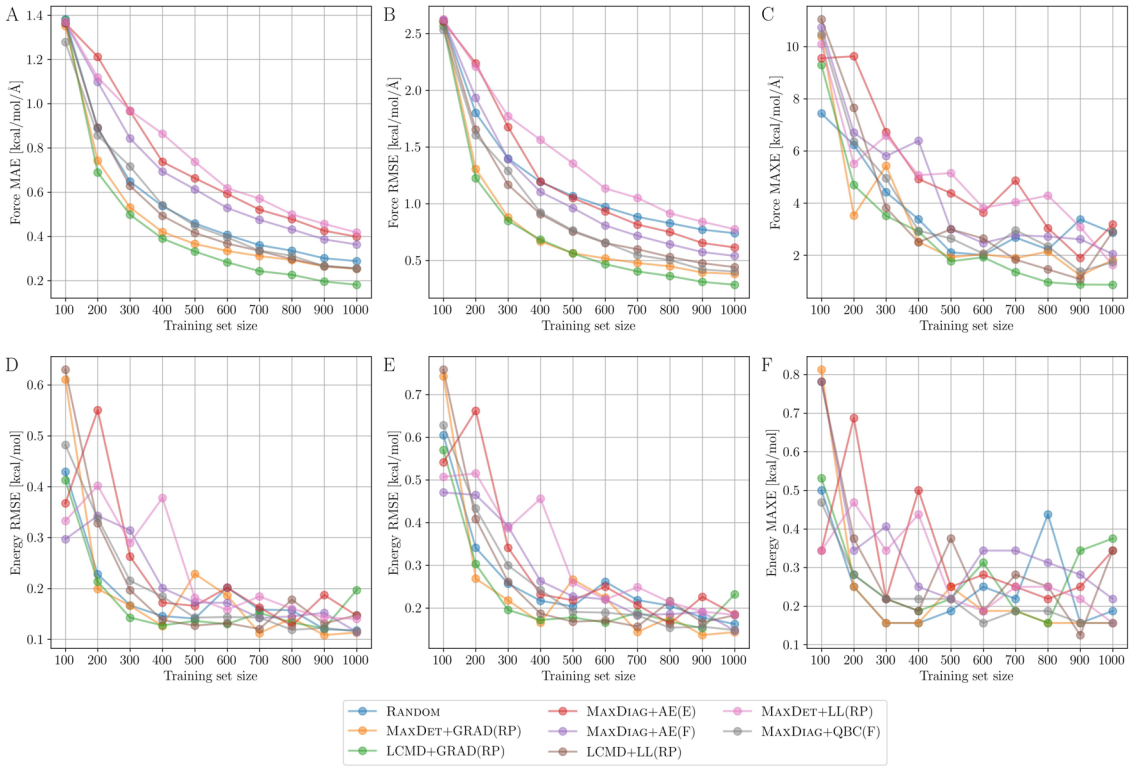

Figure 3.5: Learning curves for the salicylic acid molecule data from MD17. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

the learning curves of force error metrics, those for energy error metrics show significantly greater fluctuations. We attribute this to the excessively low loss of weight assigned to energy. We expect that either increasing the loss weight for energy or training energy-only models could yield learning curves similar to those of force metrics.

LiPS active learning tests: In realistic simulations, chemical systems typically contain a larger number of atoms than the small molecules in MD17, and many of them are periodic structures. To evaluate batch active learning strategies in more general and more challenging contexts, we incorporated two additional datasets with periodic structures. All the active learning procedures employed remain consistent with the MD17 case. Specifically, we first employ a dataset for $Li_{6.75}P_3S_{11}$ (LiPS), a crystalline superionic Li conductor with 83 atoms in a 12.38×12.26×12.44× Å triclinic cell.[140] This dataset contains 25,001 MD frames that are derived from a 50 ps NVT AIMD simulation at 520K with a timestep of 2 ps. From these, 1,000 frames serve as the validation set, 5,000 are designated for inde-

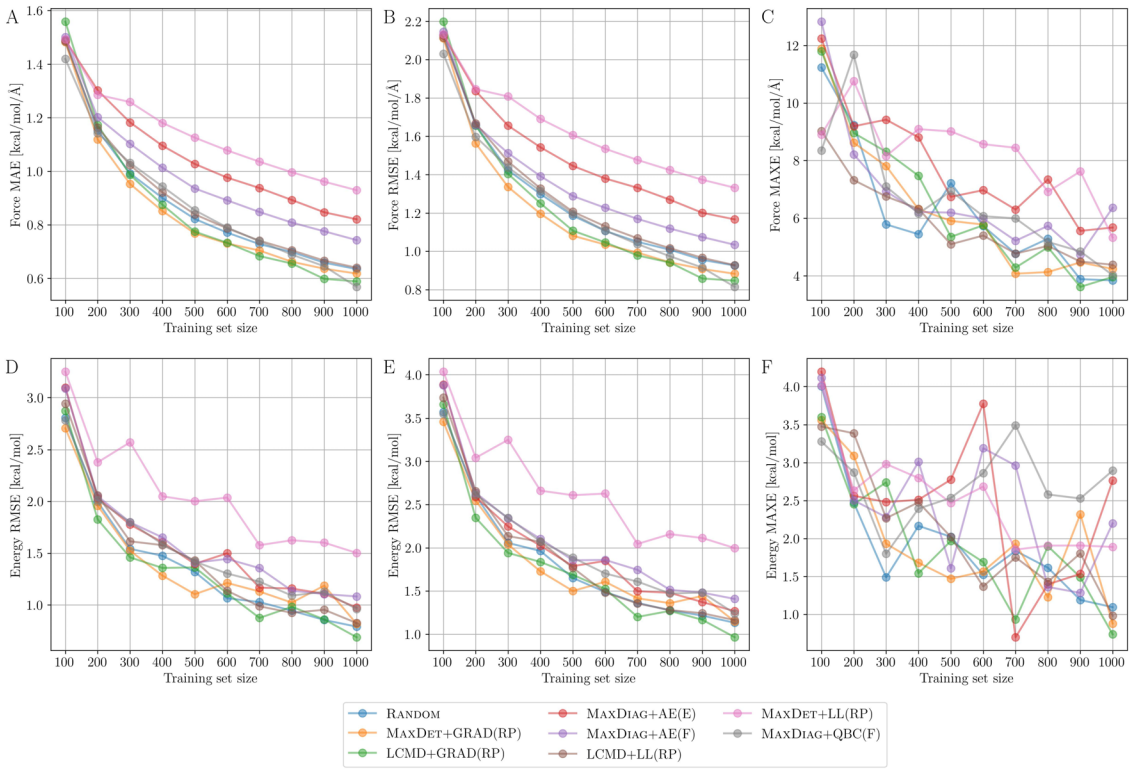

Figure 3.6: Learning curves for the salicylic acid molecule data from LiPS dataset. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

pendent testing, and the rest form the pool set for active learning selection. As shown in Figure 3.6, it is evident that LCMD+GRAP(RP) consistently surpasses other methods in terms of force MAE and RMSE, with the exception of MaxDiag+QBC(F) in the last iteration. The force MAE However, we point out that the superiority of MaxDiag+QBC(F) is not because it is a more effective active learning strategy. Instead, its advantage stems from utilizing an ensemble of five models for predictions. It is widely recognized that employing an ensemble can yield higher accuracy compared to a single model.[141, 142] We observed that LCMD+GRAP(RP) showed only a marginal improvement over Random in comparison to the MD17 cases. Meanwhile, both MaxDiag-AE(E) and MaxDiag-AE(F) methods lagged notably behind Random. We believe that the limited conformational space explored by a 50 ps AIMD simulation may be the reason. As a result, a batch size of 100 appears sufficient for the Random approach to sample a representative number of informative data points from the pool set. This leads to accuracy levels that are on par

with LCMD. In contrast, methods based on MaxDiag tend to sample data points over very short time intervals in this case, which can result in worse learning behaviors. We expect that batch active learning methods will be more crucial for the pool set with larger conformational spaces when using larger batch sizes. Additionally, the optimal batch size may vary depending on the specific chemical system under consideration.

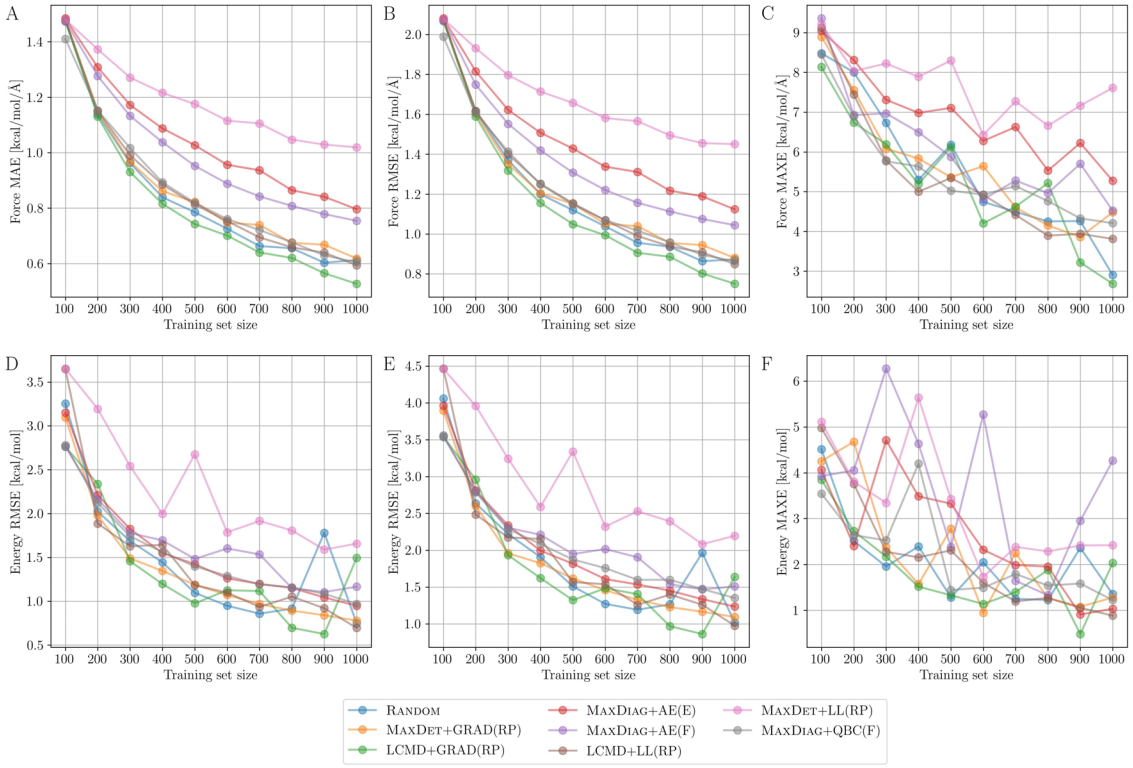

Figure 3.7: Learning curves for the salicylic acid molecule data from LiPO dataset. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

LiPO active learning tests: We have extended our tests to the more intricate system of molten glass, $Li_4P_2O_7$ (LiPO).[66] This system comprises 64 Li, 32 P, and 112 O atoms within a 10.58×13.96×16.08 cell. The dataset encompasses 25,000 MD frames, sourced from a 50 ps NVT AIMD simulation at 3000K, using a time step of 2 ps. Despite LiPO having a greater number of atoms and exhibiting higher levels of disorder compared to LiPS, we observed a notable similarity in their learning behaviors, both exhibiting significant gaps between the learning curves of MaxDiag-based methods and Random with respect to force MAE and RMSE. Consequently, we infer that the effectiveness of

batch active learning approaches is largely influenced by the conformational space of the pool set and the selected batch size, rather than the size and complexity of the chemical systems.

Based on our test results, several key insights emerge. Firstly, for pool sets with limited conformational space or when large batch sizes are used, we recommend avoiding the use of MaxDiag-based methods for data point selection. Secondly, the consistently superior performance of LCMD+GRAP(RP) across various test systems and tasks suggests it is a reliable choice for all scenarios. Finally, our analyses further reveal that the GRAD(RP) kernel consistently outperforms LL(RP) during active learning tests, emphasizing the crucial importance of selecting robust features that well represent atomic structures.

3.4 Uncertainty-aware simulation

Although MPNNs have shown outstanding performance in the sampled configurational space, they tend to perform poorly on out-of-distribution (OOD) data. In this context, the role of uncertainty estimation (UE) becomes crucial, ensuring the model predictions are always reliable during active learning iterations or production simulations. Wollschläger *et al.*[143] introduced several crucial criteria for the effective application of UE methods:

- Accuracy: Precision in simulations is of utmost importance. An effective UE method should be able to deliver reliable uncertainty metrics without compromising model accuracy.
- Speed: Ideally, a UE method should be optimized such that it introduces marginal computational overhead, especially in some computationally heavy tasks like MD simulations.
- Confidence–aware: It is crucial that the method can discern and notify when a particular atomic structure is outside the domain of training.

Ideal UE methods should meet all these criteria to effectively handle the diverse and intricate tasks presented in atomistic simulations. UE methods can be roughly categorized into two groups based on how the predictions are made: sampling-based and sampling-free methods. Sampling-based methods, such as the deep ensemble and Monte Carlo dropout, rely on the disagreements among multiple predictions to determine uncertainty. A greater variance in predictions corresponds to increased uncertainty, and vice versa. On the other hand, sampling-free methods generally utilize a single forward pass to uncertainty through the analysis of the distributions of learned features. In our workflow, the following UE methods are available for various atomistic simulations.

Deep ensemble is considered the gold standard solution for uncertainty estimation.[142] An ensemble usually comprises diverse models trained with varied architectures or initializations. When these models generate different predictions, measuring disagreements such as the standard deviation among them can provide an estimation of uncertainty. In the context of atomistic simulations, the standard deviation of energies or forces can serve as an indicator of the uncertainty associated with an atomic structure, as illustrated in Equation 3.14 and Equation 3.15. Although ensembles often improve *Accuracy* and fulfill $Confidence–aware$,[77, 144] they come at the cost of increased computational complexity, both in training and inference, thus fail at *Speed*.

Monte Carlo dropout (MCD) incorporates dropout into deep learning models, enabling the estimation of model uncertainty during predictions. By performing multiple forward passes with dropout, we can treat the collection of predictions as samples from a distribution, which captures the model uncertainty about its prediction for the input. Therefore, the formulations of energy and force uncertainties align with Equation 3.14 and Equation 3.15 as seen in the ensemble case. Since MC dropout involves deactivating neurons within a single model, it necessitates the training of only one model, thereby conserving substantial effort in the training process. Nevertheless, the random deactivation of neurons can occasionally undermine the predictive accuracy of the model. Furthermore, even though training is limited to a single model, inference still requires multiple forward passes, challenging the *Speed* criterion.

Mahalanobis distance quantifies the distance between a point and a distribution, taking into account the correlations of the data set and the scale of the features in different dimensions. Therefore, this method is very useful for detecting samples that are out of the distribution of the training data set. Formally, the Mahalanobis distance $d(\boldsymbol{S}_i, Q_S)$ between a structure $\boldsymbol{S}_i$ and a distribution of training set Q_S with mean μ_S and covariance matrix $\boldsymbol{\Sigma}_S$ is defined as:

$$d(\boldsymbol{S}_i, Q_S) = (\boldsymbol{S}_i - \mu_S)^T \boldsymbol{\Sigma}_S^{-1} (\boldsymbol{S}_i - \mu_S) \tag{3.29}$$

Clearly, the expression is equivalent to Equation 3.21 when a small noise is introduced to the covariance matrix $\boldsymbol{\Sigma}$ and the input is normalized to the training dataset. When employing the GNN base kernel, this approach can be seen as computing $k_{gnn\to avg\to gp}(\boldsymbol{S_i}, \boldsymbol{S_i})$, which is the diagonal element of kernel matrix $k_{gnn\to avg\to gp}(\boldsymbol{S}, \boldsymbol{S})$. Choosing a simple identity matrix as the covariance matrix translates to computing the Euclidean distance. We will demonstrate in subsequent tests the importance of the covariance matrix for reliable uncertainty estimation by comparing the Mahalanobis and Euclidean distances. It is worth noting that this method exclusively utilizes the features derived from a singular

model forward pass and leverages a precomputed covariance matrix to compute Mahalanobis distance. As a result, the model accuracy remains consistent with the original one, with only a marginal computational overhead introduced for evaluating the distance metrics.

Local Mahalanobis distance is different from Mahalanobis distance by reversing the order of sum and GP transformation, which can be represented as $k_{gnn\to gp\to avg}(\boldsymbol{S_i}, \boldsymbol{S_i})$. The corresponding Mahalanobis distance is then given by:

$$d(\boldsymbol{S}_i, Q_{\boldsymbol{x}}) = \sum_{x_i \in S} (\boldsymbol{x}_i - \mu_{\boldsymbol{x}})^T \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} (\boldsymbol{x}_i - \mu_{\boldsymbol{x}}) \tag{3.30}$$

An immediate advantage of this modification is that the resulting uncertainty scales to the size of atomic structures. By ensuring that uncertainty is proportional to the structural size, this method offers a refined uncertainty estimation for structures of varying system sizes.

We evaluated the accuracy of models on the MD17 dataset and the inference speed of various UE methods, as presented in Table 3.1 and Table A.2. The ensemble consists of five individual PaiNN models, each trained using different splits for training and validation. Both local Mahalanobis and Mahalanobis distance use the original single model for prediction. For MCD, we tested the results with the dropout ratio 0.01, 0.05, 0.10, and 0.20, here only 0.1 and 0.2 cases are reported in Table 3.1. In all cases, we use randomly selected 5000 structures for training, 1000 for validation, and 5000 for independent tests. Among all the UE methods, the ensemble consistently exhibited superior accuracy by leveraging predictions from various models. However, a notable decline in performance was observed when MCD was applied to the original model. From these results, we can find that only MCD fails at the accuracy criterion. Another crucial factor for UE methods is their speed. Remarkably, Mahalanobis-based UE methods are about five times faster than both the ensemble and MCD, yet they offer comparable accuracy to the ensemble. Therefore, This makes them especially suitable for running heavy simulations.

To evaluate the performance of these methods in *confidence–aware* criteria, we employ OOD detection based on the area under the receiver operating characteristic (AUC-ROC) curve. Specifically, we train a model on one molecule in MD17 and examine its ability to differentiate the remaining molecules using its uncertainty estimates. Ideally, the estimator should produce low uncertainties for the trained molecule and higher ones for the rest. Such behavior allows users to set a confidence threshold to trust model predictions when the uncertainty falls below this threshold. For performance evaluation, we calculate the area under the AUC-ROC curve of the uncertainty scores for both in-distribution (ID)

Table 3.1: MAE of PaiNN on MD17 with different UE methods (energies in kcal mol^{-1}, forces in kcal mol^{-1} Å^{-1})

		Ensemble	**Mahalanobis**	**MCD (p=0.1)**	**MCD (p=0.2)**
aspirin	Energy	**0.123**	0.129	4.324	9.932
	Forces	**0.098**	0.160	1.385	2.201
azobenzene	Energy	**0.137**	0.138	5.004	11.470
	Forces	**0.043**	0.063	1.134	1.821
ethanol	Energy	**0.051**	0.052	0.779	1.498
	Forces	**0.053**	0.088	0.707	1.101
malonaldehyde	Energy	**0.074**	**0.074**	0.855	1.749
	Forces	**0.080**	0.134	0.941	1.443
naphthalene	Energy	0.113	**0.112**	3.354	7.686
	Forces	**0.032**	0.043	1.012	1.603
paracetamol	Energy	**0.113**	0.118	3.654	8.495
	Forces	**0.068**	0.115	1.232	1.957
salicylic acid	Energy	0.107	**0.106**	2.938	6.734
	Forces	**0.055**	0.085	1.150	1.812
toluene	Energy	**0.092**	0.093	2.439	5.554
	Forces	**0.035**	0.050	1.005	1.572
uracil	Energy	0.105	**0.103**	1.629	3.624
	Forces	**0.040**	0.063	1.039	1.615

and OOD data. A score approaching 1 signifies a better ability to differentiate OOD data using the specific UE method. Figure 3.8 shows the heatmap of the AUC-ROC score for each pairwise combination of molecules obtained with different UE methods. The rows show the molecule that the model is trained on and the off-diagonal columns are the respective OOD sample. On the diagonal, we expect a score of 0.5 while a score of 1 is optimal on the off-diagonals. We found that the ensemble exhibited perfect separation between ID and OOD samples. In contrast, another sampling-based method MCD did not show a satisfactory ability for separating some pairs. Local Mahalanobis distance has shown comparable performance compared to the ensemble while using much less computational cost. From Figure 3.8d it is evident that using the local representation of atomic environments to calculate the covariance matrix can significantly improve the *confidence–aware* ability of Mahalanobis distance. The order of GP transformation in the kernels does matter. We further investigated the influence of using different kernels as shown in Figure A.9. It is clearly seen that only the GNN kernel exhibited high AUC-ROC scores on off-diagonals. Although full gradient and last-layer gradient kernels have shown

exceptional performance in representing atomic structures via t-SNE, they crucially failed at differentiating the molecules using Mahalanobis distance. (influence of covariance matrix, influence of dropout ratio). If using an identity matrix as the covariance matrix, this corresponds to Euclidean distance. We demonstrated in Figure A.10 that Euclidean distance is not able to achieve satisfactory OOD detection performance, without the use of a covariance matrix.

Our workflow provides all the mentioned UE methods described above, the users can choose suitable UE methods for their specific applications. For computationally heavy applications, local Mahalanobis distance is recommended, while for applications that need more reliable uncertainty estimations, the ensemble is recommended.

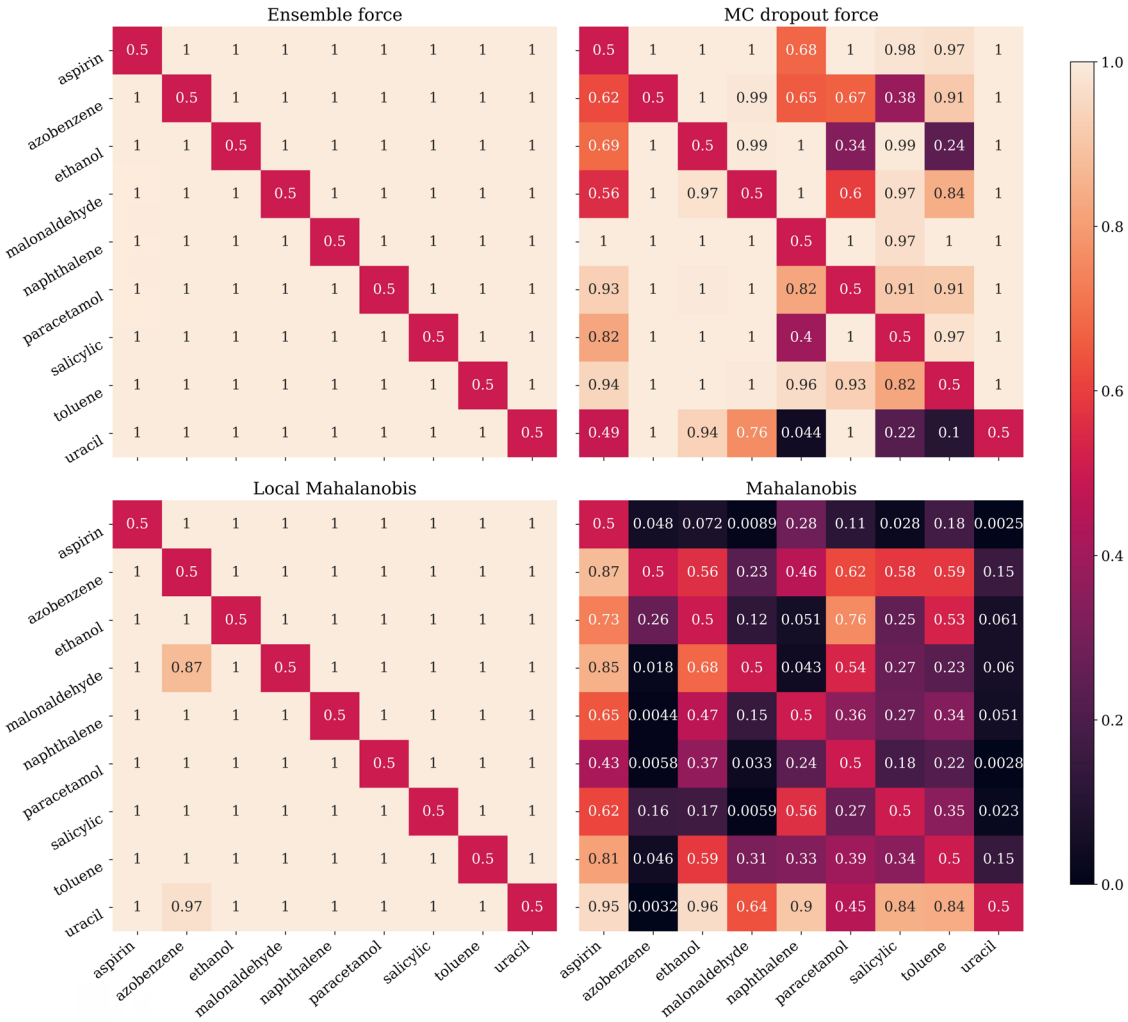

Figure 3.8: Heatmap displays the AUC-ROC values from SchNet on MD17. Each row represents a separate model, trained on the molecule listed to the left, and tested against all other molecules.

3.5 Automated workflow

Leveraging powerful selection methods and efficient UE techniques, we have established a comprehensive, automated active learning workflow. It is imperative to note that the different stages in this workflow necessitate varying resource allocations. For instance, training and machine learning-driven simulations predominantly utilize GPUs, whereas the annotation of informative batches often depends on DFT codes optimized for CPUs. Besides, these tasks are structured in a fixed sequential order. This means that one first needs to ensure the successful completion of preceding tasks before initiating certain tasks. Recognizing and adhering to these dependencies is vital to ensure the workflow operates smoothly and efficiently. To manage job assignments across diverse hardware and inspect their execution states, we employ `myqueue`,[123] a cluster job manager, to assign jobs on different hard devices and to manage these jobs. Furthermore, given the need for specifying diverse hyperparameters throughout different phases in the workflow, we have adopted the Hydra[145] configuration framework, which allows the building of hierarchical YAML configurations. In addition, we have integrated PyTorch Lightning [146] to streamline the model training process. In the sections that follow, we demonstrate how this workflow can be adeptly employed to autonomously construct the MLIPs.

Figure 3.9a displays the predefined hierarchical YAML configurations in the code. We provide a suite of default configurations for various components within the workflow, facilitating a vast number of combinations and extensive experimental runs via the command line interface (CLI). These configuration files are organized in an object-oriented manner, with each subdirectory containing files for different applications. files in an object-oriented way. Each subdirectory contains some choices for different applications. Situated in the top-level directory within the `configs` folder are four principal YAML files: `train.yaml`, `simulate.yaml`, `select.yaml`, and `label.yaml`, each corresponding to a respective phase in the active learning workflow. They define default hyperparameters, but users have the flexibility to adjust them via the CLI as needed. For instance, to train a model, one might use:

```
gnntrain model/representation=nequip data/datapath=water.traj
```

Additionally, users can craft their configuration files outside the default `configs` directory. By specifying `cfg=custom.yaml`, where `custom.yaml` is the custom configuration, one can easily employ it for desired experiments. Integrating the configuration files for each separate job leads to the overall configuration file `workflow.yaml`, which encapsulates hyperparameters for all phases in the workflow. Figure 3.9b showcases a user-defined configuration tailored for running active learning iterations. Parameters from this file will

```
|-- data
|   |-- custom.yaml
|-- labelling
|   |-- custom.yaml
|   |-- gpaw.yaml
|   |-- qe.yaml
|   |-- vasp.yaml
|-- model
|   |-- representation
|   |   |-- mace.yaml
|   |   |-- nequip.yaml
|   |   |-- painn.yaml
|   |-- nnp.yaml
|-- selection
|   |-- default_selection.yaml
|-- simulation
|   |-- custom.yaml
|   |-- mc.yaml
|   |-- md.yaml
|   |-- neb.yaml
|-- task
|   |-- optimizer
|   |   |-- adam.yaml
|   |   |-- adam_amsgrad.yaml
|   |-- scheduler
|   |   |-- exponential.yaml
|   |   |-- reduce_on_plateau.yaml
|   |-- default_task.yaml
|-- trainer
|   |-- default_trainer.yaml
|-- __init__.py
|-- label.yaml
|-- select.yaml
|-- simulate.yaml
|-- train.yaml
|-- workflow.yaml
```

(a)

```yaml
defaults:
  - model/representation: painn
  - task/optimizer: adam
  - task/scheduler: reduce_on_plateau
  - simulation: md
  - labelling: vasp

data:
  datapath: ./water_dft.traj
  cutoff: 5.0
  batch_size: 16
  num_train: 2000
  num_val: 1000
  atomic_energies: auto
  atomwise_normalization: True

model:
  representation:
    num_interactions: 3
    num_features: 64

task:
  scheduler_monitor: val_loss
  optimizer:
    lr: 0.005
  scheduler:
    factor: 0.5

simulation:
  uncertainty: local_mahalanobis
  simulator: md
  params:
    load_traj: ./water_dft.traj
    max_steps: 1000000

selection:
  kernel: full-g
  selection: lcmd_greedy
  n_random_features: 500
  batch_size: 200

labelling:
  dft_code: vasp
  num_jobs: 4
```

(b)

Figure 3.9: Main figure caption for code listings

override the defaults set in `workflow.yaml`. This design empowers users to effortlessly manage and customize their tasks, facilitating the construction of diverse MLIPs. For a more in-depth exploration and a deeper understanding of our configuration system and its functionalities, readers are encouraged to visit our codebase. This resource offers comprehensive documentation and examples, ensuring clarity and ease of use for both newcomers and experienced users.

3.6 Conclusions

In this study, we tackled the existing challenges in the development and application of machine learning interatomic potentials for atomistic simulations. Although the power of MLIPs has been previously verified, challenges such as efficient data collection, reliable tools for model confidence, and intricate procedures persisted. At the forefront of our contributions is the introduction of `CURATOR`, a comprehensive workflow that seamlessly integrates advanced active learning algorithms and reliable uncertainty estimation techniques for improving data acquisition efficiency and ensuring reliable production simulations.

The workflow encompasses state-of-the-art graph neural network models for accurate atomistic modelling. We re-implemented the gradient calculation and significantly accelerated the speed for stress calculation. We emphasized the importance of batch active learning in the collection of data sets. We show that our incorporation of batch active learning strategies effectively enables much-improved data acquisition efficiency. When evaluated on multiple benchmark datasets, our specific batch active learning strategies consistently outperformed others across various systems. This highlights their potential in significantly minimizing human efforts and computational expenses in generating MLIPs. To ensure the reliable application of trained models, we incorporated several uncertainty estimation methods into the workflow and compared their performance in terms of speed, accuracy, and confidence-awareness. The test results demonstrated that the Mahalanobis distance can serve as a fast and reliable UE method, with only fraction of the cost of ensembles while demonstrated comparable performance in accuracy and confidence-awareness.

The workflow has been made fully autonomous by combining the previously mentioned elements. By merging the Hydra-style configuration framework, Pytorch Lightning, and the robust task scheduler `myqueue`, the system is both functional and user-friendly, catering to both beginners and experts.

CHAPTER 4

Oxygen reduction at confined Au(100)-water interface

This chapter is based on the case study in paper II – "Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces". The paper is included in this thesis and corresponding supplementary information can be found in Appendix B.

4.1 Introduction

Over the past several decades, density functional theory (DFT) calculations have been extensively used for developing novel electrocatalysts towards oxygen reduction reaction (ORR) by taking advantage of well-developed theoretical methods[8, 9, 10, 7, 3] (e.g., free energy diagrams, volcano plots, and d-band theory) for predicting catalytic activities. Nevertheless, most of these calculations oversimplify the operating conditions of catalysts by either modelling the liquid waters at the electrolyte-electrode interface as static water layers,[11, 12, 13, 14] implicitly representing them via dielectric continuum models,[15, 16, 17] or even absolutely ignoring the effect of the solvents.[19, 20, 21, 22] These limitations may lead to erroneous evaluation of activity trends of catalysts as compared to experiments, for example, the oxygen reduction reaction on gold in alkaline electrolytes.[26, 27] Including solvent molecules for electrolyte-electrode interface simulations and investigating their dynamical effects could offer us a better understanding towards the reaction mechanisms of ORR and may resolve the conflicts between theoretical calculations and experiments.

While *ab initio* molecular dynamics (AIMD) is capable of capturing the dynamics of liquid water, it is prohibitively expensive for large length-scale and long time-scale simulations. For instance, the time-averaged metrics (e.g., energy and temperature) of AIMD simu-

lations can differ significantly if started from different initial configurations, while these discrepancies could be greatly mitigated if the model system is equilibrated and sampled from long enough trajectories.[37, 147, 40] The prohibitive computational cost severely limits the equilibration and sampling time scales of AIMD to only a few ps, which may significantly impair the reliability of such studies.[37, 148, 149, 150, 151, 152, 36, 35]

Recently, advances in machine learning are making great impacts in aiding the design and discovery of transitional metal based catalysts.[153, 39] By learning from data, machine learning tools can make fast predictions to find target catalysts and provide valuable insights into the nature of the reaction, which enable high-throughput screening of catalysts from a broad chemical space and automated catalyst design.[154, 155, 156] In particular, neural network potentials (NNP) have shown great promise at fitting the potential energy surface (PES) of reactive model systems by training on reference configurations that well describe the representative atomic environments.[157, 158, 159, 160] This approach could speed up MD simulations by several orders of magnitude whilst retaining the accuracy comparable to AIMD, which enables us to considerably extend the time scale and length scale of MD simulations without compromising accuracy. Initially proposed architectures of neural network potentials learned the force field by leveraging handcrafted features based on distance and angle information to capture the characteristics of local atomic environments.[49, 50, 161] Behler-Parrinello neural network potential is the first example in which the Cartesian coordinates of atoms are transformed to rotational and translational invariant atomic-centered symmetry functions.[49, 50] Recent advances in graph neural networks (GNNs) for molecule graphs have made it possible to learn representative features from the atomic structure via a graph message-passing scheme.[62, 58, 110, 66, 65] State-of-the-art GNN models leverage rotation equivariant representation of node features (i.e., features of atomic environments) to provide more accurate force predictions, which can be essential in MD simulations.[77, 66, 65] In spite of numerous novel machine learning methods for fitting PES and MD simulations driven by NNPs,[43, 40, 41, 162, 147, 163] there are few studies on simulating nonequilibrium dynamics and reactions. We have yet to find out any study performing sampling of rare events that govern chemical reactions with NNPs.[35, 164, 165] Taking ORR as an example, although NNPs can significantly accelerate MD simulations, the time scale of reactive simulation of ORR is still inaccessible, not to mention the complex ambient conditions of the catalysts. Due to the rapid development of enhanced sampling techniques like metadynamics[113, 166] (MetaD) high accuracy sampling of PES has been possible for such rare events. We envision that combining enhanced sampling methods together with high-fidelity NNPs, can enable full simulation of slow chemical reactions in atomic scale within affordable computational cost.

In this chapter, we present the full atomic simulation of ORR at Au(100)-water interface done using metadynamics simulations accelerated by equivariant graph neural network potentials.[110] The gold electrode has been extensively studied as the ORR electrocatalyst, while its exceptional activity, especially in alkaline media, is still not well-explained.[26, 27, 28, 29] This case could well demonstrate the power of our proposed simulation paradigm towards modeling of rare chemical reactions at the solid-liquid interfaces. Compared to non-reactive MD performed with NNPs, a major challenge of simulating rare events like ORR is to ensure that the machine learning model encompasses the vast configurational space far away from equilibrium. This requires adaptive sampling of representative reference structures from MD and MetaD simulations, particularly transition states that are rarely visited. In addition, quantitatively evaluating the reliability of NNPs for describing PES in the configurational space of interest is also indispensable. Here we adopt an active learning approach based on the CUR matrix decomposition[167, 168] to sample representative reference structures from MD and MetaD simulations. This method enables us to representatively sample the vast configurational spaces of ORR at the solid-liquid interface with minimal human intervention and significantly reduced computational cost. Our MD and MetaD simulations are uncertainty aware, demonstrating robust and reliable modeling of full atomic simulation of ORR with NNPs.

4.2 Computational methodology

4.2.1 Active learning framework

Our neural network potentials are constructed based on an active learning framework utilizing CUR decomposition based selective sampling as demonstrated in Figure 4.1. First, an initial dataset was generated by selectively sampling reference structures from several AIMD trajectories of Au(100)-water interfaces. Multiple interface structures with different numbers of hydroxyl or oxygen molecules are considered to ensure the diversity and versatility of the training dataset and to further study the impact of adsorbates on the dynamics of solvents. The initial AIMD trajectories contain several hundreds of thousands of configurations. Using all of them would make the training of NNPs very slow. Many structures are similar, thus models do not capture new correlations when all of those are used simultaneously. Therefore, it is crucial to sample only those structures that are representative and informative from these trajectories. We first select structures from AIMD trajectories one in every 50 MD steps, reducing the number of candidate configurations to several tens of thousands. Then the CUR matrix decomposition method[168, 167] is employed to further refine the training dataset without losing too much information.

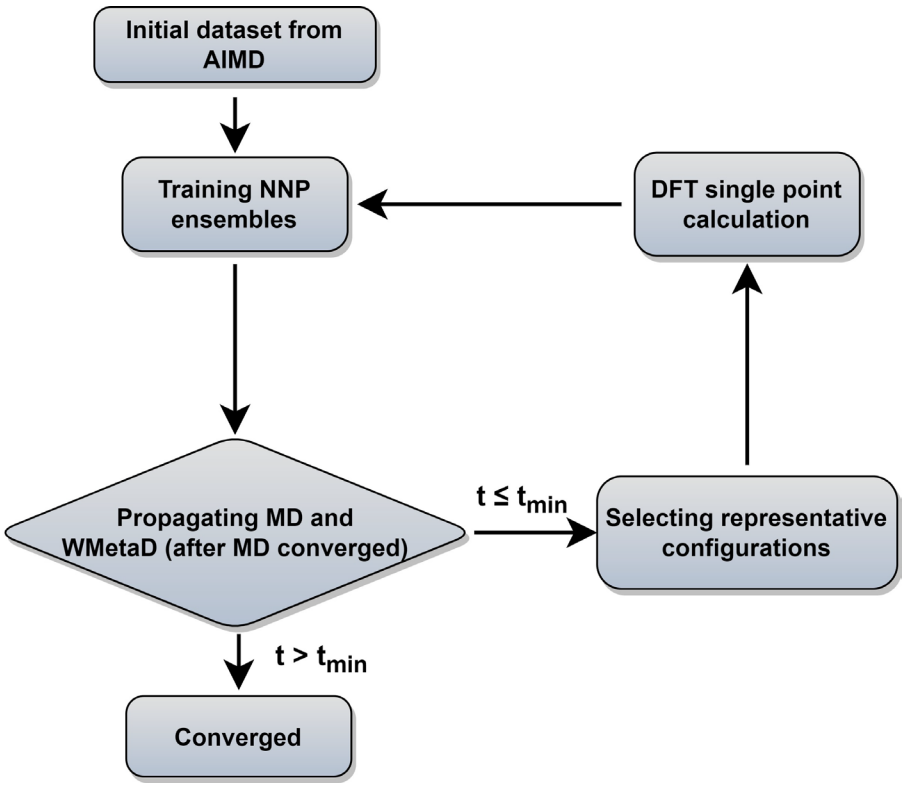

Figure 4.1: Active learning procedures used to train the neural network potentials

Given a $N \times M$ data matrix X with its rows corresponding to N atoms and its columns corresponding to M fingerprints, the objective of CUR is to minimize the information loss after ruling out some rows and columns, meanwhile minimizing the number of rows and columns to be selected. We also add an extra term in the objective function to maximize the euclidean distance between different atomic environments to ensure the diversity of sampled structures. In this process, the importance of each row and column in the data matrix can be evaluated, and the representative configurations and fingerprints can be jointly sampled. Here the fingerprints of atoms in candidate structures are described by Behler-Parrinello symmetry functions,[49, 50] which have been extensively used for fitting PES for solid-liquid interface systems.[43, 40, 41, 162, 147, 163] CUR matrix decomposition also provides an efficient way for automatically selecting symmetry function parameters that are typically non-trivial.

An ensemble of neural network potentials (NNP) was then trained on the initial dataset with 5000 reference structures after CUR selection. In order to extend their capability of exploring larger configurational space, the trained NNPs are updated adaptively by the following steps: i) propagating MD trajectories with trained NNP ensemble as energy/-force calculator; ii) selecting representative reference structures from MD trajectories by using CUR decomposition; iii) calculating these selected new data points with DFT; iv) re-

training NNPs with the expanded training dataset. Instead of propagating MD using only one NNP, we choose to combine all the trained NNPs together for the prediction of energy and forces. This strategy not only improves the predictive accuracy of our model but also provides a practical way to quantify if the model is still confident enough in the configurational space of interest.The quantification is achieved by evaluating the energy uncertainty and forces uncertainty for every step via calculating the variance of NNPs during the MD simulation. Figure B.1 compares the calculated uncertainty and true prediction error, indicating that uncertainty is an excellent indicator for true model error. Based on query-by-committee method, which has been widely used in active learning,[169, 77, 170] the configurations with relatively large uncertainty are collected to reduce the number of candidate configurations. Subsequently, the obtained structures are further sub-sampled by CUR decomposition for DFT evaluation. By adding these carefully chosen new data in the training set, we constantly improve the model prediction for new configurational space visited by MD simulations. Combining this strategy and CUR matrix decomposition significantly reduces the number of candidate structures and ensure the diversity of structures in a sampled batch. The simulations are stopped if the uncertainties are too large or too many structures with large uncertainties are collected. The iterative training of the NNP ensemble stops once all the MD simulations can be propagated to more than t_{min} steps, where t_{min} is selected as 5 ns to ensure the model systems are properly equilibrated and all the dynamical events are fully captured.[43, 40, 41] To further investigate the ORR kinetics at the gold-water interface, the iterative training procedures are repeated in the case of MetaD simulations. Notably, as the transition states are rarely visited during MetaD runs, it is critical to include enough such configurations into our training dataset and validate our MetaD simulations via uncertainty quantification.

4.2.2 Training details

The NNP ensemble we used for production consists of five neural network potentials with different architectures of polarizable atom interaction neural network (PaiNN) model.[110] In this model, all the atoms in a given configuration are treated as nodes in a graph and the information of their connections will be collected and processed by a message function, which will then be passed to an update function for updating node features. After several message passing iterations, the node features will be used as the input of a multilayer perceptron to get its atomic energy or other scalar properties. By summing up the atomic energies of a given structure, we can get its potential energy and forces by calculating the negative derivatives of energy to atomic coordinates. The model can automatically learn the relationship between chemical properties and the positions of atoms by optimizing

several hundreds of thousands of model parameters in message and update layers. In contrast, only a few hyperparameters need to be selected (the size of node features, the number of message passing layers, loss ratio of energy and forces, and the cutoff radius for collecting distance information of atoms, etc.), avoiding the need to be manually selected and test of handcrafted features like Behler-Parrinello symmetry functions.[49, 50] Besides, the model uses both scalar and vector node features to realize rotational equivariance of directional information (e.g., forces) in the graph, providing better prediction of forces.

Table 4.1: A summary of test error metrics of neural network potentials

Model	Node size	Layers	Energy error (meV/atom)		Forces error (meV/Å)			
			MAE	RMSE	l_2 MAE	l_2 RMSE	MAE	RMSE
NNP1	96	3	0.6	1.3	32.8	46.4	16.3	26.8
NNP2	112	3	0.5	1.3	31.3	45.0	15.5	26.0
NNP3	128	3	0.8	1.4	28.9	43.7	14.3	25.2
NNP4	128	4	0.4	1.2	25.4	39.3	12.6	22.7
NNP5	144	3	0.5	1.2	27.0	40.8	13.4	23.5
Ensemble	-	-	0.7	1.4	25.3	38.8	12.6	22.4

Table 4.1 reports the architectures of five models constituting our NNP ensemble and their error metrics after training on the same dataset for up to 1 000 000 steps. These models use different node feature sizes and the number of message-passing layers to induce model diversity, while their cutoff radius are all set to 5 Å. Both the model training and following production MD (MetaD) simulations are conducted on an NVIDIA GeForce RTX 3090 GPU with float32 precision. The weight parameters in these models are randomly initialized and then optimized on the same data split using stochastic gradient descent to minimize the mean square error (MSE) loss, which can be expressed as:

$$\mathcal{L} = \frac{1-\lambda}{N}\sum_{i=1}^{N}(E_i - \hat{E}_i)^2 + \frac{1-\lambda}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}(F_i^{jk} - \hat{F_i^{jk}})^2 \tag{4.1}$$

where N is the number of configurations, M is the number of atoms in a configuration, and λ is force weight that controls the relative importance between energy and force loss. Here the force weight is set to 0.99 as our tests show that using a relatively large force weight can well improve the forces prediction while only slightly undermines the precision of energy prediction. Our model parameters are trained by the Adam optimizer[171] as implemented in PyTorch[172] with an initial learning rate of 0.0001, the default parameters β_1=0.9, β_2=0.999, and the batch size of 16. An exponential decay learning rate scheduler with the coefficient of 0.96 is used to adjust learning rate for every 100 000 learning steps. The dataset is split into a training set (90%) and a validation set (10%), where

the validation set is used for early stopping when the error of forces is small enough. Note that several different error metrics are used to evaluate the performance of trained model, including mean absolute error (MAE) and root mean squared error (RMSE) for both energy and force predictions. These error metrics can be expressed as follows:

$$E_{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mid E_i - \hat{E}_i \mid \tag{4.2}$$

$$E_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(E_i - \hat{E}_i)^2} \tag{4.3}$$

$$F_{MAE} = \frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3} \mid F_i^{jk} - \hat{F_i^{jk}} \mid \tag{4.4}$$

$$F_{RMSE} = \sqrt{\frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}(F_i^{jk} - \hat{F_i^{jk}})^2} \tag{4.5}$$

4.2.3 AIMD and single point DFT calculations

The Au(100)-water interface is modelled as 30 H_2O molecules on top of a (3×3) tetragonal Au(100) surface with four atomic layers, which will be denoted as Au(100)-30H_2O hereafter. A vacuum layer larger than 15 Å is perpendicularly added into the model to eliminate the spurious interaction between periodic images. In order to simulate the interface with ORR intermediates, we also consider structures with one and two hydroxyls by removing the hydrogen atoms from water molecules near the slab, and structure with one oxygen molecule on top of Au(100) slab. These structures are denoted as Au(100)-1OH/29H_2O, Au(100)-2OH/28H_2O, and Au(100)-1O2/30H_2O, respectively. Constant temperature MD simulations are then performed in VASP[94, 95, 96, 97] by using these initial configurations with the timestep of 0.5 fs and the temperature is kept around 350K with the Nosé–Hoover thermostat.[106] The bottom two layers are kept fixed during MD run for all model systems. 50 ps, 15 ps, 15 ps, and 15 ps MD simulations are conducted for Au(100)-30H_2O, Au(100)-1OH/29H_2O, Au(100)-2OH/28H_2O, and Au(100)-1O2/30H_2O, respectively. The reason for running shorter MD simulations on the model systems with adsorbates is that their most local structures are similar to Au(100)-30H_2O system. Density functional calculations are used to calculate the potential energy and the forces for propagating AIMD and labeling representative configurations sampled by active learning. We employ an energy cutoff of 350 eV for plane-wave basis expansion and a $2 \times 2 \times 1$ Monkhorst-Pack k-grid for Brillouin zone sampling.[173] The exchange-correlation effects are approximated by PBE functional combined with the D3 Van der Waals correction.[90, 99]

4.2.4 Production MD simulations

The production MD simulations driven by the NNP ensemble have been performed using the MD engine of Atomic Simulation Environment (ASE) python library.[98] The simulation box in AIMD is too small to accommodate more adsorbates and to simulate the full reaction. Furthermore, previous studies also demonstrated that notable noise in the structural properties of the model systems could be observed when using small cell sizes.[43, 174] Considering both effects and the increased computational cost for MD and labelling, we constructed a larger model with 59 H_2O molecules on top of a (4×4) tetragonal Au(100) surface with four atomic layers, on which more adsorbates can be accommodated. With the presence of one to six *OH, corresponding hydrogen atoms are removed at the interface, producing the interface structures that could be denoted as Au(100)-1OH/58H_2O, Au(100)-2OH/57H_2O, Au(100)-3OH/56H_2O, Au(100)-4OH/55H_2O, Au(100)-5OH/54H_2O, and Au(100)-6OH/53H_2O, respectively. In order to investigate the kinetics of ORR, the initial state structure Au(100)-1O_2/57H_2O is also built by removing two H_2O molecules and placing a O_2 molecule on top of Au(100). The momenta of model systems is initiated by a Maxwell–Boltzmann distribution with the temperature set to 350 K. The MD simulations are propagated for 5 ns by Langevin dynamics with the target temperature of 350 K, the timestep of 0.25 fs, and the friction coefficient of 0.02. It is noteworthy that a smaller time step is selected for production as it can help the MD simulations reach longer time scale with smaller uncertainty. The uncertainties of frames in MD simulations are quantified as the variance and standard deviation (SD) of model outputs:

$$E_{var} = \frac{1}{N}\sum_{i=1}^{N}(E_i - \overline{E_i})^2 \tag{4.6}$$

$$F_{var} = \frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}(F_i^{jk} - \overline{F_i^{jk}})^2 \tag{4.7}$$

$$F_{sd} = \frac{1}{3M}\sum_{j=1}^{M}\sum_{k=1}^{3}\sqrt{\sum_{i=1}^{N}(F_i^{jk} - \overline{F_i^{jk}})^2} \tag{4.8}$$

where N is the number of models in the ensemble, M is the number of atoms in a frame, $\bar{E}$ and $\bar{F}$ are the average predicted energy and force, respectively. In order to ensure the reliability of MD results, the simulations will stop if F_{sd} is larger than 0.5 eV/Å or more than 2000 structures with F_{sd} larger than 0.05 ev/Å are collected.

4.2.5 Metadynamics simulations

Following the method in Ref.[37], we calculated the formation energy of *OH as the internal energy of the Au(100)-n_{OH}OH/(59-n_{OH})H_2O interface structure, plus the internal

energy of gas phase $n_{\mathrm{OH}}/2$ H_2 molecules, minus the internal energy of the Au(100)-59H_2O interface structure.

$$\begin{aligned}\Delta E =&\langle E_{\mathrm{Au(100)}-n_{\mathrm{OH}}\mathrm{OH}/(59-n_{\mathrm{OH}})\mathrm{H_2O}}\rangle_t + \frac{n_{\mathrm{OH}}}{2}(E_{\mathrm{H_2}} + \frac{3}{2}k_BT)\\ &- \langle E_{\mathrm{Au(100)}-59\mathrm{H_2O}}\rangle_t\end{aligned} \tag{4.9}$$

The internal energy of interface structures are calculated as the time averaged potential energy plus kinetic energy. And the internal energy of H_2 gas molecule is calculated as the potential energy plus $3/2k_BT$ because their center-of-mass motions are not included in the MD simulations. Likewise, the adsorption energy of O_2 are calculated as follows.

$$\begin{aligned}E_{ads} =&\langle E_{\mathrm{Au(100)}-1\mathrm{O_2}/57\mathrm{H_2O}}\rangle_t + \frac{1}{2}(E_{\mathrm{O_2}} + \frac{3}{2}k_BT)\\ &- \langle E_{\mathrm{Au(100)}-57\mathrm{H_2O}}\rangle_t\end{aligned} \tag{4.10}$$

In this chapter, all the enhanced sampling simulations are performed with the well-tempered version of metadynamics.[175] The production metadynamics simulations are propagated by Langevin dynamics for 2.5 ns in ASE. The calculation of collective variables and bias potential of metadynamics is achieved by PLUMED[121, 122, 117] which is interfaced to ASE. To construct the path CVs as described in the main text, the $\mathrm{Au(100)}-1\mathrm{O_2}/57\mathrm{H_2O}$ and $\mathrm{Au(100)}-4\mathrm{OH}/55\mathrm{H_2O}$ interface structures are selected as two reference structures. And the coordination numbers $C_{\mathrm{O_2-O}}$ and $C_{\mathrm{O_2-H}}$ are used to define the configurational space of the path. The corresponding equations and parameters for calculating $C_{\mathrm{O_2-O}}$ and $C_{\mathrm{O_2-H}}$ are shown in Table 4.2

Table 4.2: Parameters for calculating coordination numbers of O_2

CV	Definition	Parameters
C_{O_2-O}	$C_{O_2-O} = \sum_{i\in O_2} \frac{1-\left(\frac{r_{i,O}-d_0}{r_0}\right)^n}{1-\left(\frac{r_{i,O}-d_0}{r_0}\right)^m}$	$r_{i,O}$: O_i and O distance $r_0 = 1.8, d_0 = 0, n = 6, m = 12$
C_{O_2-H}	$C_{O_2-H} = \sum_{i\in O_2} \frac{1-\left(\frac{r_{i,H}-d_0}{r_0}\right)^n}{1-\left(\frac{r_{i,H}-d_0}{r_0}\right)^m}$	$r_{i,H}$: O_i and H distance $r_0 = 1.5, d_0 = 0, n = 8, m = 16$

With the defined path, the progress along the path s and the distance from the path z

can be computed as:

$$s = \frac{\sum_{i=1}^{N} i e^{-\lambda \|X - X_i\|^2}}{\sum_{i=1}^{N} e^{-\lambda \|X - X_i\|^2}} \tag{4.11}$$

$$z = -\frac{1}{\lambda} \ln \left[\sum_{i=1}^{N} e^{-\lambda \|X - X_i\|^2} \right] \tag{4.12}$$

where N is the number of reference structures, X is the structure described by $C_{\mathrm{O_2-O}}$ and $C_{\mathrm{O_2-H}}$. The parameter λ is selected as 0.25. The Gaussians adopted have an initial height of 0.1 eV and the width of 0.05 and 0.1 for s and d collective variables, respectively. The metadynamics are carried out at 350 K, employing a bias factor of 5 and desposition rate of 125 fs (every 500 steps). For every metadynamics run (except for O_2 migration as O_2 is metastable in bulk water), the system is first equilibrated for 0.5 ns.

4.3 Results and discussions

4.3.1 Validation of models

Following the active learning framework, we have obtained a final dataset with 18731 configurations. Figure B.2 shows the learning curves of our NNPs trained on the final dataset, and Table B.2 reports the detailed error metrics of best models on the validation set. It is remarkable that our NNPs exhibit exceptional accuracy towards the prediction of energy and forces, where the mean absolute errors (MAEs) of energy range between 0.4 and 0.8 meV/atom, and the MAEs of forces between 12.6 and 16.3 meV/Å. To illustrate the performance of our models on different interface structures, we also report the composition of the final dataset and corresponding error metrics for different structures as shown in Table B.3. The precision of force predictions for each species in our research system is also shown in Figure 4.2a, indicating a close numerical agreement with DFT results. All these results suggested that the trained NNPs can provide accurate energy and force predictions for different structures across the ORR configurational space in the production MD simulations. Table B.4 exhibits the comparison of model performance in terms of energy and force predictions between our model and other studies for complex systems, illustrating that our model outperforms most of these studies, especially force predictions.[111, 176, 110, 66, 177, 178] The role of accurate force prediction is emphasized in our study since that is critical in MD simulations. The performance of the trained NNP ensemble is further validated in terms of its ability to reproduce structural properties of AIMD trajectories. Figure 4.2b shows the match of radial distribution functions

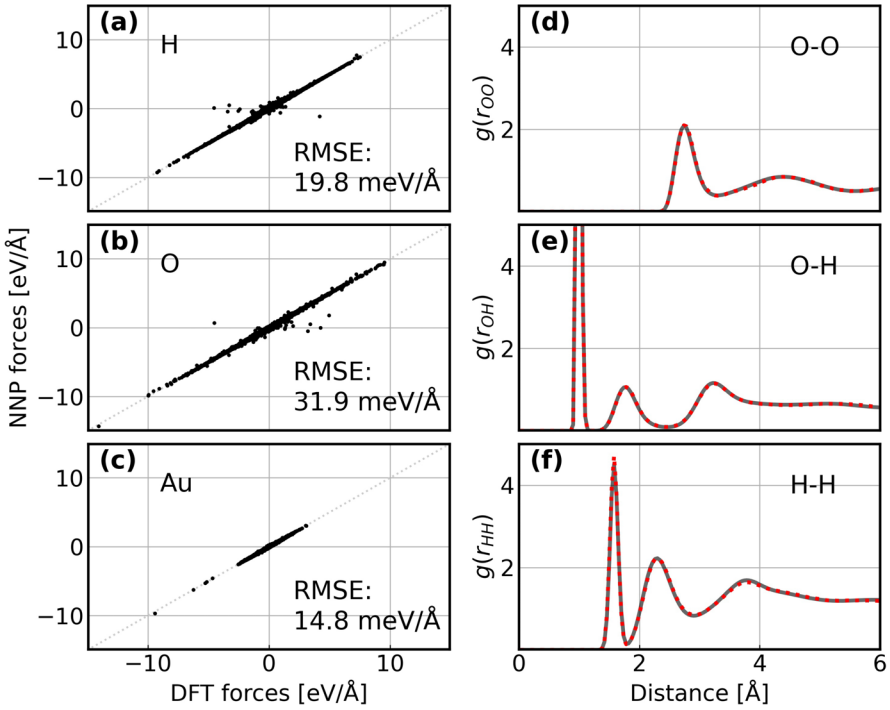

Figure 4.2: (a-c) Comparison between forces derived from DFT calculations and NNPs predicted forces for H, O, and Au. The RMSE of forces for each element are denoted inside. (d-f) Comparison between RDFs of obtained from AIMD simulations and NNPs MD simulations on Au(100)-water interface structure. The red points denote RDFs generated by AIMD calculation and the grey solid line denotes RDFs generated by NNPs calculation.

(RDFs) of all involved species in the case of Au(100)-water interface (without hydroxyls or oxygen molecules). Apparently, the RDFs generated by NNP MD simulations (solid black line) exhibit an excellent agreement with AIMD results (red points), indicating that the NNP ensemble captures the structural arrangement of the gold-water interface well. Apart from validating NNPs with existing dataset, a more important assessment for the quality of NNPs is their application domain, which can be confirmed by uncertainty measurements. Concretely, the MD runs should be ergodic to ensure the reliability of information derived from them, which indicates that all energetically relevant states must be sampled and within the manifold accessible by NNPs. For all MD simulations in this study, we not only sample the properties of interest along long-time scale MD simulations but also present the uncertainties of all steps by calculating the variance of NNPs. The low forces uncertainty of MD simulations for different interface structures verify the robustness and reliability of the trained NNP ensemble in the given configurational space (the energy and uncertainty profiles in Figure B.3 to Figure B.10). The agreement of the density profiles of water between AIMD and NNPs MD with the same box size (3×3) is reported in

Figure B.11. It can be observed that the density profile of water in NNPs MD simulation is more smooth than that in AIMD, and some disagreements are exhibited in the bulk water area. We ascribe the disagreements and the fluctuation of AIMD density profiles to the inadequate equilibration of AIMD simulations. Moreover, the average energy profiles of Au(100)-water with four *OH that started from different points are well converged as shown in Figure B.12, indicating that our MD simulations are ergodic and the time scale is long enough.

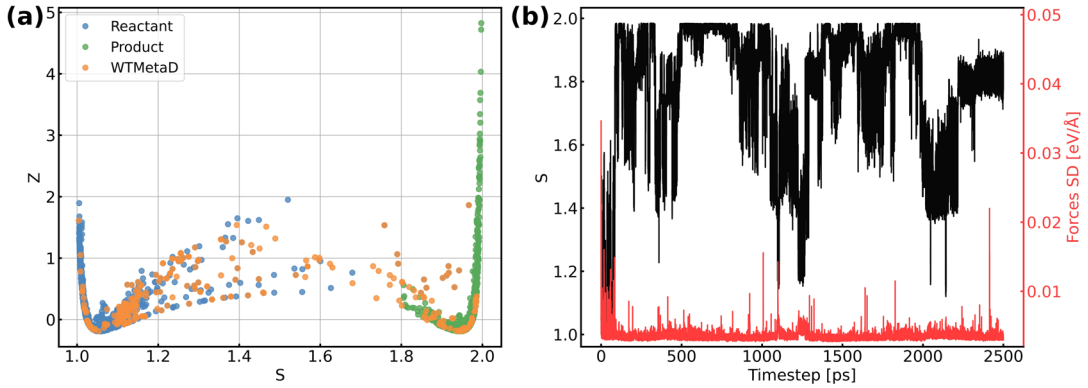

Figure 4.3: (a) Distribution of reference structures in the configurational space described by path collective variables s and z, where s represents the progress along the path between reactants and products and z represents the distance from the path. (b) Evolution of s and forces standard deviation (SD) along 2.5 ns MetaD simulation.

Except for the validation of model accuracy and reliability, we also evaluated the overall computational efficiency of the proposed scheme in terms of training the initial model, model retraining, CUR matrix decomposition, production MD simulations for 5 ns and DFT labelling as demonstrated in Figure B.13. For the systems in this study, the required computational time for 1000 AIMD steps is approximately 650 CPU hours, corresponding to 1543.8 hours in total for generating the initial AIMD dataset if using 80 CPU cores for each job. Training on the initial dataset takes about 40 hours, while the cost of retraining the new models can be substantially reduced by loading pretrained model parameters. The MD simulation driven by NNPs accounts for the highest computational cost in an active learning iteration, which takes approximately 7 days to run 5 ns simulations on an NVIDIA RTX3090 GPU. In comparison, AIMD needs more than 7 years to run 5 ns using 80 CPU cores, being about 3-400 times slower than NNP MD. To train the NNPs for a system to run more than 5 ns MD, 5 to 10 iterations are usually needed, which corresponds to 1000-2000 labelled structures as indicated in Table B.3. It is worth noting that the ASE MD engine used in this study is not specialized for GPU computing,

resulting in high overheads of data transfer between GPU and CPU. It can be expected that the computational efficiency of NNP MD can be further improved in the future by using GPU-specialized MD code.

The validation of NNPs via application in MetaD simulations is crucial as the configurational space of the full reactive process can be huge while the transitional states are rarely visited. As shown in Figure 4.3, our training data points are evenly distributed in the configurational space described by path collective variables,[116], and the force uncertainties along 2.5 ns MetaD simulations are all considerably small (all smaller than 0.05 eV/Å). Both metrics build confidence that the trained NNPs are reliable to capture the characteristics of all energetically relevant states, especially transitional states, of ORR. Furthermore, the trained models have shown excellent transferability when using them for the inference of Au(110)-water and Au(111)-water interfacial systems as demonstrated in Figure B.1a. Despite missing structural information for the two similar systems, the trained models still well predicted the energy and forces with both low errors and uncertainties for all Au(110)-water and Au(111)-water interfacial structures, which indicates that the proposed scheme and trained models can easily generalize to systems across a wide range of metals and their different facets.

4.3.2 Full metadynamics simulation of ORR

After systematic validations, the trained NNPs are used to study both adsorption energetics and kinetics of ORR at the gold-water interface. It is well-known that ORR on Au(100) in alkaline electrolytes proceeds via the complete four-electron transfer mechanism, while the partial two-electron transfer mechanism dominates on other Au facets, such as Au(111) and Au(110).[28, 29, 26] Despite the use of new techniques and persistent efforts devoted by researchers, the reason why ORR activity is exceptional and facet-dependent on gold remains elusive. There are several assumptions that may provide a clear answer to this question, including the outer-sphere mechanism of ORR,[26, 179, 180] and the role of preadsorbed species and solvents.[181, 182] All these assumptions call for a full atomic simulation that elaborately considers the ambient conditions of Au(100) and models the reaction without any simplification.

The first step of ORR on Au(100) is the O_2 activation, which is also considered a key step that determines the activity of catalysts that weakly interact with adsorbates. According to whether O_2 closely adsorbs on Au(100), the reaction can be initiated via the inner-sphere mechanism in which the slab directly transfers electrons to closely adsorbed O_2, or the outer-sphere mechanism in which ORR occurs away from the slab by several solvent layers. The adsorption energy of $*O_2$ molecule and *OH with different coverage is summarized

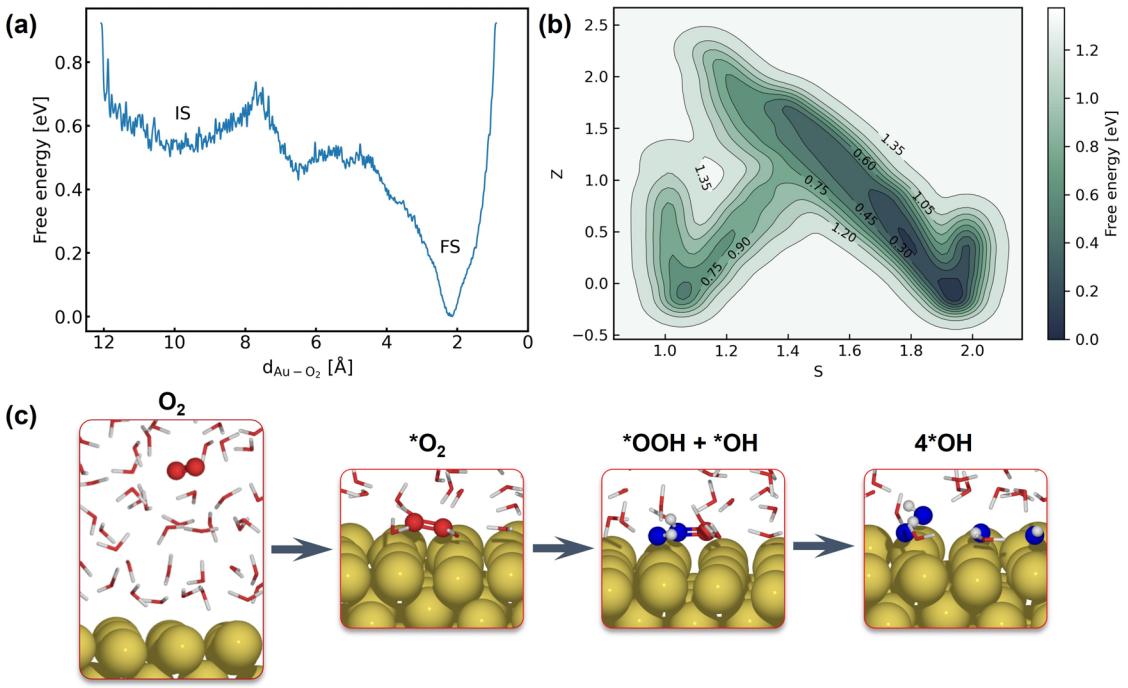

Figure 4.4: (a) Free energy landscape of O_2 migration from bulk water to Au(100) surface. (b) Free energy landscape of $*O_2$ reduction to *OH described by path collective variables. (c) Snapshots for O_2 in bulk water, initial state, transitional state, and final state.

in Table B.5, suggesting weak interaction between these species and the Au(100) slab. As demonstrated in Figure B.9, our 5 ns MD simulations at Au(100)-water interface with one O_2 molecule have shown that the O_2 molecule will be in close contact with Au(100) surface, yielding a density peak at 2.1 Å. We further carried out a MetaD simulation that models the migration of O_2 from bulk water to Au(100) surface as shown in Figure 4.4a. It is found that there is no stable local minima for O_2 saturating in bulk water, and the migration barrier can be easily overcome by the thermal fluctuation of the model system. As shown in Figure B.14, a simple MD simulation modeling the movements of O_2 in the bulk water part of the interface also proves this conclusion. After 350 ps simulations, the O_2 molecule finally moved from bulk water to Au(100) surface. Based on these results, we model the reaction process with O_2 directly adsorbed on Au(100) and believed that the bond breaking of O_2 molecule could be the rate-determining step of ORR on Au(100).

The full atomic simulation of ORR is then conducted to investigate the bond-breaking process in O_2 molecule and the formation of hydroxyls by using metadynamics simulations. The reaction coordinates of ORR are described by path collective variables (CVs)[116] with the initial state (Au(100)-1O_2/57H_2O) and final state (Au(100)-4OH/55H_2O) selected as

two reference structures. The distance to reference structures is quantified by the number of oxygen atoms (C_{O_2-O}) and hydrogen atoms (C_{O_2-H}) around the O_2 molecule. As summarized in Table B.6, these two descriptors can well capture and differentiate the structural characteristics of different possible intermediate states of ORR, including $*O_2$, *OOH, $*H_2O_2$, *O, and *OH. The well-designed CVs enable us to automatically search the reaction path without using any prior knowledge about the reaction mechanism. In this approach, instead of modeling multiple possible reaction pathways and verifying which one is energetically most favorable, we only need to incrementally extend the explored PES (with our NNPs) from equilibrium states to non-equilibrium transitional states by using active learning. Furthermore, this strategy can be easily generalized to simulate more complex model systems and chemical reactions.

Figure 4.4b shows the obtained free energy landscape of ORR as a function of path CVs, where s is the progress along the reference path, and z is the distance to the reference path. The landscape is comprised of two basins which correspond to the initial state and final state of ORR. Figure 4.3b also shows the time evolution of the s collective variable. It can be seen that the first basin in the landscape has been completely filled after approximately 100 ps, which corresponds to the transition from O_2 to hydroxyls. Filling the second basin, which can be regarded as the transition from hydroxyls to O_2, becomes much more difficult than the first one with the employed CVs in this study. However, it should be pointed out that the depth of the first basin is enough to evaluate the activation energy of bond breaking in O_2. The energy barrier of the transition from O_2 to hydroxyls is estimated to be 0.3 eV, which is in good agreement with experimental findings that Au(100) displays high ORR activity. It is noteworthy that the simulation box in this study is small in comparison with the realistic interface structure. The limited cell size can result in slightly higher formation energies of hydroxyl as demonstrated in Table B.5, which can be ascribed to the stronger repulsion between hydroxyl in smaller boxes and the possible lateral correlation of solvation shells. Besides, we also expect that the bond breaking of the O_2 molecule can be more difficult because of the easier recombination of individual oxygen atoms. Both effects can make the ORR in small cell to be less facile, while further supporting our conclusion that ORR is facile on Au(100) even modeled with a limited number of water molecules. The snapshots for O_2 in bulk water, initial state, transition state, and final state are displayed in Figure 4.4c. At first, the O_2 molecule is partially protonated by neighboring water molecules to *OOH, suggesting the associative reaction pathway proposed by Nørskov et al.[8]. However, the subsequent formation of *O is not observed in the overall reaction as the remaining oxygen atom is immediately protonated by reacting with water. Therefore, the reaction pathway observed from our

simulations can be summarized as follows:

$$^*O_2 + H_2O \rightarrow {}^*OOH + {}^*OH$$

$$^*OOH + H_2O \rightarrow 3^*OH$$

The MetaD simulation highlights the role of water molecules as a reactant of ORR, suggesting that the explicit modeling of solvents is indispensable in theoretical electrocatalysis.

4.4 Conclusions

In summary, the reactive process of ORR is investigated by MetaD simulations that are significantly accelerated by high fidelity NNPs in this study. By using an active learning strategy underpinned by CUR matrix decomposition, we obtained an NNP ensemble that exhibits exceptional performance and reliability for the prediction of structural properties and forces in the configurational space of Au(100)-water interface. By leveraging well-designed path collective variables, the ORR can be fully and automatically simulated without the need to elaborately consider multiple reaction pathways. Our MetaD simulations suggest that ORR proceeds in the associative reaction pathway, while the *OOH reaction intermediate is directly reduced to two *OH with the participation of neighboring water molecules rather than dissociating into *OH and *O. The low energy barrier of ORR predicted in this study well explains the outstanding experimental ORR activity. The longer time-scale simulations enabled by NNPs can give us deeper insight into the nature of chemical reactions, such as the facet-dependent ORR on different Au facets. Besides, the effect of cations on the ORR activity of gold is also a meaningful extension of this work. In perspective, the full atomic simulation conducted here can be conveniently extended to other model systems and become a valuable tool for investigating complex chemical reactions in a straightforward manner.

CHAPTER 5

Facet-dependent ORR on Au surfaces

This chapter is based on the case study in paper III – "A comprehensive study of facet-dependent oxygen reduction dynamics on gold surfaces using metadynamics and graph neural networks". The paper is also included in this thesis together with the corresponding supplementary information in Appendix C.

5.1 Introduction

The electrochemical reduction of oxygen is a crucial process in various energy conversion and storage devices, including fuel cells and metal-air batteries.[7, 4, 3] The efficiency and selectivity of this reaction are predominantly governed by the nature of the electrode material. Recognizing this, there is a consistent pursuit of efficient and cost-effective catalysts towards oxygen reduction reaction (ORR) in academia and industry globally. Gold, traditionally viewed as a noble and therefore catalytically inert metal, has undergone a renaissance in the realm of catalysis over the past few decades.[28, 183, 184, 185, 186, 187] In particular, gold nanoparticles have demonstrated exceptional catalytic activity for a range of reactions, from CO oxidation to selective hydrogenations.[186, 188] This unexpected catalytic activity of gold is attributed to its unique electronic properties, particle size effects, and the influence of the support material.[189, 190, 191, 192, 193]

The ORR on gold has been extensively studied in both acidic and alkaline medias.[194] In acidic solutions, gold predominantly follows a 2-electron ($2e^-$) pathway, producing hydrogen peroxide. Intriguingly, the Au(100) surface in alkaline media, not only demonstrated in a 4-electron ($4e^-$) ORR pathway but even outperforms platinum within specific potential ranges. Additionally, ORR on gold showcases a pH-dependent catalytic behavior, with Au(100) favoring a complete four-electron transfer, in contrast to the partial two-

electron transfer observed on other facets like Au(111) and Au(110).[28, 29, 179] While these experimental observations have been acknowledged for over a decade, an in-depth atomistic understanding of the reaction mechanisms remains elusive. Density functional theory (DFT) simulations combined with the computational hydrogen electrode (CHE) method have been instrumental but show limitations, especially in predicting the ORR activity of gold and the pH-dependent catalytic behavior. Using DFT calculations, Lu *et al.* highlighted the significant role of the interaction between co-adsorbed water and reaction intermediates in ORR on gold, facilitating O-O bond cleavage and thus promoting $4e^-$ reduction.[181] Duan and Henkelman suggested that the applied potential could influence the adsorption energies of ORR intermediates, resulting in the pH-dependent ORR on gold and a reduced theoretical overpotential.[27]

It is noteworthy that many studies have primarily focused on adsorption energetics, often neglecting the dynamic nature of electrochemical interfaces and its influence on ORR acativity. In our prior research,[195] we introduced a framework leveraging graph neural network (GNN) potentials to accelerate metadynamics simulations, shedding light on dynamic nature of electrochemical interfaces and the ORR kinetics at Au(100)–water interface. It offers direct insights into ORR kinetics, considering explicit solvents and long-scale molecular dynamics (MD) simulations, with a particular emphasis on the reaction kinetics of Au(100) surface.

In this extension work, we delve deeper into the ORR on prominent gold surfaces, namely Au(100), Au(110), and Au(111), incorporating explicit solvents and employing GNN-accelerated metadynamics for modeling ORR dynamics. Leveraging larger simulation boxes, we aim to provide a more comprehensive and nuanced understanding of the oxygen reduction process on gold surfaces. Our systematic exploration of interface dynamics across varied adsorbates offers an in-depth perspective on the nature of these interfaces. Notably, our simulations corroborate the facet-dependent behavior of ORR on gold, aligning well with experimental observations. Besides, our metadynamics investigations revealed the significant role of co-adsorbed species on ORR reactivity. With this research, we hope to bridge existing knowledge gaps and pave the way for the design of more efficient and robust gold-based electrocatalysts for ORR.

5.2 Computational methods

5.2.1 Generation of neural network potentials

Our methodology is built upon the techniques outlined in our previous research.[195] We utilized pretrained models from this work and derived the final dataset through the active

learning framework built in it. The composition of various interfacial structures within the dataset, along with their respective error metrics, is presented in Table C.1. We allocated 90% of the dataset for training and reserved the remaining 10% for validation, where the latter was employed for early stopping once the force error reached an acceptable threshold. To assess the performance of the trained model, we utilized multiple error metrics, including the mean absolute error (MAE) and root mean squared error (RMSE) for both energy and force predictions.

Our MD and metadynamics simulations utilized an ensemble of six neural network potentials, each based on different architectures of the polarizable atom interaction neural network (PaiNN) model.[110] The architectures of these models, along with their respective error metrics trained on the final dataset, are detailed in Table C.2. To introduce model diversity, we employed different node feature sizes, while maintaining a consistent cutoff radius of 5 Å for all models.

Both the model training and production simulations were executed on an NVIDIA GeForce RTX 3090 GPU, utilizing float32 precision. The weight parameters of models were initialized randomly and subsequently optimized on a consistent data split using stochastic gradient descent to minimize the mean square error (MSE) loss. We set the force loss weight and energy loss weight to 0.95 and 0.05, respectively, to ensure a high force prediction accuracy for propagating reliable MD simulations.

The Adam optimizer[171], as implemented in PyTorch[172], was employed to train our model parameters. We used an initial learning rate of 0.0001, default parameters of β_1=0.9 and β_2=0.999, and a batch size of 12. An exponential decay learning rate scheduler with a coefficient of 0.96 was used to adjust the learning rate every 100,000 learning steps.

5.2.2 DFT calculations

Our initial DFT dataset is derived from AIMD trajectories of Au(110)-water and Au(111)-water interfaces. Utilizing pretrained models eliminated the need for ab-initio reference structures from the Au(100)-water interface and reduced the number of reference structures for the Au(110)-water and Au(111)-water interfaces. The Au(110)-water system comprises 36 H_2O molecules atop a (2×3) tetragonal Au(110) surface (denoted as Au(110)-36H_2O). Similarly, the Au(111)-water system is modelled as 36 H_2O molecules on atop of a (3 × 3) Au(111) surface (denoted as Au(111)-36H_2O). MD simulations were conducted in VASP[94, 95, 96, 97] using these configurations for 10 ps, with a 0.5 fs timestep and a target temperature of 350K using the Nosé–Hoover thermostat[106]. The bottom two atomic layers remained fixed during the MD simulations. We adopted a 350 eV energy

cutoff for plane-wave basis and a Monkhorst-Pack k-grid with the k-point density of 0.5 $Å^{-1}$[173]. The PBE functional, combined with the D3 Van der Waals correction, was used to approximate exchange-correlation effects[90, 99]. The same parameters were employed for single-point DFT calculations during active learning iterations.

5.2.3 Production MD simulation

The production MD simulations driven by the NNP ensemble are conducted using the MD engine within Atomic Simulation Environment (ASE) python library.[98] We utilized larger simulation boxes to precisely capture the dynamics at the interface. The Au(100)-water interface was modeled with $102H_2O$ molecules on a (5×5) Au(100) surface, resulting in 431 atoms in total. The Au(110)-water interface had $115H_2O$ molecules on a (4×5) Au(110) surface, with 445 atoms. The Au(111)-water interface used a tetragonal $(5\times3\sqrt{2})$ Au(111) slab with 108 water molecules on it, with 474 atoms. The atomic structures for these interfaces are presented in Figure 5.1.

Building on these foundational interface structures, we incorporated O_2 molecules and hydroxyl groups to delve deeper into the adsorption energetics of key reaction intermediates and to set up the initial structures for metadynamics simulations. For every foundational interface, we considered 2OH, 4OH, 6OH, and 8OH cases by removing corresponding number of hydrogen atoms in the system. Besides, we also considered the $1O_2$ and $2O_2$ cases by adding corresponding number of oxygen atoms and removing corresponding number of water molecules. The resulting structures, along with their error metrics, are detailed in Table C.1. For clarity and convenience, throughout the paper, we will label each system without showing the number of H_2O as the number of water molecules is not important for discerning different systems and drawing our conclusions. For example, the Au(100)-$2O_2/98H_2O$ will be termed as Au(100)-$2O_2$ hereafter. The momentum of the model systems was initiated using a Maxwell–Boltzmann distribution, with the temperature set at 350 K. For each system, we propagated 1,500 ps MD simualations by Langevin dynamics with the target temperature of 350 K, the timestep of 0.25 fs, and the friction coefficient of 0.02. Of the 1500 ps MD trajectory, the initial 500 ps served for equilibration, while the subsequent 1000 ps was used to sample properties of interest. The uncertainty was quantified by the force standard deviation (SD), with a threshold set at 0.5 eV/Å. Simulations were halted if the force SD of a configuration exceeded this value. The formation energy of *OH and the adsorption energy of O_2 are calculated based on the method in Ref.[37] and our prior study.

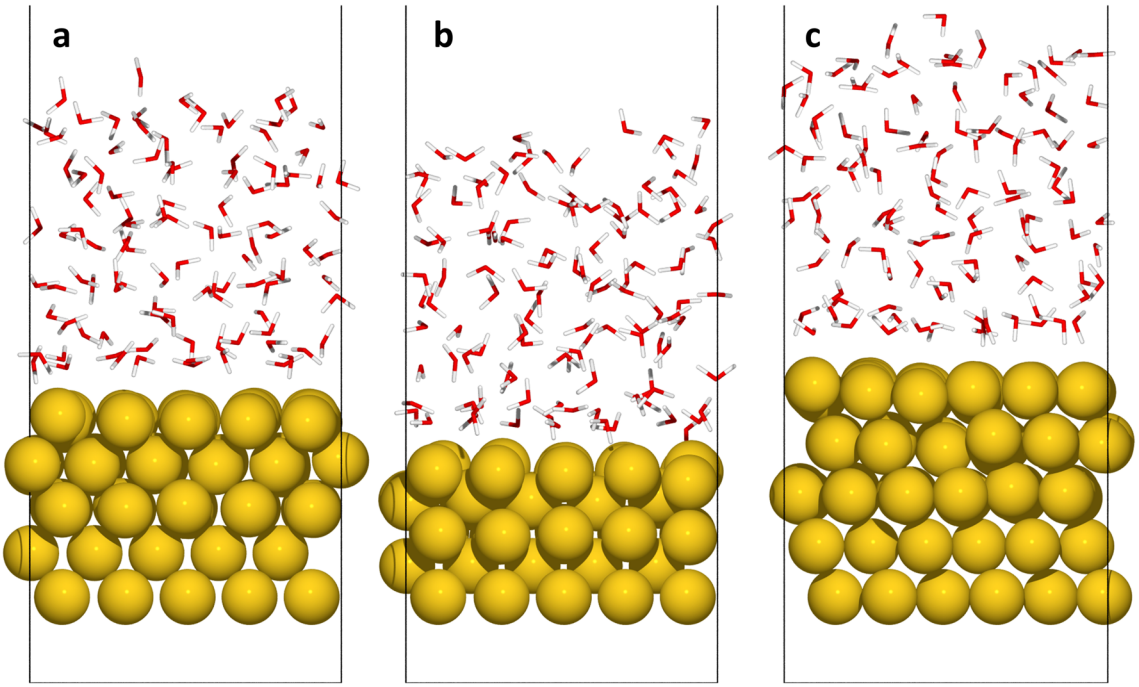

Figure 5.1: Side view of (a) Au(100)-102H_2O, (b) Au(110)-115H_2O, and (c) Au(111)-108H_2O interface structures.

5.2.4 Metadynamics simulation

In this study, all the enhanced sampling simulations are performed with the well-tempered version of metadynamics[175] The calculation of collective variables and bias potential of metadynamics is achieved by PLUMED which is interfaced to ASE.[98, 121, 122, 117] We use the path collective variables and the same parameters in ref.[195] to describe the reaction. And the coordination numbers C_{O_2-O} and C_{O_2-H} are used to define the configurational space of the path.

5.3 Results and discussion

5.3.1 Regular MD simulations

We systematically investigated the dynamics at gold-water interfaces, specifically considering the presence of adsorbed O_2 and *OH. Figure 5.2 presents the density profiles of water molecules, oxygen atoms, and hydrogen atoms relative to their distance from gold surfaces. This figure focuses on systems containing pure water, eight hydroxyls, and two oxygen molecules in the electrolyte. For detailed density profiles, average energy and uncertainties in other systems, readers are directed to Figure C.1 to Figure C.21. In these density profiles. Notably, in these density profiles, two pronounced peaks are observed for all systems within 10 Å, suggesting the presence of two structured water layers near the

slab. Among the surfaces studied, Au(110) exhibited the closest first peak to the surface at 2.5 Å. In comparison, the peaks for Au(100) and Au(111) are situated slightly farther away at 2.8 Å and 2.9 Å, respectively. The second layer of ordered water, as indicated by the second peak, is fairly consistent across the surfaces, positioned at 6.0 Å for both Au(100) and Au(110), and marginally farther at 6.1 Å for Au(111). Moving beyond 10 Å and up to 15 Å, the effects of the surface on water structuring become negligible. In this region, the density distribution of water molecules becomes almost constant and matches that of bulk water. Beyond 15 Å is the water–vacuum interface, where the densities gradually decline to zero.

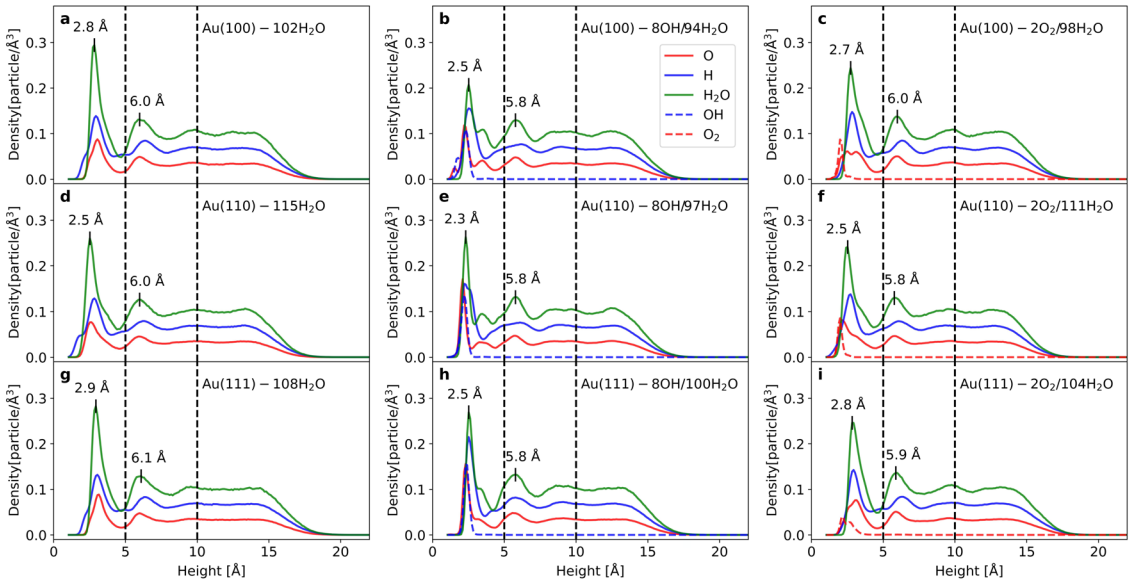

Figure 5.2: Density profiles of O, H, and H_2O as a function of the distance from Au slabs.

Upon introducing 8 *OH groups, there is a noticeable shift in the density profiles as shown in Figure 5.2b, e, and h. The positions of first peaks in the density profiles are decreased to 2.5, 2.3, and 2.5 Å for Au(100), Au(110), and Au(111), respectively. Meanwhile the second peaks consistently locate at 5.8 Å across all surfaces. The presence of hydroxyls not only modifies the peak positions but also reshapes the overall density distributions of water molecules. This behavior can be attributed to the stronger chemical adsorption of hydroxyls compared to water and the hydrogen bond network formed between these hydroxyls and surrounding water molecules.[196] While *OH forms a direct chemical bond with gold atoms through chemisorption, water primarily interacts through weaker forces like van der Waals or hydrogen bonds. The chemisorbed *OH can act as an anchor point, fixing the surrounding water molecules through hydrogen bonding. This anchoring effect can reduce the mobility of water molecules and lead to a more tightly packed first layer

of water molecules that closer to the surface.

In contrast, the inclusion of O_2 induces only minor shifts in the density distribution peaks without altering their overall shape as illustrated in Figure 5.2c, f, and i. The reason is that O_2 is a nonpolar molecule, which interacts weakly with the metal through physisorption and lacks the ability to form hydrogen bonds with water. Consequently, *OH induces more pronounced structural changes in the water layer, leading to significant alterations in the density profiles, while the influence of O_2 remains relatively subtle.

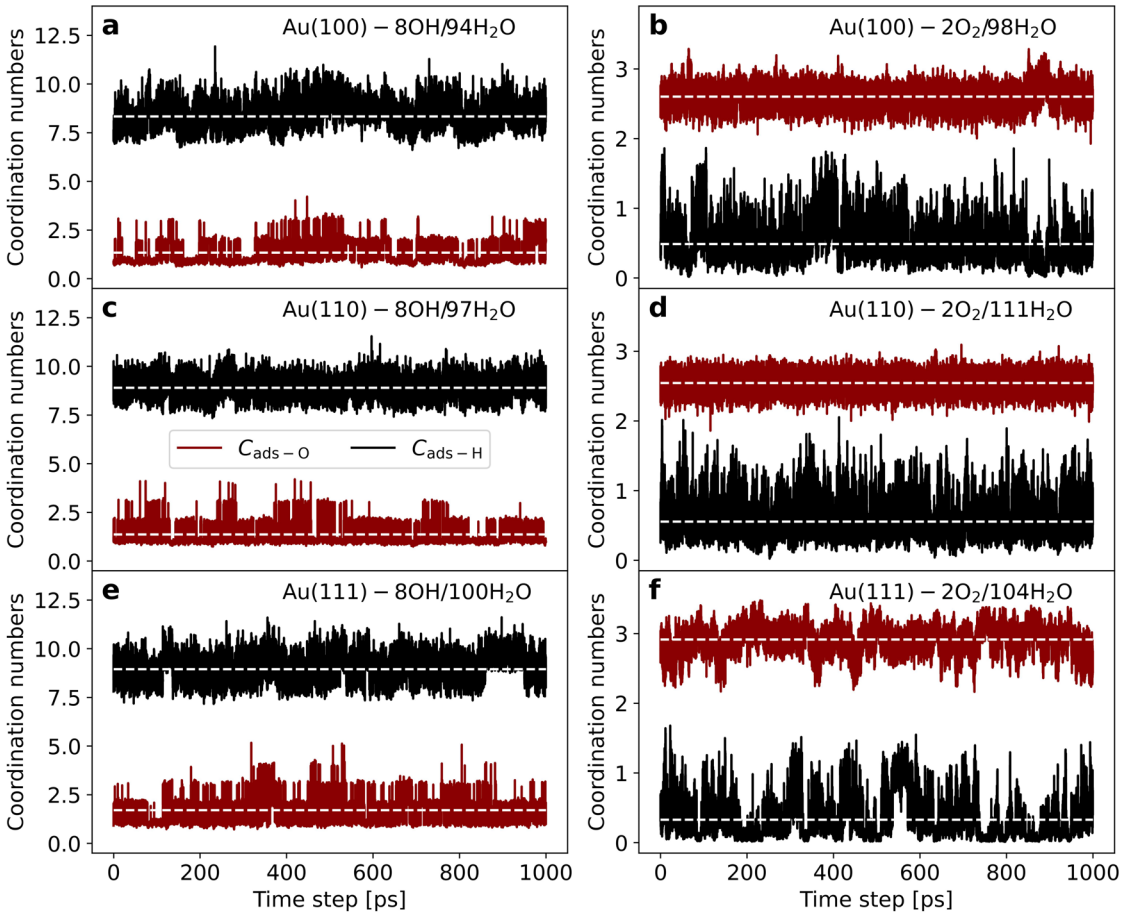

Figure 5.3: Evolution of coordination numbers $C_{\mathrm{ads-O}}$ and $C_{\mathrm{ads-H}}$ throughout MD simulations. The dashed lines denote the average coordination number.

As illustrated in Figure 5.3, we delved deeper into the evolution of coordination numbers for adsorbed species to gain insights into their local environments. Here, $C_{\mathrm{ads-O}}$ and $C_{\mathrm{ads-H}}$ represent the number of surrounding oxygen and hydrogen atoms for the adsorbates, respectively. For systems with eight hydroxyls, as depicted in Figure 5.3a, c, and e, the average coordination numbers $C_{\mathrm{ads-O}}$ for Au(100), Au(110), and Au(111) are 1.34, 1.36, and 1.70, respectively. Meanwhile, their corresponding $C_{\mathrm{ads-H}}$ values are 8.34, 8.90, and 8.94. In systems with two oxygen molecules, the average $C_{\mathrm{ads-O}}$ values for Au(100),

Au(110), and Au(111) are 2.60, 2.54, and 2.91, respectively, with $C_{\mathrm{ads-H}}$ values of 0.49, 0.56, and 0.33, respectively. A notable observation is that the fluctuation in $C_{\mathrm{ads-O}}$ is smaller than in $C_{\mathrm{ads-H}}$ across all systems. This can be attributed to the dynamic proton transfer in liquid water, which results in rapid changes in the local environments around hydroxyls. Conversely, due to the nonpolar nature of O_2, it is less inclined to form a hydrogen bond network with adjacent water molecules. Therefore, the bond breaking of O_2 can be difficult may not be accessible using regular MD simulations.

Table 5.1: Formation energies of *OH and adsorption energies of O_2 for different model systems

System	Species	θ (Coverage)	$\Delta E/n$ (eV)
Au(100)	2OH	0.080	1.131
	4OH	0.160	1.051
	6OH	0.240	1.087
	8OH	0.320	1.160
	$1O_2$	0.040	-0.657
	$2O_2$	0.080	-0.428
Au(110)	2OH	0.100	0.770
	4OH	0.200	0.755
	6OH	0.300	0.787
	8OH	0.400	0.785
	$1O_2$	0.050	-0.745
	$2O_2$	0.100	-0.736
Au(111)	2OH	0.067	0.880
	4OH	0.133	0.977
	6OH	0.200	1.055
	8OH	0.267	1.149
	$1O_2$	0.033	-0.768
	$2O_2$	0.067	-0.554

Table 5.1 exhibited the formation energies of *OH and the adsorption energies of O_2 molecule for each model system. For all three surfaces, the formation energy of *OH generally increases with increasing coverage. This suggests that as more *OH groups are adsorbed, it becomes energetically more favorable for them to form. Notably, Au(110) consistently exhibits the lowest formation energy for *OH, indicating that *OH adsorption is most favorable on this surface. Conversely, Au(100) displays the highest formation

energy, especially with 8 co-adsorbed *OH groups. The adsorption energies of O_2 are negative for all systems, indicating that the adsorption process is exothermic and energetically favorable. Combining the stable evolution observed for $C_{\mathrm{ads-O}}$ in Figure 5.3, this suggests a preference for the inner-sphere mechanism, wherein the ORR predominantly occurs near the slab.

5.3.2 Metadynamics with single oxygen molecule

With the prepared the initial structures and well-performed MLIPs, now we are able to investigate the how the reaction happens with metadynamics simulations. We firstly simulated ORR with the presence of one single oxygen molecule in the liquid water. Previous studies suggests two plausible mechanisms for ORR: the inner-sphere mechanism, where O_2 is closely adsorbed on the slabs, and the outer-sphere mechanism, wherein ORR takes place several solvent layers away from the slab.[26, 179, 180] Our regular MD simulation results, in alignment with previous studies,[195] consistently show the O_2 molecule residing in the first water layer. Consequently, our metadynamics simulations were exclusively conducted following the inner-sphere mechanism. To ensure the reliability of the production metadynamics, it is imperative to sample and include a substantial number of reference structures into the training dataset, particularly those originating from rare events on the potential energy surface. This task poses a significant challenge given the intricate reaction systems explored in this study. Specifically, driving metadynamics to escape from the final state (4*OH in the liquid) is challenging as it correlates to the oxygen evolution process, which is inherently difficult to initiate and necessitates a high applied voltage for activation. Consequently, our metadynamics simulations for each system are confined to limited length-scales.

As depicted in Figure 5.4b, on the Au(100) slab, the metadynamics simulation stops at approximately 275 ps due to the emergence of structures with excessive uncertainty (force standard deviation exceeding 0.5 eV/Å), with the O-O bond breaking being observed at 232 ps. While extending the simulation length-scale is feasible through additional active learning iterations, we ascertain that the current length-scale of the simulation is sufficient to encapsulate the overall ORR process. The atomic structures of five representative configurations are illustrated in Figure 5.4a, corresponding to key points along the reaction pathway, as marked by the white dashed line in the free energy landscape depicted in Figure 5.4c. The free energy landscape, characterized by the path CVs, manifests three obvious basins. The initial stable states encompass two different kinds of configurations: Au(100)-1O_2/H_2O where the pure O_2 molecule is adsorbed onto the slab, and Au(100)-(1OH+1OOH)/H_2O showing the presence of one *OOH and one *OH. This indicates the

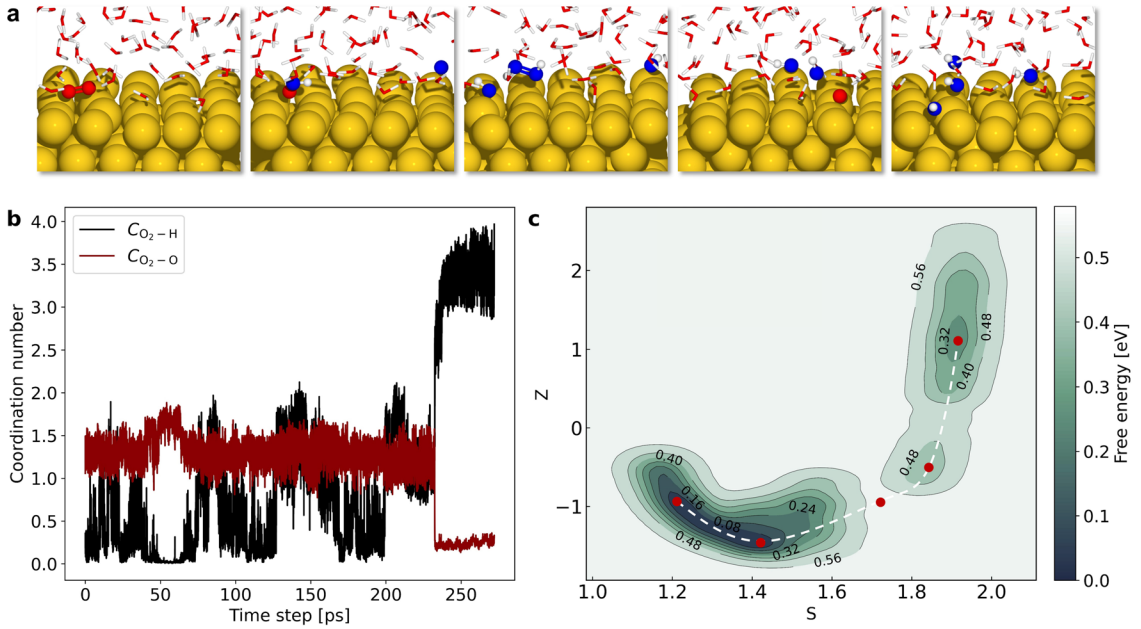

Figure 5.4: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(100) with one oxygen molecule described by path CVs.

proton transfer from the surrounding water molecules to the adsorbed O_2 molecule is facile with only negligible energy barrier. The third point along the reaction pathway is characterized by the adsorbed hydrogen peroxide (*H_2O_2), originating from the previously formed *OOH that accepted an additional proton from surrounding water molecules. However, the occurrence of this event is very rare, as illustrated by the sharp decline in C_{O_2-O} values depicted in Figure 5.4b, coupled with the high free energy of approximately 0.57 eV. The O-O and O-H bond lengths in H_2O_2 are approximately 1.48 Å and 1.03 Å respectively, with the former aligning with measurements observed in gas-phase H_2O_2, while the latter is slightly elongated in comparison to its gas-phase counterpart. The introduction of hydrogen atoms weakens the O-O bond, breaking it into two hydroxyls. Interestingly, we identified the presence of an unbonded single oxygen atom at the fourth point (Au(100)-(1O+2OH)/H_2O), corresponding to the short interval in Figure 5.4b where the C_{O_2-O} values fluctuate around 2.75. The single oxygen atom is quickly protonated by adjacent water molecules, transitioning into a hydroxyl. Our metadynamics simulation elucidated that ORR on Au(100) proceeds in a four-electron transfer reaction pathway with a reaction barrier of approximately 0.50 eV.

Transitioning to the Au(110) slab, the metadynamics simulation halts at 228 ps, with the O-O bond breaking observed at 170 ps as depicted in Figure 5.5b. The reaction pathway, as

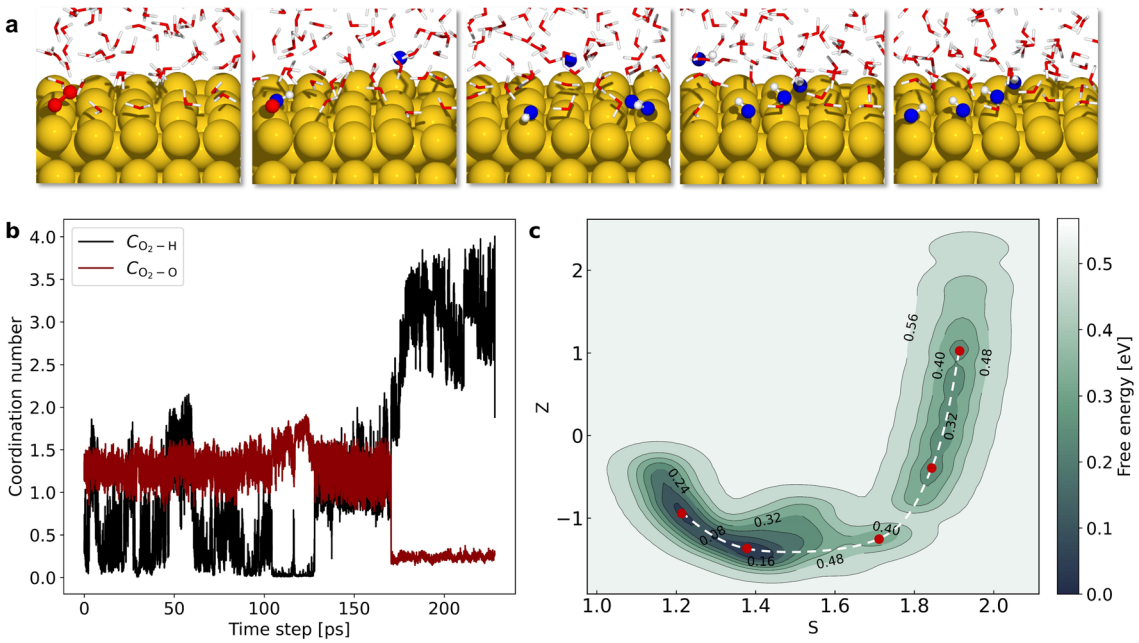

Figure 5.5: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(110) with one oxygen molecule described by path CVs.

demonstrated in Figure 5.5a, is similar to that of Au(100), albeit without the identification of the unbonded oxygen atom. The free energy landscape of Au(110) closely mirrors that of Au(100), yet with only two basins as illustrated in Figure 5.5c. A notably more stable $^*H_2O_2$ intermediate emerges in the midway during intra-basin transition. This is further evidenced by the evolution of C_{O_2-O} values within the time interval of 131 ps to 170 ps, as demonstrated in Figure 5.5b. This observation aligns with experimental findings that ORR on Au(110) and Au(111) involves the formation of $^*H_2O_2$ intermediates.[28, 29, 26] Analogous to the Au(100) case, the O-O bond dissociation emerges as the rate-determining step, albeit with a slightly lower energy barrier of 0.48 eV.

Moving onto the Au(111) slab, the simulation halts at 350 ps, with the O-O bond breaking observed at 290 ps, as depicted in Figure 5.6b. Contrary to Au(100) and Au(110) cases, the first basin exclusively includes configurations with the oxygen molecule with no *OOH identified, as shown in Figure 5.6c. The adsorbed O_2 molecule is fully protonated to $^*H_2O_2$ in a short time interval. Similarly to the Au(110) case, the existence of $^*H_2O_2$ is more stable than in the Au(100) scenario. This is further substantiated by the evolution of C_{O_2-O} values within the time interval of 210 ps to 290 ps, as demonstrated in Figure 5.6b. The free energy barrier for oxygen reduction is approximately 0.42 eV.

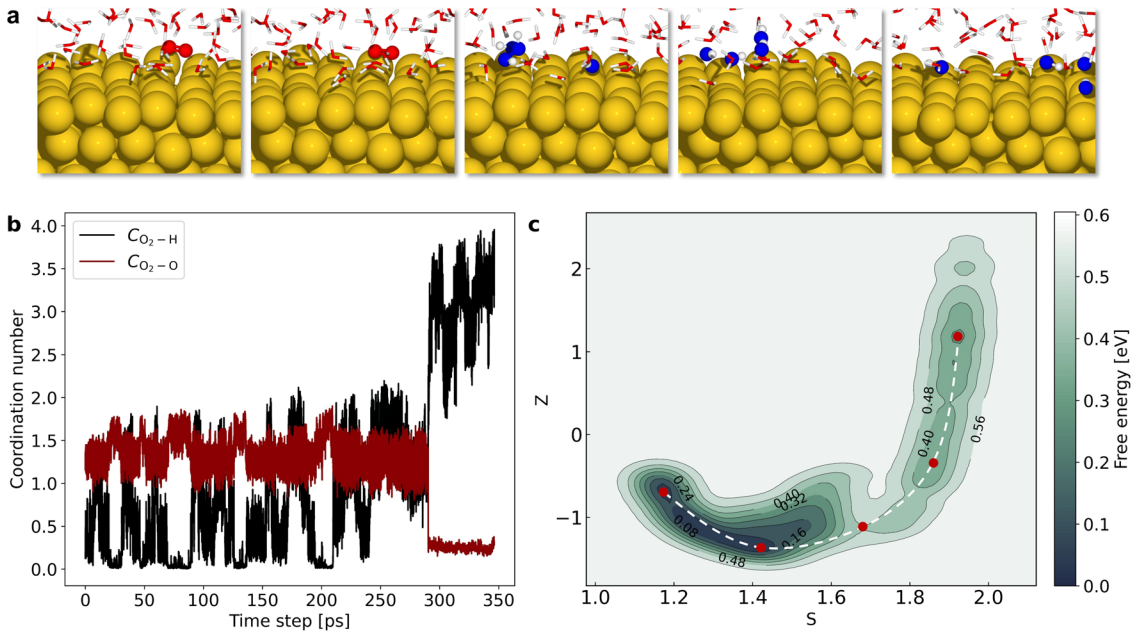

Figure 5.6: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(111) with one oxygen molecule described by path CVs.

5.3.3 Metadynamics with two oxygen molecules

While our simulations shed light on the variance in $*H_2O_2$ stability across different facets, they did not accurately predict the true ORR activity trends across these facets. We hypothesize that the co-adsorbed O_2 or hydroxyl groups on the surfaces might also exert influence on the ORR dynamics. To delve deeper into this aspect, we extended our metadynamics simulations to systems with two oxygen molecules present in the liquid water.

Figure 5.7b illustrates a metadynamics simulation conducted at the Au(100)-$2O_2/H_2O$ interface over a duration of 1480 ps, within the designated uncertainty threshold. The dissociation O-O bond of the first O_2 molecule occured rapidly at 41 ps. However, the bond in the second oxygen molecule did not break during the entire simulation. This indicated the notable impact of co-adsorbed *OH on the ORR dynamics, hindering the reduction of the remaining O_2 molecules.

Figure 5.7c presents the free energy landscape, revealing two major basins and a minor one. The first major basin corresponds to Au(100)-$2O_2/H_2O$, the initial state of the reaction. As the reaction progresses, overcoming a free energy barrier of 0.24 eV, both O_2 molecules are partially protonated to form two *OOH molecules, signified by the minor

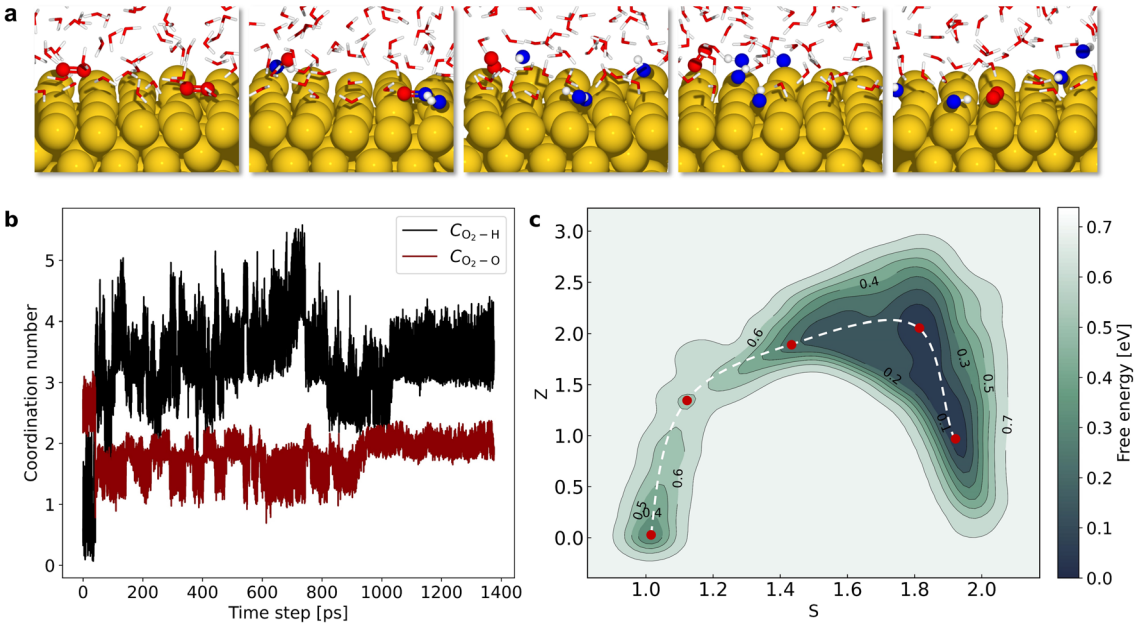

Figure 5.7: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(100) with two oxygen molecules described by path CVs.

energy basin. Following this, one of the *OOH groups undergoes O-O bond dissociation, resulting in an *OH group and an unbonded oxygen atom, which quickly accepts a proton from the surrounding water molecules to form another *OH. This *OH formation state exhibits substantial stability, persisting through the fourth and fifth points in the reaction pathway, all encapsulated within the second major energy basin.

In contrast with the Au(100)-1O_2/H_2O case, the reduction of the first O_2 molecule is considerably more facile. Besides, the presence of *H_2O_2 is not observed during the simulation further validating that the reaction mechanism found in this study aligns well with experimental findings.

Transitioning to the analysis of the Au(110)-2O_2/H_2O interface, the simulation stoped at 854 ps, with the O-O bond of the first O_2 molecule breaking at 432 ps as illustrated in Figure 5.8b. Similar to the Au(100)-2O_2/H_2O case, the second O_2 remains intact during the entire simulation. The free energy landscape at this interface showcased two major basins, signifying the initial and final states of the reaction. At the second point of the reaction pathway, the state is Au(110)-(2OOH+2OH)/H_2O, exhibiting a similar adsorption behavior to the Au(100) surface. A notable observation was the presence of *H_2O_2 at the third point, which quickly transitioned into 2*OH groups. The second

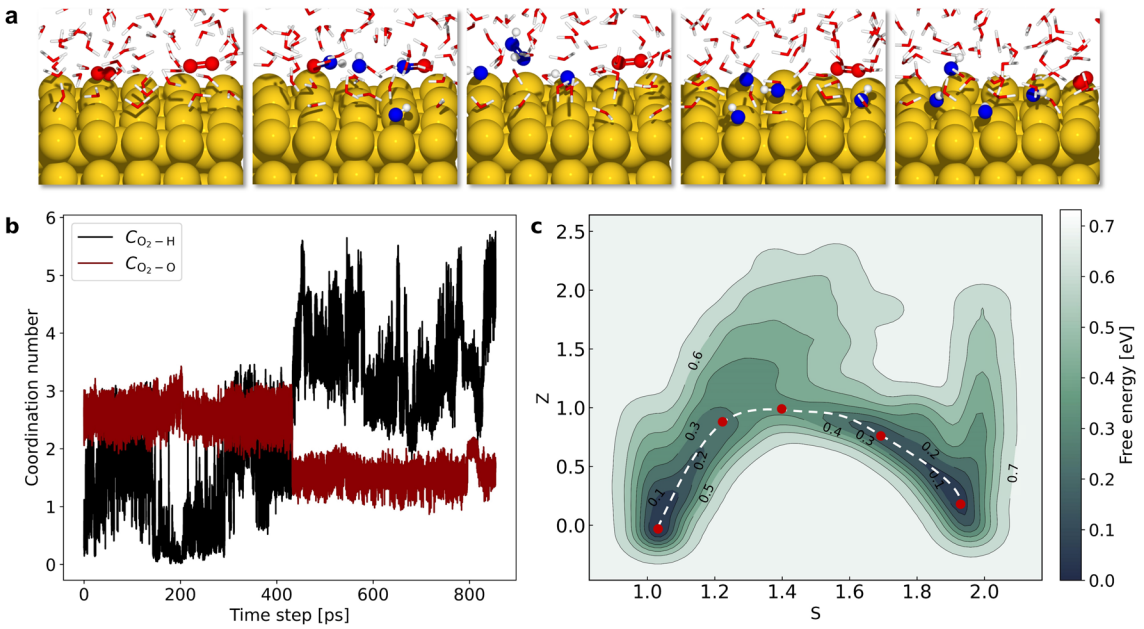

Figure 5.8: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(110) with two oxygen molecules described by path CVs.

energy basin is quite stable, encapsulating both the fourth and fifth points in the reaction pathway. The reaction barrier for the Au(110)-$2O_2/H_2O$ interface is slightly higher than that of Au(100)-$2O_2/H_2O$, being 0.32 eV.

Moving onto the Au(111)-$2O_2/H_2O$ interface, the simulation stoped at 665 ps, demonstrating the O-O bond dissociation of the first O_2 molecule at 46 ps as depicted in Figure 5.9b. This behavior aligns with the previous observations on the Au(100) and Au(110) interfaces, where the dissociation of the second O_2 molecule proved to be challenging. As demonstrated in Figure 5.9c, the free energy landscape showcased two major basins, with the first corresponding to the initial state Au(111)-$2O_2/H_2O$ and the second corresponding the final state Au(111)-($1O_2$+4OH)/H_2O. The presence of *H_2O_2 is more prominent on Au(111), expanding over a large area from the second point to the third point along the reaction trajectory. Amongst the interfaces studied, the Au(111)-$2O_2/H_2O$ interface exhibited the lowest reaction barrier, recorded at 0.21 eV.

A notable observation across all three interfaces is that the evaluated reaction barriers are lower in comparison to their counterparts in the single oxygen metadynamics case, indicating the facilitative role of co-adsorbed O_2 for ORR dynamics. Moreover, in each case, the second major basin, representing the co-adsorption of one oxygen molecule and

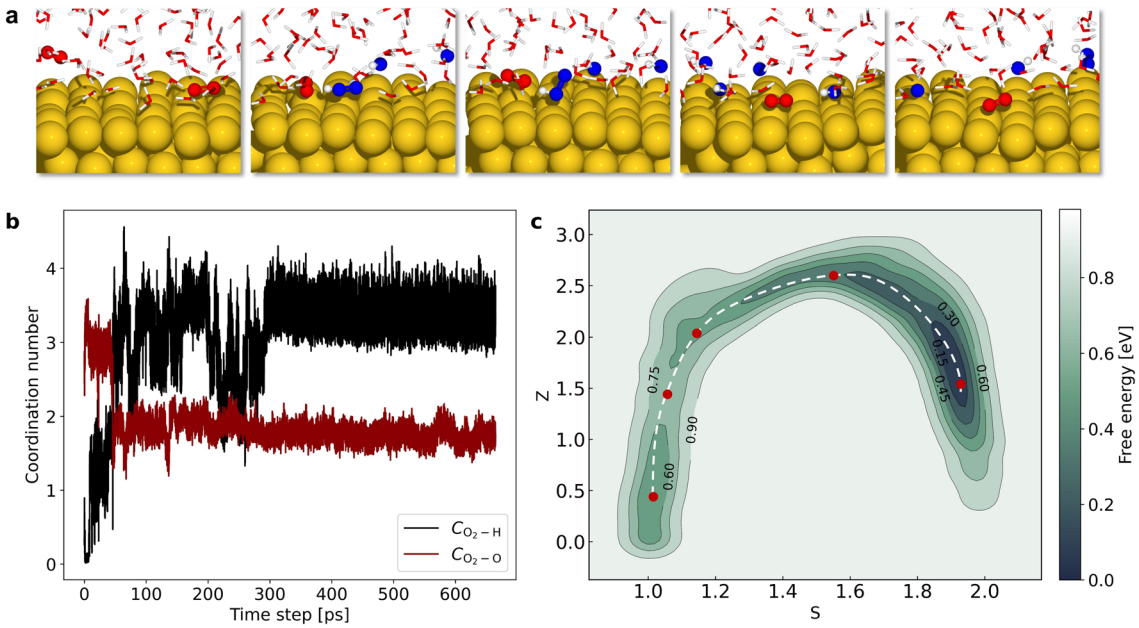

Figure 5.9: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(111) with two oxygen molecules described by path CVs.

four hydroxyl groups, was quite deep, demonstrating the remarkable stability of this state. This suggests that the presence of co-adsorbed hydroxyls can significantly influence the ORR on gold surfaces, potentially impeding the ORR process.

5.4 Conclusion and outlooks

In this comprehensive study, we have delved into the intricate dynamics of the oxygen reduction reaction (ORR) on prominent gold surfaces, specifically Au(100), Au(110), and Au(111). Our approach, which incorporated explicit solvents and utilized GNN-accelerated metadynamics, has provided a deep insight into the ORR mechanisms on these gold facets. Our regular MD simulations elucidate the dynamics at the gold-water interface, with an emphasis on the interactions involving adsorbed O_2 and *OH. The role of adsorbed *OH was found to be significant in influencing the water layer structure. Our systematic investigations have revealed the facet-dependent behavior of ORR, with particular emphasis on the stability and behavior of $*H_2O_2$ across different gold facets. For instance, the presence of $*H_2O_2$ on Au(111) and Au(110) is prominent, broadly existing in the reaction trajectory. However, the presence of $*H_2O_2$ on Au(100) is merely observed. This findings align well with experimental observations. Furthermore, our metadynamics results emphasized the role of co-adsorbed species on the ORR reactivity. The inclusion

of oxygen molecules can facilitate the reaction kinetics, while the co-adsorbed hydroxyl groups considerably impede the reaction process. The significant influence of adsorbed species, especially *OH, on the water layer structure underscores the need for a deeper understanding of their interactions and effects on other reaction intermediates. Additionally, extending this research to include different electrolytes can offer a broader perspective on how various solvents and ions modulate the ORR on gold surfaces. With the foundational knowledge established in this study, there lies the potential for the design and development of optimized gold-based electrocatalysts, paving the way for breakthroughs in electrocatalysis and related fields.

CHAPTER 6

Conclusions and Outlooks

In this thesis, we demonstrated the exploration and optimization of MLIPs for atomistic simulations, particularly focusing on the complex dynamics of the ORR on gold surfaces. If using traditional DFT calculations, the complex solid-liquid interface structures studied in this project require tremendous computational resources to rigorously and systematically study their dynamic natures. MLIPs emerged as a promising tool for extending the length-scale and time-scale of DFT calculations. However, the excellent performance of MLIPs is not taken for granted; behind this is the high-quality data that need to be collected. Gathering this data, however, is not cozy like walking in the park. It is often very tedious and one of my colleagues described this as "a monkey's job". Imagine a PhD candidate, passionate about the potential applications of MLIPs, only to find themselves stuck in the repetitive task of gathering data, rather than the exhilarating thrill of running and analyzing groundbreaking simulations.

The aim of the first work in this thesis is to free these smart folks from boring "monkey's job" and launch them into the fun world of real science. In this work, we developed an autonomous active learning workflow that can significantly reduce the human intervention and enhance the efficiency of data collecting. To improve the efficiency for collecting data, we designed several batch active learning strategies that can efficiently sample representative configurations from MLIPs-driven simulations. We demonstrated the power of different feature maps and selection algorithms on several benchmark datasets. All the tests illustrated that the LCMD+GRAD strategy exhibited the best performance, even featuring comparable low errors with half of the datapoints compared random selection. Additionally, to ensure simulations consistently operate within the application domain of MLIPs, we offer flexible uncertainty estimation tools. These tools prioritize accuracy, confidence-awareness and rapid simulation. By integrating these elements with the robust task scheduler `myqueue`, we make the automation of efficient MLIP generation possible.

The first work laid solid foundation for the subsequent two works. In the second work, we combined high-fidelity MLIPs and enhanced sampling technique together, to investigate the oxygen reduction dynamics at Au(100)-water interface. This work firstly revealed the full process of ORR in an atomistic level. We identified the associative mechanism of ORR on Au(100) and predicted a low free energy barrier of 0.3 eV. Both findings align well with experimental results. This framework can shed the light on modeling chemical reactions under complex ambient conditions.

In the third work, the simulations are extended to gold surfaces with enlarged simulation boxes. We systematically investigated the dynamics of systems across different slabs and adsorbates. It is demonstrated that the adsorbed hydroxyl groups have significant influence both on the interfacial structures and the ORR activity. The co-adsorbed hydroxyls prevented the O_2 molecules to be further protonated and reduced. We also found that the presence of $*H_2O_2$ is prominent on Au(110) and Au(111), indicating its indispensable role in the ORR on these slabs. This finding aligns well with experimental results and further demonstrated the power of our research framework.

We believe that the tools and methods presented in this thesis will gain broader acceptance and usage. We also hope our findings can pave the way for optimizing and discovering new catalysts to address the world's pressing challenges.

Bibliography

[1] Mark K Debe. "Electrocatalyst approaches and challenges for automotive fuel cells". In: *Nature* 486.7401 (2012), pp. 43–51.

[2] Hubert A Gasteiger and Nenad M Marković. "Just a dream—or future reality?" In: *science* 324.5923 (2009), pp. 48–49.

[3] Ambarish Kulkarni et al. "Understanding catalytic activity trends in the oxygen reduction reaction". In: *Chemical Reviews* 118.5 (2018), pp. 2302–2312.

[4] Minhua Shao et al. "Recent advances in electrocatalysts for oxygen reduction reaction". In: *Chemical reviews* 116.6 (2016), pp. 3594–3657.

[5] Liming Dai et al. "Metal-free catalysts for oxygen reduction reaction". In: *Chemical reviews* 115.11 (2015), pp. 4823–4892.

[6] Joseph H Montoya et al. "Materials for solar fuels and chemicals". In: *Nature materials* 16.1 (2017), pp. 70–81.

[7] Zhi Wei Seh et al. "Combining theory and experiment in electrocatalysis: Insights into materials design". In: *Science* 355.6321 (2017), eaad4998.

[8] Jens Kehlet Nørskov et al. "Origin of the overpotential for oxygen reduction at a fuel-cell cathode". In: *The Journal of Physical Chemistry B* 108.46 (2004), pp. 17886–17892.

[9] J Greeley et al. "Alloys of platinum and early transition metals as oxygen reduction electrocatalysts". In: *Nature chemistry* 1.7 (2009), pp. 552–556.

[10] Venkatasubramanian Viswanathan et al. "Universality in oxygen reduction electrocatalysis on metal surfaces". In: *Acs Catalysis* 2.8 (2012), pp. 1654–1660.

[11] Jan Rossmeisl et al. "Calculated phase diagrams for the electrochemical oxidation and reduction of water over Pt (111)". In: *The Journal of Physical Chemistry B* 110.43 (2006), pp. 21833–21839.

[12] Egill Skúlason et al. "Density functional theory calculations for the hydrogen evolution reaction in an electrochemical double layer on the Pt (111) electrode". In: *Physical Chemistry Chemical Physics* 9.25 (2007), pp. 3241–3250.

[13] Vladimir Tripkovic and Tejs Vegge. “Potential-and rate-determining step for oxygen reduction on Pt (111)”. In: *The Journal of Physical Chemistry C* 121.48 (2017), pp. 26785–26793.

[14] Heine A Hansen, Jan Rossmeisl, and Jens K Nørskov. “Surface Pourbaix diagrams and oxygen reduction activity of Pt, Ag and Ni (111) surfaces studied by DFT”. In: *Physical Chemistry Chemical Physics* 10.25 (2008), pp. 3722–3730.

[15] Yao Sha et al. “Theoretical study of solvent effects on the platinum-catalyzed oxygen reduction reaction”. In: *The Journal of Physical Chemistry Letters* 1.5 (2010), pp. 856–861.

[16] Yao Sha et al. “Oxygen hydration mechanism for the oxygen reduction reaction at Pt and Pd fuel cell catalysts”. In: *The Journal of Physical Chemistry Letters* 2.6 (2011), pp. 572–576.

[17] Alessandro Fortunelli et al. “Dramatic increase in the oxygen reduction reaction for platinum cathodes from tuning the solvent dielectric constant”. In: *Angewandte Chemie* 126.26 (2014), pp. 6787–6790.

[18] Stefan Ringe et al. “Implicit solvation methods for catalysis at electrified interfaces”. In: *Chemical Reviews* 122.12 (2021), pp. 10777–10820.

[19] Jeff Greeley and Jens K Nørskov. “Combinatorial density functional theory-based screening of surface alloys for the oxygen reduction reaction”. In: *The Journal of Physical Chemistry C* 113.12 (2009), pp. 4932–4939.

[20] Federico Calle-Vallejo, José Ignacio Martínez, and Jan Rossmeisl. “Density functional studies of functionalized graphitic materials with late transition metals for oxygen reduction reactions”. In: *Physical Chemistry Chemical Physics* 13.34 (2011), pp. 15639–15643.

[21] Vladimir Tripković et al. “The influence of particle shape and size on the activity of platinum nanoparticles for oxygen reduction reaction: a density functional theory study”. In: *Catalysis letters* 144.3 (2014), pp. 380–388.

[22] Xiaorong Zhu et al. “Activity origin and design principles for oxygen reduction on dual-metal-site catalysts: a combined density functional theory and machine learning study”. In: *The Journal of Physical Chemistry Letters* 10.24 (2019), pp. 7760–7766.

[23] Sara R Kelly et al. “Electric field effects in oxygen reduction kinetics: rationalizing pH dependence at the Pt (111), Au (111), and Au (100) electrodes”. In: *The Journal of Physical Chemistry C* 124.27 (2020), pp. 14581–14591.

[24] Nitish Govindarajan, Aoni Xu, and Karen Chan. “How pH affects electrochemical processes”. In: *Science* 375.6579 (2022), pp. 379–380.

[25] Jiang Li et al. “Modeling potential-dependent electrochemical activation barriers: revisiting the alkaline hydrogen evolution reaction”. In: *Journal of the American Chemical Society* 143.46 (2021), pp. 19341–19355.

[26] Anna Ignaczak, Elizabeth Santos, and Wolfgang Schmickler. “Oxygen reduction reaction on gold in alkaline solutions–The inner or outer sphere mechanisms in the light of recent achievements”. In: *Current Opinion in Electrochemistry* 14 (2019), pp. 180–185.

[27] Zhiyao Duan and Graeme Henkelman. “Theoretical resolution of the exceptional oxygen reduction activity of Au (100) in alkaline media”. In: *ACS Catalysis* 9.6 (2019), pp. 5567–5573.

[28] Paramaconi Rodriguez and Marc TM Koper. “Electrocatalysis on gold”. In: *Physical Chemistry Chemical Physics* 16.27 (2014), pp. 13583–13594.

[29] P Quaino et al. “Why is gold such a good catalyst for oxygen reduction in alkaline media?” In: *Angewandte Chemie International Edition* 51.52 (2012), pp. 12997–13000.

[30] Sergei Izvekov et al. “Ab initio molecular dynamics simulation of the Cu (110)–water interface”. In: *The Journal of Chemical Physics* 114.7 (2001), pp. 3248–3257.

[31] Li-Min Liu et al. “Structure and dynamics of liquid water on rutile TiO 2 (110)”. In: *Physical Review B* 82.16 (2010), p. 161415.

[32] Axel Groß et al. “Water structures at metal electrodes studied by ab initio molecular dynamics simulations”. In: *Journal of The Electrochemical Society* 161.8 (2014), E3015.

[33] Xueping Qin, Tejs Vegge, and Heine Anton Hansen. “Cation-coordinated inner-sphere CO2 electroreduction at Au–water interfaces”. In: *Journal of the American Chemical Society* 145.3 (2023), pp. 1897–1905.

[34] Phillips Hutchison et al. “Multilevel computational studies reveal the importance of axial ligand for oxygen reduction reaction on Fe–N–C materials”. In: *Journal of the American Chemical Society* 144.36 (2022), pp. 16524–16534.

[35] Tao Cheng et al. “Mechanism and kinetics of the electrocatalytic reaction responsible for the high cost of hydrogen fuel cells”. In: *Physical Chemistry Chemical Physics* 19.4 (2017), pp. 2666–2673.

[36] Tamio Ikeshoji and Minoru Otani. “Toward full simulation of the electrochemical oxygen reduction reaction on Pt using first-principles and kinetic calculations”. In: *Physical Chemistry Chemical Physics* 19.6 (2017), pp. 4447–4453.

[37] Henrik H Kristoffersen, Tejs Vegge, and Heine Anton Hansen. "OH formation and H 2 adsorption at the liquid water–Pt (111) interface". In: *Chemical science* 9.34 (2018), pp. 6912–6921.

[38] Volker L Deringer, Miguel A Caro, and Gábor Csányi. "Machine learning interatomic potentials as emerging tools for materials science". In: *Advanced Materials* 31.46 (2019), p. 1902765.

[39] Oliver T Unke et al. "Machine learning force fields". In: *Chemical Reviews* 121.16 (2021), pp. 10142–10186.

[40] Vanessa Quaranta, Matti Hellström, and Jörg Behler. "Proton-transfer mechanisms at the water–ZnO interface: The role of presolvation". In: *The journal of physical chemistry letters* 8.7 (2017), pp. 1476–1483.

[41] Vanessa Quaranta, Jörg Behler, and Matti Hellström. "Structure and dynamics of the liquid–water/zinc-oxide interface from machine learning potential simulations". In: *The Journal of Physical Chemistry C* 123.2 (2018), pp. 1293–1304.

[42] Jörg Behler. "First principles neural network potentials for reactive simulations of large molecular and condensed systems". In: *Angewandte Chemie International Edition* 56.42 (2017), pp. 12828–12840.

[43] Suresh Kondati Natarajan and Jörg Behler. "Neural network molecular dynamics simulations of solid–liquid interfaces: water at low-index copper surfaces". In: *Physical Chemistry Chemical Physics* 18.41 (2016), pp. 28704–28725.

[44] Patrick Rowe et al. "Development of a machine learning potential for graphene". In: *Physical Review B* 97.5 (2018), p. 054303.

[45] Patrick Rowe et al. "An accurate and transferable machine learning potential for carbon". In: *The Journal of Chemical Physics* 153.3 (2020).

[46] Volker L Deringer and Gábor Csányi. "Machine learning based interatomic potential for amorphous carbon". In: *Physical Review B* 95.9 (2017), p. 094203.

[47] Marco Eckhoff et al. "Closing the gap between theory and experiment for lithium manganese oxide spinels using a high-dimensional neural network potential". In: *Physical Review B* 102.17 (2020), p. 174102.

[48] Kazutoshi Miwa and Ryoji Asahi. "Molecular dynamics simulations with machine learning potential for Nb-doped lithium garnet-type oxide Li 7- x La 3 (Zr 2- x Nb x) O 12". In: *Physical Review Materials* 2.10 (2018), p. 105404.

[49] Jörg Behler and Michele Parrinello. "Generalized neural-network representation of high-dimensional potential-energy surfaces". In: *Physical review letters* 98.14 (2007), p. 146401.

[50] Jörg Behler. "Atom-centered symmetry functions for constructing high-dimensional neural network potentials". In: *The Journal of chemical physics* 134.7 (2011), p. 074106.

[51] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. "ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost". In: *Chemical science* 8.4 (2017), pp. 3192–3203.

[52] Xiang Gao et al. "TorchANI: A free and open source PyTorch-based deep learning implementation of the ANI neural network potentials". In: *Journal of chemical information and modeling* 60.7 (2020), pp. 3408–3415.

[53] Kun Yao et al. "The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics". In: *Chemical science* 9.8 (2018), pp. 2261–2269.

[54] Kyuhyun Lee et al. "SIMPLE-NN: An efficient package for training and executing neural-network interatomic potentials". In: *Computer Physics Communications* 242 (2019), pp. 95–103.

[55] Stefan Chmiela et al. "Towards exact molecular dynamics simulations with machine-learned force fields". In: *Nature communications* 9.1 (2018), p. 3887.

[56] Albert P Bartók et al. "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons". In: *Physical review letters* 104.13 (2010), p. 136403.

[57] Franco Scarselli et al. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80.

[58] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272.

[59] Kristof T Schütt et al. "Quantum-chemical insights from deep tensor neural networks". In: *Nature communications* 8.1 (2017), p. 13890.

[60] Oliver T Unke and Markus Meuwly. "PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges". In: *Journal of chemical theory and computation* 15.6 (2019), pp. 3678–3693.

[61] Kristof Schütt et al. "Schnet: A continuous-filter convolutional neural network for modeling quantum interactions". In: *Advances in neural information processing systems* 30 (2017).

[62] Kristof T Schütt et al. "Schnet–a deep learning architecture for molecules and materials". In: *The Journal of Chemical Physics* 148.24 (2018), p. 241722.

[63] Johannes Gasteiger, Janek Groß, and Stephan Günnemann. "Directional message passing for molecular graphs". In: *arXiv preprint arXiv:2003.03123* (2020).

[64] Nathaniel Thomas et al. “Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds”. In: *arXiv preprint arXiv:1802.08219* (2018).

[65] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E (n) equivariant graph neural networks”. In: *International conference on machine learning*. PMLR. 2021, pp. 9323–9332.

[66] Simon Batzner et al. “E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials”. In: *Nature Communications* 13.1 (2022), pp. 1–11.

[67] Ilyes Batatia et al. “MACE: Higher order equivariant message passing neural networks for fast and accurate force fields”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 11423–11436.

[68] Kevin Tran et al. “Methods for comparing uncertainty quantifications for material property predictions”. In: *Machine Learning: Science and Technology* 1.2 (2020), p. 025006.

[69] Albert P Bartók et al. “Machine learning a general-purpose interatomic potential for silicon”. In: *Physical Review X* 8.4 (2018), p. 041048.

[70] Volker L Deringer et al. “Gaussian process regression for materials and molecules”. In: *Chemical Reviews* 121.16 (2021), pp. 10073–10141.

[71] Felix Musil et al. “Fast and accurate uncertainty estimation in chemical machine learning”. In: *Journal of chemical theory and computation* 15.2 (2019), pp. 906–915.

[72] Johannes Gasteiger et al. “Fast and uncertainty-aware directional message passing for non-equilibrium molecules”. In: *arXiv preprint arXiv:2011.14115* (2020).

[73] Jörg Behler. “Constructing high-dimensional neural network potentials: a tutorial review”. In: *International Journal of Quantum Chemistry* 115.16 (2015), pp. 1032–1050.

[74] Justin S Smith et al. “Less is more: Sampling chemical space with active learning”. In: *The Journal of chemical physics* 148.24 (2018).

[75] Yarin Gal and Zoubin Ghahramani. “Dropout as a bayesian approximation: Representing model uncertainty in deep learning”. In: *international conference on machine learning*. PMLR. 2016, pp. 1050–1059.

[76] Mingjian Wen and Ellad B Tadmor. “Uncertainty quantification in molecular simulations with dropout neural network potentials”. In: *npj Computational Materials* 6.1 (2020), pp. 1–10.

[77] Christoph Schran et al. “Machine learning potentials for complex aqueous systems made simple”. In: *Proceedings of the National Academy of Sciences* 118.38 (2021).

[78] Jonathan Vandermause et al. “Active learning of reactive Bayesian force fields applied to heterogeneous catalysis dynamics of H/Pt”. In: *Nature Communications* 13.1 (2022), p. 5183.

[79] Jordan T Ash et al. “Deep batch active learning by diverse, uncertain gradient lower bounds”. In: *arXiv preprint arXiv:1906.03671* (2019).

[80] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. “Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning”. In: *Advances in neural information processing systems* 32 (2019).

[81] Gui Citovsky et al. “Batch active learning at scale”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 11933–11944.

[82] David Holzmüller et al. “A framework and benchmark for deep batch active learning for regression”. In: *Journal of Machine Learning Research* 24.164 (2023), pp. 1–81.

[83] Richard M Martin. *Electronic structure: basic theory and practical methods*. Cambridge university press, 2020.

[84] Jorge Kohanoff. *Electronic structure calculations for solids and molecules: theory and computational methods*. Cambridge university press, 2006.

[85] David S Sholl and Janice A Steckel. *Density functional theory: a practical introduction*. John Wiley & Sons, 2022.

[86] Max Born and Robert Oppenheimer. “On the quantum theory of molecules”. In: *Quantum Chemistry: Classic Scientific Papers*. World Scientific, 2000, pp. 1–24.

[87] Pierre Hohenberg and Walter Kohn. “Inhomogeneous electron gas”. In: *Physical review* 136.3B (1964), B864.

[88] Walter Kohn and Lu Jeu Sham. “Self-consistent equations including exchange and correlation effects”. In: *Physical review* 140.4A (1965), A1133.

[89] Viraht Sahni, K-P Bohnen, and Manoj K Harbola. “Analysis of the local-density approximation of density-functional theory”. In: *Physical Review A* 37.6 (1988), p. 1895.

[90] John P Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized gradient approximation made simple”. In: *Physical review letters* 77.18 (1996), p. 3865.

[91] Axel D Becke. “Correlation energy of an inhomogeneous electron gas: A coordinate-space model”. In: *The Journal of chemical physics* 88.2 (1988), pp. 1053–1062.

[92] Chengteh Lee, Weitao Yang, and Robert G Parr. “Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density”. In: *Physical review B* 37.2 (1988), p. 785.

[93] John P Perdew and Yue Wang. “Accurate and simple analytic representation of the electron-gas correlation energy”. In: *Physical review B* 45.23 (1992), p. 13244.

[94] Georg Kresse and Jürgen Hafner. “Ab initio molecular dynamics for liquid metals”. In: *Physical review B* 47.1 (1993), p. 558.

[95] Georg Kresse and Jürgen Hafner. “Ab initio molecular-dynamics simulation of the liquid-metal–amorphous-semiconductor transition in germanium”. In: *Physical Review B* 49.20 (1994), p. 14251.

[96] Georg Kresse and Jürgen Furthmüller. “Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set”. In: *Computational materials science* 6.1 (1996), pp. 15–50.

[97] Georg Kresse and Jürgen Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. In: *Physical review B* 54.16 (1996), p. 11169.

[98] Ask Hjorth Larsen et al. “The atomic simulation environment—a Python library for working with atoms”. In: *Journal of Physics: Condensed Matter* 29.27 (2017), p. 273002.

[99] Stefan Grimme et al. “A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu”. In: *The Journal of chemical physics* 132.15 (2010), p. 154104.

[100] Peter E Blöchl. “Projector augmented-wave method”. In: *Physical review B* 50.24 (1994), p. 17953.

[101] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

[102] Loup Verlet. “Computer” experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules”. In: *Physical review* 159.1 (1967), p. 98.

[103] Richard Car and Mark Parrinello. “Unified approach for molecular dynamics and density-functional theory”. In: *Physical review letters* 55.22 (1985), p. 2471.

[104] Hans C Andersen. “Molecular dynamics simulations at constant pressure and/or temperature”. In: *The Journal of chemical physics* 72.4 (1980), pp. 2384–2393.

[105] T Schneider and E Stoll. “Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions”. In: *Physical Review B* 17.3 (1978), p. 1302.

[106] Shuichi Nosé. “A unified formulation of the constant temperature molecular dynamics methods”. In: *The Journal of chemical physics* 81.1 (1984), pp. 511–519.

[107] William G Hoover. “Constant-pressure equations of motion”. In: *Physical Review A* 34.3 (1986), p. 2499.

[108] Dominik Marx and Jürg Hutter. *Ab initio molecular dynamics: basic theory and advanced methods*. Cambridge University Press, 2009.

[109] Jorg Behler. "Four generations of high-dimensional neural network potentials". In: *Chemical Reviews* 121.16 (2021), pp. 10037–10072.

[110] Kristof Schütt, Oliver Unke, and Michael Gastegger. "Equivariant message passing for the prediction of tensorial properties and molecular spectra". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9377–9388.

[111] Stefan Chmiela et al. "Machine learning of accurate energy-conserving molecular force fields". In: *Science advances* 3.5 (2017), e1603015.

[112] Raphael JL Townshend et al. "Geometric prediction: Moving beyond scalars". In: *arXiv preprint arXiv:2006.14163* (2020).

[113] Alessandro Laio and Michele Parrinello. "Escaping free-energy minima". In: *Proceedings of the National Academy of Sciences* 99.20 (2002), pp. 12562–12566.

[114] Giovanni Bussi, Alessandro Laio, and Michele Parrinello. "Equilibrium free energies from nonequilibrium metadynamics". In: *Physical review letters* 96.9 (2006), p. 090601.

[115] Giovanni Bussi and Alessandro Laio. "Using metadynamics to explore complex free-energy landscapes". In: *Nature Reviews Physics* 2.4 (2020), pp. 200–212.

[116] Davide Branduardi, Francesco Luigi Gervasio, and Michele Parrinello. "From A to B in free energy space". In: *The Journal of chemical physics* 126.5 (2007), p. 054103.

[117] Daniel Sucerquia et al. "Ab initio metadynamics determination of temperature-dependent free-energy landscape in ultrasmall silver clusters". In: *The Journal of Chemical Physics* 156.15 (2022), p. 154301.

[118] Glenn M Torrie and John P Valleau. "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling". In: *Journal of Computational Physics* 23.2 (1977), pp. 187–199.

[119] Shankar Kumar et al. "The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method". In: *Journal of computational chemistry* 13.8 (1992), pp. 1011–1021.

[120] Paolo Raiteri et al. "Efficient reconstruction of complex free energy landscapes by multiple walkers metadynamics". In: *The journal of physical chemistry B* 110.8 (2006), pp. 3533–3539.

[121] Gareth A Tribello et al. "PLUMED 2: New feathers for an old bird". In: *Computer physics communications* 185.2 (2014), pp. 604–613.

[122] "Promoting transparency and reproducibility in enhanced molecular simulations". In: *Nature methods* 16.8 (2019), pp. 670–673.

[123] Jens Jørgen Mortensen, Morten Gjerding, and Kristian Sommer Thygesen. "MyQueue: Task and workflow scheduling system". In: *Journal of Open Source Software* 5.45 (2020), p. 1844.

[124] Jakob Schiøtz. *Asap*. https://gitlab.com/asap. 2023.

[125] James Kermode and Lars Pastewka. *Matscipy*. https://github.com/libAtoms/matscipy. 2019.

[126] Raimondas Galvelis and Peter Eastman. *NNPOps*. https://github.com/openmm/nnpops. 2020.

[127] Adam Paszke et al. "Automatic differentiation in pytorch". In: (2017).

[128] Franz Knuth et al. "All-electron formalism for total energy strain derivatives and stress tensor components for numeric atom-centered orbitals". In: *Computer Physics Communications* 190 (2015), pp. 33–50.

[129] Jinsol Lee and Ghassan AlRegib. "Gradients as a measure of uncertainty in neural networks". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 2416–2420.

[130] Arthur Jacot, Franck Gabriel, and Clément Hongler. "Neural tangent kernel: Convergence and generalization in neural networks". In: *Advances in neural information processing systems* 31 (2018).

[131] Viktor Zaverkin and Johannes Kästner. "Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design". In: *Machine Learning: Science and Technology* 2.3 (2021), p. 035009.

[132] Sandip De et al. "Comparing molecules and solids across structural and alchemical space". In: *Physical Chemistry Chemical Physics* 18.20 (2016), pp. 13754–13769.

[133] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[134] Kimin Lee et al. "A simple unified framework for detecting out-of-distribution samples and adversarial attacks". In: *Advances in neural information processing systems* 31 (2018).

[135] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[136] Viktor Zaverkin et al. "Exploring chemical and conformational spaces by batch mode deep active learning". In: *Digital Discovery* 1.5 (2022), pp. 605–620.

[137] Maryam Pazouki and Robert Schaback. "Bases for kernel-based spaces". In: *Journal of Computational and Applied Mathematics* 236.4 (2011), pp. 575–588.

[138] Evgeny V Podryabinkin et al. “Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning”. In: *Physical Review B* 99.6 (2019), p. 064114.

[139] Yury Lysogorskiy et al. “Active learning strategies for atomic cluster expansion models”. In: *Physical Review Materials* 7.4 (2023), p. 043801.

[140] Cheol Woo Park et al. “Accurate and scalable multi-element graph neural network force field and molecular dynamics with direct force architecture”. In: *arXiv preprint arXiv:2007.14444* (2020).

[141] Leo Breiman. “Bagging predictors”. In: *Machine learning* 24 (1996), pp. 123–140.

[142] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles”. In: *Advances in neural information processing systems* 30 (2017).

[143] Tom Wollschläger et al. “Uncertainty Estimation for Molecules: Desiderata and Methods”. In: (2023).

[144] Albert Zhu et al. “Fast uncertainty estimates in deep learning interatomic potentials”. In: *The Journal of Chemical Physics* 158.16 (2023).

[145] Omry Yadan. “Hydra-a framework for elegantly configuring complex applications”. In: *Github* 2 (2019), p. 5.

[146] William A Falcon. “Pytorch lightning”. In: *GitHub* 3 (2019).

[147] August EG Mikkelsen et al. “Structure and energetics of liquid water–hydroxyl layers on Pt (111)”. In: *Physical Chemistry Chemical Physics* 24.17 (2022), pp. 9885–9890.

[148] Tian Sheng and Shi-Gang Sun. “Free energy landscape of electrocatalytic CO2 reduction to CO on aqueous FeN4 center embedded graphene studied by ab initio molecular dynamics simulations”. In: *Chemical Physics Letters* 688 (2017), pp. 37–42.

[149] Sung Sakong and Axel Groß. “Water structures on a Pt (111) electrode from ab initio molecular dynamic simulations for a variety of electrochemical conditions”. In: *Physical Chemistry Chemical Physics* 22.19 (2020), pp. 10431–10437.

[150] Tao Cheng, Hai Xiao, and William A Goddard III. “Reaction mechanisms for the electrochemical reduction of CO2 to CO and formate on the Cu (100) surface at 298 K from quantum mechanics free energy calculations with explicit water”. In: *Journal of the American Chemical Society* 138.42 (2016), pp. 13802–13805.

[151] Jeffrey A Herron, Yoshitada Morikawa, and Manos Mavrikakis. “Ab initio molecular dynamics of solvation effects on reactivity at electrified interfaces”. In: *Proceedings of the National Academy of Sciences* 113.34 (2016), E4937–E4945.

[152] Xueping Qin, Tejs Vegge, and Heine Anton Hansen. "CO2 activation at Au (110)–water interfaces: An ab initio molecular dynamics study". In: *The Journal of Chemical Physics* 155.13 (2021), p. 134703.

[153] John R Kitchin. "Machine learning in catalysis". In: *Nature Catalysis* 1.4 (2018), pp. 230–232.

[154] Pascal Friederich et al. "Machine learning dihydrogen activation in the chemical space surrounding Vaska's complex". In: *Chemical Science* 11.18 (2020), pp. 4584–4601.

[155] Benjamin Meyer et al. "Machine learning meets volcano plots: computational discovery of cross-coupling catalysts". In: *Chemical science* 9.35 (2018), pp. 7069–7077.

[156] Marco Foscato and Vidar R Jensen. "Automated in silico design of homogeneous catalysts". In: *ACS catalysis* 10.3 (2020), pp. 2354–2377.

[157] Jiayan Xu, Xiao-Ming Cao, and P Hu. "Accelerating metadynamics-based free-energy calculations with adaptive machine learning potentials". In: *Journal of chemical theory and computation* 17.7 (2021), pp. 4465–4476.

[158] Noam Bernstein, Gábor Csányi, and Volker L Deringer. "De novo exploration and self-guided learning of potential-energy surfaces". In: *npj Computational Materials* 5.1 (2019), pp. 1–9.

[159] Sina Stocker et al. "Machine learning in chemical reaction space". In: *Nature communications* 11.1 (2020), pp. 1–11.

[160] Mathias Schreiner et al. "NeuralNEB—Neural Networks can find reaction paths fast". In: *Machine Learning: Science and Technology* 3.4 (2022), p. 045022.

[161] Albert P Bartók, Risi Kondor, and Gábor Csányi. "On representing chemical environments". In: *Physical Review B* 87.18 (2013), p. 184115.

[162] Suresh Kondati Natarajan and Jörg Behler. "Self-diffusion of surface defects at copper–water interfaces". In: *The Journal of Physical Chemistry C* 121.8 (2017), pp. 4368–4383.

[163] Hossein Ghorbanfekr, Jörg Behler, and François M Peeters. "Insights into water permeation through hBN nanocapillaries by ab initio machine learning molecular dynamics simulations". In: *The Journal of Physical Chemistry Letters* 11.17 (2020), pp. 7363–7370.

[164] Manyi Yang et al. "Using metadynamics to build neural network potentials for reactive events: the case of urea decomposition in water". In: *Catalysis Today* 387 (2022), pp. 143–149.

[165] Atsushi Urakawa et al. “Towards a rational design of ruthenium CO2 hydrogenation catalysts by ab initio metadynamics”. In: *Chemistry–A European Journal* 13.24 (2007), pp. 6828–6840.

[166] Alessandro Laio and Francesco L Gervasio. “Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science”. In: *Reports on Progress in Physics* 71.12 (2008), p. 126601.

[167] Michael W Mahoney and Petros Drineas. “CUR matrix decompositions for improved data analysis”. In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702.

[168] Changsheng Li et al. “Joint active learning with feature selection via cur matrix decomposition”. In: *IEEE transactions on pattern analysis and machine intelligence* 41.6 (2018), pp. 1382–1396.

[169] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. “Query by committee”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 287–294.

[170] Jonas Busk et al. “Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks”. In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015012.

[171] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[172] Adam Paszke et al. “Pytorch: An imperative style, high-performance deep learning library”. In: *Advances in neural information processing systems* 32 (2019).

[173] Hendrik J Monkhorst and James D Pack. “Special points for Brillouin-zone integrations”. In: *Physical review B* 13.12 (1976), p. 5188.

[174] Simone Pezzotti, Alessandra Serva, and Marie-Pierre Gaigeot. “2D-HB-Network at the air-water interface: A structural and dynamical characterization by means of ab initio and classical molecular dynamics simulations”. In: *The Journal of chemical physics* 148.17 (2018), p. 174701.

[175] Hagai B Perets et al. “Realization of quantum walks with negligible decoherence in waveguide lattices”. In: *Physical review letters* 100.17 (2008), p. 170506.

[176] Raghunathan Ramakrishnan et al. “Quantum chemistry structures and properties of 134 kilo molecules”. In: *Scientific data* 1.1 (2014), pp. 1–7.

[177] Johannes Gasteiger, Florian Becker, and Stephan Günnemann. “Gemnet: Universal directional graph neural networks for molecules”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 6790–6802.

[178] Weihua Hu et al. “Forcenet: A graph neural network for large-scale quantum calculations”. In: *arXiv preprint arXiv:2103.01436* (2021).

[179] Anna Ignaczak et al. “Oxygen reduction in alkaline media—a discussion”. In: *Electrocatalysis* 8.6 (2017), pp. 554–564.

[180] Aleksej Goduljan et al. “Oxygen Reduction on Ag (100) in Alkaline Solutions—A Theoretical Study”. In: *ChemPhysChem* 17.4 (2016), pp. 500–505.

[181] Fang Lu et al. “Surface proton transfer promotes four-electron oxygen reduction on gold nanocrystal surfaces in alkaline solution”. In: *Journal of the American Chemical Society* 139.21 (2017), pp. 7310–7317.

[182] Jakub Staszak-Jirkovský et al. “Water as a promoter and catalyst for dioxygen electrochemistry in aqueous and organic media”. In: *Acs Catalysis* 5.11 (2015), pp. 6600–6607.

[183] Nuria Lopez et al. “On the origin of the catalytic activity of gold nanoparticles for low-temperature CO oxidation”. In: *Journal of Catalysis* 223.1 (2004), pp. 232–235.

[184] Manos Mavrikakis, Per Stoltze, and Jens Kehlet Nørskov. “Making gold less noble”. In: *Catalysis letters* 64 (2000), pp. 101–106.

[185] Paramaconi Rodriguez, Youngkook Kwon, and Marc TM Koper. “The promoting effect of adsorbed carbon monoxide on the oxidation of alcohols on a gold catalyst”. In: *Nature chemistry* 4.3 (2012), pp. 177–182.

[186] A Stephen K Hashmi and Graham J Hutchings. “Gold catalysis”. In: *Angewandte Chemie International Edition* 45.47 (2006), pp. 7896–7936.

[187] Bjørk Hammer and Jens K Norskov. “Why gold is the noblest of all the metals”. In: *Nature* 376.6537 (1995), pp. 238–240.

[188] Masatake Haruta et al. “Gold catalysts prepared by coprecipitation for low-temperature oxidation of hydrogen and of carbon monoxide”. In: *Journal of catalysis* 115.2 (1989), pp. 301–309.

[189] Masatake Haruta. “When gold is not noble: catalysis by nanoparticles”. In: *The chemical record* 3.2 (2003), pp. 75–87.

[190] Masatake Haruta. “Gold rush”. In: *Nature* 437.7062 (2005), pp. 1098–1099.

[191] Graham J Hutchings, Mathias Brust, and Hubert Schmidbaur. “Gold—an introductory perspective”. In: *Chemical Society Reviews* 37.9 (2008), pp. 1759–1765.

[192] TV Choudhary and DW Goodman. “Catalytically active gold: The role of cluster morphology”. In: *Applied Catalysis A: General* 291.1-2 (2005), pp. 32–36.

[193] Tamao Ishida et al. “Influence of the support and the size of gold clusters on catalytic activity for glucose oxidation”. In: *Angewandte Chemie International Edition* 47.48 (2008), pp. 9265–9268.

[194] Vojislav R Stamenkovic et al. “Energy and fuels from electrochemical interfaces”. In: *Nature materials* 16.1 (2017), pp. 57–69.

[195] Xin Yang et al. “Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces”. In: *Chemical Science* 14.14 (2023), pp. 3913–3922.

[196] Axel Groß and Sung Sakong. “Ab initio simulations of water/metal interfaces”. In: *Chemical Reviews* 122.12 (2022), pp. 10746–10776.

[197] Ri He et al. “Structural phase transitions in SrTi O 3 from deep potential molecular dynamics”. In: *Physical Review B* 105.6 (2022), p. 064104.

[198] Yuan-Bin Liu et al. “Machine learning interatomic potential developed for molecular simulations on thermal properties of β-Ga2O3”. In: *The Journal of Chemical Physics* 153.14 (2020), p. 144501.

[199] ERM Davidson et al. “Grand canonical approach to modeling hydrogen trapping at vacancies in α- Fe”. In: *Physical Review Materials* 4.6 (2020), p. 063804.

[200] Lowik Chanussot et al. “Open catalyst 2020 (OC20) dataset and community challenges”. In: *ACS Catalysis* 11.10 (2021), pp. 6059–6072.

[201] Johannes Gasteiger et al. “How Do Graph Networks Generalize to Large and Diverse Molecular Systems?” In: *arXiv preprint arXiv:2204.02782* (2022).

[202] Zijie Li et al. “Graph neural networks accelerated molecular dynamics”. In: *The Journal of Chemical Physics* 156.14 (2022), p. 144103.

Appendix A

Supplementary materials for Chapter 3: Automated active learning workflow

Supplementary Tables

Table A.1: Mean absolute errors (MAE), number of parameters, interence time for each model.

Model	Layer	Node	Inference time (ms)	#parameters	Energy MAE kcal $\cdot$mol^{-1}	Forces MAE kcal $\cdot$mol^{-1} $\cdot$ Å^{-1}
nequip(l=2)	3	64	37.054	761576	0.149	0.210
	4	64	58.620	1208552	0.139	0.166
	5	64	80.547	1655528	0.132	0.148
	6	64	101.602	2102504	0.129	0.144
	4	32	50.176	392824	0.142	0.174
	4	64	60.090	1208552	0.139	0.166
	4	96	73.762	2465624	0.130	0.135
	4	128	91.473	4164040	0.128	0.133
nequip(l=3)	3	64	107.677	1441512	0.141	0.183
	4	64	184.664	2306280	0.134	0.148
	5	64	262.095	3171048	0.129	0.135
	6	64	338.976	4035816	0.128	0.131
	4	32	109.880	764536	0.131	0.147
	4	64	184.245	2306280	0.134	0.148
	4	96	265.594	4643672	0.129	0.123
	4	128	343.956	7776712	0.128	0.121
mace(l=2)	3	64	31.100	381480	0.156	0.219
	4	64	49.463	555048	0.162	0.187
	5	64	60.733	728616	0.144	0.167
	6	64	73.655	902184	0.150	0.165
	4	32	46.367	212504	0.181	0.240
	4	64	46.391	555048	0.162	0.187
	4	96	46.843	1062456	0.150	0.164
	4	128	50.389	1734728	0.162	0.134
mace(l=3)	3	64	53.003	679656	0.148	0.201
	4	64	74.090	1003112	0.152	0.169
	5	64	96.710	1326568	0.139	0.155
	6	64	120.201	1650024	0.137	0.150
	4	32	69.652	386360	0.176	0.213
	4	64	73.907	1003112	0.157	0.169
	4	96	102.646	1885080	0.147	0.152
	4	128	129.537	3032264	0.148	0.149
painn	3	64	7.497	161025	0.150	0.271
	4	64	9.864	210753	0.145	0.253
	5	64	12.094	260481	0.143	0.253
	6	64	14.434	310209	0.144	0.249
	4	32	10.068	59297	0.180	0.339
	4	64	9.814	210753	0.145	0.253
	4	96	9.804	454369	0.140	0.243
	4	128	10.331	790145	0.134	0.216

Table A.2: Inference time for each uncertainty estimation method

	Ensemble	Mahalanobis	MCD
Runtime [ms]	42.5 ± 3.92	8.9 ± 0.86	38.4 ± 3.21
#Trainings	5	1	1
#Predictions	5	1	5

Supplementary Figures

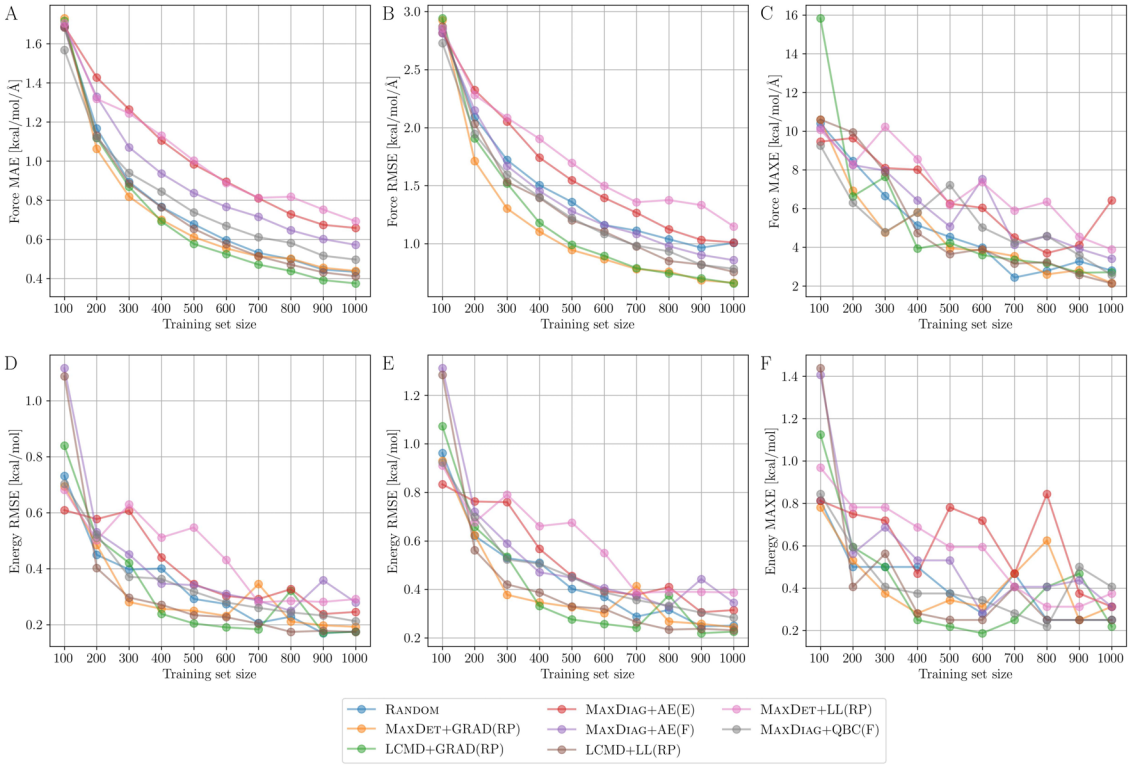

Figure A.1: Learning curves for the aspirin molecule data from MD17 dataset. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces, (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

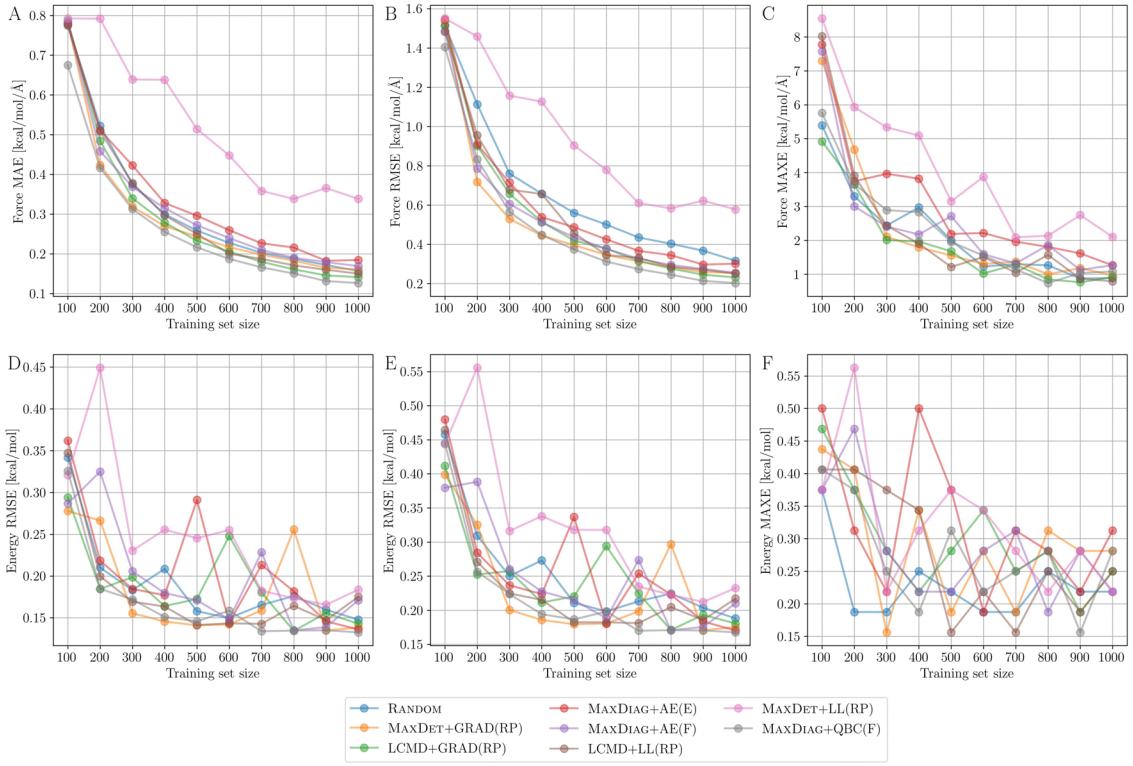

Figure A.2: Learning curves for the azobenzene molecule data from MD17 dataset.

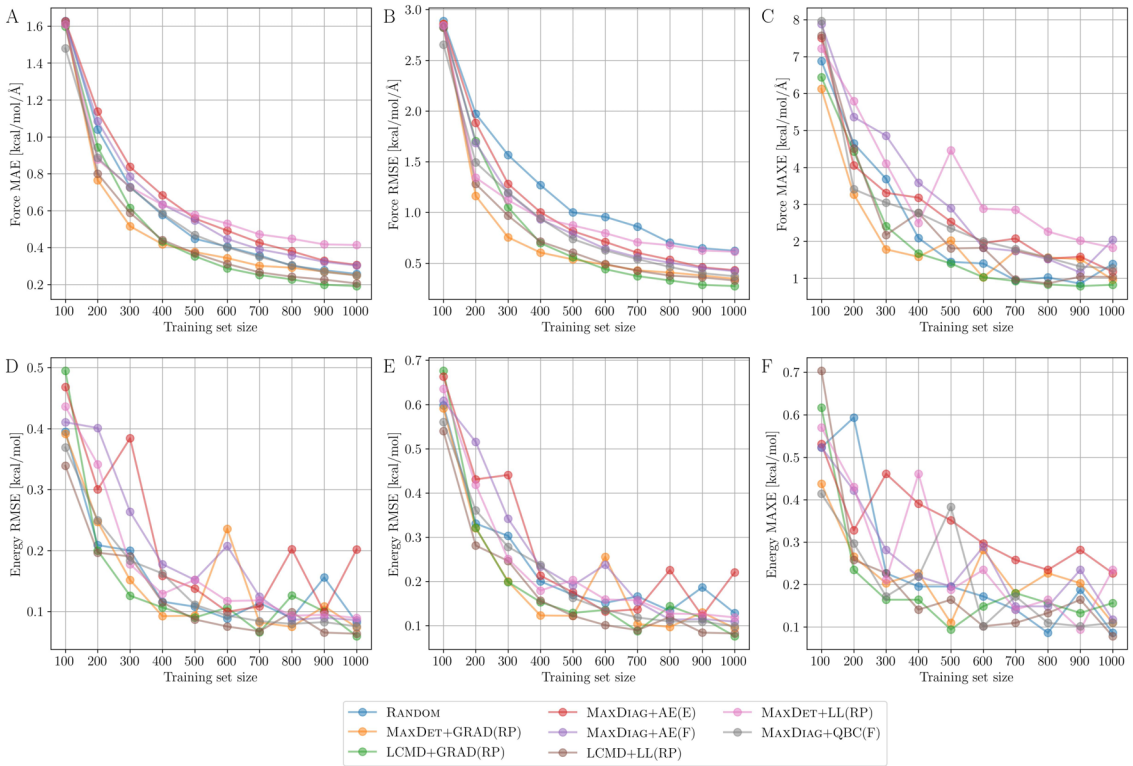

Figure A.3: Learning curves for the ethanol molecule data from MD17 dataset.

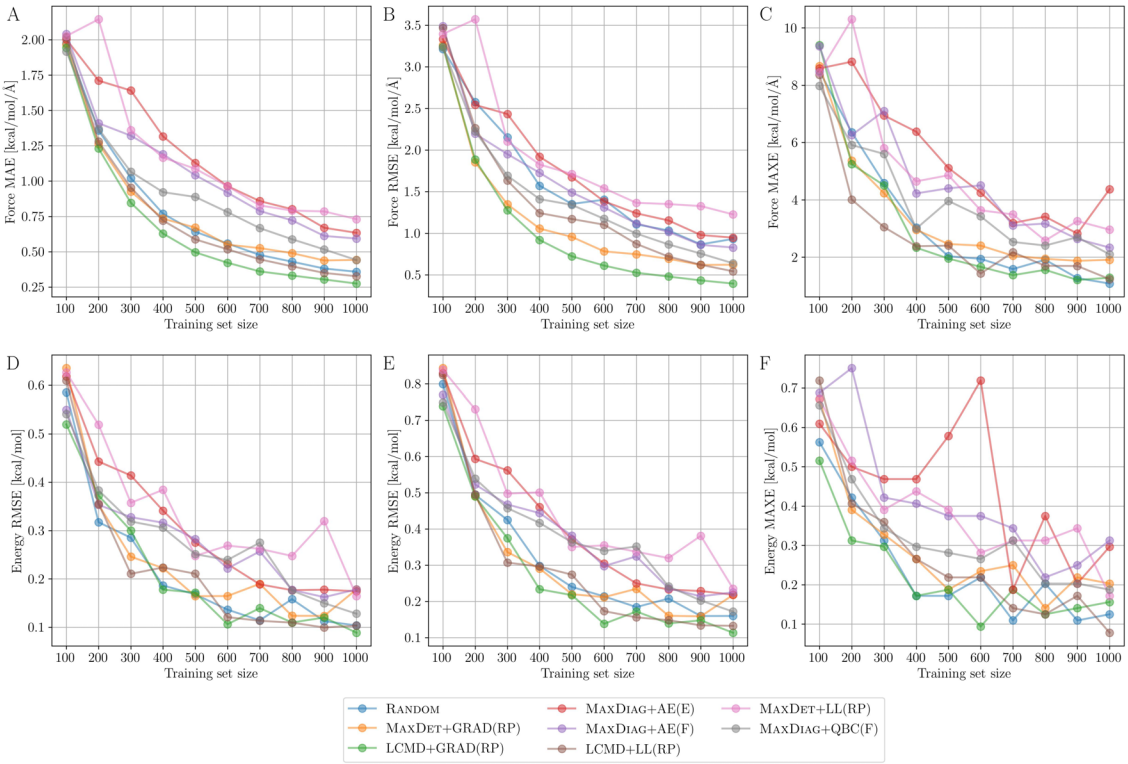

Figure A.4: Learning curves for the malonaldehyde molecule data from MD17 dataset.

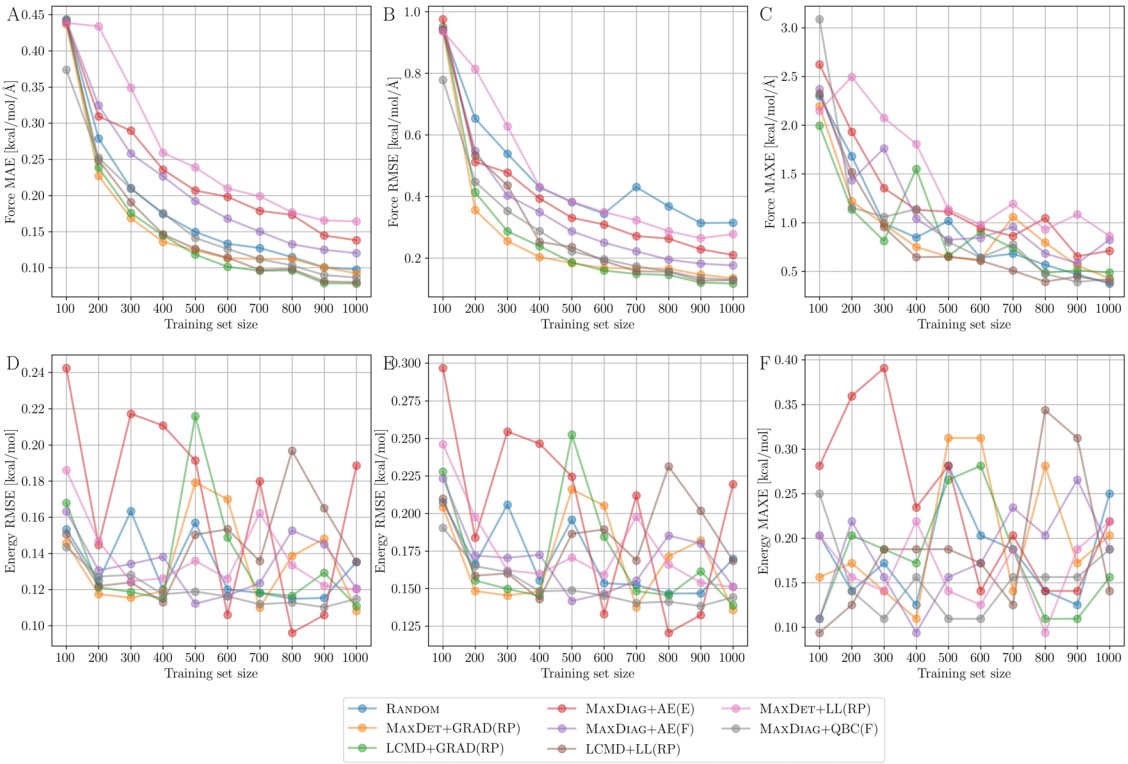

Figure A.5: Learning curves for the naphthalene molecule data from MD17 dataset.

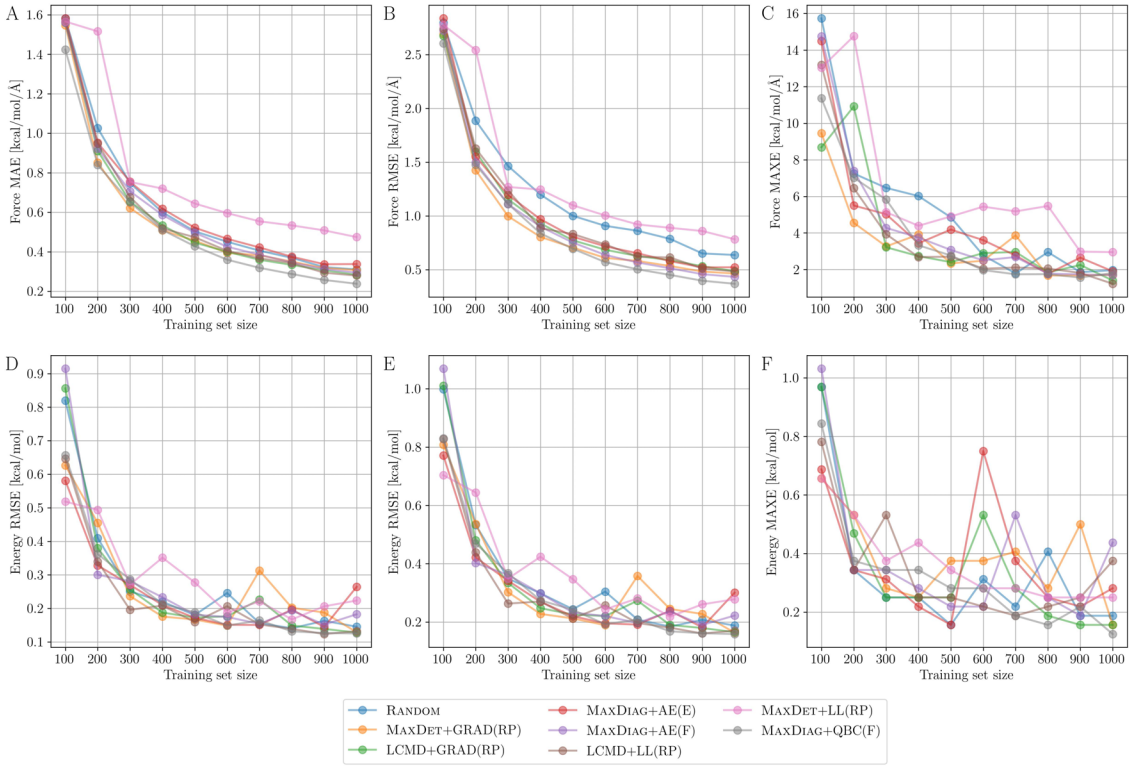

Figure A.6: Learning curves for the paracetamol molecule data from MD17 dataset.

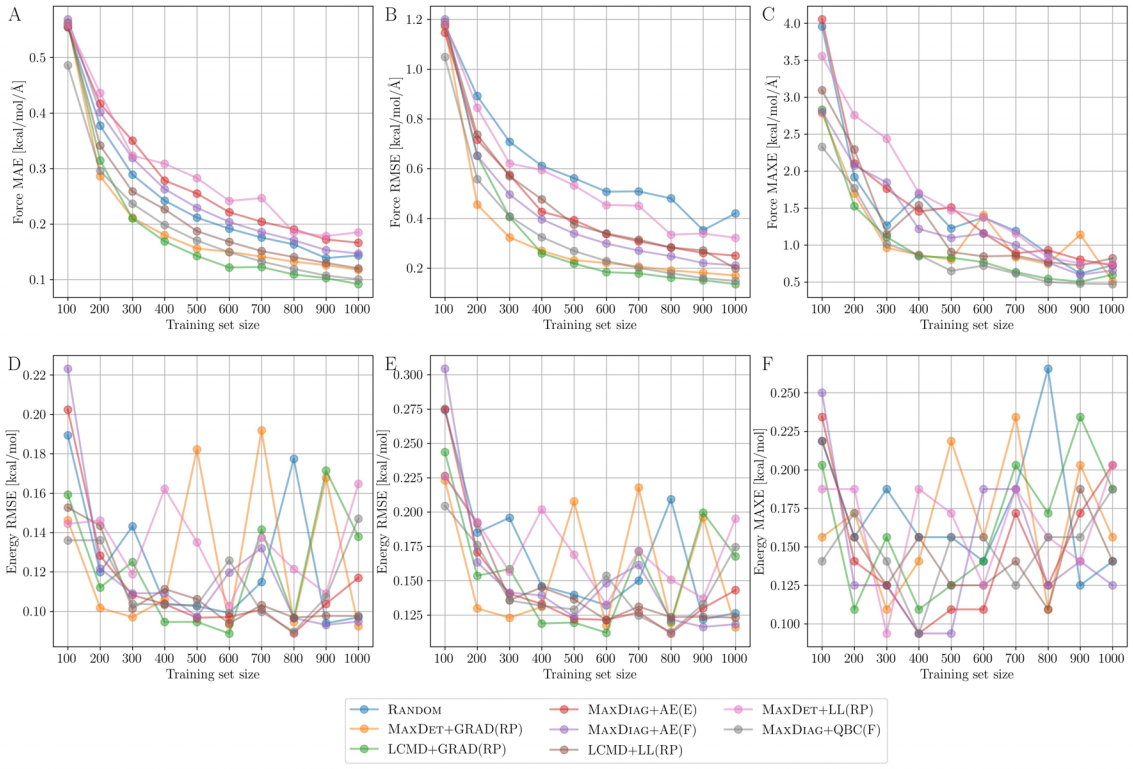

Figure A.7: Learning curves for the toluene molecule data from MD17 dataset.

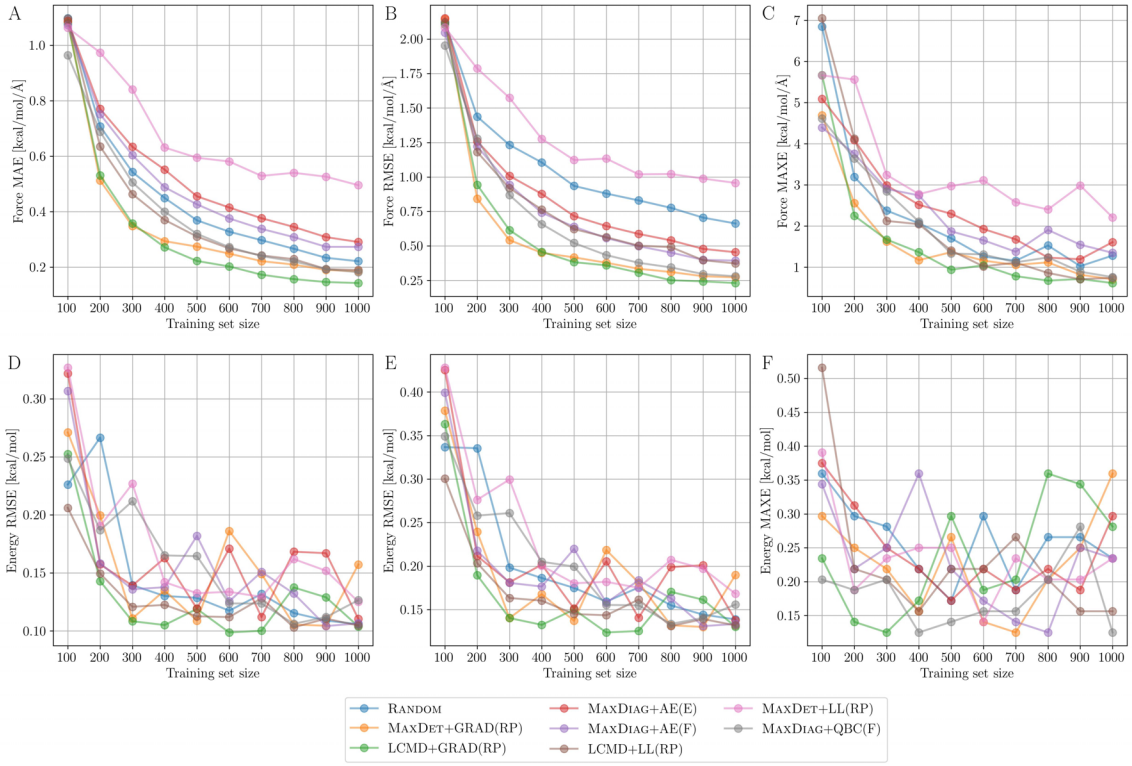

Figure A.8: Learning curves for the uracil molecule data from MD17 dataset.

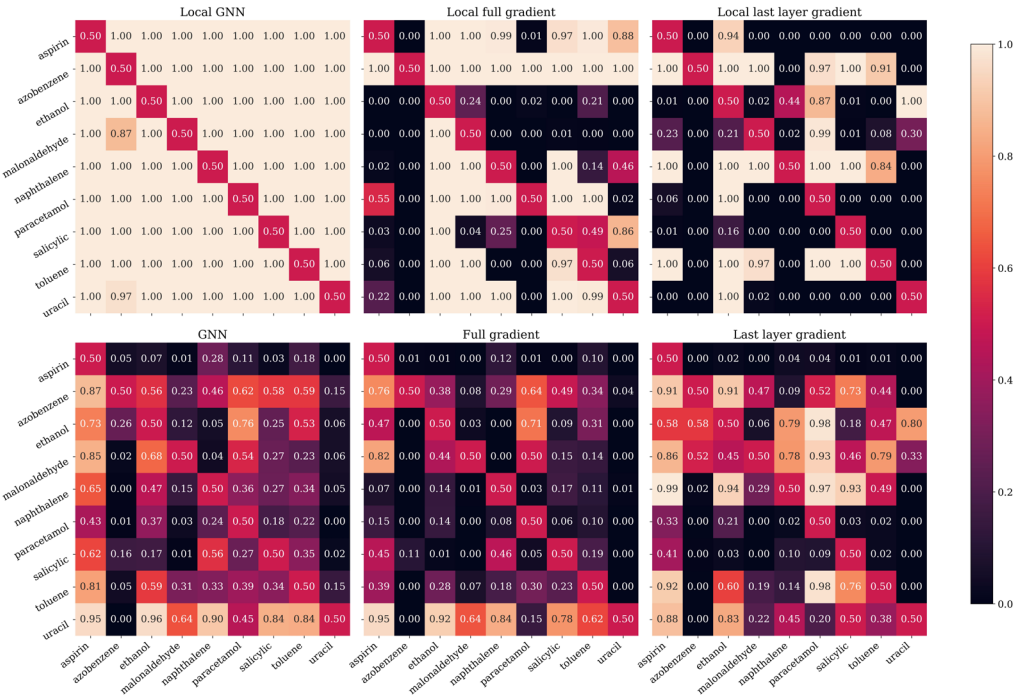

Figure A.9: Heatmap displays the AUC-ROC values derived with different kernels and Mahalanobis distance from SchNet on MD17. Each row represents a separate model, trained on the molecule listed to the left, and tested against all other molecules.

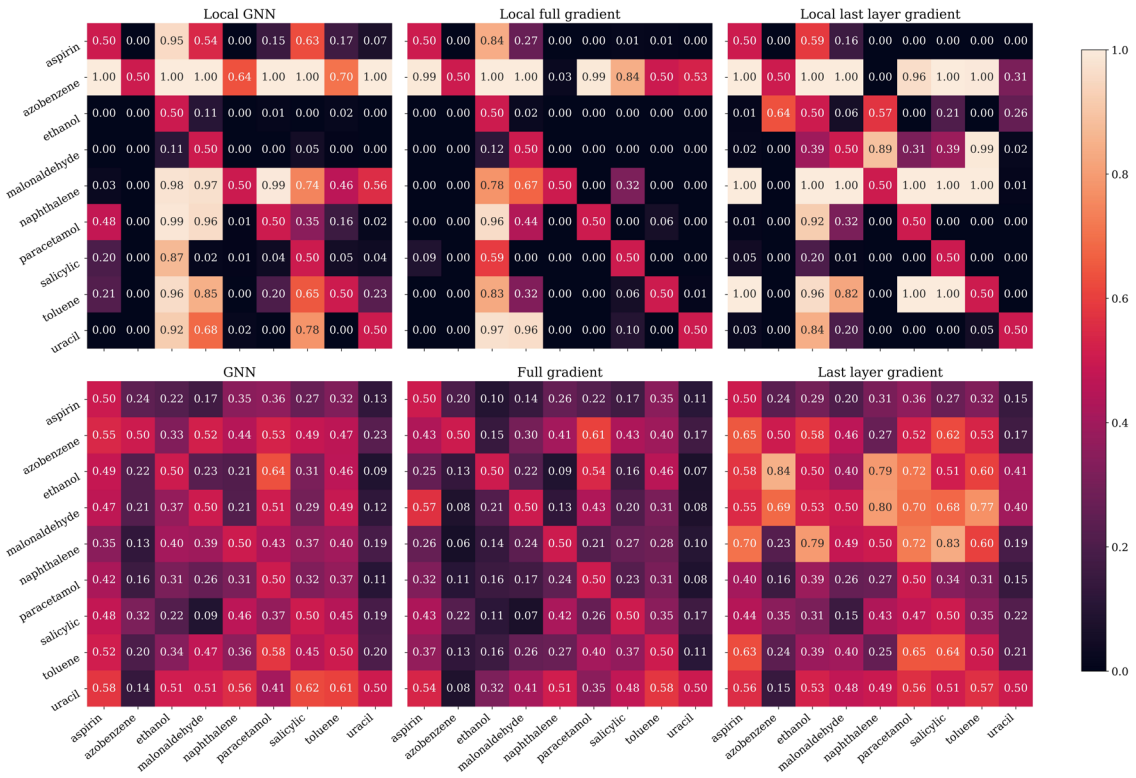

Figure A.10: Heatmap displays the AUC-ROC values derived with different kernels and Euclidean distance from SchNet on MD17. Each row represents a separate model, trained on the molecule listed to the left, and tested against all other molecules.

Appendix B

Supplementary materials for Chapter 4: Oxygen reduction at confined Au(100)-water interface

Supplementary Tables

Table B.1: Parameters for calculating coordination numbers of O_2

CV	Definition	Parameters
C_{O_2-O}	$C_{O_2-O} = \sum_{i \in O_2} \frac{1-\left(\frac{r_{i,O}-d_0}{r_0}\right)^n}{1-\left(\frac{r_{i,O}-d_0}{r_0}\right)^m}$	$r_{i,O}$: distance between O_i and O $r_0 = 1.8, d_0 = 0, n = 6, m = 12$
C_{O_2-H}	$C_{O_2-H} = \sum_{i \in O_2} \frac{1-\left(\frac{r_{i,H}-d_0}{r_0}\right)^n}{1-\left(\frac{r_{i,H}-d_0}{r_0}\right)^m}$	$r_{i,H}$: distance between O_i and H $r_0 = 1.5, d_0 = 0, n = 8, m = 16$

Table B.2: A summary of test error metrics of neural network potentials

Model	Node size	Layers	Energy error (meV/atom)		Forces error (meV/Å)			
			MAE	RMSE	l_2 MAE	l_2 RMSE	MAE	RMSE
NNP1	96	3	0.6	1.3	32.8	46.4	16.3	26.8
NNP2	112	3	0.5	1.3	31.3	45.0	15.5	26.0
NNP3	128	3	0.8	1.4	28.9	43.7	14.3	25.2
NNP4	128	4	0.4	1.2	25.4	39.3	12.6	22.7
NNP5	144	3	0.5	1.2	27.0	40.8	13.4	23.5
Ensemble	-	-	0.7	1.4	25.3	38.8	12.6	22.4

Table B.3: A summary of interface structures presented in the final dataset after CUR selection

Interface structure	Number of atoms			Number of configurations			E_{MAE} (meV/atom)	F_{MAE} (meV/Å)
	H	O	Total	Training	Validation	Total		
Au(100)-30H_2O	60	30	126	616	70	686	2.3	14.2
Au(100)-1OH/29H_2O	59	30	125	1535	172	1707	1.4	12.8
Au(100)-2OH/28H_2O	58	30	124	646	74	1694	1.1	14.0
Au(100)-1O_2/30H_2O	60	32	126	1192	136	1328	1.6	13.8
Au(100)-1OH/58H_2O	117	59	240	1458	155	1613	0.7	11.6
Au(100)-2OH/57H_2O	116	59	239	1541	153	1694	0.6	11.6
Au(100)-3OH/56H_2O	115	59	238	1667	209	1876	0.5	12.1
Au(100)-4OH/55H_2O	114	59	237	2188	245	2433	0.3	12.4
Au(100)-5OH/54H_2O	113	59	236	2110	232	2342	0.3	12.9
Au(100)-6OH/53H_2O	112	59	235	1986	229	2215	0.3	13.8
Au(100)-1O_2/57H_2O	114	59	237	1826	188	2014	0.4	12.0

Table B.4: Comparison of model performance between previous studies and this work

Ref.	System	Method	Max. N_{atom}	Training set	Errors
Natarajan et al.[43]	Cu-H_2O	BPNNP	463	10293 structures with bulk water/ice, bulk copper/cuprous oxide and water–copper interface geometries. 10% are used for validation.	E_{RMSE}: 0.9 meV/atom F_{RMSE}: 125.3 meV/Å
Quaranta et al.[40]	ZnO-H_2O	BPNNP	334	15319 structures with bulk water, bulk ZnO, and ZnO-water interface geometries. 1712 configurations are used for validation.	E_{RMSE}: 1.2 meV/atom F_{RMSE}: 143.4 meV/Å
Yang et al.[164]	Urea-water	DeepMD	110	14536 structures with 5739 reactant structures, 5217 product structures, and 3580 transition state structures.	E_{MAE}: 0.7 meV/atom F_{MAE}: 38 meV/Å
He et al.[197]	$SrTiO_3$	DeepMD	40	2600 structures with $2 \times 2 \times 2$ and $1 \times 1 \times 1$ supercells. A test set with 1500 structures of 80 atoms are used for validation.	E_{MAE}: 0.3 meV/atom F_{MAE}: 19 meV/Å
Liu et al.[198]	βGa_2O_3	GAP	160	801 training structures obtained from MD simulations at temperatures between 100 K and 1000 K. 90 structures are used for validation.	E_{RMSE}: 0.5 meV/atom F_{RMSE}: 50 meV/Å for Ga F_{RMSE}: 38 meV/Å for O
Davidson et al.[199]	$\alpha Fe - H$	GAP	128	The training data for the H-Fe interaction potential comprises snapshots from molecular dynamics trajectories of 54 and 128 Fe atoms with either 0, 1, or 2 Fe atoms removed, and a single H atom added. Altogether, about 400 configurations were used in the fit, comprising about 28k atoms.	E_{MAE}: 20 meV F_{MAE}: 10 meV/Å
Hu et al.[178]	OC20 dataset[200]	ForceNet	225	OC20 dataset [200] contains 200M+ nonequilibrium 3D atomic structures with average atom number of 73.3 from 1M+ atomic relaxation trajectories. The model is trained on 134M structures from S2F task. Four validation datasets are used to test the model performance: In Domain (ID),Out of Domain Adsorbate (OOD Adsorbate), OOD Catalyst, and OOD Both (both the adsorbate and catalyst's material are not seen in training). Each split contains 1M examples.	F_{MAEs} ID: 28.1 meV/Å OOD Ads.: 32.0 meV/Å OOD Cat.: 32.7 meV/Å OOD Both: 41.2 meV/Å Average: 33.5 meV/Å
Gasteier et al.[201]	OC20 dataset	GemNet-OC	225	The model is trained on 134M structures from S2EF task in OC20 dataset. The same test set splits are used as above.	Average: E_{MAE}: 233 meV F_{MAE}: 20.7 meV/Å
Li et al.[202]	Water	GAMD	384	7000 periodic configurations of liquid water. The number of water molecules in the unit cell ranges from 16 to 128. 723 snapshots are used for validation.	F_{MAE}: 24.28 ± 16.80 meV/Å F_{RMSE}: 35.39 ± 23.09 meV/Å
Batzner et al.[66]	$Li_4P_2O_7$	Nequip	208	The crystal structure was melted at 3000 K for 50 ps, and quenched at 600 K for another 50 ps, resulting a dataset of 25,000 AIMD frames. 1000 structures from melting phase are used for training, 100 structures for validation, and all remaining structures for independent test.	Melt(quench): E_{MAE}: 0.4(0.5) meV/atom F_{MAE}: 34.0(21.3) meV/Å E_{RMSE}: 0.8(0.5) meV/atom F_{RMSE}: 59.5(34.9) meV/Å
Ours	Au-water	PaiNN	240	18371 Au(100)-water interface structures with different number of *OH and O_2 presented in the liquid. The rare event structures in metadynamics are also included.	E_{MAE}: 0.7 meV/atom F_{MAE}: 12.6 meV/Å E_{RMSE}: 1.4 meV/atom F_{RMSE}: 22.4 meV/Å

Table B.5: Adsorption energies of different species on Au(100) surface

Species	Cell size	θ (Coverage)	$\Delta E/n_{OH}$ (eV)
1OH	(4×4)	0.063	0.717
2OH	(4×4)	0.125	0.795
3OH	(4×4)	0.188	0.862
4OH	(4×4)	0.250	0.925
5OH	(4×4)	0.313	0.958
6OH	(4×4)	0.375	0.987
1O2	(4×4)	0.063	-1.008
1OH	(3×3)	0.111	0.723
2OH	(3×3)	0.222	0.858

Table B.6: Coordination numbers of different possible intermediate structures in ORR

CV	$*O_2$	*OOH	*O + *OH	$*O_2H_2$	*OH
C_{O_2-O}	1	1	0	1	0
C_{O_2-H}	0	0.5	0.5	1	1

Supplementary Figures

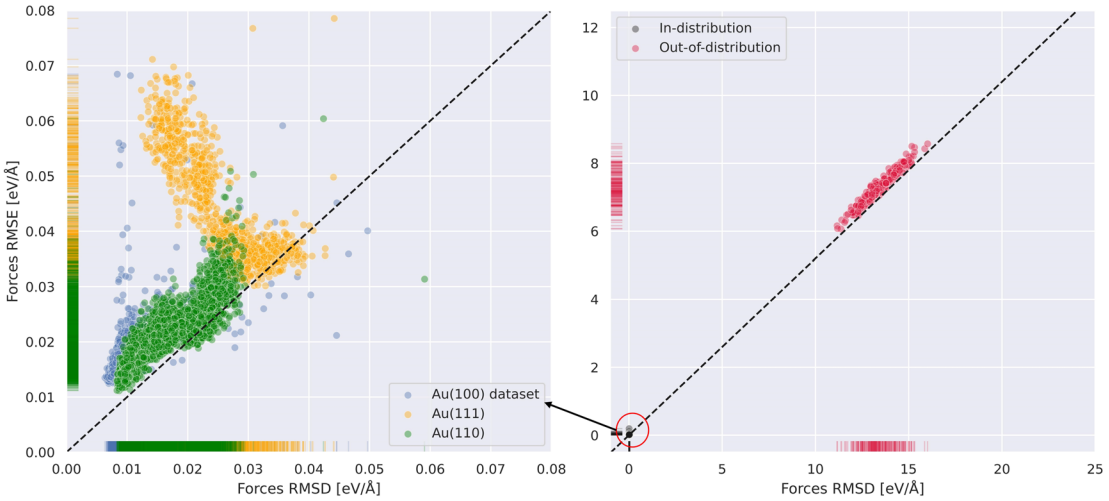

Figure B.1: Comparison between ensemble uncertainties calculated by forces root mean square deviation (RMSD) and true prediction error calculated by root mean square error (RMSE) for (a) in-distribution data, and (b) out-of-distribution data.

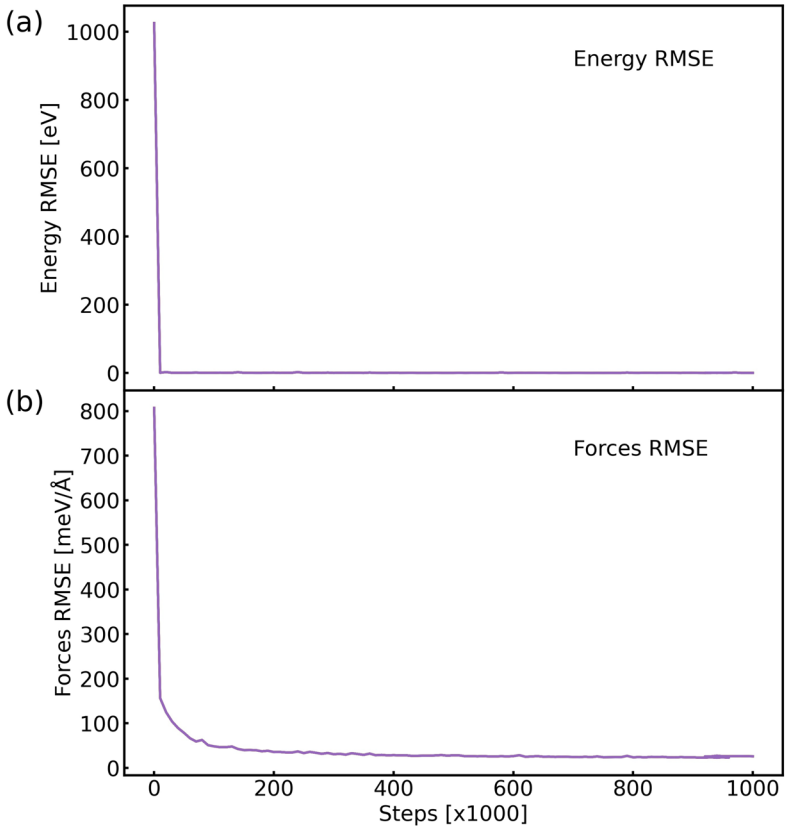

Figure B.2: Training curves of the final dataset for (a) energy root mean squared error (RMSE) and (b) forces RMSE

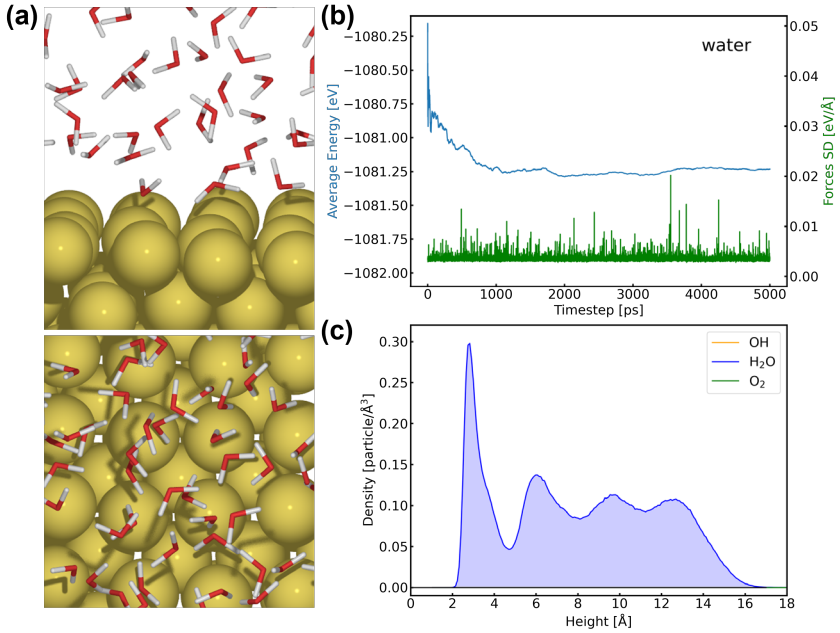

Figure B.3: (a) Side view and top view of Au(100)-59H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-59H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

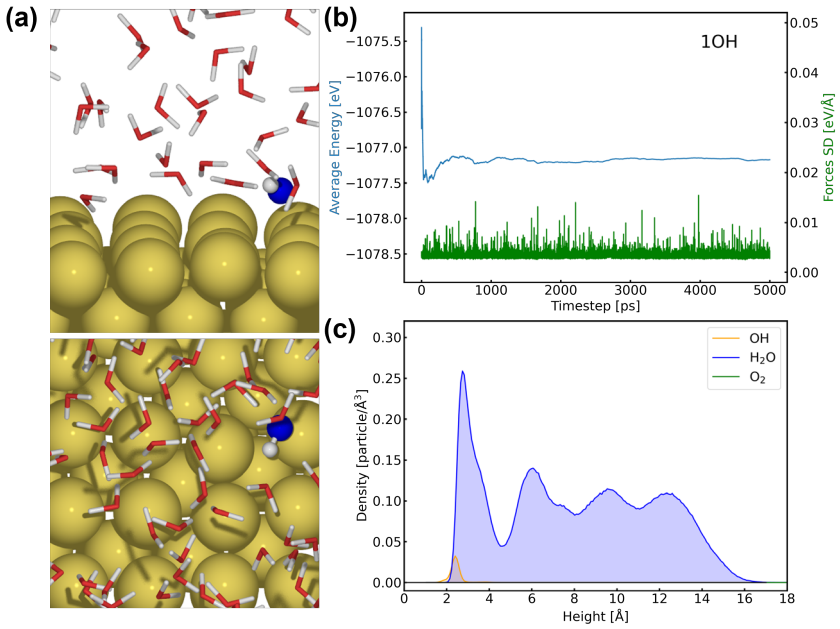

Figure B.4: (a) Side view and top view of Au(100)-1OH/58H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-1OH/58H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

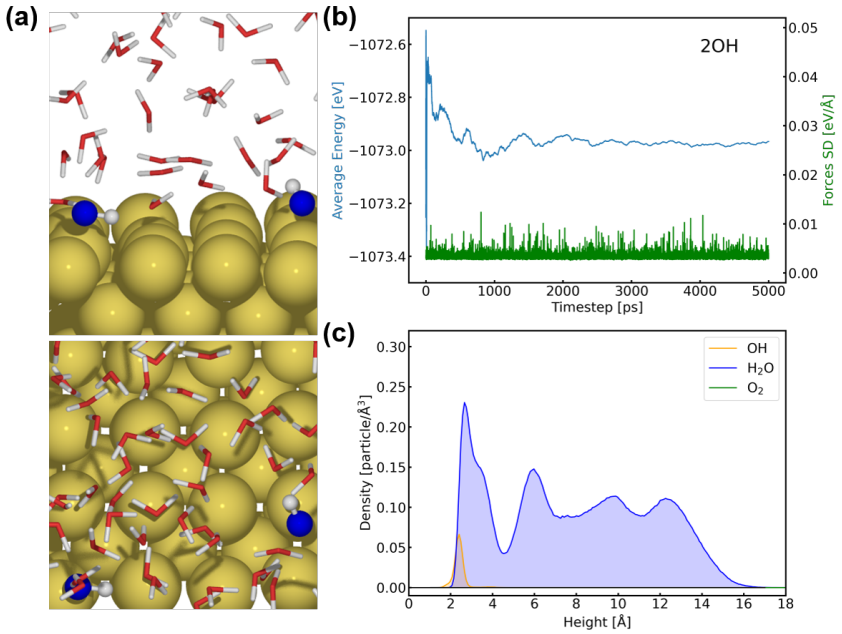

Figure B.5: (a) Side view and top view of Au(100)-2OH/57H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-2OH/57H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

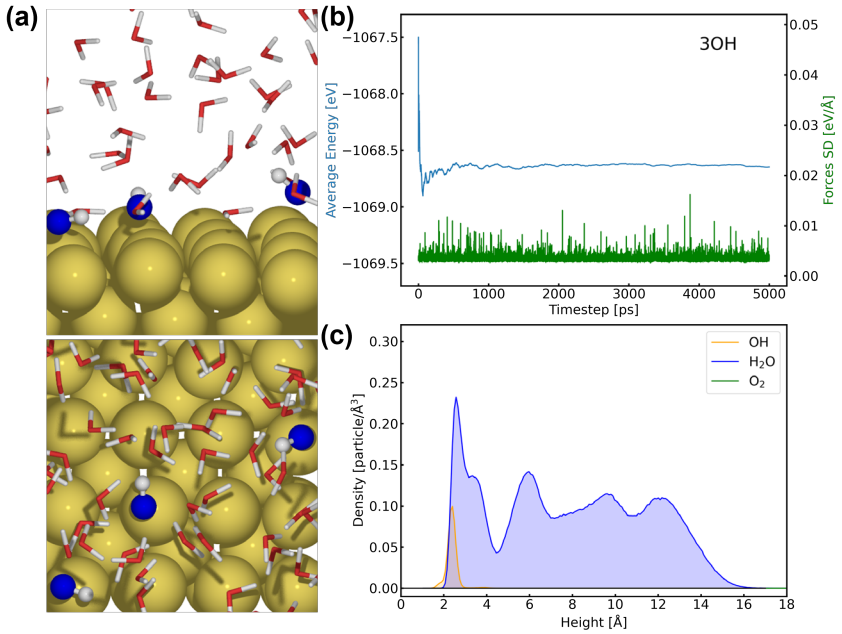

Figure B.6: (a) Side view and top view of Au(100)-3OH/56H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-3OH/56H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

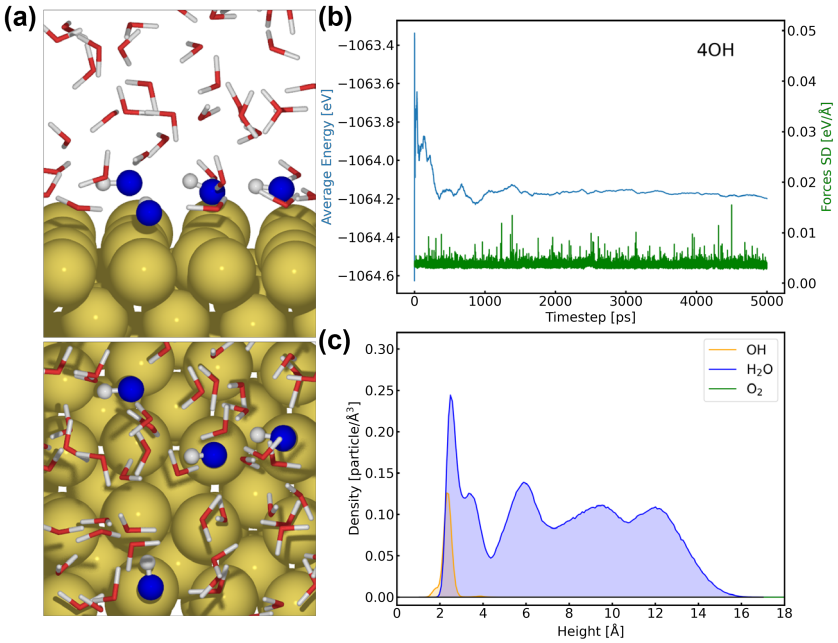

Figure B.7: (a) Side view and top view of Au(100)-4OH/55H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-4OH/55H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

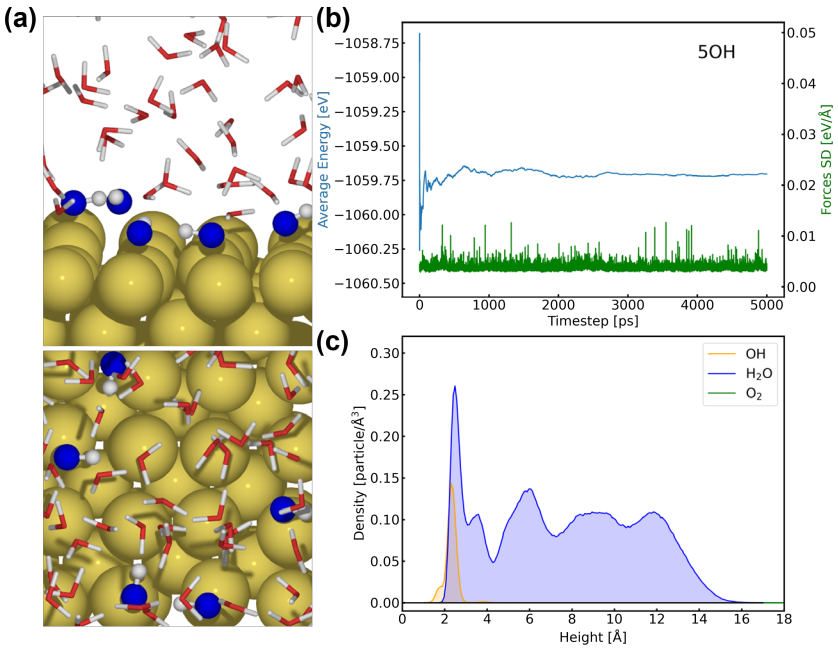

Figure B.8: (a) Side view and top view of Au(100)-5OH/54H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-5OH/54H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

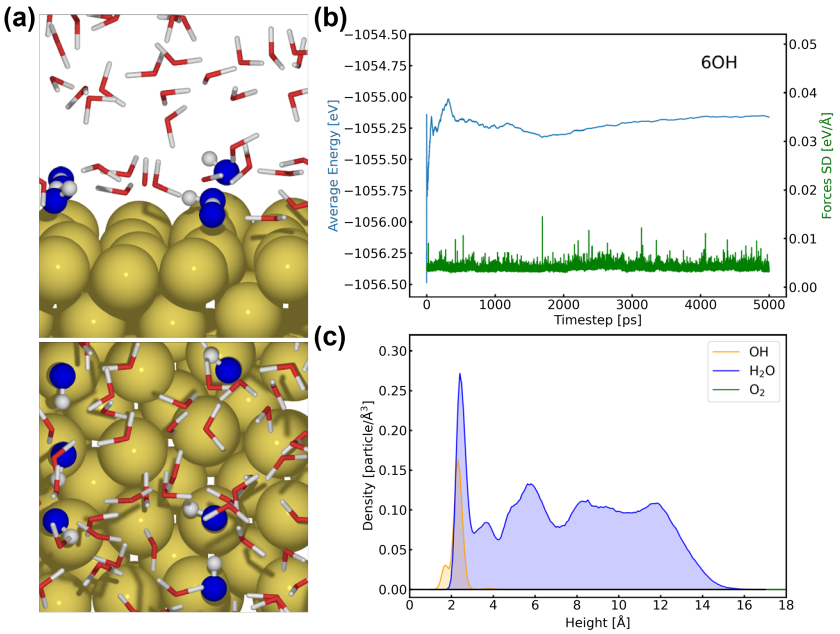

Figure B.9: (a) Side view and top view of Au(100)-6OH/53H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-6OH/53H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

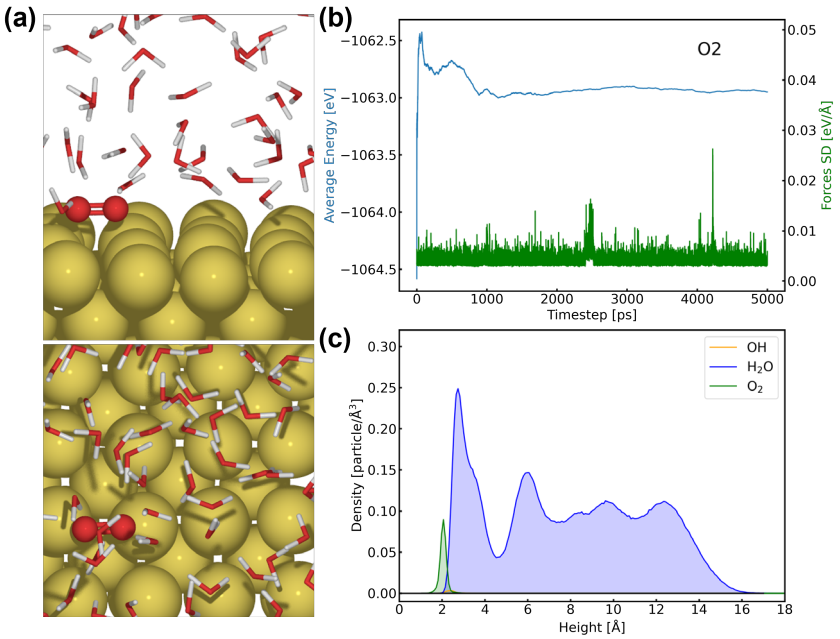

Figure B.10: (a) Side view and top view of Au(100)-1O_2/30H_2O interface structure. (b) Evolution of average energy and force standard deviations (SD) of Au(100)-1O_2/30H_2O along 5 ns MD simulations. (c) Density profiles of different species as a function of the distance from Au(100) surface.

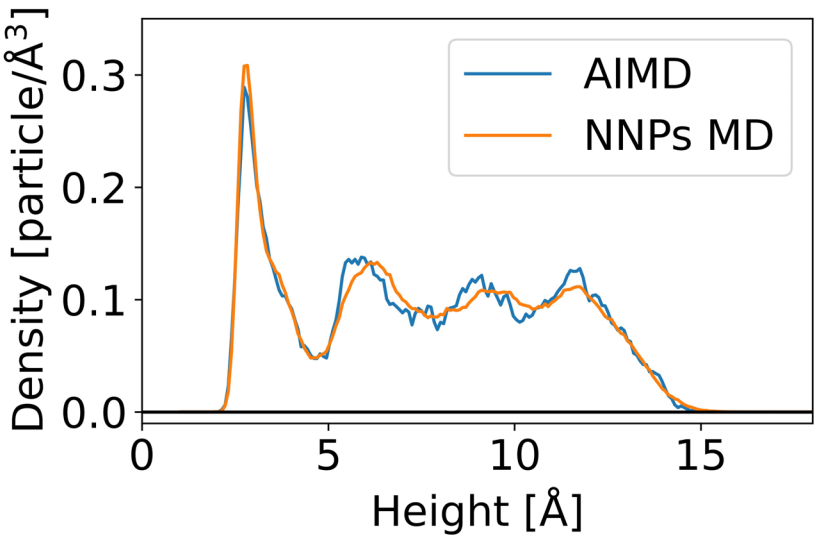

Figure B.11: Density profiles of water as a function of the distance from Au(100) surface obtained from 50 ps AIMD (blue) and 5 ns NNPs MD (orange).

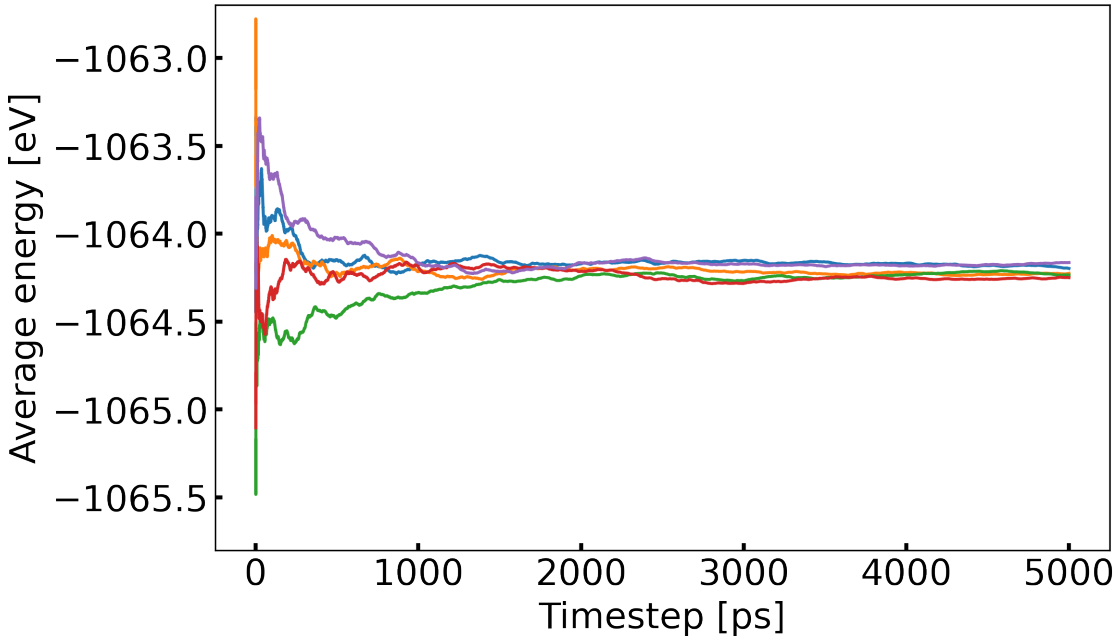

Figure B.12: Average energy profiles of Au(100)-4OH/55H_2O from five different starting configurations.

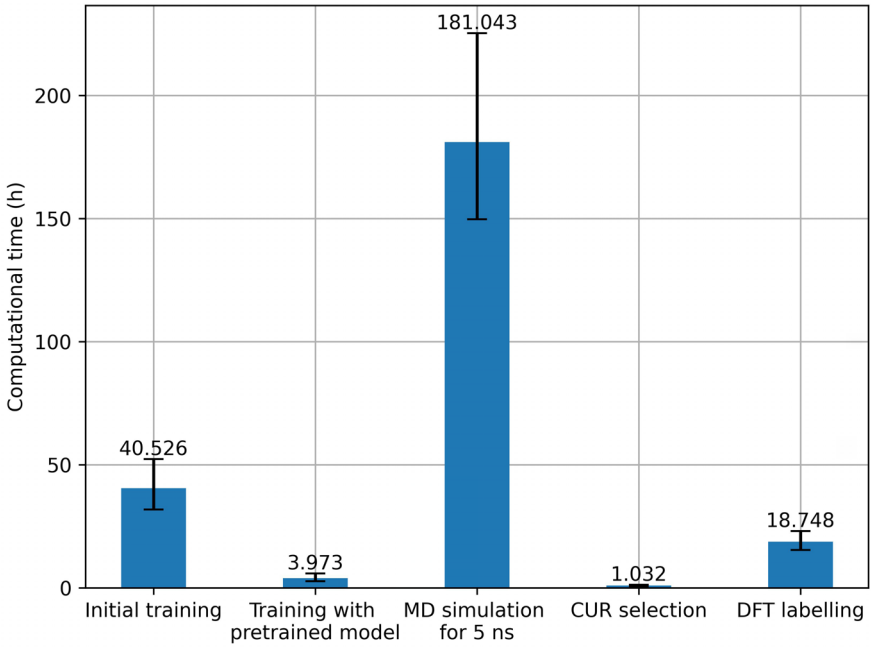

Figure B.13: Computational time for training an initial model, retraining model, 5 ns NNPs MD simulation, and CUR selection. The training time of the initial model is evaluated on 1000,000 steps for five different models. By loading pretrained model parameters, retraining takes approximately 100,000 steps on average to early-stopping. DFT labelling cost is estimated by the time of labelling 100 structures using 40 CPU cores.

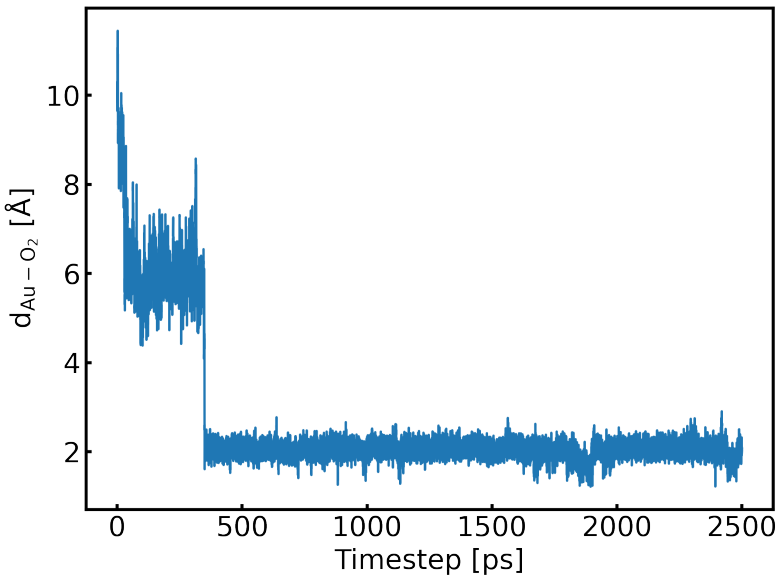

Figure B.14: Evolution of the distance between O_2 and Au(100) surface along 2.5 ns MD simulation

Appendix C

Supplementary materials for Chapter 5: Facet-dependent ORR on Au surfaces

Supplementary Tables

Table C.1: A summary of interface structures presented in the final dataset

Interface structure	atoms			Configurations
	H	O	Total	
Au(100)-67H2O	134	67	265	509
Au(100)-102H2O	204	102	431	525
Au(100)-2OH/100_2O	202	102	429	900
Au(100)-4OH/98H_2O	200	102	427	832
Au(100)-6OH/96H_2O	198	102	425	871
Au(100)-8OH/94H_2O	196	102	423	867
Au(100)-1O_2/100H_2O	200	102	427	578
Au(100)-2O_2/98H_2O	196	102	423	542
Au(100)-1O_2(MetaD)	200	102	427	480
Au(100)-2O_2(MetaD)	196	102	423	200
Au(110)-73H2O	146	73	267	835
Au(110)-115H2O	230	115	445	859
Au(110)-2OH/113_2O	228	115	443	483
Au(110)-4OH/111_2O	226	115	441	664
Au(110)-6OH/109_2O	224	115	439	565
Au(110)-8OH/107_2O	222	115	437	483
Au(110)-1O_2/113H_2O	226	115	441	560
Au(110)-2O_2/111H_2O	222	115	437	453
Au(110)-1O_2(MetaD)	226	115	441	400
Au(110)-2O_2(MetaD)	222	115	437	500
Au(111)-59H2O	146	73	267	835
Au(111)-108H2O	216	108	474	2128
Au(111)-2OH/106_2O	214	108	472	760
Au(111)-4OH/104_2O	212	108	470	886
Au(111)-6OH/102_2O	210	108	468	923
Au(111)-8OH/100_2O	208	108	466	842
Au(111)-1O_2/106H_2O	212	108	470	1180
Au(111)-2O_2/104H_2O	208	108	466	740
Au(111)-1O_2(MetaD)	212	108	470	600
Au(111)-2O_2(MetaD)	208	108	466	600

Table C.2: A summary of test error metrics of neural network potentials

Model	Node size	Layers	Energy error (meV)		Forces error (meV/Å)	
			MAE	RMSE	MAE	RMSE
NNP1	112	3	0.64	1.43	25.0	49.2
NNP2	120	3	0.605	1.34	25.4	55.1
NNP3	128	3	0.675	1.17	24.8	57.6
NNP4	136	3	0.525	1.16	23.5	49.4
NNP5	144	3	1.02	1.28	23.7	54.2
NNP6	160	3	0.55	1.14	22.2	41.6

Supplementary Figures

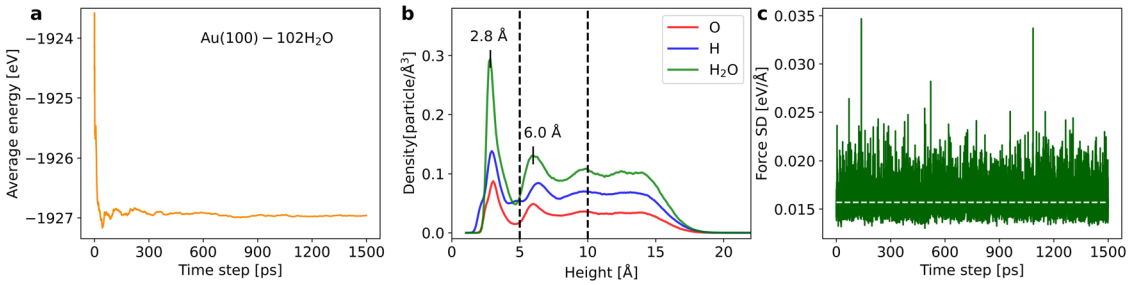

Figure C.1: (a) Evolution of average energy of Au(100)-102H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force standard deviations (SD) of Au(100)-102H_2O along 1500 ps MD simulation

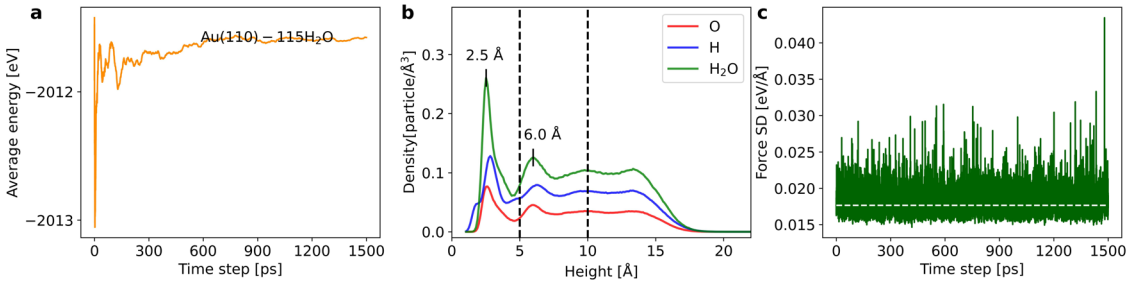

Figure C.2: (a) Evolution of average energy of Au(110)-115H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation.

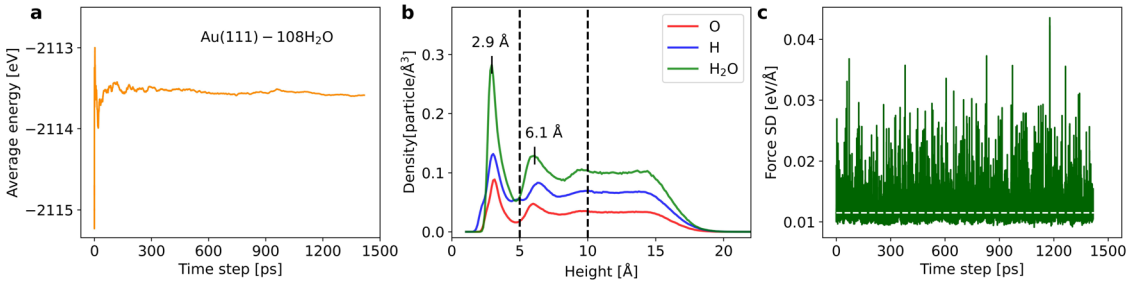

Figure C.3: (a) Evolution of average energy of Au(111)-108H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

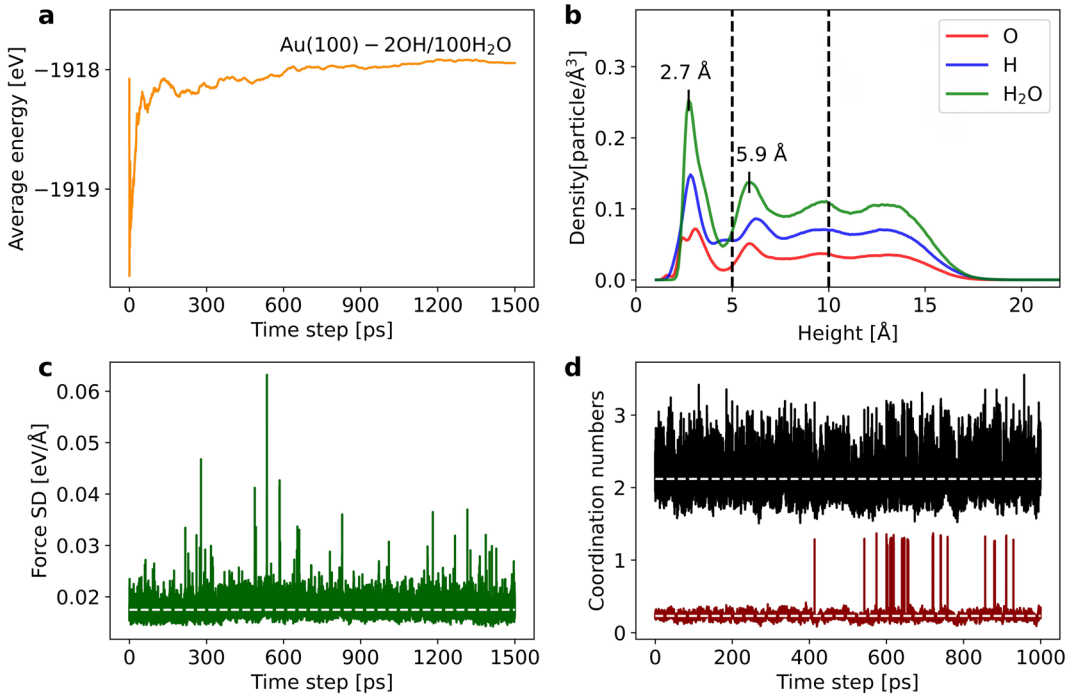

Figure C.4: (a) Evolution of average energy of Au(100)-2OH/100H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

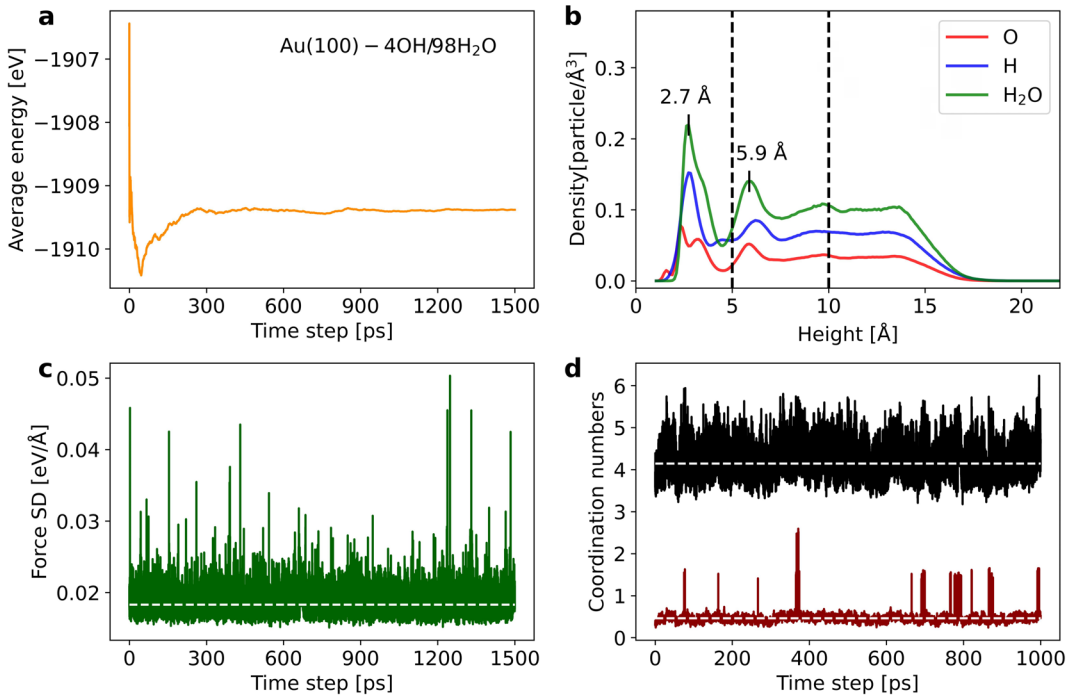

Figure C.5: (a) Evolution of average energy of Au(100)-4OH/98H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

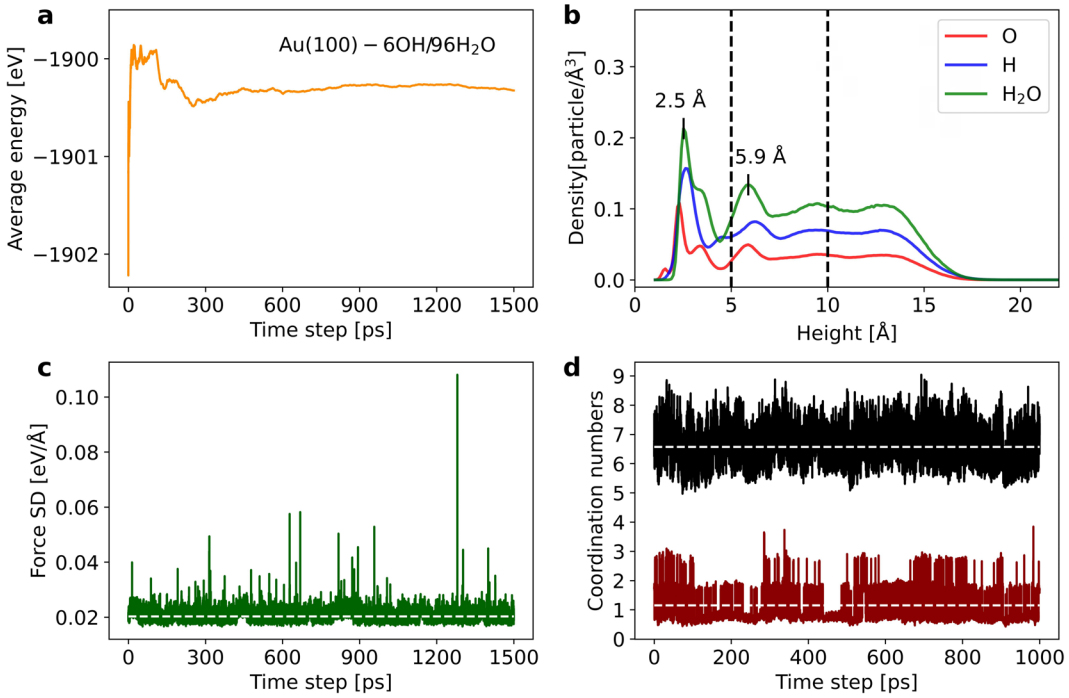

Figure C.6: (a) Evolution of average energy of Au(100)-6OH/96H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

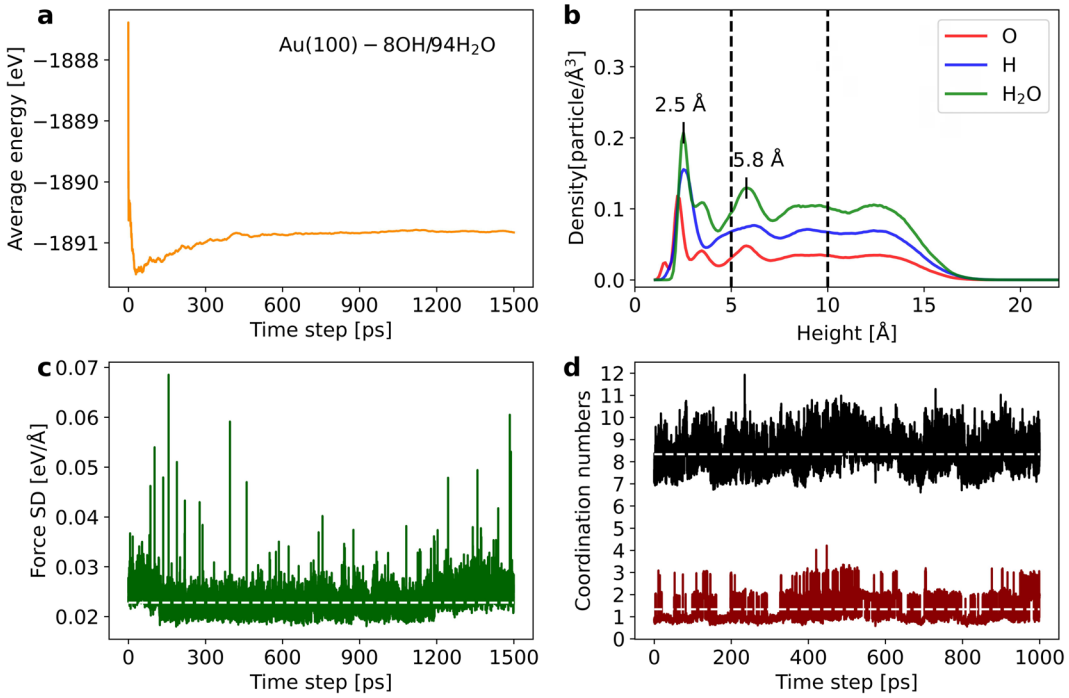

Figure C.7: (a) Evolution of average energy of Au(100)-8OH/94H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

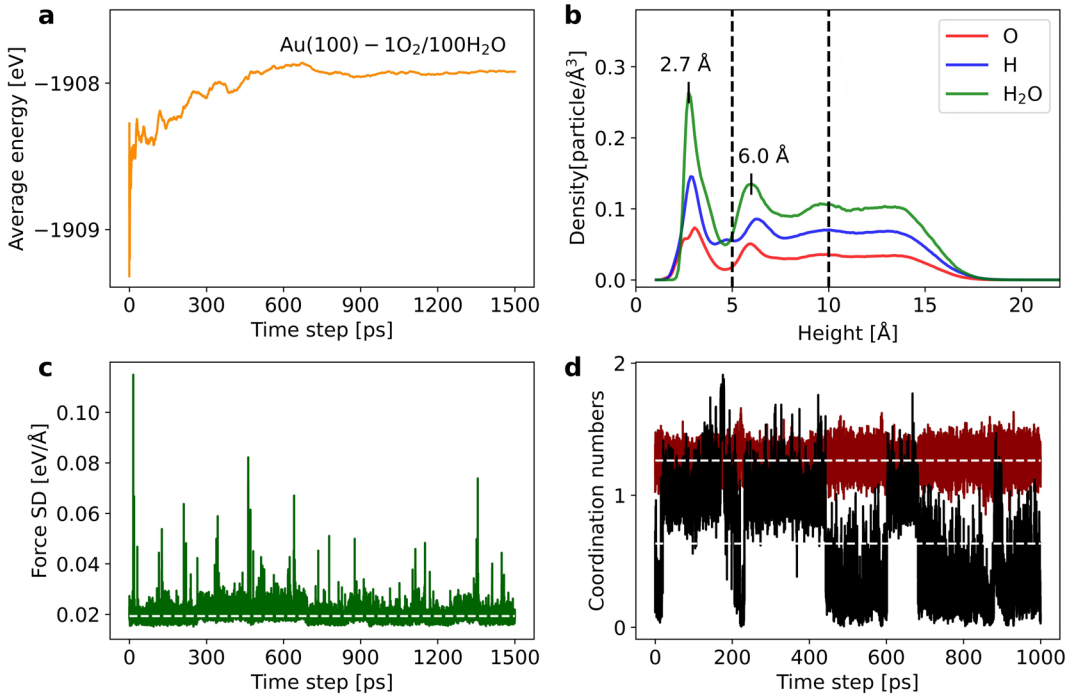

Figure C.8: (a) Evolution of average energy of Au(100)-1O_2/100H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

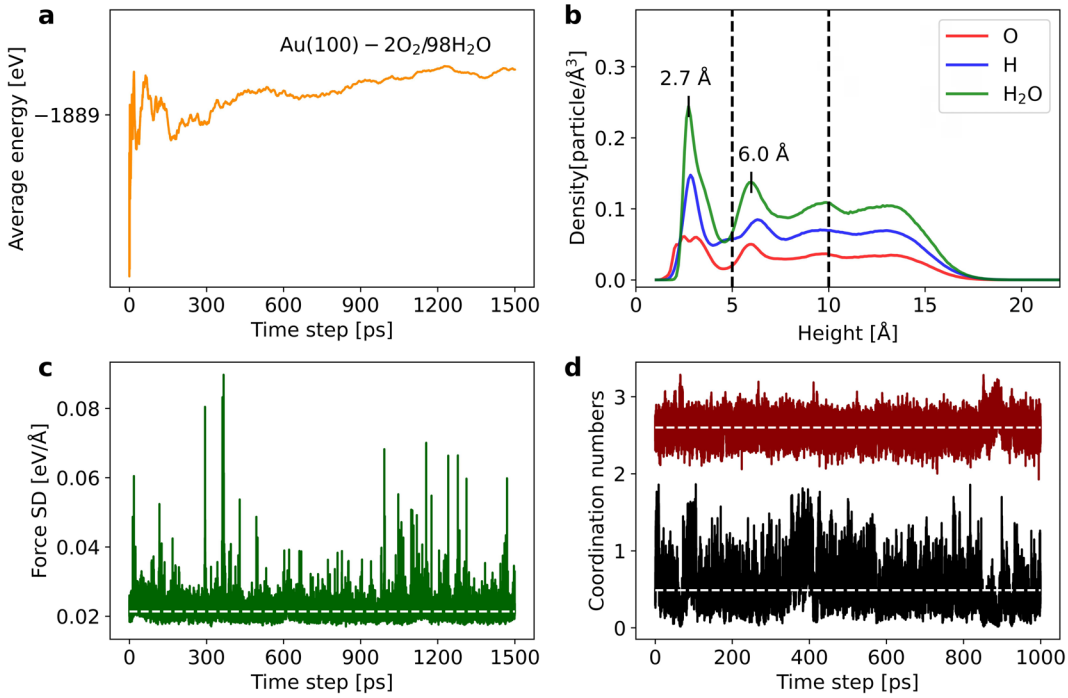

Figure C.9: (a) Evolution of average energy of Au(100)-2O_2/98H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(100) surface. (c) Evolution of force SD along 1500 ps MD simulation

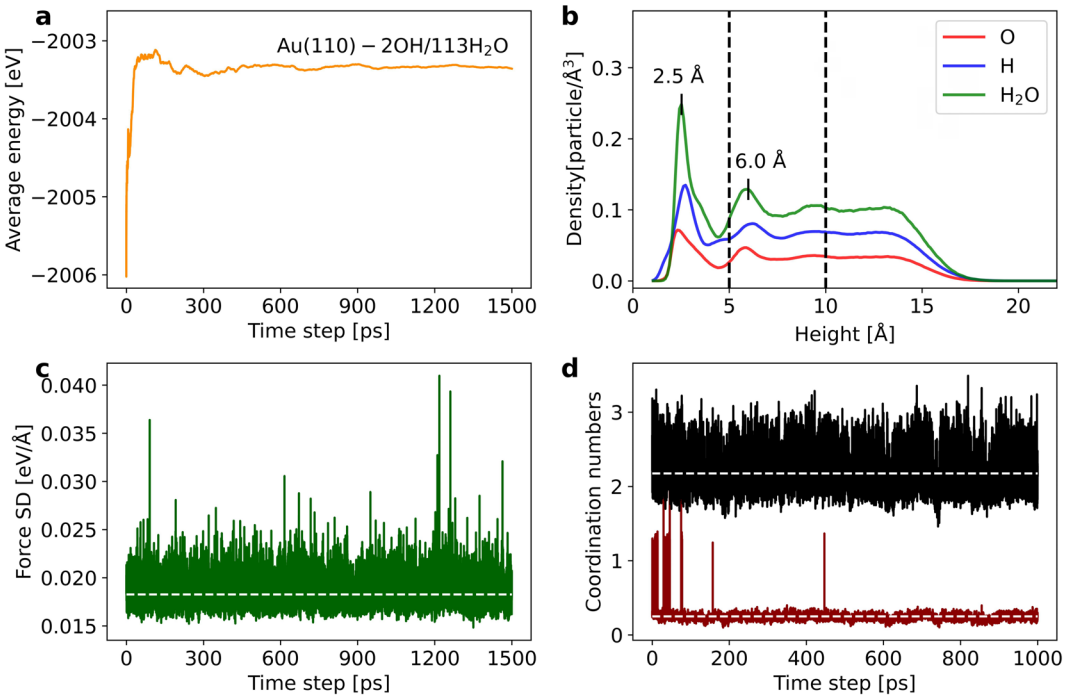

Figure C.10: (a) Evolution of average energy of Au(110)-2OH/113H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

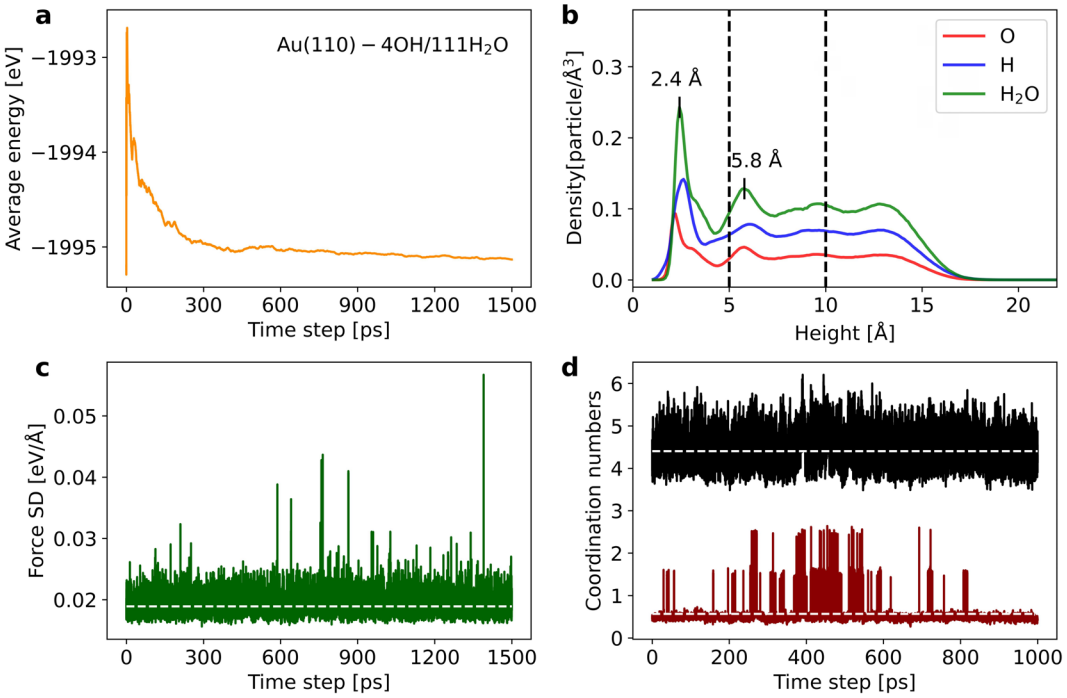

Figure C.11: (a) Evolution of average energy of Au(110)-4OH/111H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

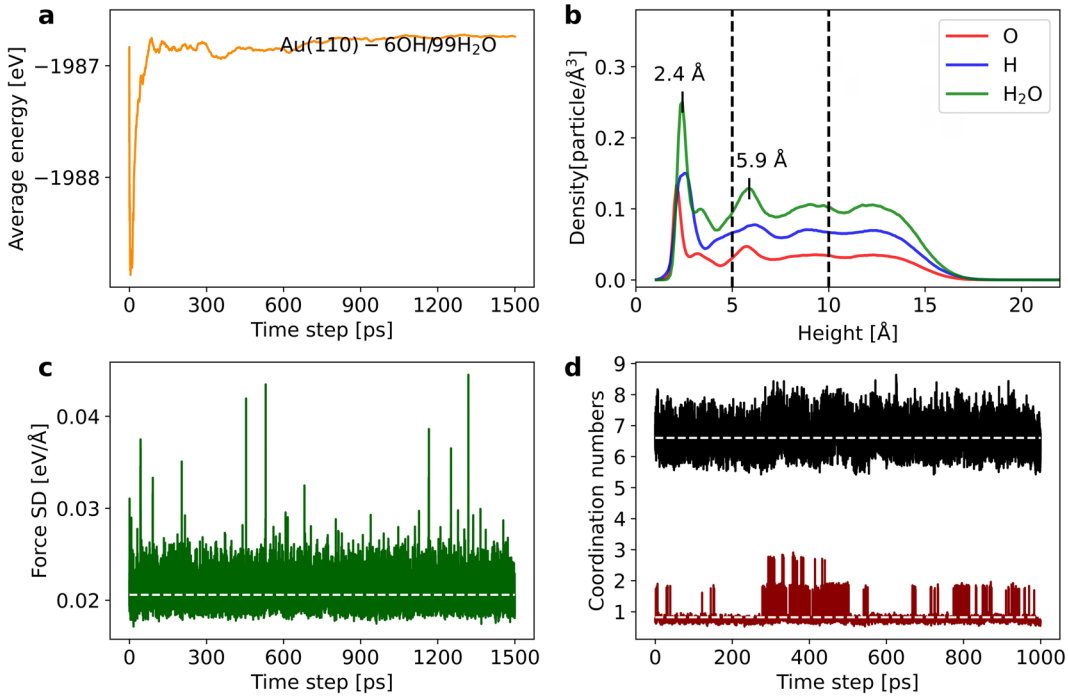

Figure C.12: (a) Evolution of average energy of Au(110)-6OH/109H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

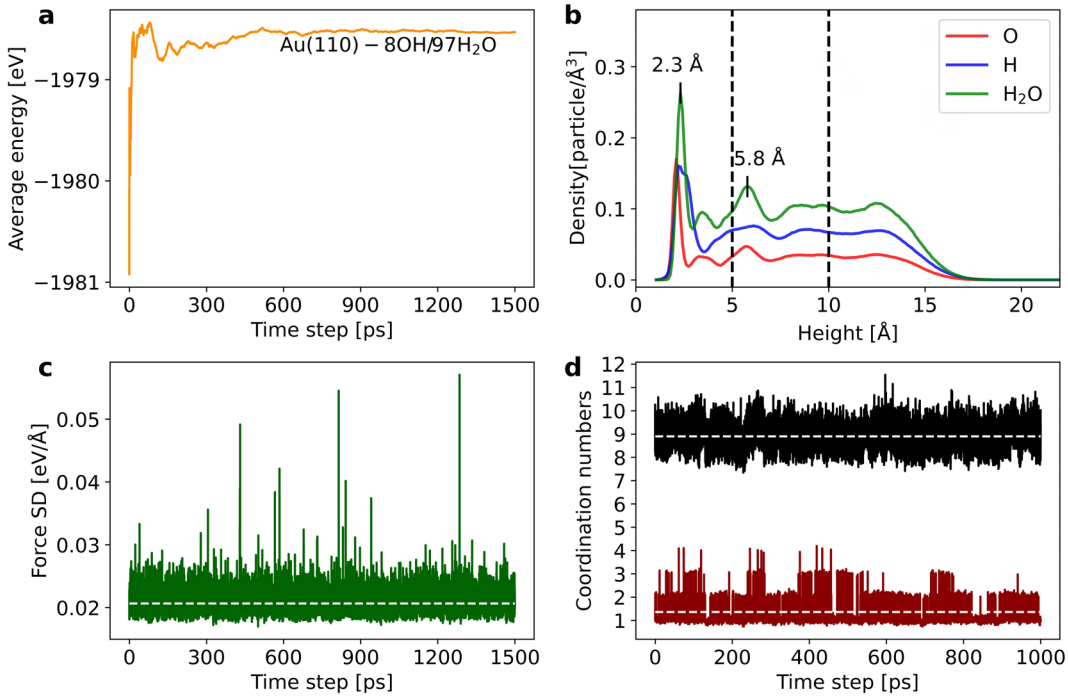

Figure C.13: (a) Evolution of average energy of Au(110)-8OH/107H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

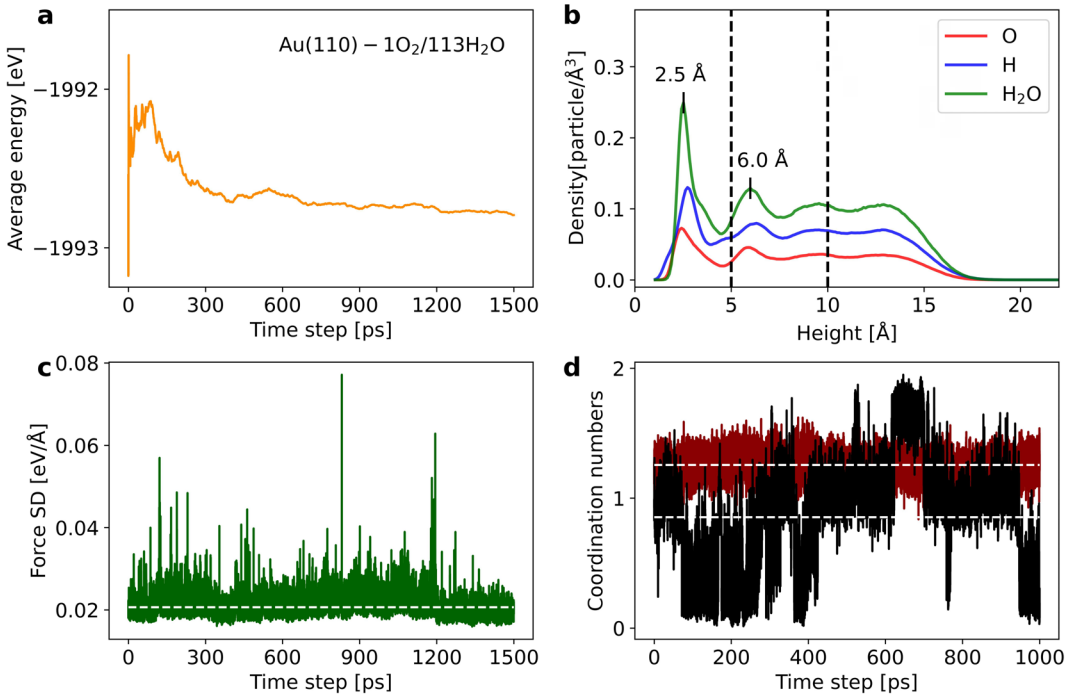

Figure C.14: (a) Evolution of average energy of Au(110)-1O_2/113H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

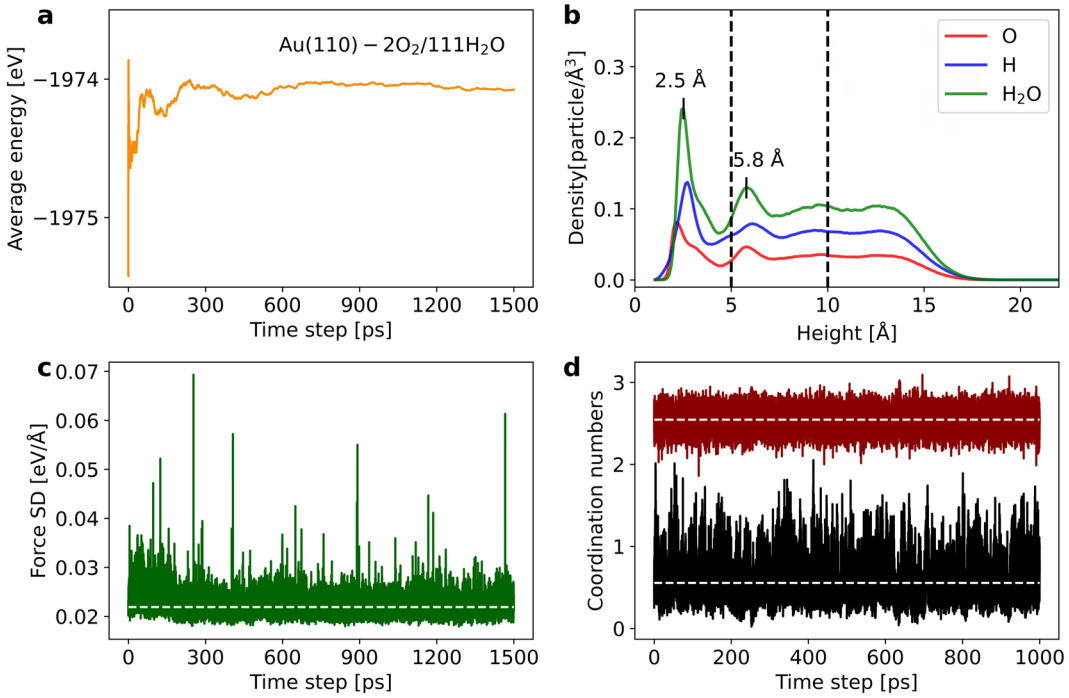

Figure C.15: (a) Evolution of average energy of Au(110)-2O_2/111H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(110) surface. (c) Evolution of force SD along 1500 ps MD simulation

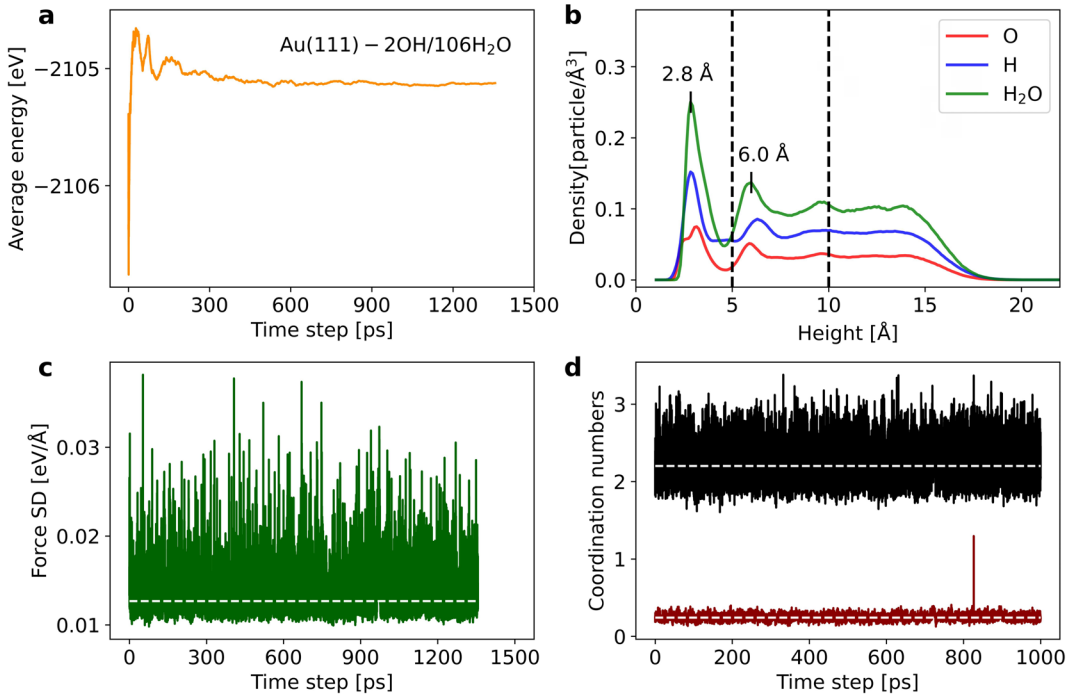

Figure C.16: (a) Evolution of average energy of Au(111)-2OH/106H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

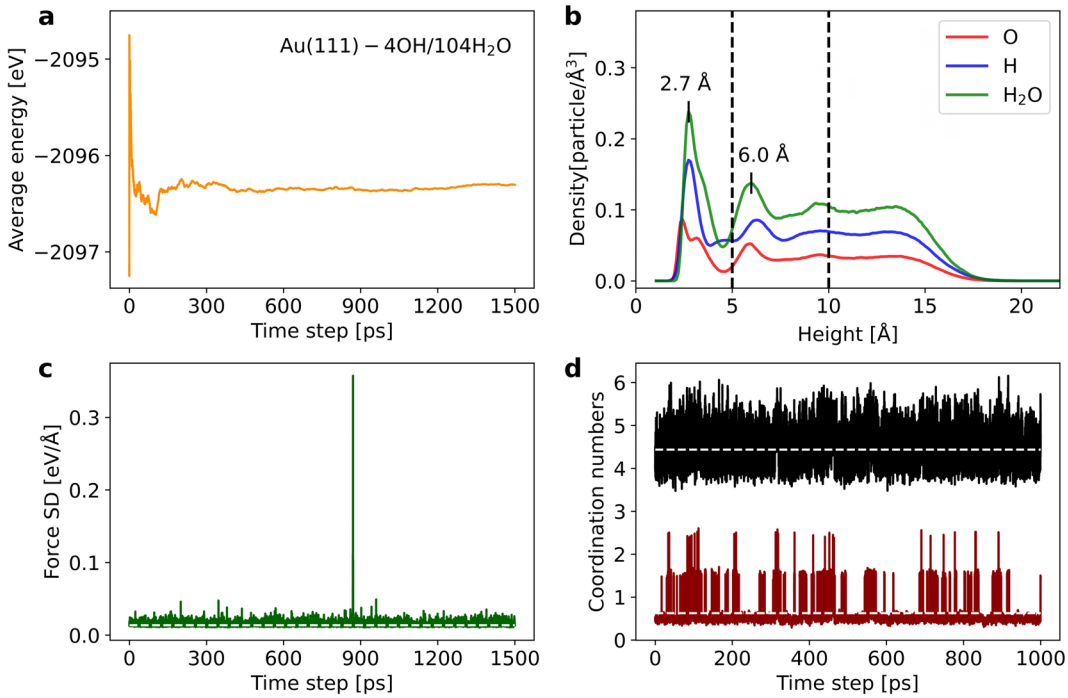

Figure C.17: (a) Evolution of average energy of Au(111)-4OH/104H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

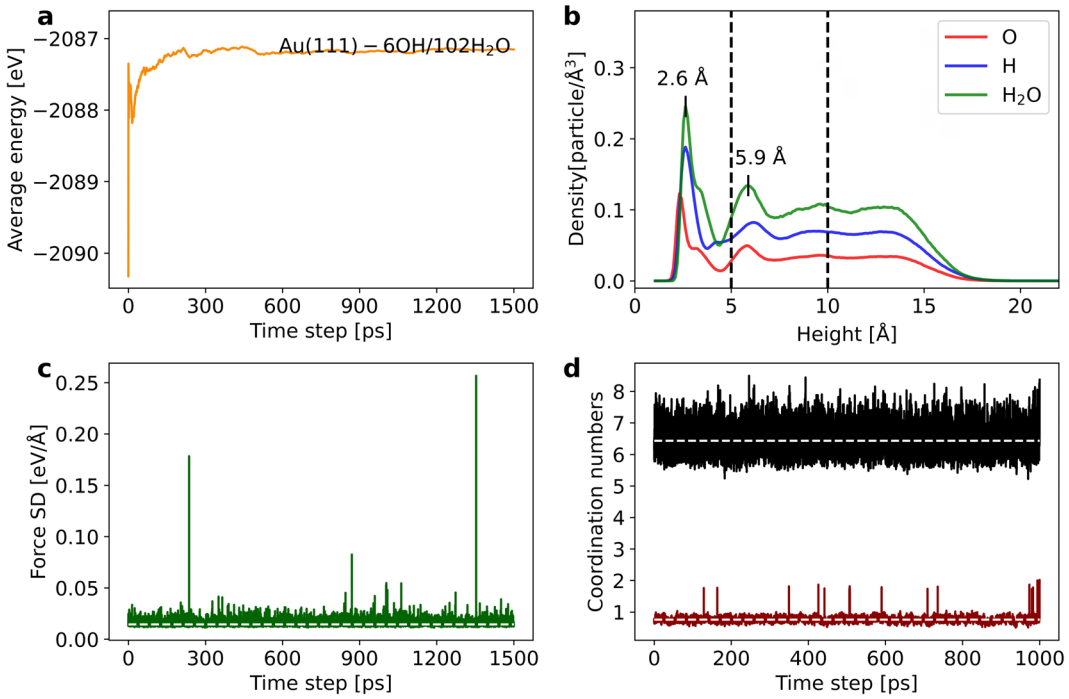

Figure C.18: (a) Evolution of average energy of Au(111)-6OH/102H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

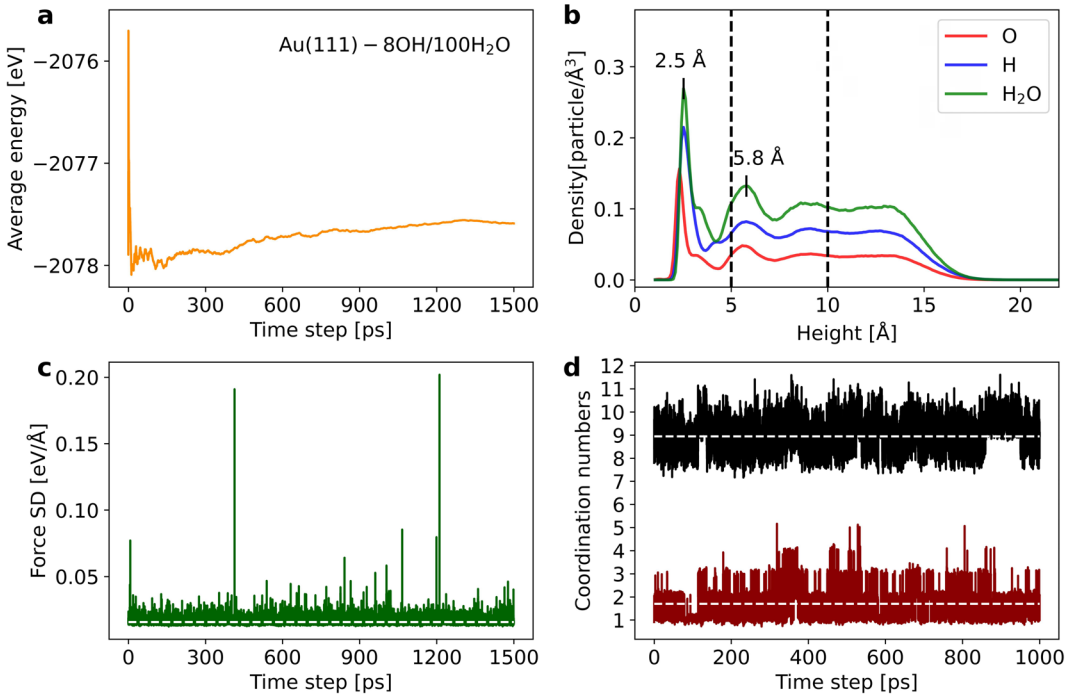

Figure C.19: (a) Evolution of average energy of Au(111)-8OH/100H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

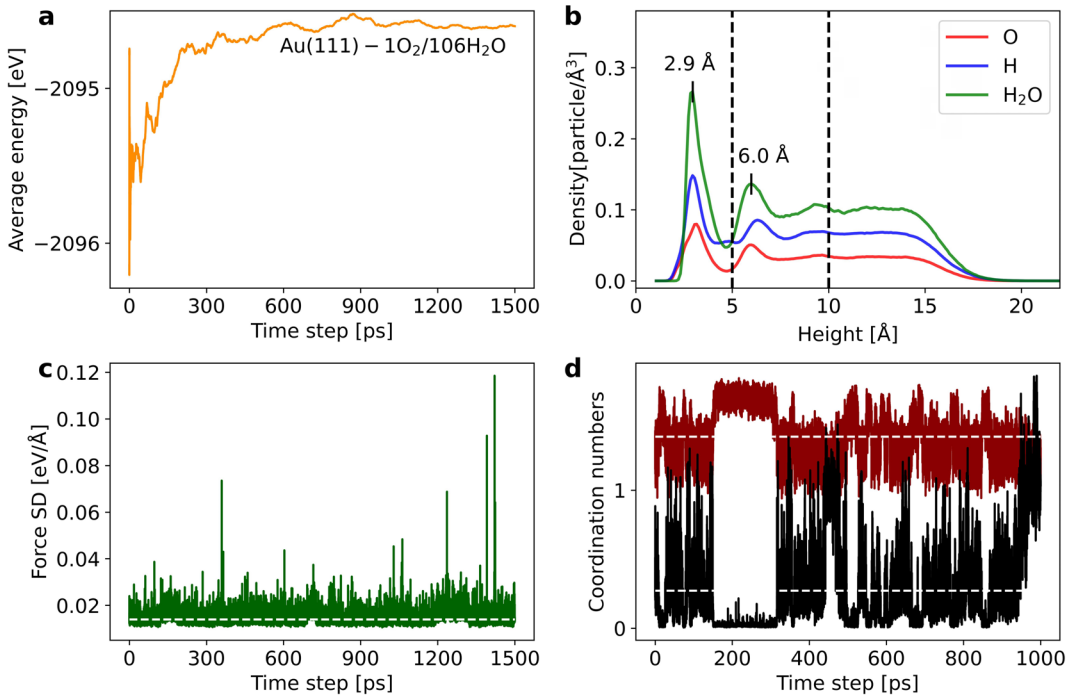

Figure C.20: (a) Evolution of average energy of Au(111)-1O2/106H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

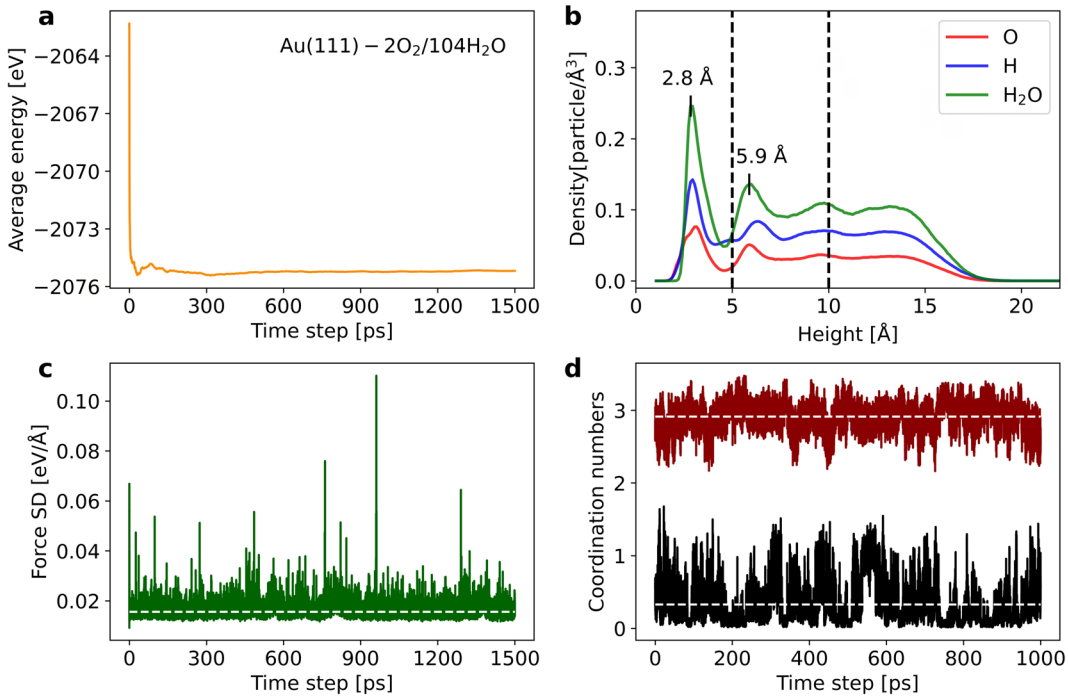

Figure C.21: (a) Evolution of average energy of Au(111)-2O2/104H_2O along 1500 ps MD simulation. (b) Density profiles of different species as a function of the distance from Au(111) surface. (c) Evolution of force SD along 1500 ps MD simulation

Paper I

Batch Active Learning at the Core: Building Robust Machine Learning Potentials for Atomistic Simulations

Xin Yang, Changzhi Ai, Sam Walton Norwood, Martin Hoffmann Petersen,Renata Sechi, Yogeshwaran Krishnan, Smobin Vincent, Jonas Busk, François Raymond J Cornet, Ole Winther, Juan Maria García Lastra, Tejs Vegge, Heine Anton Hansen, and Arghya Bhowmik

To be submitted

Batch Active Learning at the Core: Building Robust Machine Learning Potentials for Atomistic Simulations

Xin Yang,[†] Changzhi Ai,[†] Sam Walton Norwood,[†] Martin Hoffmann Petersen,[†] Renata Sechi,[†] Yogeshwaran Krishnan,[†] Smobin Vincent,[†] Jonas Busk,[†] François Raymond J Cornet,[‡] Ole Winther,[‡,¶,§] Juan Maria García Lastra,[†] Tejs Vegge,[†] Heine Anton Hansen,[*,†] and Arghya Bhowmik[*,†]

†*Department of Energy Conversion and Storage, Technical University of Denmark, Anker Engelunds Vej, 2800 Kgs. Lyngby, Denmark*

‡*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Anker Engelunds Vej, 2800 Kgs. Lyngby, Denmark.*

¶*Center for Genomic Medicine, Rigshospitalet, Copenhagen University Hospital, Denmark.*

§*Bioinformatics Centre, Department of Biology, University of Copenhagen, Denmark.*

E-mail: heih@dtu.dk; arbh@dtu.dk

Abstract

Machine-learned interatomic potentials (MLIPs) have emerged as powerful tools in the domain of atomistic simulations due to their exceptional computational efficiency and ab-initio level accuracy. At the heart of creating superior MLIPs for specific applications lies the imperative for high-quality data. However, obtaining the data for vast chemical spaces of interest is often an intricate endeavor, frequently necessitating

expensive ab-initio simulations. Active learning stands out as the ideal solution, enabling efficient acquisition of high-quality data, thereby fundamentally enhancing the robustness and applicability of MLIPs across different chemical systems.

In this light, we introduce `CURATOR`, a comprehensive autonomous active learning workflow designated for the construction of high-fidelity graph neural network potentials. This workflow integrates state-of-the-art equivariant message-passing neural networks (MPNNs) with fast and reliable uncertainty quantification techniques, pivotal for driving informative data selections. The essence of `CURATOR` lies in its effectiveness at identifying batches of structures that offer maxmimal model improvement during retraining. This is achieved by thoughtfully considering both model uncertainties and the inherent diversity of atomic configurations using efficient batch active learning algorithms. Tested rigorously across several chemical systems, our approach demonstrates a significant enhancement in data acquisition efficiency, leading to a remarkable reduction in the training duration and resources required for constructing MLIPs. In addition, the workflow provides a useful uncertainty toolbox that contains multiple uncertainty estimation methods that can provide fast and reliable estimation of model confidence. This ensures that the active learning iteration will not sample unphysical atomic configurations and that the production simulations powered by MLIPs are always reliable. We suggest the employment of the local Mahalanobis distance metric as it offers rapid and trustworthy uncertainty estimation. Moreover, the integration with the `myqueue` task scheduler ensures a seamless and automated progression of tasks on modern computer clusters, enhancing the efficiency of the workflow further.

Introduction

Recently, machine-learned interatomic potentials (MLIPs) have found successful applications across various domains such as materials science, molecular physics, and chemistry.[1] The growing acceptance of MLIPs within the community is primarily due to their computational efficiency, which rivals that of empirical force fields, and their accuracy, which

matches established ab-initio methods. Typically, a machine learning interatomic potential learns the relationship between the atomic configurations and the potential energy surface of the chemical system. For the consideration of data efficiency and accuracy, the models should be capable of exploiting the invariances/equivariances of the physical systems upon space transformations including rotation, translation, reflection, and permutation of the same type of atoms, which excludes the use of simple atom coordinates as the structural representation. Behler and Parrinello[2,3] firstly introduced the high-dimensional neural network potential (NNP) in which the total energy of a chemical system is decomposed to individual atomic energies. The atomic energies are predicted by the neural networks that take the atom-centered symmetry functions (ACSFs)[3] as the features to describe the local atomic environments, which ensures the model can be invariant with respect to translation and rotation. The advent of ACSFs has led to the emergence of numerous descriptor-based machine-learning potential designs that use predetermined rules to transform the local environment of an atom into the input vector for regression. Examples of such models include various variants of the initial Behler-Parrinello neural network (e.g. ANI,[4,5] TensorMol,[6] SimpleNN[7]) and kernel-based models like sGDML[8] and GAP.[9] A notable limitation of such MLIPs is the need for extensive testing and physical/chemical insight from experts for parameter selection to manually create the features. The efficiency of these models is strongly influenced by the selection of descriptors. Furthermore, to accurately describe multi-element systems, these models typically require a larger set of descriptors because they omit atomic type information. This omission can lead to additional computational costs and compromise the performance of the models for complex chemical systems. To overcome these challenges, end-to-end NNPs have emerged that are capable of directly learning the mapping from nuclear charges and Cartesian coordinates of atomic structures to atomic features, all within the model itself. Most end-to-end NNPs have been inspired by the graph neural network architectures,[10] specifically referred to as message-passing neural networks (MPNNs).[11] In the context of MPNNs, atomic structures are conceptualized as undirected graphs, where atoms

are depicted as nodes and atomic bonds serve as the edges between these nodes. Geometric information (radial distance and angles) from neighboring nodes within the cutoff radius is collected through a message layer to compute the features of a specific atom, which is subsequently refined by an update layer. Such a message-passing scheme is performed iteratively to refine the node features, which are finally fed into a simple feed-forward neural network to predict the desired properties of the chemical systems. Prominent examples include DTNN,[12] PhysNet,[13] Schnet,[14,15] and DimeNet[16] etc. By operating on interatomic distances and using scalar feature representations, these models ensured that the model output and atomic features were invariant to rotations and translations. However, it is noteworthy that many essential chemical properties, such as forces and dipole moments, are equivariant to rotations of the chemical systems. Using rotationally-invariant features might result in the information loss of these directional properties, compromising model performance. A practical solution is to use advanced feature representations like vectors and tensors, combined with rotationally equivariant message and update functions.[17,18] For example, Batzer et al. introduced the NequIP architecture, where model equivariance is realized by encoding relative position vectors through spherical harmonics and utilizing Clebsch-Gordon coefficients for a higher-order tensor product.[19] This approach significantly enhances model accuracy and showcases outstanding data efficiency. Additionally, incorporating higher body orders interactions, instead of limiting to just two-body terms in the message functions, can further enhance model performance, as evidenced by Batatia et al.[20] In addition, although the simulation speed of MLIPs is significantly faster than ab-initio methods, it is still far away from satisfaction. There is still room for these MLIPs to be better optimized and accelerated for realistic long- and large-scale simulations.

For MLIPs to effectively address real-world challenges, datasets derived from ab-initio calculations must comprehensively encompass the chemical conformational space. Capturing the vast chemical space requires extended ab-initio simulations, which are often prohibitively expensive. An economical alternative is to use empirical force fields or pre-trained surrogate

models, though these often yield redundant configurations. It becomes essential to select representative structures from these datasets, annotate them using density functional theory (DFT) calculations, and incorporate them into model retraining — a process known as active learning. Iteratively employing this approach can significantly reduce the time and computational cost of constructing MLIPs for realistic simulations. There are two major concerns in this approach. Firstly, when applied to the undersampled configuration space, the pre-trained MLIPs may exhibit unpredictable behaviors, yielding nonphysical structures that are not meaningful for labeling. To address this, it is crucial that these MLIPs are uncertainty-aware, enabling the simulation to cease when the models lack confidence in their predictions. Moreover, the model uncertainty also serves as a vital metric to ensure that production simulations using MLIPs remain within their designated application domain. There are a variety of uncertainty estimation (UE) methods for chemical systems.[21] Gaussian process models are known for their inherent ability of straightforward uncertainty estimations,[22–24] while methods like deep ensembles[25–27] and Monte-Carlo dropout[28,29] are commonly employed in neural network models where explicit uncertainty estimations are not available. Secondly, structures chosen from MLIP simulations should be strategically selected with effective algorithms to maximize the improvement in the quality of the training dataset. Query-by-committee (QBC) is the most commonly adopted active learning strategy where data points are selected for labeling based on the highest uncertainty or disagreement among a committee of models; in this process, individual data points are iteratively chosen and labelled for subsequent model training.[27,30,31] In the context of constructing MLIPs, acquiring batches of data points in each active learning iteration is considerably more efficient than labeling individual ones, minimizing the frequency of model retraining. While the naive QBC strategy for batch active learning ensures individual data points are informative, it does not guarantee this for the entire batch. To address this challenge, significant efforts have been dedicated to developing batch active learning algorithms that exploit both the uncertainties and diversity of the candidate datasets.[32–35]

Despite significant advancements in model development, uncertainty quantification, and active learning algorithms, a comprehensive automated workflow integrating these components remains elusive. The generation of high-quality data largely relies on the expertise of materials scientists and chemists, who may not often be familiar with machine learning and active learning concepts. A challenging learning curve can deter their vital data contributions for machine learning potentials. An intuitive, autonomous workflow for developing high-fidelity MLIPs can help bridge the knowledge gap between MLIP users and developers. This workflow should include prioritizing model performance, ensuring thorough validation of production simulations, optimizing data acquisition efficiency, and maintaining user-friendliness.

In this paper, we introduce `CURATOR`, an autonomous active learning workflow devised for the construction of high-fidelity graph neural network potentials. This workflow seamlessly integrates cutting-edge equivariant MPNNs—specifically PaiNN,[36] NequIP,[19] and MACE,[20] targeting accurate predictions for specific properties within chemical systems. To ensure the robustness of simulations driven by the trained MLIPs, our approach incorporates a variety of uncertainty quantification techniques. We have incorporated efficient active learning strategies that can efficiently identify the most informative batches of structures from production simulations, adaptively enhancing model reliability and expanding their applicability across a broader chemical space. These strategies exploit both the model uncertainties and diversity of the candidate atomic configurations, improving the efficiency of batch mode data acquisition. Through rigorous testing across diverse chemical systems—ranging from simple molecules to intricate periodic molten glass structures, we demonstrate that the algorithms can remarkably enhance data acquisition efficiency. This, in turn, substantially reduces the time required to train high-quality MLIPs. In order to further accelerate the simulation speed of these MLIPs, we have developed an efficient gradient computation method that calculates forces and stress based on the energy derivative with respect to relative position vectors. Lastly, by integrating the entire workflow with myqueue,[37] we have

achieved full automation in the task scheduling on modern computer clusters for various job types within the framework.

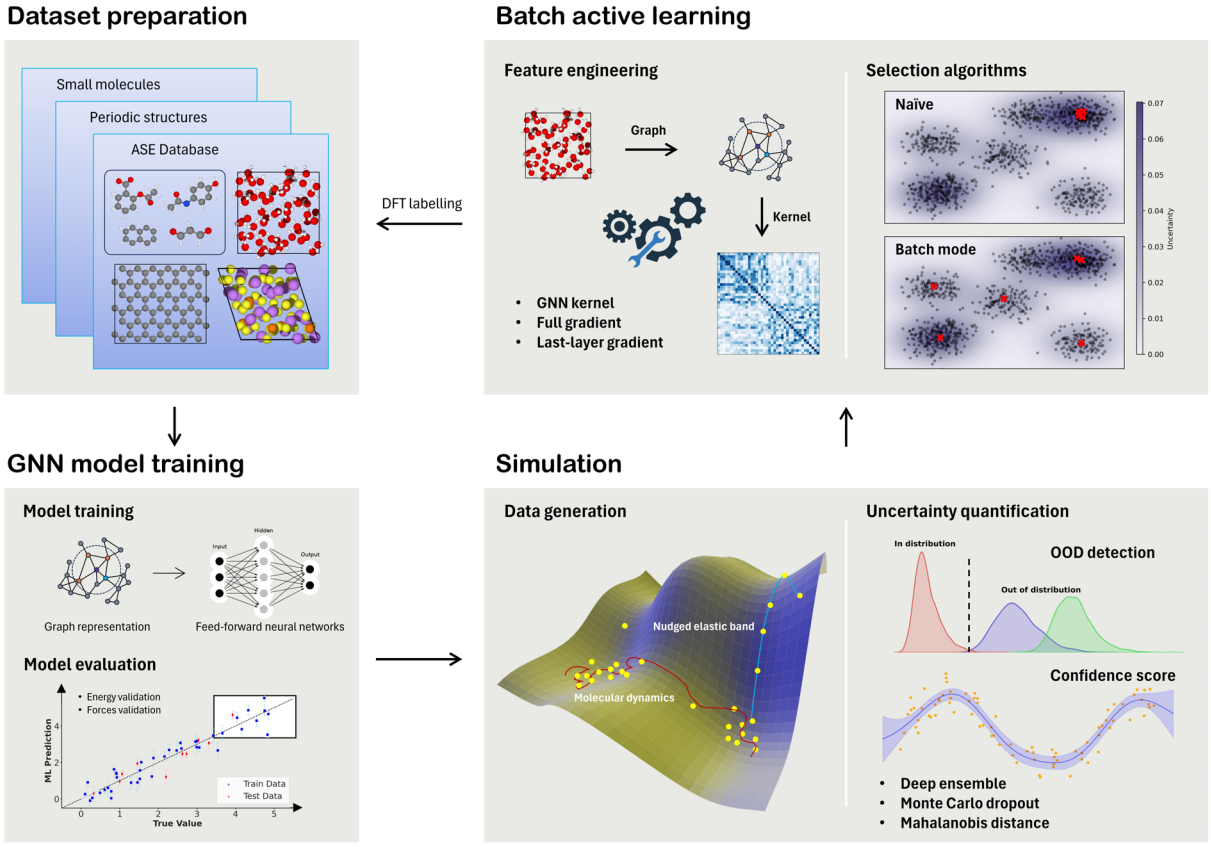

Figure 1: Schematic diagram of active learning workflow

Figure 1 outlines the various procedures for fitting MLIPs. Initially, users must provide a small dataset comprising atomic configurations derived from DFT calculations. This dataset could originate from diverse calculations, such as a short MD trajectory, configurations from structural optimizations, or nudged elastic band calculations, among others. This initial dataset serves as the foundation for training the GNN model. Within this process, atomic configurations are mapped into graph representations, which are then modeled using feed-forward neural networks. Training stops when there is no improvement in validation error over a specified number of steps. The resulting models can then be used to generate data via methods like molecular dynamics, Monte Carlo simulations, or other user-specified applications. Using our reliable uncertainty toolbox, simulations are guaranteed to stay within the

application domain of the trained models; if not, the simulations are immediately stopped. The much improved computational efficiency of MLIPs allows for the fast generation of numerous candidate structures. Batch active learning algorithms are then used to identify the most informative batches for refining the model among these candidates. This involves feature engineering for maximizing the information of individual candidate structures and minimizing the overall memory usage for storing the information, and effective algorithms that exploit the model uncertainties and data diversity. The chosen data points are subsequently labelled via DFT single-point calculations and incorporated into the initial DFT dataset for model refinement. Such a process will be iteratively performed until the derived GNN models are reliable, accurate, and stable enough for designated simulations. In the subsequent sections, we delve into the specifics of various procedures within the workflow. The remainder of this paper is structured as follows:

1. **Machine learning interatomic potentials:** We present several state-of-the-art Message Passing Neural Networks (MPNNs) utilized in our workflow, summarizing the trade-offs between model accuracy and speed. We also introduce an efficient method for gradient computation.
2. **Batch active learning:** This section introduces the active learning methods employed in our workflow, detailing the features and crucial transformations used for representing atomic structures. The efficacy of various active learning strategies is demonstrated through several selected benchmark systems.
3. **Uncertainty Estimation Methods:** We outline the uncertainty estimation methods integrated within the workflow and assess their performance against several critical criteria relevant to practical applications.
4. **Autonomous workflow:** Here, we illustrate how the aforementioned components are seamlessly incorporated into our active learning workflow.
5. **Conclusion:** Finally, we conclude with our remarks, summarizing the key findings

and implications of our work.

Machine learning interatomic potentials

Model training and evaluation

The workflow integrates a series of cutting-edge equivariant message-passing neural networks, specifically PaiNN,[36] NequIP,[19] and MACE.[20] Within these models, each atom is linked to features that encompass tensors of various orders, ranging from scalars and vectors to even more complex higher-order tensors. Leveraging these high-order features guarantees the rotational equivariance of the model, enhancing the accuracy of predictions related to directional attributes, such as dipole moments and forces. Adopting the notations from ref.,[19] the feature vectors $V_{acm}^{(l,p)}$ can be indexed by the rotation order l and the parity notation p. Here, the term "rotation order" refers to a non-negative integer l=0,1,2,... and the parity p can either be 1 or -1. Together, they label the O(3) irreducible representations of atomic features. Among the discussed models, PaiNN exclusively employs l=1 vectors and ignores parity transformation. This approach simplifies the dimensionality of the features, enabling PaiNN to achieve faster training and inference speeds without necessarily compromising on accuracy. In contrast, NequIP typically utilizes l=2 vectors and takes into account parity transformation. While this improves model accuracy, it demands greater computational resources. In both architectures, usually more than three message-passing layers are required to achieve desired accuracy levels. MACE differentiates itself by incorporating higher body-order interactions in its message functions, allowing for only two message-passing iterations to attain high accuracy. This design potentially optimizes computational efficiency while maintaining excellent model performance. For more detailed information on the models, please refer to the respective publications.

Figure 2 illustrates the trade-off between model accuracy, quantified by the mean absolute error (MAE) of forces, and inference speed. The models are trained using the aspirin

molecule data from the MD17 dataset, comprising 2,000 training, 1,000 validation, and 5,000 independent test data points. More details about the error metrics and inference speed for each model can be found in Table S1. The inference time for each model is evaluated on a diamond structure with 1000 atoms using an NVIDIA V100 GPU. This structure has an average of 86 neighbors per atom with a cutoff radius of 5.0 Å. It is important to highlight that the inference time generally remains consistent regardless of the number of atoms in the system, until all the GPU threads are occupied. As anticipated, the error diminishes with an increase in the number of message-passing layers and node feature size. However, beyond a certain threshold, increasing the number of layers and node feature size yields only marginal improvements in accuracy. It is clearly seen that both NequIP and MACE have shown outstanding model accuracy, while at a cost of much heavier computation as shown in Figure 2c and d. Therefore, it is recommended to use cheap PaiNN model for collecting training data points with active learning while to use more accurate models like MACE or NequIP for final production simulations.

Efficient gradient calculation

In the original implementations of GNN model, the total potential energy of the chemical system is calculated by aggregating individual atomic energies. Concurrently, atomic forces are generally derived from the negative gradients of the atomic energy with respect to atomic coordinates:

$$E = \sum_{i \in N} E_i \tag{1}$$

$$\vec{F_i} = -\nabla_i E \tag{2}$$

This approach respects the energy conservation constraint, which is important for improving the stability of simulations such as molecular dynamics.[38] Notice that the models never directly use the coordinates of atoms $\vec{r_i}$ to determine atomic energies. Instead, they rely

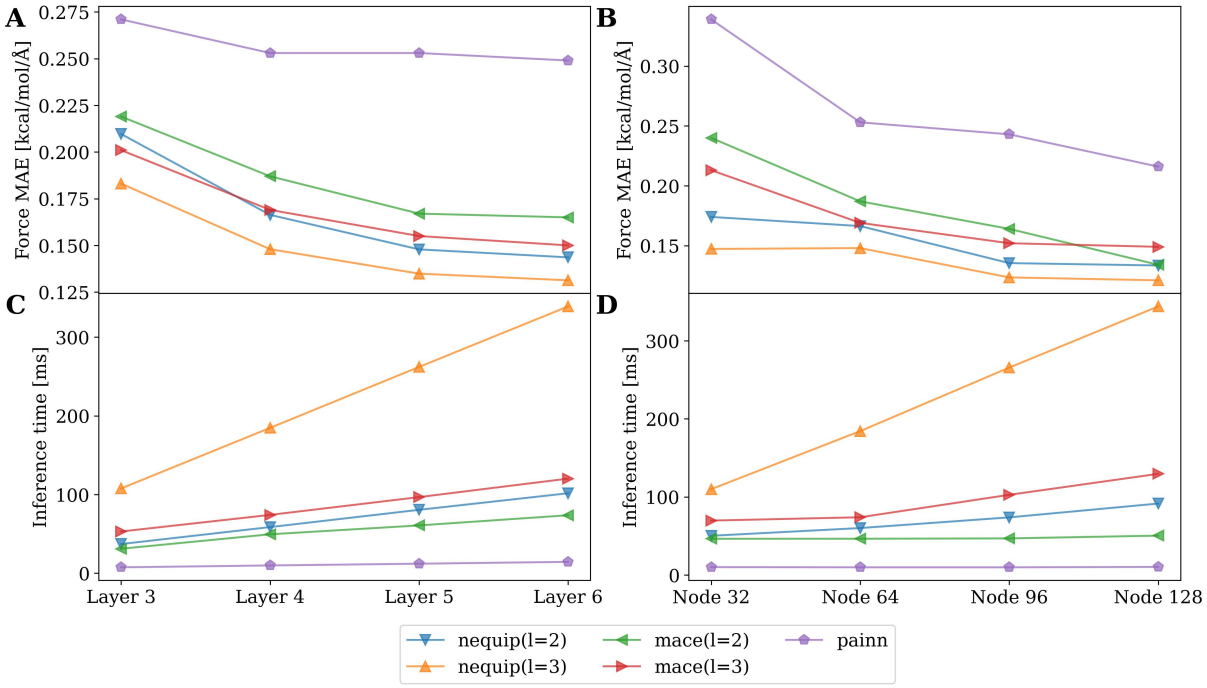

Figure 2: (a) and (b) Model accuracy (force MAE) of employed models plotted against the number of message-passing layers and the number of node features, respectively. (c) and (d) Inference time of employed models plotted against the number of message-passing layers and the number of node features, respectively.

solely on the relative position vector $\vec{r_{ij}} = \vec{r_j} - \vec{r_i}$ and its length $\| \vec{r_{ij}} \|$ in the message layers, which are typically obtained via neighbor-list algorithms from various codes like ASE,[39] ASAP3,[40] MatScipy,[41] or NNPOps.[42] Therefore, the atomic energy is exclusively a function of $\vec{r_{ij}}$:

$$E_i = E_i(\{\vec{r_{ij}}\}_{i \neq j}) \tag{3}$$

The automatic differentiation feature in PyTorch,[43] which is typically the backend of most GNN models, enables the convenient computation of negative gradients of total potential energy with respect to the model inputs, i.e. relative position vectors. Yet, the derivative $\partial \vec{r_{ij}} / \partial \vec{r_i}$ remains a missing map for force calculations. For non-periodic systems, this derivative is straightforward to compute, whereas periodic systems require consideration of cell displacements, adding extra computational overhead during data preprocessing. In contrast,

our implementation calculates forces as follows:

$$\begin{aligned}
\vec{F_i} &\equiv -\frac{\partial E}{\partial \vec{r_i}} \equiv -\sum_i \frac{\partial E_i}{\partial \vec{r_i}} \\
&= -\sum_{j \neq i} \left(\frac{\partial E_j}{\partial \vec{r_i}} \right) - \frac{\partial E_i}{\partial \vec{r_i}} \\
&= -\sum_{j \neq i} \left(\sum_{k \neq j} \frac{\partial E_j}{\partial \vec{r_{jk}}} \frac{\partial \vec{r_{jk}}}{\partial \vec{r_i}} + \frac{\partial E_i}{\partial \vec{r_{ij}}} \frac{\partial \vec{r_{ij}}}{\partial \vec{r_i}} \right) \\
&= -\sum_{j \neq i} \left(\frac{\partial E_i}{\partial \vec{r_{ij}}} - \frac{\partial E_j}{\partial \vec{r_{ji}}} \right)
\end{aligned} \tag{4}$$

In this way, the forces can be computed with only $-\partial E/\partial \vec{r_{ij}}$ that can be directly obtained with automatic differentiation. This bypasses the need to compute cell displacements and re-calculate relative position vectors, streamlining the process. Additionally, by using the neighbor list of individual atoms, we can independently determine the total forces for each atom, which offers significant potential for massively parallel implementation.

Moreover, this method can also notably reduce the effort required to compute the stress of the chemical system by using an explicit analytical expression for virial tensors. Typically, the stress tensors of a periodic system can be calculated with the first-order derivative of the total energy E with respect to small strains.[44] This method requires applying a symmetrical, infinitesimal strain deformation to the periodic system prior to the model prediction. Following this, the gradients of the total energy related to the strain tensors must be calculated. This process doubles the computational burden of gradient calculations, which represent the most significant computational expense in model prediction. In our implementation, it is worth noting that the computed force is pairwise and adheres to Newton's third law:

$$\vec{F_{ij}} = -\vec{F_{ji}} = -\frac{\partial E_i}{\partial \vec{r_{ij}}} + \frac{\partial E_j}{\partial \vec{r_{ji}}} \tag{5}$$

where $\vec{F_{ij}}$ is the force exerted by atom j on atom i. The viral tensors can then be calculated

by:

$$W = \sum_i W_i = -\frac{1}{2} \sum_i \sum_{j \neq i} r_{ij} \otimes \vec{F_{ij}} \tag{6}$$

This method employs an explicit expression for computing the virial stress tensors, eliminating the need for computationally intensive gradient calculations associated with stress tensors.

Batch active learning

Active learning operates in two primary modes. In naive active learning, the algorithm continuously selects and labels the single most informative sample, updating the model after each instance. If utilizing this strategy to select a batch instead of an individual sample, multiple informative but similar samples could be selected, potentially making the labeling of these samples redundant. This becomes especially critical in specific simulations. For example, in MLMD simulations, the extrapolative structures selected by naive active learning often present in a short time interval right before the simulation ends, as illustrated in Figure 4a and c. Even though each of these structures is individually informative, their similarity can lead to only marginal improvements during subsequent model retraining due to the overlap in information. On the other hand, batch active learning is designed to choose and annotate sets of samples simultaneously, prioritizing both uncertainty and diversity within the batch. Optimal batch active learning methods aim to choose samples that have high uncertainties while minimizing information redundancy, as illustrated in Figure 4b and d. Achieving this involves measuring atomic configuration similarities and strategically excluding similar structures from a batch. This process takes into account the features used to describe atomic structures and efficient selection algorithms, details of which will be discussed in the subsequent sections.

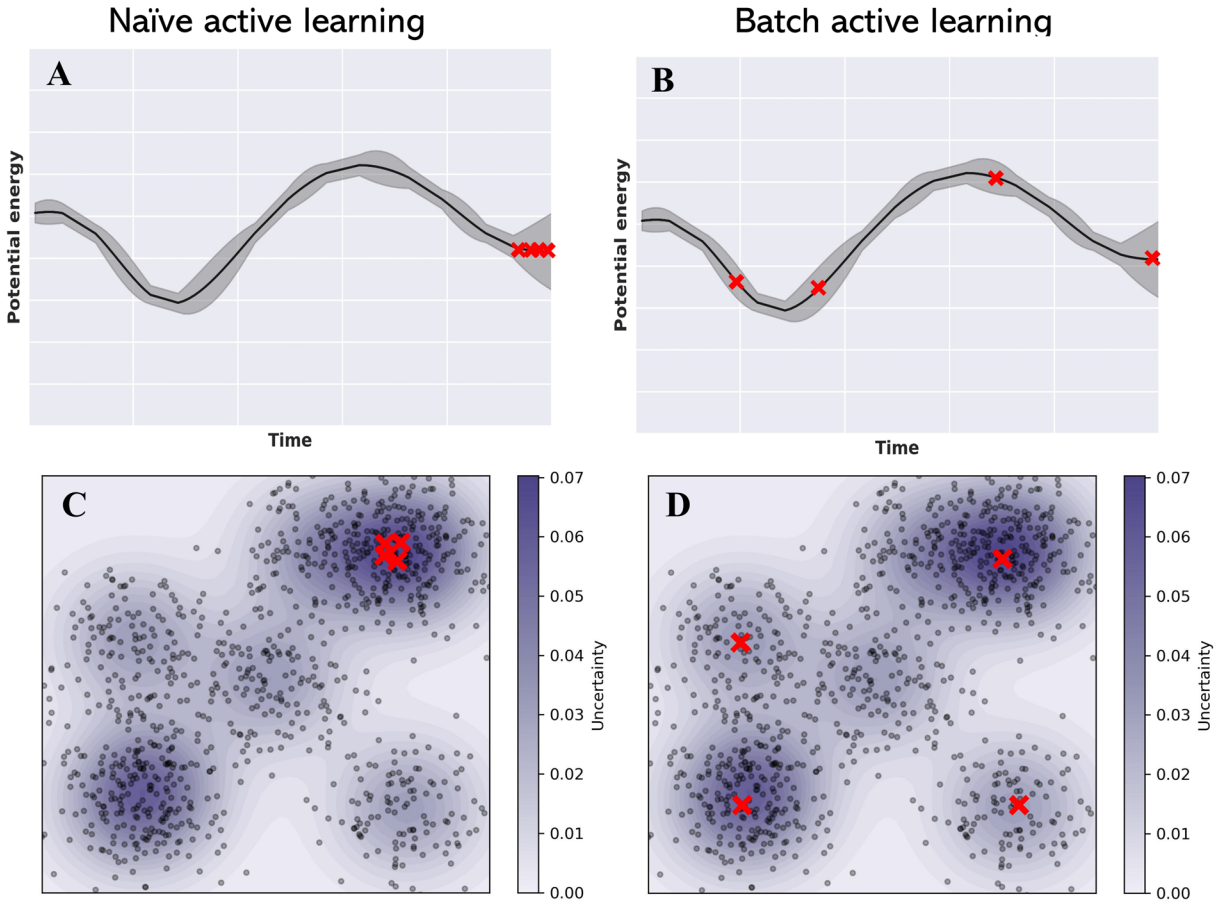

Figure 3: Schematic illustrations of active learning strategies: (a) naive active learning and (b) batch active learning methods for selecting data points from an MD simulation trajectory; (c) naive active learning and (b) batch active learning methods for selections in a two-dimensional space.

Feature engineering

Before exploring active learning selection, we must first extract features from our trained models for the candidate structures. Additionally, it is essential to understand the kernel matrix used in active learning. The following content is based on the framework from Ref.,[35] where further details are provided. For a sequence of atoms taken from these structures,

represented as $\mathcal{X} = (\boldsymbol{x_1}, ..., \boldsymbol{x_n}) \in \mathbb{R}^{n \times d}$, the corresponding feature matrix can be defined as:

$$\Phi(\boldsymbol{\mathcal{X}})^T = \begin{pmatrix} \phi(\boldsymbol{x_1})^T \\ \vdots \\ \phi(\boldsymbol{x_n})^T \end{pmatrix} \in \mathbb{R}^{n \times d_{feat}} \tag{7}$$

In this context, $\phi(x_i)$ represents the feature map for an individual atom derived from the model. The local environments of two atoms i and j then can be compared by using the similarity kernel $k(\boldsymbol{x_i}, \boldsymbol{x_j}) = \langle \phi(\boldsymbol{x_i}), \phi(\boldsymbol{x_j}) \rangle$. Expanding on this, we can compute the covariance matrix $k(\mathcal{X}, \mathcal{X}) = (k(\boldsymbol{x_i}, \boldsymbol{x_j}))_{i,j} \in \mathbb{R}^{n \times n}$ that encompasses all pairwise similarity within the feature matrix. There are various ways to construct the feature map and the kernel matrix. Besides, in order to make these kernels being more suitable to be applied for a selection method, some kernel transformation methods are often needed. In the following, we will introduce several kernels and transformation methods used in this study.

GNN kernel: The most intuitive approach for obtaining feature maps is leveraging the scalar node features derived from the outputs of the message-passing layers within MPNNs. This corresponds to the feature map ϕ_{gnn} and the graph neural network kernel k_{gnn}. Although evaluating this kernel through model prediction is generally fast and convenient, it solely contains the information necessary for computing the potential energy of the chemical system, ignoring the gradients of the systems. This could potentially limit its ability to accurately describe the atomic environment, consequently compromising the effectiveness of batch selection methods.

Full gradient kernel: Compared to GNN kernel, the full gradient kernel takes use of all gradients from the model to construct the feature map, which can be expressed as:

$$\phi_{grad}(\boldsymbol{x}) := \nabla_\theta f_{\theta_T}(\boldsymbol{x}) \tag{8}$$

where θ_T is the parameter vector of the trained model. The intuition of this method is

that the magnitude of the gradients implies the required adjustments of the parameters in different dimensions, and thus can be used to evaluate the distance between different samples. Besides, it also indicates the gap between predictions and correct values, enabling it to be a potential indicator of model uncertainty.[45]

The number of parameters in deep learning models can often be large, therefore it is intractable to get the gradients of all these parameters and to use the ultra-high-dimensional features for selection. Fortunately, the feature map ϕ_{grad} can be simplified by using the product structure of NNs, which can significantly reduce the runtime and memory usage for kernel evaluation.[35]

$$\boldsymbol{z}_i^{(l+1)} = \tilde{\boldsymbol{W}}^{(l+1)}\boldsymbol{x}_i^{(l)},\ \tilde{\boldsymbol{W}}^{(l+1)} := (\boldsymbol{W}^{(l+1)}\ \boldsymbol{b}^{(l+1)}) \in \mathbb{R}^{d_{l+1}\times(d_l+1)},\ \tilde{\boldsymbol{x}}_i^{(l)} = \begin{pmatrix} \boldsymbol{x}_i^{(l)} \\ 1 \end{pmatrix} \in \mathbb{R}^{d_l+1} \quad (9)$$

$$\phi_{grad}(\boldsymbol{x}_i^{(0)}) = \left(\frac{d\boldsymbol{z}^{(L)}}{d\tilde{\boldsymbol{W}}^{(1)}}, \ldots, \frac{d\boldsymbol{z}^{(L)}}{d\tilde{\boldsymbol{W}}^{(L)}}\right) = \left(\frac{d\boldsymbol{z}_i^{(L)}}{d\boldsymbol{z}_i^{(1)}}(\tilde{\boldsymbol{x}}_i^{(0)})^T, \ldots, \frac{d\boldsymbol{z}_i^{(L)}}{d\boldsymbol{z}_i^{(L)}}(\tilde{\boldsymbol{x}}_i^{(L-1)})^T\right) \quad (10)$$

Given that ϕ_{grad} encompasses gradient contributions across different layers, sometimes it is required to balance the magnitudes of the gradients in different layers via parameter initialization[46] or normalize the gradients post hoc. While the aforementioned derivations were initially intended for fully connected neural networks (NNs), they can also be extended for application to the FFNN component within MPNNs. This adaptation is precisely how the full gradient kernel was employed in the context of this study.

Last layer kernel: The dimensionality of a full gradient feature map can often be too large. A simple approximation to this is only using the gradients of parameters in the last layer of NNs as the feature map ϕ_{ll}.[47] From equation 10, it is evident that ϕ_{ll} is just the input of the last layer.

Average transformation: Note that the number of atoms in the pool dataset can be vast, often ranging from several millions to billions. Direct pairwise comparisons pose significant challenges in terms of memory consumption and computational efficiency. Thus,

merging the local feature maps of atoms to generate a global similarity measurement for structures is a more practical approach. When comparing two structures, a straightforward method is to use the average kernel. It is important to clarify that in this context, the term "average kernel" is somewhat misleading. It encompasses both the mean feature map of a group of atoms relative to a structure and the cumulative sum of feature maps. Mathematically, this can be represented as:

$$\phi(\boldsymbol{S_i}) = \phi_{\rightarrow avg}(\boldsymbol{x}) = \sum_{n=1}^{N_{atoms}} \phi(x_n) \tag{11}$$

where S_i denotes a structure and $\phi_{\rightarrow avg}(\boldsymbol{x})$ denotes the transformation for feature maps. This notation will also be used for other transformations hereafter. Although this method can lead to some information loss, its small computational cost can greatly accelerate the selection and minimize memory consumption. More accurate methods like regularized entropy match (REMatch) can also be used to construct the global similarity kernel.[48]

Diagonal kernels: Diagonal kernels correspond to the metrics that are used for naive active learning. These metrics can individually indicate the informativeness of selected samples while capturing no correlation between them. There are multiple ways to select these metrics. When the labels (i.e., material properties like energy and forces) of samples are known, the absolute error (AE) between true values and predictions can serve as a suitable indicator for the informativeness of individual samples. The absolute error of energy and forces are considered in this study, which can be expressed as:

$$\triangle E(S) = |E^{pred} - E^{true}| \tag{12}$$

$$\triangle F(S) = \frac{1}{3N_{atoms}} \sum_{i=1}^{N_{atoms}} \sum_{j=1}^{3} |\vec{F}_{ij}^{pred} - \vec{F}_{ij}^{true}| \tag{13}$$

These two kernels will be referred to as AE(E) and AE(F) hereafter. When the labels of samples are unknown, we can then use some sampling-based UE methods to evaluate the

disagreements between different predictions that obtained from different models or Monte-Carlo dropout, thus obtaining the uncertainty. There are multiple ways to calculate the disagreements, here we simply use the standard deviation of different predictions, which can be expressed as:

$$\sigma_E(S) = \sqrt{\sum_{n=1}^{N_{pred}} (E^{pred} - E^{true})^2} \tag{14}$$

$$\sigma_F(S) = \sqrt{\frac{1}{3N_{atoms}N_{pred}} \sum_{n=1}^{N_{pred}} \sum_{i=1}^{N_{atoms}} \sum_{j=1}^{3} (\vec{F}_{ij}^{pred} - \vec{F}_{ij}^{true})^2} \tag{15}$$

These two kernels will be referred to as QBC(E) and QBC(F) hereafter.

Random projections: Although the last-layer kernel can approximate the full gradient kernel to some extent, the information loss due to the discarded gradients can be large, undermining its ability to describe atomic environments. Random projections, also known as sketching, can be used to approximate a high-dimensional feature by a lower-dimensional feature.

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \frac{1}{\sqrt{p}} U\phi(x) \in \mathbb{R}^p \tag{16}$$

where $U \in \mathbb{R}^{p \times d_{feat}}$ is a random matrix with entries drawn from a standard normal distribution. In the case of feature map $\phi_{grad \to avg}(\boldsymbol{x})$, the following approximations are employed to simplify the sum and product of feature maps $\phi(\boldsymbol{x}) := (\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))^T$ and $\phi(\boldsymbol{x}) := \phi(\boldsymbol{x}_1) \otimes \phi(\boldsymbol{x}_1)$:

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \phi_{1 \to rp(p)}(\boldsymbol{x}_1) + \phi_{2 \to rp(p)}(\boldsymbol{x}_1) \tag{17}$$

$$\phi_{\to rp(p)}(\boldsymbol{x}) := \phi_{1 \to rp(p)}(\boldsymbol{x}_1) \otimes \phi_{2 \to rp(p)}(\boldsymbol{x}_1) \tag{18}$$

In this way, the full gradient feature map can be conveniently transformed into features with p dimensionality.

Gaussian process transformation: Gaussian process posterior transformation is derived from a Bayesian linear regression model with respect to feature $\phi(\boldsymbol{x})$, where the atom-wise property y_i can be modeled by $y_i = \boldsymbol{w}^T\phi(\boldsymbol{x}) + \varepsilon$.[49] After observing the training data $\mathcal{D}_{train}$ with inputs $\mathcal{X}_{train}$, it is well known that the posterior covariance $k(\boldsymbol{x}, \boldsymbol{x}^*|\mathcal{X}_{train})$ can be obtained by:

$$k(\boldsymbol{x}, \boldsymbol{x}'|\mathcal{X}_{train}) = k(\boldsymbol{x}, \boldsymbol{x}') - k(\boldsymbol{x}, \mathcal{X}_{train})(k(\mathcal{X}_{train}, \mathcal{X}_{train}) + \sigma^2\boldsymbol{I})^{-1}k(\mathcal{X}_{train}, \boldsymbol{x}') \tag{19}$$

$$= \phi(\boldsymbol{x})^T(I - \Phi(\mathcal{X}_{train})(\Phi(\mathcal{X}_{train})^T\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-1})\phi(\boldsymbol{x}') \tag{20}$$

Using the matrix inversion lemma (also known as the Woodbury matrix identity)[49] we can get:

$$k(\boldsymbol{x}, \boldsymbol{x}'|\mathcal{X}_{train}) = \sigma^2\phi(\boldsymbol{x})^T(\Phi(\mathcal{X}_{train})^T(\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-1}\phi(\boldsymbol{x}'), \tag{21}$$

which leads to an explicit feature map:

$$\phi_{\to gp}(\boldsymbol{x}) = \sigma(\Phi(\mathcal{X}_{train})^T(\Phi(\mathcal{X}_{train}) + \sigma^2 I)^{-\frac{1}{2}}\phi(\boldsymbol{x}). \tag{22}$$

This feature map can then be used for measuring the similarity between structures. The idea of this transformation can be seen as approximating the feed-forward NN in MPNNs as a Bayesian NN, providing a fast and robust way to evaluate model uncertainty. We will demonstrate in the subsequent section that this operation is indeed equivalent to using Mahalanobis distance for out-of-distribution detection.[50]

On the basis of the above kernels and transformations for atomic features, we come up with 6 different combinations, namely $\phi_{gnn\to avg}(x)$, $\phi_{grad\to rp\to avg}$, $\phi_{ll\to avg}$, $\phi_{grad\to rp\to avg\to GP}$, $\phi_{ll\to rp\to avg}$, $\phi_{grad\to rp\to avg\to GP}$. A suitable tool to evaluate their ability to accurately represent atomic structures and differentiate similar structures is t-SNE (t-Distributed Stochastic Neighbor Embedding),[51] which embeds the high-dimensional data points for visualization in

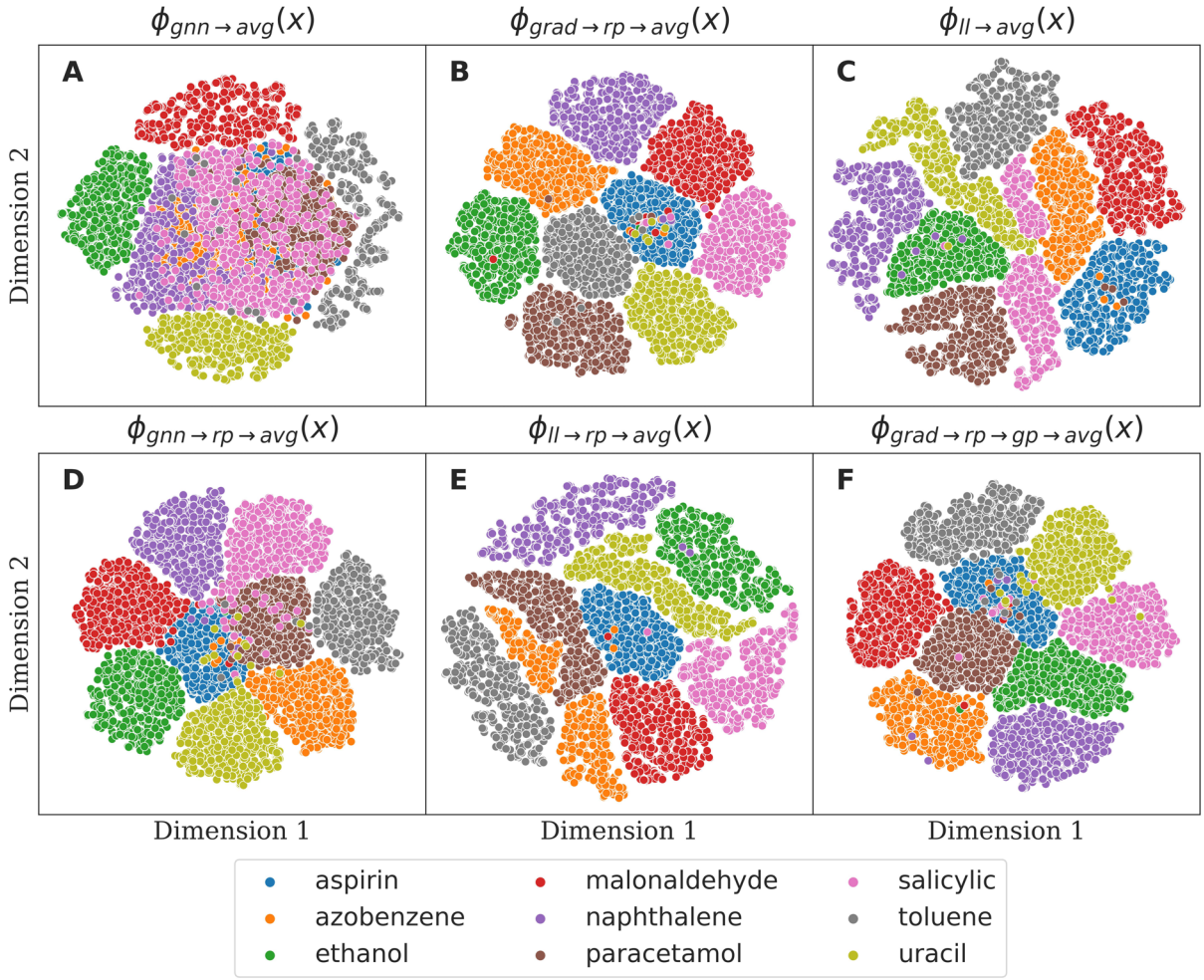

Figure 4: t-SNE plot of the applied kernels and transformations on the features derived from MD17 dataset.

a low-dimensional space by using probability distributions. Figure 4 depicts the distribution of various molecules from the MD17 dataset, visualized using t-SNE and the kernels mentioned above. From Figure 4a, it is evident that solely relying on the GNN kernel $\phi_{gnn \to avg}(x)$ is not enough to capture the structural differences between these molecules. Introducing random projection transformation to the GNN kernel offers a moderate improvement, as seen in Figure 4d, but distinguishing between different structures remains challenging. In stark contrast, both the full gradient kernel and the last gradient kernel demonstrate superior capability in capturing the structural characteristics of these molecules, evidenced by the distinct separations in Figures 4b, c, and e. We also evaluated the impact of Gaussian process trans-

formation, illustrated in Figure 4f. Regrettably, no obvious improvement is observed by using this transformation as it is mainly targeted for better uncertainty evaluation.

Batch mode selection

With the defined kernels above, we can then use some selection methods to select data points from the pool data set. Based on the above results that only $\phi_{grad\to rp\to avg}$ and $\phi_{ll\to avg}$ well represented the distribution of atomic structures, only these two kernels are employed for selection. In the following, we will briefly introduce several selection methods used in this study. '[35,52] Here we use $\mathcal{X}_{pool}$, $\mathcal{X}_{sel}$, $\mathcal{X}_{batch}$ to denote the pool dataset, selected data points, and the batch to be selected, respectively.

Random: Random selection will serve as a baseline for the selection methods and will be denoted as RANDOM. The batch data points $\mathcal{X}_{batch}$ will randomly draw from an uniform distribution

$$\textsc{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) \sim \mathcal{U}(\mathcal{X}_{pool}) \tag{23}$$

The selection continues until N_{batch} number of data points have been collected.

Naive active learning: If using $k(\boldsymbol{x}, \boldsymbol{x})$ as the uncertainty of data point $\boldsymbol{x}$, naive active learning can be conceptualized as selecting data points corresponding to the maximum diagonal elements of $k(\mathcal{X}_{pool}, \mathcal{X}_{pool})$. This method encompasses all QBC methods that employ uncertainty as their selection criterion and will be termed as MAXDIAG hereafter. The selection strategy can be expressed as

$$\textsc{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{\boldsymbol{x}\in\mathcal{X}_{pool}}{\operatorname{argmax}}\ k(\boldsymbol{x}, \boldsymbol{x}). \tag{24}$$

This method only considered the informativeness of individual data points while ignoring their similarities, which can lead to similar or even identical data points in $\mathcal{X}_{batch}$.

Greedy determinant maximization: Compared to MAXDIAG, the determinant max-

imization approach, referred to as MaxDet, curates an optimal batch $\mathcal{X}_{batch}$ by maximizing the determinant of $k(\mathcal{X}_{sel} \cup X_{batch}, \mathcal{X}_{sel} \cup X_{batch})$. This can be formalized as

$$\textsc{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{\boldsymbol{x} \in \mathcal{X}_{pool}}{\operatorname{argmax}} \det(k(\mathcal{X}_{sel} \cup \{x\}, \mathcal{X}_{sel} \cup \{x\}) + \sigma^2 I) \tag{25}$$

This method accounts for the correlation among selected points, effectively ensuring uncertainty and data diversity within the chosen batches. Calculating the determinants of batches with diverse data points is usually intractable. To alleviate computational complexity, the greedy algorithm utilizing partial pivoted matrix-free Cholesky decomposition[53] is employed. Notably, this approach aligns with the D-optimal design principles previously applied in active learning for machine learning interatomic potentials.[54,55]

Largest cluster maximum distance: Largest cluster maximum distance (LCMD) is a clustering method that aims to categorize data points from the pool set $\mathcal{X}_{pool}$ by assigning them to predefined cluster centers in $\mathcal{X}_{sel}$. Initially, every point $\boldsymbol{x}$ in $\mathcal{X}_{pool}$ is assigned to its nearest cluster center from $\mathcal{X}_{sel}$, with distances typically computed using metrics like Euclidean:

$$c(\boldsymbol{x}) := \underset{\tilde{\boldsymbol{x}} \in \mathcal{X}_{sel}}{\operatorname{argmax}} \, d_k(\boldsymbol{x}, \tilde{\boldsymbol{x}}) \tag{26}$$

The size of these clusters is determined by the sum of the distances of each member to its cluster center:

$$s(\tilde{\boldsymbol{x}}) := \sum_{c(\boldsymbol{x}) = \tilde{\boldsymbol{x}}} d_k(\boldsymbol{x}, \tilde{\boldsymbol{x}})^2 \tag{27}$$

Following this, the point in the largest cluster that is at the maximum distance from its center is chosen as the next cluster center.

$$\textsc{NextSample}(k, \mathcal{X}_{sel}, \mathcal{X}_{pool}) = \underset{s(c(\boldsymbol{x})) = \max \; s(\tilde{\boldsymbol{x}})}{\operatorname{argmax}} \; d_k(\boldsymbol{x}, c(\boldsymbol{x})) \tag{28}$$

This iterative process continues until the desired number of cluster centers matches the batch size. This method emphasizes both the representativeness and diversity of data points, thus making the batch mode selection effective.

We conducted experiments on multiple datasets to assess the effectiveness of our selection methods and kernels. These datasets include: AIMD simulations of small molecules from the MD17 dataset (non-periodic),[38] an AIMD trajectory of bulk lithium thiophosphate, $Li_{6.75}P_3S_{11}$ (periodic),[56] and an AIMD trajectory of amorphous lithium phosphate, $Li_4P_2O_7$ (periodic).[19] With a wide range of base kernels, kernel transformations, and selection modes available to us, the potential combinations were vast. This would necessitate an exhaustive number of benchmark tests, making the task unfeasibly complex. To reduce the complexity of benchmark tests and maintain clarity in our analysis, we strategically limited our focus on a select few combinations that appeared most promising. we restricted the kernels to be the full gradient and the last layer gradient, as they demonstrated a superior capability in accurately representing different structures. Both kernels are transformed by random projections with a dimensionality of 500, which can be expressed as LL(RP) and GRAP(RP). Consequently, our tests were streamlined to the following combinations: Random, MaxDiag+$\{$AE(E), AE(F), QBC(F)$\}$, MaxDet+$\{$LL(RP), GRAD(RP)$\}$, and LCMD+$\{$LL(RP), GRAD(RP)$\}$. Furthermore, all active learning tests are conducted with PaiNN model because it demonstrated superior training efficiency without compromising too much accuracy, while we anticipate that other GNN models might yield similar results.

MD17 active learning tests: We first tested these batch active learning strategies on the MD17 dataset, which comprised of MD trajectories of small molecules. The primary objective is to assess the effectiveness of various batch active learning strategies, utilizing a minimal number of data points while maximizing accuracy. The MD trajectories contain a number of frames ranging from approximately 100,000 to 1,000,000. Most of these frames are similar, making them highly suitable for active learning tests. For each molecule, a subset of 1,000 samples will be reserved as a validation dataset for early stopping, and an additional

5,000 samples will be used for an independent test of the model that exhibits the smallest validation loss in each training. The models are trained on a combined loss of energies and forces, with the energy and force weights being 0.05 and 0.95 respectively. The training began with an initial training data set of 100 samples, drawn randomly from the remaining pool dataset. Throughout each test, the training dataset is increased by a batch size of 100 until a total of 1,000 samples have been collected. The training stops when the validation loss does not improve over 150 times of validation checks. Performance was measured using multiple error metrics, including energy-based and force-based metrics like mean absolute error (MAE), root mean square error (RMSE), and maximum error (MAXE). Particularly, force error metrics were highlighted due to their pivotal role in atomic simulations such as MD, NEB, and structural optimization, where energy typically serves merely as an observer. Moreover, it is typically more challenging to achieve satisfactory force predictions.

The learning curves for the salicylic acid molecule, with a batch size of 100, are illustrated in Figure 5. Observations from other molecules mirrored these findings, as seen in Figures S1 through S8. Notably, the LCMD+GRAD(RP) combination consistently yielded the smallest force errors, with MAE, RMSE, and MAXE values being 0.182, 0.286, and 0.868 kcal/mol/Å, respectively. This is in stark contrast to the baseline method Random, which exhibited force MAE, RMSE, and MAXE values of 0.289, 0.740, and 2.860 kcal/mol/Å. Remarkably, the LCMD+GRAD(RP) combination achieved similar force accuracy to the Random method but used only half the data points (500 configurations), recording 0.332, 0.563, and 1.768 kcal/mol/Å for force MAE, RMSE, and MAXE, respectively. As anticipated, some naive active learning strategies, notably MaxDiag+AE(E) and MaxDiag+AE(F), distinctly underperformed compared to Random, highlighting the crucial importance of utilizing refined batch active learning methods. It is worth noting that the force MAXE learning curve of LCMD+GRAD(GP) is notably stable. This metric is often associated with the stability of MD simulations, as large force errors can lead to the rapid collapse of a simulation within a short time interval. Consequently, we expect that this approach will considerably enhance

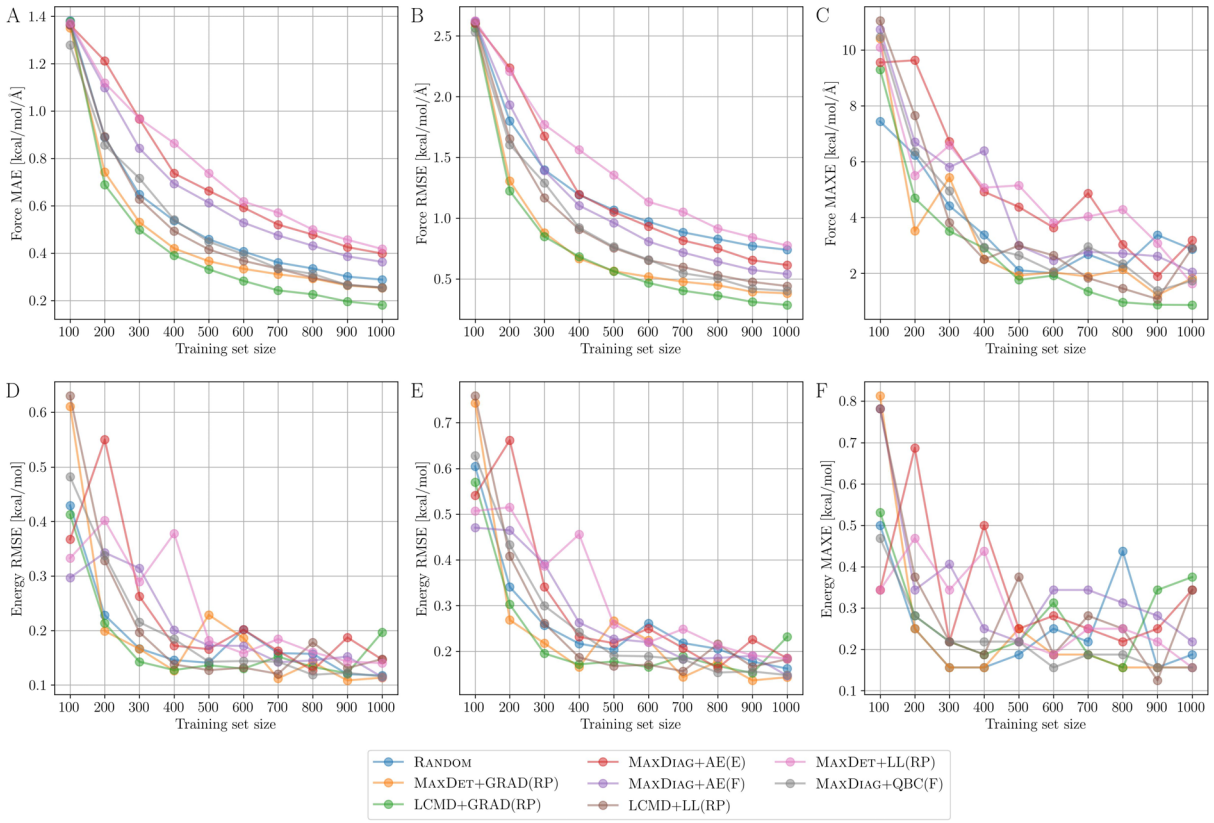

Figure 5: Learning curves for the salicylic acid molecule data from MD17. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

the stability of the simulations. Compared to the learning curves of force error metrics, those for energy error metrics show significantly greater fluctuations. We attribute this to the excessively low loss of weight assigned to energy. We expect that either increasing the loss weight for energy or training energy-only models could yield learning curves similar to those of force metrics.

LiPS active learning tests: In realistic simulations, chemical systems typically contain a larger number of atoms than the small molecules in MD17, and many of them are periodic structures. To evaluate batch active learning strategies in more general and more challenging contexts, we incorporated two additional datasets with periodic structures. All the active learning procedures employed remain consistent with the MD17 case. Specifically, we first

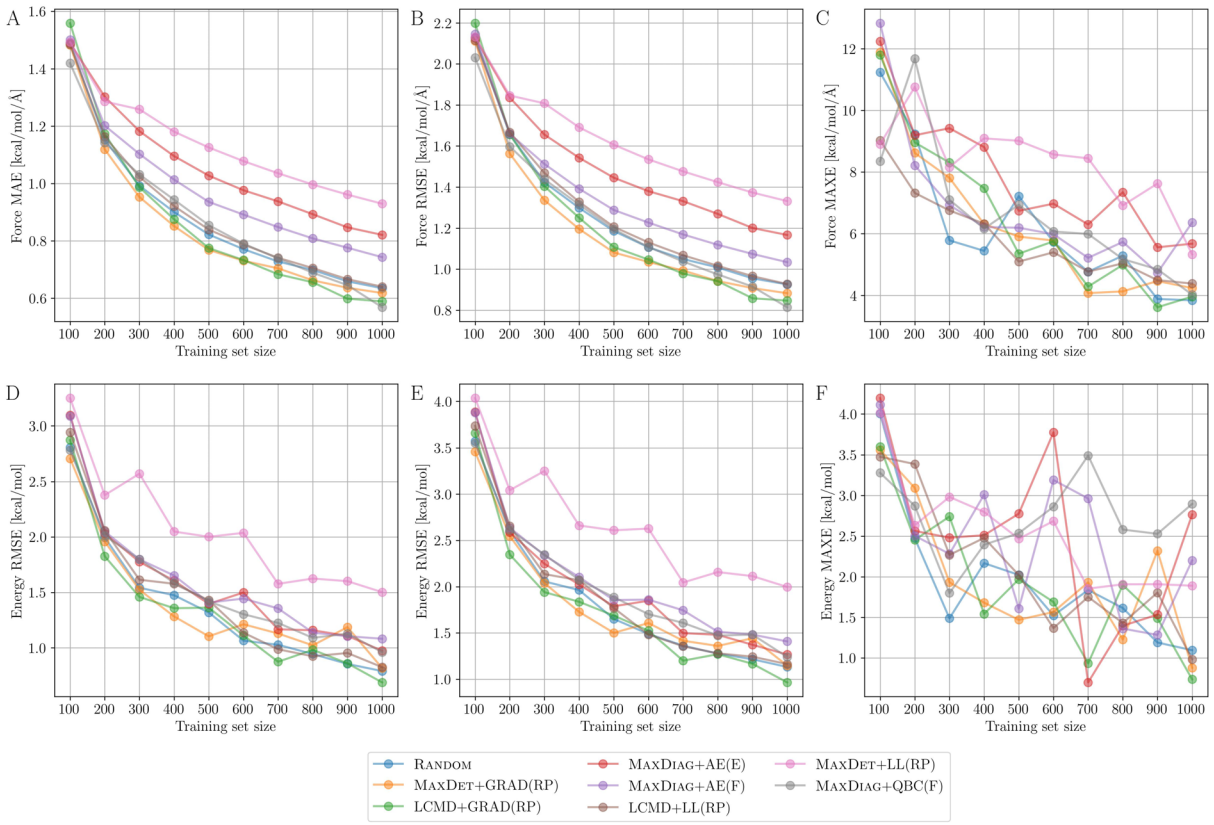

Figure 6: Learning curves for the salicylic acid molecule data from LiPS dataset. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

employ a dataset for $Li_{6.75}P_3S_{11}$ (LiPS), a crystalline superionic Li conductor with 83 atoms in a 12.38×12.26×12.44× Å triclinic cell.[56] This dataset contains 25,001 MD frames that are derived from a 50 ps NVT AIMD simulation at 520K with a timestep of 2 ps. From these, 1,000 frames serve as the validation set, 5,000 are designated for independent testing, and the rest form the pool set for active learning selection. As shown in Figure 6, it is evident that LCMD+GRAP(RP) consistently surpasses other methods in terms of force MAE and RMSE, with the exception of MaxDiag+QBC(F) in the last iteration. The force MAE However, we point out that the superiority of MaxDiag+QBC(F) is not because it is a more effective active learning strategy. Instead, its advantage stems from utilizing an ensemble of five

models for predictions. It is widely recognized that employing an ensemble can yield higher accuracy compared to a single model.[57,58] We observed that LCMD+GRAP(RP) showed only a marginal improvement over RANDOM in comparison to the MD17 cases. Meanwhile, both MAXDIAG-AE(E) and MAXDIAG-AE(F) methods lagged notably behind RANDOM. We believe that the limited conformational space explored by a 50 ps AIMD simulation may be the reason. As a result, a batch size of 100 appears sufficient for the RANDOM approach to sample a representative number of informative data points from the pool set. This leads to accuracy levels that are on par with LCMD. In contrast, methods based on MAXDIAG tend to sample data points over very short time intervals in this case, which can result in worse learning behaviors. We expect that batch active learning methods will be more crucial for the pool set with larger conformational spaces when using larger batch sizes. Additionally, the optimal batch size may vary depending on the specific chemical system under consideration.

LiPO active learning tests: We have extended our tests to the more intricate system of molten glass, $Li_4P_2O_7$ (LiPO).[19] This system comprises 64 Li, 32 P, and 112 O atoms within a 10.58×13.96×16.08 cell. The dataset encompasses 25,000 MD frames, sourced from a 50 ps NVT AIMD simulation at 3000K, using a time step of 2 ps. Despite LiPO having a greater number of atoms and exhibiting higher levels of disorder compared to LiPS, we observed a notable similarity in their learning behaviors, both exhibiting significant gaps between the learning curves of MAXDIAG-based methods and RANDOM with respect to force MAE and RMSE. Consequently, we infer that the effectiveness of batch active learning approaches is largely influenced by the conformational space of the pool set and the selected batch size, rather than the size and complexity of the chemical systems.

Based on our test results, several key insights emerge. Firstly, for pool sets with limited conformational space or when large batch sizes are used, we recommend avoiding the use of MAXDIAG-based methods for data point selection. Secondly, the consistently superior performance of LCMD+GRAP(RP) across various test systems and tasks suggests it is a reliable choice for all scenarios. Finally, our analyses further reveal that the GRAD(RP)

kernel consistently outperforms LL(RP) during active learning tests, emphasizing the crucial importance of selecting robust features that well represent atomic structures.

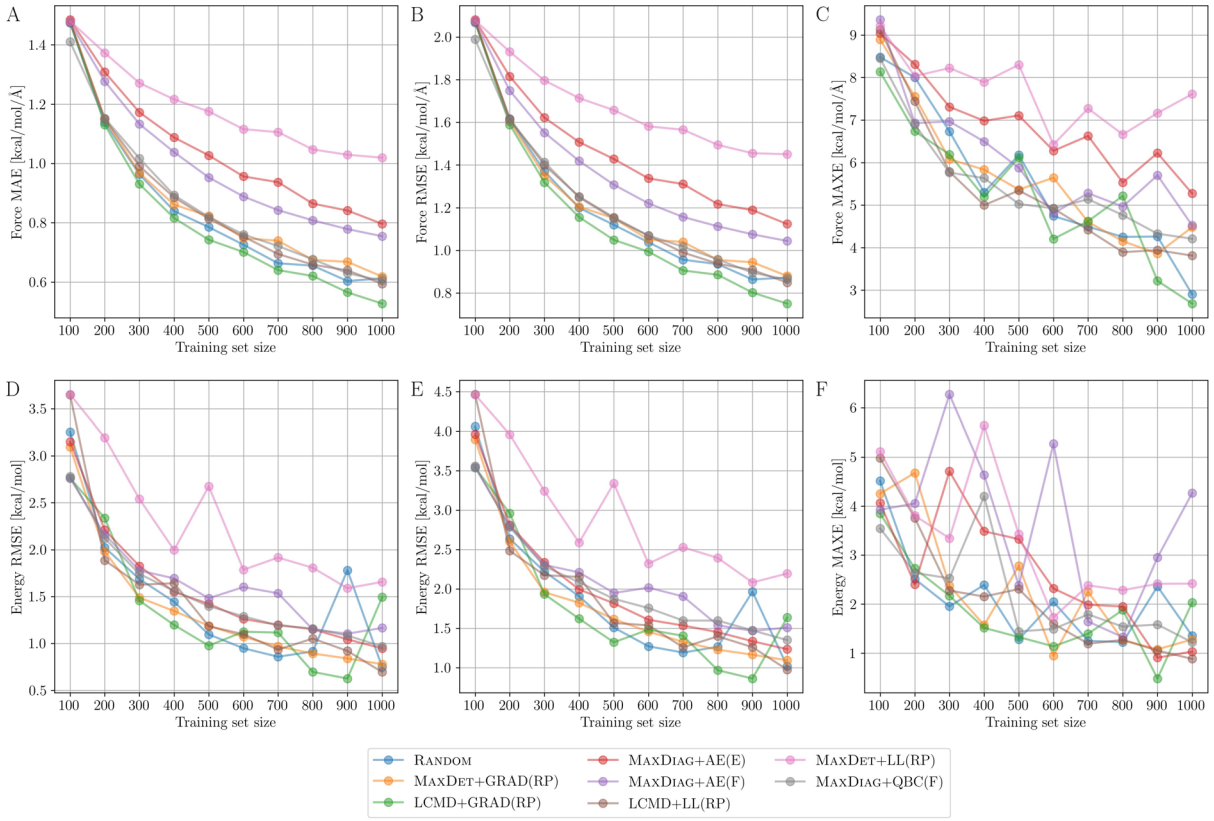

Figure 7: Learning curves for the salicylic acid molecule data from LiPO dataset. (a) The mean absolute errors (MAE), (b) root-mean-square errors (RMSE), and (c) maximum errors (MAXE) of atomic forces plotted against the training set size acquired from different active learning strategies. (d) The MAE, (e) RMSE, and (f) MAXE of total potential energies plotted against the training set size acquired from different active learning strategies.

Uncertainty-aware simulation

Although MPNNs have shown outstanding performance in the sampled configurational space, they tend to perform poorly on out-of-distribution (OOD) data. In this context, the role of uncertainty estimation (UE) becomes crucial, ensuring the model predictions are always reliable during active learning iterations or production simulations. Wollschläger *et al.*[59] introduced several crucial criteria for the effective application of UE methods:

- Accuracy: Precision in simulations is of utmost importance. An effective UE method should be able to deliver reliable uncertainty metrics without compromising model accuracy.
- Speed: Ideally, a UE method should be optimized such that it introduces marginal computational overhead, especially in some computationally heavy tasks like MD simulations.
- Confidence–aware: It is crucial that the method can discern and notify when a particular atomic structure is outside the domain of training.

Ideal UE methods should meet all these criteria to effectively handle the diverse and intricate tasks presented in atomistic simulations. UE methods can be roughly categorized into two groups based on how the predictions are made: sampling-based and sampling-free methods. Sampling-based methods, such as the deep ensemble and Monte Carlo dropout, rely on the disagreements among multiple predictions to determine uncertainty. A greater variance in predictions corresponds to increased uncertainty, and vice versa. On the other hand, sampling-free methods generally utilize a single forward pass to uncertainty through the analysis of the distributions of learned features. In our workflow, the following UE methods are available for various atomistic simulations.

Deep ensemble is considered the gold standard solution for uncertainty estimation.[58] An ensemble usually comprises diverse models trained with varied architectures or initializations. When these models generate different predictions, measuring disagreements such as the standard deviation among them can provide an estimation of uncertainty. In the context of atomistic simulations, the standard deviation of energies or forces can serve as an indicator of the uncertainty associated with an atomic structure, as illustrated in Equation 14 and Equation 15. Although ensembles often improve *Accuracy* and fulfill $Confidence–aware$,[30,60] they come at the cost of increased computational complexity, both in training and inference, thus fail at *Speed*.

Monte Carlo dropout (MCD) incorporates dropout into deep learning models, en-

abling the estimation of model uncertainty during predictions. By performing multiple forward passes with dropout, we can treat the collection of predictions as samples from a distribution, which captures the model uncertainty about its prediction for the input. Therefore, the formulations of energy and force uncertainties align with Equation 14 and Equation 15 as seen in the ensemble case. Since MC dropout involves deactivating neurons within a single model, it necessitates the training of only one model, thereby conserving substantial effort in the training process. Nevertheless, the random deactivation of neurons can occasionally undermine the predictive accuracy of the model. Furthermore, even though training is limited to a single model, inference still requires multiple forward passes, challenging the *Speed* criterion.

Mahalanobis distance quantifies the distance between a point and a distribution, taking into account the correlations of the data set and the scale of the features in different dimensions. Therefore, this method is very useful for detecting samples that are out of the distribution of the training data set. Formally, the Mahalanobis distance $d(\boldsymbol{S}_i, Q_S)$ between a structure $\boldsymbol{S}_i$ and a distribution of training set Q_S with mean μ_S and covariance matrix $\boldsymbol{\Sigma}_S$ is defined as:

$$d(\boldsymbol{S}_i, Q_S) = (\boldsymbol{S}_i - \mu_S)^T \boldsymbol{\Sigma}_S^{-1} (\boldsymbol{S}_i - \mu_S) \tag{29}$$

Clearly, the expression is equivalent to Equation 21 when a small noise is introduced to the covariance matrix $\boldsymbol{\Sigma}$ and the input is normalized to the training dataset. When employing the GNN base kernel, this approach can be seen as computing $k_{gnn\to avg\to gp}(\boldsymbol{S_i}, \boldsymbol{S_i})$, which is the diagonal element of kernel matrix $k_{gnn\to avg\to gp}(\boldsymbol{S}, \boldsymbol{S})$. Choosing a simple identity matrix as the covariance matrix translates to computing the Euclidean distance. We will demonstrate in subsequent tests the importance of the covariance matrix for reliable uncertainty estimation by comparing the Mahalanobis and Euclidean distances. It is worth noting that this method exclusively utilizes the features derived from a singular model forward pass

and leverages a precomputed covariance matrix to compute Mahalanobis distance. As a result, the model accuracy remains consistent with the original one, with only a marginal computational overhead introduced for evaluating the distance metrics.

Local Mahalanobis distance is different from Mahalanobis distance by reversing the order of sum and GP transformation, which can be represented as $k_{gnn\to gp\to avg}(\boldsymbol{S_i}, \boldsymbol{S_i})$. The corresponding Mahalanobis distance is then given by:

$$d(\boldsymbol{S}_i, Q_{\boldsymbol{x}}) = \sum_{x_i \in S} (\boldsymbol{x}_i - \mu_{\boldsymbol{x}})^T \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1} (\boldsymbol{x}_i - \mu_{\boldsymbol{x}}) \tag{30}$$

An immediate advantage of this modification is that the resulting uncertainty scales to the size of atomic structures. By ensuring that uncertainty is proportional to the structural size, this method offers a refined uncertainty estimation for structures of varying system sizes.

We evaluated the accuracy of models on the MD17 dataset and the inference speed of various UE methods, as presented in Tables 1 and S2. The ensemble consists of five individual PaiNN models, each trained using different splits for training and validation. Both local Mahalanobis and Mahalanobis distance use the original single model for prediction. For MCD, we tested the results with the dropout ratio 0.01, 0.05, 0.10, and 0.20, here only 0.1 and 0.2 cases are reported in Table 1. In all cases, we use randomly selected 5000 structures for training, 1000 for validation, and 5000 for independent tests. Among all the UE methods, the ensemble consistently exhibited superior accuracy by leveraging predictions from various models. However, a notable decline in performance was observed when MCD was applied to the original model. From these results, we can find that only MCD fails at the accuracy criterion. Another crucial factor for UE methods is their speed. Remarkably, Mahalanobis-based UE methods are about five times faster than both the ensemble and MCD, yet they offer comparable accuracy to the ensemble. Therefore, This makes them especially suitable for running heavy simulations.

To evaluate the performance of these methods in *confidence–aware* criteria, we employ

Table 1: MAE of PaiNN on MD17 with different UE methods (energies in kcal mol^{-1}, forces in kcal mol^{-1} Å^{-1})

		Ensemble	**Mahalanobis**	**MCD (p=0.1)**	**MCD (p=0.2)**
aspirin	Energy	**0.123**	0.129	4.324	9.932
	Forces	**0.098**	0.160	1.385	2.201
azobenzene	Energy	**0.137**	0.138	5.004	11.470
	Forces	**0.043**	0.063	1.134	1.821
ethanol	Energy	**0.051**	0.052	0.779	1.498
	Forces	**0.053**	0.088	0.707	1.101
malonaldehyde	Energy	**0.074**	**0.074**	0.855	1.749
	Forces	**0.080**	0.134	0.941	1.443
naphthalene	Energy	0.113	**0.112**	3.354	7.686
	Forces	**0.032**	0.043	1.012	1.603
paracetamol	Energy	**0.113**	0.118	3.654	8.495
	Forces	**0.068**	0.115	1.232	1.957
salicylic acid	Energy	0.107	**0.106**	2.938	6.734
	Forces	**0.055**	0.085	1.150	1.812
toluene	Energy	**0.092**	0.093	2.439	5.554
	Forces	**0.035**	0.050	1.005	1.572
uracil	Energy	0.105	**0.103**	1.629	3.624
	Forces	**0.040**	0.063	1.039	1.615

OOD detection based on the area under the receiver operating characteristic (AUC-ROC) curve. Specifically, we train a model on one molecule in MD17 and examine its ability to differentiate the remaining molecules using its uncertainty estimates. Ideally, the estimator should produce low uncertainties for the trained molecule and higher ones for the rest. Such behavior allows users to set a confidence threshold to trust model predictions when the uncertainty falls below this threshold. For performance evaluation, we calculate the area under the AUC-ROC curve of the uncertainty scores for both in-distribution (ID) and OOD data. A score approaching 1 signifies a better ability to differentiate OOD data using the specific UE method. Figure 8 shows the heatmap of the AUC-ROC score for each pairwise combination of molecules obtained with different UE methods. The rows show the molecule that the model is trained on and the off-diagonal columns are the respective OOD sample. On the diagonal, we expect a score of 0.5 while a score of 1 is optimal on the off-diagonals.

We found that the ensemble exhibited perfect separation between ID and OOD samples. In contrast, another sampling-based method MCD did not show a satisfactory ability for separating some pairs. Local Mahalanobis distance has shown comparable performance compared to the ensemble while using much less computational cost. From Figure 8d it is evident that using the local representation of atomic environments to calculate the covariance matrix can significantly improve the $confidence–aware$ ability of Mahalanobis distance. The order of GP transformation in the kernels does matter. We further investigated the influence of using different kernels as shown in Figure S9. It is clearly seen that only the GNN kernel exhibited high AUC-ROC scores on off-diagonals. Although full gradient and last-layer gradient kernels have shown exceptional performance in representing atomic structures via t-SNE, they crucially failed at differentiating the molecules using Mahalanobis distance. (influence of covariance matrix, influence of dropout ratio). If using an identity matrix as the covariance matrix, this corresponds to Euclidean distance. We demonstrated in Figure S10 that Euclidean distance is not able to achieve satisfactory OOD detection performance, without the use of a covariance matrix.

Our workflow provides all the mentioned UE methods described above, the users can choose suitable UE methods for their specific applications. For computationally heavy applications, local Mahalanobis distance is recommended, while for applications that need more reliable uncertainty estimations, the ensemble is recommended.

Automated workflow

Leveraging powerful selection methods and efficient UE techniques, we have established a comprehensive, automated active learning workflow. It is imperative to note that the different stages in this workflow necessitate varying resource allocations. For instance, training and machine learning-driven simulations predominantly utilize GPUs, whereas the annotation of informative batches often depends on DFT codes optimized for CPUs. Besides,

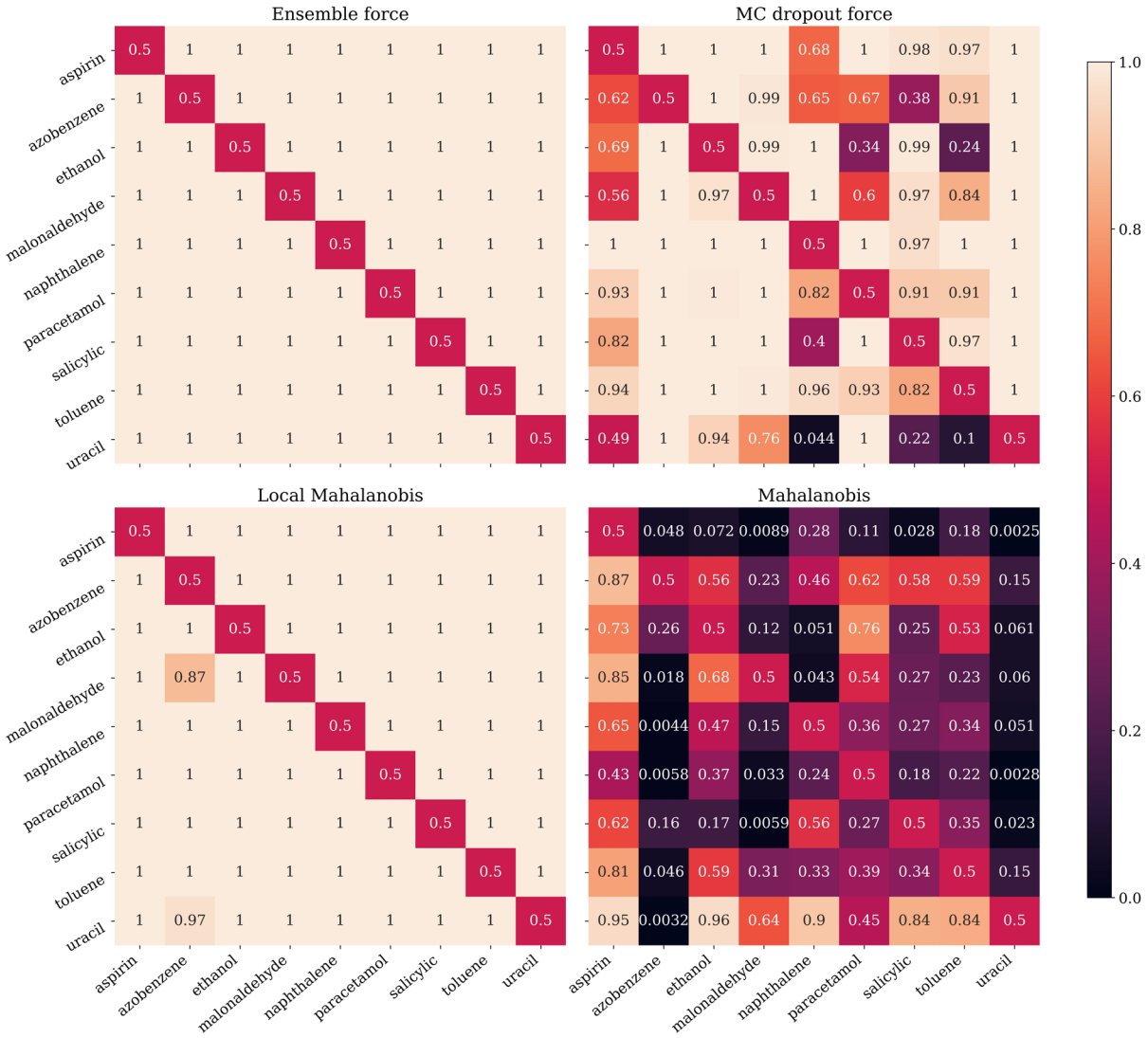

Figure 8: Heatmap displays the AUC-ROC values from SchNet on MD17. Each row represents a separate model, trained on the molecule listed to the left, and tested against all other molecules.

these tasks are structured in a fixed sequential order. This means that one first needs to ensure the successful completion of preceding tasks before initiating certain tasks. Recognizing and adhering to these dependencies is vital to ensure the workflow operates smoothly and efficiently. To manage job assignments across diverse hardware and inspect their execution states, we employ `myqueue`,[37] a cluster job manager, to assign jobs on different hard devices and to manage these jobs. Furthermore, given the need for specifying diverse hy-

perparameters throughout different phases in the workflow, we have adopted the Hydra[61] configuration framework, which allows the building of hierarchical YAML configurations. In addition, we have integrated PyTorch Lightning[62] to streamline the model training process. In the sections that follow, we demonstrate how this workflow can be adeptly employed to autonomously construct the MLIPs.

```
|-- data
|   |-- custom.yaml
|-- labelling
|   |-- custom.yaml
|   |-- gpaw.yaml
|   |-- qe.yaml
|   |-- vasp.yaml
|-- model
|   |-- representation
|   |   |-- mace.yaml
|   |   |-- nequip.yaml
|   |   |-- painn.yaml
|   |-- nnp.yaml
|-- selection
|   |-- default_selection.yaml
|-- simulation
|   |-- custom.yaml
|   |-- mc.yaml
|   |-- md.yaml
|   |-- neb.yaml
|-- task
|   |-- optimizer
|   |   |-- adam.yaml
|   |   |-- adam_amsgrad.yaml
|   |-- scheduler
|   |   |-- exponential.yaml
|   |   |-- reduce_on_plateau.yaml
|   |-- default_task.yaml
|-- trainer
|   |-- default_trainer.yaml
|-- __init__.py
|-- label.yaml
|-- select.yaml
|-- simulate.yaml
|-- train.yaml
|-- workflow.yaml
```

(a)

```yaml
defaults:
  - model/representation: painn
  - task/optimizer: adam
  - task/scheduler: reduce_on_plateau
  - simulation: md
  - labelling: vasp

data:
  datapath: ./water_dft.traj
  cutoff: 5.0
  batch_size: 16
  num_train: 2000
  num_val: 1000
  atomic_energies: auto
  atomwise_normalization: True

model:
  representation:
    num_interactions: 3
    num_features: 64

task:
  scheduler_monitor: val_loss
  optimizer:
    lr: 0.005
  scheduler:
    factor: 0.5

simulation:
  uncertainty: local_mahalanobis
  simulator: md
  params:
    load_traj: ./water_dft.traj
    max_steps: 1000000

selection:
  kernel: full-g
  selection: lcmd_greedy
  n_random_features: 500
  batch_size: 200

labelling:
  dft_code: vasp
  num_jobs: 4
```

(b)

Figure 9: Main figure caption for code listings

Figure 9a displays the predefined hierarchical YAML configurations in the code. We

provide a suite of default configurations for various components within the workflow, facilitating a vast number of combinations and extensive experimental runs via the command line interface (CLI). These configuration files are organized in an object-oriented manner, with each subdirectory containing files for different applications. files in an object-oriented way. Each subdirectory contains some choices for different applications. Situated in the top-level directory within the `configs` folder are four principal YAML files: `train.yaml`, `simulate.yaml`, `select.yaml`, and `label.yaml`, each corresponding to a respective phase in the active learning workflow. They define default hyperparameters, but users have the flexibility to adjust them via the CLI as needed. For instance, to train a model, one might use:

```
gnntrain model/representation=nequip data/datapath=water.traj
```

Additionally, users can craft their configuration files outside the default `configs` directory. By specifying `cfg=custom.yaml`, where `custom.yaml` is the custom configuration, one can easily employ it for desired experiments. Integrating the configuration files for each separate job leads to the overall configuration file `workflow.yaml`, which encapsulates hyperparameters for all phases in the workflow. Figure 9b showcases a user-defined configuration tailored for running active learning iterations. Parameters from this file will override the defaults set in `workflow.yaml`. This design empowers users to effortlessly manage and customize their tasks, facilitating the construction of diverse MLIPs. For a more in-depth exploration and a deeper understanding of our configuration system and its functionalities, readers are encouraged to visit our codebase. This resource offers comprehensive documentation and examples, ensuring clarity and ease of use for both newcomers and experienced users.

Conclusions

In this study, we tackled the existing challenges in the development and application of machine learning interatomic potentials for atomistic simulations. Although the power of

MLIPs has been previously verified, challenges such as efficient data collection, reliable tools for model confidence, and intricate procedures persisted. At the forefront of our contributions is the introduction of `CURATOR`, a comprehensive workflow that seamlessly integrates advanced active learning algorithms and reliable uncertainty estimation techniques for improving data acquisition efficiency and ensuring reliable production simulations.

The workflow encompasses state-of-the-art graph neural network models for accurate atomistic modelling. We re-implemented the gradient calculation and significantly accelerated the speed for stress calculation. We emphasized the importance of batch active learning in the collection of data sets. We show that our incorporation of batch active learning strategies effectively enables much-improved data acquisition efficiency. When evaluated on multiple benchmark datasets, our specific batch active learning strategies consistently outperformed others across various systems. This highlights their potential in significantly minimizing human efforts and computational expenses in generating MLIPs. To ensure the reliable application of trained models, we incorporated several uncertainty estimation methods into the workflow and compared their performance in terms of speed, accuracy, and confidence-awareness. The test results demonstrated that the Mahalanobis distance can serve as a fast and reliable UE method, with only fraction of the cost of ensembles while demonstrated comparable performance in accuracy and confidence-awareness.

The workflow has been made fully autonomous by combining the previously mentioned elements. By merging the Hydra-style configuration framework, Pytorch Lightning, and the robust task scheduler `myqueue`, the system is both functional and user-friendly, catering to both beginners and experts.

Acknowledgement

This work was supported by the Carlsberg Foundation through the Carlsberg Foundation Young Researcher Fellowship (Grant No. CF19-0304). The authors also thank the com-

putational resources provided by the Niflheim Linux supercomputer cluster installed at the Department of Physics at the Technical University of Denmark.

References

(1) Unke, O. T.; Chmiela, S.; Sauceda, H. E.; Gastegger, M.; Poltavsky, I.; Schütt, K. T.; Tkatchenko, A.; Müller, K.-R. Machine learning force fields. *Chemical Reviews* **2021**, *121*, 10142–10186.

(2) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters* **2007**, *98*, 146401.

(3) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of chemical physics* **2011**, *134*.

(4) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical science* **2017**, *8*, 3192–3203.

(5) Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E. TorchANI: A free and open source PyTorch-based deep learning implementation of the ANI neural network potentials. *Journal of chemical information and modeling* **2020**, *60*, 3408–3415.

(6) Yao, K.; Herr, J. E.; Toth, D. W.; Mckintyre, R.; Parkhill, J. The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics. *Chemical science* **2018**, *9*, 2261–2269.

(7) Lee, K.; Yoo, D.; Jeong, W.; Han, S. SIMPLE-NN: An efficient package for training and executing neural-network interatomic potentials. *Computer Physics Communications* **2019**, *242*, 95–103.

(8) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nature communications* **2018**, *9*, 3887.

(9) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters* **2010**, *104*, 136403.

(10) Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE transactions on neural networks* **2008**, *20*, 61–80.

(11) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. International conference on machine learning. 2017; pp 1263–1272.

(12) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nature communications* **2017**, *8*, 13890.

(13) Unke, O. T.; Meuwly, M. PhysNet: A neural network for predicting energies, forces, dipole moments, and partial charges. *Journal of chemical theory and computation* **2019**, *15*, 3678–3693.

(14) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems* **2017**, *30*.

(15) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of Chemical Physics* **2018**, *148*.

(16) Gasteiger, J.; Groß, J.; Günnemann, S. Directional message passing for molecular graphs. *arXiv preprint arXiv:2003.03123* **2020**,

(17) Thomas, N.; Smidt, T.; Kearnes, S.; Yang, L.; Li, L.; Kohlhoff, K.; Riley, P. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219* **2018**,

(18) Satorras, V. G.; Hoogeboom, E.; Welling, M. E (n) equivariant graph neural networks. International conference on machine learning. 2021; pp 9323–9332.

(19) Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications* **2022**, *13*, 2453.

(20) Batatia, I.; Kovacs, D. P.; Simm, G.; Ortner, C.; Csányi, G. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems* **2022**, *35*, 11423–11436.

(21) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. W. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology* **2020**, *1*, 025006.

(22) Bartók, A. P.; Kermode, J.; Bernstein, N.; Csányi, G. Machine learning a general-purpose interatomic potential for silicon. *Physical Review X* **2018**, *8*, 041048.

(23) Deringer, V. L.; Bartók, A. P.; Bernstein, N.; Wilkins, D. M.; Ceriotti, M.; Csányi, G. Gaussian process regression for materials and molecules. *Chemical Reviews* **2021**, *121*, 10073–10141.

(24) Musil, F.; Willatt, M. J.; Langovoy, M. A.; Ceriotti, M. Fast and accurate uncertainty estimation in chemical machine learning. *Journal of chemical theory and computation* **2019**, *15*, 906–915.

(25) Gasteiger, J.; Giri, S.; Margraf, J. T.; Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115* **2020**,

(26) Behler, J. Constructing high-dimensional neural network potentials: a tutorial review. *International Journal of Quantum Chemistry* **2015**, *115*, 1032–1050.

(27) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is more: Sampling chemical space with active learning. *The Journal of chemical physics* **2018**, *148*.

(28) Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. international conference on machine learning. 2016; pp 1050–1059.

(29) Wen, M.; Tadmor, E. B. Uncertainty quantification in molecular simulations with dropout neural network potentials. *npj computational materials* **2020**, *6*, 124.

(30) Schran, C.; Thiemann, F. L.; Rowe, P.; Müller, E. A.; Marsalek, O.; Michaelides, A. Machine learning potentials for complex aqueous systems made simple. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2110077118.

(31) Vandermause, J.; Xie, Y.; Lim, J. S.; Owen, C. J.; Kozinsky, B. Active learning of reactive Bayesian force fields applied to heterogeneous catalysis dynamics of H/Pt. *Nature Communications* **2022**, *13*, 5183.

(32) Ash, J. T.; Zhang, C.; Krishnamurthy, A.; Langford, J.; Agarwal, A. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671* **2019**,

(33) Kirsch, A.; Van Amersfoort, J.; Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems* **2019**, *32*.

(34) Citovsky, G.; DeSalvo, G.; Gentile, C.; Karydas, L.; Rajagopalan, A.; Rostamizadeh, A.; Kumar, S. Batch active learning at scale. *Advances in Neural Information Processing Systems* **2021**, *34*, 11933–11944.

(35) Holzmüller, D.; Zaverkin, V.; Kästner, J.; Steinwart, I. A framework and benchmark for deep batch active learning for regression. *Journal of Machine Learning Research* **2023**, *24*, 1–81.

(36) Schütt, K.; Unke, O.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. International Conference on Machine Learning. 2021; pp 9377–9388.

(37) Mortensen, J. J.; Gjerding, M.; Thygesen, K. S. MyQueue: Task and workflow scheduling system. *Journal of Open Source Software* **2020**, *5*, 1844.

(38) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Science advances* **2017**, *3*, e1603015.

(39) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C., et al. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(40) Schiøtz, J. Asap. `https://gitlab.com/asap`, 2023.

(41) Kermode, J.; Pastewka, L. Matscipy. `https://github.com/libAtoms/matscipy`, 2019.

(42) Galvelis, R.; Eastman, P. NNPOps. `https://github.com/openmm/nnpops`, 2020.

(43) Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. **2017**,

(44) Knuth, F.; Carbogno, C.; Atalla, V.; Blum, V.; Scheffler, M. All-electron formalism for total energy strain derivatives and stress tensor components for numeric atom-centered orbitals. *Computer Physics Communications* **2015**, *190*, 33–50.

(45) Lee, J.; AlRegib, G. Gradients as a measure of uncertainty in neural networks. 2020 IEEE International Conference on Image Processing (ICIP). 2020; pp 2416–2420.

(46) Jacot, A.; Gabriel, F.; Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems* **2018**, *31*.

(47) Zaverkin, V.; Kästner, J. Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design. *Machine Learning: Science and Technology* **2021**, *2*, 035009.

(48) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Physical Chemistry Chemical Physics* **2016**, *18*, 13754–13769.

(49) Bishop, C. M.; Nasrabadi, N. M. *Pattern recognition and machine learning*; Springer, 2006; Vol. 4.

(50) Lee, K.; Lee, K.; Lee, H.; Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems* **2018**, *31*.

(51) Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **2008**, *9*.

(52) Zaverkin, V.; Holzmüller, D.; Steinwart, I.; Kästner, J. Exploring chemical and conformational spaces by batch mode deep active learning. *Digital Discovery* **2022**, *1*, 605–620.

(53) Pazouki, M.; Schaback, R. Bases for kernel-based spaces. *Journal of Computational and Applied Mathematics* **2011**, *236*, 575–588.

(54) Podryabinkin, E. V.; Tikhonov, E. V.; Shapeev, A. V.; Oganov, A. R. Accelerating crystal structure prediction by machine-learning interatomic potentials with active learning. *Physical Review B* **2019**, *99*, 064114.

(55) Lysogorskiy, Y.; Bochkarev, A.; Mrovec, M.; Drautz, R. Active learning strategies for atomic cluster expansion models. *Physical Review Materials* **2023**, *7*, 043801.

(56) Park, C. W.; Kornbluth, M.; Vandermause, J.; Wolverton, C.; Kozinsky, B.; Mailoa, J. P. Accurate and scalable multi-element graph neural network force field and molecular dynamics with direct force architecture. *arXiv preprint arXiv:2007.14444* **2020**,

(57) Breiman, L. Bagging predictors. *Machine learning* **1996**, *24*, 123–140.

(58) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* **2017**, *30*.

(59) Wollschläger, T.; Gao, N.; Charpentier, B.; Ketata, M. A.; Günnemann, S. Uncertainty Estimation for Molecules: Desiderata and Methods. **2023**,

(60) Zhu, A.; Batzner, S.; Musaelian, A.; Kozinsky, B. Fast uncertainty estimates in deep learning interatomic potentials. *The Journal of Chemical Physics* **2023**, *158*.

(61) Yadan, O. Hydra-a framework for elegantly configuring complex applications. *Github* **2019**, *2*, 5.

(62) Falcon, W. A. Pytorch lightning. *GitHub* **2019**, *3*.

Paper II

Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces

Xin Yang, Arghya Bhowmik, Tejs Vegge and Heine Anton Hansen

Chemical Science, 2023, **14**, 3913-3922

Chemical Science

ROYAL SOCIETY OF CHEMISTRY

EDGE ARTICLE

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Chem. Sci.*, 2023, **14**, 3913

All publication charges for this article have been paid for by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces†

Xin Yang, Arghya Bhowmik, Tejs Vegge and Heine Anton Hansen *

Received 5th December 2022
Accepted 9th March 2023

DOI: 10.1039/d2sc06696c

rsc.li/chemical-science

The application of *ab initio* molecular dynamics (AIMD) for the explicit modeling of reactions at solid–liquid interfaces in electrochemical energy conversion systems like batteries and fuel cells can provide new understandings towards reaction mechanisms. However, its prohibitive computational cost severely restricts the time- and length-scales of AIMD. Equivariant graph neural network (GNN) based accurate surrogate potentials can accelerate the speed of performing molecular dynamics after learning on representative structures in a data efficient manner. In this study, we combined uncertainty-aware GNN potentials and enhanced sampling to investigate the reactive process of the oxygen reduction reaction (ORR) at an Au(100)–water interface. By using a well-established active learning framework based on CUR matrix decomposition, we can evenly sample equilibrium structures from MD simulations and non-equilibrium reaction intermediates that are rarely visited during the reaction. The trained GNNs have shown exceptional performance in terms of force prediction accuracy, the ability to reproduce structural properties, and low uncertainties when performing MD and metadynamics simulations. Furthermore, the collective variables employed in this work enabled the automatic search of reaction pathways and provide a detailed understanding towards the ORR reaction mechanism on Au(100). Our simulations identified the associative reaction mechanism without the presence of *O and a low reaction barrier of 0.3 eV, which is in agreement with experimental findings. The methodology employed in this study can pave the way for modeling complex chemical reactions at electrochemical interfaces with an explicit solvent under ambient conditions.

1 Introduction

Over the past several decades, density functional theory (DFT) calculations have been extensively used for developing novel electrocatalysts towards the oxygen reduction reaction (ORR) by taking advantage of well-developed theoretical methods[1–5] (*e.g.*, free energy diagrams, volcano plots, and d-band theory) for predicting catalytic activities. Nevertheless, most of these calculations oversimplify the operating conditions of catalysts by either modelling liquid water at the electrolyte–electrode interface as static water layers,[6–9] implicitly representing them *via* dielectric continuum models,[10–12] or even absolutely ignoring the effect of solvents.[13–16] These limitations may lead to erroneous evaluation of activity trends of catalysts as compared to experiments, for example, the oxygen reduction reaction on gold in alkaline electrolytes.[17,18] Including solvent molecules for electrolyte–electrode interface simulations and investigating their dynamical effects could offer us a better understanding towards the reaction mechanisms of the ORR and may resolve the conflicts between theoretical calculations and experiments.

While *ab initio* molecular dynamics (AIMD) is capable of capturing the dynamics of liquid water, it is prohibitively expensive for large length-scale and long time-scale simulations. For instance, the time-averaged metrics (*e.g.*, energy and temperature) of AIMD simulations can differ significantly if started from different initial configurations, while these discrepancies could be greatly mitigated if the model system is equilibrated and sampled from long enough trajectories.[19–21] The prohibitive computational cost severely limits the equilibration and sampling time scales of AIMD to only a few ps, which may significantly impair the reliability of such studies.[19,22–28]

Recently, advances in machine learning have played great roles in aiding the design and discovery of transition metal based catalysts.[29,30] By learning from data, machine learning tools can make fast predictions to find target catalysts and

Department of Energy Conversion and Storage, Technical University of Denmark, Anker Engelunds Vej, 2800 Kgs, Lyngby, Denmark. E-mail: heih@dtu.dk

† Electronic supplementary information (ESI) available: ESI figures and tables as described in the text. A summary of reference structures, a summary of error metrics of neural network potentials, a comparison of model performance between previous studies and this work, adsorption energies of difference species on Au(100), average energy profiles and density profiles of 5 ns MD simulation for different interface structures, and a movie showing the bond breaking process extracted from the transition state area. See DOI: https://doi.org/10.1039/d2sc06696c

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

provide valuable insights into the nature of the reaction, which enable high-throughput screening of catalysts from a broad chemical space and automated catalyst design.[31–33] In particular, neural network potentials (NNPs) have shown great promise at fitting the potential energy surface (PES) of reactive model systems by training on reference configurations that well describe the representative atomic environments.[34–37] This approach could speed up MD simulations by several orders of magnitude whilst retaining the accuracy comparable to AIMD, which enables us to considerably extend the time scale and length scale of MD simulations without compromising accuracy. Initially proposed architectures of neural network potentials learned the force field by leveraging handcrafted features based on distance and angle information to capture the characteristics of local atomic environments.[38–40] Behler–Parrinello neural network potential is the first example in which the Cartesian coordinates of atoms are transformed to rotational and translational invariant atomic-centered symmetry functions.[38,39] Recent advances in graph neural networks (GNNs) for molecule graphs have made it possible to learn representative features from the atomic structure *via* a graph message-passing scheme.[41–45] State-of-the-art GNN models leverage the rotation equivariant representation of node features (*i.e.*, features of atomic environments) to provide more accurate force predictions, which can be essential in MD simulations.[44–46] In spite of numerous novel machine learning methods for fitting PES and MD simulations driven by NNPs,[20,21,47–50] there are few studies on simulating nonequilibrium dynamics and reactions. We have yet to find out any study performing sampling of rare events that govern chemical reactions with NNPs.[28,51,52] Taking the ORR as an example, although NNPs can significantly accelerate MD simulations, the time scale of reactive simulation of the ORR is still inaccessible, not to mention the complex ambient conditions of the catalysts. Due to the rapid development of enhanced sampling techniques like metadynamics[53,54] (MetaD), high accuracy sampling of PES has been possible for such rare events. We envision that combining enhanced sampling methods together with high-fidelity NNPs can enable full simulation of slow chemical reactions on an atomic scale within affordable computational cost.

In this paper, we present the full atomic simulation of the ORR at an Au(100)–water interface done using metadynamics simulations accelerated by equivariant graph neural network potentials.[43] The gold electrode has been extensively studied as an ORR electrocatalyst, while its exceptional activity, especially in alkaline media, is still not well-explained.[17,18,55,56] This case could well demonstrate the power of our proposed simulation paradigm towards modeling of rare chemical reactions at solid–liquid interfaces. Compared to non-reactive MD performed with NNPs, a major challenge of simulating rare events like the ORR is to ensure that the machine learning model encompasses a vast configurational space far away from equilibrium. This requires adaptive sampling of representative reference structures from MD and MetaD simulations, particularly transition states that are rarely visited. In addition, quantitatively evaluating the reliability of NNPs for describing the PES in the configurational space of interest is also indispensable. Here we adopt an active learning approach based on CUR matrix decomposition[57,58] to sample representative reference structures from MD and MetaD simulations. This method enables us to representatively sample the vast configurational spaces of the ORR at the solid–liquid interface with minimal human intervention and significantly reduced computational cost. Our MD and MetaD simulations are uncertainty aware, demonstrating robust and reliable modeling of full atomic simulation of the ORR with NNPs.

2 Computational details

2.1 Active learning framework

Our neural network potentials are constructed based on an active learning framework utilizing CUR decomposition based selective sampling as demonstrated in Fig. 1. First, an initial dataset was generated by selectively sampling reference structures from several AIMD trajectories of Au(100)–water interfaces. Multiple interface structures with different numbers of hydroxyl or oxygen molecules are considered to ensure the diversity and versatility of the training dataset and to further study the impact of adsorbates on the dynamics of solvents. The initial AIMD trajectories contain several hundreds of thousands of configurations. Using all of them would make the training of NNPs very slow. Many structures are similar, and thus models do not capture new correlations when all of those are used simultaneously. Therefore, it is crucial to sample only those structures that are representative and informative from these trajectories. We first select structures from AIMD trajectories one in every 50 MD steps, reducing the number of candidate configurations to several tens of thousands. Then the CUR matrix decomposition method[57,58] is employed to further refine the training dataset without losing too much information. Given an N × M data matrix X with its rows corresponding to N atoms and its columns corresponding to M fingerprints, the

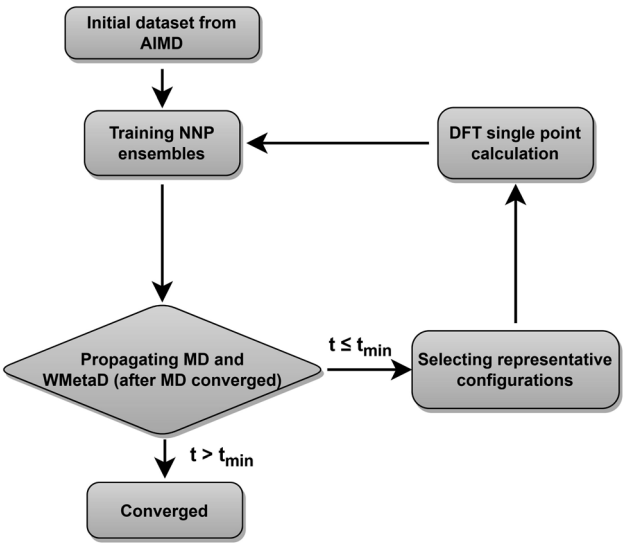

Fig. 1 Active learning procedures used to train the neural network potentials.

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

objective of CUR is to minimize the information loss after ruling out some rows and columns, while minimizing the number of rows and columns to be selected. We also add an extra term in the objective function to maximize the Euclidean distance between different atomic environments to ensure the diversity of sampled structures. In this process, the importance of each row and column in the data matrix can be evaluated, and the representative configurations and fingerprints can be jointly sampled. Here the fingerprints of atoms in candidate structures are described by Behler–Parrinello symmetry functions,[38,39] which have been extensively used for fitting the PES for solid–liquid interface systems.[20,21,47–50] CUR matrix decomposition also provides an efficient way for automatically selecting symmetry function parameters that are typically non-trivial.

An ensemble of neural network potentials (NNPs) was then trained on the initial dataset with 5000 reference structures after CUR selection. In order to extend their capability of exploring larger configurational space, the trained NNPs are updated adaptively in the following steps: (i) propagating MD trajectories with the trained NNP ensemble as an energy/force calculator; (ii) selecting representative reference structures from MD trajectories by using CUR decomposition; (iii) calculating these selected new data points with DFT; (iv) retraining NNPs with the expanded training dataset. Instead of propagating MD using only one NNP, we choose to combine all the trained NNPs together for the prediction of energy and forces. This strategy not only improves the predictive accuracy of our model but also provides a practical way to quantify if the model is still confident enough in the configurational space of interest. The quantification is achieved by evaluating the energy uncertainty and force uncertainty for every step *via* calculating the variance of NNPs during the MD simulation. Fig. S1† compares the calculated uncertainty and true prediction error, indicating that uncertainty is an excellent indicator for true model error. Based on the query-by-committee method, which has been widely used in active learning,[46,59,60] the configurations with relatively large uncertainty are collected to reduce the number of candidate configurations. Subsequently, the obtained structures are further sub-sampled by CUR decomposition for DFT evaluation. By adding these carefully chosen new data in the training set, we constantly improve the model prediction for new configurational space visited by MD simulations. Combining this strategy and CUR matrix decomposition significantly reduces the number of candidate structures and ensure the diversity of structures in a sampled batch. The simulations are stopped if the uncertainties are too large or too many structures with large uncertainties are collected. The iterative training of the NNP ensemble stops once all the MD simulations can be propagated to more than $t_{\min}$ steps, where $t_{\min}$ is selected as 5 ns to ensure that the model systems are properly equilibrated and all the dynamical events are fully captured.[21,47,48] To further investigate the ORR kinetics at the gold–water interface, the iterative training procedures are repeated in the case of MetaD simulations. Notably, as the transition states are rarely visited during MetaD runs, it is critical to include enough such configurations into our training dataset and validate our MetaD simulations *via* uncertainty quantification.

2.2 CUR matrix decomposition

Reference structures are adaptively sampled by CUR matrix decomposition[57,58] from MD simulations driven by the NNPs. CUR matrix decomposition is a low rank approximation to the input matrix, indicating that the information of the matrix can be maintained after discarding some columns and rows. Given an $n \times m$ data matrix X, our objective is to select the least number of rows and columns from X to construct a subset matrix $\tilde{X}$ while minimizing the information loss. To address this issue, Li *et al.* proposed the ALFS algorithm[58] to minimize the following objective function by using an augmented Lagrange multiplier:

$$\min_{W\in\mathbb{R}^{m\times n}} \mathcal{L} = \|X - XWX\|_F^2 + \alpha\|W\|_{2,1} + \beta\|W^T\|_{2,1} + \lambda\|T \odot (WX)\|_1 \quad (1)$$

where $W\in\mathbb{R}^{m\times n}$ is an auxiliary matrix that determines which rows and columns should be selected. Minimizing the l_2 norm of its rows and columns corresponds to minimizing the number of selected columns and rows, respectively. The weight matrix T that encodes the Euclidean distance between different rows is used to maximizing the distance between selected rows, which could effectively increase the diversity of datapoints selected in an active learning batch. The regularization parameters α, β, and γ are used to determine the priority to minimize the row numbers, column numbers or row distance, respectively. The l_2 norm of rows and columns of optimized W will be regarded as the importance score of each column and row in the data matrix, and the importance score of a configuration will be calculated as the average of the importance score of atoms inside it.

One thing that should be noted is that the overall data matrix obtained from an MD run can typically contain a few million to several billion entries with a feature size of more than one hundred, on which the implementation of CUR decomposition can be intractable. Instead of using the data matrix as a whole, we will split the large data matrix into smaller ones by rows, and then assess the importance of every row and column *via* CUR decomposition of the smaller matrices on-the-fly. This strategy significantly improves the efficiency and computational cost of CUR selection while has minor impact on the performance of CUR. The size of divided matrices is selected as 1000 entries and 90 features for each element generated by Behler–Parrinello symmetry functions. The parameters of symmetry function are also selected by CUR decomposition from a pool of 3000 symmetry functions.

2.3 AIMD and DFT single point calculations

The Au(100)–water interface is modelled as 30 H_2O molecules on top of a (3×3) tetragonal Au(100) surface with four atomic layers, which will be denoted as Au(100)–30H_2O hereafter. A vacuum layer larger than 15 Å is perpendicularly added into the model to eliminate the spurious interaction between periodic images. In order to simulate the interface with ORR intermediates, we also consider structures with one and two hydroxyls by removing the hydrogen atoms from water molecules near the slab, and a structure with one oxygen molecule on top of an Au(100) slab. These structures are denoted as Au(100)–1OH/29H_2O, Au(100)–2OH/28H_2O, and Au(100)–1O_2/30H_2O,

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

respectively. Constant temperature MD simulations are then performed in VASP[61–64] by using these initial configurations with a timestep of 0.5 fs and the temperature is kept at around 350 K with a Nosé–Hoover thermostat.[65] The bottom two layers are kept fixed during the MD run for all model systems. 50 ps, 15 ps, 15 ps, and 15 ps MD simulations are conducted for Au(100)–$30H_2O$, Au(100)–$1OH/29H_2O$, Au(100)–$2OH/28H_2O$, and Au(100)–$1O_2/30H_2O$, respectively. The reason for running shorter MD simulations on the model systems with adsorbates is that their most local structures are similar to the Au(100)–$30H_2O$ system. Density functional calculations are used to calculate the potential energy and the forces for propagating AIMD and labeling representative configurations sampled by active learning. We employ an energy cutoff of 350 eV for plane-wave basis expansion and a $2 \times 2 \times 1$ Monkhorst–Pack k-grid for Brillouin zone sampling.[66] The exchange-correlation effects are approximated by using the PBE functional combined with D3 van der Waals correction.[67,68]

2.4 Production molecular dynamics simulations

The production MD simulations driven by the NNP ensemble have been performed using the MD engine of the Atomic Simulation Environment (ASE) python library.[69] The simulation box in AIMD is too small to accommodate more adsorbates and to simulate the full reaction. Furthermore, previous studies also demonstrated that notable noise in the structural properties of the model systems could be observed when using small cell sizes.[47,70] Considering both effects and the increased computational cost for MD and labelling, we constructed a larger model with 59 H_2O molecules on top of a (4×4) tetragonal Au(100) surface with four atomic layers, on which more adsorbates can be accommodated. With the presence of one to six *OH, the corresponding hydrogen atoms are removed at the interface, producing interface structures that could be denoted as Au(100)–$1OH/58H_2O$, Au(100)–$2OH/57H_2O$, Au(100)-$3OH/56H_2O$, Au(100)–$4OH/55H_2O$, Au(100)–$5OH/54H_2O$, and Au(100)–$6OH/53H_2O$, respectively. In order to investigate the kinetics of the ORR, the initial state structure Au(100)–$1O_2/57H_2O$ is also built by removing two H_2O molecules and placing a O_2 molecule on top of Au(100). The momentum of model systems is initiated by a Maxwell–Boltzmann distribution with the temperature set to 350 K. The MD simulations are propagated for 5 ns by Langevin dynamics with a target temperature of 350 K, a timestep of 0.25 fs, and a friction coefficient of 0.02. It is noteworthy that a smaller time step is selected for production as it can help the MD simulations reach a longer time scale with smaller uncertainty. The uncertainties of frames in MD simulations are quantified as the variance and standard deviation (SD) of model outputs:

$$E_{\mathrm{var}} = \frac{1}{N}\sum_{i=1}^{N}\left(E_i - \overline{E}_i\right)^2 \tag{2}$$

$$F_{\mathrm{var}} = \frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}\left(F_i^{jk} - \widehat{F}_i^{jk}\right)^2 \tag{3}$$

$$F_{\mathrm{sd}} = \frac{1}{3M}\sum_{j=1}^{M}\sum_{k=1}^{3}\sqrt{\sum_{i=1}^{N}\left(F_i^{jk} - \widehat{F}_i^{jk}\right)^2} \tag{4}$$

where N is the number of models in the ensemble, M is the number of atoms in a frame, and *E&x0304;* and *F&x0304;* are the average predicted energy and force, respectively. In order to ensure the reliability of MD results, the simulations will stop if F_{sd} is larger than 0.5 eV $Å^{-1}$ or more than 2000 structures with F_{sd} larger than 0.05 ev $Å^{-1}$ are collected.

Following the method in ref. [19], we calculated the formation energy of *OH as the internal energy of the Au(100)–n_{OH}OH/(59-n_{OH})H_2O interface structure, plus the internal energy of gas phase $n_{\mathrm{OH}}/2H_2$ molecules, minus the internal energy of the Au(100)–$59H_2O$ interface structure.

$$\begin{aligned}\Delta E = {} & \left\langle E_{\mathrm{Au(100)}-n_{\mathrm{OH}}\mathrm{OH}/(59-n_{\mathrm{OH}})\mathrm{H_2O}}\right\rangle_t + \frac{n_{\mathrm{OH}}}{2}\left(E_{\mathrm{H_2}} + \frac{3}{2}k_{\mathrm{B}}T\right) \\ & - \left\langle E_{\mathrm{Au(100)}-59\mathrm{H_2O}}\right\rangle_t\end{aligned} \tag{5}$$

The internal energy of interface structures is calculated as the time averaged potential energy plus kinetic energy. And the internal energy of H_2 gas molecules is calculated as the potential energy plus $3/2k_{\mathrm{B}}T$ because their center-of-mass motions are not included in the MD simulations. Likewise, the adsorption energy of O_2 is calculated as follows.

$$\begin{aligned}E_{\mathrm{ads}} = {} & \left\langle E_{\mathrm{Au(100)}-1\mathrm{O_2}/57\mathrm{H_2O}}\right\rangle_t + \frac{1}{2}\left(E_{\mathrm{O_2}} + \frac{3}{2}k_{\mathrm{B}}T\right) \\ & - \left\langle E_{\mathrm{Au(100)}-57\mathrm{H_2O}}\right\rangle_t\end{aligned} \tag{6}$$

2.5 Metadynamics simulations

In this study, all the enhanced sampling simulations are performed with a well-tempered version of metadynamics.[71] The production metadynamics simulations are propagated by Langevin dynamics for 2.5 ns in ASE. The calculation of collective variables and bias potential of metadynamics is achieved by using PLUMED[72–74] which is interfaced to the ASE library. To construct the path CVs as described in the main text, the Au(100)–$1O_2/57H_2O$ and Au(100)–$4OH/55H_2O$ interface structures are selected as two reference structures. And the coordination numbers (C_{O_2}–O) and (C_{O_2}–H) are used to define the configurational space of the path. The corresponding equations and parameters for calculating (C_{O_2}–O) and (C_{O_2}–H) are shown in Table S1†

With the defined path, the progress along the path s and the distance from the path z can be computed as:

$$s = \frac{\sum_{i=1}^{N} i e^{-\lambda\|X-X_i\|^2}}{\sum_{i=1}^{N} e^{-\lambda\|X-X_i\|^2}} \tag{7}$$

$$z = -\frac{1}{\lambda}\ln\left[\sum_{i=1}^{N} e^{-\lambda\|X-X_i\|^2}\right] \tag{8}$$

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

where N is the number of reference structures and X is the structure described by (C_{O_2}–O) and (C_{O_2}–H). The parameter λ is selected as 0.25. The Gaussians adopted have an initial height of 0.1 eV and a width of 0.05 and 0.1 for s and z collective variables, respectively. The metadynamics are carried out at 350 K, employing a bias factor of 5 and a deposition rate of 125 fs (every 500 steps). For every metadynamics run (except for O_2 migration as O_2 is metastable in bulk water), the system is first equilibrated for 0.5 ns.

2.6 Training neural network potentials

The NNP ensemble we used for production consists of five neural network potentials with different architectures of the polarizable atom interaction neural network (PaiNN) model.[43] In this model, all the atoms in a given configuration are treated as nodes in a graph and the information of their connections will be collected and processed by a message function, which will then be passed to an update function for updating node features. After several message passing iterations, the node features will be used as the input of a multilayer perceptron to get its atomic energy or other scalar properties. By summing up the atomic energies of a given structure, we can get its potential energy and forces by calculating the negative derivatives of energy to atomic coordinates. The model can automatically learn the relationship between chemical properties and the positions of atoms by optimizing several hundreds of thousands of model parameters in message and update layers. In contrast, only a few hyperparameters need to be selected (the size of node features, the number of message passing layers, loss ratio of energy and forces, the cutoff radius for collecting distance information of atoms, *etc.*), avoiding the need to manually select and test handcrafted features like Behler–Parrinello symmetry functions.[38,39] Besides, the model uses both scalar and vector node features to realize rotational equivariance of directional information (*e.g.*, forces) in the graph, providing better prediction of forces.

Table S2† reports the architectures of five models constituting our NNP ensemble and their error metrics after training on the same dataset for up to 1 000 000 steps. These models use different node feature sizes and the number of message-passing layers to induce model diversity, while their cutoff radii are all set to 5 Å. Both the model training and subsequent production MD (MetaD) simulations are conducted on an NVIDIA GeForce RTX 3090 GPU with float32 precision. The weight parameters in these models are randomly initialized and then optimized on the same data split using stochastic gradient descent to minimize the mean square error (MSE) loss, which can be expressed as:

$$\mathscr{L} = \frac{1-\lambda}{N}\sum_{i=1}^{N}\left(E_i - \hat{E}_i\right)^2 + \frac{1-\lambda}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}\left(F_i^{jk} - \hat{F}_i^{jk}\right)^2 \quad (9)$$

where N is the number of configurations, M is the number of atoms in a configuration, and λ is the force weight that controls the relative importance between energy and force loss. Here the force weight is set to 0.99 as our tests show that using a relatively large force weight can well improve the force prediction while only slightly undermines the precision of energy prediction. Our model parameters are trained by the Adam optimizer[75] as implemented in PyTorch[76] with an initial learning rate of 0.0001, the default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a batch size of 16. An exponential decay learning rate scheduler with a coefficient of 0.96 is used to adjust the learning rate for every 100 000 learning steps. The dataset is split into a training set (90%) and a validation set (10%), where the validation set is used for early stopping when the error of forces is small enough. Note that several different error metrics are used to evaluate the performance of the trained model, including mean absolute error (MAE) and root mean squared error (RMSE) for both energy and force predictions. These error metrics can be expressed as follows:

$$E_{\mathrm{MAE}} = \frac{1}{N}\sum_{i=1}^{N}\left|E_i - \hat{E}_i\right| \quad (10)$$

$$E_{\mathrm{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(E_i - \hat{E}_i\right)^2} \quad (11)$$

$$F_{\mathrm{MAE}} = \frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}\left|F_i^{jk} - \hat{F}_i^{jk}\right| \quad (12)$$

$$F_{\mathrm{RMSE}} = \sqrt{\frac{1}{3NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{3}\left(F_i^{jk} - \hat{F}_i^{jk}\right)^2} \quad (13)$$

3 Results and discussion

3.1 Validation of models

Following the active learning framework, we have obtained a final dataset with 18 731 configurations. Fig. S2† shows the learning curves of our NNPs trained on the final dataset, and Table S2† reports the detailed error metrics of best models on the validation set. It is remarkable that our NNPs exhibit exceptional accuracy towards the prediction of energy and forces, where the mean absolute errors (MAEs) of energy range between 0.4 and 0.8 meV per atom, and the MAEs of forces between 12.6 and 16.3 meV Å^{-1}. To illustrate the performance of our models on different interface structures, we also report the composition of the final dataset and corresponding error metrics for different structures as shown in Table S3.† The precision of force predictions for each species in our research system is also shown in Fig. 2a, indicating close numerical agreement with DFT results. All these results suggested that the trained NNPs can provide accurate energy and force predictions for different structures across the ORR configurational space in the production MD simulations. Table S4† exhibits the comparison of model performance in terms of energy and force predictions between our model and other studies for complex systems, illustrating that our model outperforms most of these studies, especially force predictions.[43,44,77–80] The role of accurate

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

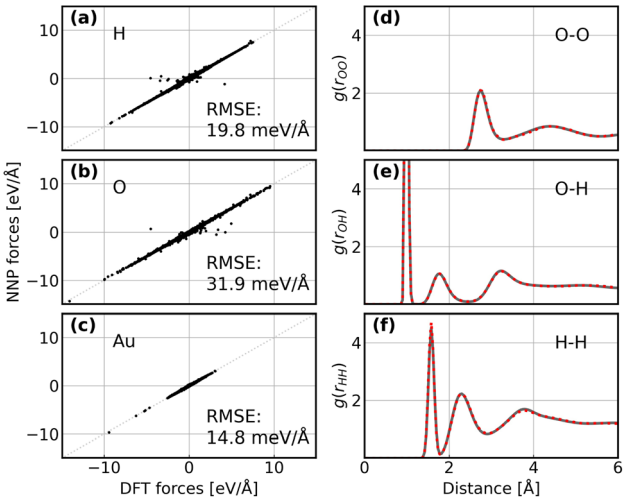

Fig. 2 (a–c) Comparison between forces derived from DFT calculations and NNP predicted forces for H, O, and Au. The RMSE of forces for each element are denoted inside. (d–f) Comparison between RDFs obtained from AIMD simulations and NNP MD simulations on the Au(100)–water interface structure. The red points denote RDFs generated by AIMD calculation and the grey solid line denotes RDFs generated by NNP calculation.

force prediction is emphasized in our study since it is critical in MD simulations. The performance of the trained NNP ensemble is further validated in terms of its ability to reproduce the structural properties of AIMD trajectories. Fig. 2b shows the match of the radial distribution functions (RDFs) of all involved species in the case of the Au(100)–water interface (without hydroxyls or oxygen molecules). Apparently, the RDFs generated by NNP MD simulations (solid black line) exhibit excellent agreement with AIMD results (red points), indicating that the NNP ensemble captures the structural arrangement of the gold–water interface well. Apart from validating NNPs with the existing dataset, a more important assessment for the quality of NNPs is their application domain, which can be confirmed by uncertainty measurements. Concretely, the MD runs should be ergodic to ensure the reliability of information derived from them, which indicates that all energetically relevant states must be sampled and within the manifold accessible by NNPs. For all MD simulations in this study, we not only sample the properties of interest along long-time scale MD simulations but also present the uncertainties of all steps by calculating the variance of NNPs. The low force uncertainty of MD simulations for different interface structures verifies the robustness and reliability of the trained NNP ensemble in the given configurational space (the energy and uncertainty profiles in Fig. S3 to S10†). The agreement of the density profiles of water between AIMD and NNP MD with the same box size (3×3) is reported in Fig. S11.† It can be observed that the density profile of water in NNP MD simulation is more smooth than that in AIMD, and some disagreements are exhibited in the bulk water area. We ascribe the disagreements and the fluctuation of AIMD density profiles to the inadequate equilibration of AIMD simulations. Moreover, the average energy profiles of Au(100)–water with four *OH that started from different points are well converged as shown in Fig. S12,† indicating that our MD simulations are ergodic and the time scale is long enough.

Except for the validation of model accuracy and reliability, we also evaluated the overall computational efficiency of the proposed scheme in terms of training the initial model, model retraining, CUR matrix decomposition, production MD simulations for 5 ns and DFT labelling as demonstrated in Fig. S13.† For the systems in this study, the required computational time for 1000 AIMD steps is approximately 650 CPU hours, corresponding to 1543.8 hours in total for generating the initial AIMD dataset if using 80 CPU cores for each job. Training on the initial dataset takes about 40 hours, while the cost of retraining the new models can be substantially reduced by loading pre-trained model parameters. The MD simulation driven by NNPs accounts for the highest computational cost in an active learning iteration, which takes approximately 7 days to run 5 ns simulations on an NVIDIA RTX3090 GPU. In comparison, AIMD needs more than 7 years to run 5 ns using 80 CPU cores, being about 3–400 times slower than NNP MD. To train the NNPs for a system to run more than 5 ns MD, 5 to 10 iterations are usually needed, which corresponds to 1000–2000 labelled structures as indicated in Table S3.† It is worth noting that the ASE MD engine used in this study is not specialized for GPU computing, resulting in high overheads of data transfer between the GPU and CPU. It can be expected that the computational efficiency of NNP MD can be further improved in the future by using a GPU-specialized MD code.

The validation of NNPs *via* application in MetaD simulations is crucial as the configurational space of the full reactive process can be huge while the transitional states are rarely visited. As shown in Fig. 3, our training data points are evenly distributed in the configurational space described by path collective variables,[81] and the force uncertainties along 2.5 ns MetaD simulations are all considerably small (all smaller than 0.05 eV $Å^{-1}$). Both metrics build confidence that the trained NNPs are reliable to capture the characteristics of all energetically relevant states, especially transitional states, of the ORR. Furthermore, the trained models have shown excellent transferability when using them for the inference of Au(110)–water and Au(111)–water interfacial systems as demonstrated in Fig. S1a.† Despite missing structural information for the two similar systems, the trained models still well predicts the energy and forces with both low errors and uncertainties for all Au(110)–water and Au(111)–water interfacial structures, which indicates that the proposed scheme and trained models can easily generalize to systems across a wide range of metals and their different facets.

3.2 Full metadynamics simulation of the ORR

After systematic validations, the trained NNPs are used to study both adsorption energetics and kinetics of the ORR at the gold–water interface. It is well known that the ORR on Au(100) in alkaline electrolytes proceeds *via* the complete four-electron transfer mechanism, while the partial two-electron transfer mechanism dominates on other Au facets, such as Au(111) and Au(110).[17,55,56] Despite the use of new techniques and persistent

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

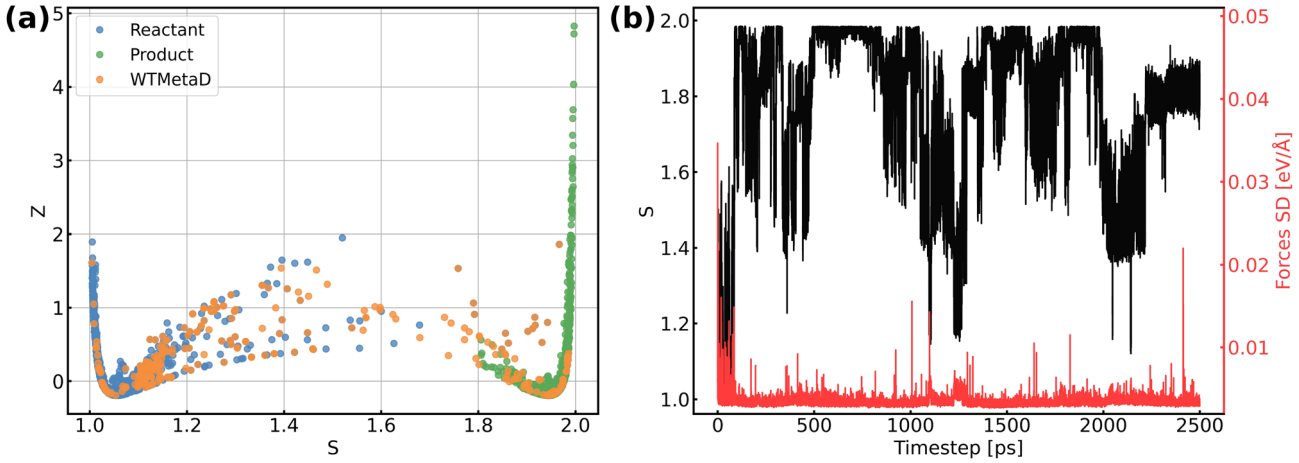

Fig. 3 (a) Distribution of reference structures in the configurational space described by path collective variables *s* and *z*, where *s* represents the progress along the path between reactants and products and *z* represents the distance from the path. (b) Evolution of *s* and force standard deviation (SD) along 2.5 ns MetaD simulation.

efforts devoted by researchers, the reason why ORR activity is exceptional and facet-dependent on gold remains elusive. There are several assumptions that may provide a clear answer to this question, including the outer-sphere mechanism of the ORR,[17,82,83] and the role of preadsorbed species and solvents.[84,85] All these assumptions call for a full atomic simulation that elaborately considers the ambient conditions of Au(100) and models the reaction without any simplification.

The first step of the ORR on Au(100) is O_2 activation, which is also considered a key step that determines the activity of catalysts that weakly interact with adsorbates. According to whether O_2 closely adsorbs on Au(100), the reaction can be initiated *via* the inner-sphere mechanism in which the slab directly transfers electrons to closely adsorbed O_2, or the outer-sphere mechanism in which the ORR occurs away from the slab by several solvent layers. The adsorption energy of the $*O_2$ molecule and *OH with different coverage is summarized in Table S5,† suggesting weak interaction between these species and the Au(100) slab. As demonstrated in Fig. S9,† our 5 ns MD simulations at the Au(100)–water interface with one O_2 molecule have shown that the O_2 molecule will be in close contact with the Au(100) surface, yielding a density peak at 2.1 Å. We further carried out a MetaD simulation that models the migration of O_2 from bulk water to the Au(100) surface as shown in Fig. 4a. It is found that there are no stable local minima for O_2 saturating in bulk water, and the migration barrier can be easily overcome by the thermal fluctuation of the model system. As shown in Fig. S14,† a simple MD simulation modeling the movements of O_2 in the bulk water part of the interface also proves this conclusion. After 350 ps simulations, the O_2 molecule finally moved from bulk water to the Au(100) surface. Based on these results, we model the reaction process with O_2 directly adsorbed on Au(100) and believed that the bond breaking of the O_2 molecule could be the rate-determining step of the ORR on Au(100).

The full atomic simulation of the ORR is then conducted to investigate the bond-breaking process in the O_2 molecule and the formation of hydroxyls by using metadynamics simulations. The reaction coordinates of the ORR are described by path collective variables (CVs)[81] with the initial state (Au(100)–$1O_2$/$57H_2O$) and final state (Au(100)–$4OH$/$55H_2O$) selected as two reference structures. The distance to reference structures is quantified by the number of oxygen atoms (C_{O_2-O}) and hydrogen atoms (C_{O_2-H}) around the O_2 molecule. As summarized in Table S6,† these two descriptors can well capture and differentiate the structural characteristics of different possible intermediate states of the ORR, including $*O_2$, *OOH, $*H_2O_2$, *O, and *OH. The well-designed CVs enable us to automatically search the reaction path without using any prior knowledge about the reaction mechanism. In this approach, instead of modeling multiple possible reaction pathways and verifying which one is energetically most favorable, we only need to incrementally extend the explored PES (with our NNPs) from equilibrium states to non-equilibrium transitional states by using active learning. Furthermore, this strategy can be easily generalized to simulate more complex model systems and chemical reactions.

Fig. 4b shows the obtained free energy landscape of the ORR as a function of path CVs, where *s* is the progress along the reference path, and *z* is the distance to the reference path. The landscape is composed of two basins which correspond to the initial state and final state of the ORR. Fig. 3b also shows the time evolution of the *s* collective variable. It can be seen that the first basin in the landscape has been completely filled after approximately 100 ps, which corresponds to the transition from O_2 to hydroxyls. Filling the second basin, which can be regarded as the transition from hydroxyls to O_2, becomes much more difficult than the first one with the employed CVs in this study. However, it should be pointed out that the depth of the first basin is enough to evaluate the activation energy of bond breaking in O_2. The energy barrier of the transition from O_2 to hydroxyls is estimated to be 0.3 eV, which is in good agreement with experimental findings that Au(100) displays high ORR activity. It is noteworthy that the simulation box in this study is small in comparison with the realistic interface structure. The

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM.
This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

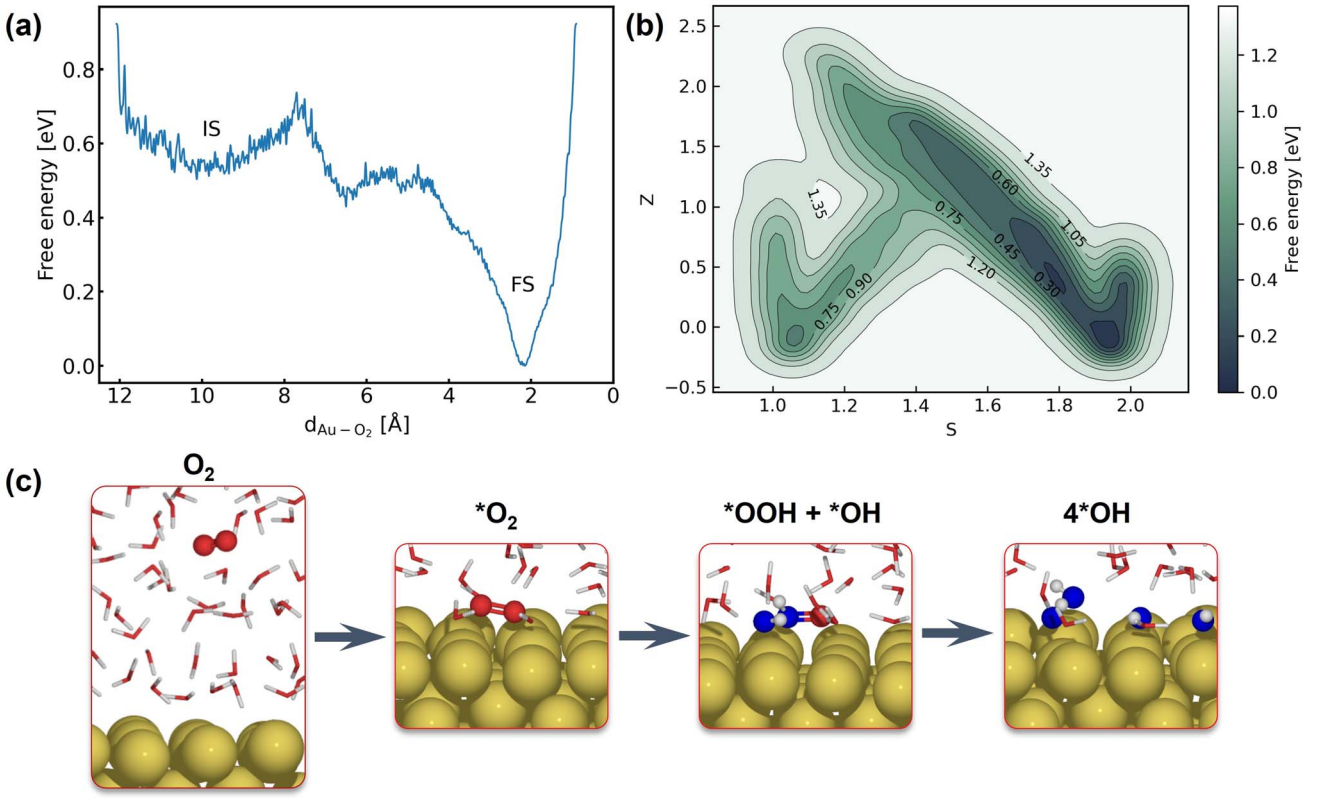

Fig. 4 (a) Free energy landscape of O_2 migration from bulk water to the Au(100) surface. (b) Free energy landscape of $*O_2$ reduction to *OH described by path collective variables. (c) Snapshots for O_2 in bulk water, the initial state, the transitional state, and the final state.

limited cell size can result in slightly higher formation energies of hydroxyl as demonstrated in Table S5,† which can be ascribed to the stronger repulsion between hydroxyl in smaller boxes and the possible lateral correlation of solvation shells. Besides, we also expect that the bond breaking of the O_2 molecule can be more difficult because of the easier recombination of individual oxygen atoms. Both effects can make the ORR in a small cell less facile, while further supporting our conclusion that the ORR is facile on Au(100) even when modeled with a limited number of water molecules. The snapshots for O_2 in bulk water, the initial state, the transition state, and the final state are displayed in Fig. 4c. At first, the O_2 molecule is partially protonated by neighboring water molecules to *OOH, suggesting the associative reaction pathway proposed by Nørskov *et al.*[1] However, the subsequent formation of *O is not observed in the overall reaction as the remaining oxygen atom is immediately protonated by reacting with water. Therefore, the reaction pathway observed from our simulations can be summarized as follows:

$$*O_2 + H_2O \rightarrow *OOH + *OH$$

$$*OOH + H_2O \rightarrow 3*OH$$

The MetaD simulation highlights the role of water molecules as a reactant of the ORR, suggesting that the explicit modeling of solvents is indispensable in theoretical electrocatalysis.

4 Conclusions

In summary, the reactive process of the ORR is investigated by MetaD simulations that are significantly accelerated by high fidelity NNPs in this study. By using an active learning strategy underpinned by CUR matrix decomposition, we obtained an NNP ensemble that exhibits exceptional performance and reliability for the prediction of structural properties and forces in the configurational space of an Au(100)–water interface. By leveraging well-designed path collective variables, the ORR can be fully and automatically simulated without the need to elaborately consider multiple reaction pathways. Our MetaD simulations suggest that the ORR proceeds in the associative reaction pathway, while the *OOH reaction intermediate is directly reduced to two *OH with the participation of neighboring water molecules rather than dissociating into *OH and *O. The low energy barrier of the ORR predicted in this study well explains the outstanding experimental ORR activity. The longer time-scale simulations enabled by NNPs can give us deeper insight into the nature of chemical reactions, such as the facet-dependent ORR on different Au facets which will be pursued in our future work. Besides, the effect of cations on the ORR activity of gold is also a meaningful extension of this work. The full atomic simulation conducted here can be conveniently extended to other model systems and become a valuable tool for investigating complex chemical reactions in a straightforward manner.

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

Data availability

The code for training the PaiNN model, performing MD and MetaD simulations, and CUR matrix decomposition is available in the following GitHub repository: **https://github.com/Yangxinsix/painn-sli**. The dataset in this study is openly available in the DTU data repository.[86]

Author contributions

X. Y. wrote the code, ran the calculations, analyzed the results, and wrote the original draft. X. Y., A. B., and H. A. H. conceived the research. A. B. and T. V. and H. A. H. supervised the research and helped revise the manuscript. All authors discussed and commented on the manuscript.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was supported by the Carlsberg Foundation through the Carlsberg Foundation Young Researcher Fellowship (Grant No. CF19-0304). The authors also acknowledge the computational resources provided by the Niflheim Linux supercomputer cluster installed at the Department of Physics at the Technical University of Denmark.

Notes and references

1 J. K. Nørskov, J. Rossmeisl, A. Logadottir, L. Lindqvist, J. R. Kitchin, T. Bligaard and H. Jonsson, *J. Phys. Chem. B*, 2004, **108**, 17886–17892.
2 J. Greeley, I. Stephens, A. Bondarenko, T. P. Johansson, H. A. Hansen, T. Jaramillo, J. Rossmeisl, I. Chorkendorff and J. K. Nørskov, *Nat. Chem.*, 2009, **1**, 552–556.
3 V. Viswanathan, H. A. Hansen, J. Rossmeisl and J. K. Nørskov, *ACS Catal.*, 2012, **2**, 1654–1660.
4 Z. W. Seh, J. Kibsgaard, C. F. Dickens, I. Chorkendorff, J. K. Nørskov and T. F. Jaramillo, *Science*, 2017, **355**, eaad4998.
5 A. Kulkarni, S. Siahrostami, A. Patel and J. K. Nørskov, *Chem. Rev.*, 2018, **118**, 2302–2312.
6 J. Rossmeisl, J. K. Nørskov, C. D. Taylor, M. J. Janik and M. Neurock, *J. Phys. Chem. B*, 2006, **110**, 21833–21839.
7 E. Skúlason, G. S. Karlberg, J. Rossmeisl, T. Bligaard, J. Greeley, H. Jónsson and J. K. Nørskov, *Phys. Chem. Chem. Phys.*, 2007, **9**, 3241–3250.
8 V. Tripkovic and T. Vegge, *J. Phys. Chem. C*, 2017, **121**, 26785–26793.
9 H. A. Hansen, J. Rossmeisl and J. K. Nørskov, *Phys. Chem. Chem. Phys.*, 2008, **10**, 3722–3730.
10 Y. Sha, T. H. Yu, Y. Liu, B. V. Merinov and W. A. Goddard III, *J. Phys. Chem. Lett.*, 2010, **1**, 856–861.
11 Y. Sha, T. H. Yu, B. V. Merinov, P. Shirvanian and W. A. Goddard III, *J. Phys. Chem. Lett.*, 2011, **2**, 572–576.
12 A. Fortunelli, W. A. Goddard, Y. Sha, T. H. Yu, L. Sementa, G. Barcaro and O. Andreussi, *Angew. Chem.*, 2014, **126**, 6787–6790.
13 J. Greeley and J. K. Nørskov, *J. Phys. Chem. C*, 2009, **113**, 4932–4939.
14 F. Calle-Vallejo, J. I. Martínez and J. Rossmeisl, *Phys. Chem. Chem. Phys.*, 2011, **13**, 15639–15643.
15 V. Tripković, I. Cerri, T. Bligaard and J. Rossmeisl, *Catal. Lett.*, 2014, **144**, 380–388.
16 X. Zhu, J. Yan, M. Gu, T. Liu, Y. Dai, Y. Gu and Y. Li, *J. Phys. Chem. Lett.*, 2019, **10**, 7760–7766.
17 A. Ignaczak, E. Santos and W. Schmickler, *Curr. Opin. Electrochem.*, 2019, **14**, 180–185.
18 Z. Duan and G. Henkelman, *ACS Catal.*, 2019, **9**, 5567–5573.
19 H. H. Kristoffersen, T. Vegge and H. A. Hansen, *Chem. Sci.*, 2018, **9**, 6912–6921.
20 A. E. Mikkelsen, H. H. Kristoffersen, J. Schiøtz, T. Vegge, H. A. Hansen and K. W. Jacobsen, *Phys. Chem. Chem. Phys.*, 2022, **24**, 9885–9890.
21 V. Quaranta, M. Hellstrom and J. Behler, *J. Phys. Chem. Lett.*, 2017, **8**, 1476–1483.
22 T. Sheng and S.-G. Sun, *Chem. Phys. Lett.*, 2017, **688**, 37–42.
23 S. Sakong and A. Groß, *Phys. Chem. Chem. Phys.*, 2020, **22**, 10431–10437.
24 T. Cheng, H. Xiao and W. A. Goddard III, *J. Am. Chem. Soc.*, 2016, **138**, 13802–13805.
25 J. A. Herron, Y. Morikawa and M. Mavrikakis, *Proc. Natl. Acad. Sci.*, 2016, **113**, E4937–E4945.
26 X. Qin, T. Vegge and H. A. Hansen, *J. Chem. Phys.*, 2021, **155**, 134703.
27 T. Ikeshoji and M. Otani, *Phys. Chem. Chem. Phys.*, 2017, **19**, 4447–4453.
28 T. Cheng, W. A. Goddard, Q. An, H. Xiao, B. Merinov and S. Morozov, *Phys. Chem. Chem. Phys.*, 2017, **19**, 2666–2673.
29 J. R. Kitchin, *Nat. Catal.*, 2018, **1**, 230–232.
30 O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schutt, A. Tkatchenko and K.-R. Muller, *Chem. Rev.*, 2021, **121**, 10142–10186.
31 P. Friederich, G. dos Passos Gomes, R. De Bin, A. Aspuru-Guzik and D. Balcells, *Chem. Sci.*, 2020, **11**, 4584–4601.
32 B. Meyer, B. Sawatlon, S. Heinen, O. A. Von Lilienfeld and C. Corminboeuf, *Chem. Sci.*, 2018, **9**, 7069–7077.
33 M. Foscato and V. R. Jensen, *ACS Catal.*, 2020, **10**, 2354–2377.
34 J. Xu, X.-M. Cao and P. Hu, *J. Chem. Theory Comput.*, 2021, **17**, 4465–4476.
35 N. Bernstein, G. Csányi and V. L. Deringer, *npj Comput. Mater.*, 2019, **5**, 1–9.
36 S. Stocker, G. Csányi, K. Reuter and J. T. Margraf, *Nat. Commun.*, 2020, **11**, 1–11.
37 M. Schreiner, A. Bhowmik, T. Vegge, P. B. Jørgensen and O. Winther, *Mach. Learn. Sci. Technol.*, 2022, **3**, 045022.
38 J. Behler and M. Parrinello, *Phys. Rev. Lett.*, 2007, **98**, 146401.
39 J. Behler, *J. Chem. Phys.*, 2011, **134**, 074106.
40 A. P. Bartók, R. Kondor and G. Csányi, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 2013, **87**, 184115.
41 K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, *J. Chem. Phys.*, 2018, **148**, 241722.

© 2023 The Author(s). Published by the Royal Society of Chemistry

Open Access Article. Published on 13 March 2023. Downloaded on 9/21/2023 8:56:06 PM. This article is licensed under a Creative Commons Attribution 3.0 Unported Licence.

42 J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, *International conference on machine learning*, 2017, pp. 1263–1272.
43 K. Schütt, O. Unke and M. Gastegger, *International Conference on Machine Learning*, 2021, pp. 9377–9388.
44 S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt and B. Kozinsky, *Nat. Commun.*, 2022, **13**, 1–11.
45 V. G. Satorras, E. Hoogeboom and M. Welling, *International conference on machine learning*, 2021, pp. 9323–9332.
46 C. Schran, F. L. Thiemann, P. Rowe, E. A. Müller, O. Marsalek and A. Michaelides, *Proc. Natl. Acad. Sci.*, 2021, **118**.
47 S. K. Natarajan and J. Behler, *Phys. Chem. Chem. Phys.*, 2016, **18**, 28704–28725.
48 V. Quaranta, J. Behler and M. Hellstrom, *J. Phys. Chem. C*, 2018, **123**, 1293–1304.
49 S. Kondati Natarajan and J. Behler, *J. Phys. Chem. C*, 2017, **121**, 4368–4383.
50 H. Ghorbanfekr, J. Behler and F. M. Peeters, *J. Phys. Chem. Lett.*, 2020, **11**, 7363–7370.
51 M. Yang, L. Bonati, D. Polino and M. Parrinello, *Catal. Today*, 2022, **387**, 143–149.
52 A. Urakawa, M. Iannuzzi, J. Hutter and A. Baiker, *Chem. - Eur. J.*, 2007, **13**, 6828–6840.
53 A. Laio and M. Parrinello, *Proc. Natl. Acad. Sci.*, 2002, **99**, 12562–12566.
54 A. Laio and F. L. Gervasio, *Rep. Prog. Phys.*, 2008, **71**, 126601.
55 P. Rodriguez and M. T. Koper, *Phys. Chem. Chem. Phys.*, 2014, **16**, 13583–13594.
56 P. Quaino, N. Luque, R. Nazmutdinov, E. Santos and W. Schmickler, *Angew. Chem., Int. Ed.*, 2012, **51**, 12997–13000.
57 M. W. Mahoney and P. Drineas, *Proc. Natl. Acad. Sci.*, 2009, **106**, 697–702.
58 C. Li, X. Wang, W. Dong, J. Yan, Q. Liu and H. Zha, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018, **41**, 1382–1396.
59 H. S. Seung, M. Opper and H. Sompolinsky, *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 287–294.
60 J. Busk, P. B. Jørgensen, A. Bhowmik, M. N. Schmidt, O. Winther and T. Vegge, *Mach. Learn.: Sci. Technol.*, 2021, **3**, 015012.
61 G. Kresse and J. Hafner, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1993, **47**, 558.
62 G. Kresse and J. Hafner, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1994, **49**, 14251.
63 G. Kresse and J. Furthmüller, *Comput. Mater. Sci.*, 1996, **6**, 15–50.
64 G. Kresse and J. Furthmüller, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1996, **54**, 11169.
65 S. Nosé, *J. Chem. Phys.*, 1984, **81**, 511–519.
66 H. J. Monkhorst and J. D. Pack, *Phys. Rev. B: Condens. Matter Mater. Phys.*, 1976, **13**, 5188.
67 J. P. Perdew, K. Burke and M. Ernzerhof, *Phys. Rev. Lett.*, 1996, **77**, 3865.
68 S. Grimme, J. Antony, S. Ehrlich and H. Krieg, *J. Chem. Phys.*, 2010, **132**, 154104.
69 A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, *et al.*, *J. Phys.: Condens. Matter*, 2017, **29**, 273002.
70 S. Pezzotti, A. Serva and M.-P. Gaigeot, *J. Chem. Phys.*, 2018, **148**, 174701.
71 H. B. Perets, Y. Lahini, F. Pozzi, M. Sorel, R. Morandotti and Y. Silberberg, *Phys. Rev. Lett.*, 2008, **100**, 170506.
72 G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni and G. Bussi, *Comput. Phys. Commun.*, 2014, **185**, 604–613.
73 M. Bonomi, G. Bussi, C. Camilloni, G. A. Tribello, P. Banáš, A. Barducci, M. Bernetti, P. G. Bolhuis, S. Bottaro and D. Branduardi, *Nat. Methods*, 2019, **16**, 670–673.
74 D. Sucerquia, C. Parra, P. Cossio and O. Lopez-Acevedo, *J. Chem. Phys.*, 2022, **156**, 154301.
75 D. P. Kingma and J. Ba, *arXiv*, 2014, preprint, arXiv:1412.6980, DOI: **10.48550/arXiv.1412.6980**.
76 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, *Advances in neural information processing systems*, 2019, 32, year.
77 S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt and K.-R. Müller, *Sci. Adv.*, 2017, **3**, e1603015.
78 R. Ramakrishnan, P. O. Dral, M. Rupp and O. A. Lilienfeld, *Sci. Data*, 2014, **1**, 1–7.
79 J. Gasteiger, F. Becker and S. Günnemann, *Adv. Neural. Inf. Process Syst.*, 2021, **34**, 6790–6802.
80 W. Hu, M. Shuaibi, A. Das, S. Goyal, A. Sriram, J. Leskovec, D. Parikh and C. L. Zitnick, *arXiv*, 2021, preprint, arXiv:2103.01436, DOI: **10.48550/arXiv.2103.01436**.
81 D. Branduardi, F. L. Gervasio and M. Parrinello, *J. Chem. Phys.*, 2007, **126**, 054103.
82 A. Ignaczak, R. Nazmutdinov, A. Goduljan, L. M. de Campos Pinto, F. Juarez, P. Quaino, G. Belletti, E. Santos and W. Schmickler, *Electrocatalysis*, 2017, **8**, 554–564.
83 A. Goduljan, L. M. de Campos Pinto, F. Juarez, E. Santos and W. Schmickler, *ChemPhysChem*, 2016, **17**, 500–505.
84 F. Lu, Y. Zhang, S. Liu, D. Lu, D. Su, M. Liu, Y. Zhang, P. Liu, J. X. Wang, R. R. Adzic, *et al.*, *J. Am. Chem. Soc.*, 2017, **139**, 7310–7317.
85 J. Staszak-Jirkovsky, R. Subbaraman, D. Strmcnik, K. L. Harrison, C. E. Diesendruck, R. Assary, O. Frank, L. Kobr, G. K. Wiberg, B. Genorio, *et al.*, *ACS Catal.*, 2015, **5**, 6600–6607.
86 X. Yang, A. Bhowmik, H. A. Hansen and T. Vegge, *DTU data*, 2023, DOI: **10.11583/DTU.22284514.v1**.

© 2023 The Author(s). Published by the Royal Society of Chemistry

Paper III

A comprehensive study of facet-dependent oxygen reduction dynamics on gold surfaces using metadynamics and graph neural networks

Xin Yang, Arghya Bhowmik, Tejs Vegge, and Heine Anton Hansen

To be submitted

A comprehensive study of facet-dependent oxygen reduction dynamics on gold surfaces using metadynamics and graph neural networks

Xin Yang, Arghya Bhowmik, Tejs Vegge, and Heine Anton Hansen*

Department of Energy Conversion and Storage, Technical University of Denmark, Anker Engelunds Vej, 2800 Kgs. Lyngby (Denmark)

E-mail: heih@dtu.dk

Abstract

Understanding the complex mechanisms of electrochemical reactions at the atomic level is at the core of optimizing energy conversion and storage devices. The exceptional activity of oxygen reduction reaction (ORR) on gold is broadly recognized while remains unexplained, especially for its facet-dependent catalytic behaviors. This research offers a deep dive into the facet-dependent dynamics of oxygen reduction on gold surfaces. Using graph neural networks (GNN) accelerated metadynamics, this study systematically investigated the ORR dynamics across different primary facets of gold, capturing the evolution of atomic structures along the reaction trajectory. Our simulations revealed the distinct formation of $^*H_2O_2$ intermediate on different surfaces and the crucial role of co-adsorbed species on the ORR dynamics. These finding can offer us deeper insights into the ORR reaction mechanism that is not accessible with traditional density functional theory (DFT) calculations, potentially paving the way for optimizing the ORR performance of gold-based catalysts.

Introduction

The electrochemical reduction of oxygen is a crucial process in various energy conversion and storage devices, including fuel cells and metal-air batteries.[1–3] The efficiency and selectivity of this reaction are predominantly governed by the nature of the electrode material. Recognizing this, there is a consistent pursuit of efficient and cost-effective catalysts towards oxygen reduction reaction (ORR) in academia and industry globally. Gold, traditionally viewed as a noble and therefore catalytically inert metal, has undergone a renaissance in the realm of catalysis over the past few decades.[4–9] In particular, gold nanoparticles have demonstrated exceptional catalytic activity for a range of reactions, from CO oxidation to selective hydrogenations.[8,10] This unexpected catalytic activity of gold is attributed to its unique electronic properties, particle size effects, and the influence of the support material.[11–15]

The ORR on gold has been extensively studied in both acidic and alkaline medias.[16] In acidic solutions, gold predominantly follows a 2-electron ($2e^-$) pathway, producing hydrogen peroxide. Intriguingly, the Au(100) surface in alkaline media, not only demonstrated in a 4-electron ($4e^-$) ORR pathway but even outperforms platinum within specific potential ranges. Additionally, ORR on gold showcases a pH-dependent catalytic behavior, with Au(100) favoring a complete four-electron transfer, in contrast to the partial two-electron transfer observed on other facets like Au(111) and Au(110).[4,17,18] While these experimental observations have been acknowledged for over a decade, an in-depth atomistic understanding of the reaction mechanisms remains elusive. Density functional theory (DFT) simulations combined with the computational hydrogen electrode (CHE) method have been instrumental but show limitations, especially in predicting the ORR activity of gold and the pH-dependent catalytic behavior. Using DFT calculations, Lu *et al.* highlighted the significant role of the interaction between co-adsorbed water and reaction intermediates in ORR on gold, facilitating O-O bond cleavage and thus promoting $4e^-$ reduction.[19] Duan and Henkelman suggested that the applied potential could influence the adsorption energies of ORR intermediates, resulting in the pH-dependent ORR on gold and a reduced theoretical overpotential.[20]

It is noteworthy that many studies have primarily focused on adsorption energetics, often neglecting the dynamic nature of electrochemical interfaces and its influence on ORR acativity. In our prior research,[21] we introduced a framework leveraging graph neural network (GNN) potentials to accelerate metadynamics simulations, shedding light on dynamic nature of electrochemical interfaces and the ORR kinetics at Au(100)–water interface. It offers direct insights into ORR kinetics, considering explicit solvents and long-scale molecular dynamics (MD) simulations, with a particular emphasis on the reaction kinetics of Au(100) surface.

In this extension work, we delve deeper into the ORR on prominent gold surfaces, namely Au(100), Au(110), and Au(111), incorporating explicit solvents and employing GNN-accelerated metadynamics for modeling ORR dynamics. Leveraging larger simulation boxes, we aim to provide a more comprehensive and nuanced understanding of the oxygen reduction process on gold surfaces. Our systematic exploration of interface dynamics across varied adsorbates offers an in-depth perspective on the nature of these interfaces. Notably, our simulations corroborate the facet-dependent behavior of ORR on gold, aligning well with experimental observations. Besides, our metadynamics investigations revealed the significant role of co-adsorbed species on ORR reactivity. With this research, we hope to bridge existing knowledge gaps and pave the way for the design of more efficient and robust gold-based electrocatalysts for ORR.

Computational methods

Generation of neural network potentials

Our methodology is built upon the techniques outlined in our previous research.[21] We utilized pretrained models from this work and derived the final dataset through the active learning framework built in it. The composition of various interfacial structures within the dataset, along with their respective error metrics, is presented in Table S1. We allocated 90% of the dataset for training and reserved the remaining 10% for validation, where the latter

was employed for early stopping once the force error reached an acceptable threshold. To assess the performance of the trained model, we utilized multiple error metrics, including the mean absolute error (MAE) and root mean squared error (RMSE) for both energy and force predictions.

Our MD and metadynamics simulations utilized an ensemble of six neural network potentials, each based on different architectures of the polarizable atom interaction neural network (PaiNN) model.[22] The architectures of these models, along with their respective error metrics trained on the final dataset, are detailed in Table S2. To introduce model diversity, we employed different node feature sizes, while maintaining a consistent cutoff radius of 5 Å for all models.

Both the model training and production simulations were executed on an NVIDIA GeForce RTX 3090 GPU, utilizing float32 precision. The weight parameters of models were initialized randomly and subsequently optimized on a consistent data split using stochastic gradient descent to minimize the mean square error (MSE) loss. We set the force loss weight and energy loss weight to 0.95 and 0.05, respectively, to ensure a high force prediction accuracy for propagating reliable MD simulations.

The Adam optimizer,[23] as implemented in PyTorch,[24] was employed to train our model parameters. We used an initial learning rate of 0.0001, default parameters of β_1=0.9 and β_2=0.999, and a batch size of 12. An exponential decay learning rate scheduler with a coefficient of 0.96 was used to adjust the learning rate every 100,000 learning steps.

DFT calculations

Our initial DFT dataset is derived from AIMD trajectories of Au(110)-water and Au(111)-water interfaces. Utilizing pretrained models eliminated the need for ab-initio reference structures from the Au(100)-water interface and reduced the number of reference structures for the Au(110)-water and Au(111)-water interfaces. The Au(110)-water system comprises 36 H_2O molecules atop a (2×3) tetragonal Au(110) surface (denoted as Au(110)-36H_2O).

Similarly, the Au(111)-water system is modelled as 36 H_2O molecules on atop of a (3×3) Au(111) surface (denoted as Au(111)-36H_2O). MD simulations were conducted in VASP[25–28] using these configurations for 10 ps, with a 0.5 fs timestep and a target temperature of 350K using the Nosé–Hoover thermostat.[29] The bottom two atomic layers remained fixed during the MD simulations. We adopted a 350 eV energy cutoff for plane-wave basis and a Monkhorst-Pack k-grid with the k-point density of 0.5 $Å^{-1}$.[30] The PBE functional, combined with the D3 Van der Waals correction, was used to approximate exchange-correlation effects.[31,32] The same parameters were employed for single-point DFT calculations during active learning iterations.

Production MD simulation

The production MD simulations driven by the NNP ensemble are conducted using the MD engine within Atomic Simulation Environment (ASE) python library.[33] We utilized larger simulation boxes to precisely capture the dynamics at the interface. The Au(100)-water interface was modeled with 102H_2O molecules on a (5×5) Au(100) surface, resulting in 431 atoms in total. The Au(110)-water interface had 115H_2O molecules on a (4×5) Au(110) surface, with 445 atoms. The Au(111)-water interface used a tetragonal $(5 \times 3\sqrt{2})$ Au(111) slab with 108 water molecules on it, with 474 atoms. The atomic structures for these interfaces are presented in Figure 1.

Building on these foundational interface structures, we incorporated O_2 molecules and hydroxyl groups to delve deeper into the adsorption energetics of key reaction intermediates and to set up the initial structures for metadynamics simulations. For every foundational interface, we considered 2OH, 4OH, 6OH, and 8OH cases by removing corresponding number of hydrogen atoms in the system. Besides, we also considered the $1O_2$ and $2O_2$ cases by adding corresponding number of oxygen atoms and removing corresponding number of water molecules. The resulting structures, along with their error metrics, are detailed in Table S1. For clarity and convenience, throughout the paper, we will label each system without

showing the number of H_2O as the number of water molecules is not important for discerning different systems and drawing our conclusions. For example, the Au(100)-$2O_2$/$98H_2O$ will be termed as Au(100)-$2O_2$ hereafter. The momentum of the model systems was initiated using a Maxwell–Boltzmann distribution, with the temperature set at 350 K. For each system, we propagated 1,500 ps MD simualations by Langevin dynamics with the target temperature of 350 K, the timestep of 0.25 fs, and the friction coefficient of 0.02. Of the 1500 ps MD trajectory, the initial 500 ps served for equilibration, while the subsequent 1000 ps was used to sample properties of interest. The uncertainty was quantified by the force standard deviation (SD), with a threshold set at 0.5 eV/Å. Simulations were halted if the force SD of a configuration exceeded this value. The formation energy of *OH and the adsorption energy of O_2 are calculated based on the method in Ref.[34] and our prior study.

Metadynamics simulation

In this study, all the enhanced sampling simulations are performed with the well-tempered version of metadynamics[35] The calculation of collective variables and bias potential of metadynamics is achieved by PLUMED which is interfaced to ASE.[33,36–38] We use the path collective variables and the same parameters in ref.[21] to describe the reaction. And the coordination numbers C_{O_2-O} and C_{O_2-H} are used to define the configurational space of the path.

Results and discussion

Regular MD simulations

We systematically investigated the dynamics at gold-water interfaces, specifically considering the presence of adsorbed O_2 and *OH. Figure 2 presents the density profiles of water molecules, oxygen atoms, and hydrogen atoms relative to their distance from gold surfaces. This figure focuses on systems containing pure water, eight hydroxyls, and two oxygen

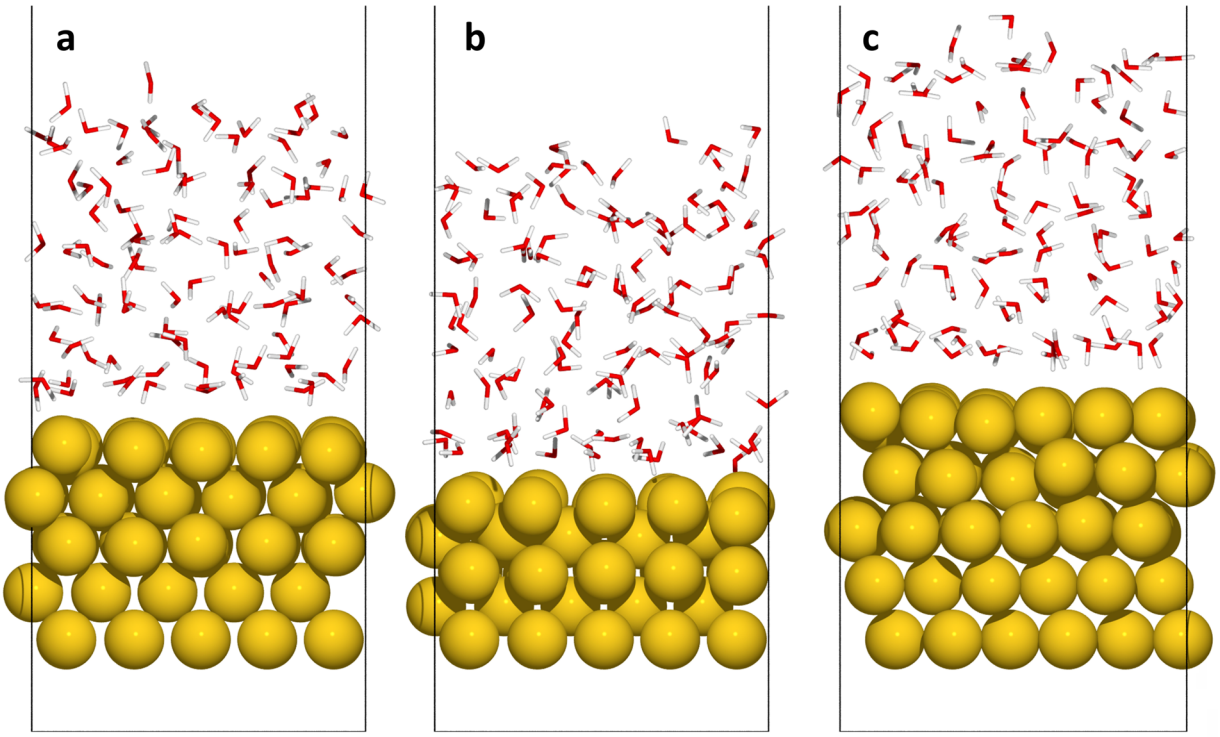

Figure 1: Side view of (a) Au(100)-102H_2O, (b) Au(110)-115H_2O, and (c) Au(111)-108H_2O interface structures.

molecules in the electrolyte. For detailed density profiles, average energy and uncertainties in other systems, readers are directed to Figure S1 to S21. In these density profiles. Notably, in these density profiles, two pronounced peaks are observed for all systems within 10 Å, suggesting the presence of two structured water layers near the slab. Among the surfaces studied, Au(110) exhibited the closest first peak to the surface at 2.5 Å. In comparison, the peaks for Au(100) and Au(111) are situated slightly farther away at 2.8 Å and 2.9 Å, respectively. The second layer of ordered water, as indicated by the second peak, is fairly consistent across the surfaces, positioned at 6.0 Å for both Au(100) and Au(110), and marginally farther at 6.1 Å for Au(111). Moving beyond 10 Å and up to 15 Å, the effects of the surface on water structuring become negligible. In this region, the density distribution of water molecules becomes almost constant and matches that of bulk water. Beyond 15 Å is the water–vacuum interface, where the densities gradually decline to zero.

Upon introducing 8 *OH groups, there is a noticeable shift in the density profiles as shown

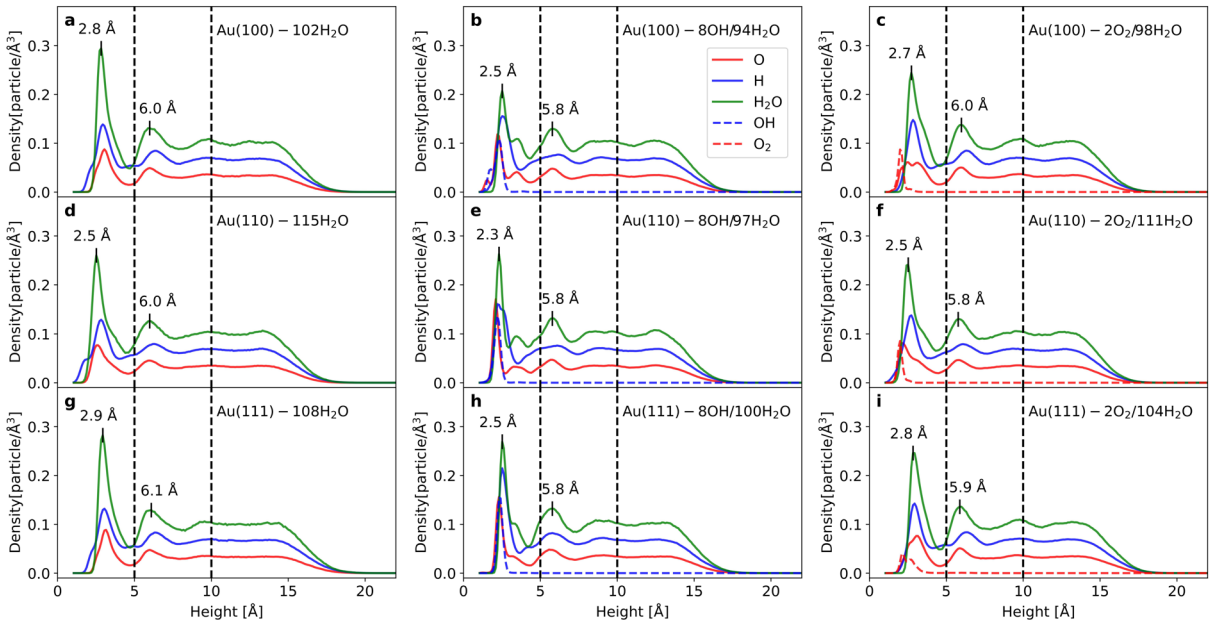

Figure 2: Density profiles of O, H, and H_2O as a function of the distance from Au slabs.

in Figure 2b, e, and h. The positions of first peaks in the density profiles are decreased to 2.5, 2.3, and 2.5 Å for Au(100), Au(110), and Au(111), respectively. Meanwhile the second peaks consistently locate at 5.8 Å across all surfaces. The presence of hydroxyls not only modifies the peak positions but also reshapes the overall density distributions of water molecules. This behavior can be attributed to the stronger chemical adsorption of hydroxyls compared to water and the hydrogen bond network formed between these hydroxyls and surrounding water molecules.[39] While *OH forms a direct chemical bond with gold atoms through chemisorption, water primarily interacts through weaker forces like van der Waals or hydrogen bonds. The chemisorbed *OH can act as an anchor point, fixing the surrounding water molecules through hydrogen bonding. This anchoring effect can reduce the mobility of water molecules and lead to a more tightly packed first layer of water molecules that closer to the surface.

In contrast, the inclusion of O_2 induces only minor shifts in the density distribution peaks without altering their overall shape as illustrated in Figure 2c, f, and i. The reason is that O_2 is a nonpolar molecule, which interacts weakly with the metal through physisorption

and lacks the ability to form hydrogen bonds with water. Consequently, *OH induces more pronounced structural changes in the water layer, leading to significant alterations in the density profiles, while the influence of O_2 remains relatively subtle.

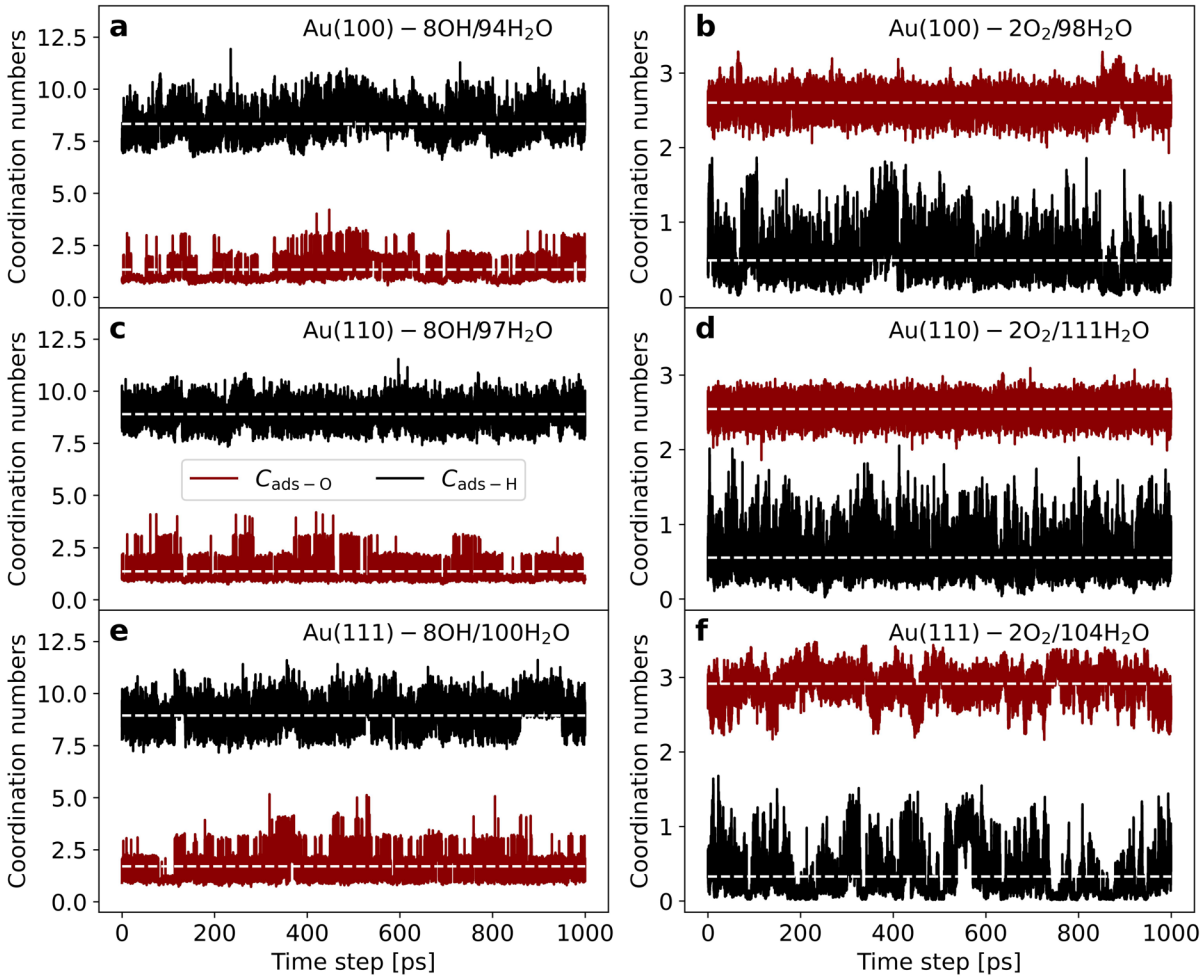

Figure 3: Evolution of coordination numbers $C_{\mathrm{ads-O}}$ and $C_{\mathrm{ads-H}}$ throughout MD simulations. The dashed lines denote the average coordination number.

As illustrated in Figure 3, we delved deeper into the evolution of coordination numbers for adsorbed species to gain insights into their local environments. Here, $C_{\mathrm{ads-O}}$ and $C_{\mathrm{ads-H}}$ represent the number of surrounding oxygen and hydrogen atoms for the adsorbates, respectively. For systems with eight hydroxyls, as depicted in Figure 3a, c, and e, the average coordination numbers $C_{\mathrm{ads-O}}$ for Au(100), Au(110), and Au(111) are 1.34, 1.36, and 1.70, respectively. Meanwhile, their corresponding $C_{\mathrm{ads-H}}$ values are 8.34, 8.90, and 8.94. In

systems with two oxygen molecules, the average $C_{\mathrm{ads-O}}$ values for Au(100), Au(110), and Au(111) are 2.60, 2.54, and 2.91, respectively, with $C_{\mathrm{ads-H}}$ values of 0.49, 0.56, and 0.33, respectively. A notable observation is that the fluctuation in $C_{\mathrm{ads-O}}$ is smaller than in $C_{\mathrm{ads-H}}$ across all systems. This can be attributed to the dynamic proton transfer in liquid water, which results in rapid changes in the local environments around hydroxyls. Conversely, due to the nonpolar nature of O_2, it is less inclined to form a hydrogen bond network with adjacent water molecules. Therefore, the bond breaking of O_2 can be difficult may not be accessible using regular MD simulations.

Table 1: Formation energies of *OH and adsorption energies of O_2 for different model systems

System	Species	θ (Coverage)	$\Delta E/n$ (eV)
Au(100)	2OH	0.080	1.131
	4OH	0.160	1.051
	6OH	0.240	1.087
	8OH	0.320	1.160
	$1O_2$	0.040	-0.657
	$2O_2$	0.080	-0.428
Au(110)	2OH	0.100	0.770
	4OH	0.200	0.755
	6OH	0.300	0.787
	8OH	0.400	0.785
	$1O_2$	0.050	-0.745
	$2O_2$	0.100	-0.736
Au(111)	2OH	0.067	0.880
	4OH	0.133	0.977
	6OH	0.200	1.055
	8OH	0.267	1.149
	$1O_2$	0.033	-0.768
	$2O_2$	0.067	-0.554

Table 1 exhibited the formation energies of *OH and the adsorption energies of O_2 molecule for each model system. For all three surfaces, the formation energy of *OH generally increases with increasing coverage. This suggests that as more *OH groups are adsorbed, it becomes energetically more favorable for them to form. Notably, Au(110) consistently exhibits the lowest formation energy for *OH, indicating that *OH adsorption is most favor-

able on this surface. Conversely, Au(100) displays the highest formation energy, especially with 8 co-adsorbed *OH groups. The adsorption energies of O_2 are negative for all systems, indicating that the adsorption process is exothermic and energetically favorable. Combining the stable evolution observed for $C_{\mathrm{ads-O}}$ in Figure 3, this suggests a preference for the inner-sphere mechanism, wherein the ORR predominantly occurs near the slab.

Metadynamics with single oxygen molecule

With the prepared the initial structures and well-performed MLIPs, now we are able to investigate the how the reaction happens with metadynamics simulations. We firstly simulated ORR with the presence of one single oxygen molecule in the liquid water. Previous studies suggests two plausible mechanisms for ORR: the inner-sphere mechanism, where O_2 is closely adsorbed on the slabs, and the outer-sphere mechanism, wherein ORR takes place several solvent layers away from the slab.[18,40,41] Our regular MD simulation results, in alignment with previous studies,[21] consistently show the O_2 molecule residing in the first water layer. Consequently, our metadynamics simulations were exclusively conducted following the inner-sphere mechanism. To ensure the reliability of the production metadynamics, it is imperative to sample and include a substantial number of reference structures into the training dataset, particularly those originating from rare events on the potential energy surface. This task poses a significant challenge given the intricate reaction systems explored in this study. Specifically, driving metadynamics to escape from the final state (4*OH in the liquid) is challenging as it correlates to the oxygen evolution process, which is inherently difficult to initiate and necessitates a high applied voltage for activation. Consequently, our metadynamics simulations for each system are confined to limited length-scales.

As depicted in Figure 4b, on the Au(100) slab, the metadynamics simulation stops at approximately 275 ps due to the emergence of structures with excessive uncertainty (force standard deviation exceeding 0.5 eV/Å), with the O-O bond breaking being observed at 232 ps. While extending the simulation length-scale is feasible through additional active learning

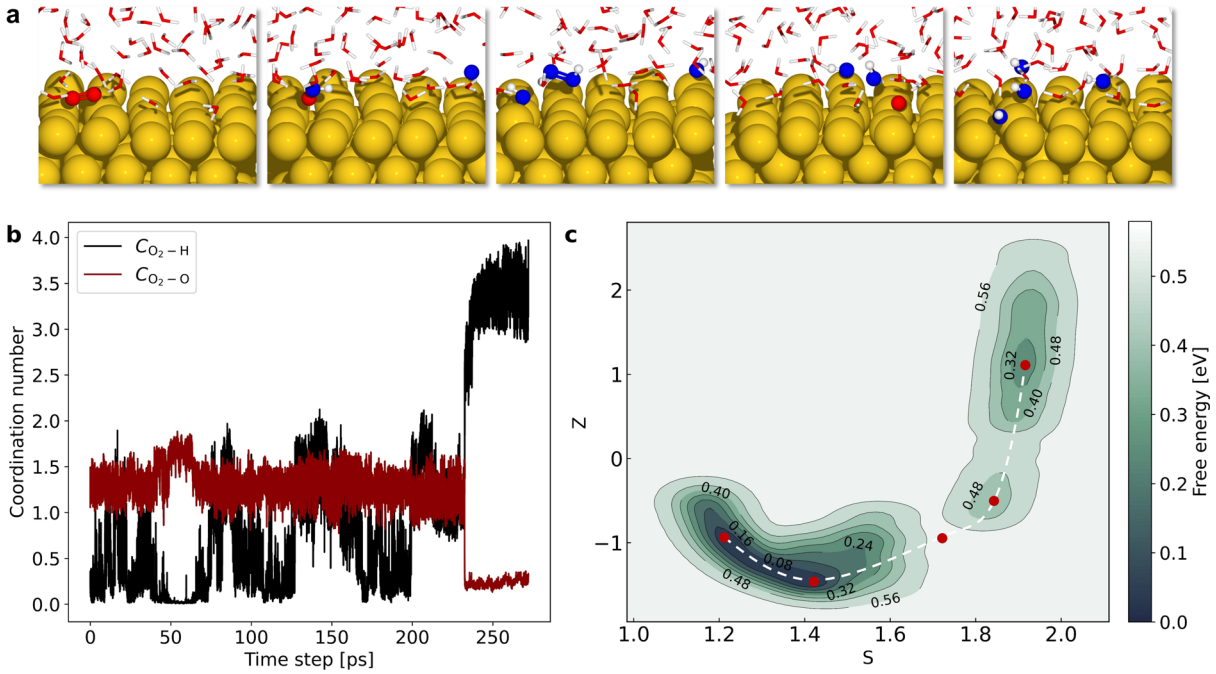

Figure 4: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(100) with one oxygen molecule described by path CVs.

iterations, we ascertain that the current length-scale of the simulation is sufficient to encapsulate the overall ORR process. The atomic structures of five representative configurations are illustrated in Figure 4a, corresponding to key points along the reaction pathway, as marked by the white dashed line in the free energy landscape depicted in Figure 4c. The free energy landscape, characterized by the path CVs, manifests three obvious basins. The initial stable states encompass two different kinds of configurations: Au(100)-1O_2/H_2O where the pure O_2 molecule is adsorbed onto the slab, and Au(100)-(1OH+1OOH)/H_2O showing the presence of one *OOH and one *OH. This indicates the proton transfer from the surrounding water molecules to the adsorbed O_2 molecule is facile with only negligible energy barrier. The third point along the reaction pathway is characterized by the adsorbed hydrogen peroxide (*H_2O_2), originating from the previously formed *OOH that accepted an additional proton from surrounding water molecules. However, the occurrence of this event is very rare, as illustrated by the sharp decline in C_{O_2-O} values depicted in Figure 4b, coupled with the

high free energy of approximately 0.57 eV. The O-O and O-H bond lengths in H_2O_2 are approximately 1.48 Å and 1.03 Å respectively, with the former aligning with measurements observed in gas-phase H_2O_2, while the latter is slightly elongated in comparison to its gas-phase counterpart. The introduction of hydrogen atoms weakens the O-O bond, breaking it into two hydroxyls. Interestingly, we identified the presence of an unbonded single oxygen atom at the fourth point (Au(100)-(1O+2OH)/H_2O), corresponding to the short interval in Figure 4b where the C_{O_2-O} values fluctuate around 2.75. The single oxygen atom is quickly protonated by adjacent water molecules, transitioning into a hydroxyl. Our metadynamics simulation elucidated that ORR on Au(100) proceeds in a four-electron transfer reaction pathway with a reaction barrier of approximately 0.50 eV.

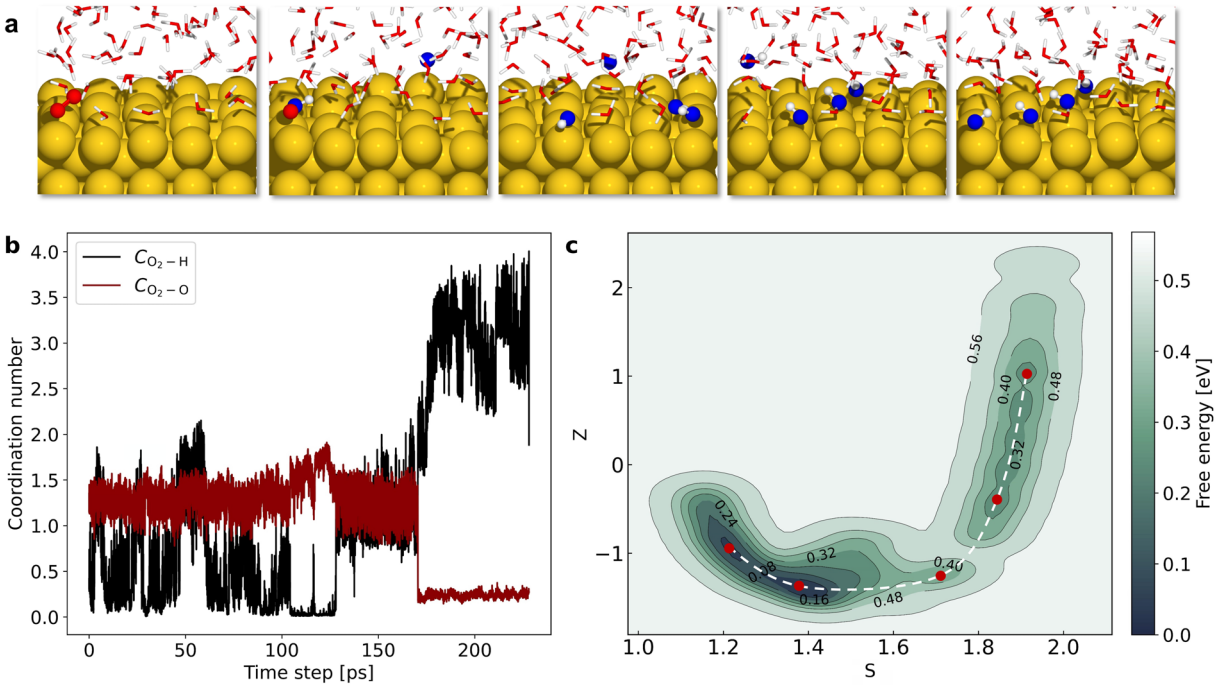

Figure 5: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(110) with one oxygen molecule described by path CVs.

Transitioning to the Au(110) slab, the metadynamics simulation halts at 228 ps, with the O-O bond breaking observed at 170 ps as depicted in Figure 5b. The reaction pathway,

as demonstrated in Figure 5a, is similar to that of Au(100), albeit without the identification of the unbonded oxygen atom. The free energy landscape of Au(110) closely mirrors that of Au(100), yet with only two basins as illustrated in Figure 5c. A notably more stable $^*H_2O_2$ intermediate emerges in the midway during intra-basin transition. This is further evidenced by the evolution of C_{O_2-O} values within the time interval of 131 ps to 170 ps, as demonstrated in Figure 5b. This observation aligns with experimental findings that ORR on Au(110) and Au(111) involves the formation of $^*H_2O_2$ intermediates.[4,17,40] Analogous to the Au(100) case, the O-O bond dissociation emerges as the rate-determining step, albeit with a slightly lower energy barrier of 0.48 eV.

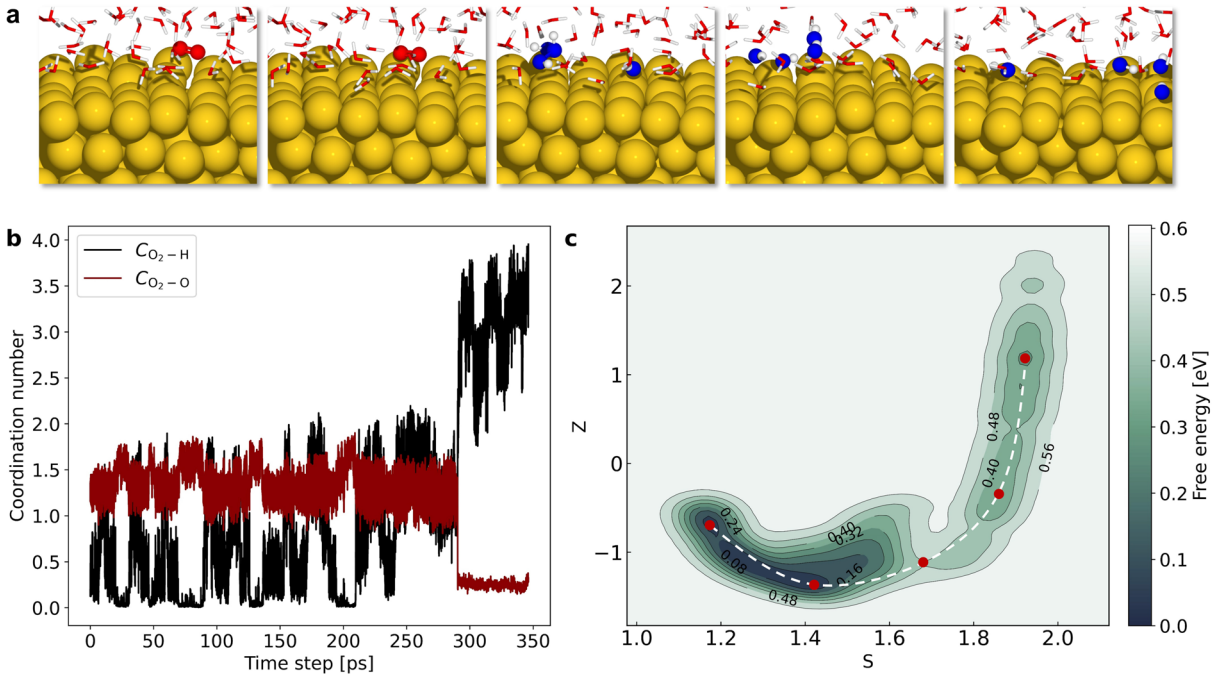

Figure 6: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(111) with one oxygen molecule described by path CVs.

Moving onto the Au(111) slab, the simulation halts at 350 ps, with the O-O bond breaking observed at 290 ps, as depicted in Figure 6b. Contrary to Au(100) and Au(110) cases, the first basin exclusively includes configurations with the oxygen molecule with no *OOH identified, as shown in Figure 6c. The adsorbed O_2 molecule is fully protonated to $^*H_2O_2$ in a short

time interval. Similarly to the Au(110) case, the existence of $*H_2O_2$ is more stable than in the Au(100) scenario. This is further substantiated by the evolution of C_{O_2-O} values within the time interval of 210 ps to 290 ps, as demonstrated in Figure 6b. The free energy barrier for oxygen reduction is approximately 0.42 eV.

Metadynamics with two oxygen molecules

While our simulations shed light on the variance in $*H_2O_2$ stability across different facets, they did not accurately predict the true ORR activity trends across these facets. We hypothesize that the co-adsorbed O_2 or hydroxyl groups on the surfaces might also exert influence on the ORR dynamics. To delve deeper into this aspect, we extended our metadynamics simulations to systems with two oxygen molecules present in the liquid water.

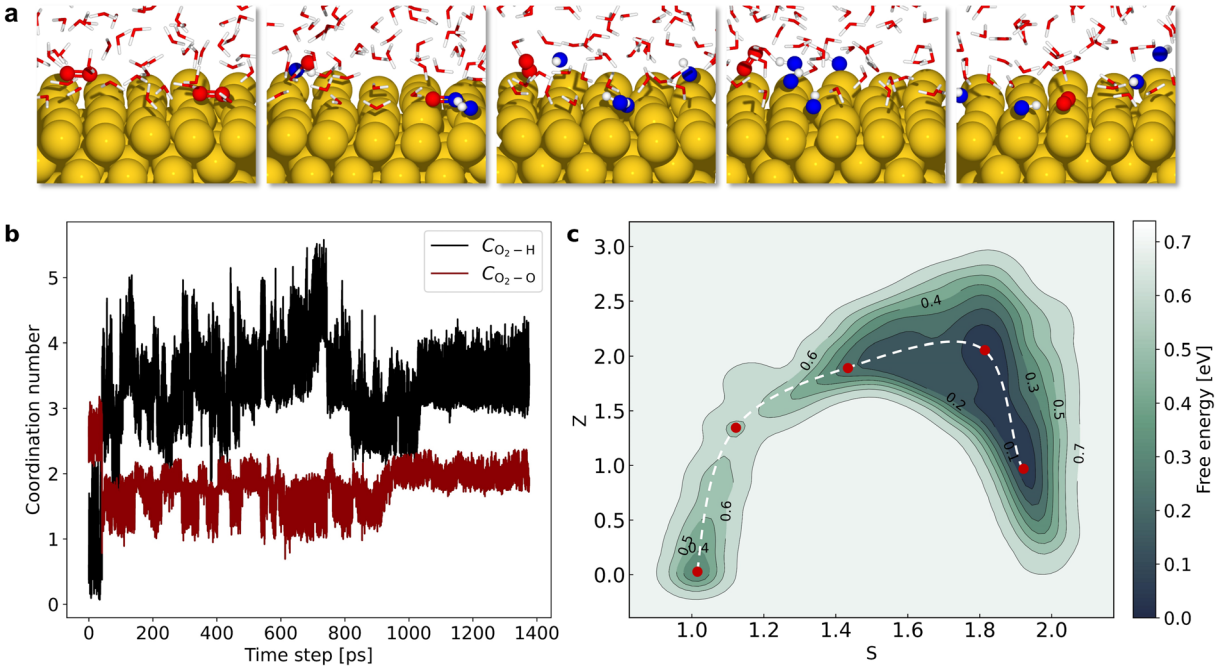

Figure 7: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(100) with two oxygen molecules described by path CVs.

Figure 7b illustrates a metadynamics simulation conducted at the Au(100)-$2O_2$/H_2O interface over a duration of 1480 ps, within the designated uncertainty threshold. The

dissociation O-O bond of the first O_2 molecule occured rapidly at 41 ps. However, the bond in the second oxygen molecule did not break during the entire simulation. This indicated the notable impact of co-adsorbed *OH on the ORR dynamics, hindering the reduction of the remaining O_2 molecules.

Figure 7c presents the free energy landscape, revealing two major basins and a minor one. The first major basin corresponds to Au(100)-$2O_2/H_2O$, the initial state of the reaction. As the reaction progresses, overcoming a free energy barrier of 0.24 eV, both O_2 molecules are partially protonated to form two *OOH molecules, signified by the minor energy basin. Following this, one of the *OOH groups undergoes O-O bond dissociation, resulting in an *OH group and an unbonded oxygen atom, which quickly accepts a proton from the surrounding water molecules to form another *OH. This *OH formation state exhibits substantial stability, persisting through the fourth and fifth points in the reaction pathway, all encapsulated within the second major energy basin.

In contrast with the Au(100)-$1O_2/H_2O$ case, the reduction of the first O_2 molecule is considerably more facile. Besides, the presence of $*H_2O_2$ is not observed during the simulation further validating that the reaction mechanism found in this study aligns well with experimental findings.

Transitioning to the analysis of the Au(110)-$2O_2/H_2O$ interface, the simulation stoped at 854 ps, with the O-O bond of the first O_2 molecule breaking at 432 ps as illustrated in Figure 8b. Similar to the Au(100)-$2O_2/H_2O$ case, the second O_2 remains intact during the entire simulation. The free energy landscape at this interface showcased two major basins, signifying the initial and final states of the reaction. At the second point of the reaction pathway, the state is Au(110)-(2OOH+2OH)/H_2O, exhibiting a similar adsorption behavior to the Au(100) surface. A notable observation was the presence of $*H_2O_2$ at the third point, which quickly transitioned into 2*OH groups. The second energy basin is quite stable, encapsulating both the fourth and fifth points in the reaction pathway. The reaction barrier for the Au(110)-$2O_2/H_2O$ interface is slightly higher than that of Au(100)-$2O_2/H_2O$,

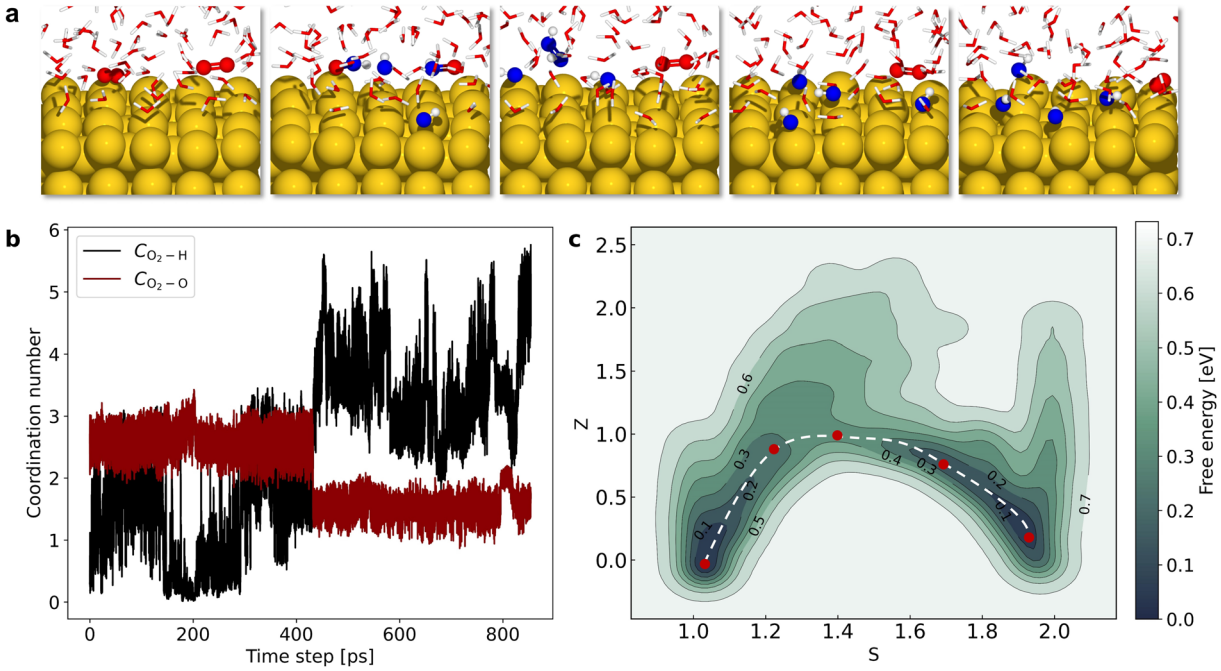

Figure 8: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(110) with two oxygen molecules described by path CVs.

being 0.32 eV.

Moving onto the Au(111)-$2O_2$/H_2O interface, the simulation stoped at 665 ps, demonstrating the O-O bond dissociation of the first O_2 molecule at 46 ps as depicted in Figure 9b. This behavior aligns with the previous observations on the Au(100) and Au(110) interfaces, where the dissociation of the second O_2 molecule proved to be challenging. As demonstrated in Figure 9c, the free energy landscape showcased two major basins, with the first corresponding to the initial state Au(111)-$2O_2$/H_2O and the second corresponding the final state Au(111)-($1O_2$+4OH)/H_2O. The presence of $*H_2O_2$ is more prominent on Au(111), expanding over a large area from the second point to the third point along the reaction trajectory. Amongst the interfaces studied, the Au(111)-$2O_2$/H_2O interface exhibited the lowest reaction barrier, recorded at 0.21 eV.

A notable observation across all three interfaces is that the evaluated reaction barriers are lower in comparison to their counterparts in the single oxygen metadynamics case, indicating

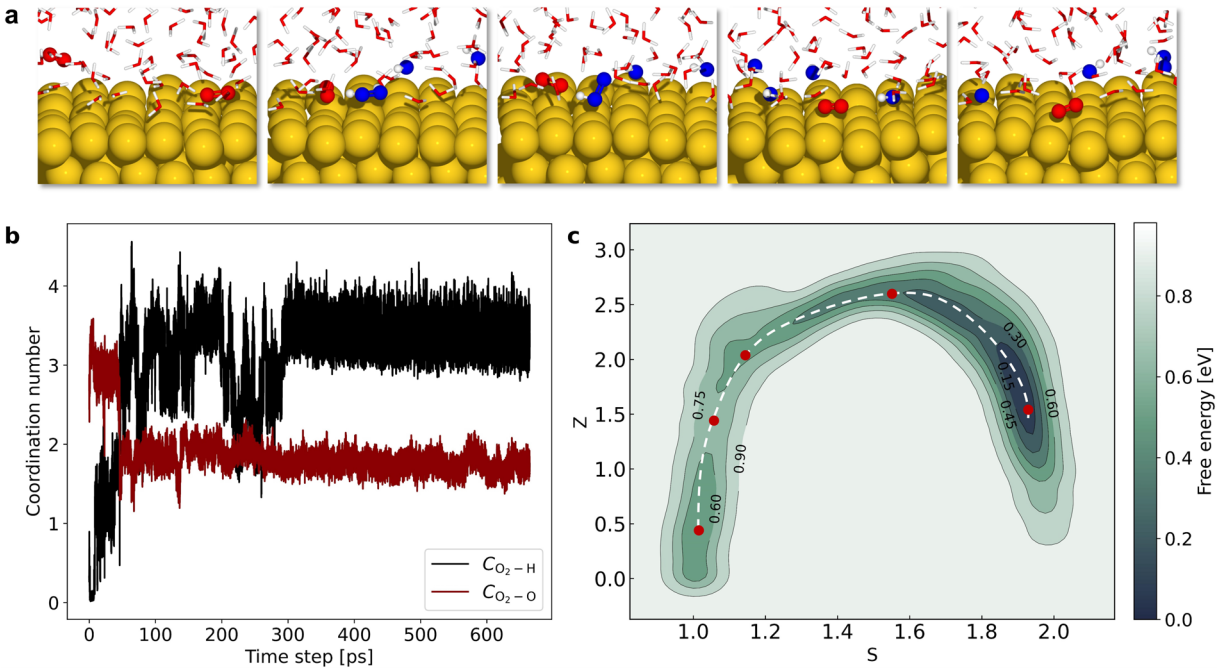

Figure 9: (a) Snapshots of representative atomic structures along the reaction trajectory. (b) Evolution of coordination numbers C_{O_2-O} and C_{O_2-H} throughout metadynamics simulation. (c) Free energy landscape of ORR on Au(111) with two oxygen molecules described by path CVs.

the facilitative role of co-adsorbed O_2 for ORR dynamics. Moreover, in each case, the second major basin, representing the co-adsorption of one oxygen molecule and four hydroxyl groups, was quite deep, demonstrating the remarkable stability of this state. This suggests that the presence of co-adsorbed hydroxyls can significantly influence the ORR on gold surfaces, potentially impeding the ORR process.

Conclusion and outlooks

In this comprehensive study, we have delved into the intricate dynamics of the oxygen reduction reaction (ORR) on prominent gold surfaces, specifically Au(100), Au(110), and Au(111). Our approach, which incorporated explicit solvents and utilized GNN-accelerated metadynamics, has provided a deep insight into the ORR mechanisms on these gold facets. Our regular MD simulations elucidate the dynamics at the gold-water interface, with an emphasis on the interactions involving adsorbed O_2 and *OH. The role of adsorbed *OH was found

to be significant in influencing the water layer structure. Our systematic investigations have revealed the facet-dependent behavior of ORR, with particular emphasis on the stability and behavior of $*H_2O_2$ across different gold facets. For instance, the presence of $*H_2O_2$ on Au(111) and Au(110) is prominent, broadly existing in the reaction trajectory. However, the presence of $*H_2O_2$ on Au(100) is merely observed. This findings align well with experimental observations. Furthermore, our metadynamics results emphasized the role of co-adsorbed species on the ORR reactivity. The inclusion of oxygen molecules can facilitate the reaction kinetics, while the co-adsorbed hydroxyl groups considerably impede the reaction process. The significant influence of adsorbed species, especially *OH, on the water layer structure underscores the need for a deeper understanding of their interactions and effects on other reaction intermediates. Additionally, extending this research to include different electrolytes can offer a broader perspective on how various solvents and ions modulate the ORR on gold surfaces. With the foundational knowledge established in this study, there lies the potential for the design and development of optimized gold-based electrocatalysts, paving the way for breakthroughs in electrocatalysis and related fields.

Acknowledgement

This work was supported by the Carlsberg Foundation through the Carlsberg Foundation Young Researcher Fellowship (Grant No. CF19-0304). The authors also thank the computational resources provided by the Niflheim Linux supercomputer cluster installed at the Department of Physics at the Technical University of Denmark.

References

(1) Seh, Z. W.; Kibsgaard, J.; Dickens, C. F.; Chorkendorff, I.; Nørskov, J. K.; Jaramillo, T. F. Combining theory and experiment in electrocatalysis: Insights into materials design. *Science* **2017**, *355*, eaad4998.

(2) Shao, M.; Chang, Q.; Dodelet, J.-P.; Chenitz, R. Recent advances in electrocatalysts for oxygen reduction reaction. *Chemical reviews* **2016**, *116*, 3594–3657.

(3) Kulkarni, A.; Siahrostami, S.; Patel, A.; Nørskov, J. K. Understanding catalytic activity trends in the oxygen reduction reaction. *Chemical Reviews* **2018**, *118*, 2302–2312.

(4) Rodriguez, P.; Koper, M. T. Electrocatalysis on gold. *Physical Chemistry Chemical Physics* **2014**, *16*, 13583–13594.

(5) Lopez, N.; Janssens, T.; Clausen, B.; Xu, Y.; Mavrikakis, M.; Bligaard, T.; Nørskov, J. K. On the origin of the catalytic activity of gold nanoparticles for low-temperature CO oxidation. *Journal of Catalysis* **2004**, *223*, 232–235.

(6) Mavrikakis, M.; Stoltze, P.; Nørskov, J. K. Making gold less noble. *Catalysis letters* **2000**, *64*, 101–106.

(7) Rodriguez, P.; Kwon, Y.; Koper, M. T. The promoting effect of adsorbed carbon monoxide on the oxidation of alcohols on a gold catalyst. *Nature chemistry* **2012**, *4*, 177–182.

(8) Hashmi, A. S. K.; Hutchings, G. J. Gold catalysis. *Angewandte Chemie International Edition* **2006**, *45*, 7896–7936.

(9) Hammer, B.; Norskov, J. K. Why gold is the noblest of all the metals. *Nature* **1995**, *376*, 238–240.

(10) Haruta, M.; Yamada, N.; Kobayashi, T.; Iijima, S. Gold catalysts prepared by coprecipitation for low-temperature oxidation of hydrogen and of carbon monoxide. *Journal of catalysis* **1989**, *115*, 301–309.

(11) Haruta, M. When gold is not noble: catalysis by nanoparticles. *The chemical record* **2003**, *3*, 75–87.

(12) Haruta, M. Gold rush. *Nature* **2005**, *437*, 1098–1099.

(13) Hutchings, G. J.; Brust, M.; Schmidbaur, H. Gold—an introductory perspective. *Chemical Society Reviews* **2008**, *37*, 1759–1765.

(14) Choudhary, T.; Goodman, D. Catalytically active gold: The role of cluster morphology. *Applied Catalysis A: General* **2005**, *291*, 32–36.

(15) Ishida, T.; Kinoshita, N.; Okatsu, H.; Akita, T.; Takei, T.; Haruta, M. Influence of the support and the size of gold clusters on catalytic activity for glucose oxidation. *Angewandte Chemie International Edition* **2008**, *47*, 9265–9268.

(16) Stamenkovic, V. R.; Strmcnik, D.; Lopes, P. P.; Markovic, N. M. Energy and fuels from electrochemical interfaces. *Nature materials* **2017**, *16*, 57–69.

(17) Quaino, P.; Luque, N.; Nazmutdinov, R.; Santos, E.; Schmickler, W. Why is gold such a good catalyst for oxygen reduction in alkaline media? *Angewandte Chemie International Edition* **2012**, *51*, 12997–13000.

(18) Ignaczak, A.; Nazmutdinov, R.; Goduljan, A.; de Campos Pinto, L. M.; Juarez, F.; Quaino, P.; Belletti, G.; Santos, E.; Schmickler, W. Oxygen reduction in alkaline media—a discussion. *Electrocatalysis* **2017**, *8*, 554–564.

(19) Lu, F.; Zhang, Y.; Liu, S.; Lu, D.; Su, D.; Liu, M.; Zhang, Y.; Liu, P.; Wang, J. X.; Adzic, R. R.; others Surface proton transfer promotes four-electron oxygen reduction on gold nanocrystal surfaces in alkaline solution. *Journal of the American Chemical Society* **2017**, *139*, 7310–7317.

(20) Duan, Z.; Henkelman, G. Theoretical resolution of the exceptional oxygen reduction activity of Au (100) in alkaline media. *ACS Catalysis* **2019**, *9*, 5567–5573.

(21) Yang, X.; Bhowmik, A.; Vegge, T.; Hansen, H. A. Neural network potentials for accelerated metadynamics of oxygen reduction kinetics at Au–water interfaces. *Chemical Science* **2023**, *14*, 3913–3922.

(22) Schütt, K.; Unke, O.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. International Conference on Machine Learning. 2021; pp 9377–9388.

(23) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(24) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; others Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **2019**, *32*.

(25) Kresse, G.; Hafner, J. Ab initio molecular dynamics for liquid metals. *Physical review B* **1993**, *47*, 558.

(26) Kresse, G.; Hafner, J. Ab initio molecular-dynamics simulation of the liquid-metal–amorphous-semiconductor transition in germanium. *Physical Review B* **1994**, *49*, 14251.

(27) Kresse, G.; Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational materials science* **1996**, *6*, 15–50.

(28) Kresse, G.; Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B* **1996**, *54*, 11169.

(29) Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics* **1984**, *81*, 511–519.

(30) Monkhorst, H. J.; Pack, J. D. Special points for Brillouin-zone integrations. *Physical review B* **1976**, *13*, 5188.

(31) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized gradient approximation made simple. *Physical review letters* **1996**, *77*, 3865.

(32) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *The Journal of chemical physics* **2010**, *132*, 154104.

(33) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; others The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter* **2017**, *29*, 273002.

(34) Kristoffersen, H. H.; Vegge, T.; Hansen, H. A. OH formation and H 2 adsorption at the liquid water–Pt (111) interface. *Chemical science* **2018**, *9*, 6912–6921.

(35) Perets, H. B.; Lahini, Y.; Pozzi, F.; Sorel, M.; Morandotti, R.; Silberberg, Y. Realization of quantum walks with negligible decoherence in waveguide lattices. *Physical review letters* **2008**, *100*, 170506.

(36) Tribello, G. A.; Bonomi, M.; Branduardi, D.; Camilloni, C.; Bussi, G. PLUMED 2: New feathers for an old bird. *Computer physics communications* **2014**, *185*, 604–613.

(37) Promoting transparency and reproducibility in enhanced molecular simulations. *Nature methods* **2019**, *16*, 670–673.

(38) Sucerquia, D.; Parra, C.; Cossio, P.; Lopez-Acevedo, O. Ab initio metadynamics determination of temperature-dependent free-energy landscape in ultrasmall silver clusters. *The Journal of Chemical Physics* **2022**, *156*, 154301.

(39) Groß, A.; Sakong, S. Ab initio simulations of water/metal interfaces. *Chemical Reviews* **2022**, *122*, 10746–10776.

(40) Ignaczak, A.; Santos, E.; Schmickler, W. Oxygen reduction reaction on gold in alkaline solutions–The inner or outer sphere mechanisms in the light of recent achievements. *Current Opinion in Electrochemistry* **2019**, *14*, 180–185.

(41) Goduljan, A.; de Campos Pinto, L. M.; Juarez, F.; Santos, E.; Schmickler, W. Oxygen Reduction on Ag (100) in Alkaline Solutions—A Theoretical Study. *ChemPhysChem* **2016**, *17*, 500–505.