Investigation of Chemical Reactivity by Machine-Learning Techniques

Von der Fakultät Chemie der Universität Stuttgart
zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Viktor Zaverkin

aus Luhansk (Ukraine)

Hauptberichter:	Prof. Dr. Johannes Kästner
Mitberichter:	Prof. Dr. Nongnuch Artrith
Prüfungsvorsitzender:	Prof. Dr. Blazej Grabowski
Tag der mündlichen Prüfung:	11. April 2022

Institut für Theoretische Chemie
Universität Stuttgart

2022

Abstract

The concepts of potential energy surface (PES) and molecular geometry, defined within the Born–Oppenheimer (BO) approximation, are essential for computational chemistry. The PES is a multi-dimensional function of atomic coordinates and can be obtained by the solution of the electronic Schrödinger equation (SE). While estimating individual points on the PES by first-principles methods, such as density functional theory (DFT), for even moderately sized molecular and material systems is computationally expensive, approximate methods allow for simulations of large systems over long time scales. Machine-learned interatomic potentials (MLIPs) have been gaining in importance since, once trained, they hold the promise to be as accurate as the reference ab-initio electronic structure method while having an efficiency on par with empirical force fields.

The derivation of a molecular representation is crucial for designing sample-efficient and accurate MLIPs, irrespective of the employed machine learning (ML) algorithm. Here, a novel molecular fingerprint referred to as Gaussian moment (GM) representation is developed. The GM representation is atom-centered, includes both structural and alchemical information of the local atomic neighborhood, and accounts for all essential invariances (translations, rotations, and permutations of like atoms). It is defined by pairwise atomic distance vectors and its runtime and memory complexity scale linearly with the number of atoms in the local neighborhood. Combined with atomistic neural networks (NNs), GM results in the Gaussian moment neural network (GM-NN) approach, which enables the generation of MLIPs with accuracy and efficiency similar to or better than other established ML models. The GM-NN source code is available free of charge from gitlab.com/zaverkin_v/gmnn.

Another intriguing aspect of MLIPs is the generation of highly informative training data sets and consequently, uniformly accurate machine-learned PESs, by applying active learning (AL) strategies. The fundamental quantity of AL is the query strategy – an algorithmic criterion for deciding whether a given configuration should be included in the training set or not. This criterion is defined here by employing the uncertainty estimate derived in the optimal experimental design (OED) framework. The proposed AL scheme allows for a more efficient estimation of the uncertainty of atomistic NNs. Thus, it allows for a more efficient

generation of transferable and uniformly accurate potentials by selecting the most informative or extrapolative configurations.

Aside from the conventional MLIPs, which typically aim to predict scalar energies, a methodology for learning the relationship between a structure and the respective tensorial property by atom-centered NNs has been proposed. To learn tensorial properties, specifically, the zero-field splitting (ZFS) tensors, the output of an NN is re-weighted by a tensor that satisfies the symmetry of the former. It has been shown that the proposed methodology can achieve high accuracy and has excellent generalization capability for out-of-sample configurations. Thus, it has been used to study the structural dependence of the ZFS tensor. Moreover, it has been demonstrated that complex processes such as spin-phonon relaxation can be investigated by employing machine-learned surrogate models.

Finally, the developed ML approaches have been used to study various surface processes in interstellar environments. Specifically, the adsorption and desorption dynamics of N and H_2 on different surfaces have been investigated, providing binding energies, sticking coefficients, and desorption temperatures. The diffusion of a nitrogen atom on the surface of amorphous solid water (ASW) at low temperatures has drawn particular attention. The study requires long time scales, short time steps in direct molecular dynamics (MD), and a very accurate PES. It has been achieved by combining MLIP driven MD simulations, free energy sampling using well-tempered metadynamics, and kinetic Monte Carlo (kMC) simulations based on the minima and saddle points on the free-energy surface (FES). The study revealed that N atoms, as a paradigmatic case for light and weakly bound adsorbates, can hardly diffuse on bare ASW at 10 K. Surface coverage may change that considerably, increasing the effective diffusion coefficient over 9–12 orders of magnitude.

Zusammenfassung

Die Konzepte der Potentialenergiefläche (PES, engl. für Potential Energy Surface) und der Molekülgeometrie sind in der Born-Oppenheimer-Näherung definiert und bilden eine Grundlage für die computergestützte Chemie. Die PES ist eine mehrdimensionale Funktion der Atomkoordinaten und kann durch die Lösung der elektronischen Schrödingergleichung erhalten werden. Die Berechnung einzelner Punkte auf der PES via First-Principles-Methoden, wie z. B. die Dichtefunktionaltheorie (DFT), wird bereits für Molekül- und Materialsysteme mittlerer Größe sehr rechenintensiv. Auf der anderen Seite ermöglichen Näherungsverfahren atomistische Simulationen großer Systeme über lange Zeitskalen. Mit ihrer, zur entsprechenden abinitio Referenzmethode ähnlichen, Präzision gewinnen die maschinell erlernten interatomaren Potentiale (MLIP, engl. für Machine Learned Interatomic Potential) an Bedeutung. Ein weiterer Vorteil ist die Recheneffizienz, vergleichbar zu empirischen Kraftfeldern.

Die Herleitung einer molekularen Repräsentation ist entscheidend für die Entwicklung von einem dateneffizienten und genauen MLIP und ist unabhängig vom maschinellen Lernverfahren. In dieser Arbeit wird eine alternative Methode entwickelt, die im Folgenden als Gauß Moment (GM) Darstellung bezeichnet wird. Die GM-Darstellung ist auf einem Atom zentriert, enthält sowohl strukturelle als auch chemische Informationen der lokalen atomaren Umgebung und berücksichtigt alle wichtigen Invarianzen (Translationen, Rotationen und Permutationen von gleichartigen Atomen). Sie wird ausschließlich durch Abstandsvektoren zwischen benachbarten Atomen definiert. Außerdem skaliert die GM linear mit der Atomanzahl in der lokalen atomaren Umgebung. Kombiniert mit atomistischen neuronalen Netzen (NNs) ergibt sich der Ansatz des Gauß Moment Neuronalen Netzwerkes (GM-NN). Dieser ermöglicht die Erzeugung von maschinell erlernten (ML, engl. für Machine Learning) Potentialen, die im Vergleich zu etablierten ML-Modellen vergleichbar oder besser in puncto Präzision und Recheneffizienz sind. Der GM-NN-Quellcode ist unter gitlab.com/zaverkin_v/gmnn frei verfügbar.

Ein weiterer wichtiger Aspekt von MLIPs ist die Generierung von hochinformativen Trainingsdatensätzen und damit gleichmäßig genauen ML-PESs. Dies kann durch Anwendung von Methoden des aktiven Lernens (AL) erreicht werden. Der Hauptbestandteil jeder AL-Methode

ist ein algorithmisches Kriterium für die Entscheidung, ob eine gegebene Konfiguration in den Trainingsdatensatz aufgenommen wird oder nicht. Ein solches Kriterium wird hier auf Basis der Unsicherheitsschätzung im Rahmen der optimalen Versuchsplanung (OED, engl. für Optimal Experimental Design) definiert. Der entwickelte AL-Algorithmus ermöglicht eine zeiteffizientere Schätzung der Unsicherheit atomistischer NNs. Durch die Auswahl der informativsten bzw. extrapolativsten Konfigurationen aus einem Trainingsdatensatz können übertragbare und gleichmäßig akkurate ML-Potentiale effizient erzeugt werden.

Neben den konventionellen MLIPs, die typischerweise skalare Energien vorhersagen, wurde hier eine Methode zum Erlernen der tensoriellen Molekül- und Materialeigenschaften durch atomzentrierte NNs eingeführt. Um die entsprechenden Eigenschaften, insbesondere den Tensor der Nullfeldaufspaltung (ZFS, engl. für Zero-Field Splitting), zu modellieren, wird die Ausgabe eines NN durch einen weiteren Tensor neu gewichtet. Dieser erfüllt die Symmetrie der zu modellierenden Eigenschaft. Die entwickelte Methode bietet eine hohe Genauigkeit und besitzt außerdem eine ausgezeichnete Generalisierungsfähigkeit auf Konfigurationen, die während des Trainings nicht benutzt wurden. Konkret wurde die Methode für die Erforschung der Abhängigkeit des ZFS-Tensors von der Molekülstruktur benutzt. Darüber hinaus konnte die Möglichkeit zur Untersuchung komplexer Prozesse, z. B. der Spin-Phonon-Relaxation, durch den Einsatz von ML-Modellen gezeigt werden.

Schließlich wurde eine Vielzahl von Oberflächenprozessen in interstellarer Umgebung untersucht, um die entwickelten ML-Methoden anwendungsbezogen zu nutzen. Insbesondere wurde die Adsorptions- und Desorptionsdynamik von N und H_2 auf verschiedenen Oberflächen simuliert sowie die Bindungsenergien, Adsorptionskoeffizienten und Desorptionstemperaturen berechnet. Ein besonderes Augenmerk wurde auf die Diffusion eines Stickstoffatoms auf amorphen Eisoberflächen bei niedrigen Temperaturen gelegt. Die entsprechende Studie erfordert lange Zeitskalen, kurze Zeitschritte in der direkten Moleküldynamik (MD) und eine hohe Genauigkeit der PES. Dies wurde durch die Kombination von MD-Simulationen auf einem MLIP, dem Sampling der Freie-Energie-Fläche (FES, engl. für Free-Energy Surface) mit der Methode der wohltemperierten Metadynamik und kinetischen Monte-Carlo-Simulationen (kMC) erreicht. Dabei wurden die Minima und Sattelpunkte auf der FES für die entsprechenden kMC-Simulationen verwendet. Das Resultat zeigte, dass N-Atome als paradigmatischer Fall für leichte und schwach gebundene Adsorbate auf den unkontaminierten, amorphen Eisoberflächen bei 10 K kaum diffundieren. Darüber hinaus konnte gezeigt werden, dass die Präsenz von anderen, inerten Atomen oder Molekülen den effektiven Diffusionskoeffizienten über neun bis zwölf Größenordnungen beeinflusst.

Peer-reviewed publications

This cumulative dissertation summarizes results that have been published in

[1]: V. Zaverkin and J. Kästner: *Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials*. Journal of Chemical Theory and Computation **16** (8), 5410–5421 (2020)

Copyright: Reprinted (adapted) with permission from Ref. [1]. Copyright 2020, American Chemical Society.

Contributions: V.Z. developed the molecular representation, conceived the neural network architecture, performed numerical simulations, prepared the figures, and wrote the first version of the manuscript. J.K. suggested the research question, managed the project and revised the first version of the manuscript. Both authors discussed results and commented on the manuscript.

[2]: V. Zaverkin and J. Kästner: *Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design*. Machine Learning: Science and Technology **2** (3), 035009 (2021)

Copyright: Reprinted (adapted) from Ref. [2]. Copyright 2021, IOP Publishing. Reproduced with permission. All rights reserved.

Contributions: V.Z. conceived the last-layer uncertainly for atomistic neural networks, implemented the proposed approach, performed numerical simulations, prepared the figures, and wrote the first version of the manuscript. J.K. managed the project and revised the first version of the manuscript. Both authors discussed results and commented on the manuscript.

[3]: V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner: *Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments*. Journal of Chemical Theory and Computation **17** (10), 6658–6670 (2021)

Copyright: Reprinted (adapted) with permission from Ref. [3]. Copyright 2021, American Chemical Society.

Contributions: V.Z. and D.H. contributed equally to this work. V.Z. conceived the GM-NN, implemented the extension to periodic systems, performed numerical simulations, including the production simulations for the TiO_2 and $Li_8Mo_2Ni_7Ti_7O_{32}$ systems, and prepared the figures. V.Z. and D.H. analyzed the symmetry of tensor contractions, designed the batch normalization for feature vectors, and improved the network training; D.H. improved the network initialization and extended the single Gaussian, used previously as a radial part of the Gaussian moment representation, to a finite sum of Gaussians. V.Z. and D.H. performed preliminary numerical experiments and wrote the first version of the manuscript. I.S. and J.K. managed the project and revised the first version of the manuscript. All authors discussed results and commented on the manuscript.

[4]: V. Zaverkin, J. Netz, F. Zills, A. Köhn, and J. Kästner: *Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments*. Journal of Chemical Theory and Computation **18** (1), 1–12 (2022)

Copyright: Reprinted (adapted) with permission from Ref. [4]. Copyright 2022, American Chemical Society.

Contributions: V.Z. conceived the machine learning method, implemented it to the GM-NN code, performed numerical simulations, and prepared figures. V.Z. and J.K. proposed the approach to modeling tensorial properties employing atomistic neural networks. J.N. and A.K. performed the ab-initio calculations, F.Z. and A.K. conducted the preliminary study. V.Z. wrote the first version of the manuscript, except for the description of ab-initio calculations written by J.N.; A.K. and J.K. managed the project and revised the first version of the manuscript. All authors discussed results and commented on the manuscript.

[5]: V. Zaverkin, G. Molpeceres, and J. Kästner: *Neural-network assisted study of nitrogen atom dynamics on amorphous solid water – II. Diffusion*. Monthly Notices of the Royal Astronomical Society **510** (2), 3063–3070 (2022)

Copyright: Reprinted (adapted) with permission from Ref. [5]. Copyright 2021, Oxford University Press.

Contributions: V.Z. conceived the method, including the implementation of the kinetic Monte Carlo algorithm and the PLUMED interface to the GM-NN code, performed numerical simulations, and wrote the first version of the manuscript. G.M. and J.K. suggested the research question, managed the project and revised the first version of the manuscript. All authors discussed results and commented on the manuscript.

Other publications by the author, not included in this thesis

[6]: G. Molpeceres, V. Zaverkin, and J. Kästner: *Neural-network assisted study of nitrogen atom dynamics on amorphous solid water – I. adsorption and desorption*. Monthly Notices of the Royal Astronomical Society **499** (1), 1373–1384 (2020)

[7]: G. Molpeceres, V. Zaverkin, N. Watanabe, and J. Kästner: *Binding energies and sticking coefficients of H_2 on crystalline and amorphous CO ice*. Astronomy & Astrophysics **648**, A84 (2021)

[8]: V. Zaverkin, D. Holzmüller, R. Schuldt, and J. Kästner: *Predicting properties of periodic systems from cluster data: A case study of liquid water.* The Journal of Chemical Physics **156** (11), 114103 (2022)

[9]: D. Holzmüller, V. Zaverkin, J. Kästner, and I. Steinwart: *A Framework and Benchmark for Deep Batch Active Learning for Regression.* ArXiv **abs/2203.09410** (2022)

[10]: V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner: *Exploring Chemical and Conformational Spaces by Batch Mode Deep Active Learning*. Digital Discovery **submitted** (2022)

[11]: V. Zaverkin, T. Lamberts, M. N. Markmeyer, and J. Kästner: *Tunnelling dominates the reactions of hydrogen atoms with unsaturated alcohols and aldehydes in the dense medium.* Astronomy & Astrophysics **617**, A25 (2018)

[12]: V. Zaverkin and J. Kästner: *Chapter 7 Instanton Theory to Calculate Tunnelling Rates and Tunnelling Splittings. Tunnelling in Molecules: Nuclear Quantum Effects from Bio to Physical Chemistry*, pp. 245–260. The Royal Society of Chemistry (2021)

Contents

Acknowledgements **XI**

List of Abbreviations **XIII**

1 Introduction **1**

2 Key concepts of quantum chemistry **5**
2.1 The Born–Oppenheimer approximation 5
2.2 Nearsightedness of electronic matter 8
2.3 Density functional theory 10

3 Artificial neural networks **15**
3.1 The basics of neural networks 15
3.1.1 Network architecture and forward propagation 15
3.1.2 Computing gradients via the backward propagation 19
3.1.3 The basics of gradient descent and convergence analysis 21
3.2 Improving the training of neural networks 23
3.2.1 Normalizing the input features 23
3.2.2 Choosing the activation function 26
3.2.3 Parameter initialization and learning rates 26
3.2.4 Momentum, adaptive learning rates, and mini-batches 29
3.3 Improving the generalization of neural networks 31
3.3.1 L^2 parameter regularization 32
3.3.2 Early stopping 34
3.3.3 The introduction of prior knowledge 38

4 On the theory of ultra-wide neural networks **41**
4.1 The basic network setup and training dynamics 41
4.2 The closed-form of training dynamics for linearized networks 44

4.3 Explaining the optimization of ultra-wide neural networks 48
4.4 Explaining the generalization of ultra-wide neural networks 49

5 Summary of research **51**
5.1 Gaussian moments and atomistic neural networks 51
5.1.1 Molecular representation 52
5.1.2 Gaussian moment neural network 56
5.2 Uncertainty of atomistic neural networks 59
5.3 Learning symmetric, traceless tensors 64
5.4 Investigating surface processes in interstellar environments 69

6 Conclusion and Outlook **75**

Bibliography **79**

Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials **97**

Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design **111**

Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments **133**

Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments **149**

Neural-network assisted study of nitrogen atom dynamics on amorphous solid water – II. Diffusion **163**

Declaration of Authorship **173**

Acknowledgements

Many people supported me in various ways during my doctoral studies and I would like to thank all of them sincerely.

First of all, I would like to express my deep gratitude to my esteemed supervisor – Prof. Johannes Kästner, for his invaluable advice and continued support during my doctoral study. His outstanding scientific knowledge and experience have inspired me greatly in my academic research and daily life. Our frequent meetings and lively discussions were pivotal for the presented work, helping me see my research from multiple perspectives and in doing so, aided me in forming a comprehensive and objective critique. I want to offer my special thanks to Prof. Nongnuch Artrith for agreeing to be the second examiner of this thesis and Prof. Blazej Grabowski for taking the chair on the examination board. My gratitude extends to the Studienstiftung des Deutschen Volkes (German National Academic Foundation) for the financial support necessary for me to undertake my doctoral studies.

Furthermore, I want to thank all my former PhD colleagues, particularly Daniel Born, whom I met during my master studies and whom I can also thank as a friend. I am particularly grateful to Germán Molpeceres for his support, fascinating joint projects, and the great time spent together in the office and social settings. In addition, I would like to thank David Holzmüller for mathematical support and his helpful comments on my work, and of course, the fruitful collaboration. I want to express my sincere gratitude to Prof. Andreas Köhn, Julia Netz, and Fabian Zills for their insightful cooperation and discussions. I wish to thank Prof. Ingo Steinwart for being my milestone examiner, Isabella Waldner for her versatile support at the beginning of and during my doctoral study, and Stefan Jagiella for technical support in my research. Moreover, I would like to extend my sincere thanks to Samuel Tovey for his helpful comments on my work.

I would like to express my deepest gratitude to my parents, Andrii and Olena, for their everlasting encouragement and unconditional support. Special thanks go to my father for guiding me as a person and scientist. I am particularly grateful to Viktor Golubev† (Mathematics) and Natalja Zolotukhina (Physics), my high school mentors, for inspiring me to pursue an academic career. I would also like to thank my friends, especially Lukas Seidel (and his family),

Marvin Poul, and Eric Weikum, for supporting me in several non-scientific ways and a sometimes necessary distraction. Above all, I want to thank Alina Makieieva for her love, patience, and constant support that accompanied me through all difficulties I have faced through my studies. This work would hardly have been possible without her.

List of Abbreviations

AIMD	Ab-initio molecular dynamics
AL	Active learning
ASW	Amorphous solid water
BN	Batch normalization
BO	Born–Oppenheimer
CP	Car–Parrinello
DFT	Density functional theory
FES	Free-energy surface
FF	Force field
GD	Gradient descent
GGA	Generalized gradient approximation
GM	Gaussian moment
GM-NN	Gaussian moment neural network
GP	Gaussian process
HF	Hartree–Fock
HK	Hohenberg–Kohn
ISM	Interstellar medium
kMC	Kinetic Monte Carlo
KS	Kohn–Sham
LDA	Local density approximation
LSDA	Local spin density approximation
MAE	Mean absolute error
MAXE	Maximal error
MBGD	Mini-batch gradient descent
MD	Molecular dynamics
ML	Machine learning
MLIP	Machine-learned interatomic potential
MML	Molecular machine learning

MSE	Mean squared error
NN	Neural network
NTK	Neural tangent kernel
NTP	Neural tangent parameterization
OED	Optimal experimental design
PES	Potential energy surface
QbC	Query by committee
RKHS	Reproducing kernel Hilbert space
RMSE	Root-mean-square error
SE	Schrödinger equation
SMM	Single-molecule magnet
SP	Standard parameterization
ZFS	Zero-field splitting
ZS	Zeeman-splitting

1 Introduction

Quantum mechanics, in one of its versions, is based on the so-called Schrödinger equation (SE) [13] that governs the wave function of a quantum-mechanical system. As Dirac noticed almost a century ago [14], the SE alone contains all necessary information to describe chemical phenomena and processes. The exact solution of the time-independent SE is possible only for the most simple systems. Therefore, to gain insights into complex chemical processes, the solution of the time-independent SE is simplified by employing the Born–Oppenheimer (BO) approximation [15]. Here, the motion of the nuclei is decoupled from the electronic movement. The BO approximation is justified by more than three orders of magnitude differences in masses of electrons and nuclei. Finally, the BO approximation introduces essential concepts including the potential energy surface (PES) and molecular geometry, thereby forming the basis of computational and theoretical chemistry, which describe chemical phenomena and processes.

Within the BO approximation, the nuclear coordinates enter the electronic SE parametrically. The resulting PES is a multidimensional real-valued function of atomic coordinates. Specifically, the PES is a $(3N_{\mathrm{at}} - 6)$-dimensional[1] effective potential energy hypersurface in which N_{at} nuclei move. One can use different techniques to estimate the individual points on the PES, from ab-initio electronic structure theory to empirical fits by force fields. While the ab-initio techniques provide chemically accurate potential energy hypersurfaces, they scale poorly with the system size. Therefore, their application in atomistic simulations, which require energies and forces for many atomic configurations, like molecular dynamics (MD) or geometry optimization, is limited to relatively small or moderately sized systems. The empirical force fields (FFs), on the contrary, are computationally very efficient and can be used to compute the PES of large systems. However, they lack accuracy, have limited transferability [16], and are generally not able to describe bond-formation and bond-breaking.

The search for methods that allow mapping atomic positions $\{\mathbf{R}_i\}_{i=1}^{N_{\mathrm{at}}}$ and nuclear charges $\{Z_i\}_{i=1}^{N_{\mathrm{at}}}$ to the PES, i.e. $f : \{Z_i, \mathbf{R}_i\} \mapsto E$, and have the accuracy comparable to first-principles methods, is part of the current research. Recently, machine-learned interatomic

[1] Translations and rotations of the whole system are excluded as they do not change the potential energy.

potentials (MLIPs) [17–27] have risen in popularity due to their ability to learn from ab-initio data and approach the accuracy as the underlying first-principles reference method. Moreover, MLIPs have proven to generalize well to unseen configurations, transfer to arbitrary-sized systems, and describe bond-breaking and bond-formation as opposed to empirical FFs [16]. For modeling chemical processes, several machine learning (ML) techniques, from linear regression models to artificial neural networks (NNs), can be used to predict a variety of chemical and physical properties of molecules and bulk solids. The approximation capabilities of NNs have promoted their broad application in computational chemistry and materials science [28]. Initially, NNs were applied to represent PESs of small atomistic systems [29, 30]. Their application to high-dimensional systems was then extended later [31]. Once trained, the computational cost of MLIPs based on NNs does not scale with the number of data points used for training as opposed to kernel-based models. Therefore, training sets can be as large as necessary to achieve the desired accuracy for applications in atomistic simulations such as MD.

The successful application of ML approaches to atomically resolved systems faces several challenges, irrespective of the specific ML algorithm type employed. One of the biggest challenges in designing MLIPs is deriving a suitable representation of an atomistic system [21, 27, 32, 33]. This challenge is manifold since the atomistic system of an arbitrary size needs to be encoded in a feature vector of fixed dimension. At the same time, the complete information about the three-dimensional configuration, including the fundamental symmetries, e.g., the rotational, translational, and permutational invariance of scalar properties, has to be encoded in this vector.

Additionally, the atomistic representation has to be systematically improvable. This means that the accuracy of predictions should increase with an increasing dimension of the feature vector. The molecular fingerprint must be computationally efficient and transferable between similar systems and their configurations. The mapping from an atomistic structure to the respective feature vector must be unique, i.e., feature vectors corresponding to different atomic arrangements should have different sets of values. Also, the extension to additional symmetries and constraints like the periodic boundary conditions in bulk solids must be easily accessible. Finally, the atomistic representation has to be differentiable with respect to the atomic coordinates to calculate forces and Hessians. This is certainly not the whole list of requirements to an "ideal" representation but is sufficient to state its relevance.

Another challenge is the generation of appropriate training data because of the dimensionality of the chemical and conformational spaces. One should also not neglect the high compu-

tational cost of ab-initio methods. In general, this problem requires the ML model to detect the most informative structures sampled in an on-the-fly fashion or contained in a pre-computed data set, such that the computationally expensive ab-initio calculations are performed only for them. It is achievable by employing active learning (AL) [34], a particular form of ML whose aim is to learn general-purpose models with a minimal number of training data. The key ingredient for an AL algorithm is the query strategy – an algorithmic criterion for deciding whether a given configuration should be included in the training set. While for the Gaussian process (GP)-based models, it can be defined employing their inherent Bayesian predictive variance, the NN-based models previously applied in computational chemistry require at least the training of an ensemble of models, the so-called query by committee (QbC) approach [34]. The search for alternative AL approaches for atomistic NNs that do not require training of an ensemble of models is challenging. However, the respective task is even more demanding since frequent re-training can be considered the computational bottleneck for applying AL to atomistic NNs, in general. Thus, the search for NN architectures that would allow fast re-training is highly desirable.

Many properties of materials and molecules are described by symmetric tensors, typically of rank two, while, for example, elastic properties of a medium require a fourth-rank tensor. Especially, magnetic properties of transition metal complexes, mediated by the Zeeman and the zero-field splitting (ZFS) interactions [35], have drawn particular attention due to the potential application of such complexes as single-molecule magnets (SMMs), molecular quantum bits, and spintronic devices [36–39]. Here, the main challenge is that the modeling of tensorial properties, in general, requires that the model respects the appropriate geometric transformations, rather than invariance, when, e.g., the reference frame rotates. The ability of modeling magnetic anisotropies, specifically, can help investigating a manifold of complex chemical and physical phenomena and processes like spin-phonon relaxation. The application of atomistic NNs to predicting tensorial properties, often defined for a specific atom rather than for the whole structure, has not been studied and is an integral part of the current research.

The last but not least challenge that was tackled during this thesis is applying MLIPs to study real-world chemical processes and phenomena, rather than using the developed approaches with benchmark systems. This challenge is two-fold, from the ML perspective. The generation of an accurate MLIP requires carefully sampled training data at the reference level of theory. Implying that obtained atomic configurations cover the relevant part of configurational and chemical space uniformly, the correct physics is imposed on the respective ML model. The assessment of the quality of the trained ML model is also challenging since it

requires to answer whether the obtained model represents correct physics or not. Particularly, the investigation of diffusion processes of adsorbates other than H and D in interstellar environments has drawn much attention and is highly disputed in the literature, see, e.g., Refs. [40–42]. However, a detailed study using ab-initio methods is unfeasible since it requires long time scales and short steps in direct MD. MLIPs providing exceptionally accurate and efficient PESs can be employed to shed some light on this intriguing research question.

The focus of this thesis is, therefore, the development of new approaches based on artificial NNs to solve the challenges mentioned above, from the derivation of a suitable representation of an atomistic system [1] to the application of developed ML approaches to the investigation of chemical phenomena and processes [5]. The main developments of this cumulative thesis are presented in Refs. [1–5] and summarized in Chapter 5. An introduction to quantum chemistry and artificial NNs is given in Chapter 2 and Chapter 3 respectively, while Chapter 4 elaborates on the recently proposed neural tangent kernel (NTK) theory [43]. Concluding remarks and an outlook are presented in Chapter 6.

2 Key concepts of quantum chemistry

This chapter presents the basic concepts of quantum chemistry, relevant for modeling chemical systems by machine learning (ML). Starting with the Born–Oppenheimer (BO) approximation [15], the most important concept is presented since the ML-based models will be trained on electronic energies defined in Section 2.1. Moreover, the nearsightedness of electronic matter is discussed in Section 2.2, following Refs. [44, 45]. This concept will be important for deriving models whose computational cost scales linearly with the number of atoms in the atomistic system. Lastly, the basic concepts of the density functional theory (DFT), probably the most broadly used first-principles method to generate reference data for ML algorithms, are reviewed in Section 2.3, following Ref. [46].

2.1 The Born–Oppenheimer approximation

Here, the Born–Oppenheimer (BO) approximation [15], essential to solving the time-independent Schrödinger equation (SE), is reviewed. This approximation allows decoupling the motion of the nuclei from electronic movement, justified by the differences in masses by a magnitude of 10^3. The BO approximation is essential for concepts such as the potential energy surface (PES) and molecular geometry. It is indispensable for conventional molecular machine learning (MML), which aims to map the atomic coordinates to the corresponding energy value

$$\{\mathbf{R}_K, Z_K\}_{K=1}^{N_{\mathrm{at}}} \mapsto E, \tag{2.1}$$

where $\mathbf{R}_K$ are the coordinates of nuclei, Z_K are the corresponding atomic numbers, and E is the electronic energy of the system.

For a general molecular system, the total (non-relativistic) Hamiltonian operator reads

$$\begin{aligned}\hat{H}_{\mathrm{tot}} &= -\sum_{K=1}^{N_{\mathrm{nuc}}} \frac{\hbar^2}{2M_K}\nabla_K^2 - \sum_{i=1}^{N_{\mathrm{el}}} \frac{\hbar^2}{2m_{\mathrm{e}}}\nabla_i^2 + \frac{e^2}{4\pi\epsilon_0}\sum_{i>j}\frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \\ &- \frac{e^2}{4\pi\epsilon_0}\sum_{K,i}\frac{Z_K}{|\mathbf{r}_i - \mathbf{R}_K|} + \frac{e^2}{4\pi\epsilon_0}\sum_{K>L}\frac{Z_K Z_L}{|\mathbf{R}_K - \mathbf{R}_L|} \\ &= \hat{T}_{\mathrm{N}} + \hat{T}_{\mathrm{e}} + \hat{V}_{\mathrm{ee}} + \hat{V}_{\mathrm{eN}} + \hat{V}_{\mathrm{NN}},\end{aligned} \tag{2.2}$$

where $\hat{T}_{\mathrm{N}}$ and $\hat{T}_{\mathrm{e}}$ are the kinetic energy operators of nuclei and electrons respectively, $\hat{V}_{\mathrm{NN}}$ is the Coulomb repulsion between nuclei, and $\hat{V}_{\mathrm{ee}}$ is the Coulomb repulsion between electrons. The remaining term is the Coulomb attraction between the nuclei and the electrons $\hat{V}_{\mathrm{eN}}$. The time-independent SE reads[1]

$$\hat{H}_{\mathrm{tot}} \left|\Psi_i^{\mathrm{tot}}(\mathbf{R},\mathbf{r})\right\rangle = E_i^{\mathrm{tot}} \left|\Psi_i^{\mathrm{tot}}(\mathbf{R},\mathbf{r})\right\rangle, \tag{2.3}$$

with $\left|\Psi_i^{\mathrm{tot}}(\mathbf{R},\mathbf{r})\right\rangle$ being the solution of the molecular SE to the energy state E_i^{total}, $\mathbf{R}$ being the coordinates of the nuclei and $\mathbf{r}$ being the electronic spatial coordinates. In general, the electronic coordinates have to be extended by the spin coordinate, neglected here for simplicity.

The following derivation of the BO approximation follows Ref. [46], and one writes for the electronic Hamiltonian operator

$$\hat{H}_{\mathrm{e}} = \hat{T}_{\mathrm{e}} + \hat{V}_{\mathrm{ee}} + \hat{V}_{\mathrm{eN}} + \hat{V}_{\mathrm{NN}} = \hat{H}_{\mathrm{tot}} - \hat{T}_{\mathrm{N}}. \tag{2.4}$$

Here, the so-called mass-polarization term has been neglected. It originates from the fact that there is no rigorous way to separate the center of mass motion from internal motion. However, the mass-polarization term leads only to energy contributions less than the electronic energy by a factor proportional to the ratio of the electronic and nuclear masses ($\sim 10^3$).

Now, fixing the coordinates of nuclei $\bar{\mathbf{R}}$ and assuming that the complete set of solutions

[1] Here, the bra-ket notation for wave functions and multi-dimensional integrals is used

$$\begin{aligned}&|\Psi\rangle \equiv \Psi, \quad \langle\Psi| \equiv \Psi^*, \\ &\int \Psi^* \Psi \mathrm{d}\mathbf{r} \equiv \langle\Psi|\Psi\rangle, \\ &\int \Psi^* \hat{H} \Psi \mathrm{d}\mathbf{r} \equiv \langle\Psi|\hat{H}|\Psi\rangle,\end{aligned}$$

where ket $|\Psi\rangle$ denotes the wave function, and bra $\langle\Psi|$ denotes its complex conjugate.

for the electronic SE is available, one may write

$$\hat{H}_{\mathrm{e}}\left(\bar{\mathbf{R}}\right)\left|\Psi_n\left(\bar{\mathbf{R}},\mathbf{r}\right)\right\rangle = E_n\left(\bar{\mathbf{R}}\right)\left|\Psi_n\left(\bar{\mathbf{R}},\mathbf{r}\right)\right\rangle, \tag{2.5}$$

where index n denotes the nth electronic state with energy $E_n\left(\bar{\mathbf{R}}\right)$. Since the Hamiltonian operator is hermitian, one can select its solutions to be orthogonal and normalized, such that the total (exact) wave function for the molecular system can be expanded in the respective basis as

$$\left|\Psi_i^{\mathrm{tot}}\left(\mathbf{R},\mathbf{r}\right)\right\rangle = \sum_{m=1}^{\infty}\Phi_{im}\left(\mathbf{R}\right)\left|\Psi_m\left(\mathbf{R},\mathbf{r}\right)\right\rangle, \tag{2.6}$$

where the expansion coefficients $\Phi_{im}\left(\mathbf{R}\right)$ are the functions of the nuclear coordinates and are solutions to the SE of nuclei.

By applying the Hamiltonian defined in Equation (2.4) to the wave function from Equation (2.6) and subsequently projecting the result on a specific electronic wave function $\left\langle\Psi_n\left(\mathbf{R},\mathbf{r}\right)\right|$ one obtains

$$\sum_{m=1}^{\infty}\left\langle\Psi_n\left(\mathbf{R},\mathbf{r}\right)\right|\hat{T}_{\mathrm{N}}\left|\Phi_{im}\left(\mathbf{R}\right)\Psi_m\left(\mathbf{R},\mathbf{r}\right)\right\rangle + E_n\left(\mathbf{R}\right)\Phi_{in}\left(\mathbf{R}\right) = E_i^{\mathrm{tot}}\Phi_{in}\left(\mathbf{R}\right), \tag{2.7}$$

where it is noted that the electronic Hamiltonian operator acts only on the electronic wave function. The first term in Equation (2.7), which includes the kinetic energy operator of the nuclei, can be written explicitly by applying the product rule as

$$\sum_{m=1}^{\infty}\left[\left\langle\Psi_n\right|\hat{T}_{\mathrm{N}}\left|\Psi_m\right\rangle - \sum_{K=1}^{N_{\mathrm{nuc}}}\frac{\hbar^2}{M_K}\left\langle\Psi_n\right|\nabla_K\left|\Psi_m\right\rangle\nabla_K\right]\Phi_{im}\left(\mathbf{R}\right) + \hat{T}_{\mathrm{N}}\Phi_{in}\left(\mathbf{R}\right), \tag{2.8}$$

where $\left|\Psi_n\right\rangle \equiv \left|\Psi_n\left(\mathbf{R},\mathbf{r}\right)\right\rangle$, for simplicity. The first two terms, the so-called non-adiabatic coupling elements, contain the electronic wave function and describe the coupling between different electronic states. In most cases, this interaction can be neglected, and the nuclei move on a potential energy surface (PES) which is a solution of the electronic SE. The SE for the nuclei in the BO approximation reads

$$\left(\hat{T}_{\mathrm{N}} + E_n\left(\mathbf{R}\right)\right)\Phi_{in}\left(\mathbf{R}\right) = E_i^{\mathrm{tot}}\Phi_{in}\left(\mathbf{R}\right). \tag{2.9}$$

Note that the BO approximation is limited and breaks, e.g., near conical intersections [47]. However, it is sufficient for most quantum chemistry applications, and this concept is broadly

used to construct a machine-learned interatomic potential (MLIP) trained on electronic energies $E_n(\mathbf{R})$. As a final remark, a few properties of $E_n(\mathbf{R})$ important to encode in MLIPs are presented. These are:

1. The electronic energy $E_n(\mathbf{R})$ is invariant to the translations of the molecular structure, i.e.
$$E_n(\mathbf{R}) = E_n(\mathbf{R} + \mathbf{a}),$$
where $\mathbf{a} \in \mathbb{R}^3$ is the vector by which the structure is shifted.

2. The electronic energy $E_n(\mathbf{R})$ is invariant to the rotations and inversions of the molecular structure, i.e.
$$E_n(\mathbf{R}) = E_n(\mathbf{U}\mathbf{R}),$$
where $\mathbf{U} \in O(3)$ and $O(3)$ is the orthogonal group in $\mathbb{R}^3$.

3. The electronic energy $E_n(\mathbf{R})$ is invariant to the permutation of like atoms, i.e.
$$E_n(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_{N_{\mathrm{at}}}) = E_n\left(\mathbf{R}_{\pi_1}, \mathbf{R}_{\pi_2}, \ldots, \mathbf{R}_{\pi_{N_{\mathrm{at}}}}\right).$$
where $\pi \in P$ and P is a set of permutations of $(1, \ldots, N_{\mathrm{at}})$.

2.2 Nearsightedness of electronic matter

Another important concept of quantum and computational chemistry is the locality of atomic interactions, which is based on a widely applicable physical principle, first introduced qualitatively in Ref. [44]. This principle is called the nearsightedness of electrons in many-atom systems and can, in many cases, be used to explain the success of machine learning (ML) when applied to chemical problems. The concept of the nearsightedness of electronic matter, which often is a good approximation for constant-chemical-potential conditions [48], suggests that for a fixed chemical potential μ, external potential $v(\mathbf{r})$, and an absence of long-range electric fields, local electronic properties, such as the electron density at $\mathbf{r}$

$$n(\mathbf{r}) = 2\sum_{i=1}^{N_{\mathrm{el}}/2} |\Psi_i(\mathbf{r})|^2, \tag{2.10}$$

do not perceive the local perturbations of the external potential $\Delta v(\mathbf{r}')$ if $\mathbf{r}'$ is far enough from $\mathbf{r}$. The relevant length scale has been introduced in Ref. [44] and is

$$\|\mathbf{r} - \mathbf{r}'\|_2 \gg \lambda, \tag{2.11}$$

where λ is the typical de Broglie wavelength occurring in the electronic ground state of a finite temperature ensemble. For example, for a finite temperature T, the de Broglie wavelength can be computed as

$$\lambda = \frac{h}{\sqrt{3 m_\mathrm{e} k_\mathrm{B} T}}, \tag{2.12}$$

where h is Planck's constant, m_e is the mass of an electron, and k_B is Boltzmann's constant. For $T = 300$ K one obtains $\lambda \approx 6.2$ nm.

A more rigorous analysis of the concept of the nearsightedness of electronic matter has been performed in Ref. [45], using the electron density defined in Equation (2.10). The authors of the latter have shown that for any perturbation of the external potential $\Delta v(\mathbf{r}')$ outside of the sphere of radius $\mathbf{r}_\mathrm{c}$, see Figure 2.1 for a 2D example, the following expression is valid

$$\lim_{\mathbf{r}_\mathrm{c} \to \infty} \overline{\Delta n}(\mathbf{r}_0, \mathbf{r}_\mathrm{c}) = 0, \tag{2.13}$$

for a broad class of systems. Here, $\overline{\Delta n}(\mathbf{r}_0, \mathbf{r}_\mathrm{c})$ is the maximum magnitude of the change in the electron density $\Delta n(\mathbf{r}_0)$ at $\mathbf{r}_0$ depending on the radius $\mathbf{r}_\mathrm{c}$ of the sphere in Figure 2.1. Moreover, for a given $\mathbf{r}_0$ and Δn, one can solve

$$\overline{\Delta n}(\mathbf{r}_0, \mathbf{r}_\mathrm{c}) = \Delta n \tag{2.14}$$

and obtain the nearsightedness range $\mathbf{r}_\mathrm{c}(\mathbf{r}_0, \Delta n)$. For technical details see the original publication [45].

In the context of quantum and computational chemistry, the principle of the nearsightedness of electronic matter can be exploited to explain the existence of first-principles methods whose computational cost scales linearly with the number of electrons N_el [44, 49–55], i.e. $\mathcal{O}(N_\mathrm{el})$. However, in this work, the nearsightedness of electronic matter is crucial to justify the decomposition of the electronic energy $E(\mathbf{R})$ into the respective atomic contributions $E_i(\mathbf{R})$ [31]

$$E(\mathbf{R}) = \sum_{i=1}^{N_\mathrm{at}} E_i(\mathbf{R}), \tag{2.15}$$

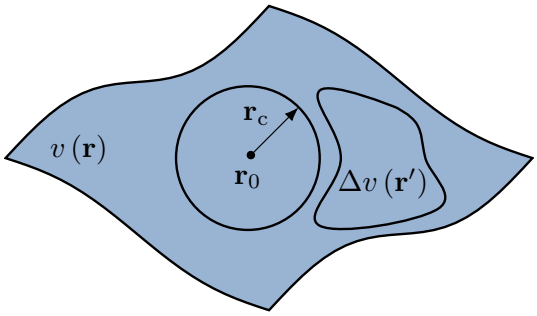


Figure 2.1: Schematic of the concept of the nearsightedness of electronic matter in 2D. $v\,(\mathbf{r})$ is the external potential in which electrons move, $\Delta v\,(\mathbf{r}')$ is the perturbation of the external potential at $\mathbf{r}'$. The perturbing potential is placed outside the sphere of radius $\mathbf{r}_\mathrm{c}$ centered at $\mathbf{r}_0$. (Reproduced with permission from Ref. [45]. Copyright (2005) National Academy of Sciences, U.S.A.)

where each atomic energy contains only local interactions within a finite cutoff radius. The selection of the cutoff radius appears to be crucial to take all relevant interactions into account since the nearsightedness range $\mathbf{r}_\mathrm{c}\,(\mathbf{r}_0, \Delta n)$ is sensitive to the system type, e.g., metallic vs. insulating [45].

2.3 Density functional theory

Probably the most frequently employed first-principles method for constructing the reference data for molecular machine learning (MML) algorithms is the density functional theory (DFT). This section gives a brief overview of the basics of DFT. The basis for DFT is the proof of the so-called Hohenberg–Kohn (HK) theorems [56], which state that the electronic ground state energy is determined entirely by the electron density n. The proof of HK theorems is straightforward and can be found in most textbooks for computational chemistry, see, e.g., Ref. [46].

The importance of the functional relationship between the ground state electronic energy and the electron density[2] is tremendous as it reduces the complexity of the problem from

[2]The electron density is given by the integral of the square of the wave function $\Psi\,(\mathbf{x}_1, \ldots \mathbf{x}_{N_\mathrm{el}})$ over $N_\mathrm{el} - 1$ electron coordinates $\mathbf{x} = (\mathbf{r}, \sigma)$

$$n\,(\mathbf{r}_1) = \int \mathrm{d}\sigma_1 \int \mathrm{d}\mathbf{x}_2 \cdots \mathrm{d}\mathbf{x}_{N_\mathrm{el}} \left|\Psi\,(\mathbf{x}_1, \ldots \mathbf{x}_{N_\mathrm{el}})\right|^2,$$

where σ denotes the spin coordinate. Thus, the electron density depends only on three spatial coordinates. Note that for each electron, three spatial and one spin coordinates are given.

dealing with $4N_{\mathrm{el}}$ degrees of freedom to dealing with only three spatial variables. Formally, one can split the energy functional $E[n]$ into three parts

$$E[n] = T[n] + E_{\mathrm{eN}}[n] + E_{\mathrm{ee}}[n], \tag{2.16}$$

with $T[n]$ being the kinetic energy, $E_{\mathrm{eN}}[n]$ being the attraction between the nuclei and electrons, and $E_{\mathrm{ee}}[n]$ being the electron–electron repulsion. The repulsion between nuclei is skipped in the above expression since it is constant in the Born–Oppenheimer (BO) approximation. In this section, atomic units (a.u.) are used to simplify the notation.[3] It is valid to split the electron–electron $E_{\mathrm{ee}}[n]$ repulsion into the classical Coulomb and a non-classical part, i.e., $J[n]$ and $K[n]$, respectively[4]

$$E_{\mathrm{ee}}[n] = J[n] + K[n]. \tag{2.17}$$

Unfortunately, there is no closed-form for the kinetic energy functional $T[n]$ and the non-classical part of the electron–electron repulsion $K[n]$, although many attempts toward their definition have been made, e.g., the Thomas–Fermi–Dirac model [57, 58]. The main flaw of these models is the poor representation of the kinetic energy functional, which, in general, is of the same order of magnitude as the total electronic energy. A solution has been proposed by Kohn and Sham [59] and is referred to as the Kohn–Sham (KS) theory. In the KS formalism, the kinetic energy functional is split into two parts, one which can be computed exactly and a small correction term.

The exact part of the kinetic energy functional is computed by assuming an auxiliary wave function Ψ_{KS} which yields the same density as the true, physical wave function. More specifically, in the KS framework, it is assumed that the electrons in this auxiliary system are non-interacting, and the solution to the electronic Schrödinger equation (SE) is given as a single

[3] All spatial coordinates are given in Bohr ($a_0 = 0.529 \times 10^{-10}$ m), while energies are in Hartree ($E_0 = 2625.5$ kJ mol^{-1}).

[4] The Coulomb term describes the classical average Coulomb potential between two charge distributions and reads

$$J[n] = \frac{1}{2}\int\int \frac{n(\mathbf{r})\, n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}',$$

while the non-classical part contains the potential correlation and exchange energy.

Slater determinant.[5] Thus, the exact part of the kinetic energy functional reads

$$T_{\mathrm{S}} = \sum_{i=1}^{N_{\mathrm{el}}} \langle \varphi_i | -\frac{1}{2}\nabla^2 | \varphi_i \rangle \,, \tag{2.18}$$

where the superscript S denotes that the kinetic energy has been calculated from the Slater determinant. The re-introducing of the so-called KS orbitals increases the complexity of the problem again to $3N_{\mathrm{el}}$ variables. Nonetheless, KS methods are independent-particle models and thus, are still far less computationally expensive than many-particle wave function-based methods [46].

Until now, no approximation has been made since the difference $T[n] - T_{\mathrm{S}}[n]$, the so-called kinetic correlation energy, together with the non-classical part $K[n] = E_{\mathrm{ee}}[n] - J[n]$ can be included in the exchange-correlation functional $E_{\mathrm{xc}}[n]$. The DFT energy reads

$$E_{\mathrm{DFT}}[n] = T_{\mathrm{S}}[n] + E_{\mathrm{eN}}[n] + J[n] + E_{\mathrm{xc}}[n] \,, \tag{2.19}$$

which is exact if the exact form of the exchange-correlation functional $E_{\mathrm{xc}}[n]$ is known. The latter has not been found so far and therefore, an approximation to it has to be derived. The advantage of the KS theory is that the exact part of the kinetic energy functional provides about ~ 99 % of the total kinetic energy $T[n]$ [46], i.e. the difference $T[n] - T_{\mathrm{S}}[n]$ is small. For example, for a neon atom, the kinetic energy is 128.9 a.u., the exchange energy is -12.1 a.u., and the correlation energy is only -0.4 a.u [46]. Therefore, one can immediately see that the KS theory is less sensitive to the exact form of the functionals as the exchange-correlation energy is about ten times smaller than the kinetic energy.

In the following, some popular classes of the exchange-correlation functionals, which build the basis for various DFT methods, are presented. It is common to separate an exchange-correlation functional into a pure exchange $E_{\mathrm{x}}[n]$ and a correlation $E_{\mathrm{c}}[n]$ contributions

$$E_{\mathrm{xc}}[n] = E_{\mathrm{x}}[n] + E_{\mathrm{c}}[n] \,. \tag{2.20}$$

[5] A Slater determinant is defined as

$$\Psi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{N_{\mathrm{el}}}) = \frac{1}{\sqrt{N_{\mathrm{el}}!}} \begin{vmatrix} \varphi_1(\mathbf{x}_1) & \varphi_2(\mathbf{x}_1) & \cdots & \varphi_{N_{\mathrm{el}}}(\mathbf{x}_1) \\ \varphi_1(\mathbf{x}_2) & \varphi_2(\mathbf{x}_2) & \cdots & \varphi_{N_{\mathrm{el}}}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{x}_{N_{\mathrm{el}}}) & \varphi_2(\mathbf{x}_{N_{\mathrm{el}}}) & \cdots & \varphi_{N_{\mathrm{el}}}(\mathbf{x}_{N_{\mathrm{el}}}) \end{vmatrix},$$

where φ_i is a monoelectronic wave function or the so-called (molecular) orbital.

One of the simplest approximations is the so-called local density approximation (LDA). Here, the local density is treated as a uniform electronic gas and the exchange-correlation functional is given by

$$E_{\mathrm{xc}}^{\mathrm{LDA}}[n] = E_{\mathrm{x}}^{\mathrm{LDA}}[n] + E_{\mathrm{c}}^{\mathrm{LDA}}[n], \tag{2.21}$$

with the exchange part given by a simple analytic expression [57, 58]

$$E_{\mathrm{x}}^{\mathrm{LDA}}[n] = -C_{\mathrm{x}} \int n^{4/3}(\mathbf{r})\, \mathrm{d}\mathbf{r}, \tag{2.22}$$

where $C_{\mathrm{x}} = \frac{3}{4}(3/\pi)^{1/3}$, and $E_{\mathrm{c}}^{\mathrm{LDA}}[n]$ is computed typically by employing parameterized fits to quantum Monte Carlo data for the uniform electron gas. Note that an extension to LDA exists for the open-shell systems called the local spin density approximation (LSDA). The LDA and LSDA approximations are especially suitable for extended metallic systems, and, therefore, they have been used extensively in the physics community. Unfortunately, these approximations are not well suited for molecular systems, underestimating the exchange energy by ~ 10 % [46].

One can achieve an improvement over LDA and LSDA by taking the gradient of the electron density $\nabla n(\mathbf{r})$ into account in the functional form. These methods are referred to as generalized gradient approximation (GGA) methods which have a general form

$$\begin{aligned} E_{\mathrm{xc}}^{\mathrm{GGA}}[n] &= E_{\mathrm{x}}^{\mathrm{GGA}}[n] + E_{\mathrm{c}}^{\mathrm{GGA}}[n] \\ &= \int f_{\mathrm{x}}(n(\mathbf{r}), \nabla n(\mathbf{r}))\, \mathrm{d}\mathbf{r} + \int f_{\mathrm{c}}(n(\mathbf{r}), \nabla n(\mathbf{r}))\, \mathrm{d}\mathbf{r}, \end{aligned} \tag{2.23}$$

where f_{x} and f_{c} are the functions of $n(\mathbf{r})$ and $\nabla n(\mathbf{r})$. Many examples of GGA functionals typically used in the literature exist, e.g., the B88 [60] and the PBE [61, 62] functionals. However, it is out of this work's scope to provide the exact analytical form of the exchange-correlation functionals mentioned above. For more details, the reader is referred to, e.g., Refs. [46, 63].

A possibility to further improve the accuracy of the exchange-correlation functional is to combine the GGA functionals with the exact Hartree–Fock (HF) exchange.[6] Such functionals are referred to as hybrid functionals. For example, from the PBE functional, one can define its

[6] The exact HF exchange is defined as

$$E_{\mathrm{x}}^{\mathrm{HF}}[n] = -\frac{1}{2}\sum_{i,j} \int\int \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \varphi_i^*(\mathbf{r})\, \varphi_j(\mathbf{r})\, \frac{1}{|\mathbf{r}-\mathbf{r}'|} \varphi_i(\mathbf{r}')\, \varphi_j^*(\mathbf{r}'),$$

where φ_i are the KS orbitals.

hybrid version PBE0 [62, 64] as

$$E_{\mathrm{xc}}^{\mathrm{PBE0}}[n] = \frac{1}{4}E_{\mathrm{x}}^{\mathrm{HF}}[n] + \frac{3}{4}E_{\mathrm{x}}^{\mathrm{PBE}}[n] + E_{\mathrm{c}}^{\mathrm{PBE}}[n], \tag{2.24}$$

such that the exchange part is a linear combination of $E_{\mathrm{x}}^{\mathrm{HF}}$ and $E_{\mathrm{x}}^{\mathrm{PBE}}$. Note that more examples of successful hybrid functionals exist, like B3LYP [65, 66]. However, a detailed overview is out of the scope of this work, and the reader is referred elsewhere [46, 63].

Finally, the computational efficiency of DFT has promoted its broad applicability in many areas of chemistry, physics and materials science. Specifically, DFT allows for the evaluation of energies and their gradients for many configurations and, therefore, allows one to study the time evolution of an atomistic system. DFT-based molecular dynamics (MD) methods (Car–Parrinello (CP) [67] and Born–Oppenheimer (BO) MD) are often referred to as ab-initio molecular dynamics (AIMD) and provide a means for studying chemical processes in an accurate and unbiased way [68–70]. Nevertheless, the length and time scales accessible with DFT are still rather limited. Machine learning (ML)-based models promise to expand the applicability of first-principle methods, in general, to larger time and length scales [5–7, 71–77].

3 Artificial neural networks

Neural network (NN) based machine learning (ML) models have achieved extraordinary performance across a wide range of tasks [78–83]. Their success has motivated the use of NNs beyond image recognition and text understanding in, for example, chemistry and physics [17–27]. Therefore, in this chapter, a bottom-up introduction to NNs is presented. Starting at the basics of NNs, Section 3.1 explains the general network architecture, the forward pass, the error back-propagation, and gradient descent (GD) for training an NN-based model. Next, Section 3.2 introduces several heuristics and best practices to improve the training of NNs and provides a Hessian-based analysis of them. Lastly, Section 3.3 discusses the generalization ability of NNs and introduces several techniques to improve it. Sections 3.1–3.3 were inspired by Refs. [84–87].

3.1 The basics of neural networks

This section elaborates on artificial feed-forward neural networks (NNs). The starting point is the general formulation of artificial NNs, the description of their architecture, and the computation of the network's output in the forward propagation step; see Section 3.1.1. Next, Section 3.1.2 discusses the calculation of gradients of the predefined loss function with respect to trainable parameters and introduces the backward propagation algorithm. From here, the basics of the gradient descent (GD) based optimization of NNs is described, and the convergence analysis of the respective algorithm is performed; see Section 3.1.3.

3.1.1 Network architecture and forward propagation

Let us consider a function defined at an abstract level as

$$f\left(\cdot;\boldsymbol{\theta}\right):\mathbb{R}^{d}\rightarrow\mathbb{R}, \tag{3.1}$$

which takes an input $\mathbf{x}\in\mathbb{R}^{d}$ and maps it to an output $f\left(\mathbf{x};\boldsymbol{\theta}\right)\in\mathbb{R}$. Note that only a single output is considered for simplicity, but the generalization to multiple outputs is straightforward.

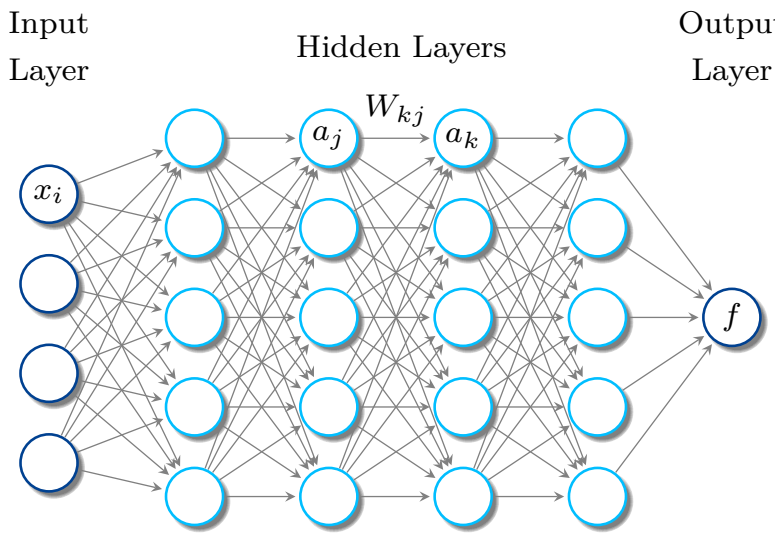


Figure 3.1: Schematic representation of a feed-forward neural network (NN) consisting of four hidden layers. In a feed-forward NN, the information propagates from the inputs through multiple hidden layers to the output.

The parameters $\boldsymbol{\theta}$ are learned from the data such that the mapping in Equation (3.1) fits the target values y. A real-valued function $f(\cdot; \boldsymbol{\theta})$ is referred to as a neural network (NN) if it is structured as a composition of many simple neurons organized in layers. Each neuron receives the output of the neurons from the previous layer as an input.

A simple scheme of a fully-connected feed-forward NN is presented in Figure 3.1. The final layer provides the output of a feed-forward NN $f(\mathbf{x}; \boldsymbol{\theta})$ and is called the output layer; $\mathbf{x}$ is the input to the NN. All other layers are called hidden layers as the training data does not show the output of these layers. One may compute the output of a neuron k in Figure 3.1 from inputs a_j in two steps

$$\begin{cases} h_k = \sum_j W_{kj} a_j + b_k, \\ a_k = \varphi(h_k), \end{cases} \tag{3.2}$$

where W_{kj} and b_k are trainable parameters of the NN and are referred to as weights and biases, respectively. The output of the first step h_k represents a weighted sum over all neurons j that neuron k receives as input. It is often referred to as a pre-activation. The second step applies an activation function $\varphi : \mathbb{R} \to \mathbb{R}$ to the pre-activation h_k yielding a result known as a post-activation. Common choices of activation functions are described at the end of this section.

In the following, let a fully-connected feed-forward NN with L hidden layers of width n_l, for $l = 0, \ldots, L+1$ be defined by the recurrence relation

$$\begin{cases} \mathbf{a}^{(0)} = \mathbf{x} \in \mathbb{R}^d, \\ \mathbf{h}^{(l+1)} = \mathbf{W}^{(l+1)}\mathbf{a}^{(l)} + \mathbf{b}^{(l+1)} \in \mathbb{R}^{n_{l+1}}, \\ \mathbf{a}^{(l+1)} = \varphi(\mathbf{h}^{(l+1)}) \in \mathbb{R}^{n_{l+1}}, \end{cases} \tag{3.3}$$

where $\varphi : \mathbb{R} \to \mathbb{R}$ is the activation function applied element-wise to an input $\mathbf{h}^{(l+1)}$, $\mathbf{W}^{(l+1)} \in \mathbb{R}^{n_{l+1} \times n_l}$ and $\mathbf{b}^{(l+1)} \in \mathbb{R}^{n_{l+1}}$ are weights and biases. Here, $n_0 = d$ and $n_{L+1} = 1$ are the input and output dimensions, respectively. Moreover, the activation for the input neurons is defined as $\mathbf{a}^{(0)} = \mathbf{x}$ and the output of the network is given by the linear output unit, i.e. $f(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{h}^{(L+1)} \in \mathbb{R}$, since throughout this work, a regression or interpolation task is considered. In general, other output units depending on the specific task are possible, see, e.g., Ref. [85].

When using feed-forward NNs, the information propagates from the input $\mathbf{x}$ through the hidden layers to an output f; see Figure 3.1. It is called the forward propagation step and is an important part of the NN training as it allows one to compute a scalar loss $\mathcal{L}(\boldsymbol{\theta})$; see the following discussion. The gradient is computed in the back-propagation step; see Section 3.1.2. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}} \subset \mathbb{R}^d \times \mathbb{R}$ denote the training set with inputs $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$. In this work, the task of regression is principally considered, thereby typically the mean squared error (MSE) loss function

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \mathbb{E}_{\mathbf{x}, y \in \mathcal{D}} \left[\| f(\mathbf{x}; \boldsymbol{\theta}) - y \|_2^2 \right], \tag{3.4}$$

will be minimized during training to attempt to find optimal trainable parameters

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \, \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}), \tag{3.5}$$

where $\boldsymbol{\theta}^*$ denotes the optimal parameters.

Note that different loss functions give various statistics. Minimizing the MSE loss on infinitely many samples drawn from the true distribution would result in an NN which predicts

the mean of y for a given $\mathbf{x}$[1]

$$f\left(\mathbf{x};\boldsymbol{\theta}^{*}\right)=\mathbb{E}_{y\in\mathcal{Y}}\left[y|\,\mathbf{x}\right], \tag{3.6}$$

while minimizing the mean absolute error (MAE) loss

$$\mathcal{L}\left(\boldsymbol{\theta};\mathcal{D}\right)=\mathbb{E}_{\mathbf{x},y\in\mathcal{D}}\left[\left\|f\left(\mathbf{x};\boldsymbol{\theta}\right)-y\right\|_{1}\right], \tag{3.7}$$

would yield a function that predicts the median value of y for a given $\mathbf{x}$. One can obtain these results by employing the calculus of variations; see Ref. [85] for more details.

Another important aspect of a successful NN architecture is the properly chosen activation function $\varphi:\mathbb{R}\rightarrow\mathbb{R}$, which gives an NN its non-linear capabilities. Here, four different popular choices are considered: ReLU [88, 89], softplus [88, 90, 91], sigmoid [92], and Swish [93–95]. The rectifier, or ReLU, activation function is defined by

$$\mathrm{ReLU}\left(h_{j}\right)=\max\left(0,h_{j}\right), \tag{3.8}$$

and is one-sided, i.e. does not enforce a sign symmetry or anti-symmetry [88].[2] Softplus is a smooth version of the rectifying non-linearity and is defined by

$$\mathrm{softplus}\left(h_{j}\right)=\ln\left(1+\exp\left(h_{j}\right)\right), \tag{3.9}$$

which has the advantage of being smooth around zero.

Another activation function class is the logistic sigmoid and the hyperbolic tangent, which

[1] One can obtain this result by employing the calculus of variations. For a squared loss function $\mathcal{L}\left(\mathbf{x},y\right)=\left(f\left(\mathbf{x}\right)-y\right)^{2}$, the average expected loss is given by

$$\mathbb{E}\left[\mathcal{L}\right]=\int\int\left(f\left(\mathbf{x}\right)-y\right)^{2}p\left(\mathbf{x},y\right)\mathrm{d}\mathbf{x}\mathrm{d}y,$$

where $p\left(\mathbf{x},y\right)$ is the joint probability distribution. To find the optimal $f^{*}\left(\mathbf{x}\right)$, the functional derivative with respect to $f\left(\mathbf{x}\right)$ is computed and reads

$$\frac{\delta\mathbb{E}\left[\mathcal{L}\right]}{\delta f\left(\mathbf{x}\right)}=2\int\left(f\left(\mathbf{x}\right)-y\right)p\left(\mathbf{x},y\right)\mathrm{d}y=0.$$

Using that $p\left(\mathbf{x},y\right)=p\left(\mathbf{x}\right)p\left(y|\,\mathbf{x}\right)$ one obtains

$$f^{*}\left(\mathbf{x}\right)=\int yp\left(y|\,\mathbf{x}\right)\mathrm{d}y=\mathbb{E}_{y}\left[y|\,\mathbf{x}\right].$$

[2] For example, the hyperbolic tangent $\tanh\left(x\right)$ imposes the sign anti-symmetry, while its absolute value [96] imposes the sign symmetry.

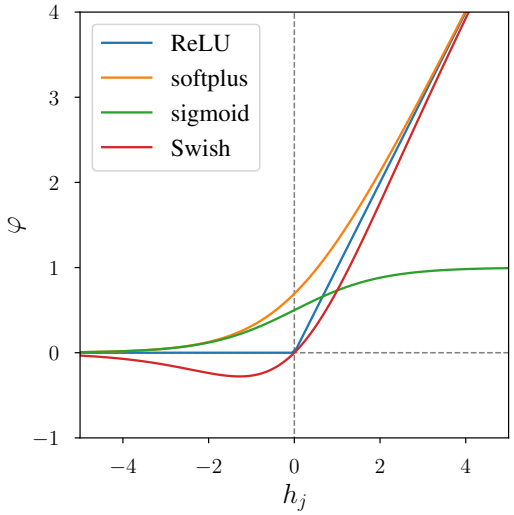


Figure 3.2: Common choices of activation functions: ReLU [88, 89], softplus [88, 90, 91], sigmoid [92], and Swish [93–95].

are equivalent to a linear transformation. Therefore, in the following discussion, only the former is considered. The sigmoid non-linearity has the following analytical form

$$\text{sigmoid}\left(h_j\right) = \frac{1}{1+\exp\left(-h_j\right)}, \tag{3.10}$$

and is a monotonically increasing function which asymptotes at some finite value as $x \to \pm\infty$. The last activation function studied here is the Swish non-linearity given by

$$\text{Swish}\left(h_j\right) = h_j \cdot \text{sigmoid}\left(h_j\right), \tag{3.11}$$

and is equivalent to the ReLU non-linearity as $h_j \to \pm\infty$. Figure 3.2 shows all activation functions described in this section.

3.1.2 Computing gradients via the backward propagation

During a training step, the parameters of a neural network (NN), $\boldsymbol{\theta} = \left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L+1)}, \mathbf{b}^{(L+1)}\right)$, are optimized in order to minimize the loss function $\mathcal{L}\left(\boldsymbol{\theta}\right)$, e.g. the squared loss defined in Equation (3.4). For the minimization of the latter, most algorithms require the computation of its gradient with respect to all parameters, i.e. $\partial\mathcal{L}\left(\boldsymbol{\theta}\right)/\partial\boldsymbol{\theta}$. Here, an algorithm called error back-propagation [97, 98] is outlined, following Ref. [86], which allows for the efficient propagation of the error feedback to the multiple

layers of the NN to compute the gradient with respect to the weights and biases that need to be adjusted.

An error back-propagation algorithm is essentially the consecutive application of the chain rule to the loss function $\mathcal{L}(\boldsymbol{\theta})$. In the following, the NN is re-written in its equivalent form as

$$f(\mathbf{x};\boldsymbol{\theta}) = \mathbf{W}^{(L+1)}\varphi\left(\mathbf{W}^{(L)}\ldots\varphi\left(\mathbf{W}^{(2)}\varphi\left(\mathbf{W}^{(1)}\mathbf{x}\right)\right)\right), \tag{3.12}$$

where biases $\mathbf{b}^{(l)}$ are omitted and $\boldsymbol{\theta} = \left(\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(L+1)}\right)$, for simplicity. For the simple squared error loss

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\left\|f(\mathbf{x};\boldsymbol{\theta}) - y\right\|_2^2, \tag{3.13}$$

where $f(\mathbf{x};\boldsymbol{\theta})$ is the output of the network and y is the corresponding label, one obtains for the gradient with respect to the weight matrix $\mathbf{W}^{(l)}$ the following expression[3]

$$\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{W}^{(l)}} = \left(\mathbf{W}^{(L+1)}\mathbf{D}^{(L)}\ldots\mathbf{W}^{(l+1)}\mathbf{D}^{(l)}\right)^T\left(f(\mathbf{x};\boldsymbol{\theta}) - y\right)\left(\mathbf{a}^{(l-1)}\right)^T. \tag{3.14}$$

Here, $\mathbf{D}^{(l)} = \operatorname{diag}\left(\varphi'\left(\mathbf{h}^{(l)}\right)\right)$ denotes the diagonal matrix with elements consisting of the derivatives of the post-activation with respect to $\mathbf{h}^{(l)}$.

Given the expression for $\partial\mathcal{L}(\boldsymbol{\theta})/\partial\mathbf{W}^{(l)}$, a sequence of the back-propagated error can be defined by the recurrence relation

$$\begin{aligned}
\boldsymbol{\delta}^{(L+1)} &= \left(f(\mathbf{x};\boldsymbol{\theta}) - y\right),\\
\boldsymbol{\delta}^{(L)} &= \left(\mathbf{W}^{(L+1)}\mathbf{D}^{(L)}\right)^T\boldsymbol{\delta}^{(L+1)},\\
&\cdots,\\
\boldsymbol{\delta}^{(l)} &= \left(\mathbf{W}^{(l+1)}\mathbf{D}^{(l)}\right)^T\boldsymbol{\delta}^{(l+1)},\\
&\cdots,\\
\boldsymbol{\delta}^{(1)} &= \left(\mathbf{W}^{(2)}\mathbf{D}^{(1)}\right)^T\boldsymbol{\delta}^{(2)},
\end{aligned}$$

which simplifies the computation of the gradient of the loss function with respect to the weight

[3]Understanding the error back-propagation algorithm requires considering the application of the chain rule to obtain derivatives to the neural networks (NNs) graphs. Let $f:\mathbb{R}\to\mathbb{R}$ denote a general real-valued function. Given the chain of functions $x = f(w)$, $y = f(x)$, and $z = f(y)$ one may write for $\partial z/\partial w$

$$\frac{\partial z}{\partial w} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w} = f'(y)\,f'(x)\,f'(w).$$

matrix $\mathbf{W}^{(l)}$ to

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \mathbf{W}^{(l)}} = \boldsymbol{\delta}^{(l)} \left(\mathbf{a}^{(l-1)}\right)^T. \tag{3.15}$$

An efficient implementation of the back-propagation algorithm stores $\mathbf{a}^{(l)}$ during the forward pass. During the backward pass, $\boldsymbol{\delta}^{(l)}$ is computed by the left-multiplying of $\boldsymbol{\delta}^{(l+1)}$ by the matrix $\left(\mathbf{W}^{(l+1)}\mathbf{D}^{(l)}\right)^T$. All vectors $\boldsymbol{\delta}^{(l)}$ are stored as well. Here, only a single output has been considered, $f(\mathbf{x};\boldsymbol{\theta}) \in \mathbb{R}$, but the generalization to multiple outputs is straightforward.

As a final remark of this section, one should mention that in practice, modern NN frameworks such as, e.g., TensorFlow [99], Pytorch [100], and JAX [101] use automatic differentiation [102], where the backward computations are produced automatically from the forward pass. Therefore, there is no need to implement the error back-propagation manually.

3.1.3 The basics of gradient descent and convergence analysis

In previous sections, the forward and backward propagation through the network, which provides pre-activations $\mathbf{a}$ and gradients $\partial\mathcal{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$, have been discussed. Thus, the main focus now is the problem of training an NN, i.e. the optimization of the trainable parameters from the data. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}} \subset \mathbb{R}^d \times \mathbb{R}$ denote the training set with inputs $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$. A large class of methods for neural network optimization are based on gradient descent (GD), which, in its basic form, reads

$$\Delta\boldsymbol{\theta} = -\alpha \frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{3.16}$$

where $\mathcal{L}(\boldsymbol{\theta})$ is the loss function, e.g., the mean squared error (MSE) loss given by

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \|f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i\|_2^2 \tag{3.17}$$

and $\alpha > 0$ is the learning rate.

In the following, the shape of the loss function close to the optimal trainable parameter $\boldsymbol{\theta}^*$ is analyzed to assess the convergence of gradient descent, following Refs. [84, 87]. For this purpose, the loss function is expanded into its Taylor series around $\boldsymbol{\theta}^*$

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}^*) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T \left.\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^2}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} (\boldsymbol{\theta} - \boldsymbol{\theta}^*) + \cdots. \tag{3.18}$$

Note that in the above expression, there is no first-order term $(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^T \partial\mathcal{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$, be-

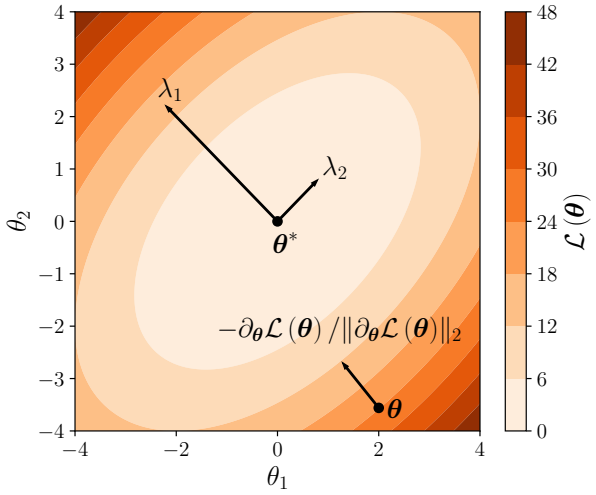


Figure 3.3: Contour plot of a loss function with $\boldsymbol{\theta} \in \mathbb{R}^2$. The highest eigenvalues correspond to the direction of the highest curvature. A large ratio λ_1/λ_2 makes the function ill-conditioned and harder to optimize with gradient descent.

cause $\boldsymbol{\theta}^*$ is defined to be a minimum, where the gradient vanishes. The eigenvalues of the Hessian matrix $\partial^2\mathcal{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}^2\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}$, namely $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_{|\boldsymbol{\theta}|} \geq 0$, provide some information about the local curvature of the loss function and how well GD may perform. Let $|\boldsymbol{\theta}|$ denote the dimension of the parameter space, i.e. the number of trainable parameters. Assuming all eigenvalues to be strictly positive, the difficulty to converge to $\boldsymbol{\theta}^*$ is given by the condition number $\lambda_1/\lambda_{|\boldsymbol{\theta}|}$. An example of a two-dimensional loss function with $\boldsymbol{\theta} \in \mathbb{R}^2$ and the respective eigenvalues is presented in Figure 3.3.

In general, the Hessian matrix of an NN can be computed by applying the chain rule for second derivatives [84]

$$\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}^2} = \frac{\partial\mathbf{O}}{\partial\boldsymbol{\theta}}^T \frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{O}^2}\frac{\partial\mathbf{O}}{\partial\boldsymbol{\theta}} + \frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\mathbf{O}}\frac{\partial^2\mathbf{O}}{\partial\boldsymbol{\theta}^2}, \tag{3.19}$$

where $\mathbf{O} = f(\mathcal{X};\boldsymbol{\theta})$. While the analysis of the high-dimensional Hessian matrix in Equation (3.19) is hardly feasible, blocks of it may have a simpler analytical form [103, 104]. Let $\langle\cdot\rangle_{\mathcal{D}}$ denote the mean over the training data, $g_k = \partial f/\partial h_k$ be the sensitivity of the neuron k to the output, and $h_k = \sum_j W_{kj}a_j$ be the output of a simple neuron. Then, the block Hessian $(\mathbf{H}_k)_{ij} = \partial^2\mathcal{L}(\boldsymbol{\theta})/\partial W_{ki}\partial W_{kj}$ focuses on the parameters of neuron k and takes the simple

form[4]

$$(\mathbf{H}_k)_{ij} = \left\langle a_i a_j g_k^2 \right\rangle_{\mathcal{D}} + \left\langle a_i \frac{\partial g_k}{\partial W_{kj}} \left(f\left(\mathbf{x}; \boldsymbol{\theta}\right) - y \right) \right\rangle_{\mathcal{D}} . \tag{3.20}$$

As a side remark, it should be mentioned that the Hessian-based analysis presented in this section will be used in the following to motivate and explain a variety of best practices and techniques used to improve the training and the generalization ability of artificial NNs.

3.2 Improving the training of neural networks

Here, several heuristics and best practices to improve the training of artificial neural networks (NNs) are presented. First, two techniques, often used in the literature to normalize the input features and inputs to hidden and output layers, standardization and batch normalization (BN), are described; see Section 3.2.1. Then, in Section 3.2.2, the selection of an appropriate activation function, defining the non-linear capability of NNs, is discussed. Particular attention in Section 3.2.3 is drawn to the parameterization of artificial NNs, which also affects the selection of learning rates and the overall performance. Finally, the key concepts of state-of-the-art optimization algorithms are presented; see Section 3.2.4.

3.2.1 Normalizing the input features

In general, a faster convergence of a neural network (NN) based model can be achieved if the average of each input variable over the training set is close to zero. To see this simple relationship, one may refer to the deductive example in Ref. [84], which considers a simple case where all the inputs are positive. It can be shown that, in this case, all updates of the trainable parameters of the first layer will have the same sign. Thus, the respective weight parameters can only all decrease or increase for a given input, and a weight vector can change its direction, if needed, only by zigzagging, which may lead to slow convergence.

In the above example, only positive inputs have been considered. However, slow convergence can, in general, be caused by any shift of the average input away from zero since it would bias the updates of the trainable parameters in a particular direction. It holds not only for the input features but for all layers in an NN. Therefore, the batch normalization (BN) technique [105, 106] is described at the end of this section, and is particularly important when using mini-batch training. See Section 3.2.4 for more details on mini-batch training.

[4] It has been used that $\partial f / \partial W_{kj} = a_j \partial f / \partial h_k$.

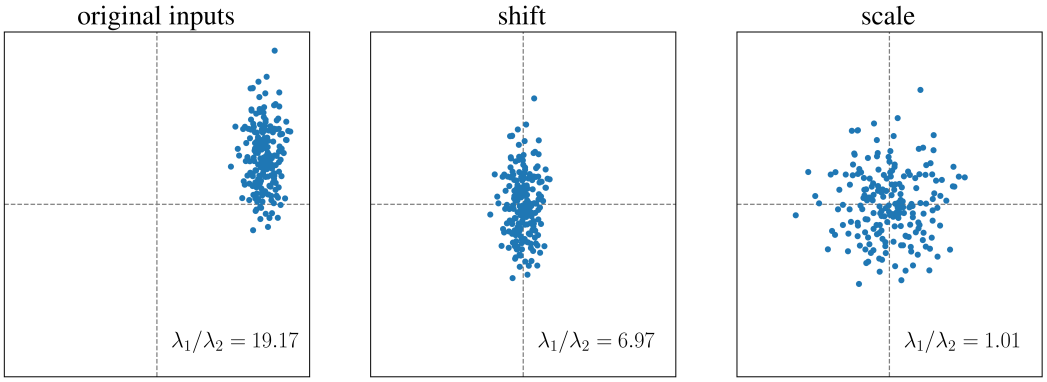


Figure 3.4: Distribution of inputs $\mathbf{x} \in \mathbb{R}^2$ and $\tilde{\mathbf{x}} \in \mathbb{R}^2$ after shifting and scaling the inputs, respectively. Condition number λ_1/λ_2 is shown as an inset. A low condition number makes the optimization easier.

Besides shifting the inputs, they can be scaled such that all of them have the same variance

$$\sigma_i^2 = \frac{1}{N_{\text{Train}}} \sum_{k=1}^{N_{\text{Train}}} \left(x_i^{(k)}\right)^2, \tag{3.21}$$

where N_{Train} is the number of training points, and σ_i^2 is the covariance of the ith input variable and $x_i^{(k)}$ is the ith component of the kth training sample. Scaling the inputs may lead to a balanced rate at which all weight parameters learn. It improves the overall convergence of the gradient descent (GD) algorithm. Additionally, by imposing $\sigma_i = 1$, one may ensure that reasonable regions of the activation function, see Section 3.2.2, are used, which may improve GD's convergence.

As a final remark, the Hessian-based analysis can be used to motivate the scaling and shifting of inputs. Considering that the inputs x_i, x_j, and the sensitivity g_k are independent, one may obtain for the Hessian matrix $\mathbf{H}_k \approx \langle \mathbf{x} \otimes \mathbf{x} g_k^2 \rangle_{\mathcal{D}}$ an upper-bound for the condition number $\lambda_1/\lambda_{|\boldsymbol{\theta}|} = \left(\max_i \sigma_i^2 + \|\mathbf{m}\|_2^2\right) / \min_i \sigma_i^2$ with $\mathbf{m} \in \mathbb{R}^d$ being the shift vector [87]. From this expression, one can see that any significant shift away from zero and a large spread in standard deviations may increase in the condition number and, thus, impose a slower convergence of the optimization of an NN-based model.

Standardization technique. In the following, the scaling and shifting of the inputs are combined to the technique often referred to as standardization. The resulting transformation is given by

$$\tilde{x_i} = \frac{x_i - m_i}{\sigma_i}, \tag{3.22}$$

for all input features $i = 1, \ldots, d$. Note that throughout this section, the input features are considered, while the same techniques may be applied to hidden layers and, especially, to the network's outputs. It also may have a positive impact on the convergence rate. An example of the standardization technique is presented in Figure 3.4 for $\mathbf{x} \in \mathbb{R}^2$. It demonstrates that centring and normalizing the inputs leads to a considerably reduced condition number ($\lambda_1/\lambda_2 = 1.01$) compared to the original inputs ($\lambda_1/\lambda_2 = 19.17$).

Batch normalization technique. Here, a brief introduction to the batch normalization (BN) technique is given. Note that BN is extensively used in literature to achieve faster training processes and better resulting models [106]. In practice, BN is implemented as an additional NN layer. In each iteration of the training process, it normalizes the inputs of the latter using the mean and the variance of the input batch. It is a beneficial technique for normalizing the inputs to hidden layers and, e.g., normalizing trainable input features [3].

In a standard BN approach, for a training step, one computes[5]

$$\mathbf{a}^{\mathrm{norm}} = \frac{\mathbf{a} - \boldsymbol{\mu}_b}{\sqrt{\boldsymbol{\sigma}_b^2 + \epsilon}}, \tag{3.23}$$

where BN is directly applied to the post-activation $\mathbf{a}$. Mean and variance of the mini-batch b are given by $\boldsymbol{\mu}_b$ and $\boldsymbol{\sigma}_b^2$, respectively. A small constant ϵ is added to the variance for numerical stability. Additionally, in each training step, the running mean and variance, which are used during the inference step, are updated according to

$$\begin{aligned} \boldsymbol{\mu} &= \gamma \boldsymbol{\mu} + (1 - \gamma)\, \boldsymbol{\mu}_b, \\ \boldsymbol{\sigma}^2 &= \gamma \boldsymbol{\sigma}^2 + (1 - \gamma)\, \boldsymbol{\sigma}_b^2, \end{aligned} \tag{3.24}$$

and the output of an BN layer at the inference step reads

$$\mathbf{a}^{\mathrm{norm}} = \frac{\mathbf{a} - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}}. \tag{3.25}$$

Typically, one must account for the bias introduced by initializing $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ to zeros. For this purpose, the respective values can be re-scaled by $s_b = 1/\left(1 - t_b + \epsilon\right)$ with ϵ being a small constant, $t_b = \gamma t_{b-1}$, and $t_0 = 1$, i.e. $\boldsymbol{\mu} \to s_b \boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2 \to s_b \boldsymbol{\sigma}^2$. Finally, the normalized value

[5] In Ref. [3], the authors observed that the averaged statistics in Equation (3.24) could also be advantageous during training for input features. It implies that the corresponding expression must be changed accordingly and is equivalent to Equation (3.25).

has to be shifted and scaled to restore the representational power of the network [105]

$$\tilde{\mathbf{a}} = \mathbf{c} \circ \mathbf{a}^{\mathrm{norm}} + \mathbf{d}, \tag{3.26}$$

where $\circ$ denotes the element-wise product, and parameters $\mathbf{c}$ and $\mathbf{d}$ are optimized during training.

3.2.2 Choosing the activation function

In the following, the Hessian-based analysis presented in Section 3.1.3 is employed to motivate the application of the activation functions described in Section 3.1.1. The discussion presented here follows Ref. [87]. Let $\mathbf{h}$ denote the pre-activations and $\mathbf{a} = \varphi(\mathbf{h})$ be the vector of post-activations received by neuron k. The respective Hessian matrix is given by $\mathbf{H}_k \approx \langle \mathbf{a} \otimes \mathbf{a} g_k^2 \rangle_{\mathcal{D}}$, where the second term of the Hessian matrix from Equation (3.20) is neglected. Moreover, the sensitivity g_k is assumed to be unity, for simplicity.

Figure 3.5 shows the distribution of pre-activations $\mathbf{h} \in \mathbb{R}^2$ and post-activations $\mathbf{a} = \varphi(\mathbf{h}) \in \mathbb{R}^2$ obtained from non-linearities in Equations (3.8–3.11) with corresponding condition number λ_1/λ_2. The application of the ReLU function results in a relatively low condition number of 1.91, which explains its broad applicability in the literature [79, 80, 107]. However, an important requirement when modeling physicochemical properties is smoothness, clearly violated by the ReLU function at zero. Therefore, in addition to the ReLU non-linearity, the softplus, sigmoid, and Swish activation functions are considered. While softplus and sigmoid tend to produce the post-activations that are not well centered, causing the conditions number to be high (38.58 and 80.67, respectively), the Swish activation function is well centered. It produces a condition number close to one (1.08).[6] Therefore, the Swish activation should be prioritized while constructing NNs employed for modeling potential energy surfaces and other physicochemical properties.

3.2.3 Parameter initialization and learning rates

Careful initialization of trainable parameters is crucial for the efficient training of NNs. They have to be initialized randomly to break artificial symmetries in the parameter space but in such a way that the reasonable part of the activation function can still be used. Additionally, it is preferable that all layers learn equally fast, i.e. all gradients are of the same magnitude. In

[6] Similar results as for the Swish non-linearity may be obtained for the shifted softplus $\ln(1 + \exp(h_j)) - \ln(2)$.

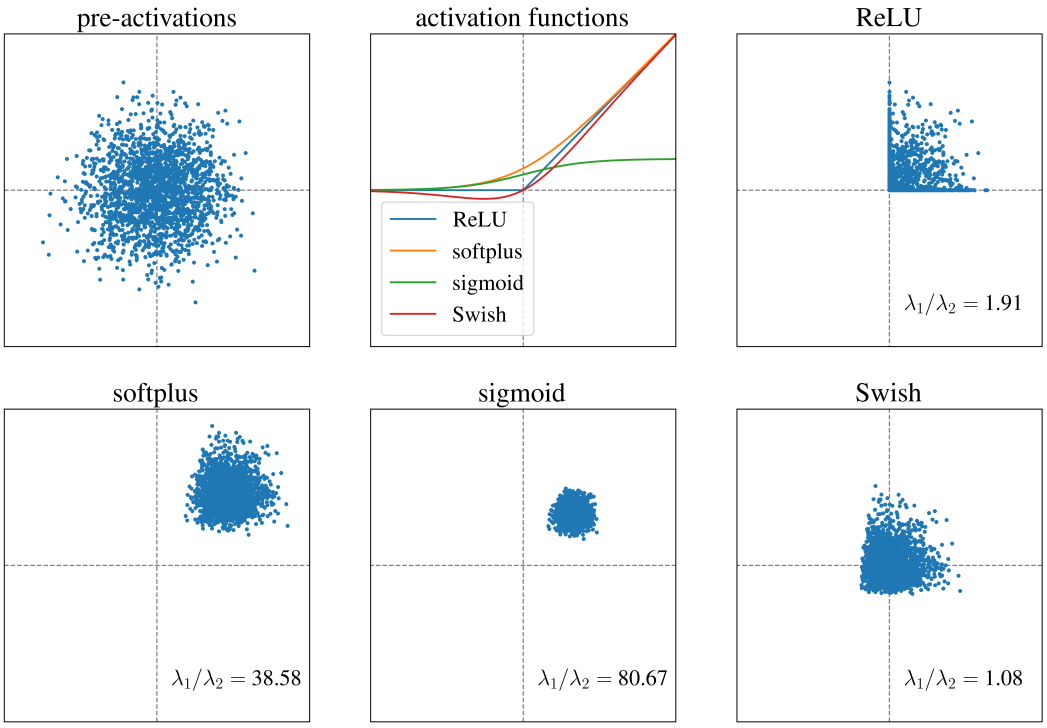


Figure 3.5: Distribution of pre-activations $\mathbf{h} \in \mathbb{R}^2$ and activations $\mathbf{a} \in \mathbb{R}^2$ resulting from the application of different non-linearities. Condition number λ_1/λ_2 is shown as an inset. A low condition number makes the optimization easier. (Reproduced with permission from Ref. [87]. Copyright 2020, The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG.)

the following, a network is again defined by a recursion

$$\begin{cases} \mathbf{a}^{(0)} = \mathbf{x} \in \mathbb{R}^d, \\ \mathbf{h}^{(l+1)} = \mathbf{W}^{(l+1)}\mathbf{a}^{(l)} + \mathbf{b}^{(l+1)} \in \mathbb{R}^{n_{l+1}}, \\ \mathbf{a}^{(l+1)} = \varphi\left(\mathbf{h}^{(l+1)}\right) \in \mathbb{R}^{n_{l+1}}, \end{cases} \tag{3.27}$$

where $\mathbf{a}^{(l+1)}$ is called the post-activation and $\mathbf{h}^{(l+1)}$ is the pre-activation. The non-linearity $\varphi : \mathbb{R} \to \mathbb{R}$ is applied entry-wise. The standard parameterization (SP) is achieved by taking the initial weight and bias elements from $W_{ij}^{(l+1)} \sim \mathcal{N}\left(0, 1/n_l\right)$ and $b_j^{(l+1)} \sim \mathcal{N}\left(0, 1\right)$, where $\mathcal{N}\left(\mu, \sigma^2\right)$ is a Gaussian distribution with a mean μ and variance σ^2. Sometimes biases are initialized at zeros, i.e. $b_j^{(l+1)} = 0$. This parameterization is sometimes referred to as LeCun [84] or Kaiming initialization [108].

Similar to the previous sections, SP can be motivated by the Hessian-based analysis [87].

Recall from Section 3.1.3 that the Hessian matrix of the network can be approximated by a block-diagonal form

$$\mathbf{H} \approx \mathrm{diag}\left(\mathbf{H}_j, \mathbf{H}_{j'}, \mathbf{H}_{j''}, \ldots, \mathbf{H}_k, \mathbf{H}_{k'}, \mathbf{H}_{k''}, \ldots\right), \tag{3.28}$$

and eigenvalues of $\mathbf{H}$ are given by the eigenvalues of the respective blocks $\mathbf{H}_k$ corresponding to neurons k. Considering that $\mathbf{H}_k \approx \langle \mathbf{a} \otimes \mathbf{a} g_k^2 \rangle_{\mathcal{D}}$, one can impose that for a small condition number, i.e. an efficient training, all post-activations have to be on the same scale, and all sensitivities also have to be on the same scale. Both requirements are satisfied for the SP if all layers have the same number of neurons.

An alternative parameterization of an NN is the neural tangent parameterization (NTP) proposed recently in Ref. [43]. In NTP, weights and biases are re-scaled as

$$\begin{aligned} \mathbf{W}^{(l+1)} &= \frac{1}{\sqrt{n_l}} \tilde{\mathbf{W}}^{(l+1)} \in \mathbb{R}^{n_{l+1} \times n_l}, \\ \mathbf{b}^{(l+1)} &= \beta \tilde{\mathbf{b}}^{(l+1)} \in \mathbb{R}^{n_{l+1}}, \end{aligned} \tag{3.29}$$

and the entries of the weight matrix $\tilde{\mathbf{W}}^{(l+1)}$ are taken from a normal distribution $\mathcal{N}(0,1)$, while biases $\tilde{\mathbf{b}}^{(l+1)}$ are initialized at zero. Note that the set of the functions that can be realized by Equation (3.27) is the same for both parameterizations, SP and NTP. However, for NTP, the gradients of the loss function with respect to the weight parameters $\partial \mathcal{L}(\boldsymbol{\theta}) / \partial \tilde{W}_{ij}^{(l)}$ and biases $\partial \mathcal{L}(\boldsymbol{\theta}) / \partial \tilde{b}_j^{(l)}$ are re-scaled by $1/\sqrt{n_l}$ and β, respectively.

For NTP, at least in a first-order Taylor approximation, the maximum possible learning rate for which gradient descent (GD) converges is asymptotically constant in width [109, 110]. Relative to SP, the NTP updates in layer l are smaller by a factor of $1/n_l$ since the gradient in layer l is multiplied by $1/\sqrt{n_l}$, and the gradient update is again multiplied by $1/\sqrt{n_l}$ in the forward pass. Hence, for SP, a learning rate of the order $1/(\max_l n_l)$ can be used if a single learning rate should be used jointly for all layers [110–112]. However, typically the input dimension is smaller than the hidden layers, $n_0 \ll n_l$ for $l = 1, \ldots, L$, and hence the learning rate in the first layer could be larger, i.e., of the constant order $1/n_0$. It is, therefore, possible to use a larger learning rate for the first layer in SP. This agrees with the empirical findings of Ref. [84], who have observed that the loss surface is often steeper in higher layers, as illustrated in Figure 3.6. In summary, NTP allows for fast training of the first layer without the use of layer-wise learning rates.

As a final remark, it should be mentioned that the factors $1/\sqrt{n_l}$ are crucial to obtaining a

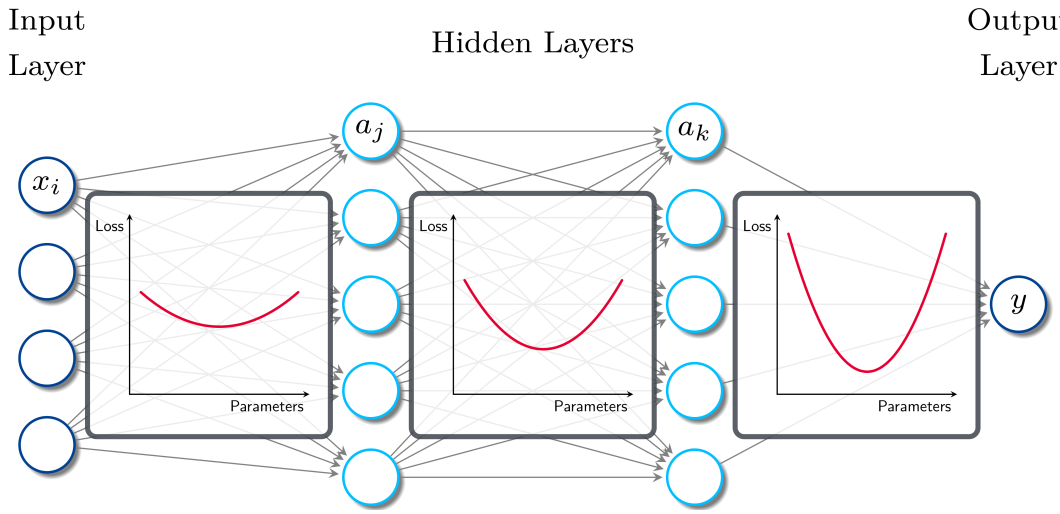


Figure 3.6: Schematic representation of the shape of the loss surface (red lines) depending on the layer. The gradient is often smaller in lower layers than in higher layers (from left to right), or equivalently, the loss surface in lower layers has smaller curvature than in higher layers. (Reproduced with permission from Ref. [84]. Copyright 2012, Springer-Verlag Berlin Heidelberg.)

consistent asymptotic behaviour of NNs as the width of the hidden layers $n_1, \ldots, n_L$ grows to infinity. The neural tangent kernel (NTK), which governs the training dynamics of infinitely wide NNs, reads

$$\left\langle \frac{\partial y}{\partial \tilde{\mathbf{W}}}, \frac{\partial y'}{\partial \tilde{\mathbf{W}}} \right\rangle = \frac{1}{n} \left\langle \frac{\partial y}{\partial \mathbf{W}}, \frac{\partial y'}{\partial \mathbf{W}} \right\rangle , \tag{3.30}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product and y is the network's output. For more details on the NTK theory, see Chapter 4. A side effect of $1/\sqrt{n_l}$ is that the influence of the connection weights during training is reduced when n_l is large. Therefore, the factor $\beta < 1$ is introduced to balance the influence of biases and connection weights. In Ref. [43], a value of 0.1 for β has been proposed.

3.2.4 Momentum, adaptive learning rates, and mini-batches

Here, other approaches aside from the heuristics and best practices described in previous sections are presented, leading to an improved convergence of gradient descent (GD). A general approach to increase the learning along the direction of low curvature of the loss surface while

damping the learning along directions with high curvature is to apply momentum in GD

$$\begin{aligned} \mathbf{g}_t &= \frac{\partial \mathcal{L}\left(\boldsymbol{\theta}\left(t\right)\right)}{\partial \boldsymbol{\theta}}, \\ \mathbf{m}_t &= \gamma \mathbf{m}_{t-1} + \left(1-\gamma\right)\mathbf{g}_t, \\ \Delta\boldsymbol{\theta}\left(t\right) &= -\alpha \mathbf{m}_t, \end{aligned} \tag{3.31}$$

where α is the learning rate, $0 \leq \gamma < 1$ is the strength of the momentum term and is usually chosen to be around 0.9 or 0.99.

Another possibility to improve the convergence of the gradient descent is to adapt the learning rate depending on the historical gradient in some previous iterations. In the most simple algorithm AdaGrad [113], this is realized by normalizing the gradients with those from all previous steps

$$\begin{aligned} \mathbf{g}_t &= \frac{\partial \mathcal{L}\left(\boldsymbol{\theta}\left(t\right)\right)}{\partial \boldsymbol{\theta}}, \\ \mathbf{v}_t &= \sum_{j=1}^{t} \mathbf{g}_j \circ \mathbf{g}_j, \\ \Delta\boldsymbol{\theta}_t &= -\alpha \mathbf{v}_t^{-1/2} \circ \mathbf{g}_t, \end{aligned} \tag{3.32}$$

where $\circ$ denotes the element-wise product.

The main drawback of such an algorithm is that it treats all past gradients equally. Thus, applying momentum to the adaptive learning rate algorithm is natural, i.e., using exponentially decaying weights for the past gradients. This modification has led to multiple improved algorithms such as, e.g., AdaDelta [114], but the most successful one is the Adam optimizer [115], which incorporates the second-order momentum $\mathbf{v}_t$

$$\begin{aligned} \mathbf{g}_t &= \frac{\partial \mathcal{L}\left(\boldsymbol{\theta}\left(t\right)\right)}{\partial \boldsymbol{\theta}}, \\ \mathbf{m}_t &= \gamma_1 \mathbf{m}_{t-1} + \left(1-\gamma_1\right)\mathbf{g}_t, \\ \mathbf{v}_t &= \gamma_2 \mathbf{v}_{t-1} + \left(1-\gamma_2\right)\mathbf{g}_t \circ \mathbf{g}_t, \\ \Delta\boldsymbol{\theta}_t &= -\alpha \mathbf{v}_t^{-1/2} \circ \mathbf{m}_t. \end{aligned} \tag{3.33}$$

Here, the calculation of the bias-corrected momentum $\mathbf{m}_t$ and normalization $\mathbf{v}_t$, necessary as the respective moving averages are initialized at zero, has been skipped for simplicity. For more details on the original algorithm, see Ref. [115]. All these algorithms, including many others, are implemented in modern NN frameworks such as, e.g., TensorFlow [99], Py-

torch [100], and JAX [101].

As a final remark, it should be mentioned that, in practice, training on the whole training set at once is not feasible as the evaluation of the loss function and its gradients with respect to the trainable parameters become very expensive, both computationally and in terms of memory. Therefore, the so-called mini-batch gradient descent (MBGD) is usually employed [116], where a group of samples is processed during training. Let N_{Batch} be the number of mini-batches and assume that N_{Train} is divisible by N_{Batch}. Then, each mini-batch contains $|I| = N_{\text{Train}}/N_{\text{Batch}}$ training samples. Now, given a random mini-batch $I \subset \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{Train}}}$ of size $|I|$, a loss function on I can be defined as

$$\mathcal{L}^{(I)}(\boldsymbol{\theta}) = \frac{1}{|I|} \sum_{i \in I} \mathcal{L}^{(i)}(\boldsymbol{\theta}), \tag{3.34}$$

where $\mathcal{L}^{(i)}(\boldsymbol{\theta})$ is the loss function for the ith training sample, e.g.

$$\mathcal{L}^{(i)}(\boldsymbol{\theta}) = \frac{1}{2} \left(f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i \right)^2, \tag{3.35}$$

and the total loss is given by

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N_{\text{Batch}}} \sum_{I=1}^{N_{\text{Batch}}} \mathcal{L}^{(I)}(\boldsymbol{\theta}). \tag{3.36}$$

Finally, one may write for the MBGD update rule

$$\Delta\boldsymbol{\theta}(t) = -\alpha \frac{\partial \mathcal{L}^{(I_t)}(\boldsymbol{\theta}(t))}{\partial \boldsymbol{\theta}}. \tag{3.37}$$

Typical mini-batch sizes range between 10 and 100 samples in the problem settings considered in this work. However, different values can be sensible in other settings, as the mini-batch size is a compromise between statistical accuracy, parallelization, and memory limitations of the given NN on the provided hardware.

3.3 Improving the generalization of neural networks

One of the central goals of machine learning (ML) is to obtain a model that performs well not only on the training samples but also on new inputs not seen during training, or in other words, generalizes well. Here, several techniques are presented to reduce the generalization error and

improve the transferability of NN-based models to new scenarios. The starting point is a regularization technique that introduces the L^2 parameter norm penalty to the loss function, the so-called weight decay technique, see Section 3.3.1. From here, in Section 3.3.2, a different approach to the regularization of an NN, so-called early stopping, is presented. Moreover, it is shown that early stopping is closely related to the weight decay regularization and advantages of the former compared to the latter are highlighted. Finally, in Section 3.3.3, the problem of the introduction of prior knowledge on an example of interatomic NN potentials is introduced, and possible solutions such as the construction of invariant representations and data augmentation are discussed.

3.3.1 L^2 parameter regularization

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}} \subset \mathbb{R}^d \times \mathbb{R}$ denote the training set with inputs $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$. Moreover, let $\boldsymbol{\theta}$ be trainable parameters of an ML model, an NN in this specific case. A common way to regularize an ML model is to add a parameter norm penalty $\Omega(\boldsymbol{\theta})$ to the original loss function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) = \mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) + \beta\Omega(\boldsymbol{\theta}). \tag{3.38}$$

Here, $\beta \in [0, \infty)$ defines the relative contribution of the norm penalty $\Omega(\boldsymbol{\theta})$ relative to the original loss function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$. This technique may be applied to all ML algorithms such as NNs, linear regression or logistic regression [85]. Here, only the application of parameter norm penalty to NN-based models is considered.

Independent of the exact form of the penalty $\Omega(\boldsymbol{\theta})$, when minimizing the regularized loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$ both the original loss function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$ and the norm of parameters $\boldsymbol{\theta}$ are reduced. In general, different choices for the parameter norm $\Omega(\boldsymbol{\theta})$ are possible. Each of them results in a different solution being preferred. Note that, typically, only weight parameters of an NN are penalized, i.e. $\boldsymbol{\theta} = \mathbf{W}$, while biases are not regularized. This is because the latter needs fewer data to accurately fit and only shifts the activation function.

While different choices of the parameter norm are possible, see, e.g., Ref. [85], here, only the L^2 norm, often referred to as weight decay, is discussed following Ref. [85]. The L^2 parameter regularization forces the weights $\mathbf{W}$ closer to zero by adding a regularization term $\beta/2\mathbf{W}^T\mathbf{W}$ to the original loss function $\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$

$$\tilde{\mathcal{L}}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) = \mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) + \frac{\beta}{2}\mathbf{W}^T\mathbf{W}. \tag{3.39}$$

Here, $\mathbf{W}$ is a vector containing weight matrix elements.

The gradient of the regularized loss function $\tilde{\mathcal{L}}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y})$ with respect to weight parameters $\mathbf{W}$ reads

$$\frac{\partial \tilde{\mathcal{L}}(\mathbf{W})}{\partial \mathbf{W}} = \frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} + \beta \mathbf{W}, \tag{3.40}$$

and the update rule for the parameters is given by

$$\mathbf{W}_{t+1} = (1 - \alpha\beta)\, \mathbf{W}_t - \alpha \frac{\partial \mathcal{L}(\mathbf{W}_t)}{\partial \mathbf{W}}, \tag{3.41}$$

where α is the learning rate of an GD algorithm. The expression in Equation (3.41) shows that the L^2 penalty reduces the weight parameters by a constant factor at each training step before performing the regular GD update. While it is a result obtained after a single iteration, in the following, it will be shown what happens over the entire course of training.

Similar to the previous section (see Section 3.2), the Hessian-based analysis may be employed to explain what happens during training when the L^2 parameter regularization term is added to the loss functions [85]. For this purpose, the original loss function $\mathcal{L}(\mathbf{W})$ is expanded into its Taylor series around the optimal weights $\mathbf{W}^*$

$$\mathcal{L}(\mathbf{W}) \approx \mathcal{L}(\mathbf{W}^*) + \frac{1}{2} (\mathbf{W} - \mathbf{W}^*)^T \mathbf{H} (\mathbf{W} - \mathbf{W}^*), \tag{3.42}$$

where $\mathbf{H} = \partial^2 \mathcal{L}(\mathbf{W}) / \partial \mathbf{W}^2 |_{\mathbf{W}=\mathbf{W}^*}$ is the Hessian matrix and is positive semi-definite since $\mathbf{W}^*$ is the location of minimum of the loss function $\mathcal{L}(\mathbf{W})$. Thus, the gradient of $\mathcal{L}(\mathbf{W})$ may be written by employing the Hessian matrix as

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{H} (\mathbf{W} - \mathbf{W}^*). \tag{3.43}$$

Now, the L^2 regularization can be added to the gradient in Equation (3.43) and the respective equation is solved such that the minimum of the regularized loss $\tilde{\mathcal{L}}(\mathbf{W})$ is found

$$\beta \mathbf{W} + \mathbf{H} (\mathbf{W} - \mathbf{W}^*) = 0, \tag{3.44}$$

$$(\mathbf{H} + \beta \mathbf{I})\, \mathbf{W} = \mathbf{H} \mathbf{W}^*, \tag{3.45}$$

$$\mathbf{W} = (\mathbf{H} + \beta \mathbf{I})^{-1} \mathbf{H} \mathbf{W}^*. \tag{3.46}$$

For small β, $\mathbf{W}$ in Equation (3.46) approaches $\mathbf{W}^*$. To see what happens for large β the Hessian matrix $\mathbf{H}$ is decomposed into a diagonal matrix $\mathbf{D}$ and the corresponding orthonormal

basis of eigenvectors $\mathbf{U}$, such that $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^T$. Applying the aforementioned decompositions to the expression in Equation (3.46) one obtains

$$\mathbf{W} = \left(\mathbf{U}\mathbf{D}\mathbf{U}^T + \beta\mathbf{I}\right)^{-1}\mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{W}^*, \tag{3.47}$$

$$\mathbf{W} = \left(\mathbf{U}\left(\mathbf{D} + \beta\mathbf{I}\right)\mathbf{U}^T\right)^{-1}\mathbf{U}\mathbf{D}\mathbf{U}^T\mathbf{W}^*, \tag{3.48}$$

$$\mathbf{W} = \mathbf{U}\left(\mathbf{D} + \beta\mathbf{I}\right)^{-1}\mathbf{D}\mathbf{U}^T\mathbf{W}^*. \tag{3.49}$$

One can see from the Hessian-based analysis that an effect of the L^2 regularization is the re-scaling of $\mathbf{W}^*$ along the axes defined by the eigenvectors of $\mathbf{H}$. The components of $\mathbf{W}^*$, aligned with the ith eigenvector of $\mathbf{H}$, are re-scaled by $\lambda_i / (\lambda_i + \beta)$. Thus, along the directions with $\lambda_i \gg \beta$, the effect of regularization is relatively small. However, along the directions with $\lambda_i \ll \beta$, the elements of the weight vector will decrease to nearly zero. The weight decay regularization technique suppresses the weight vector components for directions with a small gradient, i.e. for the directions along which the parameters do not contribute to the reduction of the loss function.

3.3.2 Early stopping

As mentioned before, when training an NN model, one usually expects as output a network with an optimal out-of-sample or generalization performance. Unfortunately, due to the over-parameterization, all standard NNs are prone to overfitting [117]: While the error on the training samples seems to get continuously better, the performance on the unseen data may deteriorate. The schematic evolution of the generalization error during training is shown in Figure 3.7.

In general, the overfitting may be conquered in two different ways. First, the number of parameters can be reduced, leading to less expressive models. Second, a regularization technique may be applied to improve an NN's generalization ability. Among the suite of existing regularization techniques [85], the most common approach is the early stopping [118]. Therefore, here, the basic ideas of the early stopping technique are presented, while several approaches for choosing, e.g., stopping criteria, can be found in the literature [119].

A standard procedure for early stopping and selecting the best set of parameters consists of the random splitting of the given data set in three parts: training, validation, and test data set. The training data set is used to adjust the trainable parameters by minimizing the loss function, e.g. the squared loss in Equation (3.17). The validation data set is used to track the model's performance on the unseen data and select the model with the lowest generalization

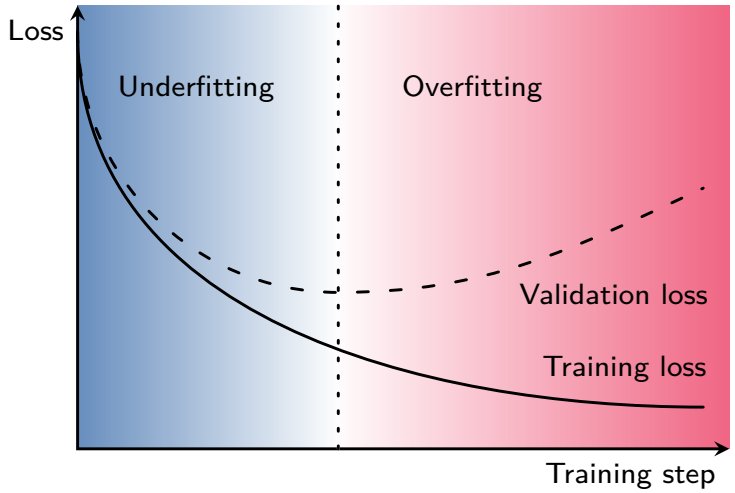


Figure 3.7: Schematic representation of the idealized training (solid line) and validation (dashed line) losses as a function of the training step. The dotted vertical line represents the best model chosen according to the minimal validation loss, while colors represent an overfitted (red) and an underfitted (blue) neural network (NN).

error. The test data set is employed to get an unbiased estimate of the model's out-of-sample performance. Note that validation data set cannot be used for this purpose since it has been used to select the final model and, thus, indirectly biased the training procedure.

An essential ingredient of the early stopping technique is the metric employed to evaluate the model's performance. Independent of the loss function used during training, the error metric should reflect what good performance for a specific task is assumed to be. Common evaluation metrics are the root-mean-square error (RMSE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{N_{\mathrm{data}}} \sum_{i=1}^{N_{\mathrm{data}}} \left(y_i - y_i^{\mathrm{ref}}\right)^2}, \tag{3.50}$$

and the mean absolute error (MAE)

$$\mathrm{MAE} = \frac{1}{N_{\mathrm{data}}} \sum_{i=1}^{N_{\mathrm{data}}} \left|y_i - y_i^{\mathrm{ref}}\right|, \tag{3.51}$$

where N_{data} is the number of samples in, e.g., the validation data set.

The main limitation of this regularization technique is that one assumes that the validation and test data sets cover a reasonably large part of the target values such that the model's

generalization ability can be assessed. This assumption is not valid in many scenarios. For example, often, one does not know whether the trained model will be exposed to some new scenarios on which it will extrapolate. To conquer this shortage of NN-based models, AL strategies [34] can be applied to track the model's performance by exploring new scenarios and adding them to the training data or selecting the most representative sample from the given data set.

Besides the basic ideas of the early stopping technique, one can show its correspondence to the L^2 regularization, see Section 3.3.1. The following will show explicitly that early stopping is a regularization technique and can be seen to be equivalent to the L^2 regularization [85]. The discussion presented below follows Ref. [85]. Similar to the previous section (Section 3.3.1), the loss function is expanded around the optimal weight parameters $\mathbf{W}^*$ in its Taylor series as

$$\mathcal{L}(\mathbf{W}) \approx \mathcal{L}(\mathbf{W}^*) + \frac{1}{2}(\mathbf{W} - \mathbf{W}^*)^T \mathbf{H}(\mathbf{W} - \mathbf{W}^*), \tag{3.52}$$

where $\mathbf{H} = \partial^2 \mathcal{L}(\mathbf{W}) / \partial \mathbf{W}^2|_{\mathbf{W}=\mathbf{W}^*}$ is the Hessian matrix and is positive semi-definite since $\mathbf{W}^*$ is the location of the minimum of $\mathcal{L}(\mathbf{W})$. The gradient under the Taylor series approximation becomes

$$\frac{\partial \mathcal{L}(\mathbf{W})}{\partial \mathbf{W}} = \mathbf{H}(\mathbf{W} - \mathbf{W}^*). \tag{3.53}$$

Let the weight vector be initialized at zeros, i.e. $\mathbf{W}_0 = \mathbf{0}$. The update of the parameters via gradient descent (GD) is defined by

$$\mathbf{W}_t = \mathbf{W}_{t-1} - \alpha \mathbf{H}(\mathbf{W}_{t-1} - \mathbf{W}^*), \tag{3.54}$$

$$\mathbf{W}_t - \mathbf{W}^* = (\mathbf{I} - \alpha \mathbf{H})(\mathbf{W}_{t-1} - \mathbf{W}^*), \tag{3.55}$$

where t denotes the training step. Applying the eigenvalue decompositions of the Hessian matrix $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{U}^T$, one obtains the following expressions for the parameter update

$$\mathbf{W}_t - \mathbf{W}^* = \left(\mathbf{I} - \alpha \mathbf{U}\mathbf{D}\mathbf{U}^T\right)(\mathbf{W}_{t-1} - \mathbf{W}^*), \tag{3.56}$$

$$\mathbf{U}^T(\mathbf{W}_t - \mathbf{W}^*) = (\mathbf{I} - \alpha \mathbf{D})\mathbf{U}^T(\mathbf{W}_{t-1} - \mathbf{W}^*). \tag{3.57}$$

Finally, choosing $\mathbf{W}^{(0)} = \mathbf{0}$ and α sufficiently small one obtains for the training dynamics of the parameters $\mathbf{W}$

$$\mathbf{U}^T\mathbf{W}_t = \left(\mathbf{I} - (\mathbf{I} - \alpha \mathbf{D})^t\right)\mathbf{U}^T\mathbf{W}^*. \tag{3.58}$$

Using the expression in Equation (3.49) for the L^2 regularization one obtains for $\mathbf{U}^T\mathbf{W}$

after re-arranging[7]

$$\mathbf{U}^T\mathbf{W} = (\mathbf{D} + \beta\mathbf{I})^{-1}\,\mathbf{D}\mathbf{U}^T\mathbf{W}^*, \tag{3.59}$$

$$\mathbf{U}^T\mathbf{W} = \left(\mathbf{I} - (\mathbf{D} + \beta\mathbf{I})^{-1}\,\beta\right)\mathbf{U}^T\mathbf{W}^*, \tag{3.60}$$

From Equation (3.58) and Equation (3.60), one can see that the early stopping and L^2 regularization can be considered to be equivalent, if for the chosen parameters α, t, and β it holds that [85]

$$(\mathbf{I} - \alpha\mathbf{D})^t = (\mathbf{D} + \beta\mathbf{I})^{-1}\,\beta. \tag{3.61}$$

Moreover, by taking the logarithm of the right and the left-hand side of Equation (3.61) and using the series expansion for $\log(1 + x)$ one obtains

$$t \approx \frac{1}{\alpha\beta}, \tag{3.62}$$

$$\beta \approx \frac{1}{t\alpha}. \tag{3.63}$$

Note that the eigenvalues of the Hessian are assumed to be small enough, i.e. $\lambda_i\alpha \ll 1$ and $\lambda_i/\beta \ll 1$. Thus, for the quadratic approximation employed in this analysis, the number of training iterations t is inversely proportional to the L^2 regularization parameter β.

Due to the equivalence of the L^2 regularization technique and the early stopping approach, at least within the quadratic approximation, it can be stated for the latter that the parameter values corresponding to the directions of large curvature of the loss function are regularized less than directions of less curvature. In the context of early stopping, the parameters corresponding to the directions of larger curvature tend to learn in the early stages of the training compared to the parameters corresponding to the directions of less curvature.

One of the main results of the Hessian-based analysis is that a trajectory of length t ends at a point that corresponds to a minimum of the L^2-regularized loss. However, early stopping goes beyond restricting the trajectory length or equivalent to the L^2 regularization. Indeed, early stopping, as has been discussed earlier, involves the evaluation of the model performance employing the validation data set error to stop the trajectory at an appropriate point in parameter space. Therefore, early stopping has the advantage over the weight decay technique in that

[7]The special case of the Woodbury matrix identity

$$(\mathbf{A} + \mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\left(\mathbf{A}\mathbf{B}^{-1} + \mathbf{I}\right)^{-1},$$

has been used to get the final result.

it automatically determines the correct amount of regularization. Instead, the weight decay technique requires many training experiments to find the best hyperparameters to achieve the same goal.

As a final remark, it should be mentioned again that the presented results assume GD training and a quadratic approximation to the loss function. These assumptions, however, can be violated in practice, and weight decay still remains a popular regularization technique.

3.3.3 The introduction of prior knowledge

Many specific problems in machine learning (ML) require the encoding of prior knowledge about the system, e.g. invariances of the electronic energy with respect to rotations, translation, and the exchange of like atoms. Introducing them is crucial to get a model with a good generalization ability. There are many possibilities to introduce prior knowledge into an ML algorithm. However, this section is dedicated mainly to the data augmentation technique and the construction of a suitable input representation. Since the current work uses NNs to build interatomic NN potentials, a case where the inputs are atomic coordinates and labels are electronic energies will be considered.

As mentioned above, one way to introduce prior knowledge to the ML algorithms is data set augmentation. Here, artificial data is created and added to the training such that the model learns, e.g., the symmetries of the electronic energy by itself. The new inputs $(\mathbf{x}, y)$ are created by transforming the inputs $\mathbf{x}$ but the corresponding labels y remain the same. For example, the invariance of the electronic energy to rotations can be considered. The respective symmetry can be induced by defining a set of matrices $\{\mathbf{U}_k\}_{k=1}^{K}$ where $\mathbf{U}_k \in O(3)$ and $O(3)$ is the orthogonal group in $\mathbb{R}^3$ and applying them to the original data to create the augmented data set

$$\tilde{\mathcal{D}} = \bigcup_{(\mathbf{x},y)\in\mathcal{D}} \left\{ (\mathbf{x}, y), \{(\mathbf{U}_k\mathbf{x}, y)\}_{k=1}^{K} \right\}. \tag{3.64}$$

While the data set augmentation can be deemed suitable for the NN-based models, it may lead to a vast amount of data due to invariances such as the invariance to the permutation of like atoms. Thus, the data set may grow quickly, requiring larger network sizes, leading to less efficient models at training and inference. Moreover, the invariances introduced via the data augmentation technique may lead to poor performance outside the data set compared to methods with explicitly encoded symmetries.

Another way to include the prior knowledge into an ML algorithm is to define an appropriate invariant representation. For the electronic energies, it would be a representation which

transforms atomic coordinates into an ML input, and imposes the invariance with respect to the global rotation, the translation, and the reflection of a molecular structure, as well as the exchange of like atoms, i.e. atoms with the same nuclear charge Z. Probably, the simplest solution, which satisfies all requirements except for the permutational invariance, can be constructed using just the scalar product of distance vectors $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ from the position of a central atom $\mathbf{r}_i$ to the positions $\mathbf{r}_j$ of all other atoms, resulting in the Weyl matrix $\mathbf{\Sigma}_i$ [120]

$$\mathbf{\Sigma}_i = \begin{pmatrix} \mathbf{r}_{i1} \cdot \mathbf{r}_{i1} & \mathbf{r}_{i1} \cdot \mathbf{r}_{i2} & \cdots \\ \mathbf{r}_{i2} \cdot \mathbf{r}_{i1} & \mathbf{r}_{i2} \cdot \mathbf{r}_{i2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} . \tag{3.65}$$

Training an ML model employing such a representation ensures that the predictions of the respective model on unseen samples also have the desired invariance.

4 On the theory of ultra-wide neural networks

The outstanding performance of artificial neural networks (NNs) in real-world applications can, in many cases, be attributed to the over-parameterization of NNs. For example, it is known that given appropriate hyperparameters, over-parameterized feed-forward NNs trained by gradient descent (GD) algorithms reach near-zero training errors [121]. Moreover, the over-parameterization of NNs is often believed to be responsible for their excellent generalization ability. Nonetheless, the detailed analysis of the convergence of NNs under GD is hindered by the fact that the training of an NN is a non-convex problem. Over-parameterization also brings new challenges in the analysis of their generalization capability. In this chapter, the neural tangent kernel (NTK) theory, first proposed in Ref. [43], is briefly described to shed light on the training dynamics, the optimization, and the generalization of over-parameterized feed-forward NNs. Starting with the basic network set-up in Section 4.1, the training dynamics governed by the NTK kernel is derived for the case of a mean squared error (MSE) loss. Next, in Section 4.2, the training dynamics of a linearized network is described with a few practical examples [122]. Lastly, Sections 4.3 and 4.4 explain the optimization and the generalization of ultra-wide NNs, respectively, by employing the NTK theory. The content of this chapter follows discussions and derivations in Refs. [43, 122–124].

4.1 The basic network setup and training dynamics

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}} \subset \mathbb{R}^d \times \mathbb{R}$ denote the training set with inputs $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ and labels $y_i \in \mathcal{Y} \subset \mathbb{R}$. For simplicity, only a single output is considered here, and the generalisation to multiple outputs is straightforward. In the following, a fully-connected feed-forward neural network (NN)

$$f(\cdot; \boldsymbol{\theta}) : \mathbb{R}^d \to \mathbb{R}, \tag{4.1}$$

with L hidden layers of width n_l, for $l = 1, \ldots, L$ is studied. Here, $n_0 = d$ and $n_{L+1} = 1$ describe the input and output dimensions, respectively, and the recurrence relation

$$\begin{cases} \mathbf{a}^{(0)} = \mathbf{x} \in \mathbb{R}^d, \\ \mathbf{h}^{(l+1)} = \frac{\sigma_w}{\sqrt{n_l}} \mathbf{W}^{(l+1)} \mathbf{a}^{(l)} + \sigma_b \mathbf{b}^{(l+1)} \in \mathbb{R}^{n_{l+1}}, \\ \mathbf{a}^{(l+1)} = \varphi(\mathbf{h}^{(l+1)}) \in \mathbb{R}^{n_{l+1}}, \end{cases} \tag{4.2}$$

defines the NN. Here, $\varphi : \mathbb{R} \to \mathbb{R}$ is the activation function applied element-wise to an input $\mathbf{h}^{(l+1)}$, $\mathbf{W}^{(l+1)} \in \mathbb{R}^{n_{l+1} \times n_l}$ and $\mathbf{b}^{(l+1)} \in \mathbb{R}^{n_{l+1}}$ are weights and biases initialized by picking the respective entries from, e.g., a standard Gaussian with zero mean and unit variance $\mathcal{N}(0, 1)$. Here, σ_w and σ_b are weight and bias scaling factors. This parameterization is referred to as the neural tangent parameterization (NTP) [43]. While a standard parameterization (SP), i.e., the Kaiming [108] or LeCun [84] initialization, normalizes only the forward dynamics of NNs, NTP normalizes their backward dynamics, too; see also Section 3.2.3.

In what follows, a vector containing all parameters is defined as

$$\boldsymbol{\theta} = \left(\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \ldots, \mathbf{W}^{(L+1)}, \mathbf{b}^{(L+1)} \right) \in \mathbb{R}^{|\boldsymbol{\theta}|}, \tag{4.3}$$

with the dimension of the parameter space given by $|\boldsymbol{\theta}| = \sum_l (n_{l-1} + 1) n_l$. Let $\boldsymbol{\theta}_t$ denote the time-dependence of the parameters, while $\boldsymbol{\theta}_0$ represent their initial values. In the following, the training of NNs by minimizing the squared loss over training data is studied

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{X}, \mathcal{Y}) = \frac{1}{2} \sum_{i=1}^{N_{\text{train}}} \left(f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i \right)^2, \tag{4.4}$$

but other choices are also possible. The output of an NN is defined as $f(\mathbf{x}; \boldsymbol{\theta}_t) = \mathbf{h}^{(L+1)}(\mathbf{x})$. The time evolution of parameters $\boldsymbol{\theta}$ under gradient flow, the continuous-time equivalent of gradient descent (GD), reads

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{\theta}_t}{\mathrm{d}t} &= -\alpha \frac{\partial \mathcal{L}(\boldsymbol{\theta}_t; \mathcal{X}, \mathcal{Y})}{\partial \boldsymbol{\theta}} \\ &= -\alpha \sum_{i=1}^{N_{\text{train}}} \left(f(\mathbf{x}_i; \boldsymbol{\theta}_t) - y_i \right) \frac{\partial f(\mathbf{x}_i; \boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}, \end{aligned} \tag{4.5}$$

where $\alpha > 0$ is the learning rate. Compared to SP, here, α is larger by a factor of the

width [109].[1] Therefore, NTP allows the usage of a universal learning rate scale irrespective of the network width; see also Section 3.2.3.

For the time evolution of the network output, one obtains employing the chain rule

$$\begin{aligned}\frac{\mathrm{d}f\left(\mathbf{x}_i;\boldsymbol{\theta}_t\right)}{\mathrm{d}t} &= \left\langle \frac{\partial f\left(\mathbf{x}_i;\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}}, \frac{\mathrm{d}\boldsymbol{\theta}}{\mathrm{d}t} \right\rangle \\ &= -\alpha \sum_{j=1}^{N_{\text{train}}} \left(f\left(\mathbf{x}_j;\boldsymbol{\theta}_t\right) - y_j\right) \left\langle \frac{\partial f\left(\mathbf{x}_i;\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}}, \frac{\partial f\left(\mathbf{x}_j;\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}} \right\rangle,\end{aligned} \tag{4.6}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. The expression in Equation (4.6) can be written in its vectorized form as

$$\frac{\mathrm{d}f\left(\mathcal{X};\boldsymbol{\theta}_t\right)}{\mathrm{d}t} = -\alpha \mathbf{H}_t\left(\mathcal{X},\mathcal{X}\right)\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right) - \mathcal{Y}\right), \tag{4.7}$$

where $\mathbf{H}_t\left(\mathcal{X},\mathcal{X}\right) \in \mathbb{R}^{N_{\text{train}} \times N_{\text{train}}}$ is a kernel matrix an element of which is defined by

$$H_t\left(\mathbf{x}_i,\mathbf{x}_j\right) = \left\langle \frac{\partial f\left(\mathbf{x}_i;\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}}, \frac{\partial f\left(\mathbf{x}_j;\boldsymbol{\theta}_t\right)}{\partial \boldsymbol{\theta}} \right\rangle. \tag{4.8}$$

In general, one can show that in the limit $n_l \to \infty$, for $l = 1, \ldots, L$, the matrix $\mathbf{H}_t\left(\mathcal{X},\mathcal{X}\right)$ remains constant during training, i.e., $\mathbf{H}_t\left(\mathcal{X},\mathcal{X}\right) = \mathbf{H}_0\left(\mathcal{X},\mathcal{X}\right)$. Moreover, under a random initialization of parameters, the random matrix $\mathbf{H}_0\left(\mathcal{X},\mathcal{X}\right)$ converges in probability to a certain deterministic kernel matrix $\mathbf{H}^{\infty}\left(\mathcal{X},\mathcal{X}\right)$ as the width goes to infinity. The respective kernel is the neural tangent kernel (NTK) evaluated on the training data [43]. It is out of this work's scope to prove the statements mentioned above. Therefore, their validity will be shown empirically in Section 4.2 by studying the linearized networks, shown to be equivalent to the infinitely wide ones [122]. For the proofs and further information on the NTK, see Refs. [43, 123, 125, 126].

Taking into account that $\mathbf{H}_t\left(\mathcal{X},\mathcal{X}\right) \to \mathbf{H}^{\infty}\left(\mathcal{X},\mathcal{X}\right)$ for $n_l \to \infty$ and $l = 1, \ldots, L$, one may write

$$\frac{\mathrm{d}f\left(\mathcal{X};\boldsymbol{\theta}_t\right)}{\mathrm{d}t} = -\alpha \mathbf{H}^{\infty}\left(\mathcal{X},\mathcal{X}\right)\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right) - \mathcal{Y}\right), \tag{4.9}$$

[1] Assume a simple layer $\mathbf{h} = \mathbf{W}\mathbf{x}$ with $\mathbf{W} = \tilde{\mathbf{W}}/\sqrt{n}$. Here, weight elements are initialized as $W_{ij} \sim \mathcal{N}\left(0, 1/n\right)$ and $\tilde{W}_{ij} \sim \mathcal{N}\left(0,1\right)$, respectively. If GD is performed on $\tilde{\mathbf{W}}$, the corresponding update rule is given by $\Delta\tilde{\mathbf{W}} = -\alpha\partial\mathcal{L}\left(\mathbf{W}\right)/\partial\tilde{\mathbf{W}}$. Thus, the update for the parameters $\mathbf{W}$ reads

$$\Delta\mathbf{W} = \frac{1}{\sqrt{n}}\Delta\tilde{\mathbf{W}} = -\frac{\alpha}{n}\frac{\partial\mathcal{L}\left(\mathbf{W}\right)}{\partial\mathbf{W}}.$$

However, if GD is performed on $\mathbf{W}$ one obtains $\Delta\mathbf{W} = -\alpha\partial\mathcal{L}\left(\mathbf{W}\right)/\partial\mathbf{W}$, which is larger by a factor of n.

and the training dynamics is governed by the NTK kernel $\mathbf{H}^{\infty}(\mathcal{X}, \mathcal{X})$. The training dynamics described by Equation (4.9) is identical to the dynamics of kernel regression under GD. Therefore, with a small modification of the NN (see Ref. [125]), for an arbitrary test point $\mathbf{x}$, one can write for $t \to \infty$ the following solution

$$f(\mathbf{x}; \boldsymbol{\theta}_{\infty}) = \mathbf{H}^{\infty}(\mathbf{x}, \mathcal{X}) \left(\mathbf{H}^{\infty}(\mathcal{X}, \mathcal{X})\right)^{-1} \mathcal{Y}. \tag{4.10}$$

4.2 The closed-form of training dynamics for linearized networks

Unlike the previous section, the training dynamics of a linearized neural network is considered here. The linearized network is obtained from the first-order Taylor expansion around its initial parameters $\boldsymbol{\theta}_0$ and reads [122]

$$f^{\text{lin}}(\mathbf{x}; \boldsymbol{\theta}_t) = f(\mathbf{x}; \boldsymbol{\theta}_0) + \left.\frac{\partial f(\mathbf{x}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \mathbf{w}_t, \tag{4.11}$$

where $\mathbf{w}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_0$. The expressions obtained for the time evolution of the parameters and the network output under GD in Equations 4.5 and 4.6 can be simplified to

$$\frac{\mathrm{d}\mathbf{w}_t}{\mathrm{d}t} = -\alpha \left(f^{\text{lin}}(\mathcal{X}; \boldsymbol{\theta}_t) - \mathcal{Y}\right) \left.\frac{\partial f(\mathcal{X}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}, \tag{4.12}$$

$$\frac{\mathrm{d}f^{\text{lin}}(\mathcal{X}; \boldsymbol{\theta}_t)}{\mathrm{d}t} = -\alpha \mathbf{H}_0(\mathcal{X}, \mathcal{X}) \left(f^{\text{lin}}(\mathcal{X}; \boldsymbol{\theta}_t) - \mathcal{Y}\right), \tag{4.13}$$

where $\mathbf{H}_0(\mathcal{X}, \mathcal{X})$ is the kernel matrix in Equation (4.8) evaluated at time $t = 0$ which elements read

$$H_0(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \left.\frac{\partial f(\mathbf{x}_i; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}, \left.\frac{\partial f(\mathbf{x}_j; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \right\rangle. \tag{4.14}$$

For the ordinary differential equations presented in Equations 4.12 and 4.13, closed-form solutions can be easily found both for the parameters

$$\mathbf{w}_t = -\left.\frac{\partial f(\mathcal{X}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^T\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} \mathbf{H}_0^{-1}(\mathcal{X}, \mathcal{X}) \left(\mathbf{I} - e^{-\alpha \mathbf{H}_0(\mathcal{X}, \mathcal{X}) t}\right) \left(f(\mathcal{X}; \boldsymbol{\theta}_0) - \mathcal{Y}\right), \tag{4.15}$$

and the network's output

$$f^{\text{lin}}\left(\mathcal{X};\boldsymbol{\theta}_t\right)=\left(\mathbf{I}-e^{-\alpha\mathbf{H}_0(\mathcal{X},\mathcal{X})t}\right)\mathcal{Y}+e^{-\alpha\mathbf{H}_0(\mathcal{X},\mathcal{X})t}f\left(\mathcal{X};\boldsymbol{\theta}_0\right). \tag{4.16}$$

Therefore, for an arbitrary test point $\mathbf{x}$, the output of the network can be obtained by plugging the expression for $\mathbf{w}_t$ in Equation (4.11). It reads

$$\begin{aligned} f^{\text{lin}}\left(\mathbf{x};\boldsymbol{\theta}_t\right)&=f\left(\mathbf{x};\boldsymbol{\theta}_0\right)\\ &\quad-\mathbf{H}_0\left(\mathbf{x},\mathcal{X}\right)\mathbf{H}_0^{-1}\left(\mathcal{X},\mathcal{X}\right)\left(\mathbf{I}-e^{-\alpha\mathbf{H}_0(\mathcal{X},\mathcal{X})t}\right)\left(f\left(\mathcal{X};\boldsymbol{\theta}_0\right)-\mathcal{Y}\right), \end{aligned} \tag{4.17}$$

such that the time evolution of the linearized neural network model has been obtained without running GD.

Interestingly, one can show that infinitely wide networks behave as linearized networks; see Ref. [122]. It leads to the same argument as in Section 4.1: The training dynamics of a neural network (NN) is described by a kernel $\mathbf{H}^{\infty}$ in the limit of $n_l\to\infty$ for $l=1,\ldots,L$. The main objective of this section is to show this empirically by running simple experiments rather than prove it analytically. For the empirical study of the training dynamics of the linearized network, a simple example of training on five points drawn randomly from $y=f\left(x\right)+\epsilon$ is emloyed. Here, $f\left(x\right)=\sin\left(x\right)$ is the sine function and $\epsilon\sim\mathcal{N}\left(0,\sigma\right)$ is Gaussian noise. The linearized training dynamics has been developed and implemented in Ref. [122] within Ref. [127] and the example presented here follows the respective interactive Colab notebook.[2]

Figure 4.1 shows the sine functions and the five selected points the model will be trained on. Note that different training data sets may result in different models and, thus, different results. It also holds for the chosen architecture. For the experiments presented here, the network architecture from the interactive Colab notebook[2] has been adapted. Namely, a fully-connected NN with two hidden layers has been used, while the number of nodes in each layer is $n_0=1$, $n_1=512$, $n_2=512$, and $n_3=1$, respectively. The Gauss error function has been used as the non-linearity for all hidden layers. All weights and biases have been initialized by drawing their elements from the standard Gaussian with zero mean and unit variance $\mathcal{N}\left(0,1\right)$, and the scale of weights and biases σ_w and σ_b were set to 1.5 and 0.05, respectively.

Figure 4.2 shows the training dynamics and the inference of the infinite (linearized) network, including the prediction variance. The figure also depicts the training dynamics of the finite network. From the figure, a correspondence between both training dynamics can be seen.

[2] `colab.research.google.com/github/google/neural-tangents/blob/main/notebooks/neural_tangents_cookbook.ipynb`

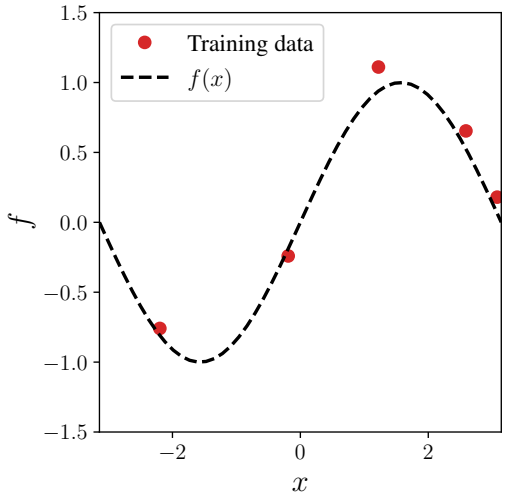


Figure 4.1: The underlying sine function with the randomly generated training data points. (Reproduced from the interactive Colab notebook.[2] Copyright 2019, Google LLC.)

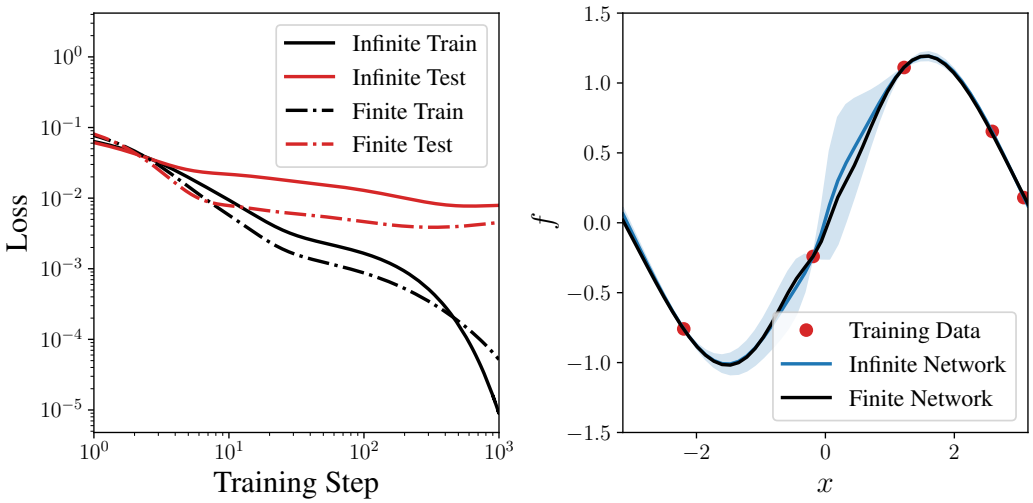


Figure 4.2: Training dynamics of the infinite (linearized) network and the finite one. The shaded area corresponds to the variance of the infinite network. (Reproduced from the interactive Colab notebook.[2] Copyright 2019, Google LLC.)

Moreover, the predictions on the test data seem to be consistent. Since the training dynamics of the linearized model is described by a kernel, it is possible to define the mean and the variance of the network predictions. The exact form of the respective equations is out of the scope of this work. For more details, the reader is referred to the original publication [122].

Furthermore, Figure 4.3 shows the training dynamics and the inference of the infinite (lin-

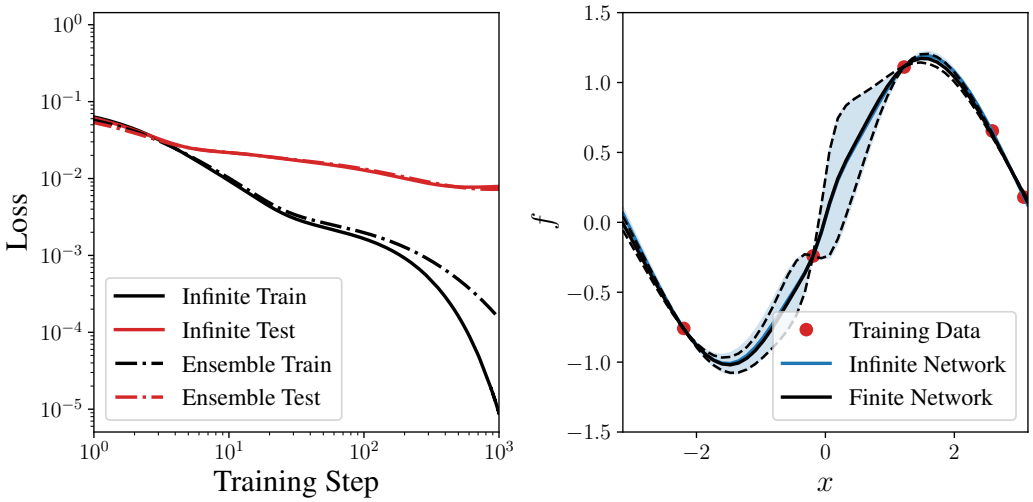


Figure 4.3: Training dynamics of the infinite (linearized) network and an ensemble of 100 finite neural networks. The shaded area corresponds to the variance of the infinite network. The dashed lines represent the variance of the ensemble. (Reproduced from the interactive Colab notebook.[2] Copyright 2019, Google LLC.)

earized) network compared to an ensemble of 100 finite networks. From the figure, one can see that the training dynamics of the linearized network corresponds nicely to one of the ensembles of finite networks. Only for the training loss, some deviations can be seen at the end of training. Moreover, the variance estimated by the ensemble is close to the one calculated by the NTK kernel.

As a final remark, it should be mentioned that NTK theory cannot be considered universal for all architectures and, thus, some layers and models may be operating in different regimes, e.g. feature learning [128]. However, the fact that a kernel may describe the training dynamics provides the means to study the convergence (trainability) and generalization ability of deep NNs by studying the properties of the former. It is the subject of the following sections.

4.3 Explaining the optimization of ultra-wide neural networks

In the previous sections, an explicit expression for the training dynamics of neural networks (NNs) under gradient descent (GD) has been established, and it reads for a general network

$$\frac{\mathrm{d}f\left(\mathcal{X};\boldsymbol{\theta}_t\right)}{\mathrm{d}t} = -\alpha\mathbf{H}^{\infty}\left(\mathcal{X},\mathcal{X}\right)\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right)-\mathcal{Y}\right), \tag{4.18}$$

where $\mathbf{H}^{\infty}\left(\mathcal{X},\mathcal{X}\right)$, denoted in the following by $\mathbf{H}^{\infty}$ to simplify the following expressions, is the neural tangent kernel (NTK). Given that a kernel governs the training dynamics, one may explain the optimization and generalization of wide NNs. This section follows mainly Ref. [123].

The training dynamics of $f\left(\mathcal{X};\boldsymbol{\theta}_t\right)$ are described by a rather simple linear dynamical system; see Equation (4.18). Thus, a standard analysis often used for this kind of system can be applied, where the kernel $\mathbf{H}^{\infty}$ is decomposed by its eigenvalues and eigenvectors as

$$\mathbf{H}^{\infty} = \sum_{i=1}^{N_{\mathrm{Train}}} \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i. \tag{4.19}$$

Here, $\otimes$ denotes the outer product, $\mathbf{v}_1,\ldots,\mathbf{v}_{N_{\mathrm{Train}}}$ are the orthonormal eigenvectors of $\mathbf{H}^{\infty}$, and $\lambda_1,\ldots,\lambda_{N_{\mathrm{Train}}}$ are the corresponding eigenvalues. By employing the eigenvalue decomposition of $\mathbf{H}^{\infty}$, the dynamics of $f\left(\mathcal{X};\boldsymbol{\theta}_t\right)$ can be studied on each eigenvector $\mathbf{v}_i$ separately. For this purpose, an eigenvector $\mathbf{v}_i$ is fixed, and the expression in Equation (4.18) is multiplied by it from both sides

$$\begin{aligned}\mathbf{v}_i^T\frac{\mathrm{d}f\left(\mathcal{X};\boldsymbol{\theta}_t\right)}{\mathrm{d}t} &= -\alpha\mathbf{v}_i^T\mathbf{H}^{\infty}\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right)-\mathcal{Y}\right)\\ &= -\alpha\lambda_i\mathbf{v}_i^T\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right)-\mathcal{Y}\right).\end{aligned} \tag{4.20}$$

From the expression in Equation (4.20), it can be seen that the dynamics of $\mathbf{v}_i^T f\left(\mathcal{X};\boldsymbol{\theta}_t\right)$ depends only on $\mathbf{v}_i^T f\left(\mathcal{X};\boldsymbol{\theta}_t\right)$ and λ_i. Thus, again a simple closed-form solution can be found for the respective one-dimensional ordinary differential equations. The latter reads

$$\mathbf{v}_i^T\left(f\left(\mathcal{X};\boldsymbol{\theta}_t\right)-\mathcal{Y}\right) = \exp\left(-\alpha\lambda_i t\right)\mathbf{v}_i^T\left(f\left(\mathcal{X};\boldsymbol{\theta}_0\right)-\mathcal{Y}\right). \tag{4.21}$$

In the following, the expression in Equation (4.21) will be used to explain the reason for the

excellent convergence of GD algorithms applied to NNs. From the respective expression, it can be seen that the difference between labels and predictions approaches zero $(f(\mathcal{X};\boldsymbol{\theta}_t)-\mathcal{Y})\to 0$ for $t\to\infty$ as fast as $\exp(-\alpha\lambda_i t)\to 0$. Furthermore, the NN optimizes faster if λ_i is large. Note, that here one has to assume that all eigenvalues of the NTK kernel matrix are strictly positive, i.e. $\lambda_i > 0\ \forall\ i\in\{1,\ldots,N_{\text{Train}}\}$ [123]. Fortunately, it can be shown that it holds under reasonably general conditions [129], for a broad range of activation functions provided the sufficient depth of the respective NNs [130]. Given the dependence of the convergence rate on the eigenvalues λ_i of the NTK kernel $\mathbf{H}^{\infty}$ the following intuitive rules can be obtained [123]

- If the labels $\mathcal{Y}$ align with a few eigenvectors corresponding to the large eigenvalues λ_i, the network optimizes at faster rates under GD.
- Suppose the labels $\mathcal{Y}$ are uniformly projected onto the basis of eigenvectors $\{\mathbf{v}_i\}_i$ or they align with eigenvectors corresponding to the small eigenvalues λ_i. In that case, the network optimizes at slower rates under GD.

For the formal proof of the convergence rate of an over-parameterized two-layer NN and experiments with it performed on the MNIST data set [131], see elsewhere [123].

4.4 Explaining the generalization of ultra-wide neural networks

Here, the generalization of ultra-wide neural networks (NNs) is explained by employing the fact that a neural tangent kernel (NTK) can describe the training dynamics of the former. For more details on NTK see Sections 4.1 and 4.2. Especially, it can be used that the prediction of a wide NN can be written as the kernel prediction function in Equation (4.10). Thus, the generalization theory for kernels can be used to examine the generalization behaviour of ultra-wide NNs.

Under certain assumptions such as the non-degeneracy and the noiselessness of the training data, the upper bound for the generalization error of a two-layer ReLU network is [123]

$$\sqrt{\frac{2\mathcal{Y}^T\left(\mathbf{H}^{\infty}\right)^{-1}\mathcal{Y}}{N_{\text{Train}}}}, \tag{4.22}$$

which is computed only from the data $\mathcal{D}=\{\mathbf{x}_i,y_i\}_{i=1}^{N_{\text{Train}}}$ and is independent of the network width. It has been shown in Ref. [123] that this generalization bound can clearly distinguish

between the random data and the actual labels. Note that different from other approaches the generalization bound in Equation (4.22) requires neither the existence of a smaller ground truth network [132] nor the training of the network to get the generalization bound [133–135].

The above analysis assumed the data to be noiseless. However, in a general setting, the data is not noiseless, and one can write

$$y_i = f^* (\mathbf{x}_i) + \epsilon_i, \tag{4.23}$$

where $f^* (\mathbf{x}_i)$ denotes the ground truth for an input $\mathbf{x}_i \in \mathbb{R}^d$, and ϵ_i is a random noise drawn from a standard Gaussian $\mathcal{N} (0, \sigma^2)$ with zero mean and finite variance σ^2. Given the result of Section 4.3, a zero loss can be achieved for $t \to \infty$, i.e.,

$$\| f (\mathbf{x}; \boldsymbol{\theta}) - y \|_2^2 \to 0. \tag{4.24}$$

For general, noisy data, it can be shown that the L_2 error with respect to the ground truth f^*

$$\| f (\mathbf{x}; \boldsymbol{\theta}) - f^* (\mathbf{x}) \|_2^2, \tag{4.25}$$

can be bounded away from zero [124]. In other words, it can be shown that overfitting can be harmful when predicting the ground truth and for the generalization ability of the network.

Nonetheless, provided the L_2 regularization, see Section 3.3.1, it can be shown that the over-parameterized network trained by gradient descent (GD) resembles the solution of kernel ridge regression[3]

$$f (\mathbf{x}; \boldsymbol{\theta}) = \mathbf{H}^\infty (\mathbf{x}, \mathcal{X}) \left(\mathbf{H}^\infty (\mathcal{X}, \mathcal{X}) + \beta \mathbf{I} \right)^{-1} \mathcal{Y}. \tag{4.26}$$

For this specific scenario, under certain assumptions, it has been shown recently that GD converges at the rate of $N_{\text{Train}}^{-d/(4d-2)}$ to the ground truth [124], where N_{Train} is the number of training samples and d is the dimensionality of the input $\mathbf{x} \in \mathbb{R}^d$. It also holds for the early stopping technique, see Section 3.3.2, and shows, in general, the effectiveness of L_2 regularization for noisy data.

[3]Recall that for an unknown function f^*, the kernel ridge regression minimizes a combination of L_2 loss defined over the data set with a weighted penalty based on the squared Hilbert norm

$$f := \underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{N_{\text{Train}}} \left(y_i - f (\mathbf{x}_i; \boldsymbol{\theta}) \right)^2 + \frac{\beta}{2} \| f \|_{\mathcal{H}}^2,$$

where β defines the strength of the regularization and $\mathcal{H}$ is the corresponding reproducing kernel Hilbert space (RKHS).

5 Summary of research

Here, a brief summary of the conducted research is presented. Starting with the derivation of the invariant molecular representation [1], which is essential when constructing accurate and computationally efficient machine-learned interatomic potentials (MLIPs), a particular machine learning (ML) method based on artificial neural networks (NNs) has been conceived [1, 3]. The latter is referred to as Gaussian moment neural network (GM-NN) and is presented in Section 5.1 along with the Gaussian moment (GM) representation. Another intriguing direction in atomistic modeling of molecules and bulk solids is the estimation of the uncertainty of the respective model. It can be used in workflows like active learning (AL). This research question has been motivated by recent applications of GM-NN in Refs. [6, 7], especially since the uncertainty estimation of atomistic NNs is only poorly developed yet. Here, the uncertainty of the atomistic NNs is derived in the framework of optimal experimental design (OED) [2]; see Section 5.2. The limitation of most state-of-the-art MLIPs to predicting scalar properties and the curiosity in investigating magnetic properties of transition metal complexes have inspired the extension of GM-NN to learning symmetric, traceless tensors [4]. It allows the study of complex physical and chemical processes like the spin-phonon relaxation outlined in Section 5.3. Last but not least, Section 5.4 demonstrates the advantage of GM-NN-based interatomic potentials when sampling free-energy surfaces (FESs). Combined with kinetic Monte Carlo (kMC) models, it provides a rigorous estimate of the mobility of a heavy adsorbate, namely nitrogen atoms, on the surface of interstellar dust grains [5].

5.1 Gaussian moments and atomistic neural networks

Here, the formal definition of the Gaussian moment (GM) representation, initially proposed in Ref. [1], is reviewed. Following the molecular representation, the architecture of atomistic neural networks (NNs) based on GMs [1, 3], referred to as Gaussian moment neural networks (GM-NNs), is described. Moreover, its performance on widely used benchmark data sets is presented. The GM-NN source code is available free of charge from gitlab.com/zaverkin_v/gmnn and doi.org/10.18419/darus-2136. The code is licensed under the MIT license.

5.1.1 Molecular representation

The derivation of a molecular representation is of immense importance for designing sample-efficient and accurate machine-learned interatomic potentials (MLIPs), irrespective of the employed machine learning (ML) algorithm. As has been mentioned in Sections 2.1 and 3.3.3, a proper molecular representation has to be invariant with respect to **(1)** global rotations, **(2)** translations, and **(3)** reflections of a molecular structure. Moreover, it has to be invariant with respect to **(4)** the exchange of like atoms. In Section 3.3.3, it has been mentioned that a solution to **(1)**–**(3)** would be the so-called Weyl matrix $\mathbf{\Sigma}_i$ [120]. However, recovering the permutation invariance **(4)** can violate the differentiability of the molecular representation [136]. Thus, it can render the Weyl matrix $\mathbf{\Sigma}_i$ representation inapplicable to problems that involve the computation of the energy gradients like the molecular dynamics (MD) simulations.

Several invariant molecular representations for atomistic ML models have been proposed [27, 32, 33]. First, most of the state-of-the-art approaches presented in literature split an atomic configuration, a molecular or a solid bulk system, into local atomic contributions [31] using hand-crafted atom-centered representations. Some examples of them are atom-centered symmetry functions [31, 137–143], power spectra or bispectra of spherical harmonics [136, 144–148], geometric moments [149–152], and permutation-invariant polynomials [153]. Alternatively, a global description of a molecule can be employed, like the distance or Coulomb matrix of the respective atomic arrangement [154–156] and derivatives of it [157–159]. A different class of models is referred to as message-passing neural networks (NNs), which learn to construct features invariant [160–165] or equivariant [166–168] with respect to rotations of the atomic configuration in a data-driven manner. Finally, most of the methods based on hand-crafted descriptors are limited to only a few atomic species [31, 145, 151, 169, 170], smaller systems [154–156], or fail to approach the accuracy of 1 kcal/mol with respect to the underlying ab-initio method [171] required for chemical applications.

Here, a different approach to the construction of molecular representation is reviewed, the so-called Gaussian moment (GM) representation, recently proposed in Ref. [1]. Similar works can be found for pattern recognition [172–175] and in atomistic ML [149, 150]. They have served as motivation and inspiration for the author's research. The central ingredient for constructing the GM features of an atom i is the definition of pair distance vectors $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ to all neighbors within the cutoff radius $r_{\max}$; see Figure 5.1. Moreover, the pair distances are split into their radial and angular components: $r_{ij} = \|\mathbf{r}_{ij}\|_2$ and $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$. The GMs are computed similarly for both molecular and periodic systems. However, for a periodic system,

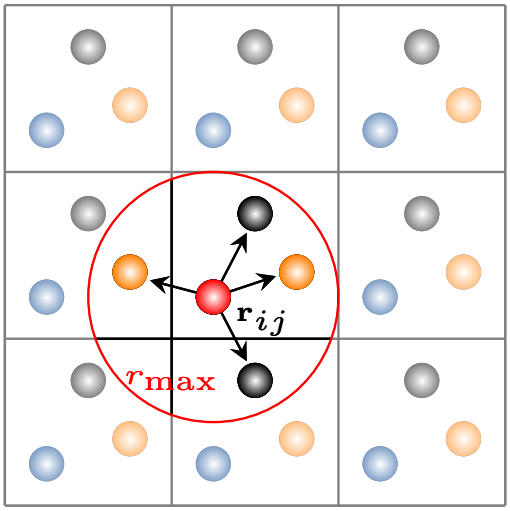


Figure 5.1: Two-dimensional schematic illustrating the local environment of an atom. (Reprinted (adapted) with permission from Ref. [3]. Copyright 2021, American Chemical Society.)

the image atoms have to be considered along with the atoms within the cell.

Employing the radial and angular components of the distance vectors, a tensorial function of a local atomic environment can be written as [1]

$$\mathbf{\Psi}_{i,L,s} = \sum_{j \neq i} R_{Z_i,Z_j,s}\left(r_{ij}, \boldsymbol{\beta}\right) \hat{\mathbf{r}}_{ij}^{\otimes L} \,, \tag{5.1}$$

where $\hat{\mathbf{r}}_{ij}^{\otimes L} = \hat{\mathbf{r}}_{ij} \otimes \cdots \otimes \hat{\mathbf{r}}_{ij}$ is the L-fold outer product of the angular components and $R_{Z_i,Z_j,s}\left(r_{ij}, \boldsymbol{\beta}\right)$ are nonlinear radial functions. The latter can be defined as a single Gaussian [1] centered equidistantly on the radial axis between $r_{\min} = 0.5$ and $r_{\max}$, or a finite sum of Gaussians as in Ref. [3]. In each case, the radial functions are re-scaled by the cosine cutoff function [31] and made trainable to encode the alchemical information into the molecular representation by employing the trainable parameters $\boldsymbol{\beta}$. The former is essential in order to make the molecular representation smooth with respect to the number of atoms within the local neighborhood. At the same time, the latter allows for employing a single NN for all species. However, the exact expression of the respective radial functions is irrelevant here and the reader is referred to the original publications [1, 3] for more details, which are part of this thesis.

As a side remark, it is important to mention that L in Equation (5.1) can be interpreted as the angular-momentum quantum number similar to spherical harmonics. Here, $L = 0$ corre-

sponds to the shape of a spherically symmetric s-orbital, $L = 1$ corresponds to the shape of a p-orbital, $L = 2$ corresponds to that of a d-orbital, and so on. The expression in Equation (5.1) also satisfies **(2)** and **(4)** by definition, while the features invariant to rotations and reflections, **(1)** and **(3)**, can be recovered by computing tensor contractions of $\mathbf{\Psi}_{i,L,s}$ [1], e.g.,

$$G_{i,s_1,s_2,s_3} = \left(\mathbf{\Psi}_{i,1,s_1}\right)_a \left(\mathbf{\Psi}_{i,1,s_2}\right)_b \left(\mathbf{\Psi}_{i,2,s_3}\right)_{a,b}, \tag{5.2}$$

where the Einstein notation has been used, i.e. the right-hand sides are summed over spatial coordinates $a, b \in \{1, 2, 3\}$. Typically, more tensor contractions are used; see the original publications [1,3]. Moreover, a variety of tensor contractions can be written down by employing generating graphs [1]. Recently, the formalism in which invariant features with respect to rotations are built from equivariant ones gained support from the community. The most recent examples are equivariant message passing NN models [167, 168].

Before defining the architecture of the specific ML model based on GMs, one can demonstrate that the respective representation is systematically improvable. That is, the accuracy of a GM-based ML model improves as the number of GMs increases. Moreover, it can be shown that the GM representation has linear runtime and memory complexity with respect to the number of atoms within the local neighborhood N_c, different from the most state-of-the-art approaches. The former characteristic of the GM representation is shown by varying the number of invariant scalars obtained in the GM framework, see Figure 5.2, while the latter requires some further elaborations.

To see which structural assumptions are used to achieve the linear complexity of GMs, let the f be an arbitrary function of radial r_{ij} and angular $\hat{\mathbf{r}}_{ij}$ components of the pair-distance vector $\mathbf{r}_{ij}$. A molecular representation, which describes, for example, a four-body arrangement, can be defined as

$$G_i = \sum_{j_1,j_2,j_3 \neq i} f(r_{ij_1}, r_{ij_2}, r_{ij_3}, \langle \hat{\mathbf{r}}_{ij_1}, \hat{\mathbf{r}}_{ij_2} \rangle, \langle \hat{\mathbf{r}}_{ij_2}, \hat{\mathbf{r}}_{ij_3} \rangle), \tag{5.3}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product of two vectors. Allowing a simpler form for f, one can approach a general expression typically employed in the state-of-the-art approaches

$$G_i = \sum_{j_1,j_2,j_3 \neq i} f\left(r_{ij_1}\right) f\left(r_{ij_2}\right) f\left(r_{ij_3}\right) \langle \hat{\mathbf{r}}_{ij_1}, \hat{\mathbf{r}}_{ij_2} \rangle \langle \hat{\mathbf{r}}_{ij_2}, \hat{\mathbf{r}}_{ij_3} \rangle. \tag{5.4}$$

The computational cost of the expression in Equation (5.4) scales as $\mathcal{O}\left(N_\mathrm{c}^3\right)$, where N_c is

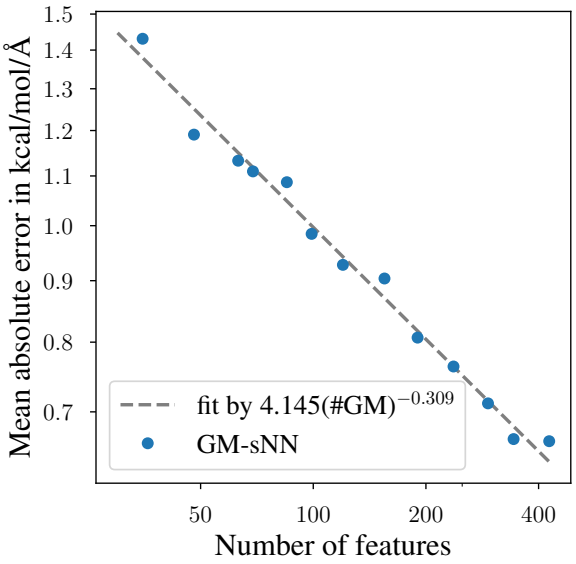


Figure 5.2: Log–log plot of the power-law decay of the mean absolute error (MAE) in predicted forces with an increasing number of Gaussian moment (GM) features. All values are computed for the aspirin molecule from MD17 data set [155, 156, 161]. GM-sNN refers to a shallow neural network (NN) with two hidden layers consisting of 256 and 128 nodes, according to Ref. [1]. (Reprinted (adapted) with permission from Ref. [1]. Copyright 2020, American Chemical Society.)

the number of atoms in the local neighborhood. An example of a representation built from expressions reminiscent of Equation (5.4) is the atom-centered symmetry functions proposed in Ref. [138], where at most three-body arrangements of atoms have been taken into account.

To approach the GM representation, one should add an index s to each function f, i.e. f_s, write the definition of a scalar product explicitly for $\langle \hat{\mathbf{r}}_{ij}, \hat{\mathbf{r}}_{ik} \rangle$, i.e, $\langle \hat{\mathbf{r}}_{ij}, \hat{\mathbf{r}}_{ik} \rangle = \sum_{a=1}^{3} (\hat{\mathbf{r}}_{ij})_a (\hat{\mathbf{r}}_{ik})_a$, and pull out the sum with respect to spatial coordinates. These steps result in an expression similar to the one from Equation (5.2)

$$\begin{aligned} G_{i,s_1,s_2,s_3} &= \sum_{a,b=1}^{3} \left(\sum_{j_1 \neq i} f_{s_1}\left(r_{ij_1}\right) \left(\hat{\mathbf{r}}_{ij_1}\right)_a \right) \left(\sum_{j_3 \neq i} f_{s_3}\left(r_{ij_3}\right) \left(\hat{\mathbf{r}}_{ij_3}\right)_b \right) \\ &\quad \times \left(\sum_{j_2 \neq i} f_{s_2}\left(r_{ij_2}\right) \left(\hat{\mathbf{r}}_{ij_2}\right)_a \left(\hat{\mathbf{r}}_{ij_2}\right)_b \right) \\ &= \sum_{a,b=1}^{3} \left(\mathbf{\Psi}_{i,1,s_1}\right)_a \left(\mathbf{\Psi}_{i,1,s_3}\right)_b \left(\mathbf{\Psi}_{i,2,s_2}\right)_{a,b}, \end{aligned} \tag{5.5}$$

implying the linear scaling of Equation (5.2) with the number of atoms in the local neighborhood. This property of GMs is somewhat arguable if only a few atoms are in the local environment of the central atom i but gains increasing importance as the latter starts to be large, i.e. when employing large cutoff radii.

5.1.2 Gaussian moment neural network

Having reviewed the molecular representation, the particular ML algorithm used to map the atomic coordinates to a scalar property (specifically, the electronic energy from Section 2.1) has to be discussed in more detail. Let the respective mapping be defined as $f : \mathcal{S} \mapsto E \in \mathbb{R}$ where $\mathcal{S} = \{\mathbf{r}_i, Z_i\}_{i=1}^{N_{\text{at}}}$ with $\mathbf{r}_i \in \mathbb{R}^3$ being the spatial coordinates of atom i and $Z_i \in \mathbb{N}$ being the respective atomic number. Here, atom-centered feed-forward neural networks (NNs) are used and therefore, for the total energy, one can write [31]

$$E\left(\mathcal{S}, \boldsymbol{\theta}\right) \approx \sum_{i=1}^{N_{\text{at}}} E_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right), \tag{5.6}$$

justified by the concept of the nearsightedness of electronic matter discussed in Section 2.2. Here, $E_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right)$ is the auxiliary atomic energy predicted by the atom-centered feed-forward NN, and $\boldsymbol{\theta}$ denotes the trainable parameters of the latter and, specific for this work, of the trainable GM representation.

In the following, the architecture of the Gaussian moment neural network (GM-NN) approach [1, 3], employed to predict electronic energies and atomic forces for molecular and material systems, is reviewed. The computational scheme of the GM-NN approach is shown in Figure 5.3. After the local neighborhood of each atom in the structure has been defined and the feature vector has been built, the latter is passed to the fully-connected feed-forward NN, which typically, in the GM-NN framework, consists of two hidden layers and is given by

$$\begin{aligned} y_i = 0.1 \cdot \mathbf{b}^{(3)} + \frac{1}{\sqrt{d_2}}\mathbf{W}^{(3)}\varphi\Bigg(0.1 \cdot \mathbf{b}^{(2)} + \\ \frac{1}{\sqrt{d_1}}\mathbf{W}^{(2)}\varphi\left(0.1 \cdot \mathbf{b}^{(1)} + \frac{1}{\sqrt{d_0}}\mathbf{W}^{(1)}\mathbf{G}_i\right)\Bigg), \end{aligned} \tag{5.7}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases, respectively, and d_l denotes the layer width. Typically, for GM-NN models, one employs $d_0 = 360$ or $d_0 = 910$ for input neurons, depending on the number of invariant features, $d_1 = d_2 = 512$ for hidden neurons, and $d_3 = 1$ for

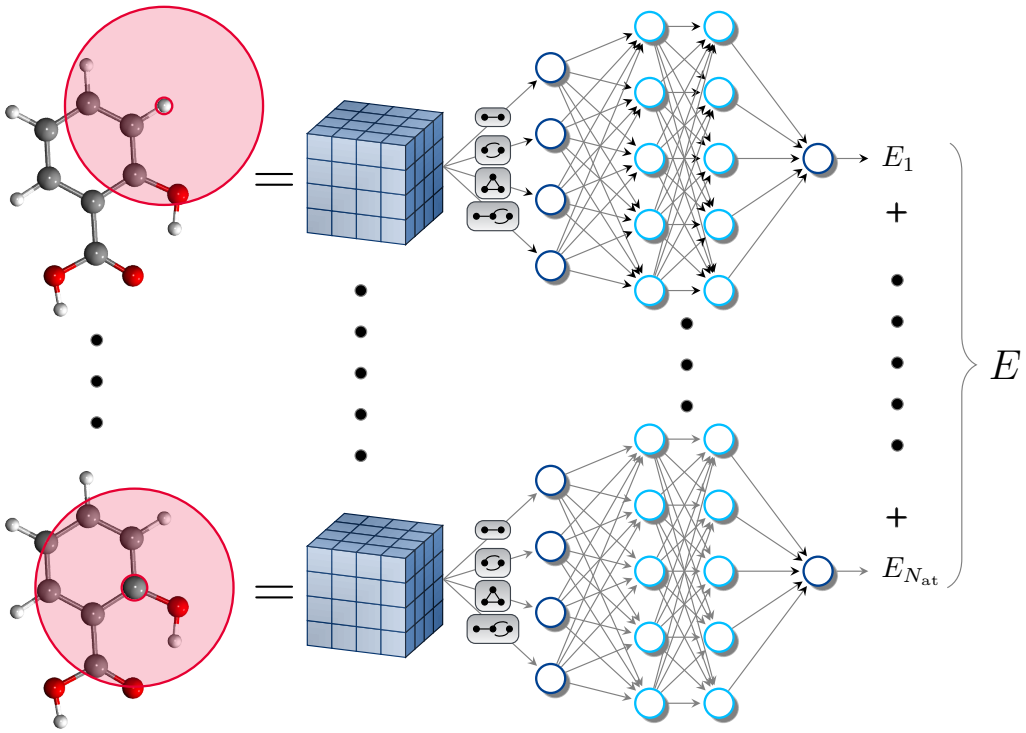


Figure 5.3: Schematic representation of the GM-NN model. (Reprinted (adapted) with permission from Ref. [1]. Copyright 2020, American Chemical Society.)

the single output neuron. Here and in Ref. [3], different from Ref. [1], the so-called neural tangent parameterization (NTP) has been employed [43]; see Section 3.2.3 for more details on NTP. The NTP parameterization improves the convergence and accuracy of GM-NN models. See the original publication attached within this thesis for more details on the network initialization [3].

Like many other state-of-the-art ML approaches, see, for example, Ref. [146], the output of GM-NN is scaled and shifted, see Section 3.2.1, by σ and μ, respectively, i.e.,

$$E_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right) = \sigma y_i + \mu, \tag{5.8}$$

to aid the training process. However, different from other ML approaches, the scale and shift parameters employed in GM-NN depend on the species of the central atom i and are trainable [1]

$$E_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right) = \sigma_{Z_i} y_i + \mu_{Z_i}, \tag{5.9}$$

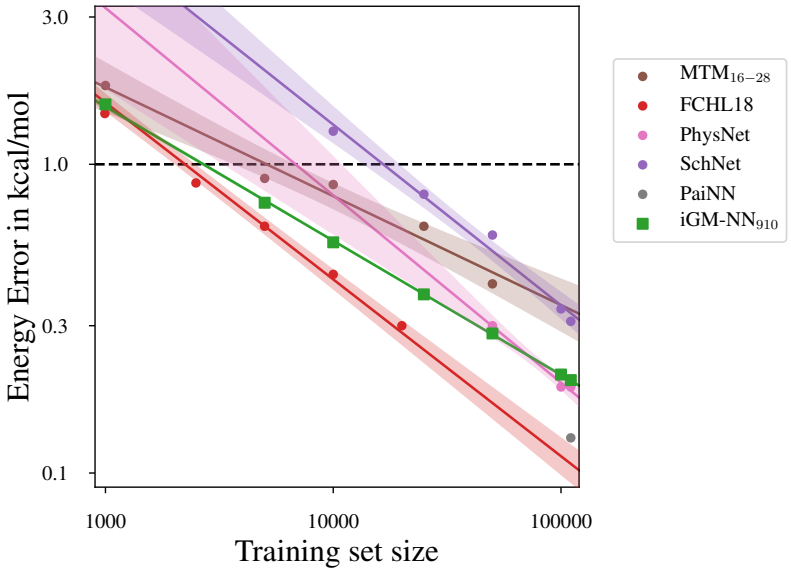


Figure 5.4: Log–log plots of the learning curves for the QM9 data set [176, 177]. The mean absolute error (MAE) of the atomization energy is plotted against the training set size. Linear fits are displayed for clarity and shaded areas denote the 95 % confidence intervals for linear regression. The dashed black line represents the desired accuracy of 1 kcal/mol. (Reprinted (adapted) with permission from Ref. [3]. Copyright 2021, American Chemical Society.)

where σ_{Z_i} and μ_{Z_i} can be initialized as the standard deviation σ and the mean μ of the per-atom average of the reference energies in the training set [1] or by solving a linear regression problem [3]. The latter leads to much better initial models and, thus, improves training considerably.

The parameters of the network and the trainable GM representation, as well as the atomic scale and shifts, are optimized by minimizing the combined loss function

$$\mathcal{L}\left(\boldsymbol{\theta}\right) = \sum_{k=1}^{N_{\text{Train}}} \Bigg[\lambda_E \| E_k^{\text{ref}} - E(\mathcal{S}_k, \boldsymbol{\theta}) \|_2^2 + \frac{\lambda_F}{3N_{\text{at}}^{(k)}} \sum_{i=1}^{N_{\text{at}}^{(k)}} \| \mathbf{F}_{i,k}^{\text{ref}} - \mathbf{F}_i\left(\mathcal{S}_k, \boldsymbol{\theta}\right) \|_2^2 \Bigg], \tag{5.10}$$

where λ_E and λ_F weight the energy and force contributions, respectively. A detailed description of the network training is out of this section's scope, and the reader is referred to Ref. [3] for more details.

Finally, GM-NN models' predictive accuracy and efficiency compared to other state-

of-the-art methods are presented on the widely used benchmark QM9 data set [176, 177]. In Figure 5.4, the best performing iGM-NN$_{910}$ model is compared to linear regression MTM$_{16-28}$ [149, 150] (formally, this approach is linear in trainable parameters only for single-component systems), kernel-based FCHL18 [178], invariant message-passing SchNet [162] and PhysNet [165], as well as equivariant message-passing PaiNN [167] models. Here, iGM-NN stands for the improved network architecture compared to Ref. [1] and 910 denotes the number of GM features.

From Figure 5.4, one can see that the performance of the GM-NN model is among the models with the lowest out-of-sample MAE of atomization energy predictions. The best performing models, FCHL18 trained on 20,000 samples and PaiNN trained on 110,426 configurations, predict the atomization energy with an MAE of 0.30 kcal/mol and 0.13 kcal/mol, respectively. The iGM-NN$_{910}$ model achieves an MAE of 0.38 kcal/mol and 0.20 kcal/mol trained on 25,000 and 110,426 configurations, respectively. Regarding the training time, the GM-NN models outperform the kernel FCHL18 model by a factor of > 60 and the message-passing PhysNet approach by a factor of six. GM-NN models are about five times more efficient than PaiNN concerning the inference time.

5.2 Uncertainty of atomistic neural networks

In Section 5.1, a machine learning (ML) approach for the construction of high-dimensional potential energy surfaces (PESs) based on the atomistic neural networks (NNs) and Gaussian moment (GM) representation has been presented. Among other intriguing directions in the field of molecular or atomistic ML, which will be discussed in the following sections, one can distinguish the generation of highly informative training data sets. It is equivalent to the generation of uniformly accurate machine-learned interatomic potentials (MLIPs). In a general setting, i.e. without the developments of this section, it requires expensive sampling of configurational and chemical space at the reference level of theory, typically employing the density functional theory (DFT), to get a sufficiently comprehensive set of reference energies and atomic forces. Therefore, for a practical application of MLIPs, the calculation of a vast amount of reference structures using DFT or similar is computationally limiting.

One can solve this problem by allowing ML models to detect the most informative structures and perform ab-initio calculations only on them. This can be done in an on-the-fly fashion, where new, non-labelled data points, i.e., data points for which no reference calculations have been performed, are generated during an atomistic simulation, e.g., molecular dynamics

(MD) simulations. Alternatively, the most informative data points can be selected offline on the fixed data set composed of a vast amount of data. Both these possibilities are related to the so-called active learning (AL) [34]. The key quantity of the latter is the query strategy, i.e., an algorithmic criterion for deciding whether a given configuration should be included in the training set.

In the context of MLIPs, it is natural to derive a query strategy for Gaussian process (GP) based models using their predictive variance [179, 180]. However, the first approach to active learning MLIPs has been proposed in Ref. [181], where the model error was evaluated employing ab-initio calculations. For the models based on atomistic NNs, a query strategy can be defined using the so-called query by committee (QbC) approach [182–185]. The latter requires the training of multiple models to get an estimate of the model's uncertainty. Other methods for the NN-based models are the Monte Carlo dropout approach [186, 187] and the distances in feature [185, 188, 189] and latent spaces [187]. Here, a different method for the estimation of the uncertainty of atomistic NNs, derived in the framework of optimal experimental design (OED) [190–192], is outlined [2]. It has been applied earlier, in the context of MLIPs, only to linear-regression-based potentials [149, 150, 193].

In the following, the problem of learning an input-output mapping $\mathcal{X} \rightarrow \mathcal{Y}$ from a set of N_{train} training samples, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{\text{train}}}$, with $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ is considered. Here, specifically, x_i are the atomic coordinates, i.e. $\mathcal{X} \subset \mathbb{R}^{N_{\text{at}} \times 3}$ with N_{at} being the number of atoms, and y_i is the respective scalar property, the electronic energy, i.e. $\mathcal{Y} \subset \mathbb{R}$. Let $\mathcal{P} = \{\mathbf{x}_i\}_{i=1}^{N_{\text{pool}}}$ denote the unlabelled pool. Here, the pool data contains the atomic coordinates, but not the reference energy and atomic forces.

A general parameterized learner can be denoted by $f(\cdot; \boldsymbol{\theta})$ with an output $f(\mathbf{x}_i; \boldsymbol{\theta})$. The parameters of the learner $\boldsymbol{\theta}$ are optimized by minimizing the mean squared loss

$$\mathcal{L} = \frac{1}{N_{\text{Train}}} \sum_{i=1}^{N_{\text{Train}}} \left(f(\mathbf{x}_i; \boldsymbol{\theta}) - y_i \right)^2 . \tag{5.11}$$

Then, the model's output variance can be estimated by employing the parameter covariance matrix $\mathbf{A}$, the so-called Fisher information matrix, computed from the sensitivity of the network output $f(\mathbf{x}_i; \boldsymbol{\theta})$ to the last layer weights $\mathbf{W}$

$$\mathbf{g}(\mathbf{x}_i) = \frac{\partial f(\mathbf{x}_i; \boldsymbol{\theta})}{\partial \mathbf{W}}, \tag{5.12}$$

where $\mathbf{W} \subset \boldsymbol{\theta}$. Formally, for the mean squared error (MSE) loss in Equation (5.11), the Fisher

information matrix can be approximated by

$$\mathbf{A} = \frac{1}{\mathcal{L}} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{W}^2} \approx \frac{1}{\mathcal{L}} \sum_{i=1}^{N_{\text{train}}} \mathbf{g}(\mathbf{x}_i) \otimes \mathbf{g}(\mathbf{x}_i), \tag{5.13}$$

if one assumes that the trained model is already close to the optimal minimum, see Ref. [2]. The estimated output variance of an NN reads [190, 191]

$$\sigma_f^2(\mathbf{x}_i) \approx \mathbf{g}^T(\mathbf{x}_i) \mathbf{A}^{-1} \mathbf{g}(\mathbf{x}_i). \tag{5.14}$$

Now, an analytic expression for the change of the parameter covariance matrix can be derived, after a new training point $\mathbf{x}^*$ has been added to the training data $\mathcal{D}$. In the presented approach the model does not have to be re-trained and the model's expected output variance is estimated based on the respective weights $\mathbf{W}$ only. The formal derivation is out of this section's scope. The resulting expression for the change in the expected model's output variance after adding a new query point $\mathbf{x}^*$ reads

$$\begin{aligned} \langle \Delta \sigma_f^2(\mathbf{x}^*) \rangle_{\mathcal{D}} &\sim \frac{\mathbf{g}^T(\mathbf{x}^*) \mathbf{A}^{-1} \mathbf{g}(\mathbf{x}^*)}{1 + \mathbf{g}^T(\mathbf{x}^*) \mathbf{A}^{-1} \mathbf{g}(\mathbf{x}^*)} \\ &\sim \mathbf{g}^T(\mathbf{x}^*) \mathbf{A}^{-1} \mathbf{g}(\mathbf{x}^*). \end{aligned} \tag{5.15}$$

Here, $\langle \cdot \rangle_{\mathcal{D}}$ denotes the average over the training set $\mathcal{D}$. Moreover, compared to the original publication attached within this thesis [2], the pre-factor has been skipped for simplicity. As a side remark, it should be mentioned that an equivalent expression can be derived by treating the last layer of an NN as a GP with linear kernel, i.e., one can write $f(\mathbf{x}_i; \boldsymbol{\theta}) = \phi(\mathbf{x}_i)^T \mathbf{W}$. Here, the feature map is defined as $\phi(\mathbf{x}_i) = \mathbf{g}(\mathbf{x}_i)$.

The query strategy is defined now by

$$\mathbf{x}^* = \underset{\mathbf{x}^* \in \mathcal{P}}{\operatorname{argmax}} \langle \Delta \sigma_f^2(\mathbf{x}^*) \rangle_{\mathcal{D}}, \tag{5.16}$$

i.e., the data points are selected which reduce the model's expected output variance the most. In general, one can now apply this query strategy to atomistic NNs to get an uncertainty estimate for electronic energies, atomic forces or a combination of them. For analytic expressions in each case, see elsewhere [2]. A schematic of the active learning cycle employed in this section is shown in Figure 5.5. First, an ML model is initialized, i.e. it is trained on the initial, randomly selected training data set. Next, the expression in Equation (5.16) is employed to

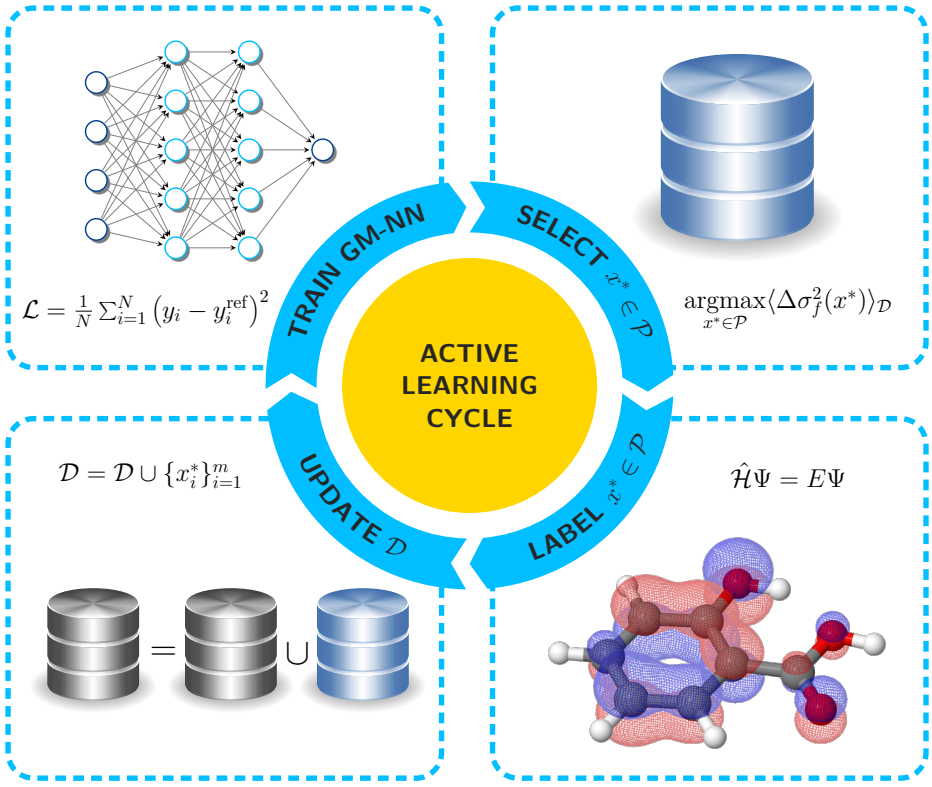


Figure 5.5: Schematic representation of the active learning cycle. (Reprinted (adapted) from Ref. [2]. Copyright 2021, IOP Publishing. Reproduced with permission. All rights reserved.)

choose new training data from a given pool of unlabelled structures $\mathcal{P}$. The ab-initio calculations are performed in the following step to get reference energies and forces. Finally, the training data set $\mathcal{D}$ is updated with the new structures and labels. The model is re-trained. The AL cycle continues until a convergence criterion, or the maximal size of the training data set, is reached. For a more detailed discussion on the AL cycle employed here, see the original publication [2], which is part of this thesis.

In general, the pool data set $\mathcal{P}$ can be generated on the fly during, e.g. a molecular dynamics (MD) simulation. However, to test the performance of the proposed algorithm and the respective uncertainty estimate, the pre-computed QM9 data set [176, 177] has been used. For example, in Figure 5.6, the mean absolute error (MAE, L_1), the root-mean-squared error (RMSE, L_2), and the maximal error (MAXE, L_∞) in predicted atomization energies obtained using the models trained on actively and randomly selected structures are depicted. All results have been obtained by averaging over three independent runs.

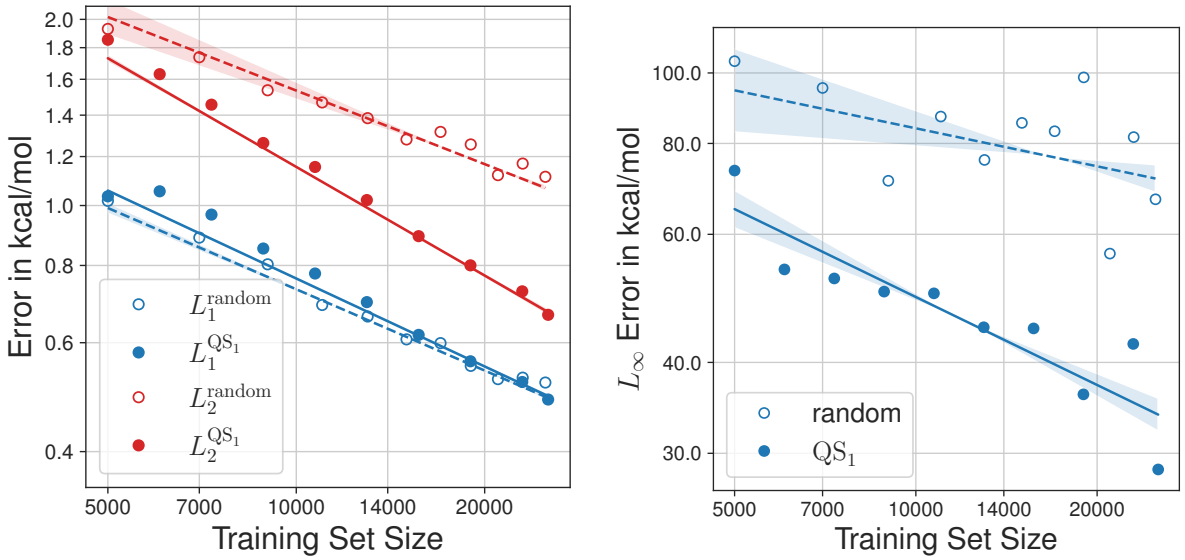


Figure 5.6: Log–log plot of (left) the mean absolute (MAE, L_1) and the root-mean-squared errors (RMSE, L_2), and (right) the maximal error (MAXE, L_∞) in the predicted energies on the QM9 data set. Structures are selected randomly or by using the query strategy based on the energy uncertainty, denoted by QS_1 in the original publication [2]. For each learning curve, a linear fit is plotted in addition. (Reprinted (adapted) from Ref. [2]. Copyright 2021, IOP Publishing. Reproduced with permission. All rights reserved.)

From Figure 5.6, one can see that for MAE, the results are quite low already when trained on randomly selected data points, and AL could not improve on that. However, the maximal error has been reduced by a factor of 2.3 when applying AL compared to the models trained on randomly selected data (MAXE of about 100 kcal/mol). Here, the results obtained for 25,000 training data has been used. It shows that the proposed algorithm selects molecules that better represent unusual molecules and reduces the overall maximal error. Moreover, while the RMSE does not reach the desired accuracy of 1 kcal/mol after training on 25,000 randomly chosen training data, the model trained by employing the query strategy in Equation (5.16) reaches the accuracy of 1 kcal/mol using only about 13,000 structures. Finally, in Ref. [1], one could obtain an RMSE of 0.63 kcal/mol when training on $110,426$ randomly selected structures. Using the AL approach, it is possible to reach an RMSE value of 0.67 kcal/mol using less than a quarter of the number of reference geometries used previously [2].

5.3 Learning symmetric, traceless tensors

Different from selecting the extrapolative configurations in Section 5.2, another intriguing research question in the field of atomistic machine learning (ML) has been investigated, namely the learning of tensorial properties by atom-centered neural networks (NNs). Most state-of-the-art atomistic ML methods are restricted to modeling potential energy surfaces (PESs) [17–27], predicting scalar electronic energies as defined in Section 2.1. However, one should mention that some of them aim to learn properties represented by a vector, e.g., dipole moments [163,165,167,170,183,194,195]. The dipole moments, specifically, can be modeled by atomic charges q_i predicted by, e.g., an atomistic NN and read

$$\boldsymbol{\mu}\left(\mathcal{S},\boldsymbol{\theta}\right)=\sum_{i=1}^{N_{\mathrm{at}}}q_i\left(\mathbf{G}_i,\boldsymbol{\theta}\right)\mathbf{r}_i, \tag{5.17}$$

where $\mathcal{S}=\{\mathbf{r}_i,Z_i\}_{i=1}^{N_{\mathrm{at}}}$ with $\mathbf{r}_i\in\mathbb{R}^3$ being the spatial coordinates of atom i and $Z_i\in\mathbb{N}$ the respective atomic number. Here, $\mathbf{G}_i$ is the molecular representation defined in Section 5.1. Nonetheless, many properties of materials and molecules are described by symmetric tensors, typically of rank two, while, for example, elastic properties of a medium require a fourth-rank tensor. The modeling of tensorial properties requires that the model respects the appropriate geometric transformations, rather than invariance, when, e.g., the reference frame rotates. There are few examples of learning tensorial properties [196–198], and all of them are restricted to kernel-based methods and modeling electric properties.

Learning the magnetic properties of molecules and materials by ML models is rare and restricted mainly to a few recent examples in Refs. [199–201]. In Ref. [199], an ML model has been proposed which reproduces both vibrational and magnetic degrees of freedom, while approaches in Refs. [200, 201] have been developed to model the zero-field splitting (ZFS) tensor $\mathbf{D}$ and the Zeeman-splitting (ZS) tensor $\mathbf{g}$. One of the goals of this work is to extend the existing schemes of learning the structure-property relationships by atomistic NNs to learning tensorial properties for general molecular geometries. Another goal is to apply the proposed formalism to modeling symmetric, traceless magnetic tensors, specifically, the $\mathbf{D}$ tensor by atomistic NNs.

Recently, a particular interest in investigating the magnetic properties of transition metal complexes has been driven by their potential application as single-molecule magnets (SMMs), molecular quantum bits, and spintronic devices [36–39]. The magnetic properties of transition metal complexes are defined by their magnetic anisotropy in the ground spin state [37, 202].

The latter is cased mainly by two fundamental interactions, the Zeeman and the ZFS interactions [35]. The latter is responsible for the spin multiplets (with spin quantum numbers $S \geq 1$) splitting characteristically even without an external magnetic field. The ZFS effect is usually described by a phenomenological spin Hamiltonian [203]

$$\hat{H}_{\mathrm{ZFS}} = \hat{\mathbf{S}} \cdot \mathbf{D} \cdot \hat{\mathbf{S}}, \tag{5.18}$$

where $\hat{\mathbf{S}}$ is a (pseudo) spin operator, see Ref. [35], and $\mathbf{D}$ – a 3×3 symmetric, traceless tensor, usually called the ZFS or $\mathbf{D}$ tensor. Typically, a large axial magnetic anisotropy, which would stabilize the magnetic moment against the thermal fluctuation, is searched for, in the case of transition metal complexes.

Now, the formalism proposed in Ref. [4] for learning symmetric tensors, specifically, the $\mathbf{D}$ tensor, by atomistic NNs is outlined. In general, the machine-learned $\mathbf{D}$ tensor has to satisfy the symmetries and invariances of the reference to allow for efficient learning and excellent generalization ability. The $\mathbf{D}$ tensor is a **(1)** traceless **(2)** symmetric tensor, which implies that $\mathrm{Tr}\,\mathbf{D} = \sum_i D_{ii} = 0$ and $D_{ij} = D_{ji}$, respectively. Additionally, it **(3)** transforms under rotation as $\tilde{\mathbf{D}} = \mathbf{R}\mathbf{D}\mathbf{R}^T$, where $\mathbf{R}$ is an orthogonal matrix, i.e., $\mathbf{D}$ is equivariant to rotations similar to atomic forces, and is **(4)** invariant to translations of the reference frame.

A tensor that satisfies **(1)**–**(3)** can be defined by the outer product of atomic coordinates $\mathbf{r}_i$, similar to the quadrupole moment, employing an atomic quantity m_i predicted by an atomistic NN [4]

$$\mathbf{D}\left(\mathcal{S}, \boldsymbol{\theta}\right) = \sum_{i=1}^{N_{\mathrm{at}}} m_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right) \cdot \left(3\mathbf{r}_i \otimes \mathbf{r}_i - \|\mathbf{r}_i\|_2^2 \mathbb{I}_3\right), \tag{5.19}$$

where $\otimes$ denotes the outer product, $\|\mathbf{r}_i\|_2$ is the length of the respective Cartesian vector, and $\mathbb{I}_3$ is a 3×3 identity matrix. The invariance to translations **(4)** is recovered by shifting $\mathbf{r}_i$ by an arbitrary shift-vector, e.g. $\bar{\mathbf{r}} = 1/N_{\mathrm{at}} \sum_i \mathbf{r}_i$ in Ref. [4]. Moreover, it has been found that normalizing $\mathbf{r}_i$, i.e. $\mathbf{r}_i \rightarrow \mathbf{r}_i / \|\mathbf{r}_i\|_2$, can further improve the learning of machine-learned $\mathbf{D}$ tensor models as well as their performance on the out-of-sample configurations. As a side remark, it should be mentioned that any symmetry of a tensorial property $\mathbf{P}$ can be modeled by the formalism proposed in Ref. [4]. For this purpose, $\mathbf{P}$ is defined employing the output of an NN m_i as

$$\mathbf{P} = \sum_{i=1}^{N_{\mathrm{at}}} m_i\left(\mathbf{G}_i, \boldsymbol{\theta}\right) \mathbf{A}_i, \tag{5.20}$$

where $\mathbf{A}_i$ is a tensor satisfying the symmetry of $\mathbf{P}$. For $\mathbf{D}$ tensors, $\mathbf{A}_i$ has been defined by

$\mathbf{A}_i = 3\mathbf{r}_i \otimes \mathbf{r}_i - \|\mathbf{r}_i\|_2^2 \mathbb{I}_3$ to impose **(1)–(4)**. Alternatively, for a property which, for example, is not traceless, one could use $\mathbf{A}_i = \mathbf{r}_i \otimes \mathbf{r}_i$.

Besides enforcing the equivariance to rotations and other symmetries described above, a network architecture has been proposed for learning symmetric, traceless tensors. The main ideas for the network architecture are similar to those presented in Section 5.1, i.e. the fully-connected feed-forward neural network consisting of two hidden layers from Equation (5.7) has been used. The main differences to Section 5.1 are the scale σ_{Z_i} and shift μ_{Z_i} parameters as well as the loss function. Here, μ_{Z_i} and σ_{Z_i} have been defined as the mean and standard deviation of $\mathbf{D}$ tensor elements with $i \leq j$, excluding one diagonal entry, since the reference tensor is traceless. The corresponding loss function reads[1]

$$\mathcal{L}_{\mathbf{D}}(\boldsymbol{\theta}) = \sum_{k=1}^{N_{\text{Train}}} \|\mathbf{D}_k^{\text{ref}} - \mathbf{D}(\mathcal{S}_k, \boldsymbol{\theta})\|_{\text{F}}^2, \tag{5.21}$$

where only elements with $i \leq j$ have been used during training. A more detailed discussion on the network architecture and training is out of the scope of this section. The reader is referred to the original publication attached within this thesis [4].

In the following, some results of the $\mathbf{D}$ tensor learning have to be elaborated on. For a detailed discussion, see the original publication [4]. In general, the proposed approach achieves a mean absolute error (MAE) of about $0.3 - 0.4$ cm^{-1} for a broad range of systems: $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ [204], $[Fe(TPA)^{Ph}]^-$ [205, 206], and $[Ni(HIM_2{-}py)_2NO_3]^+$ complexes [207]. The $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ unit cell as well as the molecule cut from it, for which reference and machine-learned $\mathbf{D}$ tensors have been computed, are depicted in Figure 5.7. For the unit cells and respective molecular geometries of $[Fe(TPA)^{Ph}]^-$ and $[Ni(HIM_2{-}py)_2NO_3]^+$ see elsewhere [4]. For the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex, specifically, a root-mean-square error (RMSE) value of 1.5 cm^{-1} has been achieved when training on 900 reference configurations. For comparison, in Ref. [201], only an RMSE of 2.2 cm^{-1} could be achieved, even though the data set employed there can be considered less diverse than the one employed in Ref. [4].

Here, a tensorial property, the $\mathbf{D}$ tensor, from Equation (5.18) has been modeled by the ML

[1] Here, the Frobenius norm has been used and is defined for an $n \times m$ matrix $\mathbf{A}$ as

$$\|\mathbf{A}\|_{\text{F}}^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij}^2,$$

where a_{ij} are the elements of the respective matrix.

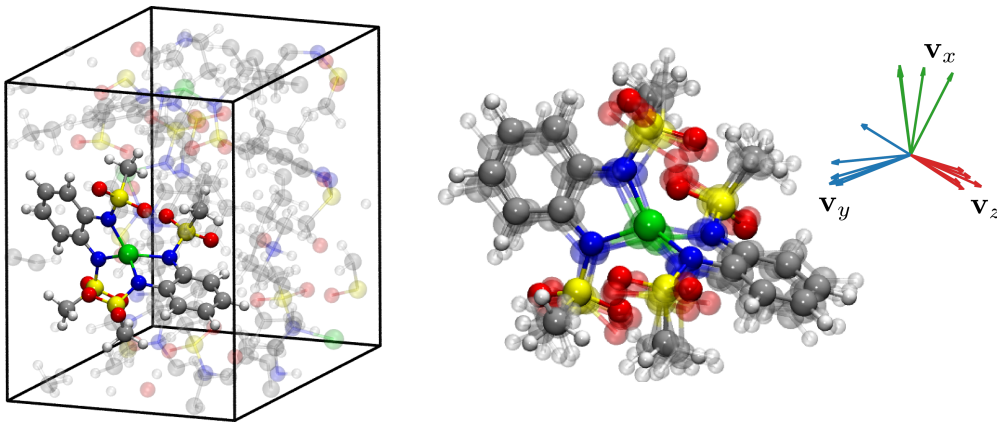


Figure 5.7: Illustration of (left) the structure of the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ unit cell and (right) the molecular geometry cut out of the periodic cell. The respective magnetic axes, i.e. the eigenvectors of $\mathbf{D} = \sum_{i=1}^{3} \lambda_i \mathbf{v}_i \otimes \mathbf{v}_i$, are shown as an inset. (Reprinted (adapted) with permission from Ref. [4]. Copyright 2021, American Chemical Society.)

method described previously. Thus, it is important to consider the correlation of each element of a 3×3 symmetric, traceless tensor predicted by the proposed method with the reference ab-initio values. Figure 5.8 shows the correlation of the reference and machine-learned results of each D_{ij} element in the $\mathbf{D}$ tensor for the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ system with $i \geq j$. Here, only those structures have been used that the ML model has not seen during training. From Figure 5.8, it can be seen that both values, machine-learned and reference ones, are perfectly correlated in accordance with the low MAE (RMSE) value of 0.30 cm^{-1} (0.58 cm^{-1}), obtained by training the model on 2900 reference structures.

As a final remark, it should be mentioned that, probably, the most promising direction for applying the proposed ML approach, in the specific case of learning magnetic anisotropy tensors, is the investigation of the dynamic behaviour of magnetic anisotropy tensors. Specifically, ML aided atomistic simulations can provide a unique insight into spin-phonon relaxation. Thus, they can help design new molecules with improved magnetic properties, e.g., large axial magnetic anisotropy [201]. In Ref. [4], this potential application has been touched by running long molecular dynamics (MD) simulations and extracting velocity-velocity and $\mathbf{D}$-$\mathbf{D}$ autocorrelation functions used subsequently to compute vibrational power spectra. The latter revealed that the most important vibrations in the spin-phonon relaxation process are the low-energy ones [4], predominantly populated under typical experimental conditions.

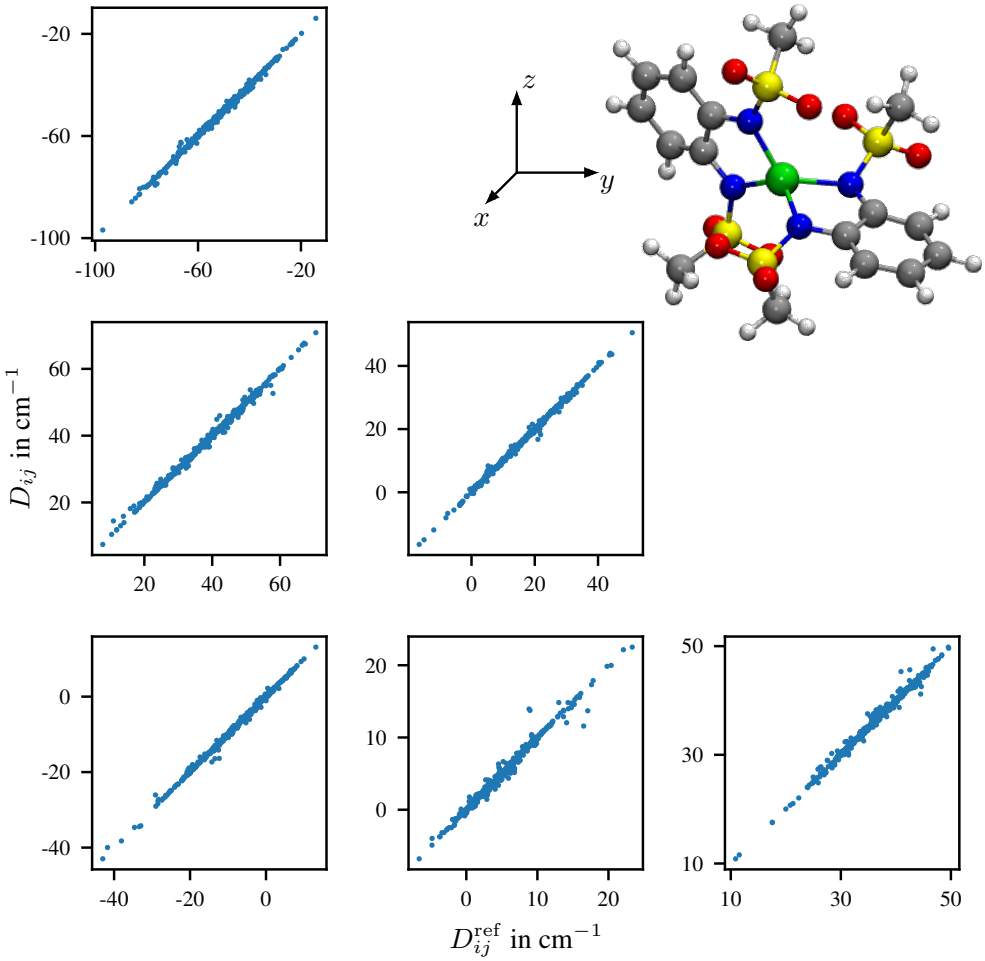


Figure 5.8: Correlation of the machine-learned symmetric elements ($i \geq j$) of the zero-field splitting tensor (D_{ij}) with the corresponding reference values (D_{ij}^{ref}) for all structures in the test $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ data. For all predictions, the model trained on 2900 reference structures was used. The respective coordinate system of the periodic box and an example substructure for which the **D** tensor was computed are shown as an inset. (Reprinted (adapted) with permission from Ref. [4]. Copyright 2021, American Chemical Society.)

5.4 Investigating surface processes in interstellar environments

Given an efficient and accurate potential energy surface (PES), it is possible to study various surface processes in interstellar environments. Four processes are essential for understanding surface chemistry in the interstellar medium (ISM): accretion, diffusion, desorption, and reaction. The adsorption and desorption dynamics of N and H_2 on amorphous solid water (ASW) and CO ice surfaces have been studied recently in Refs. [6, 7] employing the Gaussian moment neural network (GM-NN) approach [1, 3]. These studies provide binding energies, sticking coefficients, and desorption temperatures for the abovementioned systems. They show, specifically, that nitrogen atoms stick efficiently at low temperatures even though the average binding energy of N on ASW is small ($\sim$ 2.9 kJ mol^{-1}, including the zero-point vibrational frequency) in accordance with recent computational [42] and experimental [208] results. Motivated by these results, the mobility of nitrogen atom adsorbed on the ASW surface has been investigated recently [5]. Here, a brief scope of the main developments in Ref. [5] is presented. For more details on adsorption and desorption studies, the reader is referred to original publications [6, 7].

The study of diffusion processes in interstellar environments, i.e. at low temperatures and molecular abundances, requires long time scales, short time steps in direct molecular dynamics (MD), and a very accurate PES. This can be achieved by combining MD simulations on top of a surrogate machine-learned interatomic potential (MLIP), free energy sampling using well-tempered metadynamics, and kinetic Monte Carlo (kMC) simulations based on the minima and saddle points on the free-energy surface (FES). Here, the GM-NN approach [1, 3] has been used to construct an accurate and computationally efficient MLIP. It has been fitted to a training set consisting of 28,715 structures with 3 to 378 atoms each. The respective energies and forces were computed at the PBEh-3c/def2-mSVP level [209]. For more details on the data set employed in this work, see Ref. [6].

Employing the MLIP trained on the heterogeneous training data set, which covers all relevant interactions in the N/ASW system, it was possible to sample the free-energy of a region spanning 800 Å^2 by running well-tempered metadynamics simulations [210–212]. The collective variables for the metadynamics were selected to be the x and y components of the nitrogen atom diffusing on the surface. Figure 5.9 shows (left) the ASW ice surface equilibrated at 50 K with atoms colored according to their z-coordinate and (right) the respective two-dimensional FES for the adsorbed nitrogen atom on the ice surface.

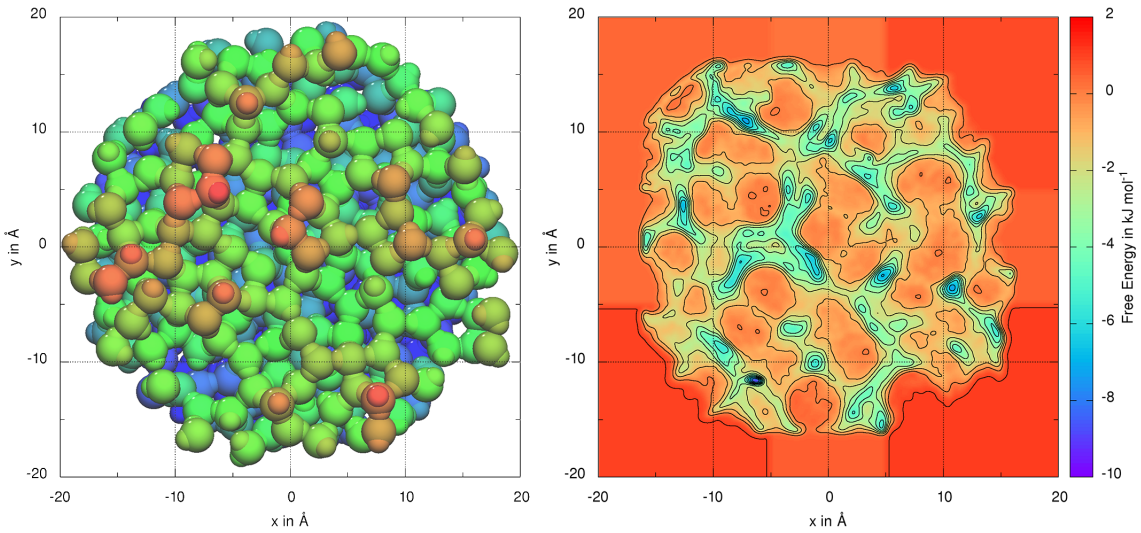


Figure 5.9: (Left) Atomic structure of the amorphous water ice equilibrated at 50 K. All atoms are colored according to their z-coordinate value (surface normal). (Right) The 2D free-energy surface (FES) for the adsorbed nitrogen atom on the ice surface. (Reprinted (adapted) with permission from Ref. [5]. Copyright 2021, Oxford University Press.)

Given the complex topology of the FES, a rather broad distribution of diffusion barriers with a mean of 2.56 kJ mol^{-1} and a standard deviation of 1.72 kJ mol^{-1} has been found, despite the relatively small model surface. Taking into account the mean separation between neighboring minima of 4.2 Å, the pre-exponential factor D_0 of classical Arrhenius equation

$$D\left(T\right) = D_0 \exp\left(-\frac{\Delta F}{RT}\right), \tag{5.22}$$

can be estimated [213] and equals to $D_0 = 1.57 \times 10^{-2}$ cm^{2}s^{-1}, while ΔF is the average diffusion barrier of 2.56 kJ mol^{-1}, T is the surface temperature, and R is the universal gas constant. In the following, a superscript to the values estimated by the topology of the respective ASW and FES is added, i.e. one writes D_0^{avg} and ΔF^{avg}, to compare them with the kMC values.

Because of the broad distribution of activation barriers, distances between neighboring minima, and the number of neighboring states, the quantities derived above, D_0^{avg} and ΔF^{avg}, are suitable for an ideal but not real, rough ice. The broad distribution of diffusion barriers, especially, results in hopping rate constants with variations over several orders of magnitude at low temperatures (from 3.8×10^{-30} to 1.1×10^{11} s^{-1} at 10 K). While a diffusion path may circumvent the higher barriers, the smallest barriers lead to oscillations between neighboring binding sites rather than to real transport of the adsorbate. Thus, a more rigorous description

of diffusion processes is needed.

The kMC approach [214, 215] provides the necessary flexibility, taking into account the connectivity between binding sites and their realistic barriers. A more detailed description of the employed kMC model is out of the scope of this section, and the reader is referred to the original publication [5], which is part of this thesis. However, it should be mentioned that each diffusion path of the kMC simulation has been split into segments to facilitate the convergence of estimated diffusion coefficients [216–218]. Additionally, a new segment has been started by reaching a binding site close to or at the border of the ASW surface if a random number between 0 and 1 was larger than 0.5 to mimic the possible hop out of our boundaries. Note that the metadynamics simulations required direct MD of 6.5×10^{-9} s, while the kMC runs covered 10^{12} s. It resulted in a more efficient sampling of nitrogen atom mobility on ASW.

The kMC model used in this work considers the complex topology of the ASW and FES surfaces and, thus, their roughness. Other important concepts which may influence the mobility of the nitrogen atoms adsorbed on ASW are surface coverage and quantum tunneling. The former has been simulated by removing specific minima for the kMC simulation, assuming that a non-reactive species, e.g., H_2, is already occupying such a state. Therefore, the distribution of binding sites has not changed by the number of adsorbed atoms. To include quantum tunneling into the kMC model, the Eckart and Bell corrections to the rate constants have been introduced. For analytic expressions of the respective tunneling corrections, see Ref. [219].

Figure 5.10 (left) shows the temperature-dependent diffusion coefficients obtained for the bare surface (no more adsorbates) employing kMC simulations along with the respective linear fit. The temperature-dependence of the diffusion constant nicely follows an Arrhenius-like behaviour in Equation (5.22) with $D_0 = (1.65 \pm 0.32) \times 10^{-2}$ cm^2 s^{-1} and $\Delta F = 6.06 \pm 0.04$ kJ mol^{-1}. While the pre-exponential factor is close to D_0^{avg}, the effective diffusion barrier is about 2.4 times larger than the averaged one ΔF^{avg}. It makes the diffusion at low temperatures less probable ($D = (3.5 \pm 1.1) \times 10^{-34}$ cm^2 s^{-1} at 10 K) compared to the estimations based on the FES topology only. Additionally, from Figure 5.10 (left), one can see that the respective diffusion coefficients accounting for tunneling are only marginally larger compared to the ones without tunneling corrections (a factor of merely 1.3–2.8). This result can be expected, taking into account the high mass of the nitrogen atom. It is in accordance with the recent results obtained for a similar atomistic system (O atom adsorbed on ASW) [41] but is at variance with a previous suggestion for oxygen atoms [40].

By running the kMC simulations, it could be observed that the low mobility, as shown in Figure 5.10 (left), is mainly caused by the domination of the energetically deepest binding

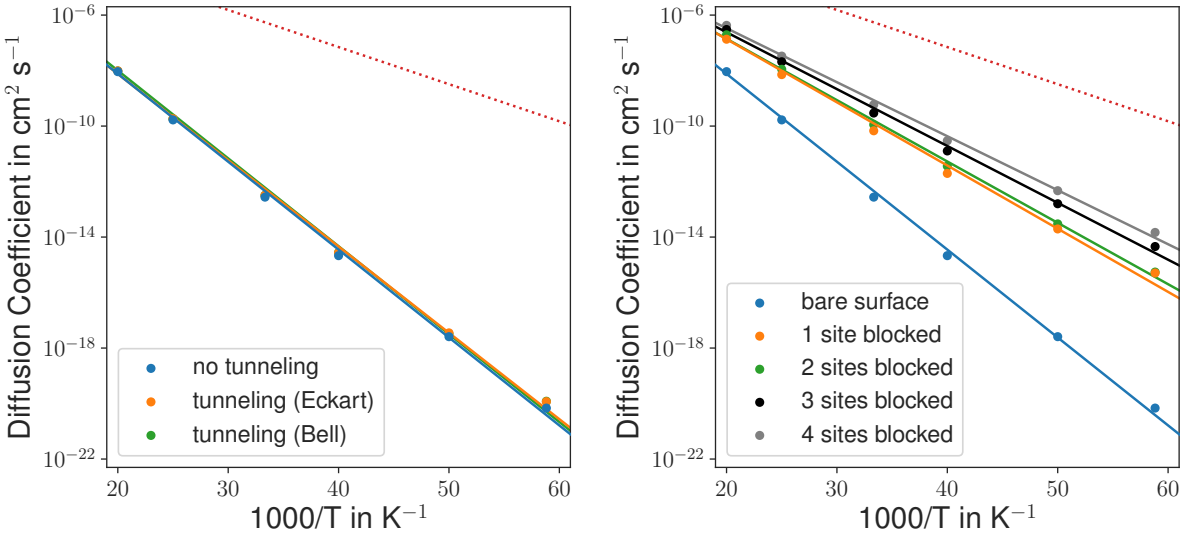


Figure 5.10: (Left) Temperature-dependence of diffusion coefficients (D) for N on amorphous solid water (ASW) for the bare surface with or without tunneling correction. (Right) Temperature-dependence of diffusion coefficients (D) for N on ASW for the bare surface and the surface with the 1–4 deepest sites blocked. Linear fits are displayed for clarity, and the red dotted line represents D^{avg}. (Reprinted (adapted) with permission from Ref. [5]. Copyright 2021, Oxford University Press.)

sites. This correlation has been observed earlier [220–222] and recently has been studied specifically on the N/ASW system [5]. It has been shown [5] that if the N adsorbate finds one of the deep binding sites (see Figure 5.9), it stays there for a long time (93 %–99.99 % of the whole simulation time, depending on the surface temperature). The employed surface model in Figure 5.9 is, with about 800 Å^2, comparably small and certainly very compact and smooth. Since in real ASW much deeper pores are expected [223], i.e. even stronger binding sites can be observed, the diffusion on a realistic surface can be expected to be even slower.

However, in an experimental setup or interstellar clouds, single nitrogen atom diffusion is highly unlikely as higher surface coverage is expected. One can expect that, under ISM conditions, deep sites are covered by other species, i.e., H_2 (H_2 is $\sim 10^5$ more abundant than N in the gas phase [224, 225]). It raises the question of the influence of additional inert species on the mobility of the adsorbed nitrogen atom, which may block the deeper binding sites. This question has been addressed by excluding the deepest binding sites, which mimics their occupation by some inert chemical species. Blocking 1 to 4 of the deepest sites results in much higher values for D of 8.2×10^{-26} to 9.0×10^{-23} cm^2 s^{-1} at 10 K, respectively. The temperature dependence of D with the deepest sites blocked is depicted in Figure 5.10 (right).

Moreover, a decrease of the effective diffusion barrier from $\Delta F = 6.06 \pm 0.04$ kJ mol^{-1} to $\Delta F = 3.73 \pm 0.03$ kJ mol^{-1} has been observed. Hence, the latter is closer to the averaged diffusion barrier obtained earlier ($\Delta F^{\text{avg}} = 2.56 \pm 1.72$ kJ mol^{-1}).

Finally, while most studies in the literature consider hopping rates, the work presented here provides the community with more rigorous estimates of the mobility of the N atom as a paradigmatic case for light and weakly bound adsorbates. In total, it has been found that the nitrogen atom is hardly able to diffuse on bare ASW surfaces. Still, surface coverage may change the mobility of N on ASW surfaces considerably, increasing the effective diffusion coefficient over 9–12 orders of magnitude.

6 Conclusion and Outlook

The presented work, in general, addresses machine learning (ML) methods applied in the context of atomistic simulations for describing various chemical and physical phenomena. More specifically, it aims to advance the recent development of ML-based methods, which typically provide surrogate potential energy surfaces (PESs), essential for most computational chemistry applications. During the last decades, machine-learned interatomic potentials (MLIPs) have risen in popularity since they promise to accurately predict materials and molecular properties while minimizing the demand on computationally inefficient ab-initio calculations. Here, a broad range of research questions has been covered, from encoding fundamental symmetries of an atomistic system into an ML-based model to applying developed methods in real-world simulations.

ML-based models applied to atomistic systems (MLIPs) can have limited accuracy and transferability due to missing fundamental symmetries of scalar properties like the electronic energy. These are the invariance with respect to translations, rotations, and reflections of the whole system. Additionally, the electronic energy similar to other scalar properties is invariant with respect to permutations of like atoms. Here, an invariant molecular fingerprint has been derived and is referred to as Gaussian moment (GM) representation [1]. This fingerprint is built from the atomic distance vectors different from most approaches in the literature, which map a 3D structure onto a 2D space, i.e. employ distances and angles between atoms [31]. Primarily, it has been shown that using the directional information leads to a systematically improvable molecular representation and ML-based methods with accuracy and sample efficiency comparable to or better than other state-of-the-art approaches.

Recently, building invariant features with respect to rotations from equivariant ones has gained support from the community. This resulted in new methods appearing, which mainly use message passing neural network (NN) architectures [167, 168]. At the same time, the idea of applying equivariant transformations to the inputs and subsequently building features invariant to rotations form the basis of much earlier works in image analysis [172–175] and more recent works in the atomistic modeling community in Refs. [149, 150] followed by Ref. [1].

Combining the GM representation with artificial NNs, the Gaussian moment neural net-

work (GM-NN) has been conceived [1,3]. Here, an architecture for an atomistic NN employing a variety of recent developments in the NN community, like the neural tangent parameterization (NTP) [43], has been proposed. NTP improves the accuracy and efficiency of the respective model considerably. As the GM representation derived in Ref. [1] depends on atomic species, GM-NN could be designed such that only a single NN has to be trained, in contrast to using an individual NN for each species as frequently done in the literature [31]. Additionally, trainable scale and shift parameters of the atomic energy have been defined and initialized by solving a linear regression problem to aid the training process. Moreover, one should mention that the parameters of the GM representation are also trainable, which improves the predictive power of GM-NN-based PESs. Finally, as a final remark on GM-NN-based PES models, the respective approach leads to overall robust and transferable potentials that facilitate the application of GM-NN-based models during real-time atomistic simulations.

Atomistic ML is about designing algorithms that solve specific problems by learning from data. Thus, generating a comprehensive training set that covers the relevant part of the configurational and chemical space is essential for obtaining a uniformly accurate MLIP with excellent generalization ability. Here, an active learning (AL) algorithm is proposed, which uses the uncertainty of NNs to detect the most informative or extrapolative configurations from an unlabeled pool data set [2]. More specifically, it is shown that the uncertainty of an atomistic NN can be derived in the optimal experimental design (OED) framework. It is referred to as the model's expected output variance. The latter correlates well with the actual error, providing linear correlation coefficients on par with other well-established approaches as the query by committee (QbC) method [2]. In Ref. [2] and Section 5.2, it has been shown that the proposed method leads to a considerable reduction of the training set size and, at the same time, to a decrease of the generalization error.

Learning scalar properties of an atomistic system, like the electronic energy, is now an established task in the ML community. However, many important properties of molecules and materials are tensorial. Thus, the presented work extends the GM-NN approach to modeling tensorial properties. The approach proposed in Ref. [4] and outlined in Section 5.3 includes encoding the symmetries of the respective tensorial property by introducing symmetry-equivalent re-scaling of the network's output. It has been proven to provide robust models with excellent generalization ability and resulted in a mean absolute error (MAE) of 0.3–0.4 cm^{-1} when the GM-NN models have been trained on zero-field splitting (ZFS) $\mathbf{D}$ tensors. Moreover, it has been shown that in combination with MLIPs based on the GM representation, the surrogate ZFS models can be used to investigate complex physical processes relevant for the dynamics

of magnetic anisotropy tensors, like the spin-phonon relaxation.

Motivated by the efficiency and out-of-sample accuracy of GM-NN-based PESs, surface processes in the interstellar medium (ISM) have been investigated. In Refs. [6, 7], the adsorption and desorption dynamics of N and H_2 on different surfaces have been investigated, providing binding energies, sticking coefficients, and desorption temperatures. In Ref. [5], the mobility of a heavy adsorbate, namely nitrogen atoms, on the surface of amorphous solid water (ASW) has been investigated. The study has been conducted by combining molecular dynamics (MD) simulations on top of a surrogate PES, free energy sampling using well-tempered metadynamics, and kinetic Monte Carlo (kMC) simulations based on the minima and saddle points on the free-energy surface (FES). The use of realistic diffusion barriers and the connectivity of binding sites, including their broad distributions, has resulted in the diffusion coefficient of nitrogen atoms on ASW of $D = (3.5 \pm 1.1) \times 10^{-34}$ cm^{2}s^{-1} at 10 K. This implies that diffusion of the nitrogen atom, as a paradigmatic case for light and weakly bound adsorbates, is effectively suppressed on bare ice surfaces at 10 K. However, surface coverage has a strong effect and modulates the value of the diffusion coefficient over 9–12 orders of magnitude. Ref. [5] has shown that tunneling has only a marginal impact on nitrogen mobility.

In summary, in this thesis, an ML method based on the GM representation and atomistic NNs has been developed and applied to modeling scalar and tensorial properties of molecular and bulk solid matter. Nevertheless, one should notice that, in a global sense, the proposed approaches only touch the complex field of ML-based modeling of physical and chemical processes. The presented research can directly introduce many intriguing questions. An example is the automated generation of training data set in an on-the-fly fashion, sensitive to the specific setting or research question. Next, the application of ML approaches to real-world problems in chemistry and physics has to be taken more seriously since currently, they are typically applied to benchmark systems only and do not find practical applications, except for relatively few examples. Finally, directions such as constructing explainable MLIPs are crucial since they aim to provide a unique insight into what the ML-based models learn the most from.

Bibliography

[1] V. Zaverkin and J. Kästner: *Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials*. J. Chem. Theory Comput. **16** (8), 5410–5421 (2020)

[2] V. Zaverkin and J. Kästner: *Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design*. Mach. Learn.: Sci. Technol. **2** (3), 035009 (2021)

[3] V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner: *Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments*. J. Chem. Theory Comput. **17** (10), 6658–6670 (2021)

[4] V. Zaverkin, J. Netz, F. Zills, A. Köhn, and J. Kästner: *Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments*. J. Chem. Theory Comput. **18** (1), 1–12 (2022)

[5] V. Zaverkin, G. Molpeceres, and J. Kästner: *Neural-network assisted study of nitrogen atom dynamics on amorphous solid water - II. Diffusion*. Mon. Not. R. Astron. Soc. **510** (2), 3063–3070 (2022)

[6] G. Molpeceres, V. Zaverkin, and J. Kästner: *Neural-network assisted study of nitrogen atom dynamics on amorphous solid water - I. adsorption and desorption*. Mon. Not. R. Astron. Soc. **499** (1), 1373–1384 (2020)

[7] G. Molpeceres, V. Zaverkin, N. Watanabe, and J. Kästner: *Binding energies and sticking coefficients of H_2 on crystalline and amorphous CO ice*. Astron. Astrophys. **648**, A84 (2021)

[8] V. Zaverkin, D. Holzmüller, R. Schuldt, and J. Kästner: *Predicting properties of periodic systems from cluster data: A case study of liquid water*. J. Chem. Phys. **156** (11), 114103 (2022)

[9] D. Holzmüller, V. Zaverkin, J. Kästner, and I. Steinwart: *A Framework and Benchmark for Deep Batch Active Learning for Regression*. ArXiv **abs/2203.09410** (2022)

[10] V. Zaverkin, D. Holzmüller, I. Steinwart, and J. Kästner: *Exploring Chemical and Conformational Spaces by Batch Mode Deep Active Learning*. Digital Discovery **submitted** (2022)

[11] V. Zaverkin, T. Lamberts, M. N. Markmeyer, and J. Kästner: *Tunnelling dominates the reactions of hydrogen atoms with unsaturated alcohols and aldehydes in the dense medium*. Astron. Astrophys. **617**, A25 (2018)

[12] V. Zaverkin and J. Kästner: *Chapter 7 Instanton Theory to Calculate Tunnelling Rates and Tunnelling Splittings*. *Tunnelling in Molecules: Nuclear Quantum Effects from Bio to Physical Chemistry*, pp. 245–260. The Royal Society of Chemistry (2021)

[13] E. Schrödinger: *An Undulatory Theory of the Mechanics of Atoms and Molecules*. Phys. Rev. **28** (6), 1049–1070 (1926)

[14] P. A. M. Dirac: *Quantum mechanics of many-electron systems*. Proc. R. Soc. Lond. A **123** (792), 714–733 (1929)

[15] M. Born and R. Oppenheimer: *Zur Quantentheorie der Molekeln*. Ann. Phys. **389** (20), 457–484 (1927)

[16] A. D. Mackerell Jr.: *Empirical force fields for biological macromolecules: Overview and issues*. J. Comput. Chem. **25** (13), 1584–1604 (2004)

[17] J. Behler: *Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations*. Phys. Chem. Chem. Phys. **13** (40), 17930–17955 (2011)

[18] J. Behler: *Perspective: Machine learning potentials for atomistic simulations*. J. Chem. Phys. **145** (17), 170901 (2016)

[19] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen et al.: *Performance and Cost Assessment of Machine Learning Interatomic Potentials*. J. Phys. Chem. A **124** (4), 731–745 (2020)

[20] O. T. Unke, D. Koner, S. Patra, S. Käser, and M. Meuwly: *High-dimensional potential energy surfaces for molecular simulations: from empiricism to machine learning*. Mach. Learn.: Sci. Technol. **1** (1), 13001 (2020)

[21] T. Mueller, A. Hernandez, and C. Wang: *Machine learning for interatomic potential models*. J. Chem. Phys. **152** (5), 50902 (2020)

[22] P. O. Dral: *Quantum Chemistry in the Age of Machine Learning*. J. Phys. Chem. Lett. **11** (6), 2336–2347 (2020)

[23] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti et al.: *Gaussian Process Regression for Materials and Molecules*. Chem. Rev. **121** (16), 10073–10141 (2021)

[24] S. Manzhos and T. Carrington: *Neural Network Potential Energy Surfaces for Small Molecules and Reactions*. Chem. Rev. **121** (16), 10187–10217 (2021)

[25] A. M. Miksch, T. Morawietz, J. Kästner, A. Urban, and N. Artrith: *Strategies for the construction of machine-learning potentials for accurate and efficient atomic-scale simulations*. Mach. Learn.: Sci. Technol. **2** (3), 31001 (2021)

[26] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky et al.: *Machine Learning Force Fields*. Chem. Rev. **121** (16), 10142–10186 (2021)

[27] P. Friederich, F. Häse, J. Proppe, and A. Aspuru-Guzik: *Machine-learned potentials for next-generation matter simulations*. Nat. Mater. **20** (6), 750–761 (2021)

[28] K. Hornik: *Approximation capabilities of multilayer feedforward networks*. Neural Netw. **4** (2), 251–257 (1991)

[29] T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren: *Neural network models of potential energy surfaces*. J. Chem. Phys. **103** (10), 4129–4137 (1995)

[30] S. Lorenz, A. Groß, and M. Scheffler: *Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks*. Chem. Phys. Lett. **395** (4), 210–215 (2004)

[31] J. Behler and M. Parrinello: *Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces*. Phys. Rev. Lett. **98** (14), 146401 (2007)

[32] M. J. Willatt, F. Musil, and M. Ceriotti: *Atom-density representations for machine learning*. J. Chem. Phys. **150** (15), 154110 (2019)

[33] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi et al.: *Physics-Inspired Structural Representations for Molecules and Materials*. Chem. Rev. **121** (16), 9759–9815 (2021)

[34] B. Settles: *Active Learning Literature Survey*. Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2009)

[35] L. F. Chibotaru and L. Ungur: *Ab initio calculation of anisotropic magnetic properties of complexes. I. Unique definition of pseudospin Hamiltonians and their derivation*. J. Chem. Phys. **137** (6), 064112 (2012)

[36] D. Gatteschi, R. Sessoli, and J. Villain: *Molecular Nanomagnets*. Oxford University Press (2006)

[37] G. A. Craig and M. Murrie: *3d single-ion magnets*. Chem. Soc. Rev. **44** (8), 2135–2147 (2015)

[38] A. Gaita-Ariño, F. Luis, S. Hill, and E. Coronado: *Molecular spins for quantum computation*. Nat. Chem. **11** (4), 301–309 (2019)

[39] S. Sanvito: *Molecular spintronics*. Chem. Soc. Rev. **40** (6), 3336–3355 (2011)

[40] M. Minissale, E. Congiu, S. Baouche, H. Chaabouni, A. Moudens et al.: *Quantum Tunneling of Oxygen Atoms on Very Cold Surfaces*. Phys. Rev. Lett. **111** (5), 053201 (2013)

[41] M. Pezzella, O. T. Unke, and M. Meuwly: *Molecular Oxygen Formation in Interstellar Ices Does Not Require Tunneling*. J. Phys. Chem. Lett. **9** (8), 1822–1826 (2018)

[42] T. Shimonishi, N. Nakatani, K. Furuya, and T. Hama: *Adsorption energies of carbon, nitrogen, and oxygen atoms on the low-temperature amorphous water ice: A systematic estimation from quantum chemistry calculations*. Astrophys. J. **855** (1), 27 (2018)

[43] A. Jacot, F. Gabriel, and C. Hongler: *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*. *NeurIPS*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi et al., volume 31, pp. 8580–8589. Curran Associates, Inc. (2018)

[44] W. Kohn: *Density Functional and Density Matrix Method Scaling Linearly with the Number of Atoms*. Phys. Rev. Lett. **76** (17), 3168–3171 (1996)

[45] E. Prodan and W. Kohn: *Nearsightedness of electronic matter*. Proc. Natl. Acad. Sci. **102** (33), 11635–11638 (2005)

[46] F. Jensen: *Introduction to Computational Chemistry*. John Wiley & Sons, Inc. (2006)

[47] A. W. Jasper, C. Zhu, S. Nangia, and D. G. Truhlar: *Introductory lecture: Nonadiabatic effects in chemical dynamics*. Faraday Discuss. **127** (0), 1–22 (2004)

[48] S. Fias, F. Heidar-Zadeh, P. Geerlings, and P. W. Ayers: *Chemical transferability of functional groups follows from the nearsightedness of electronic matter*. Proc. Natl. Acad. Sci. **114** (44), 11633–11638 (2017)

[49] W. Yang: *Direct calculation of electron density in density-functional theory*. Phys. Rev. Lett. **66** (11), 1438–1441 (1991)

[50] W. Yang and T.-S. Lee: *A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules*. J. Chem. Phys. **103** (13), 5674–5678 (1995)

[51] G. Galli and M. Parrinello: *Large scale electronic structure calculations*. Phys. Rev. Lett. **69** (24), 3547–3550 (1992)

[52] W. Kohn: *Density functional/Wannier function theory for systems of very many atoms*. Chem. Phys. Lett. **208** (3), 167–172 (1993)

[53] P. Ordejón, D. A. Drabold, R. M. Martin, and M. P. Grumbach: *Linear system-size scaling methods for electronic-structure calculations*. Phys. Rev. B **51** (3), 1456–1476 (1995)

[54] F. Mauri, G. Galli, and R. Car: *Orbital formulation for electronic-structure calculations with linear system-size scaling*. Phys. Rev. B **47** (15), 9973–9976 (1993)

[55] X.-P. Li, R. W. Nunes, and D. Vanderbilt: *Density-matrix electronic-structure method with linear system-size scaling*. Phys. Rev. B **47** (16), 10891–10894 (1993)

[56] P. Hohenberg and W. Kohn: *Inhomogeneous Electron Gas*. Phys. Rev. **136** (3B), B864–B871 (1964)

[57] F. Bloch: *Bemerkung zur Elektronentheorie des Ferromagnetismus und der elektrischen Leitfähigkeit*. Z. Physik **57** (7), 545 (1929)

[58] P. A. M. Dirac: *Note on Exchange Phenomena in the Thomas Atom*. Proc. Cambridge Phil. Soc. **26** (3), 376–385 (1930)

[59] W. Kohn and L. J. Sham: *Self-Consistent Equations Including Exchange and Correlation Effects*. Phys. Rev. **140** (4A), A1133–A1138 (1965)

[60] A. D. Becke: *Density-functional exchange-energy approximation with correct asymptotic behavior*. Phys. Rev. A **38** (6), 3098–3100 (1988)

[61] J. P. Perdew, K. Burke, and M. Ernzerhof: *Generalized Gradient Approximation Made Simple*. Phys. Rev. Lett. **77** (18), 3865–3868 (1996)

[62] J. P. Perdew, M. Ernzerhof, and K. Burke: *Rationale for mixing exact exchange with density functional approximations*. J. Chem. Phys. **105** (22), 9982–9985 (1996)

[63] M. J. Gillan, D. Alfè, and A. Michaelides: *Perspective: How good is DFT for water?* J. Chem. Phys. **144** (13), 130901 (2016)

[64] C. Adamo, M. Cossi, G. Scalmani, and V. Barone: *Accurate static polarizabilities by density functional theory: assessment of the PBE0 model*. Chem. Phys. Lett. **307** (3), 265–271 (1999)

[65] A. D. Becke: *Density-functional thermochemistry. III. The role of exact exchange*. J. Chem. Phys. **98** (7), 5648–5652 (1993)

[66] P. J. Stephens, F. J. Devlin, C. F. Chabalowski, and M. J. Frisch: *Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields*. J. Phys. Chem. **98** (45), 11623–11627 (1994)

[67] R. Car and M. Parrinello: *Unified Approach for Molecular Dynamics and Density-Functional Theory*. Phys. Rev. Lett. **55** (22), 2471–2474 (1985)

[68] M. E. Tuckerman: *Ab initio molecular dynamics: basic concepts, current trends and novel applications*. J. Phys. Condens. Matter **14** (50), R1297–R1355 (2002)

[69] R. Iftimie, P. Minary, and M. E. Tuckerman: *Ab initio molecular dynamics: Concepts, recent developments, and future trends*. Proc. Natl. Acad. Sci. **102** (19), 6654–6659 (2005)

[70] A. A. Hassanali, J. Cuny, V. Verdolino, and M. Parrinello: *Aqueous solutions: state of the art in ab initio molecular dynamics*. Philos. Trans., Math. Phys. Eng. Sci. **372** (2011), 20120482 (2014)

[71] N. Artrith and J. Behler: *High-dimensional neural network potentials for metal surfaces: A prototype study for copper*. Phys. Rev. B **85** (4), 045439 (2012)

[72] S. Tovey, A. Narayanan Krishnamoorthy, G. Sivaraman, J. Guo, C. Benmore et al.: *DFT Accurate Interatomic Potential for Molten NaCl from Machine Learning*. J. Phys. Chem. C **124** (47), 25760–25768 (2020)

[73] K. Gubaev, Y. Ikeda, F. Tasnádi, J. Neugebauer, A. V. Shapeev et al.: *Finite-temperature interplay of structural stability, chemical complexity, and elastic properties of bcc multicomponent alloys from ab initio trained machine-learning potentials*. Phys. Rev. Materials **5** (7), 073801 (2021)

[74] A. Forslund, X. Zhang, B. Grabowski, A. V. Shapeev, and A. V. Ruban: *Ab initio simulations of the surface free energy of TiN(001)*. Phys. Rev. B **103** (19), 195428 (2021)

[75] L. Zhang, H. Wang, R. Car, and W. E: *Phase Diagram of a Deep Potential Water Model*. Phys. Rev. Lett. **126** (23), 236001 (2021)

[76] H. Guo, Q. Wang, A. Stuke, A. Urban, and N. Artrith: *Accelerated Atomistic Modeling of Solid-State Battery Materials With Machine Learning*. Front. Energy Res. **9**, 695902 (2021)

[77] V. L. Deringer, N. Bernstein, G. Csányi, C. Ben Mahmoud, M. Ceriotti et al.: *Origins of structural and electronic transitions in disordered silicon*. Nature **589** (7840), 59–64 (2021)

[78] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu et al.: *Natural Language Processing (Almost) from Scratch*. J. Mach. Learn. Res. **12** (76), 2493–2537 (2011)

[79] A. Krizhevsky, I. Sutskever, and G. E. Hinton: *ImageNet Classification with Deep Convolutional Neural Networks*. *NeurIPS*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, volume 25, pp. 1097–1105. Curran Associates, Inc. (2012)

[80] Y. Kim: *Convolutional Neural Networks for Sentence Classification*. *Proceedings of EMNLP*, pp. 1746–1751. Association for Computational Linguistics (2014)

[81] K. Simonyan and A. Zisserman: *Very Deep Convolutional Networks for Large-Scale Image Recognition*. *ICLR*, edited by Y. Bengio and Y. LeCun (2015)

[82] K. He, X. Zhang, S. Ren, and J. Sun: *Deep Residual Learning for Image Recognition*. IEEE CVPR pp. 770–778 (2016)

[83] J. Devlin, M. Chang, K. Lee, and K. Toutanova: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. *NAACL-HLT*, edited by J. Burstein, C. Doran, and T. Solorio, pp. 4171–4186. Association for Computational Linguistics (2019)

[84] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller: *Efficient BackProp. Neural Networks: Tricks of the Trade (2nd ed.)*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, volume 7700 of *Lecture Notes in Computer Science*, pp. 9–48. Springer (2012)

[85] I. Goodfellow, Y. Bengio, and A. Courville: *Deep Learning*. The MIT Press (2016)

[86] R. Sun: *Optimization for deep learning: theory and algorithms*. ArXiv **abs/1912.08957** (2019)

[87] G. B. Montavon: *Introduction to Neural Networks*. *Machine Learning Meets Quantum Physics*, edited by K. T. Schütt, S. Chmiela, O. A. von Lilienfeld, A. Tkatchenko, K. Tsuda et al., volume 968 of *Lecture Notes in Physics*, pp. 37–62. Springer (2020)

[88] X. Glorot, A. Bordes, and Y. Bengio: *Deep Sparse Rectifier Neural Networks*. *Proceedings of AISTATS*, edited by G. Gordon, D. Dunson, and M. Dudík, volume 15 of *PMLR*, pp. 315–323. PMLR (2011)

[89] A. F. Agarap: *Deep Learning using Rectified Linear Units (ReLU)*. ArXiv **abs/1803.08375** (2018)

[90] C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia: *Incorporating Second-Order Functional Knowledge for Better Option Pricing*. *NeurIPS*, edited by T. Leen, T. Dietterich, and V. Tresp, volume 13, pp. 451–457. MIT Press (2000)

[91] H. Zhao, F. Liu, L. Li, and C. Luo: *A Novel Softplus Linear Unit for Deep Convolutional Neural Networks*. Appl. Intell. **48** (7), 1707–1720 (2018)

[92] J. Han and C. Moraga: *The Influence of the Sigmoid Function Parameters on the Speed of Backpropagation Learning*. *From Natural to Artificial Neural Computation*, edited by J. Mira and F. Sandoval, volume 930 of *Lecture Notes in Computer Science*. IWANN (1995)

[93] D. Hendrycks and K. Gimpel: *Gaussian Error Linear Units (GELUs)*. ArXiv **abs/1606.08415** (2016)

[94] S. Elfwing, E. Uchibe, and K. Doya: *Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning*. Neural Netw. **107**, 3–11 (2018)

[95] P. Ramachandran, B. Zoph, and Q. V. Le: *Searching for Activation Functions*. ArXiv **abs/1710.05941** (2018)

[96] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun: *What is the best multi-stage architecture for object recognition? 2009 IEEE ICCV*, pp. 2146–2153 (2009)

[97] P. J. Werbos: *Applications of advances in nonlinear sensitivity analysis*. *System Modeling and Optimization*, edited by R. F. Drenick and F. Kozin, volume 38 of *Lecture Notes in Control and Information Sciences*, pp. 762–770. Springer (1982)

[98] D. E. Rumelhart, G. E. Hinton, and R. J. Williams: *Learning representations by back-propagating errors*. Nature **323** (6088), 533–536 (1986)

[99] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen et al.: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org (2015)

[100] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury et al.: *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. *NeurIPS*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox et al., volume 32, pp. 8024–8035. Curran Associates, Inc. (2019)

[101] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary et al.: *JAX: composable transformations of Python+NumPy programs*. http://github.com/google/jax (2018)

[102] A. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind: *Automatic Differentiation in Machine Learning: a Survey*. J. Mach. Learn. Res. **18** (153), 1–43 (2018)

[103] A. Botev, H. Ritter, and D. Barber: *Practical Gauss-Newton Optimisation for Deep Learning*. *Proceedings of ICML*, edited by D. Precup and Y. W. Teh, volume 70 of *PMLR*, pp. 557–565. PMLR (2017)

[104] J. Lafond, N. Vasilache, and L. Bottou: *Diagonal Rescaling For Neural Networks*. ArXiv **abs/1705.09319** (2017)

[105] S. Ioffe and C. Szegedy: *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. *Proceedings of ICML*, edited by F. Bach and D. Blei, volume 37 of *PMLR*, pp. 448–456. PMLR (2015)

[106] X. Lian and J. Liu: *Revisit Batch Normalization: New Understanding and Refinement via Composition Optimization*. *Proceedings of AISTATS*, edited by K. Chaudhuri and M. Sugiyama, volume 89 of *PMLR*, pp. 3254–3263. PMLR (2019)

[107] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang et al.: *On rectified linear units for speech processing*. *2013 IEEE ICASSP*, pp. 3517–3521 (2013)

[108] K. He, X. Zhang, S. Ren, and J. Sun: *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015 IEEE ICCV pp. 1026–1034 (2015)

[109] D. S. Park, J. Sohl-Dickstein, Q. V. Le, and S. L. Smith: *The Effect of Network Width on Stochastic Gradient Descent and Generalization: an Empirical Study*. ArXiv **abs/1905.03776** (2019)

[110] G. Yang and E. J. Hu: *Tensor Programs IV: Feature Learning in Infinite-Width Neural Networks*. *Proceedings of ICML*, edited by M. Meila and T. Zhang, volume 139 of *PMLR*, pp. 11727–11737. PMLR (2021)

[111] R. Karakida, S. Akaho, and S. ichi Amari: *Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach*. *International Conference on Artificial Intelligence and Statistics* (2018)

[112] J. Sohl-Dickstein, R. Novak, S. S. Schoenholz, and J. Lee: *On the infinite width limit of neural networks with a standard parameterization*. ArXiv **abs/2001.07301** (2020)

[113] J. Duchi, E. Hazan, and Y. Singer: *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*. J. Mach. Learn. Res. **12** (61), 2121–2159 (2011)

[114] M. D. Zeiler: *ADADELTA: An Adaptive Learning Rate Method*. ArXiv **abs/1212.5701** (2012)

[115] D. P. Kingma and J. Ba: *Adam: A Method for Stochastic Optimization*. ArXiv **abs/1412.6980** (2015)

[116] M. Li, T. Zhang, Y. Chen, and A. J. Smola: *Efficient Mini-Batch Training for Stochastic Optimization*. *Proceedings of ACM SIGKDD*, KDD '14, pp. 661–670. Association for Computing Machinery (2014)

[117] S. Geman, E. Bienenstock, and R. Doursat: *Neural Networks and the Bias/Variance Dilemma*. Neural Comput. **4** (1), 1–58 (1992)

[118] N. Morgan and H. Bourlard: *Generalization and Parameter Estimation in Feedforward Nets: Some Experiments*. *NeurIPS*, edited by D. Touretzky, volume 2. Morgan-Kaufmann (1990)

[119] L. Prechelt: *Early Stopping — But When? Neural Networks: Tricks of the Trade: Second Edition*, edited by G. Montavon, G. B. Orr, and K.-R. Müller, pp. 53–67. Springer, Berlin, Heidelberg (2012)

[120] H. Weyl: *The Classical Groups: Their Invariants and Representations*. Princeton University Press (1966)

[121] B. Neyshabur, R. Tomioka, and N. Srebro: *In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning*. ArXiv **abs/1412.6614** (2014)

[122] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak et al.: *Wide neural networks of any depth evolve as linear models under gradient descent*. J. Stat. Mech.: Theory Exp. **2020** (12), 124002 (2020)

[123] S. Arora, S. S. Du, W. Hu, Z. Li, and R. Wang: *Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks*. ArXiv **abs/1901.08584** (2019)

[124] T. Hu, W. Wang, C. Lin, and G. Cheng: *Regularization Matters: A Nonparametric Perspective on Overparametrized Neural Network*. ArXiv **abs/2007.02486** (2021)

[125] S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov et al.: *On Exact Computation with an Infinitely Wide Neural Net*. *NeurIPS*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox et al., volume 32, pp. 8141–8150. Curran Associates, Inc. (2019)

[126] S. Arora, S. S. Du, Z. Li, R. Salakhutdinov, R. Wang et al.: *Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks*. ArXiv **abs/1910.01663** (2020)

[127] R. Novak, L. Xiao, J. Hron, J. Lee, A. A. Alemi et al.: *Neural Tangents: Fast and Easy Infinite Neural Networks in Python*. ArXiv **abs/1912.02803** (2020)

[128] G. Yang and E. J. Hu: *Feature Learning in Infinite-Width Neural Networks*. ArXiv **abs/2011.14522** (2020)

[129] Q. N. Nguyen, M. Mondelli, and G. Montúfar: *Tight Bounds on the Smallest Eigenvalue of the Neural Tangent Kernel for Deep ReLU Networks*. ArXiv **abs/2012.11654** (2021)

[130] A. Panigrahi, A. Shetty, and N. Goyal: *Effect of Activation Functions on the Training of Overparametrized Neural Nets*. ArXiv **abs/1908.05660** (2020)

[131] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner: *Gradient-based learning applied to document recognition*. Proceedings of the IEEE **86** (11), 2278–2324 (1998)

[132] Z. Allen-Zhu, Y. Li, and Y. Liang: *Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers*. ArXiv **abs/1811.04918** (2019)

[133] B. Neyshabur, R. Tomioka, and N. Srebro: *Norm-Based Capacity Control in Neural Networks*. *Proceedings of The 28th Conference on Learning Theory*, edited by P. Grünwald, E. Hazan, and S. Kale, volume 40 of *Proceedings of Machine Learning Research*, pp. 1376–1401. PMLR (2015)

[134] B. Neyshabur, S. Bhojanapalli, D. A. McAllester, and N. Srebro: *A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks*. ArXiv **abs/1707.09564** (2018)

[135] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro: *Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks*. ArXiv **abs/1805.12076** (2019)

[136] A. P. Bartók, R. Kondor, and G. Csányi: *On representing chemical environments*. Phys. Rev. B **87** (18), 184115 (2013)

[137] J. Behler, S. Lorenz, and K. Reuter: *Representing molecule-surface interactions with symmetry-adapted neural networks*. J. Chem. Phys. **127** (1), 014705 (2007)

[138] J. Behler: *Atom-centered symmetry functions for constructing high-dimensional neural network potentials*. J. Chem. Phys. **134** (7), 074106 (2011)

[139] K. V. J. Jose, N. Artrith, and J. Behler: *Construction of high-dimensional neural network potentials using environment-dependent atom pairs*. J. Chem. Phys. **136** (19), 194111 (2012)

[140] N. Artrith and A. Urban: *An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO2*. Comput. Mater. Sci. **114**, 135–150 (2016)

[141] J. S. Smith, O. Isayev, and A. E. Roitberg: *ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost*. Chem. Sci. **8** (4), 3192–3203 (2017)

[142] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand: *wACSF – Weighted atom-centered symmetry functions as descriptors in machine learning potentials*. J. Chem. Phys. **148** (24), 241709 (2018)

[143] K. Zhang, L. Yin, and G. Liu: *Physically inspired atom-centered symmetry functions for the construction of high dimensional neural network potential energy surfaces*. Comput. Mater. Sci. **186**, 110071 (2021)

[144] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi: *Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons*. Phys. Rev. Lett. **104** (13), 136403 (2010)

[145] A. Khorshidi and A. A. Peterson: *Amp: A modular approach to machine learning in atomistic simulations*. Comput. Phys. Commun. **207**, 310–324 (2016)

[146] O. T. Unke and M. Meuwly: *A reactive, scalable, and transferable model for molecular energies from a neural network approach based on local information*. J. Chem. Phys. **148** (24), 241708 (2018)

[147] E. Kocer, J. K. Mason, and H. Erturk: *A novel approach to describe chemical environments in high-dimensional neural network potentials*. J. Chem. Phys. **150** (15), 154102 (2019)

[148] E. Kocer, J. K. Mason, and H. Erturk: *Continuous and optimally complete description of chemical environments using Spherical Bessel descriptors*. AIP Adv. **10** (1), 015021 (2020)

[149] A. V. Shapeev: *Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials*. Multiscale Model. Simul. **14** (3), 1153–1173 (2016)

[150] K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev: *Machine learning of molecular properties: Locality and active learning*. J. Chem. Phys. **148** (24), 241727 (2018)

[151] Y. Zhang, C. Hu, and B. Jiang: *Embedded Atom Neural Network Potentials: Efficient and Accurate Machine Learning with a Physically Inspired Representation*. J. Phys. Chem. Lett. **10** (17), 4962–4967 (2019)

[152] M. Uhrin: *Through the eyes of a descriptor: Constructing complete, invertible descriptions of atomic environments*. Phys. Rev. B **104** (14), 144110 (2021)

[153] C. van der Oord, G. Dusson, G. Csányi, and C. Ortner: *Regularised atomic body-ordered permutation-invariant polynomials for the construction of interatomic potentials*. Mach. Learn.: Sci. Technol. **1** (1), 015004 (2020)

[154] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld: *Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning*. Phys. Rev. Lett. **108** (5), 058301 (2012)

[155] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt et al.: *Machine learning of accurate energy-conserving molecular force fields*. Sci. Adv. **3** (5) (2017)

[156] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko: *Towards exact molecular dynamics simulations with machine-learned force fields*. Nat. Commun. **9** (1), 3887 (2018)

[157] K. Hansen, F. Biegler, R. Ramakrishnan, W. Pronobis, O. A. von Lilienfeld et al.: *Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space*. J. Phys. Chem. Lett. **6** (12), 2326–2331 (2015)

[158] H. Huo and M. Rupp: *Unified Representation for Machine Learning of Molecules and Crystals*. arXiv **abs/1704.06439** (2018)

[159] A. Stuke, M. Todorovi, M. Rupp, C. Kunkel, K. Ghosh et al.: *Chemical diversity in molecular orbital energy predictions with kernel ridge regression*. J. Chem. Phys. **150** (20), 204121 (2019)

[160] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl: *Neural Message Passing for Quantum Chemistry*. *Proceedings of ICML*, edited by D. Precup and Y. W. Teh, volume 70 of *PMLR*, pp. 1263–1272. PMLR (2017)

[161] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko: *Quantum-chemical insights from deep tensor neural networks*. Nat. Commun. **8** (1), 13890 (2017)

[162] K. Schütt, P.-J. Kindermans, H. E. Sauceda Felix, S. Chmiela, A. Tkatchenko et al.: *SchNet: A continuous-filter convolutional neural network for modeling quantum interactions*. *NeurIPS*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus et al., volume 30, pp. 991–1001. Curran Associates, Inc. (2017)

[163] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller: *SchNet - A deep learning architecture for molecules and materials*. J. Chem. Phys. **148** (24), 241722 (2018)

[164] N. Lubbers, J. S. Smith, and K. Barros: *Hierarchical modeling of molecular energies using a deep neural network*. J. Chem. Phys. **148** (24), 241715 (2018)

[165] O. T. Unke and M. Meuwly: *PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges*. J. Chem. Theory Comput. **15** (6), 3678–3693 (2019)

[166] B. Anderson, T. S. Hy, and R. Kondor: *Cormorant: Covariant Molecular Neural Networks*. *NeurIPS*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox et al., volume 32, pp. 14537–14546. Curran Associates, Inc. (2019)

[167] K. T. Schütt, O. T. Unke, and M. Gastegger: *Equivariant message passing for the prediction of tensorial properties and molecular spectra*. ArXiv **abs/2102.03150** (2021)

[168] S. Batzner, T. E. Smidt, L. Sun, J. P. Mailoa, M. Kornbluth et al.: *SE(3)-Equivariant Graph Neural Networks for Data-Efficient and Accurate Interatomic Potentials*. ArXiv **abs/2101.03164** (2021)

[169] N. Artrith, A. Urban, and G. Ceder: *Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species*. Phys. Rev. B **96** (1), 014112 (2017)

[170] K. Yao, J. E. Herr, D. W. Toth, R. Mckintyre, and J. Parkhill: *The TensorMol-0.1 model chemistry: a neural network augmented with long-range physics*. Chem. Sci. **9** (8), 2261–2269 (2018)

[171] O. A. von Lilienfeld, R. Ramakrishnan, M. Rupp, and A. Knoll: *Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties*. Int. J. Quantum Chem. **115** (16), 1084–1093 (2015)

[172] J. Flusser, T. Suk, and B. Zitová: *Moment Invariants to Translation, Rotation and Scaling*, chapter 2, pp. 13–47. John Wiley & Sons, Ltd (2009)

[173] J. Flusser, T. Suk, and B. Zitová: *3D Moment Invariants to Translation, Rotation, and Scaling*, chapter 4, pp. 95–162. John Wiley & Sons, Ltd (2016)

[174] T. Suk and J. Flusser: *Tensor Method for Constructing 3D Moment Invariants*. *Computer Analysis of Images and Patterns*, edited by P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. Kropatsch, volume 6855 of *Lecture Notes in Computer Science*, pp. 212–219. Springer (2011)

[175] B. Yang, T. Suk, M. Dai, and J. Flusser: *2D and 3D Image Analysis by Gaussian-Hermite Moments*, chapter 7, pp. 143–173. Science Gate Publishing (2014)

[176] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond: *Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17*. J. Chem. Inf. Model. **52** (11), 2864–2875 (2012)

[177] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld: *Quantum chemistry structures and properties of 134 kilo molecules*. Sci. Data **1** (1), 140022 (2014)

[178] F. A. Faber, A. S. Christensen, B. Huang, and O. A. von Lilienfeld: *Alchemical and structural distribution based representation for universal quantum machine learning*. J. Chem. Phys. **148** (24), 241717 (2018)

[179] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun et al.: *On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events*. Npj Comput. Mater. **6** (20) (2020)

[180] Y. Guan, S. Yang, and D. H. Zhang: *Construction of reactive potential energy surfaces with Gaussian process regression: active data selection*. Mol. Phys. **116** (7-8), 823–834 (2018)

[181] Z. Li, J. R. Kermode, and A. De Vita: *Molecular Dynamics with On-the-Fly Machine Learning of Quantum-Mechanical Forces*. Phys. Rev. Lett. **114** (9), 096405 (2015)

[182] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg: *Less is more: Sampling chemical space with active learning*. J. Chem. Phys. **148** (24), 241733 (2018)

[183] M. Gastegger, J. Behler, and P. Marquetand: *Machine learning molecular dynamics for the simulation of infrared spectra*. Chem. Sci. **8** (10), 6924–6935 (2017)

[184] L. Zhang, D.-Y. Lin, H. Wang, R. Car, and W. E: *Active learning of uniformly accurate interatomic potentials for materials simulation*. Phys. Rev. Mater. **3** (2), 023804 (2019)

[185] C. Schran, J. Behler, and D. Marx: *Automated Fitting of Neural Network Potentials at Coupled Cluster Accuracy: Protonated Water Clusters as Testing Ground*. J. Chem. Theory Comput. **16** (1), 88–99 (2020)

[186] Y. Gal and Z. Ghahramani: *Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning*. *Proceedings of ICML*, edited by M. F. Balcan and K. Q. Weinberger, volume 48 of *PMLR*, pp. 1050–1059. PMLR, New York, New York, USA (2016)

[187] J. P. Janet, C. Duan, T. Yang, A. Nandy, and H. J. Kulik: *A quantitative uncertainty metric controls error in neural network-driven chemical discovery*. Chem. Sci. **10** (34), 7913–7922 (2019)

[188] J. P. Janet and H. J. Kulik: *Resolving Transition Metal Chemical Space: Feature Selection for Machine Learning and Structure-Property Relationships*. J. Phys. Chem. A **121** (46), 8939–8954 (2017)

[189] A. Nandy, C. Duan, J. P. Janet, S. Gugler, and H. J. Kulik: *Strategies and Software for Machine Learning Accelerated Discovery in Transition Metal Chemistry*. Ind. Eng. Chem. Res. **57** (42), 13973–13986 (2018)

[190] D. A. Cohn: *Neural Network Exploration Using Optimal Experiment Design*. Neural Netw. **9** (6), 1071–1083 (1996)

[191] D. J. C. MacKay: *Information-Based Objective Functions for Active Data Selection*. Neural Comput. **4** (4), 590–604 (1992)

[192] V. Fedorov: *Theory of optimal experiments*. New York: Academic Press (1972)

[193] E. V. Podryabinkin and A. V. Shapeev: *Active learning of linearly parametrized interatomic potentials*. Comput. Mater. Sci. **140**, 171–180 (2017)

[194] J. Westermayr, M. Gastegger, M. F. S. J. Menger, S. Mai, L. González et al.: *Machine learning enables long time scale molecular photodynamics simulations*. Chem. Sci. **10** (35), 8100–8107 (2019)

[195] M. Veit, D. M. Wilkins, Y. Yang, R. A. DiStasio, and M. Ceriotti: *Predicting molecular dipole moments by combining atomic partial charges and atomic dipoles*. J. Chem. Phys. **153** (2), 24113 (2020)

[196] A. Grisafi, D. M. Wilkins, G. Csányi, and M. Ceriotti: *Symmetry-Adapted Machine Learning for Tensorial Properties of Atomistic Systems*. Phys. Rev. Lett. **120** (3), 036002 (2018)

[197] D. M. Wilkins, A. Grisafi, Y. Yang, K. U. Lao, R. A. DiStasio et al.: *Accurate molecular polarizabilities with coupled cluster theory and machine learning*. Proc. Natl. Acad. Sci. **116** (9), 3401–3406 (2019)

[198] A. Grisafi, D. M. Wilkins, M. J. Willatt, and M. Ceriotti: *Atomic-Scale Representation and Statistical Learning of Tensorial Properties*, chapter 1, pp. 1–21. American Chemical Society (2019)

[199] I. Novikov, B. Grabowski, F. Körmann, and A. Shapeev: *Magnetic Moment Tensor Potentials for collinear spin-polarized materials reproduce different magnetic states of bcc Fe*. Npj Comput. Mater. **8** (1), 13 (2022)

[200] A. Lunghi: *Insights into the Spin-Lattice Dynamics of Organic Radicals Beyond Molecular Tumbling: A Combined Molecular Dynamics and Machine-Learning Approach*. Appl. Magn. Reson. **51** (11), 1343–1356 (2020)

[201] A. Lunghi and S. Sanvito: *Surfing Multiple Conformation-Property Landscapes via Machine Learning: Designing Single-Ion Magnetic Anisotropy*. J. Phys. Chem. C **124** (10), 5802–5806 (2020)

[202] D. Gatteschi and R. Sessoli: *Quantum Tunneling of Magnetization and Related Phenomena in Molecular Materials*. Angew. Chem. Int. Ed. **42** (3), 268–297 (2003)

[203] A. Abragam and B. Bleaney: *Electron Paramagnetic Resonance of Transition Ions*. Clarendon (1970)

[204] Y. Rechkemmer, F. D. Breitgoff, M. van der Meer, M. Atanasov, M. Hakl et al.: *A four-coordinate cobalt(II) single-ion magnet with coercivity and a very high energy barrier*. Nat. Commun. **7** (1), 10467 (2016)

[205] W. H. Harman, T. D. Harris, D. E. Freedman, H. Fong, A. Chang et al.: *Slow Magnetic Relaxation in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes*. J. Am. Chem. Soc. **132** (51), 18115–18126 (2010)

[206] M. Atanasov, D. Ganyushin, D. A. Pantazis, K. Sivalingam, and F. Neese: *Detailed Ab Initio First-Principles Study of the Magnetic Anisotropy in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes*. Inorg. Chem. **50** (16), 7460–7477 (2011)

[207] G. Rogez, J.-N. Rebilly, A.-L. Barra, L. Sorace, G. Blondin et al.: *Very Large Ising-Type Magnetic Anisotropy in a Mononuclear Ni(II) Complex*. Angew. Chem. Int. Ed. **44** (12), 1876–1879 (2005)

[208] Minissale, M., Congiu, E., and Dulieu, F.: *Direct measurement of desorption and diffusion energies of O and N atoms physisorbed on amorphous surfaces*. Astron. Astrophys. **585**, A146 (2016)

[209] S. Grimme, J. G. Brandenburg, C. Bannwarth, and A. Hansen: *Consistent structures and interactions by density functional theory with small atomic orbital basis sets*. J. Chem. Phys. **143** (5), 054107 (2015)

[210] T. Huber, A. E. Torda, and W. F. van Gunsteren: *Local elevation: A method for improving the searching properties of molecular dynamics simulation*. J. Comput. Aid. Mol. Des. **8** (6), 695–708 (1994)

[211] A. Laio and M. Parrinello: *Escaping free-energy minima*. Proc. Natl. Acad. Sci. **99** (20), 12562–12566 (2002)

[212] A. Barducci, G. Bussi, and M. Parrinello: *Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method*. Phys. Rev. Lett. **100** (2), 020603 (2008)

[213] Y. A. Du, J. Rogal, and R. Drautz: *Diffusion of hydrogen within idealized grains of bcc Fe: A kinetic Monte Carlo study*. Phys. Rev. B **86** (17), 174110 (2012)

[214] A. B. Bortz, M. H. Kalos, and J. L. Lebowitz: *A new algorithm for Monte Carlo simulation of Ising spin systems*. J. Comput. Phys. **17** (1), 10–18 (1975)

[215] D. T. Gillespie: *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*. J. Comput. Phys. **22** (4), 403–434 (1976)

[216] R. Kirchheim: *Monte-carlo simulations of interstitial diffusion and trapping – I. One type of traps and dislocations*. Acta Metall. **35** (2), 271–280 (1987)

[217] R. Kirchheim: *Hydrogen solubility and diffusivity in defective and amorphous metals*. Prog. Mater. Sci. **32** (4), 261–325 (1988)

[218] A. Ramasubramaniam, M. Itakura, M. Ortiz, and E. Carter: *Effect of atomic scale plasticity on hydrogen diffusion in iron: Quantum mechanically informed and on-the-fly kinetic Monte Carlo simulations*. J. Mater. Res. **23** (10), 2757–2773 (2008)

[219] S. McConnell and J. Kästner: *Instanton rate constant calculations close to and above the crossover temperature*. J. Comput. Chem. **38** (30), 2570–2580 (2017)

[220] Karssemeijer, L. J. and Cuppen, H. M.: *Diffusion-desorption ratio of adsorbed CO and CO_2 on water ice*. Astron. Astrophys. **569**, A107 (2014)

[221] B. Senevirathne, S. Andersson, F. Dulieu, and G. Nyman: *Hydrogen atom mobility, kinetic isotope effects and tunneling on interstellar ices (Ih and ASW)*. Mol. Astrophys. **6**, 59–69 (2017)

[222] V. Ásgeirsson, H. Jónsson, and K. T. Wikfeldt: *Long-Time Scale Simulations of Tunneling-Assisted Diffusion of Hydrogen on Ice Surfaces at Low Temperature*. J. Phys. Chem. C **121** (3), 1648–1657 (2017)

[223] J.-B. Bossa, B. Maté, C. Fransen, S. Cazaux, S. Pilling et al.: *POROSITY AND BAND-STRENGTH MEASUREMENTS OF MULTI-PHASE COMPOSITE ICES*. Astrophys. J. **814** (1), 47 (2015)

[224] E. B. Jenkins: *A UNIFIED REPRESENTATION OF GAS-PHASE ELEMENT DEPLETIONS IN THE INTERSTELLAR MEDIUM*. Astrophys. J. **700** (2), 1299–1348 (2009)

[225] M. Ruaud, V. Wakelam, and F. Hersant: *Gas and grain chemical composition in cold cores as predicted by the Nautilus three-phase model*. Mon. Not. R. Astron. Soc. **459** (4), 3756–3767 (2016)

Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

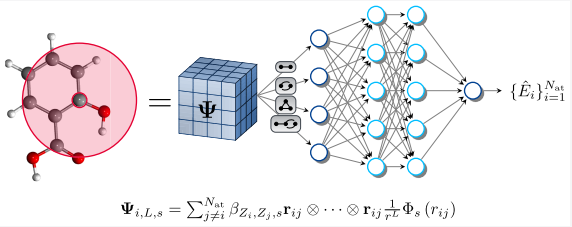
Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials

V. Zaverkin and J. Kästner*

Cite This: J. Chem. Theory Comput. 2020, 16, 5410−5421 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Machine learning techniques allow a direct mapping of atomic positions and nuclear charges to the potential energy surface with almost ab initio accuracy and the computational efficiency of empirical potentials. In this work, we propose a machine learning method for constructing high-dimensional potential energy surfaces based on feed-forward neural networks. As input to the neural network, we propose an extendable invariant local molecular descriptor constructed from geometric moments. Their formulation via pairwise distance vectors and tensor contractions allows a very efficient implementation on graphical processing units (GPUs). The atomic species is encoded in the molecular descriptor, which allows the restriction to one neural network for the training of all atomic species in the data set. We demonstrate that the accuracy of the developed approach in representing both chemical and configurational spaces is comparable to the one of several established machine learning models. Due to its high accuracy and efficiency, the proposed machine-learned potentials can be used for any further tasks, for example, the optimization of molecular geometries, the calculation of rate constants, or molecular dynamics.



1. INTRODUCTION

Most applications in computational chemistry require the use of potential energy surfaces (PESs). The PES is a multidimensional real-valued function of atomic coordinates. It can be obtained by the solution of the electronic Schrödinger equation in the Born−Oppenheimer approximation.[1] For the estimation of individual points on the PES, different techniques can be used, from ab initio electronic structure theory to empirical fits by force fields. Specifically, highly accurate estimates are computationally expensive; thus, applications that require energies and forces for a large number of atomic configurations, like molecular dynamics (MD) or geometry optimization, require significant amounts of computational time.

MD simulations of big systems, e.g., proteins or other macromolecules, are currently infeasible at the ab initio level of theory. In such cases, empirical force fields provide the necessary computational efficiency at the drawback of limited transferability[2] and their general inability to describe bond formation and bond breaking. Therefore, a method which allows a direct mapping of atomic positions and nuclear charges to the PES, i.e., $f : \{z_i, \mathbf{r}_i\} \mapsto E$, with maximal accuracy is required.

Machine learning (ML) techniques can be applied for an efficient approximation of the PES, since, once trained, they hold the promise to combine the accuracy of ab initio electronic structure methods with the efficiency of empirical force fields. For chemical applications, several ML techniques can be used to predict a variety of chemical and physical properties of molecules and solids. The most frequently used approach is feed-forward neural networks (NN).

The construction of a reliable machine-learned mapping from atomic positions to potential energies requires a carefully chosen representation of the input to the ML algorithm defined by the atomic coordinates and nuclear charges. This is because the ML methodology does not exploit any information about the physics of the problem, in our case neither the invariance of a chemical system with respect to translation, reflection, or rotation of the whole molecule nor to permutation of atoms with the same nuclear charge (atomic species). Therefore, a transformation to a suitable set of coordinates, i.e., a suitable descriptor, is required in order to obtain the desired accuracy in energy and gradient predictions.

Several descriptors for ML models have been proposed. Some of the approaches split the molecules into atomic contributions and use hand-crafted descriptors, e.g., atom-centered symmetry functions (ACSF),[3,4] power spectra or bispectra of spherical harmonics,[5−9] or geometric moments.[10,11] Others use the Coulomb matrix of the whole molecule.[12] A different class of models is referred to as

Received: April 9, 2020
Published: July 16, 2020

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

message-passing high-dimensional NNs, which learn to construct invariant features in a data-driven manner.[13−18] Most of the methods based on hand-crafted descriptors are limited to only a few atomic species[3,4,19,20] or smaller systems[21,22] or fail to approach the accuracy of 1 kcal/mol with respect to the underlying ab initio method[23] required for chemical applications.

The requirements of a PES fit for successful application in chemistry are summarized in the following. It has to approximate the PES sufficiently accurately with an error below 1 kcal/mol in the energies with respect to the underlying ab initio method and a comparable error for the forces. The approximation should be differentiable with respect to the atomic coordinates to allow for the calculation of forces and Hessians. It has to fulfill the invariances mentioned above: translation, rotation, and permutation of like atoms. The fit should also be systematically improvable; i.e., the accuracy of predictions should increase with increasing size of the training data set. Finally, the machine learning model should be general; i.e., it should be transferable between similar systems and their configurations.[24] Unfortunately, existing models and respective potential energy surfaces fulfill only a subset of these requirements.

In this work, we introduce a novel, physically inspired molecular descriptor, which can be used as input for any ML algorithm. We refer to it as Gaussian Moments (GM) since it was inspired by Gaussian-type atomic orbitals and derived from geometric moments previously used for pattern recognition.[25−28] We have chosen feed-forward NNs as an ML method for our applications. In addition to the structural description, we encode the information about the atomic species in the molecular representation. This allows us to use a single NN for all atomic species, in contrast to using an individual NN for each species as frequently necessary previously.[3,7,19,20,29] It is shown that the ML potentials built with the GM descriptor match or improve upon the state-of-the-art performance on standard benchmark data sets.

This article has the following structure: first, we formulate the molecular representation based on GM and explain our machine learning model describing details on its training. Then, in Section 3, we apply our machine learning model to the QM9,[30,31] MD17,[14,21,22] and ISO17[14,15,31] benchmark data sets and compare it to various models published in the literature. Additionally, we use it to predict vibrational frequencies based on a newly generated training set. The concluding remarks are given in Section 4.

2. METHOD

As mentioned above, a suitable descriptor, which converts atomic coordinates into ML input, should ensure the same global invariances as the physical system. These are (1) the global rotation, (2) the translation, and (3) the reflection of a molecular structure, as well as (4) the exchange of atoms of the same atomic species, i.e., with the same nuclear charge Z. One simple solution, which satisfies requirements 1−3, can be constructed using just the scalar product of vectors $\mathbf{r}_{ij}$ from the position of a central atom i to the positions j of all other atoms, resulting in the Weyl matrix[32] Σ_i

$$\Sigma_i = \begin{pmatrix} \mathbf{r}_{i1}\cdot\mathbf{r}_{i1} & \mathbf{r}_{i1}\cdot\mathbf{r}_{i2} & \cdots \\ \mathbf{r}_{i2}\cdot\mathbf{r}_{i1} & \mathbf{r}_{i2}\cdot\mathbf{r}_{i2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{1}$$

However, any molecular system is invariant with respect to the exchange of two atoms of the same type. Therefore, a proper molecular representation has to incorporate this property as well. Unfortunately, introducing the permutation invariance into the above representation makes it intractable whenever one deals with large systems and, moreover, can violate the differentiability of the molecular representation.[6] Therefore, the main focus of this section is to introduce a class of molecular representations, which satisfies permutation invariance and is at least a C^2 function of the atomic positions; i.e., it is at least twice differentiable.

2.1. Molecular Descriptor.

The methodology of this study is based on the fact that the PES is the expectation value of electronic Hamiltonian $\hat{H}$; i.e., it is a solution of the electronic Schrödinger equation

$$\hat{H}\Psi(\mathbf{r}) = E(\mathbf{r})\Psi(\mathbf{r}) \tag{2}$$

Here, Ψ is the electronic wave function which depends on the atomic position vector $\mathbf{r}$. Thus, the energy of a molecular system is a functional of the electronic wave function

$$E = \mathcal{F}[\Psi] \tag{3}$$

The electronic wave function can be efficiently expanded into atom-centered Gaussian-type orbitals, which inspired our choice of the molecular descriptors. Note that the descriptor uses exclusively atomic positions rather than electronic coordinates.

We split the descriptor for the whole chemical system into functions, which describe the environment of each atom individually. Those can subsequently be combined to describe whole molecular or periodic systems. The environment of each atom is described by a function reminiscent of a Gaussian-type orbital[33] (GTO)

$$\varphi_{s,l_x,l_y,l_z}(\mathbf{r}) = \frac{x^{l_x}y^{l_y}z^{l_z}}{r^L}\Phi_s(r) \tag{4}$$

with $\mathbf{r} = (x, y, z)$ being an atom's position relative to a central atom, r being its absolute value, and L defined as $L = l_x + l_y + l_z$. The prefactor $x^{l_x}y^{l_y}z^{l_z}/r^L$ covers the angular dependence of the GTO, which we deal with in eq 9. The radial part $\Phi_s(r)$ was chosen to be a single normalized Gaussian with a radial cutoff defined as

$$\Phi_s(r) = \left(\frac{2N_{\text{Gauss}}^2}{\pi R_{\text{max}}^2}\right)^{1/4} e^{-N_{\text{Gauss}}^2/R_{\text{max}}^2(r-\gamma_s)^2} f_{\text{cut}}(r) \tag{5}$$

The width of each Gaussian depends on the total number N_{Gauss} of functions used and the cutoff radius R_{max}. Each Gaussian is centered at γ_s, which is chosen evenly spaced between R_{min} and R_{max}

$$\gamma_s = R_{\text{min}} + \frac{s-1}{N_{\text{Gauss}}-1}(R_{\text{max}} - R_{\text{min}}) \tag{6}$$

with s being an index from 1 to N_{Gauss}. As discussed in Section 3, we typically use $N_{\text{Gauss}} = 7$ and $R_{\text{min}} = 0.5$ Å. R_{max} depends on the specific case. Note that γ_s is defined for the whole data set. An example of the radial basis functions $\Phi_s(r)$ is shown in Figure 1.

Each radial function incorporates a cutoff function $f_{\text{cut}}(r)$, which restricts the descriptor to the local neighborhood of the atom and decays smoothly to zero at the cutoff radius R_{max}. In

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

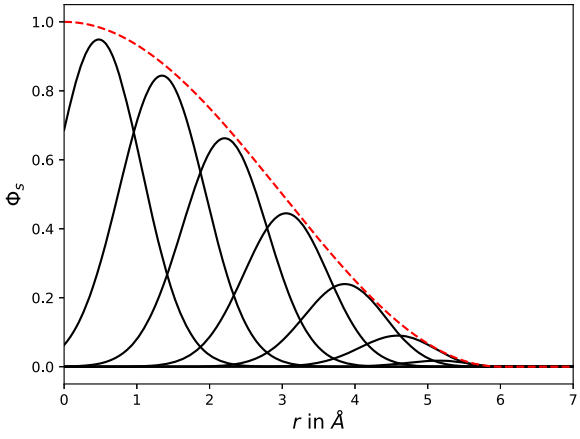


Figure 1. Radial basis functions $\Phi_s(r)$ (black) and the cutoff function $f_{cut}(r)$ (red, dashed) for R_{max} = 6.0 Å and N_{Gauss} = 7.

this work, we have chosen the cosine cutoff function,[3] see Figure 1

$$f_{cut}(r) = \begin{cases} \frac{1}{2}\left(\cos\left(\pi\frac{r}{R_{max}}\right) + 1\right) & r \le R_{max} \\ 0 & r > R_{max} \end{cases} \quad (7)$$

Periodic boundary conditions are incorporated by including the periodic images of atoms in the local neighborhood. The GM descriptor is constructed from the coordinates of the image atoms and the atoms within the cell. However, a more thorough discussion of periodic calculations is beyond the scope of this work.

In the next step, we form a linear combination of the atomic "wave" functions $\varphi_{s,l_x,l_y,l_z}(\mathbf{r})$, similar to the linear combination of atomic orbitals (LCAO), again inspired by quantum chemistry. The total molecular wave function centered at an atom i reads

$$\Psi_{i,L,s} = \sum_{j\neq i}^{N_{at}} \beta_{Z_i,Z_j,s}\varphi_{s,l_x,l_y,l_z}(\mathbf{r}_{ij}) \quad (8)$$

where Z_i and Z_j are the nuclear charges of the central atom i and its atomic neighbors j. The coefficients $\beta_{Z_i,Z_j,s}$ distinguish between nuclear charges and radial shells. They are optimized in the training procedure. For a given i, L, and s, $\Psi_{i,L,s}$ is a tensor of rank L.

Equation 8 preserves invariances 2 and 4 by construction. $\Psi_{i,L,s}$ is invariant with respect to translation 2 owing to its dependence on the atomic distance vectors $\mathbf{r}_{ij}$. The invariance with respect to permutation of like atoms (4) is ensured by the sum. However, the prefactor $x^{l_x}y^{l_y}z^{l_z}/r^L$ still violates the invariance with respect to rotation and reflection for $L > 0$. Consequently, $\Psi_{i,L,s}$ cannot be used directly as input to ML algorithms, and further treatment is necessary.

One can interpret L as an angular momentum similar to spherical harmonics. For example, $L = 0$ corresponds to the shape of a spherically symmetric s-orbital; $L = 1$ corresponds to the shape of a p-orbital, $L = 2$ to that of a d-orbital, and so on. To construct a rotationally invariant basis, we look deeper into the mathematical properties of $\Psi_{i,L,s}$. GTO functions in eq 4 can be written as a Cartesian tensor. For L = 0,1,2, we can write $\Psi_{i,L,s}$ when rewriting the angular dependence in terms of atomic distance vectors $\mathbf{r}_{ij}$ rather than in terms of its components, as

$$\Psi_{i,0,s} = \sum_{j\neq i}^{N_{at}} \beta_{Z_i,Z_j,s}\Phi_s(r_{ij})$$

$$\Psi_{i,1,s} = \sum_{j\neq i}^{N_{at}} \beta_{Z_i,Z_j,s}\frac{\mathbf{r}_{ij}}{r}\Phi_s(r_{ij})$$

$$\Psi_{i,2,s} = \sum_{j\neq i}^{N_{at}} \beta_{Z_i,Z_j,s}\frac{\mathbf{r}_{ij}\otimes\mathbf{r}_{ij}}{r^2}\Phi_s(r_{ij}) \quad (9)$$

where $\otimes$ denotes the tensor product. For an arbitrary angular momentum L, one can write

$$\Psi_{i,L,s} = \sum_{j\neq i}^{N_{at}} \beta_{Z_i,Z_j,s}\underbrace{\mathbf{r}_{ij}\otimes\cdots\otimes\mathbf{r}_{ij}}_{L\text{ times}}\frac{1}{r^L}\Phi_s(r_{ij}) \quad (10)$$

The tensor $\mathbf{r}_{ij}\otimes\cdots\otimes\mathbf{r}_{ij}$ has rank L and is, in the following discussion, referred to as $T_{i_1,i_2,\cdots,i_L} = (\mathbf{r}_{ij}\otimes\cdots\otimes\mathbf{r}_{ij})_{i_1,i_2,\cdots,i_L}$ to simplify the notation. Since $T_{i_1,i_2,\cdots,i_L}$ is a Cartesian tensor, it behaves under rotation according to the rule

$$\hat{T}_{\alpha_1,\alpha_2,\cdots,\alpha_L} = R_{\alpha_1,i_1}R_{\alpha_2,i_2}\cdots R_{\alpha_L,i_L}T_{i_1,i_2,\cdots,i_L} \quad (11)$$

where R_{α_1,i_1} is an arbitrary orthonormal matrix, e.g., a rotation or reflection. From linear algebra, it is known that any full contraction of a Cartesian tensor or of a product of Cartesian tensors is a rotationally invariant scalar. The radial function does not affect this property due to its inherent invariance with respect to rotations. The same holds for reflections. Consequently, an invariant basis, which satisfies all the requirements, can be constructed by calculating the full contractions of the molecular wave function.

Note that the concept of constructing rotational invariants using contractions of Cartesian tensors was initially introduced by Flusser et al.,[25−28] where geometric and Gaussian−Hermite moments were used to address pattern recognition problems. Additionally, geometric moments were used to construct rotationally invariant bases for linear regression in PES construction.[10,11]

Inspired by previous work on invariants obtained using geometric moments,[10,11,25−28] we refer to scalars obtained by contracting $\Psi_{i,L,s}$ as Gaussian moments (GM). To simplify the generation of contractions, we employed graphs.[27] Some examples are shown in Figure 2. However, one can find a direct correspondence to index-matrices[10] and use them instead.

In general for the representation of a molecular structure, at least a $(3N_{at} - 6)$-dimensional descriptor is needed. This can be fulfilled by using only rather few contractions. It turned out to be sufficient to restrict the total angular momentum to $L \le 3$ and the maximal number of contracted tensors to 3. This

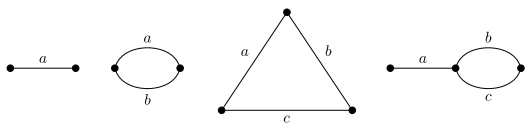


Figure 2. Generating graphs for the tensor contractions 12.2, 12.3, 12.6, and 12.7 of eq 12.

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

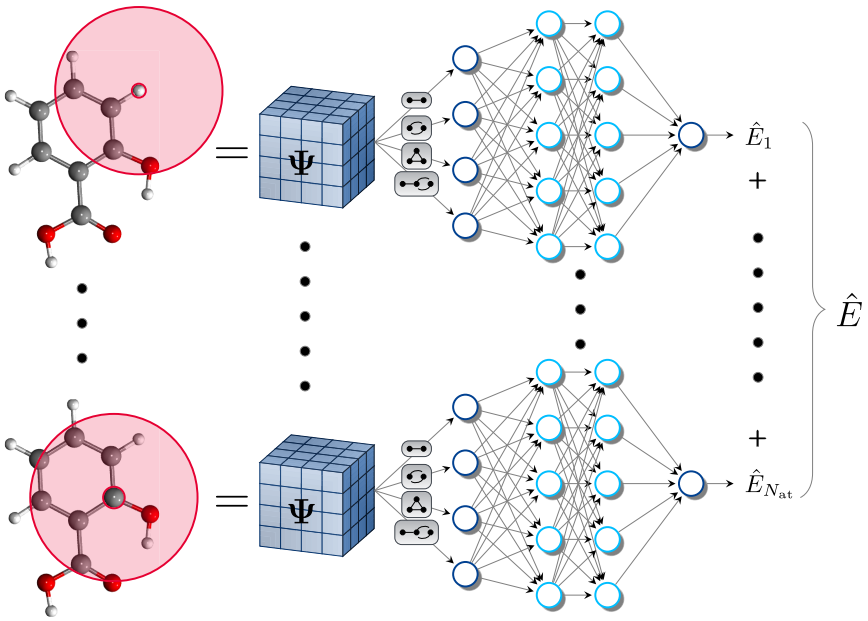


Figure 3. Schematic representation of the model used in this work for calculating molecular energies and forces.

results in a total of eight contractions, i.e., Gaussian moments, that we used throughout this work

$$\begin{aligned}
\rho_{i,s_1} &= \Psi_{i,0,s_1} \\
\rho_{i,s_1,s_2} &= (\Psi_{i,1,s_1})_a(\Psi_{i,1,s_2})_a \\
\rho_{i,s_1,s_2} &= (\Psi_{i,2,s_1})_{a,b}(\Psi_{i,2,s_2})_{a,b} \\
\rho_{i,s_1,s_2} &= (\Psi_{i,3,s_1})_{a,b,c}(\Psi_{i,3,s_2})_{a,b,c} \\
\rho_{i,s_1,s_2,s_3} &= (\Psi_{i,2,s_1})_{a,b}(\Psi_{i,1,s_2})_a(\Psi_{i,1,s_3})_b \\
\rho_{i,s_1,s_2,s_3} &= (\Psi_{i,2,s_1})_{a,b}(\Psi_{i,2,s_2})_{a,c}(\Psi_{i,2,s_3})_{b,c} \\
\rho_{i,s_1,s_2,s_3} &= (\Psi_{i,1,s_1})_a(\Psi_{i,3,s_2})_{a,b,c}(\Psi_{i,2,s_3})_{b,c} \\
\rho_{i,s_1,s_2,s_3} &= (\Psi_{i,3,s_1})_{a,b,c}(\Psi_{i,3,s_2})_{a,b,d}(\Psi_{i,2,s_3})_{c,d}
\end{aligned} \tag{12}$$

Here, Einstein's notation was used for tensor contractions, i.e., the sum is taken over double indices, to simplify the expressions. All these tensors are symmetric. We use only upper triangular entries as descriptors.

In total, using $N_{\text{Gauss}} = 7$ and all contractions given in eq 12, we obtained $7 + 28 \cdot 3 + 84 \cdot 4 = 427$ rotationally invariant scalars for each atom. These constitute the molecular descriptor, which was used as input for the NN in Section 3. All elements of the molecular descriptor depend on the atomic species of the central atom and its atomic neighborhood. This dependence is encoded using the coefficients $\beta_{Z_i,Z_j,s}$ which are optimized during training.

Contractions of two wave functions can be related to electronic densities with an angular momentum L. Electronic densities were recently used for the construction of a molecular representation in ML.[29] However, the approach presented here is more general than electronic densities as it allows to contract more (and less) than two wave functions to construct rotational invariants. Thus, much more insight in the angular and radial distribution of the atomic environment can be incorporated into the machine learning algorithms at the same computational cost.

2.2. Atomistic Neural Networks. Artificial neural networks (NN) have been proven to be capable of approximating any nonlinear functional relationship.[34] Therefore, they are of particular interest for reproducing high-dimensional potential energy surfaces (PES). Behler and Parrinello suggested a construction, which allows the application of NNs to systems of different sizes.[3] In their approach, the total energy $\hat{E}$ of a molecular system is decomposed into a sum of atomic contributions $\hat{E}_i$

$$\hat{E} = \sum_{i=1}^{N_{\text{at}}} \hat{E}_i = \sum_{i=1}^{N_{\text{at}}} \text{NN}_{Z_i}(\mathbf{x}_{\text{in}}^{(i)}) \tag{13}$$

where NN_{Z_i} denotes the neural network output, and $\mathbf{x}_{\text{in}}$ is a molecular representation. In their approach, an individual neural network NN_{Z_i} is constructed and trained for each atomic species Z_i. In our approach, a similar construction is used. Since the Gaussian moment representation $\rho_{i,s_1,s_2,\ldots}$ contains the information about the atomic species via the coefficients $\beta_{Z_i,Z_j,s}$, a single NN is constructed and trained for all species. This results in the expression for the total energy

$$\hat{E} = \sum_{i=1}^{N_{\text{at}}} \text{NN}(\mathbf{x}_{\text{in}}^{(i)} = \{\rho_{i,s_1}, \rho_{i,s_1,s_2}, \ldots\}) \tag{14}$$

The approach presented here is atom-centered and, thus, allows the modeling of molecular systems with a variable number of atoms.

In this work, a feed-forward neural network is used. In a feed-forward NN, an input layer is connected to an output layer via one or multiple hidden layers. The information in the network passes only in a single direction toward the output layer. The local molecular descriptor, i.e., $\mathbf{x}_{\text{in}}^{(i)} = \{\rho_{i,s_1}, \rho_{i,s_1,s_2}, \ldots\}$,

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

provides the values of the neurons in the input layer, while the output of the NN is the atomic energy, $\hat{E}_i$. A linear transformation is applied to the input data for each layer followed by a nonlinear activation function, i.e., for two hidden layers

$$\mathbf{y}_{\text{out}} = \phi_{\text{out}}(\phi_2(\phi_1(\mathbf{x}_{\text{in}}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2)\mathbf{W}_{\text{out}} + \mathbf{b}_{\text{out}}) \quad (15)$$

where $\mathbf{W}_k$ are the weight matrices, $\mathbf{b}_k$ are the biases, and ϕ_k are activation functions. For the output layer, a linear activation function is used, whereas for the hidden layers nonlinear activation functions are applied. In this work, a "rectifier"-like function, the soft-plus function $\phi_i(x) = \ln(1 + \exp(x))$, was chosen as the nonlinear activation function. We found it to perform better than other standard activation functions for the data sets used here. In order to maximize the use of the nonlinear region of the activation functions, the atomic energy is scaled and shifted as $\hat{E}_i = \sigma_{Z_i} y_{\text{out},i} + \mu_{Z_i}$. The parameters σ_{Z_i} and μ_{Z_i} depend on atomic species and are optimized during the training procedure. The initialization of σ_{Z_i} and μ_{Z_i} is performed by using the standard deviation and mean of the per-atom average of the reference energies in the training set to improve the convergence of the model.

A schematic representation of an atom-centered feed-forward NN and the computational procedure of the presented GM model is shown in Figure 3. First, a neighborhood of all atoms within the cutoff radius R_{max} is assigned to each atom i. Next, given the parameters γ_s, the radial functions $\Phi_s(r)$ are evaluated. Using the coefficients $\beta_{Z_i,Z_j,s}$, which are initiated randomly, the tensor-valued function Ψ_i centered at the atom i is constructed. Then, the predefined tensor contractions are applied, and the molecular representation ρ is calculated. It is used as input to the feed-forward NN which outputs scaled atomic energies, $y_{\text{out},i}$. These are transformed back to nonscaled values, $\hat{E}_i$, which are summed to result in the total energy of the system.

In total, two network architectures, a shallow and a deep NN, are constructed to test our model on benchmark data sets as discussed in Section 3. The shallow network has two hidden layers with [256,128] nodes, respectively. The deep network consists of five hidden layers with [1024,512,256,128,64] nodes each. We refer to the shallow model as GM-sNN and to the deep model as GM-dNN.

2.3. Training. In this work, we are interested in the prediction of energies and forces and possibly Hessians in the future. Therefore, prior to describing the training procedure, a few sentences are dedicated to the importance of the incorporation of forces into the training. For quantum chemical training data, obtaining forces for all atoms is about as computationally expensive as obtaining the energy. Thus, forces provide additional training data which are comparably cheap to obtain. Therefore, they are included in the training of the model.

To optimize weights and biases of each layer of the GM model, the training loss function is defined as

$$\mathcal{L} = w_{\text{E}} \| \hat{E} - E^{\text{ref}} \|^2 + \frac{w_{\text{F}}}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{k=1}^{3} \| \hat{F}_{i,k} - F_{i,k}^{\text{ref}} \|^2 \quad (16)$$

To control the energy and force contribution during the training, we define the adjustable parameters w_{E} and w_{F}. The parameters were set to $w_{\text{E}} = 1$ and $w_{\text{F}} = 100$ Å^2 for all models. The higher weight of the force error is motivated by the fact that forces alone determine the dynamics of a chemical system. Consequently, the accurate force prediction is most important for MD simulations. In case the model is trained only on energies, the parameter w_{F} is set to zero. The parameters, w_{E} and w_{F}, were chosen according to performance tests of the GM-NN model. However, optimal values are likely to depend on the system under study, and the parameters should be adjusted accordingly. A more thorough investigation of the dependence of the performance on the parameters is planned for the future works.

The reference values for the force and energy are denoted by E^{ref} and $\mathbf{F}^{\text{ref}}$, respectively. Atomic forces $\hat{\mathbf{F}}$ are calculated from the total energy $\hat{E}$ analytically by taking the partial derivative with respect to atomic positions. For an atom i along the component $k \in \{x,y,z\}$, the atomic force is defined as

$$\hat{F}_{i,k}(Z_1, Z_2, ..., Z_{N_{\text{at}}}, \mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{N_{\text{at}}}) = -\frac{\partial \hat{E}}{\partial r_{i,k}}(Z_1, Z_2, ..., Z_{N_{\text{at}}}, \mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{N_{\text{at}}}) \quad (17)$$

All models used in Section 3 were implemented in the Tensorflow[35] framework. Atomic forces were calculated using automatic differentiation.[36] The training loss in eq 16 was minimized using the AMSGrad optimizer[37] with 32 molecules per mini-batch with an exception of the models trained on the ISO17[14,15,31] data set, where a mini-batch of 128 molecules was used. The learning rate was set to 10^{-3} for all models and kept constant throughout the whole training procedure. Each optimization took 5000 training epochs with an exception of the models trained on 1000 MD17[14,21,22] samples, where we optimized for 10,000 epochs. Overfitting was prevented using the early stopping technique.[38] After each epoch, the training loss was evaluated on a validation set. After training, the model that performed best on the validation set was selected for further application on the test sets. So, although the validation data was not used directly in the training procedure, it indirectly influenced models chosen at the end.

2.4. Scalability and Computational Cost. To achieve linear scaling of the computational cost and memory usage, the GM-NN model uses atom neighbor lists as implemented in ASE.[39] This allows the calculation of the energy and gradient for a structure with up to 100,000 atoms in less than 230 s on a single Intel Xeon CPU E5-2670 0. The memory required for the respective calculations with up to 25,000 atoms is about 9.7 GB. This allows efficient training and inference on typical GPUs for large systems. Further information on the computational cost and memory usage, including details on the trained model, is provided in the Supporting Information.

3. RESULTS

Here, we apply the NN model based on Gaussian moments (GM-NN) to three well-established quantum chemistry data sets: QM9,[30,31] MD17,[14,21,22] and ISO17.[14,15,31] These data sets are designed such that different aspects of chemical space are covered. For all data sets, we report the mean absolute error (MAE) and the root-mean-square error (RMSE) in kcal/mol for the energies and in kcal/mol/Å for the forces.

The deep network model GM-dNN was tested only on large training sets, i.e., 50,000 training samples from the MD17 data set and 400,000 training samples from the ISO17 data set. The reason for this is that for smaller training sets, e.g., 1000 samples from the MD17 data set and the QM9 data set, the

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

shallow GM-sNN model is already sufficient to reach an acceptable accuracy within the given number of training epochs. The deep architecture is prone to overfitting, especially for small training sets. The deep architecture is promising for large and complex training sets, because it is known that the additional hidden layers enhance the capability of neural networks to capture complexity and high nonlinearity of functional dependence.[40,41]

The input layer for both architectures has 427 neurons as discussed in Section 2.1. The only remaining adjustable parameter of the descriptor is the cutoff radius R_{max}. It was set to 3.0 Å for the QM9 data set and to 4.0 Å for the MD17 and ISO17 data sets. In each experiment, the data set is split into a training set of size N and a validation set containing 2000 structures used for early stopping. The remaining data was used for testing the models.

3.1. QM9. QM9[30,31] is a widely used benchmark for the prediction of several properties of molecules in equilibrium. Thus, all forces vanish. They were not included into the training loss function. Only shallow GM-sNN models were trained on the QM9 data set.

The QM9 data set consists of 133,885 neutral, closed-shell organic molecules with up to 9 heavy atoms (C, O, N, F) and a varying number of hydrogen (H) atoms. The largest structure in the data set contains 29 atoms in total. Since 3054 molecules from the original QM9 data set failed a consistency test,[31] we used only the remaining 130,831 structures in the following experiments.

For QM9, a cutoff radius of R_{max} = 3.0 Å was chosen. This is rather small compared to the 10 Å used in the message-passing architectures, e.g., SchNet[15,16] or PhysNet.[13] However, the sphere defined by the small cutoff radius of 3.0 Å includes already a maximum of 24 neighbors out of 28 possible neighboring atoms for the largest structures in the data set. This holds for central atoms of the respective structures. For the side atoms, smaller local environments can be found which can be transferred to the smaller structures in the data set. So, the smaller cutoff improves the ability of the model to generalize. Thus, the cutoff radius has to be increased only in the case some important interactions are neglected, which is not the case for the QM9 data set.

The learning curves of the model are shown in Figure 4. They show the dependence of the MAE and the RMSE on the training set size. For training set sizes of 1000, 5000, and 10,000, the results are obtained by averaging over five independent choices of the training set. For 25,000, 50,000, 100,000, and 110,426 structures, only three independent choices of the training set are averaged. The GM-sNN trained on 110,426 reference energies predicts energies of the remaining structures with a MAE of 0.27 kcal/mol and a RMSE of 0.63 kcal/mol. The required accuracy of 1 kcal/mol in the case of the MAE is reached already when training on 5000 reference structures.

A comparison of the GM-sNN model to the various models published in the literature can be found in Table 1. It can be seen that the performance of the GM-sNN model is comparable to all methods shown. However, one can see that the MTM_{16-28} model[11] and the model in ref 8 perform slightly better when training on 1000 and 5000 reference samples. The MTM_{16-28} model employs geometric moments to construct rotationally invariant bases for linear regression. The model in ref 8 uses NNs as an ML method and the power spectrum of spherical harmonics as a structural descriptor. In

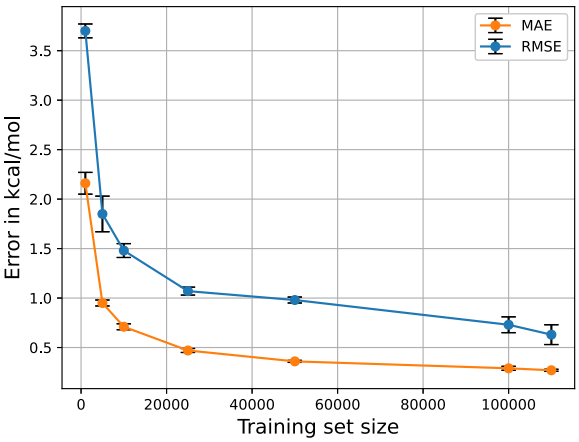


Figure 4. Mean absolute error (MAE) and root-mean-square error (RMSE) in kcal/mol of the energy prediction on the QM9 data set depending on the number of structures in the training set. Results for all training set sizes are averaged over three and five independent randomly chosen training data sets, see text. The error bars indicate the standard deviation.

both methods, atomic species and environment are encoded simultaneously. With an increasing number of training samples, the GM-sNN model outperforms the MTM_{16-28} model and the model in ref 8. The GM-sNN model reaches an accuracy comparable to the message-passing models, e.g., SchNet,[15,16] HIP-NN,[17] and PhysNet.[13] The message-passing models learn to construct invariant features from nuclear charges and interatomic distances in a data-driven manner. This approach was first introduced by the DTNN.[14]

We also investigated how well a model trained on small molecules transfers to larger systems. For this purpose, the QM9 data set was divided into two subsets. The first subset contains molecules with up to 15 atoms and has 24,978 structures in total. The other subset which is used for testing has molecules with more than 15 atoms and has 105,853 structures in total. We used 22,978 structures of the first subset for training and another 2000 for validation. The errors on the test set of all 105,853 structures are averaged over three independent choices of the training set and are MAE = 1.01 kcal/mol and RMSE = 1.65 kcal/mol. This demonstrates that the trained models can be transferred from small to large structures. However, the performance deteriorates compared with the randomly chosen structures (see Table 1).

All models for the QM9 data set were trained on an NVIDIA Tesla V100-SXM2-32GB GPU. The training of 5000 epochs took from 1 h (1000 structures) to 3 days (110,426 structures).

3.2. MD17. The MD17 data set[14,21,22] is a collection of structures, energies, and atomic forces of eight small organic molecules obtained from ab initio molecular dynamics (MD). For each molecule, a large variety of conformations is covered. The data set varies in size from 150,000 to almost 1,000,000 conformations. It covers energy differences from 20 to 48 kcal/mol and force components ranging from 266 to 570 kcal/mol/Å. The task of this experiment is to predict energies and forces for these molecules using various models.

We have chosen a cutoff radius of R_{max} = 4.0 Å, since already 19 of the 20 possible neighboring atoms of the central atoms of the aspirin molecule (acetylsalicylic acid), the largest molecule

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

Table 1. MAEs in kcal/mol for Energy Prediction on QM9 Data Set[30,31] for Various Models Reported in Literature and Different Sizes of Training Set[a]

Training Set Size	DTNN[14]	SchNet[15,16]	PhysNet[13]	HIP-NN[17]	MTM_{16-28}[11]	Ref 8	GM-sNN
1000	–	–	–	–	**1.8**	1.85	2.16
5000	–	–	–	–	**0.90**[c]	0.95	0.95
10,000	–	1.28[b]	–	–	0.86	0.73	**0.71**
25,000	1.04	0.80[b]	–	–	0.63	0.55	**0.47**
50,000	0.94	0.59	**0.30**	0.35	0.41	0.46	0.36
100,000	0.84	0.34	**0.19**	0.26	–	0.41	0.29
110,426	–	0.31	**0.19**	0.26	–	–	0.27

[a]Results of the GM-sNN model are averaged over three and five independent randomly chosen training data sets, see text. [b]As estimated from the graphs in ref 16. [c]As estimated from the graphs in ref 11.

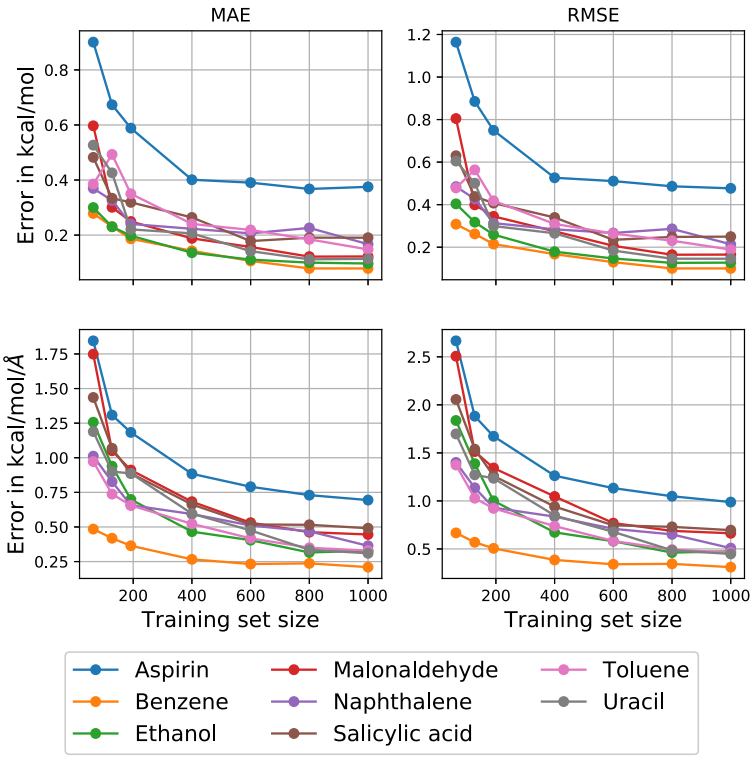


Figure 5. Mean absolute error (MAE) and root-mean-square error (RMSE) in kcal/mol and kcal/mol/Å of energy (top) and force (bottom) predictions, respectively, on the MD17 data set depending on the number of structures in the training data set. All results were obtained using the shallow architecture GM-sNN.

of the MD17 data set, lie withing a sphere defined by this cutoff.

In a first test, we investigated the learning curves of the shallow GM-sNN model trained on structures from the MD17 data set. For this purpose, we trained the model on 64, 128, 192, 400, 600, 800, and 1000 randomly chosen samples. The respective learning curves for eight small organic molecules are presented in Figure 5. From the figure, it is noticeable that already 64 samples are enough to achieve an accuracy of 1 kcal/mol in energy. For most molecules, at least 192 training samples are necessary to achieve an accuracy of 1 kcal/mol/Å of the forces. Aspirin requires 400 samples, but benzene requires only 64 due to its rigid conformation.

It may be noticed that the force learning curves look smoother than the energy learning curves. This is because a large weighting factor of w_F = 100 Å^2 was used in the loss function for the forces. Thus, most emphasis was given on the force training. In a typical example, ≈99.7% of the loss at the end of the training is caused by the forces. However, in all cases, training could be continued which would lead to smaller force errors and to a higher impact of energies on the training. Further training would make the energy learning curves smoother.

A comparison of GM-NN models to several models recently published in the literature can be found in Table 2. The GM-sNN models were trained on N = 1000 and N = 50,000 samples; the GM-dNN models were trained on N = 50,000 samples. The results of all models are averaged over three randomly chosen training sets. From Table 2, we see that the GM-NN models yield an accuracy which is comparable with those of all well-established methods. The best training result is written in bold face. The shallow GM-sNN model outperforms

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

Table 2. Mean Absolute Errors for Energy and Force Prediction in kcal/mol and kcal/mol/Å, Respectively[a]

		N = 1000				N = 50,000			
		GDML[21]	EANN[29]	SchNet[15]	GM-sNN	SchNet[15]	PhysNet[13]	GM-sNN	GM-dNN
Benzene	energy	**0.07**	–	0.08	0.08 (0.008)	**0.07**	**0.07** (0.002)	**0.07** (0.003)	**0.07** (<0.001)
	force	0.23	–	0.31	**0.21** (0.021)	0.17	0.15 (0.001)	**0.14** (<0.001)	**0.14** (0.001)
Toluene	energy	0.12	**0.11**	0.12	0.15 (0.009)	**0.09**	0.10 (0.004)	0.10 (0.006)	**0.09** (0.003)
	force	**0.24**	0.38	0.57	0.34 (0.012)	0.09	**0.03** (0.002)	0.10(0.003)	0.06 (<0.001)
Malonaldehyde	energy	0.16	0.14	0.13	**0.12** (0.012)	0.08	**0.07** (<0.001)	**0.07** (0.003)	**0.07** (<0.001)
	force	0.8	0.62	0.66	**0.45** (0.014)	0.08	**0.04** (0.002)	0.08 (0.006)	0.05 (0.006)
Salicylic acid	energy	**0.12**	0.14	0.20	0.19 (0.020)	**0.10**	0.11 (0.005)	0.11 (0.002)	0.11 (0.002)
	force	**0.28**	0.51	0.85	0.49 (0.021)	0.19	**0.04** (**0.001**)	0.14 (0.001)	0.08 (0.002)
Aspirin	energy	**0.27**	0.33	0.37	0.38 (0.015)	**0.12**	**0.12** (0.005)	0.19 (0.006)	0.13 (0.004)
	force	0.99	0.99	1.35	**0.69** (0.025)	0.33	**0.06** (0.002)	0.26 (0.009)	0.12 (0.008)
Ethanol	energy	0.15	0.10	**0.08**	0.10 (0.007)	**0.05**	**0.05** (<0.001)	**0.05** (<0.001)	**0.05** (0.002)
	force	0.79	0.47	0.39	**0.33** (0.017)	0.05	**0.03** (<0.001)	0.06 (0.005)	0.04 (0.001)
Uracil	energy	**0.11**	**0.11**	0.14	0.12 (0.008)	**0.10**	**0.10** (0.001)	**0.10** (<0.001)	**0.10** (0.001)
	force	**0.24**	0.35	0.56	0.33 (0.016)	0.11	**0.03** (<0.001)	0.07 (0.005)	0.04 (<0.001)
Naphthalene	energy	**0.12**	**0.12**	0.16	0.17 (0.011)	**0.11**	0.12 (0.011)	0.13 (0.017)	**0.11** (0.004)
	force	**0.23**	0.27	0.58	0.36 (0.023)	0.11	**0.04** (0.001)	0.13 (0.012)	0.08 (0.008)

[a]The results are obtained by averaging over three independent choices of the training sets; their standard deviation is given in parentheses. The GM-NN models are trained on 1000 and 50,000 training samples. All models are trained on energies and forces with the exception of the GDML[21] model, which is trained on forces only.

the message-passing model SchNet when trained on 1000 and 50,000 reference samples. The deep GM-dNN model reaches the accuracy of the PhysNet model. All mentioned message-passing models have more complicated mathematical forms and deeper NN architectures than our GM-NN models. Therefore, their capability of interpolation can potentially be better.

The GDML model is more accurate than our GM-sNN for the smaller molecules, although even there the difference is small (see Table 2). Note that the GDML[21] model was trained on forces only and, in general, scales badly with the number of reference structures due to its kernel nature. For small data sets and complex molecules, like aspirin, our GM-sNN model outperforms all presented methods in the force prediction. The force error on the aspirin data set is smaller by 0.3 kcal/mol/Å than the respective predictions of the GDML and EANN models and smaller by 0.66 kcal/mol/Å than the SchNet predictions. The EANN model employs density-like descriptors and NNs as an ML method. The errors of GM-sNN in energy prediction could be improved training for more epochs (see the previous discussion).

In addition to the models listed in Table 2, we can compare to sGDML[22] an extension of the GDML model that incorporates rigid space group symmetries and dynamic nonrigid symmetries, e.g., methyl group rotations. The performance is similar. For example, the accuracy of the sGDML force prediction is 0.68 kcal/mol/Å for the aspirin data set, while GM-sNN results in 0.69 kcal/mol/Å. The GM-sNN model needs fewer reference structures, less than 400, to achieve an accuracy of 1 kcal/mol/Å, compared to the sGDML model, which needs about 600 reference structures. Note that in this comparison it was assumed that the chosen training data is similarly correlated.

We use the MD17 data set to test the dependence of the performance of the GM-sNN model on the size of our descriptor, the number of Gaussian moments (#GM). Figure 6 shows that the force error is reduced algebraically with the increasing size of the descriptor. For aspirin and $N = 1000$, we obtain an MAE of the forces of about $4.145 \cdot (\#GM)^{-0.309}$ kcal/mol/Å. A similar algebraic convergence of the error in the energy prediction is illustrated in Figure S2 of the Supporting Information. To compare the performance with typical hand-crafted descriptors, atom-centered symmetry functions (ACSF)[3,4] were chosen. In ref 18, it was shown that a typical Behler–Parrinello model with ACSFs as molecular descriptors is consistently outperformed by the SchNet model. For example, on the aspirin data set, a MAE of 1.92 kcal/mol/Å in predicted forces was achieved using 51 ACSF invariant scalars with 1000 training structures. For comparison, the GM-sNN model achieves an MAE of 1.43 kcal/mol/Å in predicted forces using only 35 GM descriptors and an MAE of 1.19 kcal/mol/Å with 48 GM descriptors (see Figure 6). This shows that the proposed descriptor outperforms ACSFs and captures all necessary information about the molecular structure as efficiently as message-passing architectures. Due to their particular mathematical form, GM achieve the desired

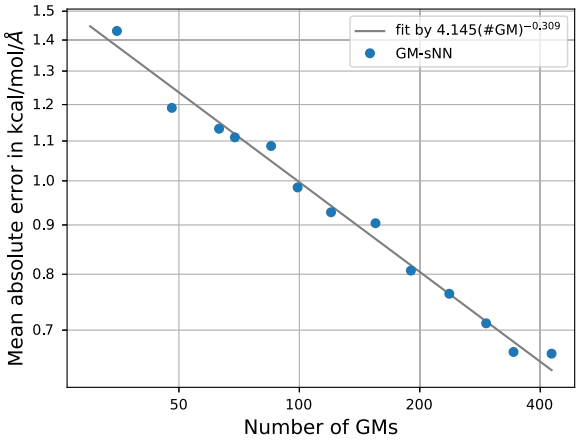


Figure 6. Log–log plot of the algebraic decrease of the error in the predicted forces with an increasing number of Gaussian moments. All values are given on the test data of the aspirin data set for the GM-sNN model ($N = 1000$).

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410–5421

Table 3. Mean Absolute Errors for Energy and Force Prediction on Two Variants of ISO17 Benchmark in kcal/mol and kcal/mol/Å, Respectively[a]

		SchNet[15]	PhysNet[13]	GM-sNN	GM-dNN
known molecules/unknown conformations	energy	0.36	**0.10** (<0.001)	0.40 (0.020)	0.17 (0.003)
	force	1.00	**0.12** (0.002)	0.65 (0.019)	0.28 (0.011)
unknown molecules/unknown conformations	energy	2.40	2.94 (0.260)	**1.97** (0.414)	2.71 (0.640)
	force	2.18	**1.38** (0.060)	1.66 (0.082)	1.96 (0.189)

[a]The results are obtained by averaging over three independent choices of the training sets; their standard deviation is given in parentheses.

flexibility and, thus, even GM-sNN outperforms the SchNet model in several tests (see Table 2).

All GM-NN models for the MD17 data set were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each. The training of the GM-sNN model on 1000 structures for 10,000 epochs took 4 h, and the training on 50,000 structures for 5000 epochs was carried out during 2 days. The GM-dNN model required at most 2 days and 15 h for the training.

3.3. ISO17. The ISO17 data set[14,15,31] contains short MD trajectories of 127 isomers with the composition $C_7O_2H_{10}$, drawn randomly from the QM9 data set. For all molecules, energies and atomic forces are provided. Each trajectory samples 5000 conformations. In total, the data set contains 635,000 structures.

The experiment was arranged as follows. The data set was split into two subsets. The first subset contained the data of ≈80% of all molecules. From this subset, 400,000 structures were taken randomly for training, and another 4000 structures were used for validation. The remaining 101,000 structures were used for testing the model. This test is referred to as "known molecules/unknown conformations". Then, we applied the trained model to the remaining ≈20% of all molecules, those which the model had not seen before. This second test is referred to as "unknown molecules/unknown conformations". It allows us to test the generalization capability of the GM-NN model.

The results of both tests obtained with the GM-sNN and GM-dNN models are compared to recent literature data in Table 3. The results of the GM-NN models are obtained by averaging over three randomly chosen training sets. From the table, it is noticeable that the GM-sNN model outperforms the SchNet model in three of the four tests. The shallow model also outperforms both message-passing models in the energy prediction for "unknown molecules/unknown conformations". The energy error is about 0.43 kcal/mol lower than the SchNet prediction and 0.97 kcal/mol lower than the PhysNet prediction. This shows that GM-sNN generalizes better than the models from the literature.

The deep GM-dNN model outperforms the shallow GM-sNN model and approaches the accuracy of PhysNet when applied to the "known molecules/unknown conformations" test. However, using the deep architecture deteriorates the performance on the "unknown molecules/unknown conformations" test. This indicates that the larger, more flexible network learns more details of the "known molecules/unknown conformations" test set at the expense of generalization capabilities, tested on the unknown molecules.[42] This example shows that a thorough choice of the network architecture is of crucial importance for the specific task for which the model is to be designed.

All GM-NN models were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each for 5000 training epochs. The training of the GM-sNN model took ≈7 days; the training of the GM-dNN took ≈7 days and 6 h. Note that the results of the PhysNet model were obtained after training for ≈1 month.[13]

3.4. MD of Ethanol with Ab Initio Accuracy. The predictive power of the machine-learned potentials was tested on a simple organic molecule, namely, ethanol. We calculated the energy profile for the ethanol rotamers, i.e., for the rotation of the OH group around the C−O bond and the rotation of the CH_3 group around the C−C bond. A comparison of the predictions is made based on machine-learned potentials to the potential energy profile calculated at the PBE-D3(BJ)/6-31G* level of theory[43−46] using Turbomole 7.1[47] within ChemShell[48,49] and is shown in Figure 7.

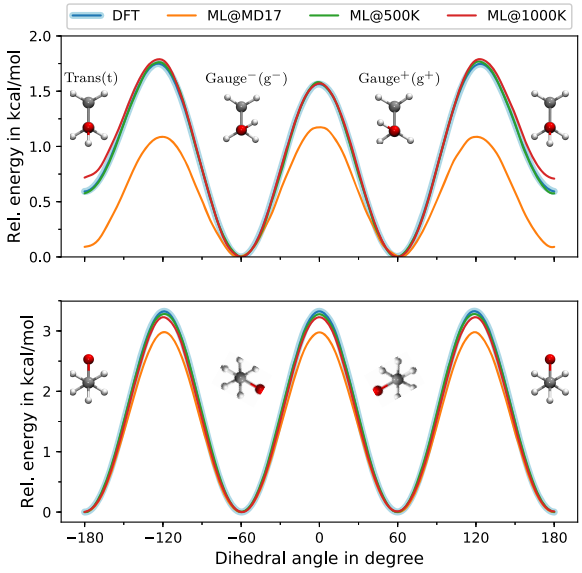


Figure 7. Potential energy profile of the dihedral angle describing the rotation (top) of the OH group around the C−O bond and (bottom) of the rotation of the CH_3 group around the C−C bond. The GM-sNN model was trained on the MD17 data set, ML@MD17, on the 500 K data set, ML@500K, and on the 1000 K data set, ML@1000K.

It is noticeable that the model trained on the MD17 data set for ethanol, ML@MD17, (we took the GM-sNN model trained on 50,000 structures, Section 3.2) shows large deviations in the barrier heights. This is probably caused by the slightly different levels of theory: MD17 used PBE+vdW-TS (we were unable to find information on the basis used to create MD17[21]). While the functionals are the same, the different treatment of dispersion and the different basis set in the reference may lead to the deviation of the energy profiles.

To ensure the reproducibility of the tests, we generated two different data sets for ethanol on the same level of theory as for

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

the respective DFT profile. The data sets were taken from ab initio MDs at 500 and 1000 K. In the following, we describe the generation of the data sets. First, we performed Born−Oppenheimer MD at 500 and 1000 K in the *NVT* ensemble using the Berendsen thermostat with GFN2-xTB[50,51] as the underlying quantum mechanical method. The time step was set to 0.5 fs, and the dynamics was run for 50,000 steps resulting in 25.0 ps of dynamics. Every 10 steps, a geometry was taken from the dynamics, and the energy, as well as atomic forces, were recalculated at the PBE-D3(BJ)/6-31G* level of theory. The MD was performed within ChemShell, and for the refinement with DFT, we used Turbomole 7.1 within ChemShell. For each data set, we obtained in total 5000 structures. The additional data set at 1000 K was created because the barrier for the rotation of the CH_3 group around the C−C bond is way higher than 500 K. Both data sets can be found in a git repository.[52]

The GM-sNN model was trained using 4000 reference structures for 5000 training epochs. Training of the model was performed on an NVIDIA Tesla V100-SXM2-32GB GPU, and it took about 4.5 h for each data set. The remaining 1000 structures were used for validation. We refer to the model trained on the 500 K data set as ML@500K and to the one trained on the 1000 K data set as ML@1000K. From Figure 7, it can be seen that the model trained on the generated data sets fits the DFT profile well, and all deviations are small. All barriers are given in Table 4.

Table 4. Energetic Barriers in kcal/mol Predicted by Machine-Learned Potentials and Calculated at the PBE-D3(BJ)/6-31G* Level of Theory

	OH			CH_3
	$t \rightarrow g^-$	$g^- \rightarrow t$	$g^- \rightarrow g^+$	
PBE-D3(BJ)/6-31G*	1.16	1.75	1.56	3.32
ML@MD17	1.00	1.09	1.17	2.98
ML@500K	1.19	1.76	1.58	3.28
ML@1000K	1.07	1.79	1.57	3.23

To test the prediction of frequencies, even though only energies and forces were used for the training, we calculated the vibrational power spectrum of ethanol based on the ML@1000K model using the velocity−velocity autocorrelation function. In this formalism, the intensity of a transition is proportional to

$$I \propto \left| \int \langle v(t_0) v(t + t_0) \rangle \exp(-i\omega t) \mathrm{d}t \right|^2 \tag{18}$$

Velocities for the calculation of the power spectrum were obtained by running MD trajectories on the ML@1000K model within ASE[39] using a Langevin thermostat at the temperatures of 500 and 100 K. The time step was set to 0.5 fs, and the dynamics were run for 40 ps. The first 1 ps was ignored. The final spectra obtained from MDs at 100 and 500 K are shown in Figure 8.

In Figure 8, one can, for example, find bands at 3611 cm^{-1} (500 K) and 3644 cm^{-1} (100 K) which correspond to the O−H stretching of alcohol. This is very similar to the corresponding harmonic frequency from DFT at the PBE-D3(BJ)/6-31G* level, 3638 cm^{-1}. The experimental values for ethanol in the gas phase range from 3649 to 3682 cm^{-1},[53]

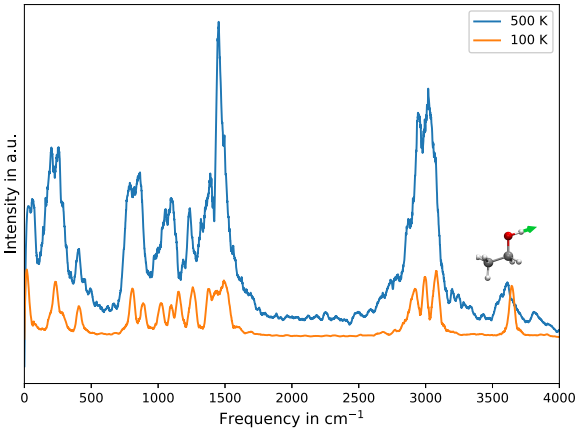


Figure 8. Vibrational power spectrum of ethanol obtained via velocity−velocity autocorrelation function and the expression in eq 18. The velocities are obtained from MD at 500 and 100 K using the GM-sNN model trained on the 1000 K data set, see text for details.

which are also close to the values predicted using ML potentials.

4. CONCLUSIONS

In the present work, we proposed Gaussian moments as a representation for molecular structures that incorporates global symmetries, i.e., the invariances with respect to rotation and translation of the entire system and the invariance with respect to permutation of atoms of the same species. The particular advantage of constructing GM is that the GM representation can be written in terms of pairwise distance vectors and tensor contractions. This allows for an efficient calculation of them on graphics processing units (GPUs). The representation can easily be extended by generating further rotationally invariant scalars from additional generating graphs. Thus, an even larger basis can be constructed if needed, at almost the same computational cost.

We have demonstrated that the GM descriptor can be used as input for machine learning algorithms. In this work, we used feed-forward NNs as a machine learning method for the regression. We evaluated the GM-NN models on three different quantum-chemical benchmark data sets, which cover both chemical and conformational variability. On the basis of the performed tests, we can argue that the GM-NN models show comparable or better accuracy with respect to the state-of-the-art machine learning models. The performance of GM with only two hidden layers is similar to that of message-passing models, such as SchNet[15,16] and PhysNet,[13] which have much deeper and mathematically more complicated NN architectures.

We have shown that a GM model trained on small reference structures is able to generalize to larger structures. Additionally, it was shown that the respective GM descriptor is able to capture all necessary information about the molecular structure so that the machine learns as efficiently as respective models which include all possible symmetries explicitly.

In addition to the benchmark data sets, machine-learned potentials based on Gaussian moments were applied to predict rotamers and the vibrational power spectrum of the ethanol molecule. We have seen that the GM-NN potentials are capable of capturing differences between the gauge and trans

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

conformations of ethanol and to capture vibrational frequencies even though they were trained on energies and forces only.

In summary, we have presented an approach for constructing a machine learning model based on tensor contractions, which fulfills physical constraints and is inspired by the molecular wave function. This model has been proven to be generally applicable to molecular systems and, therefore, can potentially be applied to large scale molecular simulations.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.0c00347.

Additional data on scalability and computational cost, as well as a figure showing the decrease in the error in the energy prediction with an increase in the descriptor size (PDF)

■ AUTHOR INFORMATION

Corresponding Author

J. Kästner − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0001-6178-7669; Email: kaestner@theochem.uni-stuttgart.de

Author

V. Zaverkin − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.0c00347

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors acknowledge financial support received in the form of a Ph.D. scholarship from the Studienstiftung des Deutschen Volkes (German National Academic Foundation). We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075-390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech) and the European Union's Horizon 2020 research and innovation programme (Grant Agreement No. 646717, TUNNELCHEM). We also would like to acknowledge the support by the Institute for Parallel and Distributed Systems (IPVS) of the University of Stuttgart for providing computer time.

■ REFERENCES

(1) Born, M.; Oppenheimer, R. Zur Quantentheorie der Molekeln. *Ann. Phys.* **1927**, *389*, 457−484.

(2) Mackerell, A. D., Jr. Empirical force fields for biological macromolecules: Overview and issues. *J. Comput. Chem.* **2004**, *25*, 1584−1604.

(3) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(4) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, No. 074106.

(5) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.

(6) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *87*, 184115.

(7) Khorshidi, A.; Peterson, A. A. Amp: A modular approach to machine learning in atomistic simulations. *Comput. Phys. Commun.* **2016**, *207*, 310−324.

(8) Unke, O. T.; Meuwly, M. A reactive, scalable, and transferable model for molecular energies from a neural network approach based on local information. *J. Chem. Phys.* **2018**, *148*, 241708.

(9) Kocer, E.; Mason, J. K.; Erturk, H. A novel approach to describe chemical environments in high-dimensional neural network potentials. *J. Chem. Phys.* **2019**, *150*, 154102.

(10) Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **2016**, *14*, 1153−1173.

(11) Gubaev, K.; Podryabinkin, E. V.; Shapeev, A. V. Machine learning of molecular properties: Locality and active learning. *J. Chem. Phys.* **2018**, *148*, 241727.

(12) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, No. 058301.

(13) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678−3693.

(14) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **2017**, *8*, 13890.

(15) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 991−1001.

(16) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet A deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722.

(17) Lubbers, N.; Smith, J. S.; Barros, K. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.* **2018**, *148*, 241715.

(18) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K.-R. SchNetPack: A Deep Learning Toolbox For Atomistic Systems. *J. Chem. Theory Comput.* **2019**, *15*, 448−455.

(19) Artrith, N.; Urban, A.; Ceder, G. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *96*, No. 014112.

(20) 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. *Chem. Sci.* **2018**, *9*, 2261−2269.

(21) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **2017**, *3*, e1603015.

(22) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **2018**, *9*, 3887.

(23) von Lilienfeld, O. A.; Ramakrishnan, R.; Rupp, M.; Knoll, A. Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties. *Int. J. Quantum Chem.* **2015**, *115*, 1084−1093.

(24) Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930−17955.

(25) Flusser, J.; Suk, T.; Zitová, B. *Moments and Moment Invariants in Pattern Recognition*; John Wiley & Sons, Ltd., 2009; Chapter 2, pp 13−47.

(26) Flusser, J.; Suk, T.; Zitová, B. *2D and 3D Image Analysis by Moments*; John Wiley & Sons, Ltd., 2016; Chapter 4, pp 95−162.

(27) Suk, T.; Flusser, J. Tensor Method for Constructing 3D Moment Invariants. In *Computer Analysis of Images and Patterns*; Real, P., Diaz-Perni, D., Molina-Abril, H., Berciano, A., Kropatsch, W., Eds.; Springer: Berlin, Heidelberg, 2011; pp 212−219.
(28) Yang, B.; Suk, T.; Dai, M.; Flusser, J. 2D and 3D Image Analysis by Gaussian-Hermite Moments. *Gate to Computer Science and Research* **2014**, *1*, 143−173.
(29) Zhang, Y.; Hu, C.; Jiang, B. Embedded Atom Neural Network Potentials: Efficient and Accurate Machine Learning with a Physically Inspired Representation. *J. Phys. Chem. Lett.* **2019**, *10*, 4962−4967.
(30) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(31) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.
(32) Weyl, H. *The Classical Groups: Their Invariants and Representations*; Princeton University Press: Princeton, NJ, 1966.
(33) Boys, S. F.; Egerton, A. C. Electronic wave functions - I. A general method of calculation for the stationary states of any molecular system. *Proc. R. Soc. London A* **1950**, *200*, 542−554.
(34) Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251−257.
(35) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*; TensorFlow, 2015. https://www.tensorflow.org/ (accessed July 2020).
(36) Baydin, A.; Pearlmutter, B. A.; Radul, A. A.; Siskind, J. M. Automatic Differentiation in Machine Learning: a Survey. *J. Mach. Learn. Res.* **2018**, *18*, 1−43.
(37) Reddi, S. J.; Kale, S.; Kumar, S. On the Convergence of Adam and Beyond, *arXiv:1904.09237*, 2019.
(38) Prechelt, L. In *Neural Networks: Tricks of the Trade*, Second ed.; Montavon, G., Orr, G. B., Müller, K.-R., Eds.; Springer: Berlin, Heidelberg, 2012; pp 53−67.
(39) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dulak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The atomic simulation environmenta Python library for working with atoms. *J. Phys.: Condens. Matter* **2017**, *29*, 273002.
(40) Bengio, Y.; Lamblin, P.; Popovici, D.; Larochelle, H. In *Advances in Neural Information Processing Systems 19*; Schölkopf, B., Platt, J. C., Hoffman, T., Eds.; MIT Press, 2007; pp 153−160.
(41) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C. J. C., Bottou, L., Weinberger, K. Q., Eds.; Curran Associates, Inc., 2012; pp 1097−1105.
(42) Neyshabur, B.; Tomioka, R.; Srebro, N. In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning, *arXiv:1412.6614*, 2014.
(43) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(44) 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. *J. Chem. Phys.* **2010**, *132*, 154104.
(45) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the damping function in dispersion corrected density functional theory. *J. Comput. Chem.* **2011**, *32*, 1456−1465.
(46) Rassolov, V. A.; Pople, J. A.; Ratner, M. A.; Windus, T. L. 6-31G* basis set for atoms K through Zn. *J. Chem. Phys.* **1998**, *109*, 1223−1229.
(47) Furche, F.; Ahlrichs, R.; Hättig, C.; Klopper, W.; Sierka, M.; Weigend, F. Turbomole. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2014**, *4*, 91−100.
(48) Metz, S.; Kästner, J.; Sokol, A. A.; Keal, T. W.; Sherwood, P. ChemShell—a modular software package for QM/MM simulations. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2014**, *4*, 101−110.
(49) Sherwood, P.; de Vries, A. H.; Guest, M. F.; Schreckenbach, G.; Catlow, C. A.; French, S. A.; Sokol, A. A.; Bromley, S. T.; Thiel, W.; Turner, A. J.; Billeter, S.; Terstegen, F.; Thiel, S.; Kendrick, J.; Rogers, S. C.; Casci, J.; Watson, M.; King, F.; Karlsen, E.; Sjøvoll, M.; Fahmi, A.; Schäfer, A.; Lennartz, C. QUASI: A general purpose implementation of the QM/MM approach and its application to problems in catalysis. *J. Mol. Struct.: THEOCHEM* **2003**, *632*, 1−28.
(50) Grimme, S.; Bannwarth, C.; Shushkov, P. A Robust and Accurate Tight-Binding Quantum Chemical Method for Structures, Vibrational Frequencies, and Noncovalent Interactions of Large Molecular Systems Parametrized for All spd-Block Elements (Z = 186). *J. Chem. Theory Comput.* **2017**, *13*, 1989−2009.
(51) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-xTB−An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15*, 1652−1671.
(52) *Ethanol Datasets*. https://github.com/zaverkin/ethanol_datasets_git (accessed July 2020).
(53) Linstrom, P. J., Mallard, W. G., Eds. *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*; National Institute of Standards and Technology: Gaithersburg, MD, 2016.

https://dx.doi.org/10.1021/acs.jctc.0c00347
J. Chem. Theory Comput. 2020, 16, 5410−5421

Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design

Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035009 https://doi.org/10.1088/2632-2153/abe294

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
9 September 2020

REVISED
11 January 2021

ACCEPTED FOR PUBLICATION
2 February 2021

PUBLISHED
12 May 2021

PAPER

Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design

Viktor Zaverkin and Johannes Kästner

Institute for Theoretical Chemistry, University of Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany

E-mail: kaestner@theochem.uni-stuttgart.de

Keywords: molecular machine learning, atomistic neural networks, active learning, optimal experimental design, computational chemistry

Abstract

Machine learning has been proven to have the potential to bridge the gap between the accuracy of *ab initio* methods and the efficiency of empirical force fields. Neural networks are one of the most frequently used approaches to construct high-dimensional potential energy surfaces. Unfortunately, they lack an inherent uncertainty estimation which is necessary for efficient and automated sampling through the chemical and conformational space to find extrapolative configurations. The identification of the latter is needed for the construction of transferable and uniformly accurate potential energy surfaces. In this paper, we propose an active learning approach that uses the estimated model's output variance derived in the framework of the optimal experimental design. This method has several advantages compared to the established active learning approaches, e.g. Query-by-Committee, Monte Carlo dropout, feature and latent distances, in terms of the predictive power and computational efficiency. We have shown that the application of the proposed active learning scheme leads to transferable and uniformly accurate potential energy surfaces constructed using only a small fraction of data points. Additionally, it is possible to define a natural threshold value for the proposed uncertainty metric which offers the possibility to generate highly informative training data on-the-fly.

1. Introduction

Quantum chemistry (QC) aims to describe the physical and chemical properties of atomistic systems using quantum mechanics. Computational chemistry (CC) uses QC approaches to obtain potential energy surfaces (PESs). Most other physical and chemical properties can be derived from the latter. Application of QC methods to even moderately large atomistic systems is computationally very expensive and, therefore, the development of empirical force fields (FFs) became the cornerstone of the modern CC [1–3]. Empirical FFs are highly efficient but suffer from limited transferability [4] and are generally not able to describe bond breaking and bond formation. Thus, there is a great demand for efficient and accurate PES models.

The recent development of machine learning (ML) methods changes the way of modeling molecular and material systems [5]. Being able to learn efficiently complex and highly non-linear functional relationships ML methods give the promise to bridge the gap between the computational efficiency of FFs and the accuracy of QC. In this paper, we use the ML approach recently developed in our group, which is referred to as Gaussian moment neural networks (GM-NNs) [6].

Employing ML algorithms, it is possible now to parametrize PESs using *ab initio* data to obtain models that can predict energies, atomic forces and Hessians with the *ab initio* accuracy and efficiency of FFs. An important issue appears; trained on QC data there is no guarantee that the parametrized model will properly predict properties of configurations far from the training data set. The generation of appropriate training data appears to be an especially challenging task if one takes into account the dimensionality of the chemical and conformational spaces [7]. Additionally, data sets built based on human intuition tend to be clustered,

sparse, and incomplete. They contain thousands to millions of data points each of them required the calculation of *ab initio* energies and forces. The latter can prohibit the application of ML methods due to the high computational cost.

This problem can be resolved by allowing the ML models to detect the most informative structures and perform the *ab initio* calculations only for them. This can be done on-the-fly, selecting extrapolative structures when running, e.g. the molecular dynamics (MD) simulation, or offline on the fixed data sets improving the generalization and the transferability of the potential. Both possibilities are related to active learning [8], an area of supervised learning whose aim is to learn general-purpose models with a minimal number of training data. The key quantity needed to perform active learning is the query strategy, i.e. an algorithmic criterion for deciding whether a given configuration has to be included in the training set.

A general overview of AL approaches can be found in [8]. In the context of interatomic potentials, a very natural query strategy can be defined for Gaussian process (GP) models using their inherent Bayesian predictive variance. Recently, this approach was successfully applied to model PESs of single- and multi-element systems on-the-fly [9] as well as to construct reactive PESs for H_3 and two prototypical reactive systems [10]. The on-the-fly training of machine-learned force fields was first proposed in [11], while the model error was evaluated employing *ab initio* calculations due to the poor correlation between the internal error of their GP model and the true model error [12]. Besides uncertainty-driven AL algorithms, genetic algorithms were applied for the optimization of training data sets [13] as well as a method based on selecting small building blocks, AMONs, from a dictionary to generate training instances on-the-fly has been recently proposed [14].

In this paper, we focus on methods that can be applied to neural networks. Query-by-Committee (QBC) is one of the most frequently used AL approaches in the literature [15–18]. It estimates the uncertainty of NNs using an ensemble of NN models. While widely employed in the chemistry community, training an ensemble of models increases the computational effort to the number of models used. Another approach to obtain the uncertainty of NNs is the Monte Carlo dropout approach [19, 20]. The cost is reduced to running the model multiple times rather than of the training of an ensemble. Finally, the uncertainty metric can be constructed by measuring the distances in the feature [18, 21, 22] and the latent spaces [20]. This can be prohibited due to the size of the system, the dimension of the feature space, and the size of the NN.

In this work, we propose another AL approach for atomistic NNs which uses the expected change in the model's output variance obtained in the framework of optimal experimental design (OED) [23–25]. To the best of our knowledge, a different OED framework was used previously for the linear regression problems [26, 27], and no application to atomistic NNs was proposed.

In the proposed AL scheme, the model's output variance is calculated using the Fisher information matrix computed using only the weights of the output layer. This can be done since the successive layers act to filter the redundant information present in the previous layers increasing the informativeness of the parameters of the last layer. The proposed AL approach can be applied to select new query points according to the model's output variance in both energies and forces.

The advantages of this approach are that (a) it introduces no overhead into model training or evaluation, (b) it can be applied to both simple and complex NN architectures that have been used for chemical property prediction, (c) it naturally ignores the redundant information present in the input layer and selects new query points based only on the model parameters, (d) it is possible to define an efficient algorithm which can query a large amount of data within few minutes.

The paper has the following structure: first, we shortly introduce the GM-NN model, derive the estimated output variance of NNs, and propose three different query strategies for atomistic NNs. Then, in section 3, we apply our active learning method to the ethanol [6], the QM9 [28, 29], and the N-ASW [7] data sets and compare the results to those obtained by the random selection strategy. We show that the expected change in the estimated output variance is correlated with the generalization error, and discuss the possibility of applying the proposed approach on-the-fly. The concluding remarks are given in section 4.

2. Method

In this work, we consider the problem of learning an input-output mapping $\mathcal{X} \to \mathcal{Y}$ from a set of N_{train} training samples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N_{\text{train}}}$, with $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ [23]. In the case of molecular machine learning x_i are typically the atomic coordinates, i.e. $\mathcal{X} \subset \mathbb{R}^{N_{\text{at}} \times 3}$ with N_{at} being the number of atoms, and y_i are molecular physicochemical properties. Here we consider y_i being the scalar total energy of the system, i.e. $\mathcal{Y} \subset \mathbb{R}$, or its atomic forces, i.e. $\mathcal{Y} \subset \mathbb{R}^{N_{\text{at}} \times 3}$.

We denote a general parametrized learner as $f(\mathbf{w}, \cdot)$. Its output can be written as $\hat{y}_i = f(\mathbf{w}, x_i)$. The learner is trained by adjusting parameters $\mathbf{w}$ so that the mean squared loss,

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(\hat{y}_i - y_i\right)^2, \tag{1}$$

is minimized. Converged training procedure results in the best set of parameters, $\hat{\mathbf{w}}$, used subsequently for predictions on the test data or during a real-time simulation.

For this work, we use the GM-NN [6] as the parametrized learner with corresponding weight matrices and bias vectors $\mathbf{w} = \{\boldsymbol{W}_k, \boldsymbol{b}_k\}$. A brief overview of the architecture, molecular descriptor, and training procedure of the machine learning (ML) model is given in section 2.1. For more details about the employed model, see elsewhere [6].

The main focus of this work is the setting in which it is allowed to the parametrized learner, $f(\hat{\mathbf{w}}, \cdot)$, to select a new training input x^* from a set of candidate inputs, which we call the pool:

$$\mathcal{P} = \{x_i\}_{i=1}^{N_{\text{pool}}} \subset \mathbb{R}^{N_{\text{at}} \times 3}, \tag{2}$$

with N_{pool} unlabeled instances. Labeling, here the calculation of *ab initio* energies and forces, is performed only on the selected instances since this process is assumed to be computationally expensive.

Given the above conditions, the main issue remains the selection of new training samples without labeling, which would minimize the generalization error of the model. For this purpose, we propose an active learning scheme that selects new training instances according to the expected change in the estimated output variance of the learner. The latter is derived by employing techniques from the field of optimal experimental design (OED) [25]. This work is based on the applications of OED to feed-forward NNs dated from the end of the 20th century [23, 24] and can be referred to as variance reduction query strategy [8].

Section 2.2 briefly reviews the derivation of the expected change in the estimated variance of NNs when adding a new training instance. Section 2.3 presents query strategies used for the active learning of atomistic NNs.

2.1. GM-NN model

As was mentioned before, we have selected the GM-NN approach [6], which uses feed-forward NNs to represent the high-dimensional potential energy surface (PES), as the parametrized learner. In this approach, a single potential energy E of a molecular or solid-state structure is written as a sum of 'atomic' energy contributions:

$$\hat{E} = \sum_i \hat{E}_i \left(\{\mathbf{R}_{ij}, Z_i, Z_j\}_{j \neq i}\right). \tag{3}$$

These $\hat{E}_i$ depend on the local environment of the atom i within a predefined cutoff sphere of the radius R_c. In the above equation index j runs only over all neighbors within R_c. The choice of R_c depends strongly on the studied system and, therefore, will be specified separately for each data set in section 3.

The description of the local environment in the GM-NN model is given through a set of novel symmetry-preserving local atomic descriptors, the Gaussian moments (GM). In addition to the geometric information, GMs include information about the atomic species of both the central and neighbor atoms. Therefore, for all 'atomic' energy contributions, only a single NN has to be trained, in contrast to using an individual NN for each species as frequently required in the literature. The computational cost and memory usage of the GM-NN model scale linearly with the system size because atomic neighbor lists are employed. Throughout this work, we use a shallow neural network with two hidden layers consisting of 256 and 128 nodes each (abbreviated GM-sNN in [6]).

To train the GM-sNN model the loss function:

$$\mathcal{L} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} w_{\text{E}} \left\|\Delta E_i\right\|^2 + \frac{w_{\text{F}}}{3N_{\text{at}}} \sum_{j=1}^{N_{\text{at}}} \left\|\Delta \mathbf{F}_{ij}\right\|^2, \tag{4}$$

is minimized. Here, N_{at} is the number of atoms in the respective structure, ΔE_i and $\Delta \mathbf{F}_{ij}$ are the differences between the GM-sNN prediction and the reference data for energies and forces, respectively. The parameters w_{E} and w_{F} were set to 1 au and 100 au Å^2, respectively. Every GM-sNN model was trained using the AMSGrad optimizer [30] with 32 molecules per mini-batch. The learning rate was set to 10^{-3} and kept constant throughout the whole training procedure. All models used in section 3 were implemented in the Tensorflow framework [31] and were trained for 5000 training epochs on an NVIDIA Tesla V100-SXM2-32GB GPU.

2.2. Variance estimation for NNs

The purpose of this section is to derive an estimator for the output variance of NNs and the respective change in the variance when a new data point is added to the training set. The latter can be used to indirectly minimize the generalization error of NNs. This holds since the learner's expected future error can be decomposed into three terms [32]: (a) the *noise* of the data introduced by, e.g. the *ab initio* method, which is independent of the model and the training set; (b) the *bias* of the model, i.e. the error introduced by the model class itself; (c) the *variance* of the model. Thus, minimizing the variance of the model is guaranteed to minimize the future generalization error of the model [8].

Following derivations in references [23, 24] the estimated output variance of the NN at the training point x_i can be written as

$$\sigma^2_{\hat{y}}(x_i) \approx \mathcal{L}\left(\frac{\partial \hat{y}_i}{\partial \mathbf{w}}\right)^T \left(\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2}\right)^{-1} \left(\frac{\partial \hat{y}_i}{\partial \mathbf{w}}\right), \tag{5}$$

where $\mathcal{L}$ is the mean squared loss of the NN given in equation (1) or equation (4), the network sensitivity is defined by $\mathbf{g}(x_i) = \partial \hat{y}_i / \partial \mathbf{w}$, and the Fisher information matrix, $\mathbf{A}$, is defined as

$$\mathbf{A} = \frac{1}{\mathcal{L}} \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^2}. \tag{6}$$

The inverse of the Fischer information matrix can be referred to as the parameter covariance matrix.

By using the chain rule one can easily obtain the Fisher information matrix element:

$$A_{ab} = \frac{2}{\mathcal{L} N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \frac{\partial \hat{y}_i}{\partial w_a} \frac{\partial \hat{y}_i}{\partial w_b} + (\hat{y}_i - y_i) \frac{\partial^2 \hat{y}_i}{\partial w_a \partial w_b}, \tag{7}$$

where the prefactor can be seen as the noise level. For the final expression given in equation (14) the noise parameter results in a multiplicative prefactor and, therefore, does not have any impact on the method performance. To be consistent with [23] we use the training residual $\mathcal{L}$ as a noise estimate. Assuming that the trained model is already close to the optimal minimum, i.e. the prediction for x_i is fairly good, the Fisher information matrix can be approximated as

$$\mathbf{A} \approx \frac{1}{\mathcal{L}} \sum_{i=1}^{N_{\text{train}}} \mathbf{g}(x_i) \mathbf{g}^T(x_i). \tag{8}$$

We have mentioned before that the generalization error of the model is correlated with its output variance. The variance reduction query becomes

$$x^* = \underset{x^* \in \mathcal{P}}{\operatorname{argmin}} \langle \sigma^2_{\hat{y}}(x_i, x^*) \rangle_{x_i \in \mathcal{D}}, \tag{9}$$

where the expression $\langle \sigma^2_{\hat{y}}(x_i, x^*) \rangle_{x_i \in \mathcal{D}}$ is the estimated mean output variance across the input distribution, i.e. the learner's output variance averaged over the training samples after the model has been retrained on the instance x^* and its corresponding label. Note that in the following we write instead of $\langle \cdot \rangle_{x_i \in \mathcal{D}}$ only $\langle \cdot \rangle_{\mathcal{D}}$ to make the notation somewhat shorter.

Different approaches can be used to find an optimal instance x^* without re-training the model. For example, in references [26, 27] the so-called D-optimality approach was employed. In this work, we follow the approach proposed in [23].

After adding a new training instance x^* the Fisher information matrix can be approximated as

$$\mathbf{A}^* \approx \mathbf{A} + \frac{1}{\mathcal{L}} \mathbf{g}(x^*) \mathbf{g}^T(x^*). \tag{10}$$

Its inverse can be easily calculated by the Woodbury matrix identity:

$$(\mathbf{A}^*)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1} \mathbf{g}(x^*) \mathbf{g}^T(x^*) \mathbf{A}^{-1}}{\mathcal{L} + \mathbf{g}^T(x^*) \mathbf{A}^{-1} \mathbf{g}(x^*)}. \tag{11}$$

The output variance of the model, after adding the data point x^*, can be estimated at the reference point x_i without re-training the model as

$$\begin{aligned} \sigma^2_{\hat{y}}(x_i, x^*) &= \mathbf{g}^T(x_i) \left(\mathbf{A}^*\right)^{-1} \mathbf{g}(x_i) \\ &= \mathbf{g}^T(x_i) \mathbf{A}^{-1} \mathbf{g}(x_i) - \frac{\left[\mathbf{g}^T(x_i) \mathbf{A}^{-1} \mathbf{g}(x^*)\right]^2}{\mathcal{L} + \mathbf{g}^T(x^*) \mathbf{A}^{-1} \mathbf{g}(x^*)}, \end{aligned} \tag{12}$$

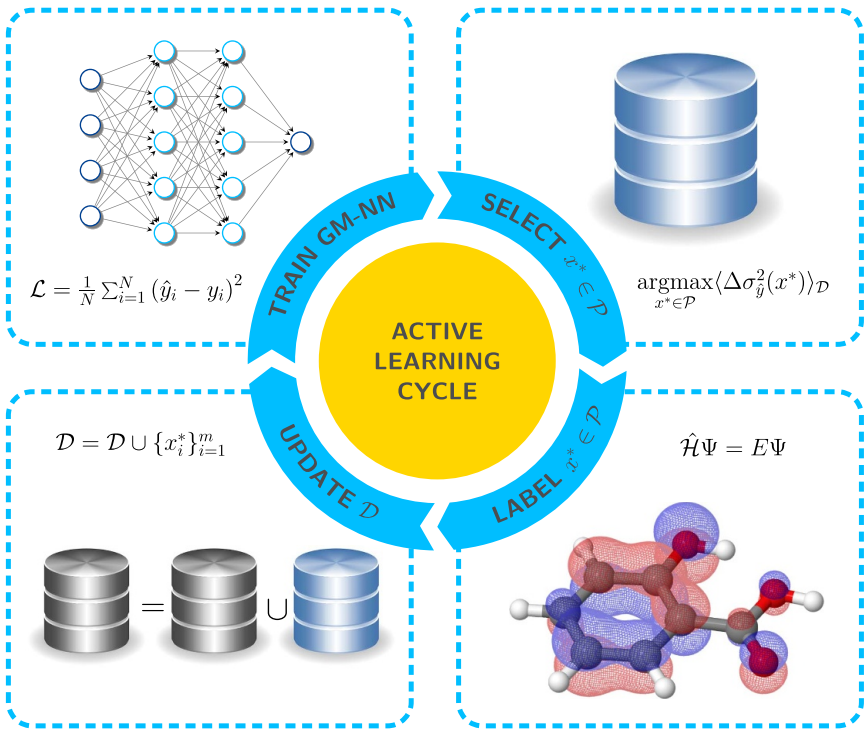


Figure 1. Schematic representation of the active learning cycle.

where the first term, $\mathbf{g}^T(x_i)\mathbf{A}^{-1}\mathbf{g}(x_i) = \sigma^2_{\hat{y}}(x_i)$, is the original output variance of the model. The second term is the expected change in the model's output variance after querying a new data point x^*. Since our interest is the average variance change one has to calculate the average for each instance in the pool. This is inefficient if the pool contains a large number of data. To make this step computationally efficient one can approximate the respective expression by

$$\langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}} \approx \frac{\mathbf{g}^T(x^*)\mathbf{A}^{-1}\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i) \rangle_{\mathcal{D}}\mathbf{A}^{-1}\mathbf{g}(x^*)}{\mathcal{L} + \mathbf{g}^T(x^*)\mathbf{A}^{-1}\mathbf{g}(x^*)}. \tag{13}$$

In the above expression, one calculates the average over training samples once when calculating $\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i) \rangle_{\mathcal{D}}$ and can re-use it for all instances in the pool. Given this expression, one solves the problem defined in equation (9), i.e. one can define an instance $x^* \in \mathcal{P}$ which minimizes the model's output variance without either labeling the data or re-training the model. We want to remind the reader that labeling, calculation of *ab initio* energies and atomic forces, is assumed to be computationally expensive.

The expression in equation (13) can be simplified even further using the definition of the Fisher information matrix $\mathbf{A}$ and of the average $\langle \mathbf{g}(x_i)\mathbf{g}^T(x_i) \rangle_{\mathcal{D}}$ and reads

$$\langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}} = \frac{\mathcal{L}}{N_{\text{train}}} \cdot \frac{\mathbf{g}^T(x^*)\tilde{\mathbf{A}}^{-1}\mathbf{g}(x^*)}{1 + \mathbf{g}^T(x^*)\tilde{\mathbf{A}}^{-1}\mathbf{g}(x^*)}, \tag{14}$$

where $\tilde{\mathbf{A}} = \mathcal{L}\mathbf{A}$. Note that we neglect the prefactor $\mathcal{L}/N_{\text{train}}$ in the following discussion since it is independent of the new data point x^* and only rescales the expected change in the model's output variance.

2.3. Active learning: atomistic neural networks

An active learning scheme has to be able to select the most informative instances from the unlabeled pool of data. Here, we make use of equation (14) derived in section 2.2 and of the fact that the model's output variance is correlated with the generalization error.

The active learning scenario proposed in this work is schematically shown in figure 1. In the first step, the ML model is initialized, i.e. it is trained on the initial, randomly selected, training data set of size N_{train}. Next, using the expression in equation (14), m structures are selected from the pool. The respective structures are selected such that the expected change in the output variance of the model is maximal:

$$x^* = \underset{x^* \in \mathcal{P}}{\operatorname{argmax}} \langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}}. \tag{15}$$

The above expression correlates with equation (9). Larger values of the expected change in the output variance imply reduction of the output variance itself, see equation (12). Note that due to the possible correlations of the data in the pool, the active learning algorithm selects a maximum of $m = 0.1 \cdot N_{\text{train}}$ new samples per iteration. The size of the training data set increases after each active learning iteration by m samples, respectively. Finally, after the algorithm has selected m query instances, the respective labels are calculated using, e.g. *ab initio* quantum chemistry (QC) and the training set is updated by these query samples. In this work, we re-train the GM-NN model on the updated training set using re-initialized weights and biases to study the performance of the active learning scheme more thoroughly. However, we should mention that one can start from the pre-trained parameters from the previous iteration which would speed-up the re-training.

The active learning continues until either the maximal size of the training set is reached or the queried instances are not sufficiently informative anymore. We use only the first criterion in section 3, but discuss ways to define the second criterion. Note that the latter is important to perform the so-called learning on-the-fly, where the active learning algorithm queries structures obtained during the simulation. In this work, we use only data sets that are already labeled to test the proposed approach in terms of its applicability to the sampling of configurational and chemical spaces. The maximal size of the training data set depends on the data set. Therefore, the upper limit is defined separately for each data set in section 3.

Before introducing the possible query strategies for atomistic NNs we want to briefly discuss an issue caused by the size of the parameter space of NNs. In general, the parameter space of atomistic NNs is large, which makes the proposed approach, in the first view, intractable on typical computers. For example, for the shallow GM-sNN model with only two hidden layers and 427 invariant molecular descriptors, one obtains more than 142 000 weight parameters. Note that we take weights into account for active learning but no biases because their influence is expected to be negligible. Therefore, one needs to make additional assumptions to make the presented approach applicable to the atomistic NNs.

To tackle the size problem we assume that the weight parameters of the output layer contribute most to the estimation of the model's output variance. One can argue that the preceding layers of the NN contain some amount of redundant information that is filtered when passing through the network. This makes the input of the output layer and, thus, the respective weights more sensitive to the relevant changes in the queried structures. A similar assumption was made for the latent distances in [20]. The performed experiments prove the above premise, see section 3. Using only the weight parameters of the output layer, the Fisher information matrix is only 128×128-dimensional and can be easily inverted.

Now we want to focus on particular query strategies that can be used to select the most informative structures for atomistic NNs. In the following, three different possibilities are presented, based on the energy, force, and the total squared loss.

2.3.1. Query strategy QS_1: energy squared loss

The first query strategy, labeled as QS_1, uses only the energy squared loss:

$$\mathcal{L}_{\hat{E}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left\| E_i - \hat{E}_i \right\|^2. \tag{16}$$

Given the energy loss, one can easily define the network sensitivity, $\mathbf{g}_{\hat{E}}(x_i) = \partial \hat{E}_i / \partial \mathbf{w}$, and the corresponding Fisher information matrix reads

$$\mathbf{A}_{\hat{E}} = \frac{1}{\mathcal{L}_{\hat{E}}} \sum_{i=1}^{N_{\text{train}}} \mathbf{g}_{\hat{E}}(x_i)\, \mathbf{g}_{\hat{E}}(x_i)^T. \tag{17}$$

Configurations are selected from the pool of unlabeled structures using equations (14) and (15) with the respective network sensitivity, Fisher information matrix, and squared loss.

2.3.2. Query strategy QS_2: force squared loss

The second query strategy, labeled as QS_2, uses only the force squared loss:

$$\mathcal{L}_{\hat{F}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \frac{1}{3N_{\text{at}}} \sum_{j=1}^{N_{\text{at}}} \sum_{k=1}^{3} \left\| F_{ijk} - \hat{F}_{ijk} \right\|^2. \tag{18}$$

In this case, the network sensitivity is calculated as the gradient of the atomic force element with respect to the model parameters:

$$\mathbf{g}_{\hat{F}_{ijk}}(x_i) = \frac{\partial \hat{F}_{ijk}}{\partial \mathbf{w}}. \tag{19}$$

This implies that one obtains a tensor of rank 3 for the whole molecular structure instead of a vector as in QS_1. This makes the query strategy QS_2 less efficient. However, it can be advantageous if the most informative local atomic environments have to be found.

Using the corresponding network sensitivity and the force squared loss one can write an expression for the Fisher information matrix:

$$\mathbf{A}_{\hat{F}} = \frac{1}{\mathcal{L}_{\hat{F}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N_{\text{at}}} \sum_{k=1}^{3} \mathbf{g}_{\hat{F}_{jk}}(x_i)\, \mathbf{g}_{\hat{F}_{jk}}(x_i)^T. \tag{20}$$

Similar to QS_1, configurations are selected from the pool of unlabeled structures using equations (14) and (15) with the respective network sensitivity, Fisher information matrix, and squared loss.

In contrary to QS_1 one obtains a matrix, $(\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}$, of size $N_{\text{at}} \times 3$ for each structure x^* in the pool. We propose to use the mean of that matrix as the final value employed to select the most informative structures, i.e.

$$\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}} = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} (\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}. \tag{21}$$

Alternatively, one could use the maximal value of the matrix, $(\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}})_{ij}$, selecting structures according to the most informative local environments. For the data sets used in section 3, we have found that using the mean gives the best correlation between the calculated metric, $\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle_{\mathcal{D}}$, and the actual absolute error in predicted force. Therefore, we employed only this approach in section 3.

2.3.3. Query strategy QS_3: total squared loss

It is also possible to calculate the expected change in the estimated output variance of the model using the total loss function presented in equation (4). The uncertainty of the model evaluated for an instance $x^* \in \mathcal{P}$ can be written as a weighted sum of results obtained in sections 2.3.1 and 2.3.2:

$$\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{total}}_{\mathcal{D}} = \langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{energy}}_{\mathcal{D}} + \beta \langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle^{\text{force}}_{\mathcal{D}}, \tag{22}$$

where $\langle \Delta\sigma^2_{\hat{y}}(x^*)\rangle^{\text{energy}}_{\mathcal{D}}$ is the expected change in the model's output variance obtained using the energy loss and $\langle \Delta\bar{\sigma}^2_{\hat{y}}(x^*)\rangle^{\text{force}}_{\mathcal{D}}$ is the uncertainty obtained from the force loss. The force contribution is scaled by a factor β, similar to equation (4). Note that in section 3.1 and section 3.3 we have found that this approach is of no practical use for the respective systems. This is due to the high correlation between QS_1 and QS_2 metric, high computational cost of the latter, and only minor improvement in terms of the correlation with the actual error.

3. Results

In this section, we test the proposed active learning (AL) scheme on three different molecular data sets. In section 3.1, we confirm the applicability of our approach to atomistic NNs in practice using the ethanol molecular dynamics (MD) data set [6], which samples the configurational space. In section 3.2, we simulate the chemical space sampling employing the well-established QM9 data set [28, 29]. Lastly, in section 3.3, the OED uncertainty metric is used to construct a transferable and uniformly accurate NN potential using the N-ASW data set [7]. Additionally, we discuss the possibility of using the proposed approach on-the-fly.

All experiments described in this section were run using the GM-NN approach [6], which uses feed-forward NN with two hidden layers containing [256, 128] nodes each. For representing molecular structures the selected approach uses 427 rotationally invariant scalars referred to as GMs. The only hyper-parameter needed to be defined is the cutoff radius. It is set up for each experiment separately.

3.1. Ethanol: molecular dynamics data

We start by applying the proposed AL approach to the ethanol data set [6] to confirm its general applicability to molecular systems. The ethanol data set contains Cartesian coordinates, total energies, and atomic forces of 5000 conformations obtained from an *ab initio* MD at 1000 K. Energies and atomic forces were calculated at the PBE-D3(BJ)/6-31G* [33–36] level of theory. In this section, we set the cutoff radius to 4 Å, i.e. the whole molecule is within the cutoff sphere.

Before discussing the obtained results we want to define all hyper-parameters of AL scheme used in this section. Each AL cycle is initialized drawing randomly 100 and 200 structures from the data set. The GM-sNN is trained on 100 structures, the other 200 structures were used for early stopping [37]. Then, the parameters of the trained model are used to calculate the OED metric defined in equation (14) for all conformations in the pool. Note that the pool comprises the structures remaining after the selection of the training samples and includes structures used for validation. $0.1 \cdot N_{\text{train}}$ structures with the maximal OED metric are selected and added to the training set. Finally, 200 new conformations are drawn randomly from the pool for validation and the model is re-trained. In total, we performed 32 active learning iterations including the initialization, which results in a maximal training set size of 1863. All structures which were not employed during training (e.g. for the last AL iteration 2937 structures) were used to test the performance of the model.

It must be emphasized that the gains predicted in the OED framework are expected gains and depend on several approximations discussed in section 2.2. Therefore, we want to confirm that the OED uncertainty metric correlates with the absolute error in energies and atomic forces. Figure 2 shows the correlation of the QS_1 (top left) and QS_2 (top right) metric with the actual force errors. Each dot in figure 2 corresponds to one structure in the pool. Note that forces alone determine the dynamics of a chemical system. Therefore, we want to confirm that the AL approach can select those structures which would improve force prediction on extrapolative conformations. All metrics presented in figure 2 are calculated according to equation (14), if not stated otherwise, and normalized to $[0, 1]$ for comparison.

From figure 2 one can notice that the correlation between the actual error and the estimated uncertainty of the model is not ideal. The imperfect correlation between uncertainties and actual errors can originate from inductive biases of the model and from the fact that a large uncertainty does not necessarily imply a large error. To estimate the influence of the latter contribution, we compute the correlation between the uncertainty and random errors sampled from the posterior predictive distribution as

$$\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right| , \tag{23}$$

with $\mathbf{F_i} \sim \mathcal{N}\left(\hat{\mathbf{F}}_i, \langle \Delta\sigma^2_{\hat{y}}(x^*) \rangle_{\mathcal{D}}\right)$. Using this method, we find a correlation of 0.86 for QS_2 if the posterior predictive distribution was accurate. The fact that the real correlation is lower can be attributed to the inductive biases of the ML method.

Figure 2 shows a slightly better correlation of the QS_2 metric with the actual error in the forces. The correlation coefficients obtained for QS_1 and QS_2 are 0.53 and 0.60, respectively. Note that for the QS_1 uncertainty we used only the numerator of equation (14) for the plot since we have found that the whole expression shows a non-linear relationship with the actual error. This can be done as long as the expression in equation (14) and its numerator are monotonically increasing functions, which is the case for the current study. Regarding the computational cost, the QS_2 metric needs 80 times more CPU time (6 s for the QS_1 metric on a single Intel Xeon CPU E5-2640 4) and 20 times more memory (140 MB for the QS_1 metric) to be evaluated for all structures in the pool. Both, the CPU time and the memory usage, depend only slightly on the active learning iteration.

Additionally, we have studied the correlation between QS_1 and QS_2 uncertainties. In figure 2 (bottom left) one can see that the uncertainty metrics are strongly correlated with a correlation coefficient of 0.75. The combination of QS_1 and QS_2 uncertainties comprising the QS_3 query strategy, see section 2.3.3, have not shown any considerable improvement. We could reach only a correlation coefficient of 0.61 with $\beta = 1.5$, see figure 2 (bottom right). For that reason, in the following, we consider only two other query strategies described in section 2.3.1 and section 2.3.2.

Considering the efficiency of the QS_1 metric and the sufficient correlation with the actual force error it seems superior over the QS_2 and QS_3 metric. The QS_2 metric can directly provide information on the most informative local atomic environment. However, because the total energy is decomposed into 'atomic' energies, it is also possible to identify the most informative local environment by using the QS_1 metric.

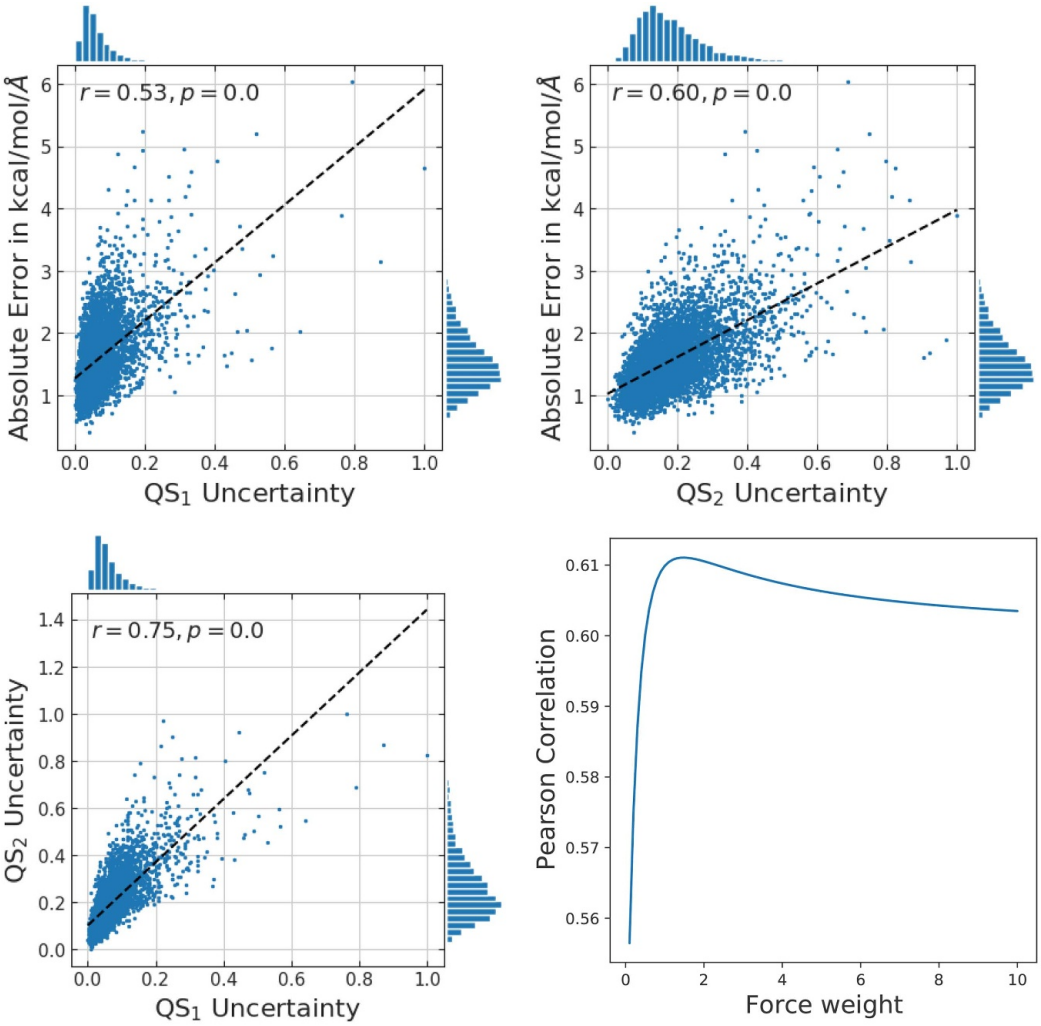


Figure 2. Correlation of the (top left) QS_1 and (top right) QS_2 metric with the absolute force error $\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$ on the ethanol data set. For both query strategies, the correlation coefficient is shown as an inset along with the best fit line. (Bottom left) Correlation between QS_1 and QS_2 uncertainty metric. The correlation coefficient is shown as an inset along with the best fit line. (Bottom right) The dependence of the correlation coefficient obtained for the QS_3 metric and the absolute force error on the force weight β in equation (22).

The success of the AL algorithms can be measured by comparing to randomly chosen training sets. Figure 3 shows three different error measures obtained for the force predictions of the GM-sNN model trained on actively and randomly selected data. The respective error measures are the mean absolute error (MAE), L_1, the root-mean-squared error (RMSE), L_2, and the maximal error (MAXE), L_∞. All results are averaged over three independent runs. While the MAE and RMSE of the model trained on the structures selected by the AL algorithm are improved only by factors of 1.15 to 1.23, the MAXE is reduced by factors larger than 2.0, compared to the results obtained with randomly selected training data. Strong improvement of the MAXE, i.e. the identification of extrapolative or unusual configurations, shows that our AL scheme leads to the generation of uniformly accurate machine-learned potentials. Note that only about 1150 structures were needed for a maximal error of around 1 kcal mol^{-1} Å^{-1}, the accuracy required for molecular simulations.

All models for the ethanol data set were trained on an NVIDIA Tesla V100-SXM2-32GB GPU. The training of 5000 epochs took from 13 min (100 structures) to about 2 h (1863 structures).

3.2. QM9

In this section, we use the QM9 data set [28, 29] to assess the performance of our AL scheme when sampling the chemical space. QM9 is a widely used benchmark for the prediction of several properties of molecules in

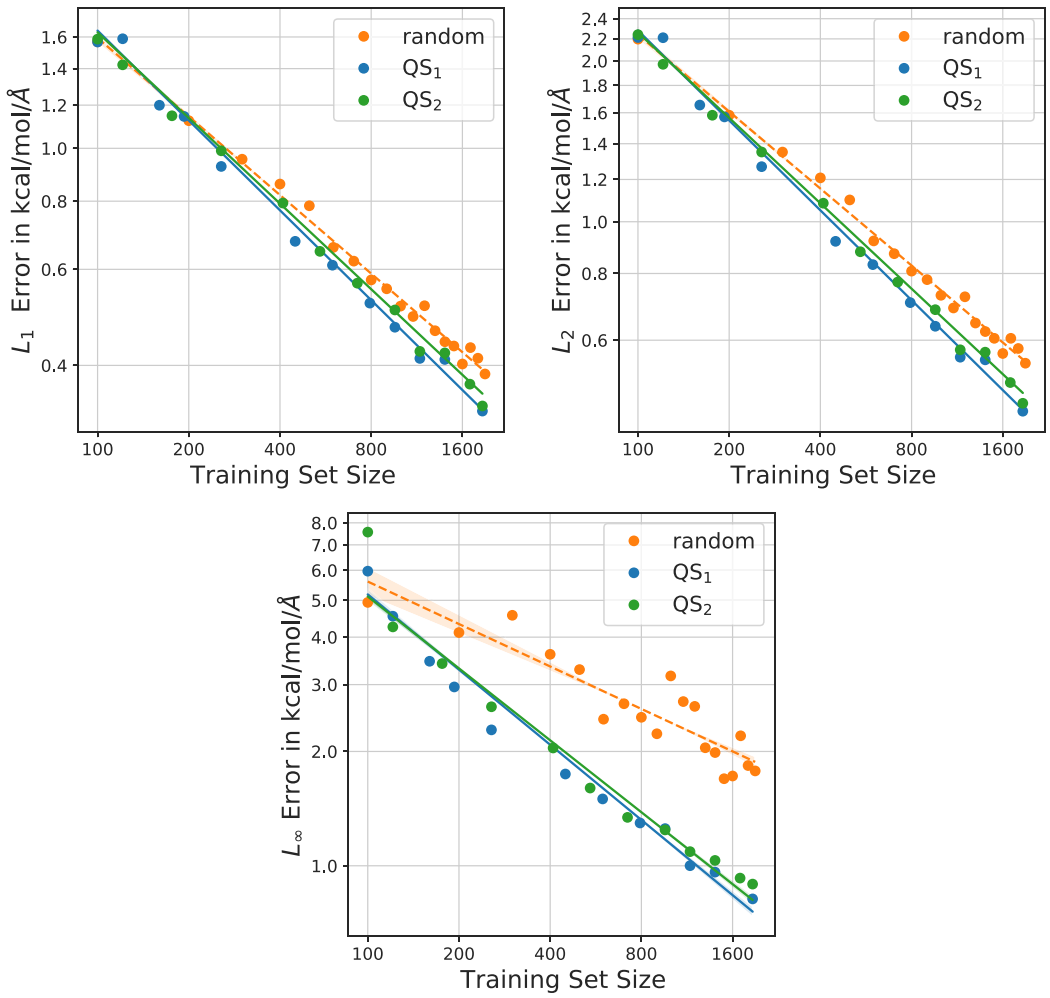


Figure 3. Log–log plot of (top left) the mean absolute, (top right) the root-mean-squared, and (bottom) the maximal error of the predicted forces on the ethanol data set depending on the size of the training set. Structures are selected randomly or using query strategies QS_1 and QS_2. For each learning curve a linear fit is plotted in addition.

equilibrium. Thus, all forces vanish and the model is trained using only the energy squared loss. For AL we use only the query strategy QS_1.

The QM9 data set consists of 133 885 neutral, closed-shell organic molecules with up to 9 heavy atoms (C, O, N, F) and a varying number of hydrogen (H) atoms. The largest structure in the data set contains 29 atoms in total. Since 3054 molecules from the original QM9 data set failed a consistency test [29], we used only the remaining 130 831 structures in the following experiments. Similar to the previous work we used a cutoff radius of 3.0 Å [6].

For initializing AL cycles we selected randomly 5000 samples to train the model and another 2000 structures to validate its performance during the training procedure. The parameters of the converged model were employed to select new training samples from the pool comprising 125 831 structures using equation (14). Note that the structures used for early stopping (validation set) are also added to the pool. In every iteration the AL algorithm selects $0.1 \cdot N_{\text{train}}$ new structures and adds them to the training set. The AL cycle was stopped when the training set size reached a value of 25 261, i.e. after 18 iterations including the initialization.

We want to make an additional remark on the computational cost of the QS_1 strategy. To select 500 samples out of 125 831 structures in the pool we needed about 135 s on the single Intel Xeon CPU E5-2640 4. The memory used is about 850 MB. Both values are almost independent of the AL step.

Similar to the previous section, figure 4 (top left) shows the correlation of the OED metric with the absolute energy error. We have found a correlation coefficient of 0.44 for the QS_1 uncertainty metric. Note that in the figure we used the square root of the expression in equation (14) to be consistent with the results

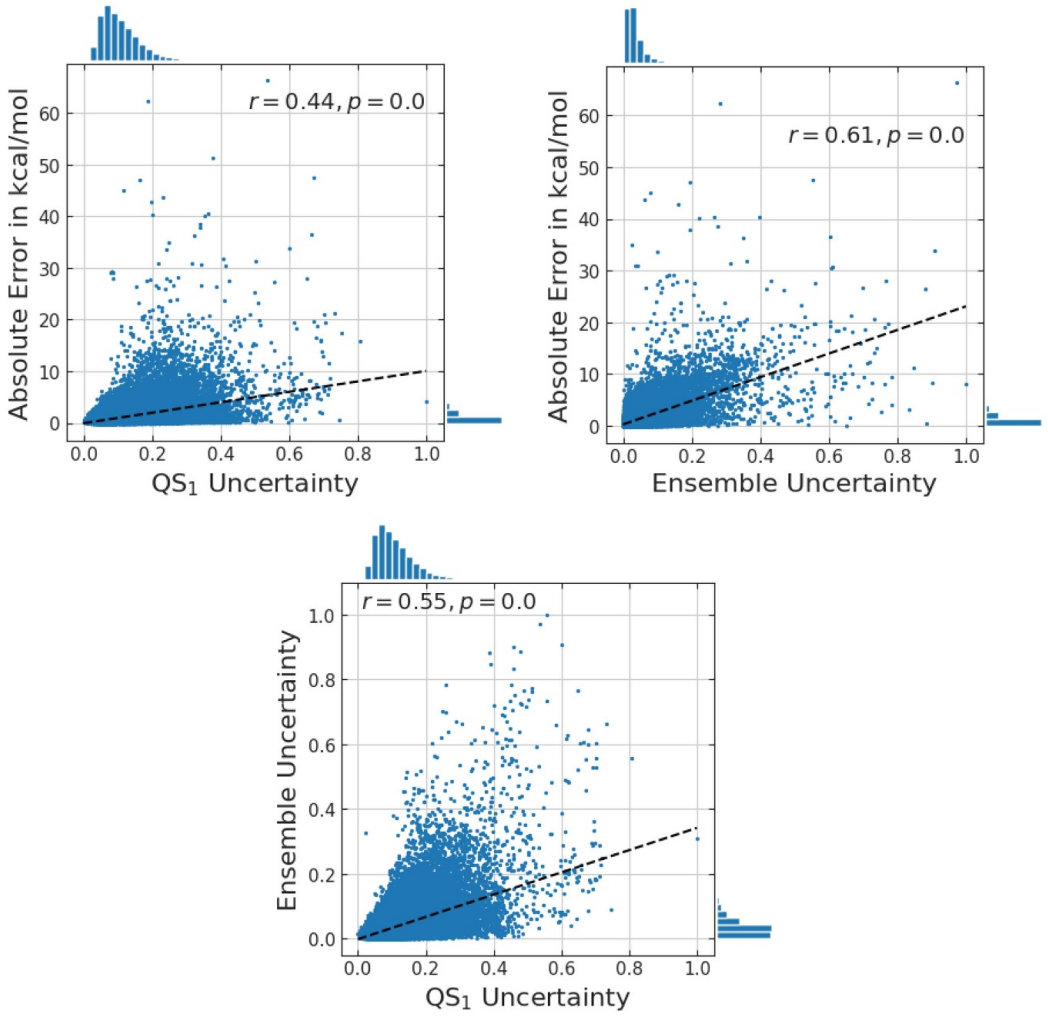


Figure 4. Correlation of (top left) the QS_1 metric and (top right) the ensemble variance with the absolute energy error, $\Delta E = \left|E - \hat{E}\right|$, for QM9 test data. (Bottom) Correlation between the variance estimated by the ensemble and the QS_1 uncertainty metric. The correlation coefficients are shown as insets along with the best fit lines.

obtained below for ensembling. Structures predicted with higher energy error obtain a higher uncertainty by the OED approach. To demonstrate that the performance of the OED approach is comparable to well-established AL techniques we trained the committee of three models. The uncertainty in the Query-by-Committee (QbC) approach can be measured as

$$\sigma_{\text{ens}}\left(x^*\right) = \sqrt{\frac{1}{N_{\text{ens}}}\sum_{i=1}^{N_{\text{ens}}}\left(\hat{y}_i\left(x^*\right) - \hat{\bar{y}}\left(x^*\right)\right)^2}, \tag{24}$$

where N_{ens} is the number of models in the committee, i.e. $N_{\text{ens}} = 3$. $\hat{\bar{y}}\left(x^*\right) = 1/N_{\text{ens}}\sum_{i=1}^{N_{\text{ens}}}\hat{y}_i\left(x^*\right)$ is the mean of the energy prediction over the committee.

Figure 4 (top right) shows that the correlation of the absolute energy error with the uncertainty metric obtained employing a committee is comparable to the correlation obtained using our method. The correlation coefficient has a value of 0.61. The difference is negligible when one takes into account that the OED approach does not need to train multiple models. In general, the proposed approach is N_{ens}-times more efficient than the QbC method.

For the sake of completeness, we studied the correlation between uncertainty estimates. Figure 4 (bottom) shows a strong correlation between them with a correlation coefficient of 0.55. For the comparison of the QbC approach with a few other approaches, e.g. Monte Carlo dropout [19, 20], feature space distances [18, 21, 22] and latent space distances [20], see elsewhere [20].

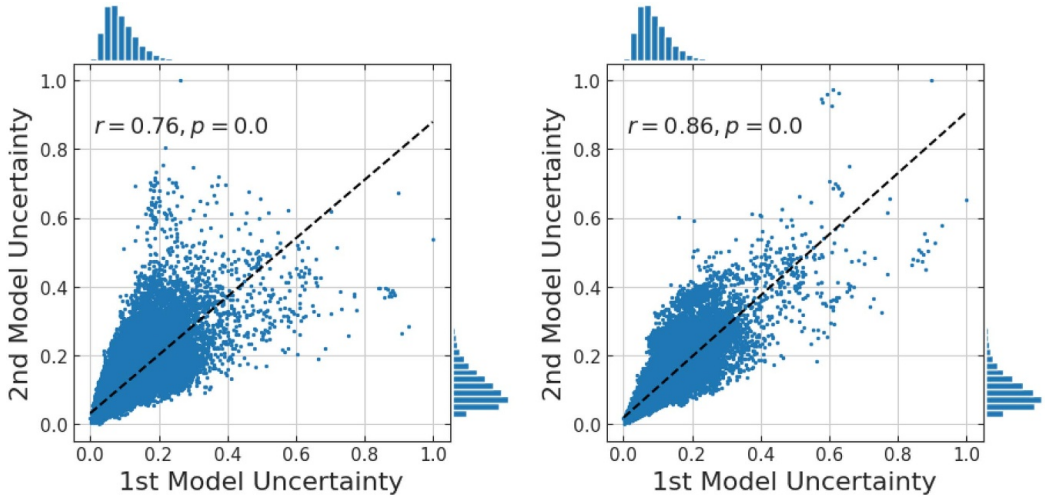


Figure 5. (Left) Correlation of the OED uncertainty metrics calculated using parameters of two independently initialized and trained GM-NN models. Both models have been trained to the same degree. (Right) Correlation of the OED uncertainty metrics calculated using the same model but trained to different degrees. The first model has been trained for 5000 epochs. The second model has been trained for 112 epochs.

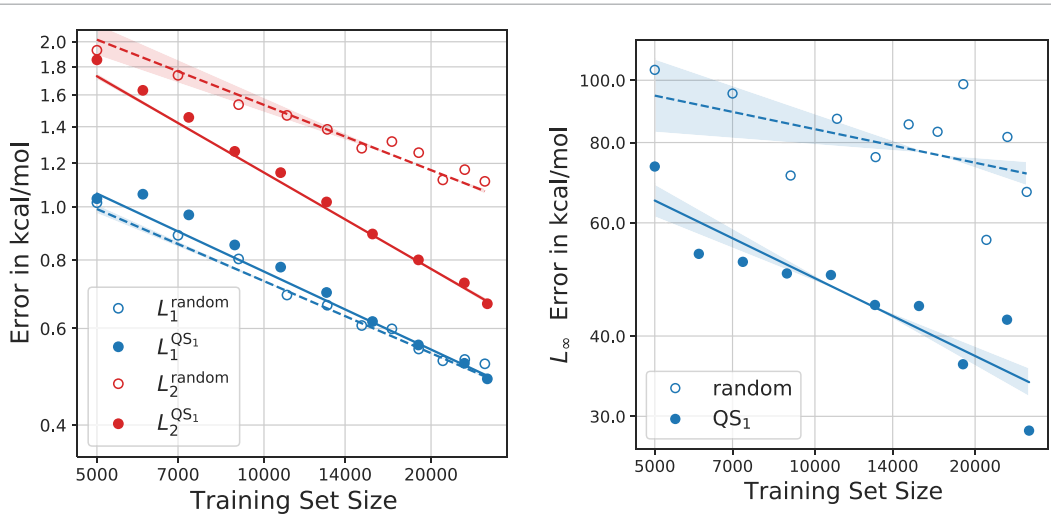


Figure 6. Log–log plot of (left) the mean absolute (L_1) and the root-mean-squared errors (L_2), and (right) the maximal error (L_∞) in the predicted energies on the QM9 data set. Structures are selected randomly or using query strategy QS_1. For each learning curve a linear fit is plotted in addition.

Besides the correlation of the OED uncertainty with the actual error, the correlation between uncertainties obtained for different local minima of the model has been studied. For that purpose, on the one hand, two models were trained using the same training data but independent randomly initialized NN parameters. Figure 5 (left) shows the correlation between uncertainties of two converged, i.e. trained to the same degree, models with a linear correlation coefficient of 0.76. On the other hand, the correlation between uncertainties of the same model but trained to a different degree is shown in figure 5 (right). In this case, we have found that the uncertainty obtained after training for 112 epochs correlates strongly with the uncertainty estimated for the converged model with a correlation coefficient of 0.86. These findings show that different local minima produce similar results. In particular, these results show that it is enough to train the model only shortly to obtain the desired uncertainty estimate, which improves the computational efficiency of the proposed approach considerably. Equivalent results can be obtained for other training data sets and, for the sake of brevity, will be left out in other sections. Given the deviation between the estimated NN uncertainties, an interesting question arises about the ensembling of them as well as about a combination with the QbC variance to achieve an even better correlation with the actual error. Unfortunately, this is out of the scope of this paper and will be studied in our future works.

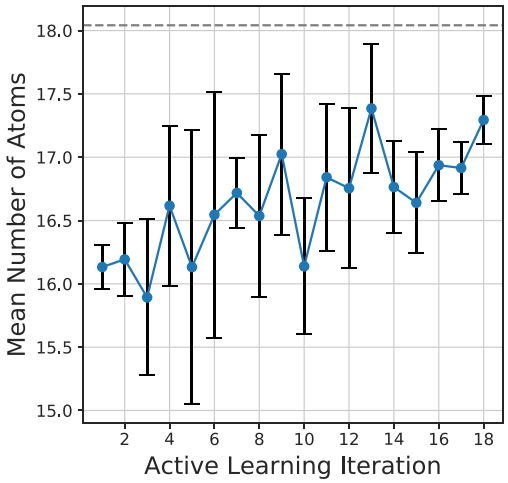


Figure 7. The mean number of atoms in molecules selected from the QM9 data set using the QS_1 metric. Error bars represent the standard deviation. The dashed gray line represents the mean number of atoms obtained using the random selection.

Figure 6 compares the mean absolute error (MAE, L_1), the root-mean-squared error (RMSE, L_2), and the maximal error (MAXE, L_∞) in predicted energies obtained using the models trained on actively and randomly selected structures. All results are obtained averaging over three independent runs. The GM-NN model results is quite low values for MAE already when trained on randomly selected data points. AL can not improve on that. However, the RMSE does not reach the desired accuracy of 1 kcal mol^{-1} for randomly chosen training data even after training on 25 000 structures. The MAXE can have values of about 100 kcal mol^{-1} which makes the model too inaccurate for the energy prediction of these unusual or even extrapolative structures.

Applying the AL scheme we were able to reduce the maximal error by a factor of 2.3, measured for the models trained on 25 000 structures. This confirms that the AL algorithm selects molecules that better represent unusual molecules and, therefore, reduces the overall maximal error. The RMSE is reduced by a factor of about 1.7, again for the models trained on 25 000 structures. We were able to reach the accuracy of 1 kcal mol^{-1} using only about 13 000 structures. In reference [6], we had obtained an RMSE of 0.63 kcal mol^{-1} when training on 110 426 randomly selected structures. Using our AL approach it was now possible to reach an RMSE value of 0.67 using less than a quarter of the number of structures used previously.

To further test the proposed AL approach we have built a training data set containing 105 508 structures. The uncertainty of the GM-NN model at each AL iteration was estimated after training for 250 epochs. In total, 32 AL iteration were performed. Training the NN on the final training data for 5000 epochs we obtained an RMSE value of 0.28 kcal mol^{-1} and an MAE value of 0.21 kcal mol^{-1}. Compared to the results obtained in reference [6] for 110 426 randomly selected structures there is an improvement of a factor 2.25 and 1.29, respectively. Additionally, using the QS_1 strategy we could reduce the maximal error from 62.06 kcal mol^{-1} to a value of 2.24 kcal mol^{-1} and lower the number of structures with absolute error larger than 1 kcal mol^{-1} by a factor of 6 (about 90 structures for AL). With that we can conclude that the presented AL approach allows us to construct transferable and uniformly accurate machine-learned potentials, which used to be impossible using random selection.

Finally, we investigated the sizes of the molecules selected by the active learning approach at each iteration. Figure 7 shows that the QS_1 approach selects on the average smaller structures than the ones drawn randomly. This implies that the smaller structures in the QM9 data set contain local environments relevant to the larger structures. Note that a similar trend was obtained in [26]. For example, authors in [26] obtained the mean number of atoms of around 16 for the training data size of 6000 which is similar to our result. However, in contrast to their results we see that the model tries to select larger structures with increasing iteration step. This difference most probably comes from the different sizes of the training sets.

All GM-sNN models were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each for 5000 training epochs. The training took from about 3 h (5000 structures) to 14 h (25 261 structures).

3.3. N-ASW: molecular dynamics data

As a final test, we apply our AL approach to the data set recently used to study the adsorption and desorption dynamics of nitrogen atoms on top of amorphous solid water (ASW), which is relevant in astrochemical

processes [7]. The N-ASW data set is available directly from reference [38]. The goal of this section is to show to which extent the AL algorithm can be useful for real-time chemical simulations.

The purpose of the N-ASW data set was to describe the interaction of a nitrogen atom with an ASW surface. The model used in reference [7] contained 1498 atoms. To train a NN for this system highly heterogeneous data was needed. Therefore, the N-ASW data set contains structures with 3 to 378 atoms which result in 28 715 structures in total. Energies and atomic forces are calculated at the PBEh-3c/def2-mSVP [39] level of theory.

In this work, we split the data into five classes. The first class, C1, contains structures with $3 \leqslant N_{\text{at}} \leqslant 9$. The second class, C2, contains structures with $N_{\text{at}} = 22$ atoms and is the largest one with 18 735 structures in total. Other classes, C3, C4, and C5, contain structures with $N_{\text{at}} = 37$, $N_{\text{at}} = \{90, 91\}$, and $N_{\text{at}} \geqslant 100$, respectively. For this data set, a cutoff radius of 5.5 Å was used to include the long-range interactions of the nitrogen with the water ice surface.

Each AL cycle was initialized by randomly drawing 1000 structures for training and another 1000 for validation. Using the parameters of the trained models the uncertainty metric given in equation (14) was calculated and new structures were selected from the pool. Similar to the previous sections the pool contained the structures used for validation. In each iteration the model was allowed to select $0.1 \cdot N_{\text{train}}$ new structures and add them to the training set until the latter reached the size of 6105. This took 20 AL iterations in total, including the initialization.

The computational cost of the active learning algorithms was almost independent of the active learning iteration. The selection of m new training samples using the query strategy QS_1 required about 9 min on a single Intel Xeon CPU E5-2640 4. The query strategy QS_2 was about 80 times slower.

Figure 8 shows the correlation of the QS_1 metric (top left) and the QS_2 metric (top right) with the absolute force error, $\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$. From the figure one can see that both OED metrics are strongly correlated with the absolute force error. We obtained correlation coefficients of 0.88 and 0.79 for the QS_1 and QS_2 metrics, respectively. Here we have found that the QS_1 metric correlates better with the force error than QS_2. This appears to be caused mainly by the class C1 of the training data. Removing it from the data set results in correlation coefficients of 0.94 for QS_2 and 0.92 for QS_1. This can be explained as follows. The AL algorithm recognizes that the smallest structures are underrepresented in the data set since the model can barely transfer the knowledge obtained from the bigger structures using a cutoff of 5.5 Å. Therefore, it labels them with larger values of the uncertainty metric, but the force error is rather small for these simple structures. Interestingly, the QS_1 metric is less sensitive to it and can recognize the minor importance of these structures. Figure 8 shows that both AL algorithms recognize correctly that the last two data classes, C4 and C5, are the most relevant ones for a better generalization of the model. The most abundant data class, C2, is already well represented using the initial 1000 training points. Both algorithms assign small uncertainty metric values to C2 structures.

Similar to section 3.1 we calculated the correlation between both uncertainty metrics, QS_1 and QS_2. It should be mentioned that we have found a good correlation for all data classes except for C1 which contains structures with $3 \leqslant N_{\text{at}} \leqslant 9$. Therefore, it was left out in the correlation plot, see figure 8 (bottom left), as well as in figure 8 (bottom right). We have obtained a correlation coefficient of 0.94 between QS_1 and QS_2 metric. Any considerable improvement could not be found when using combined uncertainty estimation QS_3. The linear correlation coefficient reached only a value of 0.95 for $\beta = 4.6$.

Figure 9 compares the root-mean-squared error (RMSE) and the maximal error (MAXE) in predicted force and energy errors depending on the training set size. The structures used for the training of the respective models are selected randomly, or employing the QS_1 and QS_2 approaches. All results are obtained averaging over three independent runs. It can be seen that employing AL algorithms improves both the RMSE and MAXE in predicted energies and forces.

The RMSE in predicted forces is reduced by a factor of 1.2 for the training set size of 6105 when using the QS_1 metric. The MAXE is reduced by a factor of 2.4 for the training set size of 6105, and the desired accuracy of 1 kcal mol^{-1} Å^{-1} is reached already after training on about 2500 structures. Using random selection even after training on 6105 structures the maximal error is about 1.29 kcal mol^{-1} Å^{-1} and around 22 structures have an error in predicted forces larger than 1 kcal mol^{-1} Å^{-1}. After training on 3000 randomly selected structures we obtained around 120 structures that had force errors larger than 1 kcal mol^{-1} Å^{-1}. The MAXE obtained using around 3000 actively selected structures is 0.70 kcal mol^{-1} Å^{-1}. These values were obtained with the QS_1 metric, the results for QS_2 are similar.

In figure 9 one can see an even stronger improvement in RMSE and MAXE for the total energies when employing the AL scheme. The RMSE is lower by a factor of 1.8 than the one obtained using the model trained on randomly selected data points and equals 0.25 kcal mol^{-1} (6105 training points). The MAXE is reduced from 8.34 to 1.14 kcal mol^{-1}. Only around 9 structures show an energy error larger than 1 kcal mol^{-1} employing the model which was trained on 6105 actively selected samples (1000 of them were drawn

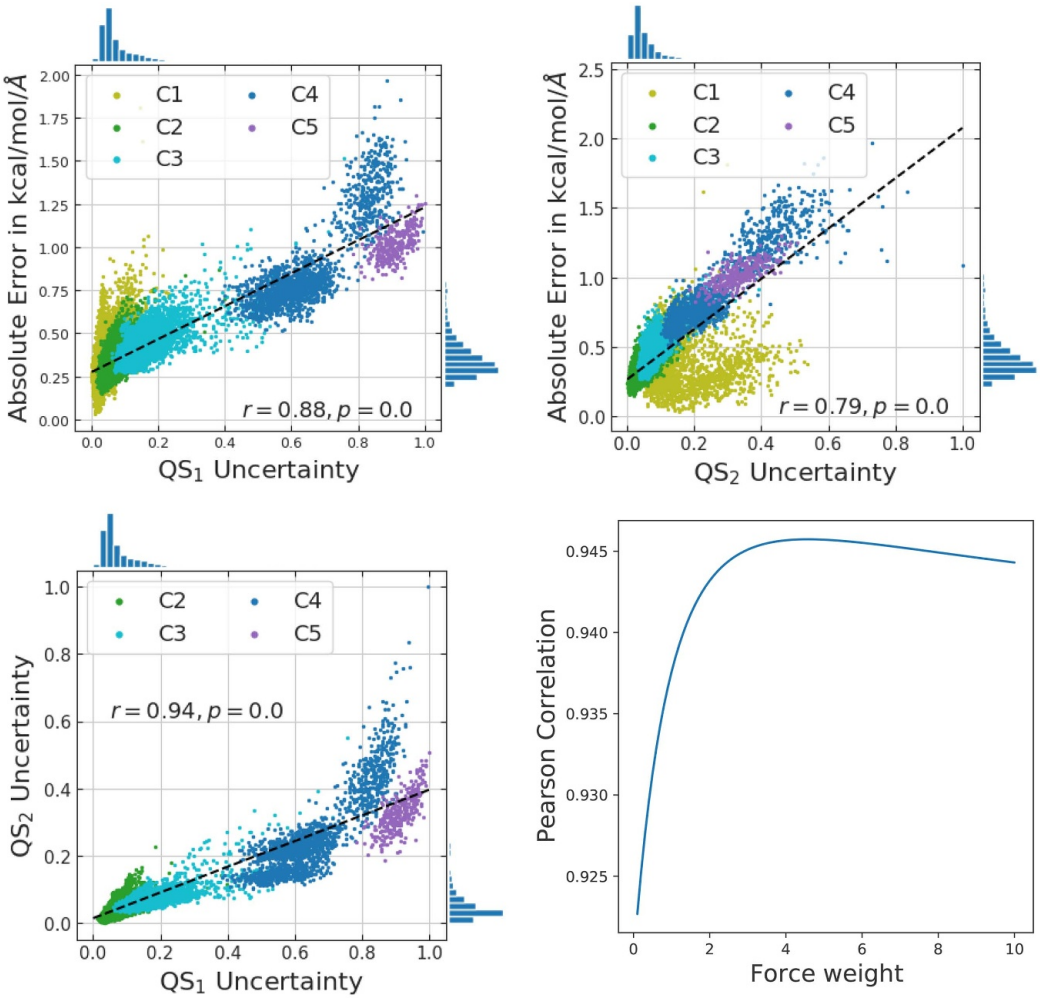


Figure 8. Correlation of the (top left) QS_1 metric and the (top right) QS_2 metric with the absolute force error $\Delta F = \frac{1}{3N_{\text{at}}} \sum_{i=1}^{N_{\text{at}}} \sum_{j=1}^{3} \left| F_{ij} - \hat{F}_{ij} \right|$ on the N-ASW data set. For both graphs, the correlation coefficient is shown as an inset along with the best fit line. (Bottom left) Correlation between QS_1 and QS_2 uncertainty metric. The correlation coefficient is shown as an inset along with the best fit line. (Bottom right) The dependence of the correlation coefficient obtained for the QS_3 metric and the absolute force error on the force weight β in equation (22). For both graphs, the first data class C1 was left out, see text.

randomly from the data set in the initialization step), while for 6105 randomly selected samples around 570 show an energy error larger than 1 kcal mol^{-1}. Similar to the discussion of the force errors all values were obtained with the QS_1 metric. The QS_2 approach leads to a similar outcome for the RMSE in predicted energies. However, the performance in MAXE is deteriorated by a factor of 2.4 compared to the QS_1 approach. This shows that the QS_1 is superior to QS_2 since it leads to a comparable improvement in force errors but much better performance in predicting total energies.

The above observations concerning the RMSE and MAXE in predicted forces and energies let us draw the following conclusion. Employing the proposed AL scheme allows us to create a uniformly accurate and transferable potential trained on 6105 structures, which can describe the nitrogen adsorption and desorption with the desired accuracy. To generate an equally accurate potential using the randomly selected structures would require much more data. Note that the potential obtained using only around 2500 data points can already be used for MD simulations. However, reference [7] aimed at binding energies, which required a high accuracy in the energies as well.

Finally, we want to study different approaches to define the threshold value of the OED uncertainty metric which is important for the application of the AL algorithm on-the-fly. We propose two approaches: (1) one can define the threshold value as the mean of the model's uncertainty over the training data, $\sigma_{\text{th}}^{(1)} = 1/N_{\text{train}} \sum_{i=1}^{N_{\text{train}}} \langle \Delta\sigma_{\hat{y}}^2(x_i) \rangle_{\mathcal{D}}$; (2) one can define the threshold value as the median of the model's uncertainty over the training data, $\sigma_{\text{th}}^{(2)} = \underset{x_i \in \mathcal{D}}{\text{median}} \langle \Delta\sigma_{\hat{y}}^2(x_i) \rangle_{\mathcal{D}}$. As a measure of informativeness of the

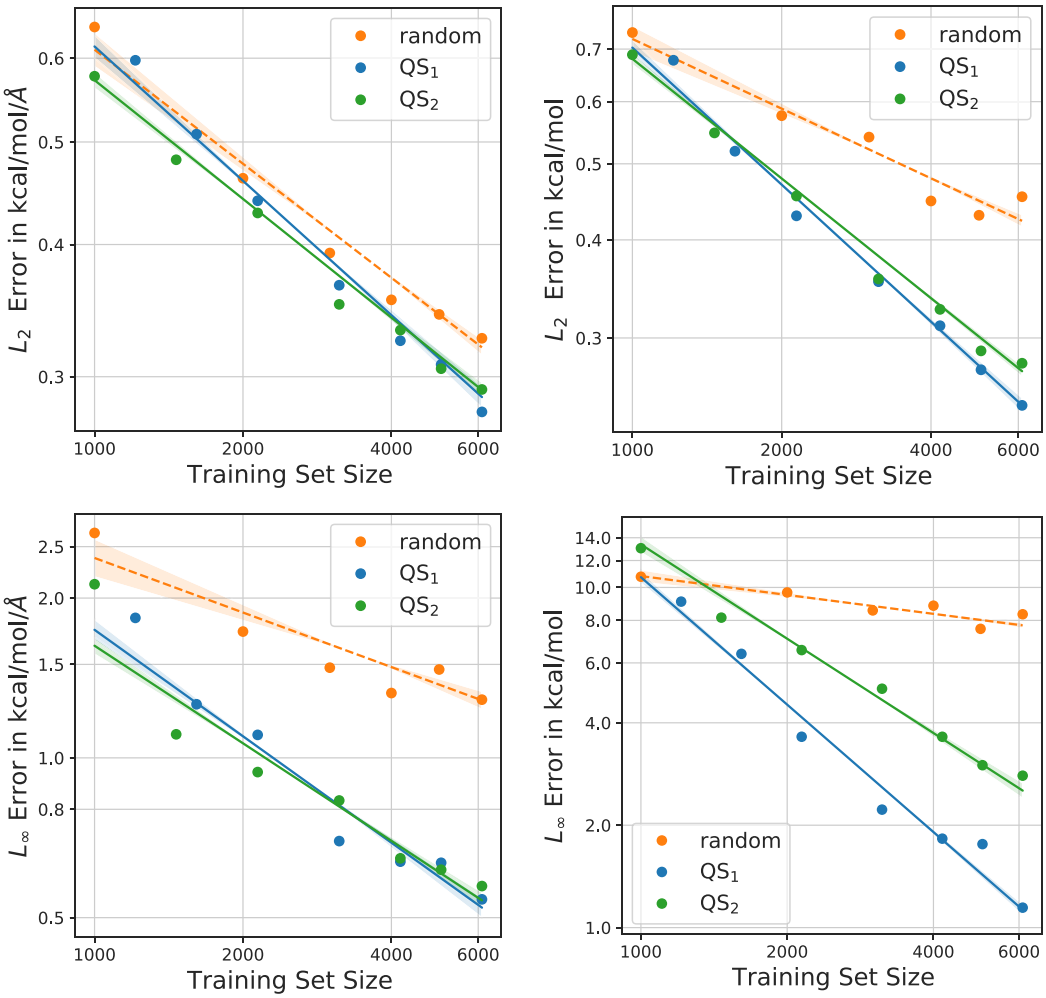


Figure 9. Log–log plot of the root-mean-squared error in predicted (top left) atomic forces and (top right) total energies on the N-ASW data set depending on the training set size. Log–log plot of the maximal error in predicted (bottom left) atomic forces and (bottom right) total energies depending on the training set size. Structures are selected randomly or using query strategies QS_1 and QS_2. For each learning curve a liner fit is plotted in addition.

selected structures we define the fraction $I = \langle \Delta\sigma^2_{\tilde{y}}(x^*)\rangle_{\mathcal{D}}/\sigma_{\text{th}}$. Structures with $I > 1$ are considered to be informative.

Figure 10 (left) shows the relative number of structures with $I > 1$ selected by both query strategies. Figure 10 (right) shows the ratio of the averaged informativeness of all structures selected in one AL iteration, i.e.

$$\bar{I}^{(1)} = \frac{1}{\#x^*}\sum_{x^*}\langle \Delta\sigma^2_{\tilde{y}}(x^*)\rangle_{\mathcal{D}}/\sigma^{(1)}_{\text{th}}$$

or

$$\bar{I}^{(2)} = \underset{x^*\in\mathcal{P}}{\text{median}}\langle \Delta\sigma^2_{\tilde{y}}(x^*)\rangle_{\mathcal{D}}/\sigma^{(2)}_{\text{th}}.$$

From figure 10 one can see that strongest improvement is achieved at around $N_{\text{train}} = 3000$, in accordance with the MAXE shown in figure 9. The relative number of informative structures decreases strongly for all query strategies except for $\sigma^{(2)}_{\text{th}}(\text{QS}_2)$. Additionally, $\sigma^{(2)}_{\text{th}}(\text{QS}_1)$ shows an increase in figure 10 (top left) after reaching $N_{\text{train}} = 4500$. This can be explained as follows. Many of the selected structures have I close to 1, in case of QS_2, or slightly less than 1, in case of QS_1. Therefore, we see only a small averaged informativeness in figure 10 (top right). However, some amount of the selected structures are nevertheless able to reduce the model's output variance significantly and, thus, have high values of OED uncertainty. Note that one would include only the most informative structures when learning on-the-fly and, therefore, reduce

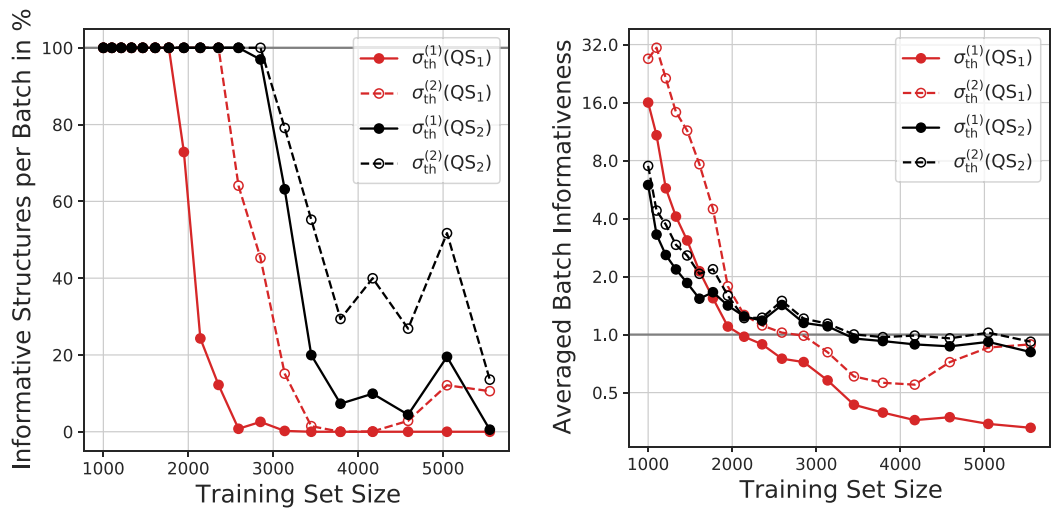


Figure 10. Comparison of different threshold values $\sigma_{th}^{(1)}$ and $\sigma_{th}^{(2)}$. (Left) The relative number of informative structures and (right) the log plot of the averaged informativeness of all structures selected in each AL iteration.

the number of structures necessary for the desired performance even further. In this work, we included $0.1 \cdot N_{train}$ new structures at each AL iteration which is less efficient, however allows us to test the performance of the AL algorithm and define the threshold values, $\sigma_{th}^{(1)}$ and $\sigma_{th}^{(2)}$.

Taking a more thorough look at the structures in the training data we have found that some of them were still underrepresented. These structures built the minority of the training data and have high values of the QS_1 and QS_2 metric. Therefore, they bias the threshold $\sigma_{th}^{(1)}$ to higher values. Taking this into account one can argue that $\sigma_{th}^{(2)}$ is superior to the latter giving the possibility to include the structures underrepresented but already included in the training set. This relationship can be easily seen comparing figure 9 and figure 10.

All GM-NN models for the N-ASW data set were trained on one NVIDIA Tesla V100-SXM2-32GB GPU each. The training of the model on 1000 structures for 5000 epochs took less than 4 h, and the training on 6105 structures for 5000 epochs was carried out during 17 h.

4. Conclusion

Machine learned potentials have been proven to have the potential to bridge the accuracy of *ab initio* methods and efficiency of empirical potentials. To construct potentials that are transferable and uniformly accurate for the chemical and conformational spaces of interest the model has to be able to define and select the extrapolative and most unusual structures for which, subsequently, the *ab initio* atomic forces and energies are calculated. Unfortunately, neural networks, the most frequently used machine learning approach in computational chemistry, have no inherent uncertainty estimators, as, for example, Gaussian processes have. For this reason it is not *a priori* clear, which data has to be included in the training set. The data sets employed are usually way larger than required or miss important regions of the configurational and chemical space.

In this paper, we proposed a novel active learning scheme for atomistic neural network potentials defined in the framework of the optimal experimental design. This approach uses the expected change in the estimated model's output variance to select structures that can be expected to reduce the generalization error of the model. The output variance is derived using the squared loss. Therefore, we were able to define three different query strategies based on the energy, the force, and the total losses.

To test the active learning approach we employed three different data sets. First, query strategies based on the energy and force squared losses were applied to an MD data set sampling the conformational space. Here, we have confirmed that the estimated gains are strongly correlated with the actual errors in predicted forces. We have seen that the active learning approach leads to a considerable reduction of the maximal error indicating that the most extrapolative and unusual structures are selected by the algorithm.

Next, the query strategy based on the energy squared loss was applied to a chemically diverse data set, the QM9 set [28, 29]. We have seen that employing the active learning scheme leads to a more accurate potential over the whole data set. We could achieve an RMSE close to the one obtained previously in reference [6] after training on randomly selected 110 426 structures, using only a fraction of the data set.

Finally, the AL approach was tested on the data set recently used to study the adsorption and desorption dynamics of nitrogen atoms on top of amorphous solid water. As a result, we obtained a model which predicts atomic forces with a maximal error of 0.54 kcal mol^{-1} Å^{-1} (0 structures with force error $\geqslant$1 kcal mol^{-1} Å^{-1}) and total energies with a maximal error of 1.14 kcal mol^{-1} (around 9 structures with energy error of $\geqslant$1 kcal mol^{-1}). The model was trained on only 6105 structures. This is a great improvement over the model trained on 6105 randomly selected structures, which resulted in a maximal error of the predicted total energies of about 8.34 kcal mol^{-1} and failed to predict the total energies with the desired accuracy of 1 kcal mol^{-1} for around 570 structures. The maximal error in predicted forces is around 1.29 kcal mol^{-1} Å^{-1}.

We also studied the possible threshold values for the OED uncertainty metrics which is necessary to perform learning on-the-fly. We have found that the median of the expected change in the estimated output variance over the training set is the better choice compared to the mean. This is due to the fact that the mean value is more sensitive to the structures already present but underrepresented in the training set. In general, we observed a good correlation of the estimated informativeness of selected structures with the maximal error reduction. This indicates that the threshold value can be defined naturally using both the mean and the median over the data used for training. Therefore, it is possible to use the proposed algorithm on-the-fly.

In summary, we presented an efficient approach for the active selection of the most informative structures from molecular data sets. We have shown that it leads to a considerable reduction of the training set sizes and at the same time to a reduction of the generalization error. Additionally, we have shown that it is possible to naturally define a threshold value for the OED uncertainty metric. This allows the application of the proposed method to the generation of transferable and uniformly accurate potentials on-the-fly.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors acknowledge financial support received in the form of a Ph.D. scholarship from the Studienstiftung des deutschen Volkes (German National Academic Foundation). We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075 – 390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). We also like to acknowledge the support by the Institute for Parallel and Distributed Systems (IPVS) of the University of Stuttgart for providing computer time.

ORCID iDs

Viktor Zaverkin https://orcid.org/0000-0001-9940-8548
Johannes Kästner https://orcid.org/0000-0001-6178-7669

References

[1] Hornak V, Abel R, Okur A, Strockbine B, Roitberg A and Simmerling C 2006 *Proteins* **65** 712–25
[2] Vanommeslaeghe K *et al* 2010 *J. Comput. Chem.* **31** 671–90
[3] Halgren T A 1996 *J. Comput. Chem.* **17** 490–519
[4] Mackerell Jr A D 2004 *J. Comput. Chem.* **25** 1584–604
[5] Dral P O 2020 *J. Phys. Chem. Lett.* **11** 2336–47
[6] Zaverkin V and Kästner J 2020 *J. Chem. Theory Comput.* **16** 5410–21
[7] Molpeceres G, Zaverkin V and Kästner J 2020 *Mon. Not. R. Astron. Soc.* **499** 1373–84
[8] Settles B 2009 Active learning literature survey Computer Sciences *Technical Report* 1648 (University of Wisconsin–Madison)
[9] Vandermause J, Torrisi S B, Batzner S, Xie Y, Sun L, Kolpak A M and Kozinsky B 2020 *npj Comput. Mater.* **6** 20
[10] Guan Y, Yang S and Zhang D H 2018 *Mol. Phys.* **116** 823–34
[11] Li Z, Kermode J R and De Vita A 2015 *Phys. Rev. Lett.* **114** 096405
[12] Li Z 2014 On-the-fly machine learning of quantum mechanical forces and its potential applications for large scale molecular dynamics PhD Thesis King's College, London
[13] Browning N J, Ramakrishnan R, von Lilienfeld O A and Roethlisberger U 2017 *J. Phys. Chem. Lett.* **8** 1351–9
[14] Huang B and von Lilienfeld O A 2020 *Nat. Chem.* **12** 945–51
[15] Smith J S, Nebgen B, Lubbers N, Isayev O and Roitberg A E 2018 *J. Chem. Phys.* **148** 241733
[16] Gastegger M, Behler J and Marquetand P 2017 *Chem. Sci.* **8** 6924–35
[17] Zhang L, Lin D Y, Wang H, Car R and E W 2019 *Phys. Rev. Mater.* **3** 023804
[18] Schran C, Behler J and Marx D 2020 *J. Chem. Theory Comput.* **16** 88–99

[19] Gal Y and Ghahramani Z 2016 Dropout as a Bayesian approximation: representing model uncertainty in deep learning *Proc. 33rd Int. Conf. Machine Learning* (*Proc. Machine Learning Research* vol 48), eds M F Balcan and K Q Weinberger (New York, USA: PMLR) pp 1050–9
[20] Janet J P, Duan C, Yang T, Nandy A and Kulik H J 2019 *Chem. Sci.* **10** 7913–22
[21] Janet J P and Kulik H J 2017 *J. Phys. Chem.* A **121** 8939–54
[22] Nandy A, Duan C, Janet J P, Gugler S and Kulik H J 2018 *Ind. Eng. Chem. Res.* **57** 13973–86
[23] Cohn D A 1996 *Neural Netw.* **9** 1071–83
[24] MacKay D J C 1992 *Neural Comput.* **4** 590–604
[25] Fedorov V 1972 *Theory of Optimal Experiments* (New York: Academic)
[26] Gubaev K, Podryabinkin E V and Shapeev A V 2018 *J. Chem. Phys.* **148** 241727
[27] Podryabinkin E V and Shapeev A V 2017 *Comput. Mater. Sci.* **140** 171–80
[28] Ruddigkeit L, van Deursen R, Blum L C and Reymond J L 2012 *J. Chem. Inf. Model.* **52** 2864
[29] Ramakrishnan R, Dral P O, Rupp M and von Lilienfeld O A 2014 *Sci. Data* **1** 140022
[30] Reddi S J, Kale S and Kumar S 2019 (arXiv:1904.09237) [cs.LG]
[31] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: https://www.tensorflow.org/)
[32] Geman S, Bienenstock E and Doursat R 1992 *Neural Comput.* **4** 1–58
[33] Perdew J P, Burke K and Ernzerhof M 1996 *Phys. Rev. Lett.* **77** 3865
[34] Grimme S, Antony J, Ehrlich S and Krieg H 2010 *J. Chem. Phys.* **132** 154104
[35] Grimme S, Ehrlich S and Goerigk L 2011 *J. Comput. Chem.* **32** 1456
[36] Rassolov V A, Pople J A, Ratner M A and Windus T L 1998 *J. Chem. Phys.* **109** 1223
[37] Prechelt H 2012 *Neural Networks: Tricks of the Trade* (Berlin: Springer)
[38] Molpeceres G, Zaverkin V and Kästner J 2020 N-ASW: molecular dynamics data (v1) (available at: http://doi.org/10.5281/zenodo.4013889)
[39] Grimme S, Brandenburg J G, Bannwarth C and Hansen A 2015 *J. Chem. Phys.* **143** 054107

Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

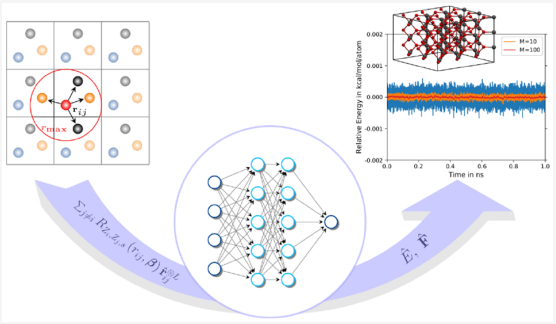
Fast and Sample-Efficient Interatomic Neural Network Potentials for Molecules and Materials Based on Gaussian Moments

Viktor Zaverkin, David Holzmüller, Ingo Steinwart, and Johannes Kästner*

Cite This: *J. Chem. Theory Comput.* 2021, 17, 6658−6670 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Artificial neural networks (NNs) are one of the most frequently used machine learning approaches to construct interatomic potentials and enable efficient large-scale atomistic simulations with almost ab initio accuracy. However, the simultaneous training of NNs on energies and forces, which are a prerequisite for, e.g., molecular dynamics simulations, can be demanding. In this work, we present an improved NN architecture based on the previous GM-NN model [Zaverkin V.; Kästner, J. *J. Chem. Theory Comput.* **2020**, *16*, 5410−5421], which shows an improved prediction accuracy and considerably reduced training times. Moreover, we extend the applicability of Gaussian moment-based interatomic potentials to periodic systems and demonstrate the overall excellent transferability and robustness of the respective models. The fast training by the improved methodology is a prerequisite for training-heavy workflows such as active learning or learning-on-the-fly.



1. INTRODUCTION

Approximate methods, such as empirical force fields (FFs),[1−3] are an integral part of modern computational chemistry and materials science. While the application of first-principles methods, such as density functional theory (DFT), to even moderately sized molecular and material systems is computationally very expensive, approximate methods allow for simulations of large systems over long time scales. During the last decades, machine-learned potentials (MLPs)[4−33] have risen in popularity due to their ability to be as accurate as the respective first-principles reference methods, the transferability to arbitrary-sized systems, and the capability of describing bond breaking and bond formation as opposed to empirical FFs.[34]

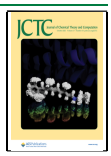
Interpolating abilities of neural networks (NNs)[35] promoted their broad application in computational chemistry and materials science. NNs were initially applied to represent potential energy surfaces (PESs) of small atomistic systems[36,37] and were later extended to high-dimensional systems.[21] Once trained, the computational cost of MLPs based on NNs does not scale with the number of data points used for training as opposed to kernel-based models.[5−8] Therefore, training sets can be as large as necessary to achieve the desired interpolation accuracy for applications in atomistic simulations, such as Monte Carlo (MC) sampling or molecular dynamics (MD).

A central ingredient for the construction of robust and accurate MLPs is a carefully chosen molecular representation. A wide variety of such representations have been proposed in the literature.[4−20] In our previous work,[19] we have shown that geometric moments can be successfully used for this purpose. Combined with radial basis functions and used with neural networks, we arrived at Gaussian moment neural networks (GM-NNs).[19] We have proposed an NN architecture that allowed us to use a single NN for all atomic species, in contrast to using an individual NN for each species as frequently necessary previously.[12,21,23] This improved the times needed for an individual evaluation of energies and atomic forces.

In this work, we present an improved GM-NN (iGM-NN) architecture for molecular and materials systems, which allows for very fast training, 10−20 times faster compared to the previous version, and shows improved prediction accuracy as well as excellent transferability to configurations not seen during training. Besides the improved NN architecture, including more flexible radial functions used for the construction of the Gaussian-moment representation, we extend the latter to the application on periodic systems.

While our improved NN allows for convenient training on a given data set, it is especially useful when building the training data set on-the-fly, e.g., during a molecular dynamics (MD) simulation, or offline, drawing new training structures from an

Received: May 27, 2021
Published: September 29, 2021



unlabeled data set. These workflows are referred to as learning-on-the-fly or active learning,[27,38−46] respectively. They reduce the number of required ab initio calculations but require frequent retraining of the model from updated training sets. The application of state-of-the-art machine learning models based on artificial NNs in such training-heavy workflows is somewhat hindered by their training time, which often ranges from several hours to several days.[19,30,47] We expect that our short training times will render active learning and learning-on-the-fly approaches much more attractive. Moreover, without access to GPUs, one can still train iGM-NN quickly on a CPU-only system.

To assess the quality of MLPs based on the improved methodology, we thoroughly benchmark the predictive accuracy as well as inference and training times on established molecular data sets from the literature, QM9[48,49] and MD17.[24−26] Finally, we studied two solid-state systems, TiO_2[23] and $Li_8Mo_2Ni_7Ti_7O_{32}$,[31] to investigate the applicability of iGM-NN potentials to periodic systems as well as to assess their robustness and transferability during real-time atomistic simulations.

2. THEORY

This section first introduces the representation used to describe the local atomic environments of atoms in molecular and solid-state systems throughout this work and the changes compared to our previous work.[19] Second, we describe the architecture of the feed-forward neural network used to learn the nonlinear map between physicochemical properties and the representation as well as its training.

2.1. Representation. In this work, we denote an atomic structure as $S^{(k)} = \{\mathbf{r}_i, Z_i\}_{i=1}^{N_{at}^{(k)}}$, where $\mathbf{r}_i \in \mathbb{R}^3$ are the Cartesian coordinates of atom i, and $Z_i \in \mathbb{N}$ encodes its species (e.g., $Z_i = 1$ for an H atom). A potential energy surface (PES) is a function that maps an atomic structure to a scalar energy, i.e., $f: S^{(k)} \to E \in \mathbb{R}$. The purpose of molecular machine learning (ML) approaches is to learn this mapping without solving the electronic Schrödinger equation.

A machine-learned PES has to satisfy several symmetries. These are the invariance with respect to global rotations, translations, and reflections of a molecular structure, as well as with respect to the exchange of atoms of the same atomic species. Additionally, it should allow the generalization to larger structures. The latter requirement is satisfied by decomposing the total energy of a system into a sum of "auxiliary" atomic energies[21]

$$\hat{E}(S^{(k)}, \boldsymbol{\theta}) \approx \sum_{i=1}^{N_{at}} \hat{E}_i(\mathbf{G}_i, \boldsymbol{\theta}) \tag{1}$$

where the invariances are encoded in the local representation $\mathbf{G}_i$, and $\boldsymbol{\theta}$ are the trainable parameters of the ML model. Making the representation dependent on the atomic species, i.e., $\mathbf{G}_i = \mathbf{G}_i(\mathbf{r}_i, Z_i, \{\mathbf{r}_j, Z_j\}_{j=1}^{N_c})$ where N_c is the number of atoms in the cutoff sphere, it is possible to train a single neural network (NN) for all species in the training set,[19] different than, e.g., ref 21. Note that in our approach the local representation is trainable, too.

One of the main challenges is to find an appropriate molecular representation that introduces all required symmetries into the ML model. Among the suite of existing invariant molecular representations, we build upon the trainable Gaussian moments (GM) that we introduced recently.[19] In this approach, the atomic distance vectors $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ are split into their radial and angular components, i.e., $r_{ij} = \|\mathbf{r}_{ij}\|_2$ and $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$, respectively. Note that for a periodic system one has to include the distance vectors to periodic images of atoms in the local neighborhood, see Figure 1. A tensor-valued function of the local atomic environment can then be written as

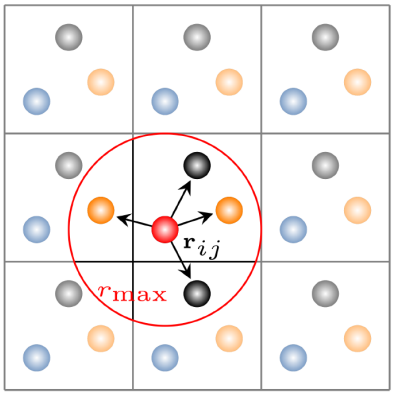


Figure 1. Two-dimensional schematic illustrating the local environment of an atom.

$$\boldsymbol{\Psi}_{i,L,s} = \sum_{j \neq i} R_{Z_i,Z_j,s}(r_{ij}, \boldsymbol{\beta}) \hat{\mathbf{r}}_{ij}^{\otimes L} \tag{2}$$

where $\hat{\mathbf{r}}_{ij}^{\otimes L} = \hat{\mathbf{r}}_{ij} \otimes \cdots \otimes \hat{\mathbf{r}}_{ij}$ is the L-fold tensor product of the angular components, and $R_{Z_i,Z_j,s}(r_{ij}, \boldsymbol{\beta})$ are nonlinear radial functions with trainable parameters $\boldsymbol{\beta}$. Note that in difference to ref 19, we extend the trainable parameter by an index s from 1 to N_{Basis}, i.e., $\beta_{Z_i,Z_j,s,s'}$ with s' being index from 1 to N_{Gauss}. The radial function has, similar to the one presented in ref 27, the form

$$R_{Z_i,Z_j,s}(r_{ij}, \boldsymbol{\beta}) = \frac{1}{\sqrt{N_{\text{Gauss}}}} \sum_{s'=1}^{N_{\text{Gauss}}} \beta_{Z_i,Z_j,s,s'} \Phi_{s'}(r_{ij}) \tag{3}$$

where N_{Gauss} is the number of Gaussian functions $\Phi_{s'}(r_{ij})$ placed along the radial coordinate, and the factor $1/\sqrt{N_{\text{Gauss}}}$ influences the effective learning rate during optimization inspired by the NTK parametrization.[50] The radial function

$$\Phi_{s'}(r_{ij}) = \left(\frac{2N_{\text{Gauss}}}{\pi r_{\text{max}}^2}\right)^{1/4} e^{-N_{\text{Gauss}}^2 (r_{ij} - \gamma_{s'})^2 / r_{\text{max}}^2} f_{\text{cut}}(r_{ij}) \tag{4}$$

is centered at $\gamma_{s'} = r_{\text{min}} + \frac{s'-1}{N_{\text{Gauss}}-1}(r_{\text{max}} - r_{\text{min}})$ and rescaled by the cosine cutoff function[21] $f_{\text{cut}}(r)$ with a cutoff radius $r_{\text{max}} > r_{\text{min}} = 0.5$ Å.

We initialize the trainable coefficients $\beta_{Z_i,Z_j,s,s'}$ independently uniformly on the interval $[-1, 1]$, similar to ref 19. However, we have found that on the QM9 data set,[48,49] better accuracy can be achieved by initializing $\beta_{Z_i,Z_j,s,s'} = \delta_{s,s'}$. We assume that the reason for this is that the distance r_{ij} between atoms already contains some information on the species Z_i and Z_j if the respective data set contains only structures in equilibrium, like the QM9 data set.

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

The tensors $\mathbf{\Psi}_{i,L,s}$ are invariant with respect to translations and to permutations of atoms of the same atomic species. A rotationally invariant representation is obtained by performing full contractions

$$
\begin{aligned}
G^{(1)}_{i,s_1} &= \mathbf{\Psi}_{i,0,s_1} \\
G^{(2)}_{i,s_1,s_2} &= (\mathbf{\Psi}_{i,1,s_1})_a (\mathbf{\Psi}_{i,1,s_2})_a \\
G^{(3)}_{i,s_1,s_2} &= (\mathbf{\Psi}_{i,2,s_1})_{a,b} (\mathbf{\Psi}_{i,2,s_2})_{a,b} \\
G^{(4)}_{i,s_1,s_2} &= (\mathbf{\Psi}_{i,3,s_1})_{a,b,c} (\mathbf{\Psi}_{i,3,s_2})_{a,b,c} \\
G^{(5)}_{i,s_1,s_2,s_3} &= (\mathbf{\Psi}_{i,1,s_1})_a (\mathbf{\Psi}_{i,1,s_2})_b (\mathbf{\Psi}_{i,2,s_3})_{a,b} \\
G^{(6)}_{i,s_1,s_2,s_3} &= (\mathbf{\Psi}_{i,2,s_1})_{a,b} (\mathbf{\Psi}_{i,2,s_2})_{a,c} (\mathbf{\Psi}_{i,2,s_3})_{b,c} \\
G^{(7)}_{i,s_1,s_2,s_3} &= (\mathbf{\Psi}_{i,1,s_1})_a (\mathbf{\Psi}_{i,3,s_2})_{a,b,c} (\mathbf{\Psi}_{i,2,s_3})_{b,c} \\
G^{(8)}_{i,s_1,s_2,s_3} &= (\mathbf{\Psi}_{i,3,s_1})_{a,b,c} (\mathbf{\Psi}_{i,3,s_2})_{a,b,d} (\mathbf{\Psi}_{i,2,s_3})_{c,d}
\end{aligned} \tag{5}
$$

where we use Einstein notation, i.e., the right-hand sides are summed over $a, b, c, d \in \{1, 2, 3\}$.

By construction, the tensors $\mathbf{G}^{(k)}$ in eq 5 have certain symmetries. In order to avoid including duplicate elements into the descriptor, in ref 19, we used upper triangular elements, i.e., those with indices satisfying $s_1 \leq s_2 \leq s_3$. However, tensors $\mathbf{G}^{(5)}$ and $\mathbf{G}^{(8)}$ possess symmetry only with respect to permutation of indices s_1 and s_2, whereas $\mathbf{G}^{(7)}$ does not show any symmetry to index permutation. Therefore, using upper triangular elements may lead to the results which depend on the specific order of contractions presented in eq 5. To deal with this shortage, we now use all elements of $\mathbf{G}^{(7)}$ and only elements with $s_1 \leq s_2$ for $\mathbf{G}^{(5)}$ and $\mathbf{G}^{(8)}$. For $N_{\text{Basis}} = 7$, this yields $N_{\text{Feature}} = 910$ invariant descriptors instead of $N_{\text{Feature}} = 427$ used in ref 19 at almost no computational overhead since in a practical implementation whole tensors $\mathbf{G}^{(k)}$ are computed. Additionally, this allows us to reduce the number of basis functions to, e.g., $N_{\text{Basis}} = 5$ corresponding to $N_{\text{Feature}} = 360$ invariant features, which reduces the overall computational cost.

2.2. Machine Learning. *2.2.1. Normalization of Input Features.* The normalization of feature vectors prior to training a neural network is known to improve the performance of the latter and, therefore, is usually applied to neural networks. For an architecture that uses trainable representation based on Gaussian moments (GM), we have found it to be redundant[19] except for data sets such as QM9,[48,49] a quantum chemistry data set that covers a wide range of chemical space. For the latter, we have found that normalizing the input features leads to a considerable improvement of the predictive accuracy. This is likely due to the specific initialization of $\beta_{Z_i Z_j s s'}$ for QM9, see Section 2.1, which results in non-negative radial functions $R_{Z_i Z_j s}(r_{ij}, \boldsymbol{\beta})$ at initialization. Therefore, we describe here a way to center and standardize the trainable local representation inspired by the Batch Normalization technique.[51] First, during training, the mean and the standard deviations over the current minibatch b containing N_{Struct} structures are calculated as

$$
\begin{aligned}
\boldsymbol{\mu}_b &= \frac{\sum_{k=1}^{N^{(b)}_{\text{Struct}}} \sum_{i=1}^{N^{(k)}_{\text{at}}} \mathbf{G}_i}{\sum_{k=1}^{N^{(b)}_{\text{Struct}}} \sum_{i=1}^{N^{(k)}_{\text{at}}} 1} \\
\sigma_b &= \frac{\sum_{k=1}^{N^{(b)}_{\text{Struct}}} \sum_{i=1}^{N^{(k)}_{\text{at}}} \left\| \mathbf{G}_i - \boldsymbol{\mu}_b \right\|_2^2}{\sum_{k=1}^{N^{(b)}_{\text{Struct}}} \sum_{i=1}^{N^{(k)}_{\text{at}}} N_{\text{Feature}}}
\end{aligned} \tag{6}
$$

The running statistics are computed as an exponentially moving average over the previous batches, i.e., as

$$
\begin{aligned}
\bar{\boldsymbol{\mu}}_b &= \gamma \bar{\boldsymbol{\mu}}_{b-1} + (1-\gamma)\boldsymbol{\mu}_b \\
\bar{\sigma}_b &= \gamma \bar{\sigma}_{b-1} + (1-\gamma)\sigma_b
\end{aligned} \tag{7}
$$

where $\bar{\boldsymbol{\mu}}_0$ and $\bar{\sigma}_0$ are initialized to zeros. The final values for the scale and shift of feature vector are rescaled by a factor $s_b = 1/(1 - t_b + \epsilon)$ with $\epsilon = 10^{-8}$ being a small number used to avoid division by zero and $t_b = \gamma \cdot t_{b-1}$ with $t_0 = 1$. The factor $1/(1 - t_b)$ is used to correct for the bias introduced by initializing $\bar{\boldsymbol{\mu}}_0$ and $\bar{\sigma}_0$ to zeros. We obtained good results by setting the momentum as $\gamma = 0.5^{1/N_{\text{Batch}}}$, where N_{Batch} is the number of batches per epoch, such that the information from one epoch ago is decayed by a factor of 0.5. The normalized feature vector reads

$$
\tilde{\mathbf{G}}_i = c_{\text{Scale}} \frac{\mathbf{G}_i - s_b \bar{\boldsymbol{\mu}}_b}{\sqrt{s_b \bar{\sigma}_b + \epsilon}} \tag{8}
$$

where b is the index of the last training batch. Different from standard Batch Normalization, we use the averaged statistics also during training and compute the variance over all feature vectors in the batch instead of normalizing all features individually. The former reduces the noisy nature of single-batch statistics, while the latter preserves the relative importance of the features. No trainable scale and shift parameters for the normalized feature vector are used since the normalization layer is followed by a fully connected layer, which can already learn to scale and shift the features. Instead of that, the normalized feature vector is rescaled by a constant $c_{\text{Scale}} = 0.4$, the benefits of which might be explained by the detrimental effects of a large initial neural network function.[52−54]

2.2.2. Network Architecture. Artificial neural networks (NNs) have been proven to be capable of approximating any nonlinear function relationship.[35] Therefore, they are a perfect candidate for learning the map between the structure and the respective physicochemical property. In this work, we use a fully connected feed-forward neural network consisting of two hidden layers of the following functional form

$$
\begin{aligned}
\hat{y}_i = 0.1 \cdot \mathbf{b}^{(3)} + \frac{1}{\sqrt{d_2}} \mathbf{W}^{(3)} \phi \Bigg(0.1 \cdot \mathbf{b}^{(2)} \\
+ \frac{1}{\sqrt{d_1}} \mathbf{W}^{(2)} \phi \left(0.1 \cdot \mathbf{b}^{(1)} + \frac{1}{\sqrt{d_0}} \mathbf{W}^{(1)} \mathbf{G}_i \right) \Bigg)
\end{aligned} \tag{9}
$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are weights and biases of the respective layer l. The parameters 0.1 and $1/\sqrt{d_l}$ correspond to the so-called NTK parametrization,[50] which is a theoretically motivated parametrization for fully connected layers. We initialize weights of the fully connected part by picking the respective entries from a normal distribution with zero mean

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

and unit variance. The trainable bias vectors are initialized to zero.

Our network parametrization and initialization are motivated by two goals. First, the variance of the neurons should be on the order of one in all network layers, and it should be approximately independent of the layer width d_l. This ensures that reasonable regions of the activation function are used and that the initialization does not have to be modified when changing the layer widths. Second, the amount that a trainable parameter needs to change during training should be on the order of one. This ensures that the updates performed by the Adam optimizer, which essentially uses scaled normalized gradients, roughly train all parameters equally fast. In contrast to the Kaiming initialization,[55] which has been motivated by the first goal, the NTK parametrization additionally satisfies our second goal since the weights are initialized with unit variance.

As an activation function, we use the Swish/SiLU activation function[56−58] $\phi(x) = \alpha x/(1 + \exp(-x))$ multiplied by a scalar α, instead of the softplus function used in ref 19. We choose $\alpha \approx 1.6765$ such that $\mathbb{E}_{x\sim\mathcal{N}(0,1)}\phi(x)^2 = 1$, i.e., the activation function preserves the second moment if the input is standard Gaussian.[59−61]

Throughout this work we used two different sizes of the feature vector with $d_0 = 910$ and $d_0 = 360$, obtained using $N_{\text{Basis}} = 7$ and $N_{\text{Basis}} = 5$, respectively. The computed local molecular representation $\mathbf{G}_i$ passes through two hidden layers with $d_1 = d_2 = 512$ hidden neurons. The output layer has a single $d_3 = 1$ output neuron since we predict a scalar energy.

2.2.3. Scaling and Shifting the Atomic Energy. In order to aid the training process, the output of the neural network can be scaled and shifted by the standard deviation σ and the mean μ of the per-atom average of the reference energies in the training set. In ref 19, we have shown that the convergence of the model can be improved by making these parameters trainable as well as dependent on the atomic species, i.e., σ_{Z_i} and μ_{Z_i}. Here, we improve on this idea by a proper initialization of atomic shifts μ_{Z_i} and scaling parameters σ_{Z_i}.

We have found that on data sets with different elemental compositions of structures, a better species-dependent initialization for the shifts μ_{Z_i} can be obtained using linear regression. Specifically, we solve the linear regression problem

$$\hat{\boldsymbol{\delta}} = \underset{\boldsymbol{\delta}\in\mathbb{R}^{N_Z}}{\operatorname{argmin}} \sum_{k=1}^{N_{\text{Train}}} (E(S_k) - \bar{E}(S_k) - \boldsymbol{\phi}(S_k)^T\boldsymbol{\delta})^2 + \lambda\|\boldsymbol{\delta}\|_2^2$$

$$\boldsymbol{\phi}(S_k) = \begin{pmatrix} \#\{i\colon Z_i = 1\} \\ \#\{i\colon Z_i = 2\} \\ \vdots \\ \#\{i\colon Z_i = N_Z\} \end{pmatrix} \tag{10}$$

where $\bar{E}(S_k) = N_{\text{at}}^{(k)}\mu$ is the structure's mean energy, N_Z is the number of species, $\lambda = 1$ is a regularization parameter, and $\boldsymbol{\delta}$ are the learned species-dependent differences to the mean atomic energy. Here, only the difference is regularized, which makes this method invariant to shifts in the total energy. This linear regression problem can be solved in a matter of seconds and scales linearly with the training set size.

Using the output $\hat{y}_i$ of the fully connected layers, we compute the predicted atomic energy as

$$\hat{E}_i(\mathbf{G}_i, \boldsymbol{\theta}) = c\cdot(\sigma_{Z_i}\hat{y}_i + \mu_{Z_i}) \tag{11}$$

where σ_{Z_i} and μ_{Z_i} are trainable parameters, initialized to 1 and to $(\mu + \hat{\delta}_{Z_i})/c$, respectively. We set the constant c to be the root-mean-squared error (RMSE) per atom of the mean atomic energy, i.e.

$$c = \sqrt{\frac{1}{N_{\text{total}}}\sum_{k=1}^{N_{\text{Train}}}\frac{(\bar{E}(S_k) - E(S_k))^2}{N_{\text{at}}^{(k)}}} \tag{12}$$

where $N_{\text{total}} = \sum_{k=1}^{N_{\text{Train}}} N_{\text{at}}^{(k)}$ is the total number of atoms in the training set. The introduction of the constant parameter c is motivated, like our use of the NTK parametrization, to achieve uniform learning speed across layers and data sets.

2.2.4. Loss Function and Training. In this work, we are interested in predicting total energies as well as atomic forces. Therefore, we minimize the following combined loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{k=1}^{N_{\text{Train}}}\left[\lambda_E\|E_k^{\text{ref}} - \hat{E}(S_k, \boldsymbol{\theta})\|_2^2 + \frac{\lambda_F}{3N_{\text{at}}^{(k)}}\sum_{i=1}^{N_{\text{at}}^{(k)}}\|\mathbf{F}_{i,k}^{\text{ref}} - \hat{\mathbf{F}}_i(S_k, \boldsymbol{\theta})\|_2^2\right] \tag{13}$$

to optimize the respective parameters of the trainable representation, fully connected neural network part as well as the parameters which scale and shift the output of the neural network. The reference values for the energy and atomic force are denoted by E_k^{ref} and $\mathbf{F}_{i,k}^{\text{ref}}$, respectively. The atomic force for an atom i is calculated by taking the partial derivative of the total energy with respect to the respective atomic position, i.e., $\hat{\mathbf{F}}_i(S_k, \boldsymbol{\theta}) = -\nabla_{r_i}\hat{E}(S_k, \boldsymbol{\theta})$.

Atomic forces containing $3N_{\text{at}}$ scalars provide much more information about a molecular structure compared to total energies. Moreover, atomic forces alone determine the dynamics of a chemical system and, therefore, play a crucial role during atomistic simulations such as molecular dynamics simulations. For this purpose, it is usual to weight the force error by a larger factor $\lambda_F > \lambda_E$ compared to total energies. In ref 19, we have found empirically parameters $\lambda_E = 1$ au and $\lambda_F = 100$ au Å^2 which already lead to improved predictive accuracy. In this work, we could improve on that by making the force weight dependent on N_{at}, i.e., $\lambda_F = 12N_{\text{at}}$ au Å^2. For data sets that contain only total energies, like QM9, we use only the energy loss, i.e., $\lambda_F = 0$ au Å^2.

To minimize the loss function in eq 13, the Adam optimizer[62] with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$ and a minibatch of 32 structures is employed. Moreover, we allow for layer-wise learning rates which decay linearly to zero by multiplying them with $(1 - r)$ where $r =$ step/max_step. For all data sets except QM9, we use an initial learning rate of 0.03 for the parameters of the fully connected layers, 0.02 for the parameters of the trainable GM representation, and 0.05 and 0.001 for the shift and scale parameters of atomic energies, respectively. To prevent overfitting on the QM9 data set, we use lower learning rates of 0.005, 0.0025, 0.05, and 0.001, respectively. This might be explained by noisier stochastic gradients due to the lack of force labels.

The model is trained for 500 epochs for the QM9 data set containing only total energies as reference values and for 1000 epochs for all other data sets used in Section 4. Overfitting was prevented using the early stopping technique.[63] After each epoch, the mean absolute errors (MAEs) of energies and forces

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

were evaluated on the validation set. After training, the model with the minimal sum of energy and force MAEs on the validation set was selected for further application on the test sets. While these hyperparameters worked reasonably well on the selected benchmarks for us, we want to emphasize that other trade-offs between training speed and accuracy can be achieved by increasing/decreasing the number of epochs and simultaneously decreasing/increasing initial learning rates.

3. DATA SETS

This section contains a brief description of the data sets used to benchmark the iGM-NN models in Section 4 on molecular and materials systems.

3.1. QM9. The QM9 data set[48,49] is a widely used benchmark for the prediction of several properties of molecules in equilibrium. Thus, all forces vanish, and only energies are used to train the model. The QM9 data set consists of 133,885 neutral, closed-shell organic molecules with up to 9 heavy atoms (C, O, N, F) and a varying number of hydrogen (H) atoms. However, since 3054 molecules from the original QM9 data set failed a consistency test,[49] we used only the remaining 130,831 structures in the experiments presented in Section 4.1. For this data set, a cutoff radius of r_{max} = 3.0 Å was chosen, similar to ref 19.

3.2. MD17. The MD17 data set[24−26] is a collection of structures, energies, and atomic forces of eight small organic molecules obtained from ab initio molecular dynamics (AIMD). For each molecule, a large variety of conformations is covered. The data set size varies from 150,000 to almost 1,000,000 conformations. It covers energy differences from 20 to 48 kcal/mol. Force components range from 266 to 570 kcal/mol/Å. We have chosen a cutoff radius of r_{max} = 4.0 Å for the MD17 data set, similar to ref 19. Note that we excluded the data set for the benzene molecule from experiments since in ref 18 the respective energies were found to be noisy.

3.3. Bulk TiO_2. The TiO_2 bulk data set[23] contains structures, energies, and atomic forces obtained from density functional theory (DFT) calculations using the PBE exchange-correlation functional[64] as implemented in PWSCF of the Quantum ESPRESSO package.[65] The data set contains distorted rutile, anatase, and brookite structures as well as the configurations sampled from short molecular dynamics simulations. In addition, supercell structures with oxygen vacancies are included. In total, the TiO_2 data set contains 7815 structures ranging in the size from 6 to 95 atoms. We used a cutoff of r_{max} = 6.5 Å for training the iGM-NN model in Section 4.3.

3.4. Quaternary Metal Oxide. To test the applicability of the iGM-NN approach to materials with a more complex chemical composition, we used the Li−Mo−Ni−Ti oxide (LMNTO) data set presented in ref 31. The reference energies and atomic forces are extracted from a 50 ps long AIMD simulation at 400 K. The AIMD simulations of the LMNTO system employed the strongly constrained and appropriately normed (SCAN) semilocal density functional.[66] The LMNTO data set contains 2616 periodic structures with 56 atoms each ($Li_8Mo_2Ni_7Ti_7O_{32}$). The data set is available free-of-charge from ref 67. We employed a cutoff radius of r_{max} = 6.5 Å for training the iGM-NN model in Section 4.4.

4. RESULTS

In this section, we apply the improved Gaussian moment neural network (iGM-NN) architecture to the molecular and materials data sets presented in Section 3.

4.1. Results for QM9. In Figure 2, we compare the predictive accuracy of a number of well-established models for the atomization energy of molecules in the QM9 data set. SchNet[11,24] and PhysNet[30] are message-passing neural networks (MPNNs) with distance-based interactions. PaiNN[20] is a recently proposed rotationally equivariant message-passing neural network architecture. Besides neural network-based approaches, we compare with the kernel-based model FCHL18[13] and its faster successor FCHL19[18] as well as with the linear model MTM.[27] Note that we compare mainly with FCHL19 and PaiNN since these have state-of-the-art accuracy and are also considered to be relatively fast. Finally, we compare the improved Gaussian moment neural network (iGM-NN) model with the previous Gaussian moment neural network (GM-NN) architecture.[19]

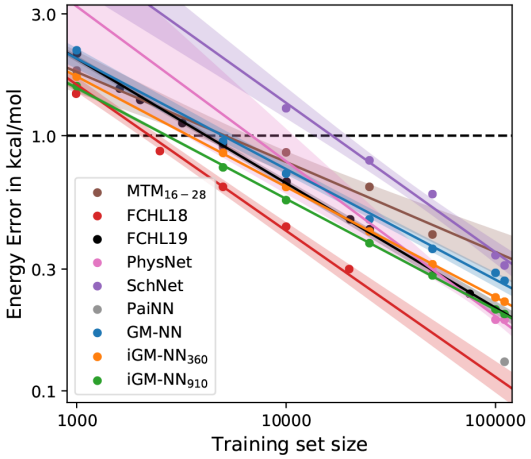


Figure 2. Learning curves for the QM9 data set. The mean absolute error (MAE) of atomization energy is plotted against the training set size. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression. The dashed black line represents the desired accuracy of 1 kcal/mol.

For the QM9 data set, we find the performance of the iGM-NN model to be among the models with the lowest out-of-sample MAE of atomization energy predictions. Compared to the GM-NN model, the MAE at 25000 training samples is 0.38, 0.42, and 0.47 kcal/mol for iGM-NN_{910}, iGM-NN_{360}, and GM-NN, respectively. The best performing model, FCHL18, predicts the atomization energy with an MAE of 0.30 kcal/mol, and the FCHL19 model predicts the atomization energy with an MAE of 0.47 kcal/mol, both trained on 20000 samples. Beyond that, the iGM-NN model shows a considerable improvement over GM-NN in the predictive accuracy when trained on 1000 samples and shows a comparable accuracy to the relatively slow FCHL18 method. We obtained an MAE of 1.56, 1.70, and 2.19 kcal/mol for iGM-NN_{910}, iGM-NN_{360}, and GM-NN, respectively. Compared to the best message-passing architecture, PaiNN, the MAE at 110,426 training samples is 0.13 and 0.20 kcal/mol for PaiNN and iGM-NN_{910}, respectively. As seen from Figure 2, the predictive accuracy of the iGM-NN model is equivalent to that of the PhysNet

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

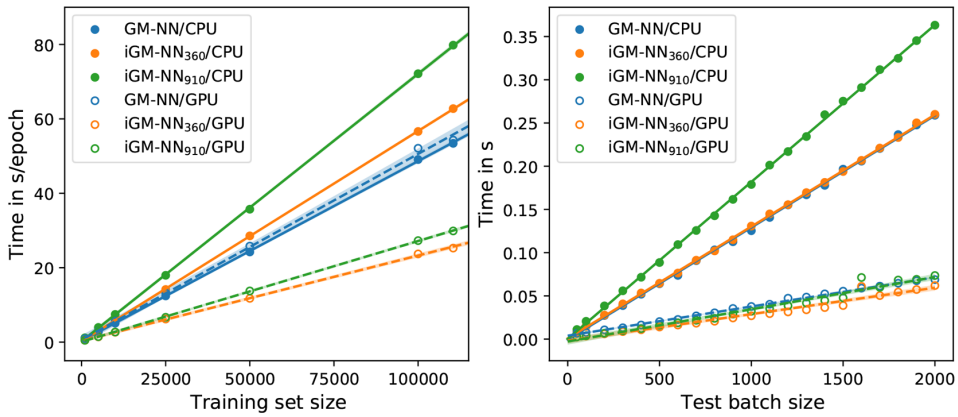


Figure 3. Training (left) and inference (right) times on the QM9 data set against the training set and test batch sizes, respectively. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression.

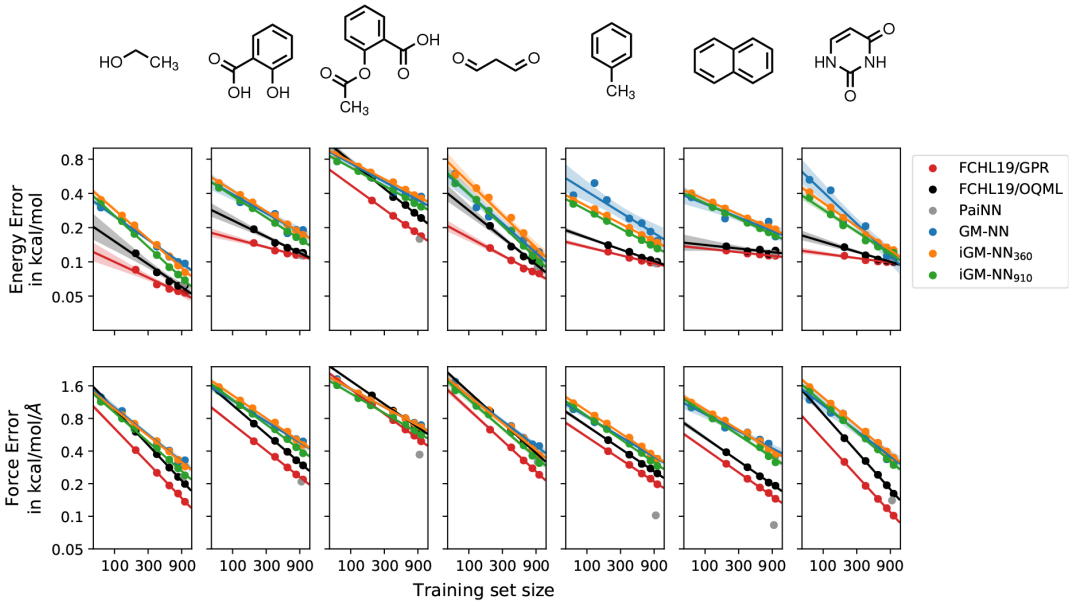


Figure 4. Learning curves for seven molecules from the MD17 data set. The mean absolute errors (MAEs) of total energies and atomic forces are plotted against the training set size. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression. The respective values are given from left to right for: ethanol, salicylic acid, aspirin, malonaldehyde, toluene, naphthalene, and uracil.

architecture and is better than the accuracy of the SchNet architecture.

In the following, we compare training and inference times of the iGM-NN model with the respective values for the GM-NN model, for the PaiNN architecture, and for the kernel-based FCHL18 and FCHL19 models. The respective values for GM-NN and iGM-NN are presented in Figure 3. The training and inference times for the respective literature methods are taken from refs 13, 18, and 20.

Compared to the GM-NN model, iGM-NN_{360} and iGM-NN_{910} need factors of 1.8−2.2 less time for a single training epoch at a single NVIDIA Tesla V100-SXM-32GB GPU. This speedup can be explained by the parallel data loading implemented in the iGM-NN model. On a 20-core node equipped with two Intel Xeon CPU E5-2640 v4 @ 2.40 GHz CPUs, we observed a slightly degraded efficiency of the model. This can be explained by the inclusion of batch normalization and an overall larger model compared to GM-NN. However, the improved convergence of the iGM-NN model results in a significantly reduced number of training epochs (500 instead of 5000) and therefore leads to a much shorter overall training time compared to the previous GM-NN model.

With the training times ranging from 4.0 h (GPU) to 11 h (CPU), obtained for 110,426 samples, iGM-NN is about 3−8 times faster than the kernel-based FCHL19 model. The latter needs 27 h to calculate the kernel matrix for all structures in the QM9 data set on a 24-core node equipped with two Intel Xeon E5-2680v3 @ 2.50 GHz CPUs.[18] Note that FCHL19 is about 20 times faster than FCHL18.[18] The reported times to train a PhysNet model are in the range of 1−2 days.[30]

The inference times of iGM-NN and GM-NN are similar, although for the iGM-NN model a larger neural network was

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

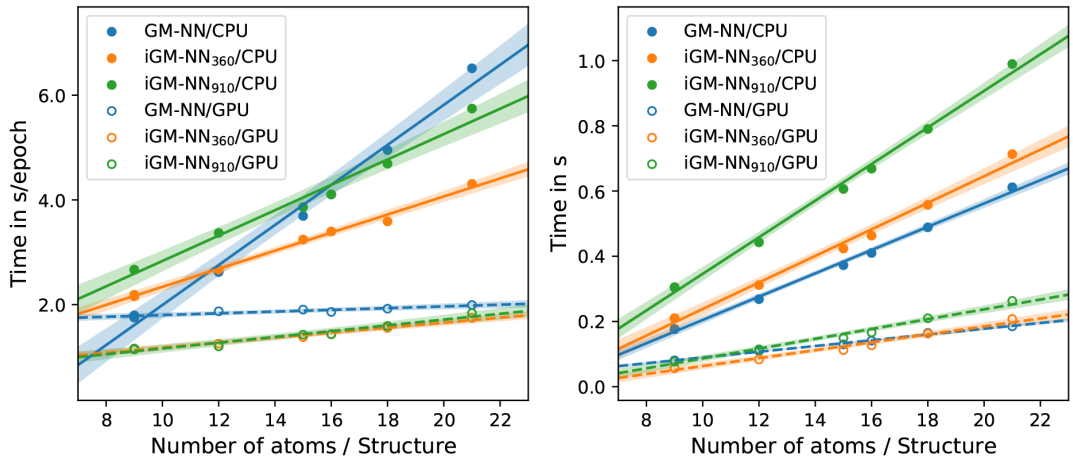


Figure 5. Training (left) and inference (right) times on the MD17 data set against the training set and test batch sizes, respectively. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression. All values are given for a training set size of 1000 structures and test batch size of 2000 structures.

employed. Evaluating the iGM-NN model on a random minibatch of 2000 structures, we obtained inference time in the range of 62−74 ms, while for the GM-NN model, we acquired 69 ms. Additionally, for a random minibatch of 50 molecules from QM9, the inference time of iGM-NN models is 2.5−2.6 ms, while for the PaiNN model, 13 ms were reported.[20]

In total, we observed that the iGM-NN model performs on par with the best kernel-based methods, which have been deemed to be more data-efficient than neural networks, and with the message-passing architectures. Moreover, we find that the proposed method is about 5 times faster per inference step than PaiNN.[20]

4.2. Results for MD17. We evaluate the ability to predict combined energies and forces on the MD17 benchmark. Figure 4 reports the MAE of force and energy predictions as a function of the number of training samples taken for seven molecules from the data set. Note that we excluded benzene from the experiments due to noisy energies. To demonstrate the data efficiency of iGM-NN, we use the more challenging training set sizes with up to 1000 structures. We compare the iGM-NN model to the kernel-based FCHL19/GPR and FCHL19/OQML approaches,[18] which are state-of-the-art kernel-based methods. The first approach incorporates derivatives in the training set within the framework of Gaussian process regression. The second one, operator quantum machine learning (OQML), allows for simultaneous training on the energies and forces. In addition, we compare to the recently proposed equivariant message-passing architecture, PaiNN.[20]

In general, we note that the improved GM-NN (iGM-NN) architecture leads to models that have similar or improved accuracy compared to the prediction errors reported for the GM-NN architecture in our previous paper.[19] For all molecules in the MD17 data set, the iGM-NN models learn somewhat faster compared to GM-NN for both energy and force training. As a general trend, the iGM-NN_{910} model needs about 20−40% fewer data to get the same accuracy as the GM-NN model. The iGM-NN_{360} model requires the same amount or about 10−20% fewer data to reach the same predictive accuracy as the GM-NN model, while using fewer invariant descriptors. For example, for aspirin, an MAE force error of about 0.7 kcal/mol/Å is obtained at roughly 600, 800, and 1000 samples for iGM-NN_{910}, iGM-NN_{360}, and GM-NN, respectively. Learning curves for all these models are presented in Figure 4.

As seen in Figure 4, iGM-NN provides a predictive accuracy similar to the kernel-based methods FCHL19/GPR and FCHL19/OQML. This implies, similar to the previous section, that the proposed architecture has a sample efficiency comparable to kernel-based approaches. Note that our approach is outperformed only by the FCHL19/GPR approach, which has an unfavorable trade-off for the time-to-train compared with the iGM-NN and FCHL19/OQML approaches. The equivariant message passing architecture PaiNN trained on 950 samples shows predictive accuracy on par with or better than the FCHL19/GPR approach; however, it is less efficient at inference compared to the iGM-NN approach, see below.

In the following, we report timings for the training and validation of the iGM-NN models for a training set of 1000 molecules and a validation batch of 2000 molecules taken from the MD17 data set. The respective values are presented in Figure 5. Compared to the GM-NN models, the speedup for a single training epoch is about a factor of 2−3 on an NVIDIA Tesla V100-SXM-32GB GPU and a 20-core node equipped with two Intel Xeon CPU E5-2640 v4 @ 2.40 GHz CPUs. Additionally, a 10-fold speedup is observed due to the reduced number of training epochs (from 10,000 to 1000).

For the FCHL19/OQML model, training times vary between 51 s (malonaldehyde) and 527 s (aspirin).[18] The iGM-NN_{360} model, with 573 s for malonaldehyde and 873 s for aspirin on a GPU, is clearly slower for smaller molecules compared to the FCHL19/OQML but comparably fast on larger ones. Additionally, the iGM-NN timing scales more favorably with the system size and the number of training samples. Compared to the FCHL19/GPR model (training time ranges from 1926 s to 101,451 s), the training of iGM-NN_{360} even on a 20-core CPU node is about 2−47 times faster. The timings for kernel evaluation in the case of the

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

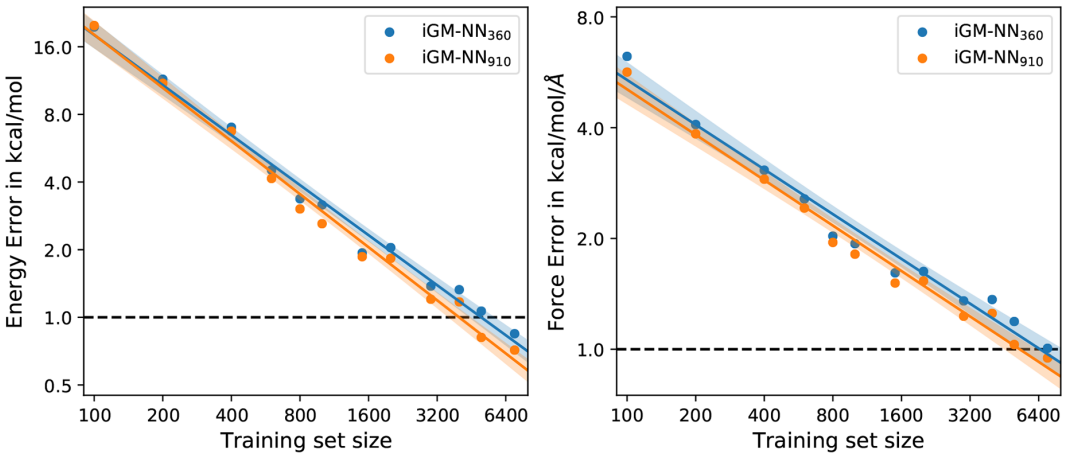


Figure 6. Learning curves for the TiO_2 data set. The mean absolute errors (MAEs) of total energies and atomic forces are plotted against the training set size. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression. The dashed black lines represent the desired accuracies of 1 kcal/mol and 1 kcal/mol/Å, respectively.

FCHL19/OQML and FCHL19/GPR models were performed on a 24-core node equipped with two Intel Xeon CPU E5-2680 v3 @ 2.50 GHz CPUs.[18]

The inference time of the iGM-NN_{360} model is 4.7 ms for a single ethanol and aspirin structure on a single GPU. The respective values correspond to a single energy and force evaluation times ranging from 0.5 ms/atom for ethanol to 0.2 ms/atom for aspirin, while PaiNN architecture required for a single reference calculation 14 ms (1.5 ms/atom) for ethanol and 15 ms (0.6 ms/atom) for aspirin. The inference times of the iGM-NN_{360} model, evaluated on a single GPU for a minibatch of 2000 structures, range from 57 ms for the smallest molecules, ethanol and malonaldehyde, to 208 ms for aspirin and are similar to those obtained for the GM-NN model, see Figure 5. These values correspond to a single energy and force evaluation times of 3−5 μs/atom, indicating a favorable scaling of inference times with the system size.

On a 20-core CPU node, we obtained, evaluating the iGM-NN_{360} model on single ethanol and aspirin structure, inference times ranging from 0.3 ms/atom to 0.1 ms/atom, respectively. Evaluating the respective model on a minibatch of 2000 structures, we acquired inference times ranging from 12 μs/atom for ethanol to 17 μs/atom for aspirin. The respective force prediction times for the FCHL19/OQML model are in the range of 5.7−23.5 ms/atom.[18]

Overall, we see that the iGM-NN, as well as GM-NN models, are at least one order of magnitude more efficient at inference compared to other state-of-the-art models. Therefore, the proposed models are favorable for atomistic simulations which require large system sizes and long time scales, e.g., molecular dynamics, while still showing comparable predictive accuracy.

4.3. Results for Bulk TiO_2. To determine the performance of the iGM-NN approach for a material system, we investigated titanium dioxide (TiO_2). The application of TiO_2 in the industry is versatile and ranges from the use as a pigment in paints[68] to the use as a photocatalyst for energy-related applications, e.g., photocatalytic water splitting.[69,70] TiO_2 occurs naturally in three polymorphs: rutile, anatase, and brookite. These phases build the training set used in this section, see Section 3.3. We should note at this point that density functional theory (DFT) without semiempirical corrections,[71,72] as used to generate our training data,[23] incorrectly predicts anatase to be more stable than rutile at standard conditions.[73,74] Machine learning approaches cannot improve on that. Finally, to test the transferability of the iGM-NN approach, we predict properties of two high-pressure TiO_2 phases, columbite and baddeleyite, which were not part of the training data.

Figure 6 shows learning curves for the iGM-NN predictions of the total energy and atomic forces based on 360 and 910 local invariant molecular descriptors. As seen from the figure, the desired accuracy of 1 kcal/mol for total energies is reached for a training set size of around 4000 and 5000 reference structures, using 910 and 360 invariant features, respectively. The desired accuracy of 1 kcal/mol/Å for atomic forces is reached after training on 5000 and 7000 reference structures. In general, we see that iGM-NN_{910} models show only slightly lower MAE values compared to iGM-NN_{360} at the cost of higher training and inference times (see previous sections). Therefore, for the following experiments, we use only machine-learned potentials (MLPs) based on 360 invariant atomic features.

The MAEs of total energies and atomic forces, discussed above, are abstract quality measures of MLPs. In practice, the robustness and reliability of the potential in real-time applications provide a more rigorous assessment. Therefore, we assess the robustness and smoothness of the iGM-NN potential by applying it to a 3 × 3 × 3 rutile TiO_2 supercell containing 54 TiO_2 formula units (162 atoms) to run a molecular dynamics (MD) simulation. Note that for an MD simulation a smooth, continuous energy surface is required to facilitate the numerical integration of the equation of motion. Figure 7 shows the total energy during MD simulation in the microcanonical (NVE) statistical ensemble, carried out within the ASE simulation package[75] over 1.0 ns using a time step of 1.0 fs. The atomic velocities were initialized with a Maxwell−Boltzmann distribution for a temperature of 1000 K.

As seen in Figure 7, the total energy of the system is well conserved with fluctuations below 0.5 × 10^{-3} kcal/mol/atom. Since the raw data of the total energy fluctuates relatively strongly (blue line in Figure 7), we computed the running

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

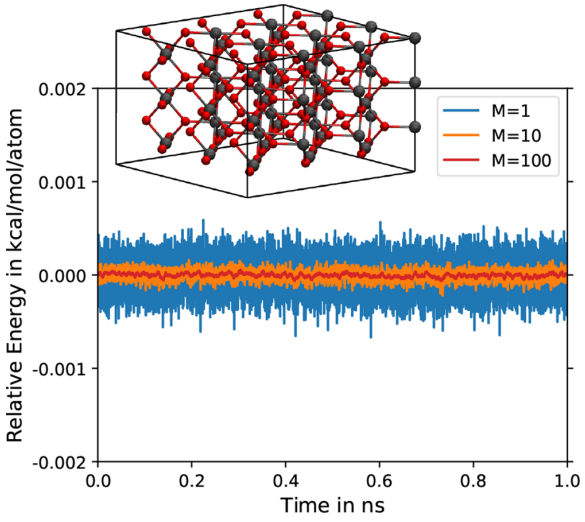


Figure 7. Fluctuation of the total energy during MD simulations of a $3 \times 3 \times 3$ rutile TiO_2 supercell containing 162 atoms over 1 ns using a time step of 1 fs in the microcanonical (NVE) ensemble employing the iGM-NN_{360} interatomic potential trained on 5000 reference structures. The atomic velocities were initialized with a Maxwell–Boltzmann distribution for a temperature of 1000 K. M is the window size for the running average in eq 14.

averages of the total energy to see whether the observable displays a time-drift. The running average for an observable O_i can be computed as

$$\hat{O}_i = \frac{1}{M+1} \sum_{j=i-M/2}^{i+M/2} O_j \tag{14}$$

where M is the window size of the running average. From Figure 7, it can be observed that the total energy shows only a slight energy drift of $(-2.0 \pm 0.4) \times 10^{-5}$ kcal/mol/atom/ns, which is another indication for the numerical stability of the machine-learned potential.

Along with the aforementioned energy conservation, we investigate whether the iGM-NN models correctly reproduce the relative phase stability. As can be seen from Figure 8, the iGM-NN potential smoothly reproduces the energy as a function of the lattice volume of rutile, anatase, and brookite. Moreover, we observe a smooth energy dependence on the lattice volume for the columbite and baddeleyite high-pressure phases which were not included in the training set. While the former exhibits similarities to anatase and rutile polymorphs, the latter is denser, and each titanium is 7-fold coordinated by oxygen, as opposed to the octahedral coordination in other phases. Therefore, we clearly see that the iGM-NN approach results in smooth potentials even when extrapolating to configurations and crystal structures that have not been seen before.

To study the relative phase stability more rigorously, we computed the relative phase energies, unit cell volumes, and bulk moduli by a fit of the stabilized jellium equation of state (SJEOS),[76] as implemented in ASE, to the iGM-NN potential energy curves. The reference DFT values are taken from ref 23, where they were computed by a fit of the Birch–Murnaghan equation of state[77] to the DFT potential energy curves. Table 1

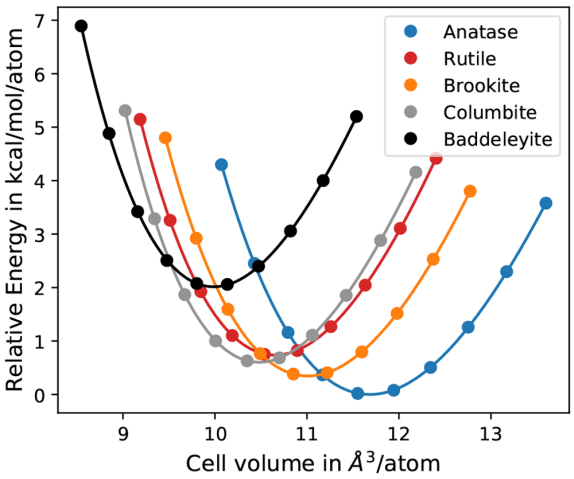


Figure 8. Relative energy per atom as a function of the cell volume for the five TiO_2 crystal structures: anatase, rutile, brookite, columbite, and baddeleyite. Symbols correspond to the iGM-NN_{360} potential energies, and lines represent the fit of the stabilized jellium equation of state (SJEOS).[76] All energies are obtained employing the iGM-NN_{360} model trained on 5000 reference structures.

shows the respective values for the relative phase energies, unit cell volumes, and bulk moduli obtained using DFT and MLPs.

Table 1. Relative Energies (E_0) in kcal/mol/atom, Unit Cell Volumes (V_0) in Å^3/atom, and Bulk Moduli (B_0) in GPa of Five Different TiO_2 Phases Obtained Employing the Machine-Learned Interatomic Potential (iGM-NN_{360}) and Respective Reference Values as Computed Using Density Functional Theory (DFT)

	DFT			iGM-NN_{360}		
phase	E_0	V_0	B_0	E_0	V_0	B_0
anatase ($I4_1/amd$)	0.00	11.71	211	0.00	11.68	208
rutile ($P4_2/mnm$)	0.76	10.68	235	0.74	10.65	234
brookite ($Pbca$)	0.37	11.01	225	0.35	11.01	220
columbite ($Pbcn$)	0.67	10.50	230	0.60	10.48	241
baddeleyite ($P2_1/c$)	1.52	9.96	240	2.01	9.99	243

For three TiO_2 phases that were included in the training set (rutile, anatase, and brookite), we found an excellent agreement with the reference DFT values. The relative phase energies, unit cell volumes, and bulk moduli deviate by at most 0.02 kcal/mol/atom (corresponds to max. 5.4%), 0.03 Å^3/atom (0.3%), and 5 GPa (2.2%). For the high-pressure TiO_2 phases columbite and baddeleyite, which are not included in the training set, we have found the relative phase energies to deviate by 0.07 kcal/mol/atom and 0.49 kcal/mol/atom. In ref 23, somewhat larger deviations of about 0.83 kcal/mol/atom and 1.20 kcal/mol/atom were observed for columbite and baddeleyite, respectively. The unit cell volumes and bulk moduli predicted by the iGM-NN potential are in excellent agreement with DFT and deviate from the reference values only by at most 0.3% and 4.8%, respectively.

In total, the obtained results demonstrate the applicability of the iGM-NN potentials to material systems. While being accurate and sample-efficient, the iGM-NN approach shows an improved transferability to polymorphs not seen during

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658–6670

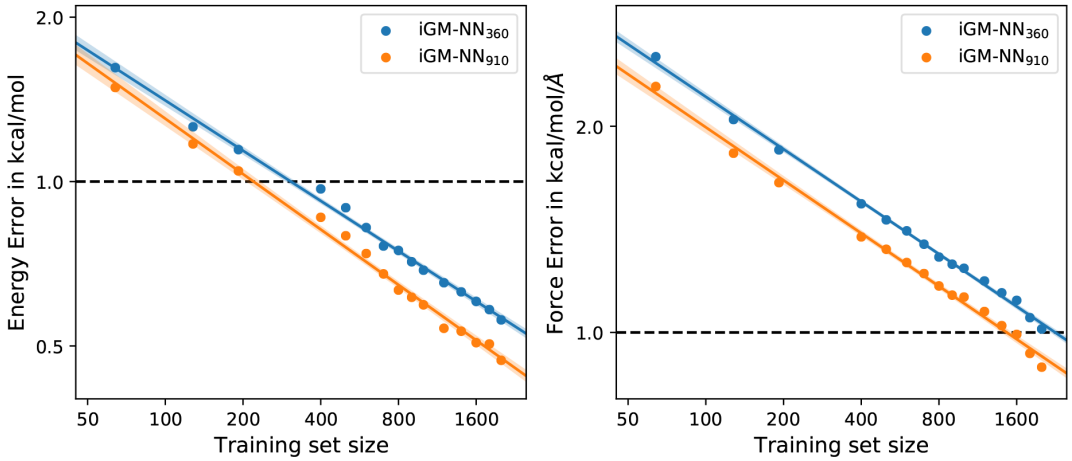


Figure 9. Learning curves for the LMNTO data set. The mean absolute errors (MAEs) of total energies (left) and atomic forces (right) are plotted against the training set size. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression. The dashed black lines represent the desired accuracies of 1 kcal/mol and 1 kcal/mol/Å, respectively.

training, compared to other machine learning approaches described in the literature. This makes it feasible to study, for example in this specific case, other TiO_2 phases without directly including them in the training set and, thus, facilitating the study and discovery of new materials.

4.4. Results for Quaternary Metal Oxide. To assess the applicability of machine-learned potentials (MLPs) based on the iGM-NN approach to a material with a more complex chemical composition, we finally studied Li−Mo−Ni−Ti oxide (LMNTO, $Li_8Mo_2Ni_7Ti_7O_{32}$). The respective data set is described in Section 3.4 and in ref 31. Note that LMNTO is a potential high-capacity positive electrode material for lithium-ion batteries and, therefore, is of high technological relevance.[78]

Figure 9 depicts learning curves of the total energy and atomic forces for the iGM-NN_{360} and iGM-NN_{910} models. For this system, we can compare our iGM-NN approach directly to the ænet code. The MAE for total energies at 700 training samples is 0.01 kcal/mol/atom, 0.01 kcal/mol/atom, and 0.11 kcal/mol/atom for iGM-NN_{360}, iGM-NN_{910}, and ænet,[12,23,31] respectively. For the MAE of atomic forces, we obtained a value of 1.35 kcal/mol/Å and 1.22 kcal/mol/Å with iGM-NN_{360} and iGM-NN_{910}, respectively, trained on 700 reference structures, while ænet results in an MAE of 15.22 kcal/mol/Å. Note that the ænet potential was trained using the recently proposed Taylor-expansion data enhancement approach,[31] which is more data-efficient, i.e., it uses less than 50% of the information contained in the data set.

As can be seen from Figure 9, both iGM-NN_{360} and iGM-NN_{910} reach the desired accuracy in predicted total energies already after training on 400 and 200 reference structures, respectively. The MAE target in predicted atomic forces of 1 kcal/mol/Å is reached after training on 2000 and 1400 reference structures for iGM-NN_{360} and iGM-NN_{910}, respectively. However, because of the similar performance of iGM-NN_{360} and iGM-NN_{910}, in the following experiments we use only iGM-NN_{360}, similar to Section 4.3.

As discussed in Section 4.3, the robustness and reliability of the MLP in a real-time simulation are a better assessment of its quality than abstract measures such as MAE values for predictions. Figure 10 shows the total energy during an MD

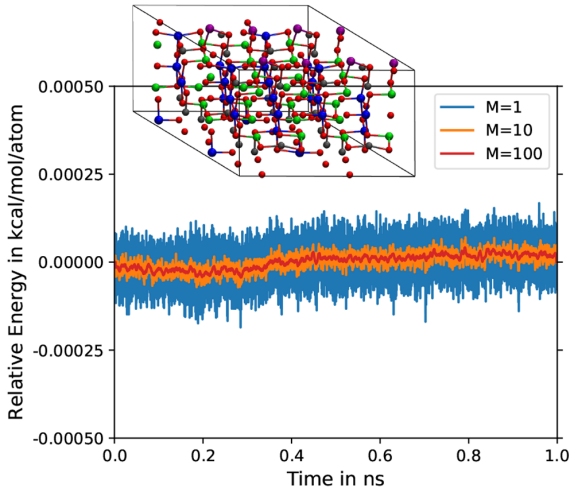


Figure 10. Fluctuation of the total energy during MD simulations of a 2 × 2 × 1 LMNTO supercell containing 224 atoms over 1 ns using a time step of 1 fs in the microcanonical (NVE) ensemble employing the iGM-NN_{360} interatomic potential trained on 2000 reference structures. The atomic velocities were initialized with a Maxwell−Boltzmann distribution for a temperature of 300 K. M is the window size for the running average in eq 14.

simulation in the microcanonical (NVE) statistical ensemble, carried out within the ASE simulation package[75] over 1.0 ns using a time step of 1.0 fs. The total energy of the system is well-conserved, and fluctuations remain below 0.1×10^{-3} kcal/mol/atom. Another indication for the numerical stability of the MLP based on the iGM-NN approach is the very low energy drift of merely $(5.9 \pm 0.1) \times 10^{-5}$ kcal/mol/atom/ns.

Overall, from the above experiments, we have seen that the iGM-NN approach is able to produce reliable and robust machine-learned potentials for multicomponent periodic systems. Additionally, we can argue that the proposed approach shows an excellent transferability since it allowed to run a stable MD simulation for 1 ns after training on only 2000 reference structures. Note that usually it is stated that

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

training sets with tens of thousands of reference structures are required to construct accurate MLPs.[31]

5. CONCLUSIONS

We have presented an improved version of the Gaussian moment neural network[19] which enables the generation of machine-learned potentials with an accuracy comparable to the established machine learning models. The presented modifications to the neural network architecture, the training process, and the implementation allow for much faster model training compared to the previous version in ref 19 and extend the applicability of Gaussian Moments to periodic structures. Fast training is a prerequisite of workflows with frequent retraining, such as active learning or learning-on-the-fly, which, in turn, reduce the number of required ab initio calculations.

Machine-learned potentials evaluated on two standard benchmark data sets, QM9 and MD17, have shown prediction accuracy comparable to state-of-the-art machine learning models. For models trained on atomization energies only, such as QM9, the accuracy is close to 0.2 kcal/mol and, thus, approaches the accuracy of message-passing models, such as PhysNet. The training time was reduced by factors of 10−20 compared to GM-NN.

Testing the proposed approach on the MD17 data set, which contains total energies and atomic forces, we have observed only a slight improvement in the prediction accuracy but a considerable reduction (factors of 10−20) of training times compared to the previous GM-NN. Note that GM-NN already could achieve an accuracy comparable to the state-of-the-art machine learning models. We should emphasize that the iGM-NN models outperform current state-of-the-art literature ML methods at inference time in terms of time required for single energy and force calculation.

We have shown that the iGM-NN approach can be used to generate accurate and robust machine-learned potentials for periodic systems, such as titanium dioxide (TiO_2) and LMNTO ($Li_8Mo_2Ni_7Ti_7O_{32}$). For both systems, the proposed approach could achieve the desired accuracy of 1 kcal/mol and 1 kcal/mol/Å in predicted total energies and atomic forces, respectively. Additionally, we assessed the quality of the machine-learned potentials by inspecting the energy conservation during a microcanonical molecular dynamics simulation. Along with energy fluctuations of the order of 10^{-4} kcal/mol/atom, we observed negligible time-drift, which demonstrates the smoothness and robustness of the iGM-NN potentials.

Testing the relative stability of TiO_2 polymorphs, we have observed an excellent transferability of the iGM-NN potentials. Specifically, we could obtain an accurate prediction for relative energies, unit cell volumes, and bulk moduli on high-pressure TiO_2 phases columbite and baddeleyite, which were not given in the training set.

In summary, the developments of this work aim to improve the applicability of machine-learned potentials to run various simulations for molecules and materials. Future work will also deal with long-range interactions important for, e.g., ionic liquids.

AUTHOR INFORMATION

Corresponding Author

Johannes Kästner − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0001-6178-7669; Email: kaestner@theochem.uni-stuttgart.de

Authors

Viktor Zaverkin − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0001-9940-8548

David Holzmüller − *Institute for Stochastics and Applications, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0002-9443-0049

Ingo Steinwart − *Institute for Stochastics and Applications, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0002-4436-7109

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.1c00527

Author Contributions

V.Z. and D.H. contributed equally to this work.

Notes

The authors declare no competing financial interest.

The full source code is available free-of charge at https://gitlab.com/zaverkin_v/gmnn.

ACKNOWLEDGMENTS

We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075-390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). The authors acknowledge support by the state of Baden-Württemberg through the bwHPC consortium for providing computer and GPU time. D.H. thanks the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for their support. V.Z. acknowledges the financial support received in the form of a Ph.D. scholarship from the Studienstiftung des Deutschen Volkes (German National Academic Foundation).

REFERENCES

(1) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins: Struct., Funct., Genet.* **2006**, *65*, 712−725.

(2) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; Mackerell, A. D., Jr. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2010**, *31*, 671−690.

(3) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996**, *17*, 490−519.

(4) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(5) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.

(6) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(7) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *87*, 184115.

(8) von Lilienfeld, O. A.; Ramakrishnan, R.; Rupp, M.; Knoll, A. Fourier series of atomic radial distribution functions: A molecular

fingerprint for machine learning models of quantum chemical properties. *Int. J. Quantum Chem.* **2015**, *115*, 1084−1093.
(9) Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **2016**, *14*, 1153−1173.
(10) Khorshidi, A.; Peterson, A. A. Amp: A modular approach to machine learning in atomistic simulations. *Comput. Phys. Commun.* **2016**, *207*, 310−324.
(11) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. In *NeurIPS 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: 2017; pp 991−1001.
(12) Artrith, N.; Urban, A.; Ceder, G. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *96*, 014112.
(13) Faber, F. A.; Christensen, A. S.; Huang, B.; von Lilienfeld, O. A. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.* **2018**, *148*, 241717.
(14) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Phys. Rev. Lett.* **2018**, *120*, 143001.
(15) Zhang, L.; Han, J.; Wang, H.; Saidi, W.; Car, R.; E, W. End-to-end Symmetry Preserving Inter-atomic Potential Energy Model for Finite and Extended Systems. *NeurIPS*; 2018; p 4436.
(16) Kocer, E.; Mason, J. K.; Erturk, H. A novel approach to describe chemical environments in high-dimensional neural network potentials. *J. Chem. Phys.* **2019**, *150*, 154102.
(17) Zhang, Y.; Hu, C.; Jiang, B. Embedded Atom Neural Network Potentials: Efficient and Accurate Machine Learning with a Physically Inspired Representation. *J. Phys. Chem. Lett.* **2019**, *10*, 4962−4967.
(18) Christensen, A. S.; Bratholm, L. A.; Faber, F. A.; von Lilienfeld, O. FCHL revisited: Faster and more accurate quantum machine learning. *J. Chem. Phys.* **2020**, *152*, 044107.
(19) Zaverkin, V.; Kästner, J. Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials. *J. Chem. Theory Comput.* **2020**, *16*, 5410−5421.
(20) Schütt, K. T.; Unke, O. T.; Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. 2021, abs/2102.03150. *ArXiv*. https://arxiv.org/abs/2102.03150 (accessed 2021-09-22).
(21) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.
(22) Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930−17955.
(23) Artrith, N.; Urban, A. An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for TiO2. *Comput. Mater. Sci.* **2016**, *114*, 135−150.
(24) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **2017**, *8*, 13890.
(25) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **2017**, *3*, e1603015.
(26) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **2018**, *9*, 3887.
(27) Gubaev, K.; Podryabinkin, E. V.; Shapeev, A. V. Machine learning of molecular properties: Locality and active learning. *J. Chem. Phys.* **2018**, *148*, 241727.
(28) Lubbers, N.; Smith, J. S.; Barros, K. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.* **2018**, *148*, 241715.
(29) 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. *Chem. Sci.* **2018**, *9*, 2261−2269.
(30) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678−3693.
(31) Cooper, A. M.; Kästner, J.; Urban, A.; Artrith, N. Efficient training of ANN potentials by including atomic forces via Taylor expansion and application to water and a transition-metal oxide. *npj Comput. Mater.* **2020**, *6*, 54.
(32) Smith, J. S.; Nebgen, B. T.; Zubatyuk, R.; Lubbers, N.; Devereux, C.; Barros, K.; Tretiak, S.; Isayev, O.; Roitberg, A. E. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **2019**, *10*, 2903.
(33) Sivaraman, G.; Gallington, L.; Krishnamoorthy, A. N.; Stan, M.; Csányi, G.; Vázquez-Mayagoitia, A.; Benmore, C. J. Experimentally Driven Automated Machine-Learned Interatomic Potential for a Refractory Oxide. *Phys. Rev. Lett.* **2021**, *126*, 156002.
(34) Mackerell, A. D., Jr. Empirical force fields for biological macromolecules: Overview and issues. *J. Comput. Chem.* **2004**, *25*, 1584−1604.
(35) Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251−257.
(36) Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **1995**, *103*, 4129−4137.
(37) Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* **2004**, *395*, 210−215.
(38) Settles, B. *Active Learning Literature Survey*; Computer Sciences Technical Report 1648; 2009.
(39) Li, Z.; Kermode, J. R.; De Vita, A. Molecular Dynamics with On-the-Fly Machine Learning of Quantum-Mechanical Forces. *Phys. Rev. Lett.* **2015**, *114*, 096405.
(40) Podryabinkin, E. V.; Shapeev, A. V. Active learning of linearly parametrized interatomic potentials. *Comput. Mater. Sci.* **2017**, *140*, 171−180.
(41) Smith, J. S.; Nebgen, B.; Lubbers, N.; Isayev, O.; Roitberg, A. E. Less is more: Sampling chemical space with active learning. *J. Chem. Phys.* **2018**, *148*, 241733.
(42) Zhang, L.; Lin, D.-Y.; Wang, H.; Car, R.; E, W. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Materials* **2019**, *3*, 023804.
(43) Gastegger, M.; Behler, J.; Marquetand, P. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.* **2017**, *8*, 6924−6935.
(44) Zaverkin, V.; Kästner, J. Exploration of transferable and uniformly accurate neural network interatomic potentials using optimal experimental design. *Mach. Learn.: Sci. Technol.* **2021**, *2*, 035009.
(45) Janet, J. P.; Duan, C.; Yang, T.; Nandy, A.; Kulik, H. J. A quantitative uncertainty metric controls error in neural network-driven chemical discovery. *Chem. Sci.* **2019**, *10*, 7913−7922.
(46) Vandermause, J.; Torrisi, S. B.; Batzner, S.; Xie, Y.; Sun, L.; Kolpak, A. M.; Kozinsky, B. On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events. *Npj Comput. Mater.* **2020**, *6*, 20.
(47) Schütt, K. T.; Kessel, P.; Gastegger, M.; Nicoli, K. A.; Tkatchenko, A.; Müller, K.-R. SchNetPack: ADeep Learning Toolbox For Atomistic Systems. *J. Chem. Theory Comput.* **2019**, *15*, 448−455.
(48) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(49) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.
(50) Jacot, A.; Gabriel, F.; Hongler, C. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *NeurIPS*; 2018.

(51) Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *PMLR* **2015**, *37*, 448−456.
(52) Lee, J.; Xiao, L.; Schoenholz, S. S.; Bahri, Y.; Novak, R.; Sohl-Dickstein, J.; Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *J. Stat. Mech.: Theory Exp.* **2020**, *2020*, 124002.
(53) Chizat, L.; Oyallon, E.; Bach, F. On Lazy Training in Differentiable Programming. *NeurIPS* **2019**, *32*, 2937−2947.
(54) Nonnenmacher, M.; Reeb, D.; Steinwart, I. Which Minimizer Does My Neural Network Converge To? 2020, abs/2011.02408. *ArXiv*. https://arxiv.org/abs/2011.02408 (accessed 2021-09-22).
(55) He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE ICCV* **2015**, 1026−1034.
(56) Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). 2016, abs/1606.08415. *arXiv*. https://arxiv.org/abs/1606.08415 (accessed 2021-09-22).
(57) Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Netw.* **2018**, *107*, 3−11.
(58) Ramachandran, P.; Zoph, B.; Le, Q. V. Searching for Activation Functions. 2018, abs/1710.05941. *ArXiv*. https://arxiv.org/abs/1710.05941 (accessed 2021-09-22).
(59) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. *NeurIPS*; 2017.
(60) Arora, S.; Du, S.; Hu, W.; Li, Z.; Salakhutdinov, R.; Wang, R. On Exact Computation with an Infinitely Wide Neural Net. *NeurIPS*; 2019.
(61) Lu, Y.; Gould, S.; Ajanthan, T. Bidirectional Self-Normalizing Neural Networks. 2020, abs/2006.12169. *ArXiv*. https://arxiv.org/abs/2006.12169 (accessed 2021-09-22).
(62) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2015, abs/1412.6980. *CoRR*. https://arxiv.org/abs/1412.6980 (accessed 2021-09-22).
(63) Prechelt, L. In *Neural Networks: Tricks of the Trade*, 2nd ed.; Montavon, G., Orr, G. B., Müller, K.-R., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; pp 53−67, DOI: 10.1007/978-3-642-35289-8_5.
(64) Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.
(65) Giannozzi, P.; Baroni, S.; Bonini, N.; Calandra, M.; Car, R.; Cavazzoni, C.; Ceresoli, D.; Chiarotti, G. L.; Cococcioni, M.; Dabo, I.; Corso, A. D.; de Gironcoli, S.; Fabris, S.; Fratesi, G.; Gebauer, R.; Gerstmann, U.; Gougoussis, C.; Kokalj, A.; Lazzeri, M.; Martin-Samos, L.; Marzari, N.; Mauri, F.; Mazzarello, R.; Paolini, S.; Pasquarello, A.; Paulatto, L.; Sbraccia, C.; Scandolo, S.; Sclauzero, G.; Seitsonen, A. P.; Smogunov, A.; Umari, P.; Wentzcovitch, R. M. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *J. Phys.: Condens. Matter* **2009**, *21*, 395502.
(66) Sun, J.; Ruzsinszky, A.; Perdew, J. P. Strongly Constrained and Appropriately Normed Semilocal Density Functional. *Phys. Rev. Lett.* **2015**, *115*, 036402.
(67) Cooper, A.; Kästner, J.; Urban, A.; Artrith, N. *Efficient Training of ANN Potentials by Including Atomic Forces via Taylor Expansion and Application to Water and a Transition-Metal Oxide* **2020**, *6*, 54.
(68) Buxbaum, G. *Industrial Inorganic Pigments*; John Wiley & Sons: 2008.
(69) Kavan, L.; Grätzel, M.; Gilbert, S. E.; Klemenz, C.; Scheel, H. J. Electrochemical and Photoelectrochemical Investigation of Single-Crystal Anatase. *J. Am. Chem. Soc.* **1996**, *118*, 6716−6723.
(70) Khan, S. U. M.; Al-Shahry, M.; Ingler, W. B. Efficient Photochemical Water Splitting by a Chemically Modified n-TiO2. *Science* **2002**, *297*, 2243−2245.
(71) Conesa, J. C. The Relevance of Dispersion Interactions for the Stability of Oxide Phases. *J. Phys. Chem. C* **2010**, *114*, 22718−22726.
(72) Arroyo-de Dompablo, M. E.; Morales-García, A.; Taravillo, M. DFT+U calculations of crystal lattice, electronic structure, and phase stability under pressure of TiO2 polymorphs. *J. Chem. Phys.* **2011**, *135*, 054503.
(73) Muscat, J.; Swamy, V.; Harrison, N. M. First-principles calculations of the phase stability of TiO_2. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2002**, *65*, 224112.
(74) Labat, F.; Baranek, P.; Domain, C.; Minot, C.; Adamo, C. Density functional theory analysis of the structural and electronic properties of TiO2 rutile and anatase polytypes: Performances of different exchange-correlation functionals. *J. Chem. Phys.* **2007**, *126*, 154703.
(75) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The atomic simulation environment−a Python library for working with atoms. *J. Phys.: Condens. Matter* **2017**, *29*, 273002.
(76) Alchagirov, A. B.; Perdew, J. P.; Boettger, J. C.; Albers, R. C.; Fiolhais, C. Reply to "Comment on 'Energy and pressure versus volume: Equations of state motivated by the stabilized jellium model. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2003**, *67*, 026103.
(77) Birch, F. Finite Elastic Strain of Cubic Crystals. *Phys. Rev.* **1947**, *71*, 809−824.
(78) Lee, J.; Seo, D.-H.; Balasubramanian, M.; Twu, N.; Li, X.; Ceder, G. A new class of high capacity cation-disordered oxides for rechargeable lithium batteries: Li−Ni−Ti−Mo oxides. *Energy Environ. Sci.* **2015**, *8*, 3255−3265.

https://doi.org/10.1021/acs.jctc.1c00527
J. Chem. Theory Comput. 2021, 17, 6658−6670

Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

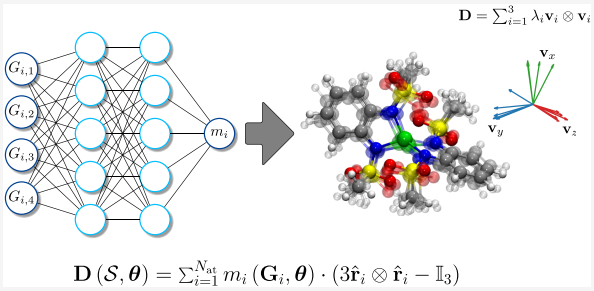
Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments

Viktor Zaverkin, Julia Netz, Fabian Zills, Andreas Köhn, and Johannes Kästner*

Cite This: J. Chem. Theory Comput. 2022, 18, 1−12 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: We propose a machine learning method to model molecular tensorial quantities, namely, the magnetic anisotropy tensor, based on the Gaussian moment neural network approach. We demonstrate that the proposed methodology can achieve an accuracy of 0.3−0.4 cm^{-1} and has excellent generalization capability for out-of-sample configurations. Moreover, in combination with machine-learned interatomic potential energies based on Gaussian moments, our approach can be applied to study the dynamic behavior of magnetic anisotropy tensors and provide a unique insight into spin−phonon relaxation.



1. INTRODUCTION

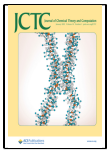
There is an ongoing interest in the investigation of magnetic properties of transition metal complexes and their possible applications as single-molecule magnets (SMMs), molecular quantum bits, and spintronic devices.[1−4] One of the most important factors influencing the magnetic properties is the magnetic anisotropy in the ground spin state.[2,5] Tailoring promising complexes in a way to achieve a large barrier for magnetic relaxation and minimizing other decay pathways[6−9] requires a detailed understanding of the underlying principles that determine the structure−property relationships.

While sampling based on first-principles methods can provide some insight, it is usually limited by its high computational cost. Approximate methods allow for simulations even taking into account the large size of conformational and chemical space of interest. Machine learning approaches with their ability to learn any complex nonlinear relationship between a structure and its related property and with their generalization capability are perfect candidates for the specific task of modeling magnetic anisotropy. During the last decades, machine learning methods have been gaining importance in several fields in computational chemistry allowing for, e.g., the construction of machine-learned force fields and prediction of vibrational spectra.[10−30] The ability of neural networks (NNs) to interpolate any nonlinear functional relationship[31] promoted their broad application in computational chemistry and materials science. NNs were initially applied to represent potential energy surfaces (PESs) of small atomistic systems[10,11] and were later extended to high-dimensional systems.[13] Once trained, the computational cost of machine-learned potentials (MLPs) based on NNs is independent of the number of data points used for training.

Case studies of the application of machine learning algorithms to study the magnetic properties of molecules and materials are somewhat rare. Recently, an approach for the construction of machine-learned interatomic potentials that are capable of reproducing both vibrational and magnetic degrees of freedom was proposed[32] and applied to bcc iron. Closer to the present work are the investigations presented in refs 33 and 34, where tensorial properties such as the zero-field splitting (ZFS) tensor **D** and the Zeeman-splitting (ZS) tensor **g** were modeled by machine learning approaches.

In this work, we build upon the methodology developed in our group for constructing efficient and accurate interatomic potentials, referred to as Gaussian moment neural network (GM-NN).[24,30] The GM-NN uses neural networks (NNs) to map novel symmetry-preserving local atomic descriptors, Gaussian moments (GMs), to auxiliary atomic quantities and includes both the geometric and alchemical information about the atomic species of both the central and neighbor atoms. For all atomic contributions, only a single NN has to be trained, in contrast to using an individual NN for each species as frequently done in the literature.[13,15] To allow for efficient training on properties such as **D** tensors, we introduce a novel approach for the encoding of relevant invariances into the output of the respective machine learning model. Moreover, we propose a neural network architecture for the task of modeling the magnetic anisotropy of SMMs.

Received: August 24, 2021
Published: December 9, 2021



https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

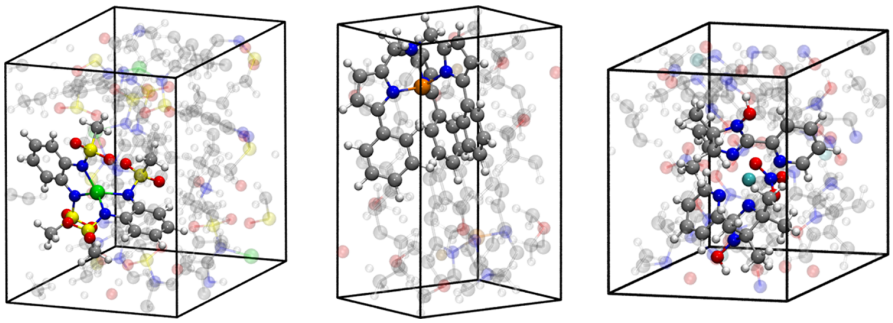


Figure 1. Illustration of the structure of (left) the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ unit cell, (middle) the $[Fe(tpa)^{Ph}]^-$ unit cell, and (right) the $[Ni(HIM_2{-}py)_2NO_3]^+$ unit cell. The color code is: Co, green; Fe, orange; Ni, turquoise; Na, brown; S, yellow; C, gray; O, red; N, blue; and H, white.

To assess the quality of the surrogate models obtained employing the proposed approach, we thoroughly benchmark the predictive accuracy on three promising SMMs: $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$,[6] $[Fe(tpa)^{Ph}]^-$,[7,35] and $[Ni(HIM_2{-}py)_2NO_3]^+$ complexes.[8] Finally, we use the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex as a case study to investigate the applicability of the proposed approach to the dynamics of the **D** tensor and examine its ability to model spin−phonon coupling effects.

In this work, we employ the open-source package Gaussian moment neural network (GM-NN) for constructing MLPs and applying them in atomistic simulations. The GM-NN source code is available free of charge from https://gitlab.com/zaverkin_v/gmnn.

2. MACHINE LEARNING MODELS

This section first introduces the representation of an atomistic system used to encode molecular and solid-state structures suitable for machine learning (ML) models. Second, we describe the approach to machine learning modeling the magnetic anisotropy tensors including the neural network architecture and its training, based on our previous work on Gaussian moments.[24,30] Finally, we outline the construction of the machine-learned interatomic potential energy used in Section 4.2 for molecular dynamics (MD) simulations of $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$.

2.1. Molecular Representation. A molecular or solid-state system is defined by its Cartesian coordinates $\mathbf{r}_i \in \mathbb{R}^3$ and its atomic numbers Z_i which, in the following discussion, are combined to $\mathcal{S} = \{\mathbf{r}_i, Z_i\}_{i=1}^{N_{at}}$ for simplicity. In the community of neural network (NN) model chemistry, it is usual to approximate properties by a sum of their atomic contributions. For example, the total energy of a system can be approximated[13] by a sum of atomic energies $\mathcal{E}_i$

$$\mathcal{E}(\mathcal{S}, \boldsymbol{\theta}) = \sum_{i=1}^{N_{at}} \mathcal{E}_i(\mathbf{G}_i, \boldsymbol{\theta}) \tag{1}$$

Total charge and the dipole moments can be defined via the atomic point charges q_i

$$Q_{tot}(\mathcal{S}, \boldsymbol{\theta}) = \sum_{i=1}^{N_{at}} q_i(\mathbf{G}_i, \boldsymbol{\theta})$$
$$\boldsymbol{\mu}(\mathcal{S}, \boldsymbol{\theta}) = \sum_{i=1}^{N_{at}} q_i(\mathbf{G}_i, \boldsymbol{\theta})\mathbf{r}_i \tag{2}$$

where each $\mathcal{E}_i$ and q_i are given as the output of an artificial NN, and all relevant invariances are encoded in the local representation $\mathbf{G}_i(\mathcal{S}, \boldsymbol{\beta})$ with $\boldsymbol{\beta}$ being trainable parameters. Thus, the main purpose of machine learning is to find such parameters $\boldsymbol{\theta}$ that the mapping $f\colon \mathcal{S} \to \{E, Q_{tot}, \boldsymbol{\mu}, ...\}$ is as close as possible to the reference values, $\{E^{ref}, Q_{tot}^{ref}, \boldsymbol{\mu}^{ref}, ...\}$.

The most challenging aspect of a machine learning model applied to a physicochemical problem is the definition of an appropriate representation of an atomistic system. In this work, we employ Gaussian moment representation, which defines the local atomic environment as[24,30]

$$\boldsymbol{\Psi}_{i,L,s} = \sum_{j \neq i} R_{Z_i,Z_j,s}(r_{ij}, \boldsymbol{\beta})\hat{\mathbf{r}}_{ij}^{\otimes L} \tag{3}$$

where we use the radial and angular components of the atomic distance vector $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, i.e., $r_{ij} = \|\mathbf{r}_{ij}\|_2$ and $\hat{\mathbf{r}}_{ij} = \mathbf{r}_{ij}/r_{ij}$, as inputs to nonlinear radial functions $R_{Z_i,Z_j,s}(r_{ij}, \boldsymbol{\beta})$ with trainable parameters $\boldsymbol{\beta}$ and the L-fold tensor product $\hat{\mathbf{r}}_{ij}^{\otimes L} = \hat{\mathbf{r}}_{ij} \otimes \cdots \otimes \hat{\mathbf{r}}_{ij}$, respectively. Z_i and Z_j correspond to the nuclear charges of the central atom i and its atomic neighbors j. As nonlinear radial functions, we employ a weighted sum of Gaussian functions, rescaled by the cosine cutoff function;[13] see refs 24 and 30. The parameters $\boldsymbol{\beta}$ are optimized during training similar to other parameters of an atomistic NN by minimization of a loss function. A rotationally invariant representation is obtained by computing full tensor contractions as described in refs 24 and 30.

As a final remark on the representation of an atomistic system, we want to discuss the possibility of learning different physicochemical properties by a single ML model. In general, it should be possible, given that the corresponding properties are available for the same atomistic system. Unfortunately, in this work, the energy and atomic forces are calculated for the full periodic system while the magnetic anisotropy tensors are computed for isolated molecules cut from them; see Figure 1. That implies that both atomistic systems provide different sets of input features and, therefore, cannot directly be combined. For this reason, in the following sections, we describe the

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

construction of two separate ML models for learning **D** tensors and potential energy surfaces.

2.2. D Tensor. Mediated by spin−orbit coupling, the components of spin multiplets (with spin quantum numbers $S \geq 1$) split characteristically even in the absence of an external magnetic field (zero-field splitting, ZFS). This effect is usually described by a phenomenological spin Hamiltonian[36]

$$H_{\mathrm{ZFS}} = \hat{\mathbf{S}} \cdot \mathbf{D} \cdot \hat{\mathbf{S}} \tag{4}$$

where $\hat{\mathbf{S}}$ is a (pseudo) spin operator and **D** is a 3 × 3 symmetric, traceless tensor, usually called a ZFS tensor or a **D** tensor. The anisotropy parameters D and E (often simply referred to as D and E values, respectively) can be then obtained from the eigenvalues of **D** (X, Y, and Z) as

$$D = \frac{3}{2} Z$$
$$E = \frac{1}{2}(X - Y) \tag{5}$$

where Z corresponds to the eigenvalue of **D** with the largest absolute value. The normalized eigenvectors of **D** are the anisotropy axes of the system.

2.2.1. Incorporation of Symmetries into the ML Framework. The aim of this work is to predict **D** tensors using artificial neural networks (NNs). The **D** tensor is a (**1**) traceless (**2**) symmetric tensor, which implies that $\mathrm{Tr}\,\mathbf{D} = \sum_i D_{ii} = 0$ and $D_{ij} = D_{ji}$. Additionally, it (**3**) transforms under rotation as $\tilde{\mathbf{D}} = \mathbf{R}\mathbf{D}\mathbf{R}^{\mathrm{T}}$, where **R** is an orthogonal matrix, but is (**4**) invariant to translations. These properties have to be encoded into the respective machine learning model to allow for efficient training. Moreover, unlike the energy, the **D** tensor cannot be reduced to a natural scalar atomic property which makes its prediction more difficult using the atomistic neural networks.

To the best of our knowledge, there are only two studies in the literature of modeling tensorial properties such as **g** and **D** tensors.[33,34] In ref 33, where ridge regression was used to fit **D** tensors, the rotational equivariance was imposed by requiring the regression coefficients to transform as spherical tensors, i.e., employing the sum over the Wigner matrices corresponding to rigid rotations. In ref 34, where artificial neural networks were used to fit **g** tensors, a rotation operator Λ was defined by, e.g., the Kabsh algorithm to impose the desired symmetries. In this work, we present a more rigorous and efficient way of encoding the symmetries (**1**)−(**4**) into machine learning algorithms.

We approach the modeling of zero-field splitting tensors by introducing a fictitious atomic quantity m_i assigned to each atom i. A tensor which satisfies (**1**)−(**3**) can be defined using the tensor product of Cartesian vectors, similar to the quadrupole moment, employing the aforementioned atomic quantity m_i

$$\mathbf{D}(\mathcal{S}, \boldsymbol{\theta}) = \sum_{i=1}^{N_{\mathrm{at}}} m_i(\mathbf{G}_i, \boldsymbol{\theta}) \cdot (3\mathbf{r}_i \otimes \mathbf{r}_i - \|\mathbf{r}_i\|_2^2 \mathbb{I}_3) \tag{6}$$

where ⊗ is the tensor product, $\|\mathbf{r}_i\|_2$ is the length of the respective Cartesian vector, and $\mathbb{I}_3$ is a 3 × 3 identity matrix.

Unfortunately, the above expression violates translational invariance (**4**) of the respective tensorial property. This issue can be resolved by shifting the coordinate system by, e.g., $\bar{\mathbf{r}} = 1/N_{\mathrm{at}} \sum_i \mathbf{r}_i$, which is system-dependent. Note that the shift can be defined by any other procedure since its aim is only to impose the translational invariance (**4**). However, we should emphasize that we have found that using the pairwise distances $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$, where i can be selected to be the metal atom, could be disadvantageous in terms of predictive accuracy due to missing contributions from the central atom.

Using the respective shift vector $\bar{\mathbf{r}}$ results in a new position vector $\mathbf{r}_i \rightarrow \mathbf{r}_i - \bar{\mathbf{r}}$. Additionally, we have found that it is advantageous to use the normalized vectors $\hat{\mathbf{r}}_i = \mathbf{r}_i / \|\mathbf{r}_i\|_2$. In total, the resulting expression reads

$$\mathbf{D}(\mathcal{S}, \boldsymbol{\theta}) = \sum_{i=1}^{N_{\mathrm{at}}} m_i(\mathbf{G}_i, \boldsymbol{\theta}) \cdot (3\hat{\mathbf{r}}_i \otimes \hat{\mathbf{r}}_i - \mathbb{I}_3) \tag{7}$$

where m_i is predicted by an atomistic NN.

As a final remark, we want to point out that any symmetry of a tensorial property **P** can be modeled by the procedure proposed above if one defines $\mathbf{P} = \sum_i m_i \mathbf{A}_i$, where $\mathbf{A}_i$ is a tensor satisfying the symmetry of **P** and m_i is a machine-learned scalar value. In our example, we define $\mathbf{A}_i = 3\hat{\mathbf{r}}_i \otimes \hat{\mathbf{r}}_i - \mathbb{I}_3$ to impose (**1**)−(**4**), but if the respective property is, e.g, not traceless one could use $\mathbf{A}_i = \hat{\mathbf{r}}_i \otimes \hat{\mathbf{r}}_i$.

2.2.2. Network Architecture and Training. We use a fully connected feed-forward neural network consisting of two hidden layers of the following functional form, similar to our previous work[30]

$$y_i(\mathbf{G}_i, \boldsymbol{\theta}) = 0.1 \cdot \mathbf{b}^{(3)} + \frac{1}{\sqrt{d_2}} \mathbf{W}^{(3)} \phi\left(0.1 \cdot \mathbf{b}^{(2)} + \frac{1}{\sqrt{d_1}} \mathbf{W}^{(2)} \phi\left(0.1 \cdot \mathbf{b}^{(1)} + \frac{1}{\sqrt{d_0}} \mathbf{W}^{(1)} \mathbf{G}_i\right)\right) \tag{8}$$

where $\mathbf{W}^{(l)}$ and $\mathbf{b}^{(l)}$ are weights and biases of the respective layer l, respectively. As an input to the neural network, $\mathbf{G}_i$, we use the recently proposed trainable local invariant representation based on Gaussian moments.[24,30] The parameters 0.1 and $1/\sqrt{d_l}$ correspond to the so-called NTK parameterization.[37] We initialize weights of the fully connected part by drawing the respective entries from a normal distribution with zero mean and unit variance. The trainable bias vectors are initialized to zero. As an activation function, we use the Swish/SiLU activation function[38−40] $\phi(x) = \alpha x/(1 + \exp(-x))$ multiplied by a scalar α. We choose $\alpha \approx 1.6765$ such that $\mathbb{E}_{x \sim \mathcal{N}(0,1)} \phi(x)^2 = 1$, i.e., the activation function preserves the second moment if the input is standard Gaussian.[41−43]

To aid the training process, the output of the neural network can be scaled and shifted by the standard deviation σ and the mean μ of the reference $\mathbf{D}^{\mathrm{ref}}$ tensor values, similar to the scaling and shifting the atomic energy output.[24,30] Note that we used for the computation of σ and μ only those elements of D_{ij}, which satisfy $i \geq j$ to avoid double counting and excluded one diagonal element since the respective tensor is traceless. The convergence of the model can be improved even further by making these parameters trainable as well as dependent on the atomic species, i.e., σ_{Z_i} and μ_{Z_i}. The final output of the network reads

$$m_i(\mathbf{G}_i, \boldsymbol{\theta}) = y_i(\mathbf{G}_i, \boldsymbol{\theta}) \sigma_{Z_i} + \mu_{Z_i} \tag{9}$$

To train the neural network on reference values for $\mathbf{D}^{\mathrm{ref}}$ tensors, we minimize the following loss function

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

$$\mathcal{L}_{\mathbf{D}}(\boldsymbol{\theta}) = \sum_{k=1}^{N_{\text{Train}}} \|\mathbf{D}_k^{\text{ref}} - \mathbf{D}(\mathcal{S}_k, \boldsymbol{\theta})\|_2^2 \tag{10}$$

to optimize the respective parameters of the trainable representation, fully connected neural network part as well as the parameters, which scale and shift the output of the neural network. Note that we train only on those elements of $(\mathbf{D})_{ij}$ tensor that satisfy $i \geq j$ and we define $\mathbf{D}^{\text{ref}}$ and $\mathbf{D}$ as

$$\mathbf{D}^{\text{ref}} = (D_{11}^{\text{ref}}, D_{12}^{\text{ref}}, D_{13}^{\text{ref}}, D_{22}^{\text{ref}}, D_{23}^{\text{ref}}, D_{33}^{\text{ref}})^{\text{T}}$$
$$\mathbf{D} = (D_{11}, D_{12}, D_{13}, D_{22}, D_{23}, -(D_{11} + D_{22}))^{\text{T}} \tag{11}$$

To minimize the loss function in eq 10, the Adam optimizer[44] with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$, and a mini-batch of 32 structures is employed. Moreover, we allow for layer-wise learning rates, which decay linearly to zero by multiplying them with $(1 - r)$, where r = step/max_step. Overall, we use an initial learning rate of 0.02 for the parameters of the fully connected layers, 0.02 for the parameters of the trainable GM representation, 0.025 and 0.025 for the shift and scale parameters.

Throughout this work, we used an architecture with an input dimension of $d_0 = 360$, two hidden layers with $d_1 = d_2 = 512$ hidden neurons, and an output layer that has a single $d_3 = 1$ output neuron. Each model in Section 4 is trained for 1000 epochs. Overfitting was prevented using the early stopping technique.[45] After each epoch, the mean absolute errors (MAE) of the $\mathbf{D}$ tensor were evaluated on the validation set. After training, the model with the minimal MAE on the validation set was selected for further applications. While the selected hyperparameters worked reasonably well on the selected systems for us, we want to emphasize that other trade-offs between the number of training epochs and the initial learning rates can be achieved.

Note that to compute machine-learned $\mathbf{D}$ tensors during an MD simulation we interfaced our approach with the ASE package (v3.21.0).[46] For tracking the accuracy, we employed the query-by-committee (QbC) approach[47] during MD simulations. For this purpose, we trained a committee of three models on the same split of the data set but using randomly initialized parameters and reported the obtained uncertainty

$$\sigma_{\text{ens}}(\mathcal{S}) = \sqrt{\frac{1}{N_{\text{ens}}} \sum_{i=1}^{N_{\text{ens}}} (y_i(\mathcal{S}) - \bar{y}(\mathcal{S}))^2} \tag{12}$$

where N_{ens} is the number of models in the committee, i.e., $N_{\text{ens}} = 3$. $\bar{y}(\mathcal{S}) = 1/N_{\text{ens}} \sum_{i=1}^{N_{\text{ens}}} y_i(\mathcal{S})$ is the mean of the property prediction (energy, atomic force element, or $\mathbf{D}$ tensor element, respectively) over the committee. All models were trained within the Tensorflow framework[48] on an NVIDIA Tesla V100-SXM-32GB GPU. The training of an ensemble of three models for 1000 epochs took from 4 min (100 $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ structures) to 3 h (2900 $[Ni(HIM_2{-}py)_2NO_3]^+$ structures).

2.3. Potential Energy. Thermal averaging of magnetic anisotropy tensors requires an interatomic potential that fulfills two main premises. The underlying model has to produce sufficiently accurate atomic forces for molecular dynamics (MD) simulations, i.e., comparable to the level of theory employed for the generation of the training data, and allow for efficient computations, i.e., comparable to empirical force fields. Machine learning algorithms have found a broad application in computational chemistry since they satisfy both conditions provided a suitable molecular descriptor that encodes structural and alchemical information.

As for the $\mathbf{D}$ tensors, we employed the Gaussian moment neural network (GM-NN) approach[24,30] to construct the PES. Again, we use a neural network with an input dimension of $d_0 = 360$, two hidden layers with $d_1 = d_2 = 512$ hidden neurons, and an output layer, which has a single $d_3 = 1$ output neuron.

To train the GM-NN model the combined loss function

$$\mathcal{L}_{\mathcal{E},\mathbf{F}}(\boldsymbol{\theta}) = \sum_{k=1}^{N_{\text{Train}}} \left[\lambda_{\mathcal{E}} \|\mathcal{E}_k^{\text{ref}} - \mathcal{E}(\mathcal{S}_k, \boldsymbol{\theta})\|_2^2 + \frac{\lambda_F}{3N_{\text{at}}^{(k)}} \sum_{i=1}^{N_{\text{at}}^{(k)}} \|\mathbf{F}_{i,k}^{\text{ref}} - \mathbf{F}_i(\mathcal{S}_k, \boldsymbol{\theta})\|_2^2 \right] \tag{13}$$

is minimized. Here, $N_{\text{at}}^{(k)}$ is the number of atoms in the respective structure. The reference values for the energy and atomic force are denoted by $\mathcal{E}_k^{\text{ref}}$ and $\mathbf{F}_{i,k}^{\text{ref}}$, respectively. The parameters $\lambda_{\mathcal{E}}$ and λ_{F} were set to 1 au and $12N_{\text{at}}^{(k)}$ au Å^2, respectively. The network was trained using the Adam optimizer[44] with 32 molecules per mini-batch. The layer-wise learning rate was set to 0.03 for the parameters of the fully connected layers, 0.02 for the trainable representation, 0.05 and 0.001 for the shift and scale parameters of atomic energies, respectively. We allowed all learning rates to decay linearly to zero. For more information, see ref 30.

To run MD simulations with the machine-learned potentials (MLPs), we interfaced the GM-NN approach with the ASE package (v3.21.0).[46] For tracking the accuracy of MLPs, we employed the query-by-committee (QbC) approach[47] during MD simulations. For this purpose, we trained a committee of three models on the same split of the data set but using randomly initialized parameters and reported on the attained uncertainty; see eq 12. All models were trained within the Tensorflow framework[48] on an NVIDIA Tesla V100-SXM-32GB GPU. The training of an ensemble of three models for 1000 epochs took about 40 h (3100 $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ periodic structures).

3. TEST SYSTEMS AND COMPUTATIONAL DETAILS

In this section, we describe the generation of the data used to construct machine-learned interatomic potentials as well as machine learning models for $\mathbf{D}$ anisotropy tensors and an overview of selected test systems.

3.1. Test System Description. We selected the following three promising candidates for SMMs (in fact even single-ion magnets, SIMs): $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$,[6] $[Fe(tpa)^{Ph}]^-$,[7,35] and $[Ni(HIM_2{-}py)_2NO_3]^+$.[8] Figure 1 illustrates the unit cells of the systems and the corresponding complexes used in the cluster models.

3.1.1. $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$. One of the most promising high-anisotropy SIMs has been reported by Rechkemmer et al.[6] It has a single cobalt ion center that is bound to two doubly deprotonated 1,2-bis(methanesulfonamido)benzene ligands, resulting in a distorted tetrahedral coordination sphere. In the crystal, the charge is compensated by two $NHEt_3^+$ cations. The zero-field splitting parameter $D^{\text{exp}} = -115 \pm 20$ cm^{-1} has been experimentally determined by fitting to AC and DC susceptibility and magnetic hysteresis measurements.[6]

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

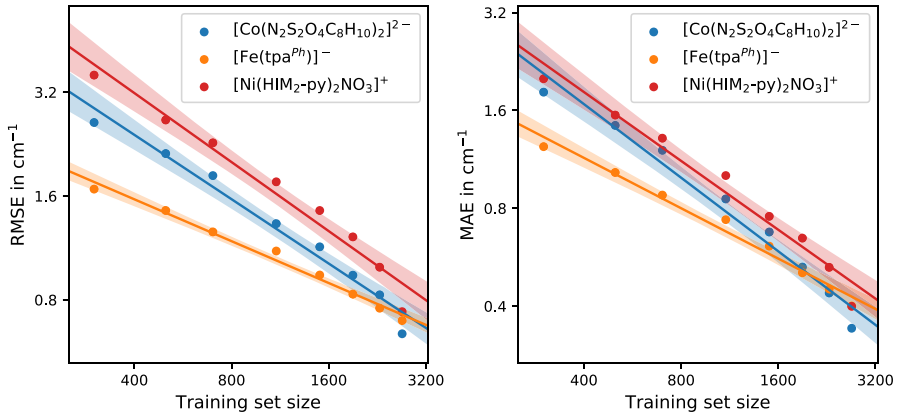


Figure 2. Learning curves for the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, $[Fe(tpa)^{Ph}]^-$, and $[Ni(HIM_2{-}py)_2NO_3]^+$ data sets. The root-mean-squared errors (RMSE) and the mean absolute errors (MAE) of the **D** tensor components are plotted against the training set size. Linear fits are displayed for clarity, and shaded areas denote the 95% confidence intervals for linear regression.

3.1.2. [Fe(tpa)^{Ph}]$^-$. This is one example out of a big family of similar complexes which all feature an iron center ion in trigonal-pyramidal surrounding of a pyrrolide ligand tpa^R.[7] The complex studied here has phenyl attached to the main pyrrolide ligand (R = Ph). The counterion in the crystal is $Na^+(H_3COC_2H_4OCH_3)_3$. Experimentally the magnetic parameters have been determined to be $D^{exp} = -26 \pm 2$ cm^{-1} and $E^{exp} = 5$ cm^{-1} by AC and DC magnetic susceptibility measurements and fitting procedures.[7] This family of complexes was also theoretically studied by Atanasov et al. on the level of complete active space self-consistent field (CASSCF) wave functions in conjunction with *N*-electron valence perturbation theory (NEVPT2) and quasi-degenerate perturbation theory (QDPT).[35]

3.1.3. [Ni(HIM$_2$−py)$_2$NO$_3$]$^+$. One example of a nickel complex with a high magnetic anisotropy has been synthesized by Rogez et al.[8] The nickel ion is influenced by a highly distorted octahedral coordination sphere consisting of two bidentate ligands HIM$_2$−py as well as an O,O′-chelating nitrate ligand. In the crystal, the positive charge is compensated by a second NO_3^- ion without direct contact with the nickel ion. Measurements of the magnetization versus field, HF-HFEPR, and FDMRS resulted in fitted magnetic parameters $D^{exp} = -10.1 \pm 0.1$ cm^{-1} and $E^{exp} = 0.202 \pm 0.01$ cm^{-1}.[8]

3.2. Data Generated via Ab Initio Molecular Dynamics (AIMD). For all training data sets, an ab initio molecular dynamics (AIMD) calculation has been carried out using the projector augmented wave (PAW) method[49,50] with the Perdew−Burke−Ernzerhof (PBE) functional,[51] as implemented in the VASP program package.[52−54] A Hubbard *U* correction term has been used with the simplified (rotationally invariant) approach introduced by Dudarev et al.[55] The used values of *U* are 3.3 eV for Co, 4.0 eV for Fe, and 6.4 eV for Ni.[56] Dispersion corrections were applied by the zero damping DFT-D3 method.[57]

As a starting point, the crystal structure was first optimized using an energy cutoff for the plane-wave basis of 600 eV and the "accurate precision" settings in VASP, the projection operators were evaluated in real space. The Brillouin zone was sampled by a Monkhorst−Pack grid (Co system: 2 × 2 × 2; Fe system: 3 × 3 × 2; Ni system: 4 × 4 × 2). The stress tensor was calculated, and all degrees of freedom were allowed to change in relaxation.

After that, we performed an AIMD simulation with VASP, using the same settings as before but with a cutoff energy of 400 eV and the "normal precision" settings. Only the Γ point was considered. For each system shown in Figure 1, we calculated 5000 time steps of 1 fs for the temperatures 100, 300, 400, 450, and 500 K each using a Nosé−Hoover thermostat as implemented in VASP.

We chose more structures from the MD runs at higher temperatures, i.e., 200, 600, 800, 900, and 1000 sample structures randomly chosen from the MD runs at 100, 300, 400, 450, and 500 K, respectively. This was done because more diverse conformations are visited at higher temperatures and therefore the dynamics performed at a higher temperature contains more information necessary for the construction of reliable interatomic potentials.

3.3. D Tensor Data. To generate reference values for the **D** tensor on which models were subsequently trained, we cut the corresponding molecules from the periodic structures of the AIMD simulation, as illustrated in Figure 1. Effects of the neighboring molecules and the crystal field on the **D** tensor have thus been neglected for the present study. In total, we obtained 3500 configurations for each test system, for which magnetic properties were calculated using the Molpro program package.[58] The orbitals were optimized by the configuration-averaged Hartree−Fock (CAHF) procedure[59−62] using the Karlsruhe def2-SVP basis sets.[63] The active space consisted of the five d orbitals in each of the systems (see below). Based on these orbitals, a complete active space configuration interaction (CASCI) calculation was performed to obtain the spin-free states. These were afterward used in a spin−orbit configuration interaction (SO-CI) calculation, using a mean-field spin−orbit operator,[64−66] based on the CAHF average density for the mean field.

The following setup is used for each of the test systems: For $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, we chose a CAS(7,5) and the SO-CI calculation was carried out using 40 doublet and 10 quartet CASCI states; for $[Fe(tpa)^{Ph}]^-$, we chose a CAS(6,5) and the SO-CI calculation was based on 50 singlet, 45 triplet, and 5 quintet CASCI states; and for $[Ni(HIM_2{-}py)_2NO_3]^+$, we finally chose a CAS(8,5) and the SO-CI calculation used 15 singlet and 10 triplet CASCI states.

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

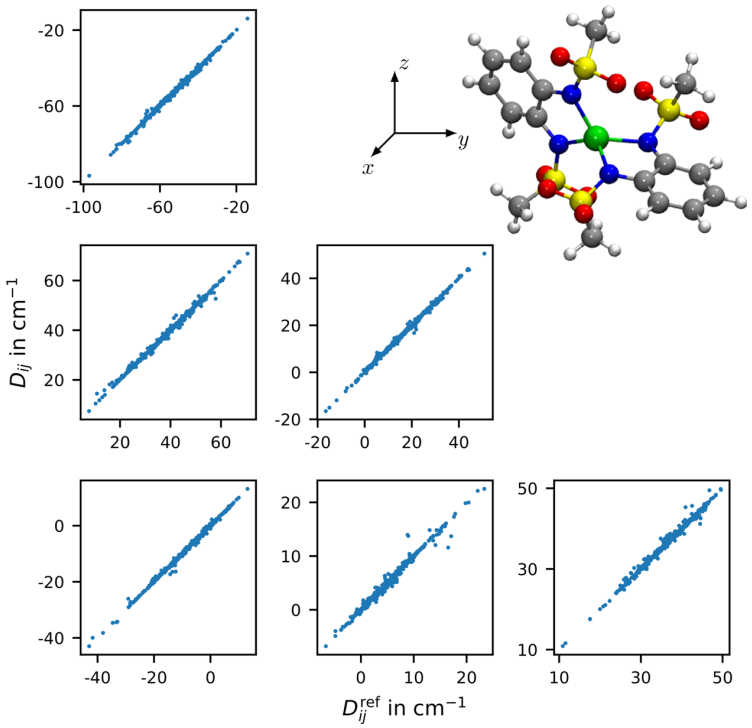


Figure 3. Correlation of the machine-learned symmetric elements ($i \geq j$) of the zero-field splitting tensor (D_{ij}) with the corresponding reference values (D_{ij}^{ref}) for all structures in the test $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ data. For all predictions, the model trained on 2900 reference structures was used. The respective coordinate system of the periodic box as well as an example substructure for which **D** tensor was computed are shown as an inset.

From the SO-CI calculations, the **D** tensor was extracted using the pseudo-spin procedure described by Chibotaru and Ungur.[67]

4. RESULTS AND DISCUSSION

Here, we apply the proposed approach for machine learning symmetric traceless tensors, namely, **D** tensors, to systems described in Section 3. For the example of $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, we demonstrate the applicability of our approach to studying relevant magnetic properties of SMMs.

4.1. Machine Learning of the D Tensor. To study the performance of the proposed approach for machine learning magnetic anisotropy tensors, or specifically **D** tensors, we trained our model on $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, $[Fe(tpa)^{Ph}]^-$, and $[Ni(HIM_2{-}py)_2NO_3]^+$ data. Each data set contained structures and **D** tensor elements of 3500 configurations. In Figure 2, we report the mean absolute error (MAE) and the root-mean-square error (RMSE) in cm^{-1}. The cutoff radius employed in the definition of invariant atomic representation was set to 8.0 Å. All other hyperparameters are described in Section 2.2.2.

Figure 2 reports the MAE and RMSE of **D** tensor predictions as a function of the number of training samples. For all training set sizes, we have randomly drawn 300 additional structures as validation data to track overfitting during training. Since the validation data indirectly influence the selected set of trainable parameters, all values presented in Figure 2 are obtained for the test data that have not been seen during training. For example, for 2900 training data, we have used the remaining 300 structures to test the model, while for 1500 training data, 1700 structures were used for the same purpose.

In general, we notice that the proposed approach leads to models that learn quite efficiently on the reference data. For example, the RMSE for the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ data set is reduced by a factor of 2 when doubling the training data set size. It can be observed that the RMSE for $[Ni(HIM_2{-}py)_2NO_3]^+$ is slightly higher compared to other systems, while $[Fe(tpa)^{Ph}]^-$ shows lower RMSE values for smaller training data set sizes. The former observation can be explained by the higher flexibility of $[Ni(HIM_2{-}py)_2NO_3]^+$. This leads to a larger conformational space sampled during ab initio molecular dynamics (AIMD) in Section 3.2 and, as a result, to a broader range of **D** tensor elements; see Figure S2. For $[Fe(tpa)^{Ph}]^-$, the situation is different since structural differences lead only to a slight variation in **D** tensor elements; see Figure S1, compared to other systems, Figures 3 and S2.

A direct comparison to previous models dealing with machine learning magnetic anisotropy tensors, e.g., refs 33 and 34, is impossible since no standard benchmark is available in the literature, in contrast to, e.g., QM9[68,69] and MD17[70−72] data sets for testing machine-learned interatomic potentials. Therefore, we use the presented data sets, available free-of-charge from ref 73, to benchmark the models developed for predicting **D** tensors.

In previous work,[33] for $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, an RMSE value of 2.2 cm^{-1} was obtained when training on 900 structures, while we obtain 1.5 cm^{-1}. In ref 33, the training data set was generated starting from an optimized structure in a vacuum and then displacing atoms by a maximum of ±0.05 Å (500 structures), ±0.1 Å (500 structures), and ±0.2 Å (500 structures), while we extracted the conformations from an

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

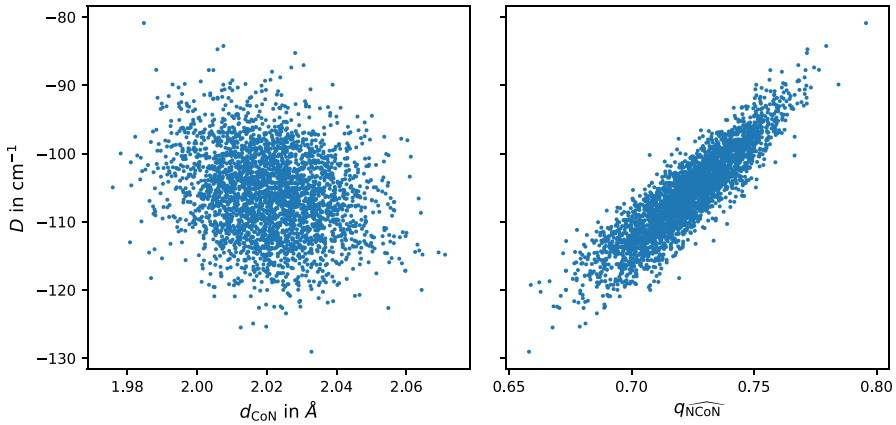


Figure 4. Correlation of the magnetic axial anisotropy $D = 3/2Z$, where Z is the largest eigenvalue of the **D** tensor, with (left) the average Co−N distance d_{CoN} and (right) tetrahedral order parameter $q_{\widehat{NCoN}}$. The latter yields a value of 1 for the perfect tetrahedral structure, 0.5 for a perfect quadratic planar structure, and 0 for a random mutual arrangement of central and neighboring atoms. The tetrahedral order parameter $q_{\widehat{NCoN}}$ yields a linear correlation coefficient of 0.88, while the average Co−N distance does not correlate with D and yields a linear correlation coefficient of −0.22.

AIMD simulation. Thus, we expect that our data sets cover a broader conformational space, making the training more difficult (which would lead to higher MAE/RMSE values) but facilitating the prediction of properties for out-of-sample configurations.

In Figure 3, we show the correlation of the reference and machine-learned results of each D_{ij} element in the **D** tensor for the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ system. We again use only those structures the machine learning model has not seen during training. From Figure 3, we clearly see a perfect correlation between both values in accordance with the low MAE (RMSE) value of 0.30 cm^{-1} (0.58 cm^{-1}), obtained by training the model on 2900 reference structures. Similar results were obtained for $[Fe(tpa)^{Ph}]^-$ and $[Ni(HIM_2{-}py)_2NO_3]^+$; see Figures S1 and S2. In total, taking into account the broadly sampled conformational space and an excellent out-of-sample predictive accuracy, our approach should be applicable to large-scale molecular dynamics simulations.

4.2. Dynamics of the D Tensor. We assess the quality of machine-learned **D** tensors by applying the machine learning methodology to the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ system to study its magnetic/physicochemical properties. While a detailed analysis is out of the scope of this paper, we provide merely a qualitative overview to demonstrate the broad applicability of our approach.

To analyze the properties of $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, we ran molecular dynamics (MD) simulations in the canonical (NVT) statistical ensemble, carried out within the ASE simulation package[46] using a Langevin thermostat at the temperatures of 25, 50, 75, and 100 K. All MD runs were performed over 2.5 ns using a time step of 0.5 fs. The atomic velocities were initialized with a Maxwell−Boltzmann distribution for the temperatures of 25, 50, 75, and 100 K, respectively.

Forces for molecular dynamics were generated by an ensemble of three machine-learned interatomic potentials; see Section 2.3. To train the machine-learned interatomic potentials, a cutoff radius of 6.5 Å was employed. The ensembling technique provides us with an error estimate of the potential during simulation. We have found the machine-learned potential to be very accurate with the uncertainty between models ranging from 0.10 ± 0.06 to 0.15 ± 0.10 kcal/(mol Å); see Figure S3. For **D** tensor predictions, we also use an ensemble of three models with an uncertainty ranging from 0.11 ± 0.06 to 0.20 ± 0.11 cm^{-1}; see Figure S3. These values allow us to claim that our machine-learned models are suitable for the following analysis.

4.2.1. Dependence of the D Tensor on the Structure. To study the structural dependence of the **D** tensor, we evaluated the average Co−N distance d_{CoN} and the tetrahedral order parameter $q_{\widehat{NCoN}}$ as[74]

$$d_{CoN} = \frac{1}{4}\sum_{i=1}^{4} d_i$$

$$q_{\widehat{NCoN}} = 1 - \frac{3}{8}\sum_{i=1}^{3}\sum_{j=i+1}^{4}\left(\cos(\theta_{ij}) + \frac{1}{3}\right)^2 \quad (14)$$

where d_i and θ_{ij} are the distance and angle formed by the metal center and its neighboring nitrogen atoms, respectively. Note that in the literature usually the average angle between nitrogen atoms belonging to the same ligand is used as an order parameter,[33] or its deviation from a perfect tetrahedral angle $T_d = 109.5°$ is discussed.[75] We have employed the tetrahedral order parameter $q_{\widehat{NCoN}}$ often used when studying the structure of liquid water.[76] This parameter contains the information about angular distribution but is rescaled in such a way that its value varies between 0 (if the arrangement of all atoms is random) and 1 (in a perfect tetrahedral network). Moreover, it has a value of 0.5 for a perfect quadratic planar configuration, which allows us to look into both possible configurations of the first coordination sphere of $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, tetrahedral and quadratic planar.

Figure 4 shows the correlation of the magnetic axial anisotropy $D = 3/2Z$, where Z is the eigenvalue of the **D** tensor with the largest absolute value, with the order parameters presented above. Data obtained from MD at 100 K are shown since they provide the broadest range of conformations and the broadest range of D values. From Figure 4, we see that the average Co−N distance d_{CoN} correlates only marginally with the magnetic axial anisotropy D for which we obtained a linear correlation coefficient of

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

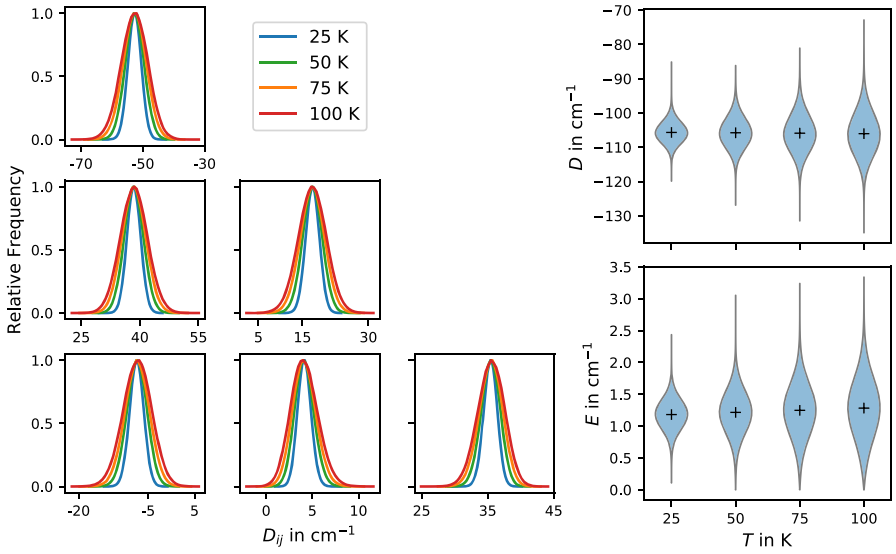


Figure 5. (Left) Thermal broadening of the elements D_{ij} with $i \geq j$ of the zero-field splitting tensor **D** obtained from sampling configurations over 2.5 ns long molecular dynamics. All values were computed employing the machine learning model described in Section 2.2. For all predictions, the model trained on 2900 reference structures was used. (Right) Temperature dependence of $D = 3/2Z$ and $E = |X - Y|/2$, where X, Y, and Z are the eigenvalues of the **D** tensor, with Z being its largest absolute eigenvalue.

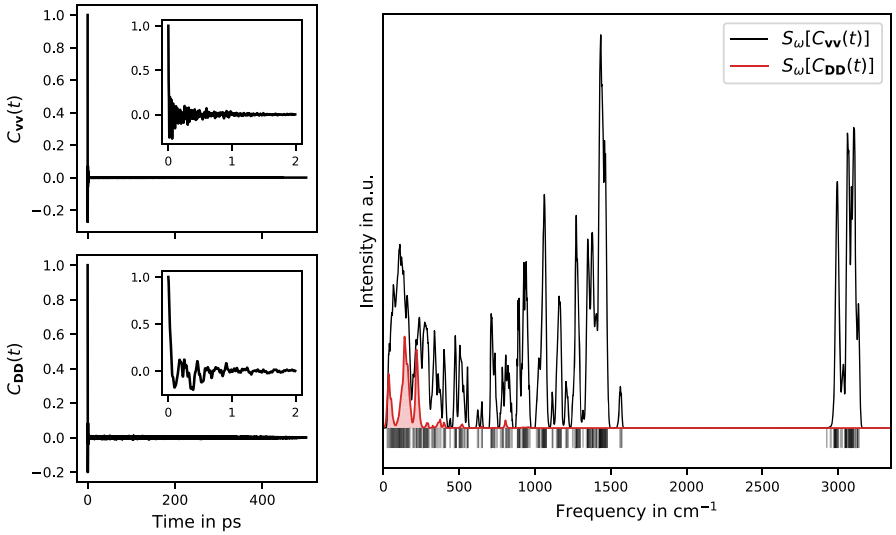


Figure 6. (Left) Velocity and **D** tensor autocorrelation functions, $C_{vv}(t)$ and $C_{DD}(t)$, respectively, obtained by averaging over five independent MD trajectories at 50 K of a length of 500 ps. (Right) Power spectral density function S_ω of the velocity and **D** tensor autocorrelation functions. The gray vertical lines at the bottom of the graph correspond to harmonic frequencies.

−0.22. This is in agreement with the results found in ref 33 taking into consideration that a minimal average Co−N distance of about 1.98 Å could be sampled at 100 K. Note that in ref 33, a stronger correlation was found for distances less than 2.0 Å. However, from our MD simulations, we see that such configurations are too rare at 100 K to be taken into account.

Figure 4 clearly shows a strong dependence of the magnetic axial anisotropy D on the tetrahedral order parameter $q_{\widehat{\mathrm{NCoN}}}$ for which we obtained a linear correlation coefficient of 0.88. When changing the structure from tetrahedral coordination to a quadratic planar one, a strong increase of the magnetic anisotropy is observed. Our results are in perfect agreement with recent results that suggest that the $\widehat{\mathrm{NCoN}}$ angle represents the main path for improving Co^{2+} single-ion magnets.[6,33,77−80]

4.2.2. Thermal Distribution of the D Tensor. Besides the correlation of the magnetic axial anisotropy D with structural order parameters d_{CoN} and $d_{\widehat{\mathrm{NCoN}}}$, we study the variation in magnitude and orientation due to thermal fluctuations in the **D** anisotropy tensor. Figure 5 (left) shows the distribution of the individual D_{ij} components of the **D** tensor sampled over 2.5 ns. It can be seen that all elements D_{ij} are approximately normally distributed with broader distributions for higher temperatures. The elements are symmetric with $D_{ij} = D_{ji}$; hence, only D_{ij} for $i \geq j$ are shown. It should be noted that we have found that the

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

mean of the distribution of each element D_{ij} remains almost unchanged and the maximal deviation equals 0.1 cm^{-1}. However, the distribution of each element D_{ij} is broadened by approximately a factor of 2 when increasing the temperature from 25 to 100 K.

To relate the dynamics of the magnetic axial anisotropy $D = 3/2Z$ to the structural dynamics of the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex, we display the respective distribution in Figure 5 (right) along with the distribution for $E = |X - Y|/2$. Note that X, Y, and Z are the eigenvalues of $\mathbf{D}$, with Z being the one with the largest absolute value. We have found the values for D to range from -105.7 ± 3.1 to -106.0 ± 6.2 cm^{-1}, i.e., the mean is almost temperature-independent while the standard deviation is doubled when increasing the temperature by a factor of 4 in accordance with results for D_{ij}. Note that our values estimated from MD simulations with machine learning are very close to the experimental one of -115 ± 20 cm^{-1}.[6] The values for E range from 1.2 ± 0.2 to 1.3 ± 0.5 cm^{-1}.

4.2.3. Time Correlation Functions and Spin−Phonon Coupling. Finally, and mainly as an outlook to future applications of our approach, we study the coupling of the $\mathbf{D}$ tensor of the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex to the dynamics of the periodic atomic structure (velocity vectors $\mathbf{v} = \dot{\mathbf{x}}$). For this purpose, we define time-dependent $\mathbf{D}$ tensor and velocity autocorrelation functions (ACFs) as

$$C_{\mathbf{DD}}(t) = \frac{1}{6}\sum_{i=1}^{3}\sum_{j\geq i}^{3}\frac{\langle D_{ij}(0)D_{ij}(t)\rangle}{\langle D_{ij}(0)D_{ij}(0)\rangle}$$

$$C_{\mathbf{vv}}(t) = \frac{1}{3N_{\mathrm{at}}}\sum_{i=1}^{N_{\mathrm{at}}}\sum_{j=1}^{3}\frac{\langle v_{ij}(0)v_{ij}(t)\rangle}{\langle v_{ij}(0)v_{ij}(0)\rangle} \quad (15)$$

respectively. Using these ACFs, it is possible to compute the corresponding spectra by performing a Fourier transform or by employing the maximum-entropy approach.[81−83] In this work, we compute the power spectral density function S_ω of interest employing the `memspectrum` package.[84]

Figure 6 shows the velocity and $\mathbf{D}$ tensor ACFs (left) as well as the respective power spectral density functions $S_\omega[C_{\mathbf{vv}}(t)]$ and $S_\omega[C_{\mathbf{DD}}(t)]$ (right). From the latter, in principle, the spin−phonon coupling coefficients $\partial D/\partial q_\alpha$ beyond the harmonic approximation can be calculated, which provide the interaction strength between the spin and the atomic movements.[85] The present study is restricted to the demonstration of the capability of the proposed approach, a detailed analysis will be presented in future work. For now, we computed the spectra for the Γ-point only, while the inclusion of the full Brillouin zone (k-dependence) may be important to estimate the spin lifetime. Even with our preliminary spectra, we are able to deduce similar conclusions as previous work[85] found for $[Fe(tpa)^{Ph}]^-$ in that the most important vibrations in the spin−phonon relaxation process are the low-energy ones. They are predominantly populated under typical experimental conditions.

5. CONCLUSIONS

In this work, we have presented a machine learning approach based on Gaussian moments[24,30] for tensorial properties on the example of the zero-field splitting ($\mathbf{D}$) tensor. It enables an efficient prediction and modeling of magnetic properties of single-molecule magnets. The presented approach was extensively tested on three systems $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, $[Fe(tpa)^{Ph}]^-$, and $[Ni(HIM_2{-}py)_2NO_3]^+$ in terms of its predictive accuracy and its reliability during real-time simulations.

Training the proposed model on the respective reference $\mathbf{D}$ values for the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$, we have observed an improved accuracy compared to previous studies,[33] especially taking into account the larger sampled configurational space. In total, for all systems tested in this work, we could achieve an MAE of 0.3−0.4 cm^{-1} and an RMSE of 0.6−0.7 cm^{-1} for the models trained on 2900 reference structures. Moreover, we have shown that the proposed approach, once trained on a sufficiently large configurational space, is able to predict $\mathbf{D}$ tensor values for millions of conformations not seen before with a negligibly small uncertainty of 0.11−0.20 cm^{-1} obtained by employing the query-by-committee approach. This demonstrates the excellent generalization capability of our approach.

In combination with machine-learned interatomic potentials, we were able to run 2.5 ns long molecular dynamics simulations at temperatures of 25, 50, 75, and 100 K. Using the respective trajectories, we could analyze several properties of the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex taken as an example. Analyzing the dependence of the magnetic axial anisotropy on average Co−N distance and an $\widehat{\mathrm{NCoN}}$ angle-dependent order parameter $q_{\widehat{\mathrm{NCoN}}}$, we have found that the $\widehat{\mathrm{NCoN}}$ angle represents the main path for improving Co^{2+} single-ion magnets, in accordance with recent results.[6,33,77−80] Moreover, we could estimate the thermal average of D and E values. For the former, a very good agreement with the experiment has been found, while for the latter, no experimental data are available.

Besides the structure, we investigated the dynamic behavior of the $\mathbf{D}$ tensor of the $[Co(N_2S_2O_4C_8H_{10})_2]^{2-}$ complex. Even in such preliminary work, we observed the expected behavior that the low-energy vibrations are important for the spin−phonon relaxation process.

In summary, our developments aim to provide an alternative way for the efficient modeling of magnetic properties of single molecular magnets via machine learning. While the current setup is based on a relatively simple complete active space configuration interaction treatment of the molecular properties, our approach can be easily extended to more elaborate methods, e.g., using transfer learning. Future work will furthermore deal with an application of the proposed methodology to allow for a detailed analysis of spin−phonon relaxation processes.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.1c00853.

Correlation of the machine-learned symmetric elements of the zero-field splitting tensor (D_{ij}) with the corresponding reference values (D_{ij}^{ref}) for all structures in the test sets $[Fe(tpa)^{Ph}]^-$ and $[Ni(HIM_2{-}py)_2NO_3]^+$; uncertainty distributions for atomic forces and $\mathbf{D}$ tensor by the query-by-committee (QbC) approach (PDF)

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

AUTHOR INFORMATION

Corresponding Author

Johannes Kästner − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0001-6178-7669; Email: kaestner@theochem.uni-stuttgart.de

Authors

Viktor Zaverkin − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0001-9940-8548

Julia Netz − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0003-1223-0391

Fabian Zills − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany*

Andreas Köhn − *Institute for Theoretical Chemistry, University of Stuttgart, 70569 Stuttgart, Germany;* orcid.org/0000-0002-0844-842X

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.1c00853

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075-390740016 under Germany's Excellence Strategy and through grant no INST 40/575-1 FUGG (JUSTUS 2 cluster). They acknowledge the support by the Stuttgart Center for Simulation Science (SimTech). They also acknowledge the support by the state of Baden-Württemberg through the bwHPC consortium for providing computer time. V.Z. acknowledges financial support received in the form of a Ph.D. scholarship from the Studienstiftung des Deutschen Volkes (German National Academic Foundation).

REFERENCES

(1) Gatteschi, D.; Sessoli, R.; Villain, J. *Molecular Nanomagnets*; Oxford University Press, 2006.

(2) Craig, G. A.; Murrie, M. 3d single-ion magnets. *Chem. Soc. Rev.* **2015**, *44*, 2135−2147.

(3) Gaita-Ariño, A.; Luis, F.; Hill, S.; Coronado, E. Molecular spins for quantum computation. *Nat. Chem.* **2019**, *11*, 301−309.

(4) Sanvito, S. Molecular spintronics. *Chem. Soc. Rev.* **2011**, *40*, 3336−3355.

(5) Gatteschi, D.; Sessoli, R. Quantum Tunneling of Magnetization and Related Phenomena in Molecular Materials. *Angew. Chem., Int. Ed.* **2003**, *42*, 268−297.

(6) Rechkemmer, Y.; Breitgoff, F. D.; van der Meer, M.; Atanasov, M.; Hakl, M.; Orlita, M.; Neugebauer, P.; Neese, F.; Sarkar, B.; van Slageren, J. A four-coordinate cobalt(II) single-ion magnet with coercivity and a very high energy barrier. *Nat. Commun.* **2016**, *7*, No. 10467.

(7) Harman, W. H.; Harris, T. D.; Freedman, D. E.; Fong, H.; Chang, A.; Rinehart, J. D.; Ozarowski, A.; Sougrati, M. T.; Grandjean, F.; Long, G. J.; Long, J. R.; Chang, C. J. Slow Magnetic Relaxation in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes. *J. Am. Chem. Soc.* **2010**, *132*, 18115−18126.

(8) Rogez, G.; Rebilly, J.-N.; Barra, A.-L.; Sorace, L.; Blondin, G.; Kirchner, N.; Duran, M.; van Slageren, J.; Parsons, S.; Ricard, L.; Marvilliers, A.; Mallah, T. Very Large Ising-Type Magnetic Anisotropy in a Mononuclear Ni(II) Complex. *Angew. Chem., Int. Ed.* **2005**, *44*, 1876−1879.

(9) Bamberger, H.; Albold, U.; Midlíková, J. D.; Su, C.-Y.; Deibel, N.; Hunger, D.; Hallmen, P. P.; Neugebauer, P.; Beerhues, J.; Demeshko, S.; Meyer, F.; Sarkar, B.; van Slageren, J. Iron(II), Cobalt(II), and Nickel(II) Complexes of Bis(sulfonamido)benzenes: Redox Properties, Large Zero-Field Splittings, and Single-Ion Magnets. *Inorg. Chem.* **2021**, *60*, 2953−2963.

(10) Blank, T. B.; Brown, S. D.; Calhoun, A. W.; Doren, D. J. Neural network models of potential energy surfaces. *J. Chem. Phys.* **1995**, *103*, 4129−4137.

(11) Lorenz, S.; Groß, A.; Scheffler, M. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* **2004**, *395*, 210−215.

(12) Lorenz, S.; Scheffler, M.; Gross, A. Descriptions of surface chemical reactions using a neural network representation of the potential-energy surface. *Phys. Rev. B* **2006**, *73*, No. 115431.

(13) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98*, No. 146401.

(14) Behler, J. Neural network potential-energy surfaces in chemistry: a tool for large-scale simulations. *Phys. Chem. Chem. Phys.* **2011**, *13*, 17930−17955.

(15) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016**, *145*, No. 170901.

(16) Shapeev, A. V. Moment Tensor Potentials: A Class of Systematically Improvable Interatomic Potentials. *Multiscale Model. Simul.* **2016**, *14*, 1153−1173.

(17) Gastegger, M.; Behler, J.; Marquetand, P. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.* **2017**, *8*, 6924−6935.

(18) Gubaev, K.; Podryabinkin, E. V.; Shapeev, A. V. Machine learning of molecular properties: Locality and active learning. *J. Chem. Phys.* **2018**, *148*, No. 241727.

(19) Dral, P. O.; Barbatti, M.; Thiel, W. Nonadiabatic Excited-State Dynamics with Machine Learning. *J. Phys. Chem. Lett.* **2018**, *9*, 5660−5663.

(20) Zhang, L.; Han, J.; Wang, H.; Car, R.; Weinan, E. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Phys. Rev. Lett.* **2018**, *120*, No. 143001.

(21) 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. *Chem. Sci.* **2018**, *9*, 2261−2269.

(22) Westermayr, J.; Gastegger, M.; Menger, M. F. S. J.; Mai, S.; González, L.; Marquetand, P. Machine learning enables long time scale molecular photodynamics simulations. *Chem. Sci.* **2019**, *10*, 8100−8107.

(23) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *J. Chem. Theory Comput.* **2019**, *15*, 3678−3693.

(24) Zaverkin, V.; Kästner, J. Gaussian Moments as Physically Inspired Molecular Descriptors for Accurate and Scalable Machine Learning Potentials. *J. Chem. Theory Comput.* **2020**, *16*, 5410−5421.

(25) Molpeceres, G.; Zaverkin, V.; Kästner, J. Neural-network assisted study of nitrogen atom dynamics on amorphous solid water − I. adsorption and desorption. *Mon. Not. R. Astron. Soc.* **2020**, *499*, 1373−1384.

(26) Dral, P. O. Quantum Chemistry in the Age of Machine Learning. *J. Phys. Chem. Lett.* **2020**, *11*, 2336−2347.

(27) Molpeceres, G.; Zaverkin, V.; Watanabe, N.; Kästner, J. Binding energies and sticking coefficients of H2 on crystalline and amorphous CO ice. *Astron. Astrophys.* **2021**, *648*, No. A84.

(28) Christensen, A. S.; Bratholm, L. A.; Faber, F. A.; von Lilienfeld, O. FCHL revisited: Faster and more accurate quantum machine learning. *J. Chem. Phys.* **2020**, *152*, No. 044107.

(29) Faber, F. A.; Christensen, A. S.; Huang, B.; von Lilienfeld, O. A. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.* **2018**, *148*, No. 241717.

(30) Zaverkin, V.; Holzmüller, D.; Steinwart, I.; Kästner, J. Fast and Sample-Efficient Interatomic Neural Network Potentials for Mole-

cules and Materials Based on Gaussian Moments. *J. Chem. Theory Comput.* **2021**, *17*, 6658−6670.
(31) Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* **1991**, *4*, 251−257.
(32) Novikov, I.; Grabowski, B.; Kormann, F.; Shapeev, A. Machine-Learning Interatomic Potentials Reproduce Vibrational and Magnetic Degrees of Freedom. 2021, arXiv:2012.12763. arXiv.org e-Print archive. https://arxiv.org/abs/2012.12763.
(33) Lunghi, A.; Sanvito, S. Surfing Multiple Conformation-Property Landscapes via Machine Learning: Designing Single-Ion Magnetic Anisotropy. *J. Phys. Chem. C* **2020**, *124*, 5802−5806.
(34) Lunghi, A. Insights into the Spin-Lattice Dynamics of Organic Radicals Beyond Molecular Tumbling: A Combined Molecular Dynamics and Machine-Learning Approach. *Appl. Magn. Reson.* **2020**, *51*, 1343−1356.
(35) Atanasov, M.; Ganyushin, D.; Pantazis, D. A.; Sivalingam, K.; Neese, F. Detailed Ab Initio First-Principles Study of the Magnetic Anisotropy in a Family of Trigonal Pyramidal Iron(II) Pyrrolide Complexes. *Inorg. Chem.* **2011**, *50*, 7460−7477.
(36) Abragam, A.; Bleaney, B. *Electron Paramagnetic Resonance of Transition Ions*; Clarendon Press, 1970.
(37) Jacot, A.; Gabriel, F.; Hongler, C. In *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*, NeurIPS Proceedings, 2018.
(38) Hendrycks, D.; Gimpel, K. Gaussian Error Linear Units (GELUs). 2016, arXiv:1606.08415. arXiv.org e-Print archive. https://arxiv.org/abs/1606.08415.
(39) Elfwing, S.; Uchibe, E.; Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural Networks* **2018**, *107*, 3−11.
(40) Ramachandran, P.; Zoph, B.; Le, Q. V. Searching for Activation Functions. 2018, arXiv:1710.05941. arXiv.org e-Print archive. https://arxiv.org/abs/1710.05941.
(41) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. In *Self-Normalizing Neural Networks*, NeurIPS Proceedings, 2017.
(42) Arora, S.; Du, S.; Hu, W.; Li, Z.; Salakhutdinov, R.; Wang, R. In *On Exact Computation with an Infinitely Wide Neural Net*, NeurIPS Proceedings, 2019.
(43) Lu, Y.; Gould, S.; Ajanthan, T. Bidirectional Self-Normalizing Neural Networks. 2020, arXiv:2006.12169. arXiv.org e-Print archive. https://arxiv.org/abs/2006.12169.
(44) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 2015, arXiv:1412.6980. arXiv.org e-Print archive. https://arxiv.org/abs/1412.6980.
(45) Prechelt, L. *Neural Networks: Tricks of the Trade*, 2nd ed.; Montavon, G.; Orr, G. B.; Müller, K.-R., Eds.; Springer: Berlin, Heidelberg, 2012; pp 53−67.
(46) Larsen, A. H.; Mortensen, J. J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; Hermes, E. D.; Jennings, P. C.; Jensen, P. B.; Kermode, J.; Kitchin, J. R.; Kolsbjerg, E. L.; Kubal, J.; Kaasbjerg, K.; Lysgaard, S.; Maronsson, J. B.; Maxson, T.; Olsen, T.; Pastewka, L.; Peterson, A.; Rostgaard, C.; Schiøtz, J.; Schütt, O.; Strange, M.; Thygesen, K. S.; Vegge, T.; Vilhelmsen, L.; Walter, M.; Zeng, Z.; Jacobsen, K. W. The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter* **2017**, *29*, No. 273002.
(47) Settles, B. *Active Learning Literature Survey*, Computer Sciences Technical Report 1648; University of Wisconsin−Madison, 2009.
(48) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org. https://www.tensorflow.org/.
(49) Blöchl, P. E. Projector augmented-wave method. *Phys. Rev. B* **1994**, *50*, 17953.
(50) Kresse, G.; Joubert, D. From ultrasoft pseudopotentials to the projector augmented-wave method. *Phys. Rev. B* **1999**, *59*, 1758−1775.
(51) Perdew, J.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **1997**, *78*, 1396.
(52) Kresse, G.; Hafner, J. Ab initio molecular dynamics for liquid metals. *Phys. Rev. B* **1993**, *47*, 558.
(53) Kresse, G.; Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **1996**, *6*, 15−50.
(54) Kresse, G.; Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **1996**, *54*, 11169.
(55) Dudarev, S. L.; Botton, G. A.; Savrasov, S. Y.; Humphreys, C. J.; Sutton, A. P. Electron-energy-loss spectra and the structural stability of nickel oxide: An LSDA+U study. *Phys. Rev. B* **1998**, *57*, 1505.
(56) Mann, G. W.; Lee, K.; Cococcioni, M.; Smit, B.; Neaton, J. B. First-principles Hubbard U approach for small molecule binding in metal-organic frameworks. *J. Chem. Phys.* **2016**, *144*, No. 174104.
(57) 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. *J. Chem. Phys.* **2010**, *132*, No. 154104.
(58) Werner, H.-J.; Knowles, P. J.; Knizia, G.; Manby, F. R.; Schütz, M.et al.*MOLPRO, A Package of Ab Initio Programs*, version 2020.0, 2020. https://www.molpro.net.
(59) McWeeny, R. *Methods of Molecular Quantum Mechanics*; Academic Press, 1996.
(60) McWeeny, R. SCF theory for excited states. *Mol. Phys.* **1974**, *28*, 1273−1282.
(61) Hallmen, P. P.; Köppl, C.; Rauhut, G.; Stoll, H.; Van Slageren, J. Fast and reliable ab initio calculation of crystal field splittings in lanthanide complexes. *J. Chem. Phys.* **2017**, *147*, No. 164101.
(62) Calvello, S.; Piccardo, M.; Rao, S. V.; Soncini, A. CERES: An ab initio code dedicated to the calculation of the electronic structure and magnetic properties of lanthanide complexes. *J. Comput. Chem.* **2018**, *39*, 328−337.
(63) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−3305.
(64) Marian, C. M.; Wahlgren, U. A new mean-field and ECP-based spin-orbit method. Applications to Pt and PtH. *Chem. Phys. Lett.* **1996**, *251*, 357−364.
(65) Heß, B. A.; Marian, C. M.; Wahlgren, U.; Gropen, O. A mean-field spin-orbit method applicable to correlated wavefunctions. *Chem. Phys. Lett.* **1996**, *251*, 365−371.
(66) Berning, A.; Schweizer, M.; Werner, H.-J.; Knowles, P. J.; Palmieri, P. Spin-orbit matrix elements for internally contracted multireference configuration interaction wavefunctions. *Mol. Phys.* **2000**, *98*, 1823−1833.
(67) Chibotaru, L. F.; Ungur, L. Ab initio calculation of anisotropic magnetic properties of complexes. I. Unique definition of pseudospin Hamiltonians and their derivation. *J. Chem. Phys.* **2012**, *137*, No. 064112.
(68) Ruddigkeit, L.; van Deursen, R.; Blum, L. C.; Reymond, J.-L. Enumeration of 166 Billion Organic Small Molecules in the Chemical Universe Database GDB-17. *J. Chem. Inf. Model.* **2012**, *52*, 2864−2875.
(69) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, No. 140022.
(70) Schütt, K. T.; Arbabzadah, F.; Chmiela, S.; Müller, K. R.; Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **2017**, *8*, No. 13890.

(71) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **2017**, *3*, No. e1603015.
(72) Chmiela, S.; Sauceda, H. E.; Müller, K.-R.; Tkatchenko, A. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **2018**, *9*, No. 3887.
(73) Zaverkin, V.; Netz, J.; Zills, F.; Köhn, A.; Kästner, J. *Thermally Averaged Magnetic Anisotropy Tensors via Machine Learning Based on Gaussian Moments*, version v1; Zenodo, 2021. https://doi.org/10.5281/zenodo.5172156.
(74) Errington, J.; Debenedetti, P. Relationship between structural order and the anomalies of liquid water. *Nature* **2001**, *409*, 318−321.
(75) Titiš, J.; Miklovič, J.; Boča, R. Magnetostructural study of tetracoordinate cobalt(II) complexes. *Inorg. Chem. Commun.* **2013**, *35*, 72−75.
(76) Liu, J.; He, X.; Zhang, J. Z. H.; Qi, L.-W. Hydrogen-bond structure dynamics in bulk water: insights from ab initio simulations with coupled cluster theory. *Chem. Sci.* **2018**, *9*, 2065−2073.
(77) Fataftah, M. S.; Zadrozny, J. M.; Rogers, D. M.; Freedman, D. E. A Mononuclear Transition Metal Single-Molecule Magnet in a Nuclear Spin-Free Ligand Environment. *Inorg. Chem.* **2014**, *53*, 10716−10721.
(78) Carl, E.; Demeshko, S.; Meyer, F.; Stalke, D. Triimidosulfonates as Acute Bite-Angle Chelates: Slow Relaxation of the Magnetization in Zero Field and Hysteresis Loop of a CoII Complex. *Chem.—Eur. J.* **2015**, *21*, 10109−10115.
(79) Suturina, E. A.; Nehrkorn, J.; Zadrozny, J. M.; Liu, J.; Atanasov, M.; Weyhermüller, T.; Maganas, D.; Hill, S.; Schnegg, A.; Bill, E.; Long, J. R.; Neese, F. Magneto-Structural Correlations in Pseudotetrahedral Forms of the [Co(SPh)4]2− Complex Probed by Magnetometry, MCD Spectroscopy, Advanced EPR Techniques, and ab Initio Electronic Structure Calculations. *Inorg. Chem.* **2017**, *56*, 3102−3118.
(80) Wu, T.; Zhai, Y.-Q.; Deng, Y.-F.; Chen, W.-P.; Zhang, T.; Zheng, Y.-Z. Correlating magnetic anisotropy with the subtle coordination geometry variation of a series of cobalt(ii)-sulfonamide complexes. *Dalton Trans.* **2019**, *48*, 15419−15426.
(81) Burg, J. P. *Maximum Entropy Spectral Analysis*; Stanford University, 1975.
(82) Vos, K. A Fast Implementation of Burg's Method, 2013. Available: www.opuscodec.org/docs/vos_fastburg.pdf.
(83) Martini, A.; Schmidt, S.; del Pozzo, W. Maximum Entropy Spectral Analysis: A Case Study. 2021, arXiv:2106.09499. arXiv.org e-Print archive. https://arxiv.org/abs/2106.09499.
(84) Martini, A.; Schmidt, S.; del Pozzo, W. Maximum Entropy Spectrum, 2021. https://github.com/martini-alessandro/Maximum-Entropy-Spectrum.
(85) Lunghi, A.; Totti, F.; Sanvito, S.; Sessoli, R. Intra-molecular origin of the spin-phonon coupling in slow-relaxing molecular magnets. *Chem. Sci.* **2017**, *8*, 6051−6059.

https://doi.org/10.1021/acs.jctc.1c00853
J. Chem. Theory Comput. 2022, 18, 1−12

Neural-network assisted study of nitrogen atom dynamics on amorphous solid water – II. Diffusion

Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **510,** 3063–3070 (2022)
Advance Access publication 2021 December 13

https://doi.org/10.1093/mnras/stab3631

Neural-network assisted study of nitrogen atom dynamics on amorphous solid water – II. Diffusion

Viktor Zaverkin, Germán Molpeceres and Johannes Kästner ★

Institute for Theoretical Chemistry, University of Stuttgart, Pfaffenwaldring 55, D-70569 Stuttgart, Germany

Accepted 2021 December 9. Received 2021 December 9; in original form 2021 July 19

ABSTRACT

The diffusion of atoms and radicals on interstellar dust grains is a fundamental ingredient for predicting accurate molecular abundances in astronomical environments. Quantitative values of diffusivity and diffusion barriers usually rely heavily on empirical rules. In this paper, we compute the diffusion coefficients of adsorbed nitrogen atoms by combining machine learned interatomic potentials, metadynamics, and kinetic Monte Carlo simulations. With this approach, we obtain a diffusion coefficient of nitrogen atoms on the surface of amorphous solid water of merely $(3.5 \pm 1.1) \times 10^{-34}$ cm^2 s^{-1} at 10 K for a bare ice surface. Thus, we find that nitrogen, as a paradigmatic case for light and weakly bound adsorbates, is unable to diffuse on bare amorphous solid water at 10 K. Surface coverage has a strong effect on the diffusion coefficient by modulating its value over 9–12 orders of magnitude at 10 K and enables diffusion for specific conditions. In addition, we have found that atom tunnelling has a negligible effect. Average diffusion barriers of the potential energy surface (2.56 kJ mol^{-1}) differ strongly from the effective diffusion barrier obtained from the diffusion coefficient for a bare surface (6.06 kJ mol^{-1}) and are, thus, inappropriate for diffusion modelling. Our findings suggest that the thermal diffusion of N on water ice is a process that is highly dependent on the physical conditions of the ice.

Key words: astrochemistry – molecular data – methods: numerical – ISM: molecules.

1 INTRODUCTION

The mobility of atoms and molecules on the surface of interstellar dust grains is crucial for surface processes like the formation of complex organic molecules (COMs). Observed abundances can be explained only by a combination of gas-phase reactions and surface chemistry (Herbst & van Dishoeck 2009). It is believed that diffusive processes, like the Langmuir–Hinshelwood mechanism, prevail in surface chemistry at low temperatures (Herbst & van Dishoeck 2009; Ruaud et al. 2015), although the Eley–Rideal and 'hot-atom' mechanisms may have significant importance (He, Emtiaz & Vidali 2017).

The rate-limiting step of diffusive mechanisms is the mobility of adsorbates on the surface. At the average temperature of molecular clouds of 10–20 K (Snow & McCall 2006), diffusion by thermal hopping is limited. The mass and binding energy are the main factors determining if an adsorbate diffuses or not. There is evidence from experiment (Tsong 2001; Hama et al. 2012; Kuwahata et al. 2015) and simulation (Ásgeirsson, Jónsson & Wikfeldt 2017; Senevirathne et al. 2017) for efficient diffusion of H, D, H_2, and He on ice surfaces at temperatures as low as 10 K. Especially, the diffusion of H may be facilitated by tunnelling (Kuwahata et al. 2015) on polycrystalline water ice. On amorphous solid water (ASW), the influence of tunnelling greatly depends on the adsorption site under consideration (Hama et al. 2012; Ásgeirsson et al. 2017; Senevirathne et al. 2017)

A different situation arises for the next set of light particles with relevance in astrochemistry, namely the first-row atoms C, N, and O, for which rich chemistry is expected. The interaction of these atoms with ASW has been theoretically studied recently (Shimonishi et al. 2018), finding that C forms a tightly bound complex and is unable to diffuse, while O and especially N are much weaker bound. Quantum tunnelling was claimed by indirect evidence to be responsible for O diffusion (Minissale et al. 2013), a finding disputed later (Pezzella, Unke & Meuwly 2018). Here, we report on the diffusivity of nitrogen atoms on ASW surfaces.

Recently, we investigated the adsorption dynamics of the N atom on ASW (Molpeceres, Zaverkin & Kästner 2020b) using ab-initio molecular dynamics employing a neural-network potential [machine learned potential (MLP), Zaverkin & Kästner 2020], finding sticking to be extremely effective at low temperatures and desorption to occur in a window in between 23–28 K, in agreement with previous experiments (Minissale, Congiu & Dulieu 2016). The average binding energy of N on ASW was found to be very small (∼2.9 kJ mol^{-1}, including zero-point vibrational energies), also in agreement with the average value provided in recent simulations (Shimonishi et al. 2018). Our reported distribution of binding energies is also in accordance with the experimental values of the literature (Minissale et al. 2016), with a lower average value (by a factor of 2) but a significant amount of binding sites in their provided range (∼5.8 kJ mol^{-1}; see fig. 2 of Molpeceres et al. 2020b).

The small binding energy, in combination with the small mass of N and the possibility of simulating long time scales thanks to the MLP, has motivated us to explicitly study the diffusion of an adsorbate other than H and D with relevance to interstellar surface chemistry. Using a combination of accelerated sampling techniques (Laio & Parrinello 2002; Barducci, Bussi & Parrinello 2008) to construct a 2D free-energy surface (FES) experienced by the N atoms and kinetic Monte Carlo (kMC) simulations, we have estimated the diffusion coefficient of N as a function of the temperature, as well as other diffusion-related properties.

★ E-mail: kaestner@theochem.uni-stuttgart.de

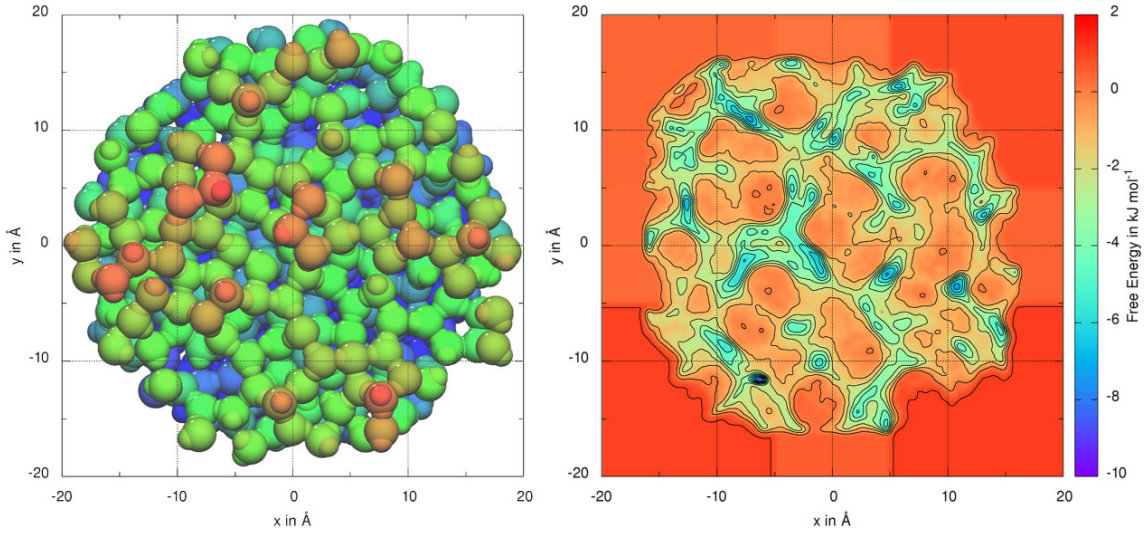


Figure 1. (Left-hand panel) Atomic structure of the amorphous water ice equilibrated at 50 K. All atoms are coloured according to their z-coordinate value (surface normal) with red atoms sticking out of the surface and blue atoms denoting cavities. (Right-hand panel) The 2D FES for the adsorbed nitrogen atom on the ice surface.

This paper has the following structure: First, we briefly introduce the relevant computational details in Section 2, and describe the obtained results in Section 3. The justification of our finding is presented in Section 4, while a rationalization of previous results in the light of our recent simulations is discussed in the last section (Section 5). The results presented here have important implications for the chemistry of dense, interstellar clouds.

2 COMPUTATIONAL DETAILS

The study of diffusion requires long time scales, short time steps in direct molecular dynamics, and a very accurate potential energy surface. We have achieved this by combining ab-initio molecular dynamics with a neural-network potential (MLP) (Zaverkin & Kästner 2020; Molpeceres et al. 2020b), free-energy sampling using metadynamics, and kMC based on the minima and saddle points on the FES.

2.1 Machine learned potential

To create an accurate interatomic potential, we employed the Gaussian moment neural-network (GM-NN) approach (Zaverkin & Kästner 2020) recently developed in our group. Our MLP was fitted to a training set of energies and gradients at the PBEh-3c/def2-mSVP (Grimme et al. 2015) level of 28 715 structures with 3 to 378 atoms each. For more details on the data set employed in this work see elsewhere (Molpeceres et al. 2020b). The training set structures are accessible free of charge (Molpeceres, Zaverkin & Kästner 2020a). For details on the construction of the respective MLP and on the analysis of its accuracy during molecular dynamics simulations, see Supplementary Information.

2.2 Free-energy sampling

The free-energy sampling of a region spanning 800 Å^2 was performed using well-tempered metadynamics simulations (Huber, Torda & van Gunsteren 1994; Laio & Parrinello 2002; Barducci et al. 2008). For that, we interfaced our neural-network potential with the Atomic Simulation Environment (ASE) (Hjorth Larsen et al. 2017) and the PLUMED package (Bonomi et al. 2009; Tribello et al. 2014; The PLUMED consortium et al. 2019). The internal modes of water were flexible. The collective variables for the metadynamics were selected to be the x and y components of the nitrogen atom diffusing on the surface. The parameters of the Gaussian bias potential in the metadynamics were a rate of deposition of 125 fs, a Gaussian height of 0.025 kJ mol^{-1}, a Gaussian width of 0.25 Å for each of the collective variables, and a bias factor of 6. These parameters were obtained after extensive testing to produce a smooth FES. In addition, we included an arbitrarily high wall potential (spherical) to avoid nitrogen escaping via the borders of the ice. The complete sampling of such a big region is impossible, even with metadynamics simulations. Thus, we have divided the complete surface into nine different sub-regions of reduced size running metadynamics simulations in each one of them. Jumps between the different sub-regions were avoided by a harmonic potential wall. The FES was then reconstructed by overlapping each sub-region using a weighting function to ensure a smooth potential. For more details, see Supplementary Information. Each sub-region was sampled at a temperature of 50 K for a total time of 6.5 ns each. The FES was reconstructed from the negative of the history-dependent bias potential (Laio & Parrinello 2002). Our structural model of the ice and the reconstructed 2D FES for the movement of the nitrogen atom in x and y directions are shown in Fig. 1.

2.3 Minima and transition state optimization

Using the analytic FES and its derivatives, we optimized the minima and transition states using the optimization library DL-FIND (Kästner et al. 2009). We obtained 139 minima and coarse-grained them to 60 by combining close-lying minima separated by negligible barriers. We calculated the transition states that can be reduced to a single elementary step using the Nudged Elastic Band (NEB) method (Jónsson, Mills & Jacobsen 1998; Henkelman & Jónsson 2000; Henkelman, Uberuaga & Jónsson 2000). We ended up with 60 minima interconnected by 107 transition states. All relevant data are

given in Supplementary Information, to ensure the reproducibility of this work. Rate constants for each possible transition (direct and reverse) were obtained in the context of transition state theory by applying Eyring's equation. Explicit consideration of tunnelling for each transition was incorporated using Eckart and Bell-type tunnelling corrections for each rate constant.

2.4 Modelling of diffusion

We constructed a kMC model (Bortz, Kalos & Lebowitz 1975; Gillespie 1976) explicitly considering all the activation barriers and minima on our FES. We ran 10^8 kMC steps at each temperature (10^9 steps at 17 K). After that, the probability of finding the N adatom at each binding site in kMC resembled its Boltzmann probability. This was achievable down to 17 K. To reach lower temperatures, we had to extrapolate, see Fig. 3. To take our model's boundaries into account appropriately, we divided the diffusion paths into segments (Kirchheim 1987, 1988; Ramasubramaniam et al. 2008) that end once the N atom reaches the confining potential wall. A new segment is started if a random number between 0 and 1 is larger than 0.5 to mimic the possible hop out of our boundaries. Additionally, we started a new kMC segment if, during 10^5 iterations, no border was reached to obtain better statistics on estimated diffusion coefficients. The diffusion coefficient was calculated from the mean squared displacement (MSD) of the nitrogen atom by time-averaging over the segments i:

$$D = \sum_i \frac{D_i \triangle t_i}{t}, \tag{1}$$

where $\triangle t_i = t_i - t_{i-1}$ is the time length of the segment i, t is the total time of the kMC simulation, and the respective diffusion coefficient D_i is calculated as

$$D_i = \frac{(\boldsymbol{r}(t_i) - \boldsymbol{r}(t_{i-1}))^T (\boldsymbol{r}(t_i) - \boldsymbol{r}(t_{i-1}))}{2d \triangle t_i}, \tag{2}$$

where $d = 2$ for a two-dimensional system. We carefully validated all simulation parameters, also the non-periodic model, by comparison between easy-to-sample periodic and non-periodic auxiliary models, see Supplementary Information. Finally, the simulation of different degrees of surface coverage was done by removing specific minima for the kMC simulation, assuming a non-reactive species is already occupying such state, therefore the distribution of binding sites do not change by the number of adsorbed atoms.

The metadynamics simulations required direct molecular dynamics of 6.5 ns, while the kMC runs covered 10^{12} s. This protocol resulted in temperature-dependent diffusion coefficients as time averages of our kMC trajectories.

3 COMPUTATIONAL RESULTS

3.1 Free-energy surface

Fig. 1 (left-hand panel) shows the ASW ice surface equilibrated at 50 K with atoms coloured according to their z-coordinate. Fig. 1 (right-hand panel) represents the respective 2D FES for the adsorbed nitrogen atom on the ice surface. From these figures, the complexity of the ASW and FES surface topologies can be deduced. Moreover, the presence of small pores is expected to be crucial when computing the diffusion coefficients. Note that we assume that differences between FESs obtained for temperatures equal to or lower than 50 K are small since the variation of the entropy of the system should be minute. Therefore, it is sufficient to sample the FES at 50 K only and use the obtained diffusion barriers within the kMC framework for other temperatures.

Fig. 2 (left-hand panel) shows the distribution of activation barriers for diffusion obtained for the coarse-grained FES. It can be seen that the respective distribution is relatively broad with a mean value of 2.56 kJ mol^{-1} and a standard deviation value of 1.72 kJ mol^{-1}. Taking into account the broad distribution of activation energies, one could claim that the assumption about the relation of the effective diffusion barrier to the mean binding energy, i.e. $E_{\rm diff} \sim 0.55 E_{\rm bin}$ found in the literature (Minissale et al. 2016), depends on the physical conditions under consideration since it does not directly compare with our value of 0.76 (computed from our distribution of binding energies excluding zero-point vibrational energies, see Molpeceres et al. 2020b).

In addition, one may estimate the pre-exponential factor, D_0, of the classical Arrhenius expression

$$D = D_0 \exp\left(-\frac{\Delta F}{RT}\right), \tag{3}$$

directly from FES. For this purpose, one may write for D_0, similar to Du, Rogal & Drautz (2012),

$$D_0 = \Gamma a_0^2 \nu_0, \tag{4}$$

to which we add a superscript, i.e. we write $D_0^{\rm avg}$, to simplify the comparison of the obtained result to the corresponding values presented in Section 3.2. In the expression in equation (3), we set the effective diffusion barrier ΔF to $\Delta F^{\rm avg} = 2.56$ kJ mol^{-1} and, in equation (4), a_0 is the mean jump distance, ν_0 is the attempt frequency, and Γ is the geometric pre-factor related to the connectivity of each site to its neighbouring sites.

The attempt frequency ν_0 can be estimated as the vibrational frequency of the adatom averaged over all sites. We obtained a value of $\nu_0 = 8.9 \cdot 10^{12}$ s^{-1}. The respective harmonic frequencies are calculated from the Hessian matrix computed for the binding sites on the coarse-grained FES.

The jump distance can be estimated from the distance distribution between neighbouring minima, shown in Fig. 2 (right-hand panel). Thus, we obtain a mean jump distance value of $a_0 = 4.20$ Å. For the estimation of geometric pre-factor Γ, we assume isotropic diffusion, which results in the following expression (Allnatt & Lidiard 1993):

$$\Gamma = \frac{n}{2d}, \tag{5}$$

where n is the number of neighbouring states and d is the dimensionality of the system. The former can be determined as the mean of the connectivity of all sites and equals to 4. Thus, we obtained the pre-exponential factor of $D_0^{\rm avg} = 1.57 \times 10^{-2}$ cm^2 s^{-1}, setting $d = 2$. The respective Arrhenius plot is depicted in Fig. 3.

Note that due to the broad distribution of activation barriers, distances between neighbouring minima, and the number of neighbouring states the quantities derived in this section, $D_0^{\rm avg}$ and $\Delta F^{\rm avg}$, may deviate from the ground truth. Thus, a more rigorous description of diffusion processes is needed. The kMC approach, employed in Section 3.2, provides us with the necessary flexibility taking into account the connectivity between binding sites with their realistic barriers.

3.2 Kinetic Monte Carlo

In Section 3.1, we have seen that the 2D FES for the adsorbed nitrogen atom has a broad distribution of diffusion barriers, distances between neighbouring sites, and the number of neighbours. Therefore, to

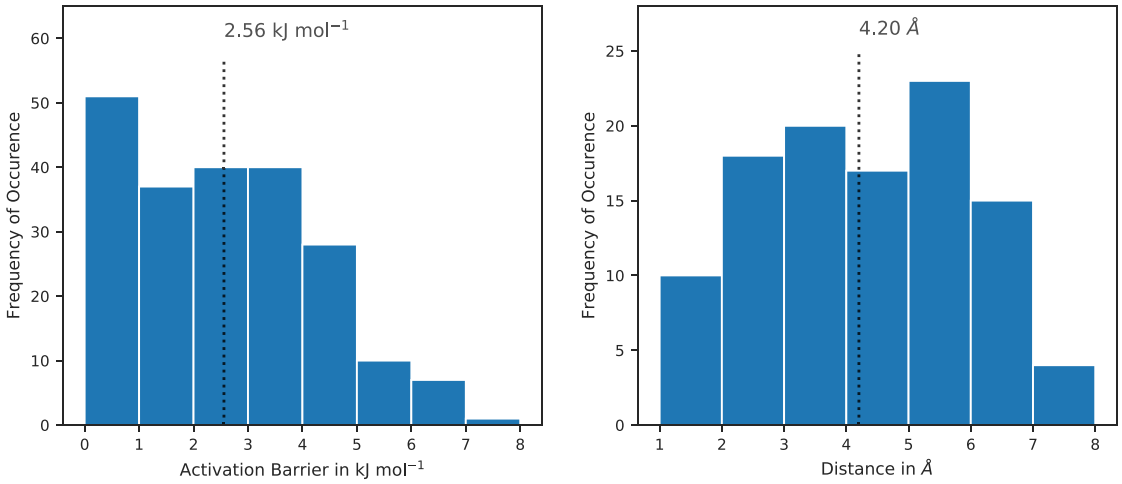


Figure 2. (Left-hand panel) Distribution of diffusion barriers with a mean of 2.56 kJ mol^{-1} and a standard deviation of 1.72 kJ mol^{-1}. (Right-hand panel) Distribution of distances between neighbouring sites on the FES with a mean of 4.20 Å and a standard deviation of 1.63 Å.

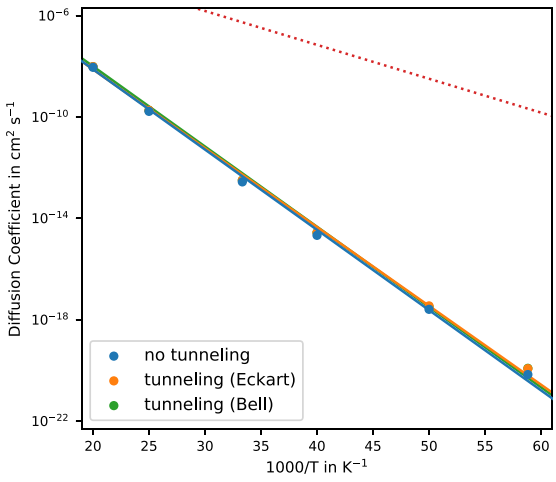


Figure 3. Temperature dependence of diffusion coefficients (D) for N on ASW for the bare surface with or without tunnelling correction. Linear fits are displayed for clarity, and the red dotted line represents D^{avg} estimated in Section 3.1.

avoid hopping between two states separated by a low activation barrier and, thus, improve the efficiency of kMC simulations, we employ the coarse-grained FES, the generation of which is briefly discussed in Section 2.3.

3.2.1 Single adsorbed nitrogen

To study the mobility of the nitrogen atom on the water surface, we performed kMC simulations at six different substrate temperatures $T = \{17, 20, 25, 30, 40, \text{and } 50\}$ K. Each simulation was started from a randomly selected site and was performed for 10^8–10^9 steps resulting in total times ranging from seconds (50 K) to several thousands of years (17 K). For each temperature value, we performed 25 independent kMC runs with the exception of $T = 17$ K for with 10 independent kMC runs were performed. The reason for that is the increased computational cost compared to other temperature values due to the increased number of steps (10^9). Converging kMC simulations at 10 K was impossible in a reasonable time.

To make sure that the sampling was performed long enough, we compared kMC probability to find the nitrogen atom at a certain binding site to its Boltzmann equivalent. The kMC probability can be calculated as $p_i \propto t_i/t_{\mathrm{total}}$, where t_i is the time spend in the corresponding binding site i and t_{total} is the total time covered by the simulation.

Fig. 3 shows the temperature-dependent diffusion coefficients obtained for the bare surface employing kMC simulations along with the respective linear fit. Fitting the respective values by the Arrhenius expression presented in equation (3), we obtained the pre-exponential factor value of $D_0 = (1.65 \pm 0.32) \times 10^{-2}$ cm^2 s^{-1} and the effective diffusion barrier value of $\Delta F = (6.06 \pm 0.04)$ kJ mol^{-1}. It should be noted that diffusion coefficients predicted employing kMC simulations are different from D^{avg}, obtained using the averaged diffusion barrier in Section 3.1, by several orders of magnitude. While the pre-exponential factor is close to the one obtained in Section 3.1 ($D_0^{\mathrm{avg}} = 1.57 \times 10^{-2}$ cm^2 s^{-1}), the average diffusion barrier ($\Delta F^{\mathrm{avg}} = 2.56$ kJ mol^{-1}) is about 2.4 times smaller compared to the respective value obtained by kMC simulations. This makes the diffusion at low temperatures less probable compared to the estimations based on the FES only.

Using the data from the linear fit in Fig. 3, we estimated the diffusion coefficient at 10 K and obtained D (10 K) $= (3.47 \pm 1.07) \times 10^{-34}$ cm^2 s^{-1}. This is much lower than the estimate obtained using parameters from Section 3.1 [D^{avg} (10 K) $= 6.67 \times 10^{-16}$ cm^2 s^{-1}]. All numerical values of diffusion coefficients are presented in Table 1.

Note that it is advisable to perform companion molecular dynamics simulations to ensure that kMC simulations lead to correct state-to-state evolution of the studied system (Voter 2007). However, many theoretical studies have been performed on similar systems (Karssemeijer & Cuppen 2014; Karssemeijer et al. 2014; Ásgeirsson et al. 2017; Senevirathne et al. 2017) like the molecular dynamics study of Pezzella et al. (2018), which we use as a reference in this work.

From fig. 2A of Pezzella et al. (2018), one finds an MSD of 133.64 Å^2 for the adsorbed oxygen atom sampled over 500 ns at 50 K. This results in a diffusion coefficient of 6.68×10^{-9} cm^2 s^{-1} similar to the one obtained by us for the nitrogen atom (9.16×10^{-9} cm^2 s^{-1}). Similar values for the diffusion coefficients might be expected

Table 1. Diffusion coefficients in $cm^2\ s^{-1}$ obtained for the bare surface, including tunnelling, and with the one to four deepest sites blocked by running kMC simulations.

T (K)	Bare surface	Tunnelling (Eckart)	Tunnelling (Bell)	One site blocked	Two sites blocked	Three sites blocked	Four sites blocked
10	3.47×10^{-34} (1.07)	9.82×10^{-34} (2.79)	4.67×10^{-34} (1.98)	8.20×10^{-26} (3.30)	2.87×10^{-25} (0.33)	9.71×10^{-24} (2.26)	9.04×10^{-23} (1.86)
17	6.87×10^{-21} (2.22)	1.17×10^{-20} (0.26)	1.21×10^{-20} (0.33)	5.06×10^{-16} (1.93)	5.47×10^{-16} (0.76)	4.56×10^{-15} (1.30)	1.47×10^{-14} (0.35)
20	2.57×10^{-18} (0.58)	3.52×10^{-18} (0.98)	3.22×10^{-18} (1.55)	1.96×10^{-14} (0.67)	2.98×10^{-14} (0.12)	1.59×10^{-13} (0.13)	4.69×10^{-13} (0.49)
25	2.14×10^{-15} (0.11)	2.75×10^{-15} (0.19)	2.77×10^{-15} (0.18)	1.97×10^{-12} (0.15)	3.61×10^{-12} (0.04)	1.27×10^{-11} (0.02)	2.96×10^{-11} (0.07)
30	2.75×10^{-13} (0.11)	3.20×10^{-13} (0.09)	3.20×10^{-13} (0.12)	6.77×10^{-11} (0.58)	1.12×10^{-10} (0.02)	2.95×10^{-10} (0.06)	5.88×10^{-10} (0.15)
40	1.67×10^{-10} (0.02)	1.84×10^{-10} (0.02)	1.82×10^{-10} (0.02)	7.17×10^{-9} (0.13)	1.11×10^{-8} (0.03)	2.08×10^{-8} (0.06)	3.32×10^{-8} (0.09)
50	9.16×10^{-9} (0.04)	9.74×10^{-9} (0.05)	9.71×10^{-9} (0.06)	1.35×10^{-7} (0.03)	1.94×10^{-7} (0.03)	3.02×10^{-7} (0.04)	4.26×10^{-7} (0.06)
D_0	1.65×10^{-2} (0.32)	1.27×10^{-2} (0.23)	2.16×10^{-2} (0.57)	4.88×10^{-3} (1.23)	3.69×10^{-3} (0.27)	3.02×10^{-3} (0.44)	2.63×10^{-3} (0.34)
ΔF	6.06 (0.04)	5.96 (0.04)	6.06 (0.06)	4.36 (0.05)	4.23 (0.02)	3.92 (0.03)	3.73 (0.03)

Standard deviation is given in parentheses. Values for 10 K are estimated from the linear fit.

because the estimated average diffusion barriers are close for both systems and are 2.29 kJ mol^{-1} (Pezzella et al. 2018) for the oxygen atom and 2.56 kJ mol^{-1} for the nitrogen atom.

Additionally, we analysed direct molecular dynamics simulations of the nitrogen atom adsorbed on top of the ASW surface performed over 4 and 2 ns at 10 and 50 K, respectively. The analysis revealed the good correspondence of the diffusion coefficients obtained by the direct molecular dynamics and by our kMC simulations. For details, see Supplementary Information. Based on these results, we may argue that the kMC dynamics produces correct time evolution of the system and, thus, is statistically indistinguishable from a long molecular dynamics simulation.

Quantum tunnelling may potentially influence the mobility of nitrogen atoms. We use the Eckart and Bell corrections to the rate constants of the kMC simulations. For analytic expressions of the respective tunnelling corrections, see McConnell & Kästner (2017). The diffusion coefficients obtained using the corresponding corrected rate constants are shown in Fig. 3. The corresponding numerical values, as well as D_0 and ΔF, can be found in Table 1.

From Fig. 3, we see that the diffusion coefficients accounting for tunnelling are only marginally larger compared to the ones without tunnelling corrections. This result can be expected, taking into account the high mass of the nitrogen atom.

Another concept that has to be accounted for is the roughness of the underlying potential at the atomic length-scales as introduced by Pezzella et al. (2018). Following the work of Zwanzig (1988), we may write for the effective diffusion coefficient

$$D^* = D\exp\left(-\left(\frac{\epsilon}{RT}\right)^2\right), \tag{6}$$

where we use the Arrhenius expression for D from equation (3) and ϵ resembles the variations of the potential on atomic length-scales. Note that in contrast to Pezzella et al. (2018), we obtain a modified expression

$$D^* = D_0\exp\left(-\frac{\Delta F}{RT}-\left(\frac{\epsilon}{RT}\right)^2\right), \tag{7}$$

which is more suitable for fitting properties that vary on several orders of magnitude like the diffusion coefficient.

Fitting the expression in equation (7) to the kMC data results in a negligibly small roughness parameter ϵ and an increase of diffusion coefficient at 10 K of a factor of 2. Therefore, we skip the respective analysis in the subsequent sections.

3.2.2 *Blocking surface sites*

In general, the studied water surface has regions containing smaller pores on different parts of the ice surface that are characterized by high binding energy and a high number of neighbours. Note that in real ASW, much deeper pores are expected (Bossa et al. 2015). While performing kMC simulations on the bare surface, see Section 3.2.1, we observed that the nitrogen atom is able to diffuse quickly into one of these sites. Once the nitrogen atom was trapped by one of these sites, it stays there for most of the time. We have found that the nitrogen atom spends about 93 per cent of the simulation time in the site associated with the deepest free energy at 50 K, found at around $(x, y) = (-6.3, -11.6)$ Å for the respective FES shown in Fig. 1. For lower temperatures, the nitrogen atom stays in the corresponding site up to 99.99 per cent of the total simulation time. The smallest barrier to get out of the respective minimum is $\Delta F = 5.90$ kJ mol^{-1}.

Moreover, in an experimental setup or interstellar clouds, single nitrogen atom diffusion is highly unlikely due to the high atomic fluxes employed in the former and the relative molecular abundances, e.g. with respect to H_2 (Ruaud, Wakelam & Hersant 2016), in the latter. Higher surface coverage is expected in both cases. This raises the question of the influence of the presence of additional inert species on the mobility of the adsorbed nitrogen atom. To address this question, we exclude the deepest binding sites (one to four), which mimics their occupation by some inert chemical species. Note that in this section, we neglect tunnelling, since we have found that it has only a marginal impact on the mobility of the nitrogen atom on the water ice surface.

Similar to Section 3.2.1, we performed kMC simulations at different substrate temperatures ranging from 17 to 50 K. All kMC simulations were performed for 10^6–10^8 steps since the Boltzmann distribution could be achieved faster when excluding the deeper sites. For all temperatures, 25 independent kMC runs were performed. Simulations at 10 K were impossible in a reasonable time due to computational cost even after excluding four deeper binding sites.

Fig. 4 shows the temperature-dependent diffusion coefficients of the nitrogen atom along with the corresponding linear fits for the bare surface and the surface with one to four binding sites excluded from the simulation. From the figure, one can see that the mobility of the nitrogen atom on the water ice surface increases strongly with the increasing number of blocked sites. The strongest effect was observed by excluding the first deepest site. While the difference of diffusion coefficients for the bare surface and the surface with multiple occupied binding sites is smaller for temperatures above 30 K, the effect on the mobility of the nitrogen is greater at lower temperatures. We have found that the diffusion coefficient for the adsorbed nitrogen atom at 10 K is 9–12 orders of magnitude larger

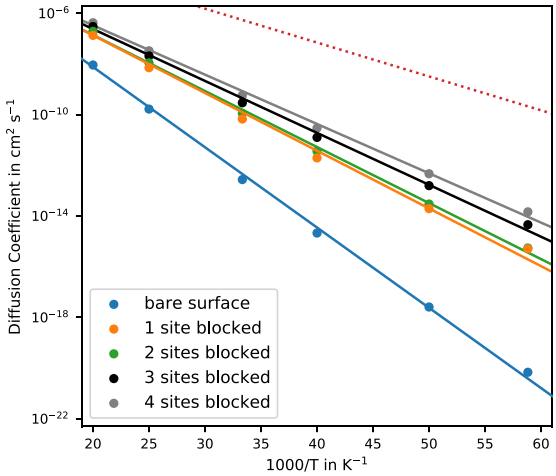


Figure 4. Temperature dependence of diffusion coefficients (D) for N on ASW for the bare surface and the surface with the one to four deepest sites blocked. Linear fits are displayed for clarity, and the red dotted line represents D^{avg}.

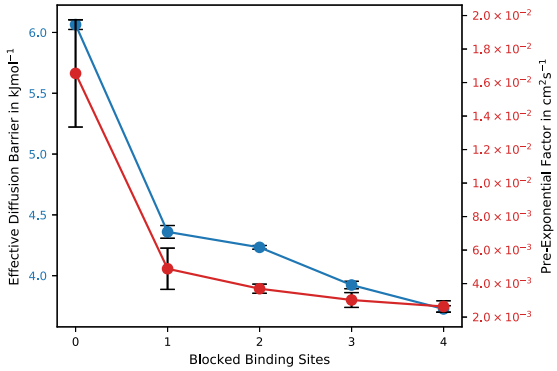


Figure 5. Pre-exponential factor, D_0, and effective diffusion barrier, ΔF, depending on the number of blocked sites. The blue line shows the effective diffusion barrier, while the red one represents the pre-exponential factor.

for the surface with the blocked deeper binding sites compared to the bare surface. All numerical values of diffusion coefficients at different substrate temperatures can be found in Table 1.

Fig. 5 shows the dependence of the pre-exponential factor D_0 and the effective barrier ΔF, obtained by fitting kMC values of diffusion coefficients by the Arrhenius equation from equation (3), on the number of occupied binding sites. The respective numerical values can be found in Table 1. From the figure, we see that the strongest decrease in both values was achieved already by excluding the deepest minimum in agreement with the observed increase of diffusion coefficients discussed above. The effective diffusion barrier decreases from 6.06 to 4.36 kJ mol^{-1}, which is now closer to the averaged one obtained in Section 3.1 ($\Delta F^{\mathrm{avg}} = 2.56 \pm 1.72$ kJ mol^{-1}), especially taking into account its large standard deviation. The pre-exponential factor decreases by one order of magnitude and remains almost unchanged after excluding the first binding site.

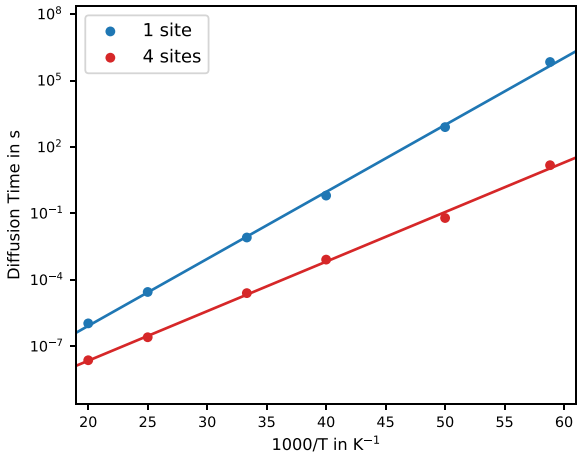


Figure 6. Diffusion times of the nitrogen atom adsorbed on top of the ASW surface until it is trapped by the deepest binding site (one site) or by one of the four deepest binding sites (four sites).

3.2.3 Diffusion times

Besides diffusion coefficients, we measured times the nitrogen could freely diffuse on the bare surface before being trapped by the deepest binding site or by one of the four deepest binding sites. Fig. 6 shows the temperature dependence of the diffusion time for both scenarios. We have found that the nitrogen atom is trapped by the deepest binding site already after 1.10×10^{-6} s at 50 K and after about 8 d at 17 K. The diffusion time depends exponentially on the inverse temperature, i.e. we may write

$$t = A \exp\left(\frac{B}{T}\right) , \tag{8}$$

and obtain parameters A and B by fitting the respective expression to measured data. The corresponding linear fits are presented in Fig. 6. For the nitrogen being trapped by the deepest binding site, we obtained $A = (7.61 \pm 3.05) \times 10^{-13}$ s and $B = (696.01 \pm 9.95)$ K. For the nitrogen being trapped by one of four deepest binding sites, we obtained $A = (7.65 \pm 3.53) \times 10^{-13}$ s and $B = (514.79 \pm 11.47)$ K.

Using the parameters A and B, we can compute the time the nitrogen atom will be able to freely diffuse until it reaches the deepest or one of four deepest binding sites. The respective values are 1.28×10^{18} and 1.74×10^{10} s (age of the Universe is 4×10^{17} s). This is equivalent to effective distances covered by the nitrogen atom ($4Dt = r^2$) of 4.22 and 5×10^{-4} Å, respectively. While the times are large, these distances are negligibly small compared to the size of typical dust grain of 1 μm.

4 DISCUSSION

Running the kMC simulations at $T = 50$ K, we obtain a diffusion coefficient of a nitrogen atom on ASW of 9.2×10^{-9} cm^2 s^{-1}, which is already low, but still even higher than the experimental results (Maté et al. 2020) obtained for CH_4, a molecule with very similar weight and affinity with the ASW to N, of 10^{-12} cm^2 s^{-1}. The influence of the temperature is severe, at $T = 17$ K we obtain $D = (6.9 \pm 2.2) \times 10^{-21}$ cm^2 s^{-1}, which is in much better agreement with estimates by He et al. (He, Emtiaz & Vidali 2018) of 10^{-20} cm^2 s^{-1} for CH_4 on ASW. Extrapolating to $T = 10$ K, the typical temperature in a molecular cloud, we arrive at $D = (3.5 \pm 1.1) \times 10^{-34}$ cm^2 s^{-1}, a very low value. Since the MSD covered by diffusion in two

dimensions is $\langle r^2 \rangle = 4Dt$, the average time for such a particle to scan a 1 μm dust grain is approx 10^{25} s, which is orders of magnitude longer than the age of the Universe (4×10^{17} s). Thus, a Langmuir–Hinshelwood mechanism of nitrogen atoms diffusing to meet reaction partners is found to be unlikely.

The temperature dependence of the diffusion constant nicely follows an Arrhenius-like behaviour of $D(T) = D_0 \exp(-\Delta F/RT)$ with $D_0 = (1.65 \pm 0.32) \times 10^{-2}$ cm^2 s^{-1} and $\Delta F = 6.06 \pm 0.04$ kJ mol^{-1} (see Fig. 3).

A closer analysis of our data reveals that the low diffusivity is caused by the domination of the strongest binding sites, as has been extensively discussed in Karssemeijer & Cuppen (2014), Senevirathne et al. (2017), and Ásgeirsson et al. (2017). Whenever the N adsorbate finds one of the deep binding sites, it stays there for a long time. Our surface model is, with about 800 Å^2, comparably small and certainly very compact and smooth. In real ASW, much deeper pores are expected, which should lead to even stronger binding. Thus, on a more realistic surface, the diffusion can be expected to be even slower. However, deep sites may be blocked by other adsorbates. That can be easily simulated with our model. Blocking one to four of the deepest sites results in much higher values for D of 8.2×10^{-26} to 9.0×10^{-23} cm^2 s^{-1} at 10 K, respectively. The temperature dependence of D with the deepest sites blocked is illustrated in Fig. 4. We expect that, under interstellar medium conditions, deep sites will be covered by other species, i.e. H_2, with a consequent increment in D.

Diffusion on the surface may be accelerated by quantum mechanical tunnelling of the adsorbate. We took that into account in the kMC simulations by correcting the rate constants with the transmission coefficients of Eckart and Bell barriers fitted to the free-energy barriers. The effect by atom tunnelling is marginal, even at 10 K: The diffusion constant is increased by a factor of merely 1.3–2.8. Thus, we find tunnelling not to be relevant for nitrogen atom diffusion at 10 K, which can be expected from the high mass of N, which agrees with previous results (Pezzella et al. 2018) but is at variance with a previous suggestion for oxygen atoms (Minissale et al. 2013).

Is it justifiable to consider diffusion to be fast under a homogeneous regime, as considered by some theoretical models? A common attempt is to describe diffusion via hopping rates from one minimum to a neighbouring one. This may be a justified model if all barriers and all binding sites were similar, like on a crystalline surface. However, even on our comparably small surface model, the barriers of the individual hopping events vary from 0.05 to 7.8 kJ mol^{-1} with an average of 2.56 kJ mol^{-1}, which result in hopping rate constants with a huge variation from 3.8×10^{-30} to 1.1×10^{11} s^{-1} at 10 K. Depending on the surface morphology, some of the higher barriers may be circumvented by the diffusion path. However, the smallest barriers usually merely lead to oscillations between neighbouring binding sites rather than to real transport of the adsorbate. Overall, a model that takes the connectivity between binding sites with their realistic barriers into account, like our kMC model, is required to estimate the diffusivity accurately. Effective diffusion barriers (ΔF) should be estimated from such models.

Most of these arguments have been previously used in the theoretical study of H atoms diffusion on ASW versus polycrystalline ice (Kuwahata et al. 2015; Ásgeirsson et al. 2017; Senevirathne et al. 2017) and hold here. In the picture presented in Hama et al. (2012), H atoms block the deep binding sites and recombination of H_2 is possible from another incoming H atom. In the case of nitrogen, however, this mechanism is less likely. Given the similar abundances of H and N in molecular clouds, in the event of complete coverage of binding sites, H atoms diffusion will surpass N atoms diffusion. However, a smaller fraction of N chemistry due to diffusion cannot be discarded under the high-coverage regime.

In the limit of low coverage of ices, we conclude that reactions via the Langmuir–Hinshelwood mechanism with N are extremely unlikely at 10 K. A closer look at other diffusive mechanisms is warranted in this context. Recently, we have shown that hot-atom diffusion just after exothermic adsorption is a very short-ranged process, with the adsorbate molecules moving only for about 1–2 ps before they become thermalized (Molpeceres et al. 2020b). They thermalize anywhere on the surface, not necessarily in deep sites. However, deep binding sites are abundant enough to block any significant spatial movement of the adsorbate effectively. As mentioned before, a real surface will be even rougher than our model, so more deep binding sites are expected. After trapping, the adsorbate is effectively removed from the reaction, waiting for an additional reaction partner.

In the limit of high coverage, on the other hand, diffusion of N could, in principle, proceed. Our diffusion constants at 25 K for high coverage are comparable with those of H at 25 K at low coverage (Ásgeirsson et al. 2017), indicating the extreme importance of the number of adsorbates pre-adsorbed on the surface. In experiments, even under the sub-monolayer regime, deep-binding sites are readily occupied. Hence the diffusion of reactive species (such as O or N) is measured as an upper limit of the range of possible diffusion coefficients.

Above which temperature can we expect diffusion to become relevant for thermal diffusion of N on a pristine ASW surface? While it is difficult to assign an absolute number to D sufficient for surface reactivity, we can use the hydrogen atom on ASW as an estimate. $D_H = 1.09 \times 10^{-5}$ cm^2 s^{-1} was obtained at 10 K on ASW (Al-Halabi & Van Dishoeck 2007), a value later claimed (Hama et al. 2012) to be somewhat high and recently corrected by the values of Ásgeirsson et al. (2017) to $D_H = 5.80 \times 10^{-11}$ cm^2 s^{-1} at 25 K in the limit of low coverage of the surface and $D_H = 3.30 \times 10^{-7}$ cm^2 s^{-1} at the same temperature when the deepest adsorption sites are occupied. To match the results at 10 K of Al-Halabi & Van Dishoeck (2007), a temperature of about 100 K is necessary, while to match the results of Ásgeirsson et al. (2017) at 25 K for a bare surface, a temperature close to 40 K would be necessary.

We finally want to raise awareness of issues arising from the use of average diffusion barriers as fractions of the average binding energy, which is common practice in astrochemical models. From our calculations of the diffusion barrier, we arrived at an average value of 2.56 kJ mol^{-1}, which is 0.76 times the average binding energy. The values that we obtained from the Arrhenius fit in Table 1, however, show that the effective barrier for diffusion varies between 3.73–6.06 kJ mol^{-1} (a ratio of 1.1–1.8, using binding energies without zero-point vibrational energies) due to the prevalence of deep binding sites. Such value can decrease below 1.0 for higher surface coverage, as discussed above, meaning that the use of $\Delta F^{\rm avg}$ should be only justified in this context. It is worth mentioning, however, that previous modelling studies equivalent to the study presented here found diffusion/binding energy ratios below 1.0 (Karssemeijer & Cuppen 2014; Ásgeirsson et al. 2017), so further investigation is highly desirable.

5 CONCLUSION

In light of our simulations and in accordance with Pezzella et al. (2018), we conclude that diffusion of nitrogen atoms and implicitly

of reactive species heavier and tighter bound by water ice (which, to the best of our knowledge, includes most reactive species) is hindered for a wide range of conditions, and hydrogen diffusion must dominate surface chemistry in these environments. When the surface of the ice is sufficiently populated by other adsorbates, the diffusion of N may be enabled. Alternatively, non-thermal diffusion after hydrogenation can be invoked to explain the formation of molecules such as CO_2 (Ioppolo et al. 2011), organic alcohols (Qasim et al. 2019b), formaldehyde (Qasim et al. 2019a), or, very recently, glycine (Ioppolo et al. 2021). We also emphasize here that the importance of non-diffusive mechanisms (Eley–Rideal) should also be re-evaluated in the context of the formation of COMs (Herbst & van Dishoeck 2009; He et al. 2017).

ACKNOWLEDGEMENTS

We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding EXC 2075 – 390740016 under Germany's Excellence Strategy. We acknowledge the support by the Stuttgart Center for Simulation Science (SimTech) and the European Union's Horizon 2020 research and innovation programme (grant agreement No. 646717, TUNNELCHEM). We also like to acknowledge the support by the Institute for Parallel and Distributed Systems (IPVS) of the University of Stuttgart and by the state of Baden-Württemberg through the bwHPC consortium for providing computer time. GM thanks the Alexander von Humboldt Foundation for a postdoctoral research fellowship. VZ acknowledges the financial support received in the form of a Ph.D. scholarship from the Studienstiftung des Deutschen Volkes (German National Academic Foundation).

DATA AVAILABILITY

The data obtained in this paper will be shared at reasonable request to the corresponding author. All binding sites and transition states used for kMC simulations can be found in Supporting Information.

REFERENCES

Al-Halabi A., Van Dishoeck E. F., 2007, MNRAS, 382, 1648
Allnatt A. R., Lidiard A. B., 1993, Atomic Transport in Solids. Cambridge Univ. Press, Cambridge
Ásgeirsson V., Jónsson H., Wikfeldt K. T., 2017, J. Phys. Chem., 121, 1648
Barducci A., Bussi G., Parrinello M., 2008, Phys. Rev. Lett., 100, 020603
Bonomi M. et al., 2009, Comput. Phys. Commun., 180, 1961
Bortz A. B., Kalos M. H., Lebowitz J. L., 1975, J. Comput. Phys., 17, 10
Bossa J.-B., Maté B., Fransen C., Cazaux S., Pilling S., Rocha W. R. M., Ortigoso J., Linnartz H., 2015, ApJ, 814, 47
Du Y. A., Rogal J., Drautz R., 2012, Phys. Rev., 86, 174110
Gillespie D. T., 1976, J. Comput. Phys., 22, 403
Grimme S., Brandenburg J. G., Bannwarth C., Hansen A., 2015, J. Chem. Phys., 143, 054107
Hama T., Kuwahata K., Watanabe N., Kouchi A., Kimura Y., Chigai T., Pirronello V., 2012, ApJ, 757, 185
He J., Emtiaz S. M., Vidali G., 2017, ApJ, 851, 104
He J., Emtiaz S., Vidali G., 2018, ApJ, 863, 156
Henkelman G., Jónsson H., 2000, J. Chem. Phys., 113, 9978
Henkelman G., Uberuaga B. P., Jónsson H., 2000, J. Chem. Phys., 113, 9901
Herbst E., van Dishoeck E. F., 2009, ARA&A, 47, 427
Hjorth Larsen A. et al., 2017, J. Phys.: Condens. Matter., 29, 273002
Huber T., Torda A. E., van Gunsteren W. F., 1994, J. Comput.-Aided Mol. Des., 8, 695
Ioppolo S., van Boheemen Y., Cuppen H. M., van Dishoeck E. F., Linnartz H., 2011, MNRAS, 413, 2281
Ioppolo S. et al., 2021, Nature Astron., 5, 197
Jónsson H., Mills G., Jacobsen K. W., 1998. Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions. World Scientific Publishing Co. Pte. Ltd., p. 385. https://www.worldscientific.com/doi/abs/10.1142/9789812839664_0016
Karssemeijer L. J., Cuppen H. M., 2014, A&A, 569, A107
Karssemeijer L. J., Ioppolo S., Van Hemert M. C., Van Der Avoird A., Allodi M. A., Blake G. A., Cuppen H. M., 2014, ApJ, 781, 16
Kästner J., Carr J. M., Keal T. W., Thiel W., Wander A., Sherwood P., 2009, J. Phys. Chem., 113, 11856
Kirchheim R., 1987, Acta Metall., 35, 271
Kirchheim R., 1988, Prog. Mater. Sci., 32, 261
Kuwahata K., Hama T., Kouchi A., Watanabe N., 2015, Phys. Rev. Lett., 115, 133201
Laio A., Parrinello M., 2002, Proc. Natl. Acad. Sci., 99, 12562
Maté B., Cazaux S., Satorre M. Á., Molpeceres G., Ortigoso J., Millán C., Santonja C., 2020, A&A, 643, A163
McConnell S., Kästner J., 2017, J. Comput. Chem., 38, 2570
Minissale M. et al., 2013, Phys. Rev. Lett., 111, 053201
Minissale M., Congiu E., Dulieu F., 2016, A&A, 585, A146
Molpeceres G., Zaverkin V., Kästner J., 2020a, https://doi.org/10.5281/zenodo.4013889, last accessed on 29 December 2021
Molpeceres G., Zaverkin V., Kästner J., 2020b, MNRAS, 499, 1373
Pezzella M., Unke O. T., Meuwly M., 2018, J. Phys. Chem. Lett., 9, 1822
Qasim D., Fedoseev G., Lamberts T., Chuang K. J., He J., Ioppolo S., Kästner J., Linnartz H., 2019a, ACS Earth Space Chem., 3, 986
Qasim D., Lamberts T., He J., Chuang K. J., Fedoseev G., Ioppolo S., Boogert A. C., Linnartz H., 2019b, A&A, 626, A118
Ramasubramaniam A., Itakura M., Ortiz M., Carter E., 2008, J. Mater. Res., 23, 2757
Ruaud M., Loison J. C., Hickson K. M., Gratier P., Hersant F., Wakelam V., 2015, MNRAS, 447, 4004
Ruaud M., Wakelam V., Hersant F., 2016, MNRAS, 459, 3756
Senevirathne B., Andersson S., Dulieu F., Nyman G., 2017, Mol. Astrophys., 6, 59
Shimonishi T., Nakatani N., Furuya K., Hama T., 2018, ApJ, 855, 27
Snow T. P., McCall B. J., 2006, ARA&A, 44, 367
The PLUMED consortium et al., 2019, Nature Methods, 16, 670
Tribello G. A., Bonomi M., Branduardi D., Camilloni C., Bussi G., 2014, Comput. Phys. Commun., 185, 604
Tsong T. T., 2001, Prog. Surf. Sci., 67, 235
Voter A. F., 2007, Introduction to the Kinetic Monte Carlo Method. Springer, Dordrecht
Zaverkin V., Kästner J., 2020, J. Chem. Theory Comput., 16, 5410
Zwanzig R., 1988, Proc. Natl. Acad. Sci. USA, 85, 2029

SUPPORTING INFORMATION

Supplementary data are available at *MNRAS* online.

Please note: Oxford University Press is not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for this paper.

This paper has been typeset from a TEX/LATEX file prepared by the author.

Erklärung über die Eigenständigkeit der Dissertation

Ich versichere, dass ich die vorliegende Arbeit mit dem Titel *Investigation of Chemical Reactivity by Machine-Learning Techniques* selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe; aus fremden Quellen entnommene Passagen und Gedanken sind als solche kenntlich gemacht.

Declaration of Authorship

I hereby certify that the dissertation entitled *Investigation of Chemical Reactivity by Machine-Learning Techniques* is entirely my own work except where otherwise indicated. Passages and ideas from other sources have been clearly indicated.

Stuttgart, der 29. April 2022

Viktor Zaverkin