Technische Universität München
TUM School of Engineering and Design

Advances in Neural Network Potentials for Molecular Dynamics Simulations: Physics-Informed Training and Uncertainty Quantification

Stephan Thaler, M.Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design der Technischen Universität München zur Erlangung eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigten Dissertation.

Vorsitz: Prof. Dr.-Ing. Nikolaus A. Adams

Prüfer*innen der Dissertation:

1. Prof. Dr. Julija Zavadlav
2. Prof. Phaedon-Stelios Koutsourelakis, Ph.D.
3. Prof. Matej Praprotnik, Ph.D.

Die Dissertation wurde am 26.06.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 01.10.2023 angenommen.

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble."

– Paul Dirac, 1929

Abstract

Molecular dynamics (MD) simulations are a cornerstone of material science: By enabling experiments in-silico, MD simulations can support the screening of a large database of candidate compounds for applications in material design and drug discovery. However, the accuracy and reliability of MD simulations depends critically on the choice of the potential energy function that defines molecular interactions. Neural network (NN) potentials are promising due to their ability to represent many-body interactions and their large model capacity. Their accuracy is primarily limited by the quality and quantity of the available training data.

This thesis presents a series of methodological advancements aimed at maximizing the information gain from the available training data and allowing for trustworthy NN potential-based MD simulations: The first paper introduces the Differentiable Trajectory Reweighting (DiffTRe) method that facilitates training NN potentials on experimental data. In particular, DiffTRe allows the combination of experimental and first principles data, which is particularly relevant for larger systems that are inaccessible to accurate computational quantum mechanics simulations.

The second paper demonstrates that training NN potentials via relative entropy (RE) minimization is a highly data efficient training scheme that also corrects for numerical errors by sampling from the NN potential via an MD simulation during training. Thus, RE minimization enables more accurate coarse-grained MD simulations while reducing the computational effort for data generation.

The third paper shows that scalable uncertainty quantification (UQ) schemes for NN potentials allow the estimation of reliable credible intervals of MD observables. Specifically, both the Deep Ensemble method and stochastic gradient Markov chain Monte Carlo (SG-MCMC) with multiple Markov chains are found to be reliable UQ schemes in this context. However, it was also shown that further research into SG-MCMC schemes is needed in order to leverage the theoretical advantage of additional sampling of the posterior volume. To this end, the fourth paper introduced the *JaxSGMC* library, which aims to accelerate the development of novel SG-MCMC samplers through reusable algorithmic building blocks. Additionally, *JaxSGMC* promotes Bayesian UQ of NNs by providing an easy-to-use interface to state-of-the-art SG-MCMC samplers.

The fifth and final paper presents the Energy Minimized Atomistic Insertion (EMATI) method, which reduces interface artefacts in the Adaptive Resolution Scheme. EMATI achieves this by inserting atoms based on the local chemical environment rather than randomly. Minimizing interface artefacts is a prerequisite for using accurate NN potentials in concurrent multiscale MD simulations, such that large interface errors do not outweigh the increased accuracy of NN potentials.

In sum, the methodological advancements presented in this thesis pave the way towards accurate and reliable NN potential-based MD simulations, which support real-world decision-making in a broad range of material science applications.

Kurzfassung

Moleküldynamiksimulationen (MD) sind ein Eckpfeiler der Materialwissenschaften, weil durch in-silico Experimente das Screening von großen Datenbanken von Materialkandidaten unterstützt werden kann. Dies lässt sich für Anwendungen im Materialdesign und in der Arzneimittelentdeckung nutzen. Die Genauigkeit und Zuverlässigkeit von MD-Simulationen hängt jedoch entscheidend von der Wahl der potenziellen Energiefunktion ab, welche die molekularen Wechselwirkungen definiert. Neuronale Netzwerkpotentiale (NNPs) sind vielversprechend, da sie in der Lage sind, Mehrkörper-Wechselwirkungen darzustellen und eine große Modellkapazität aufweisen. Ihre Genauigkeit wird in erster Linie durch die Qualität und Quantität der verfügbaren Trainingsdaten begrenzt.

In dieser Dissertation werden eine Reihe von methodischen Entwicklungen vorgestellt, die darauf abzielen, den Informationsgewinn aus den verfügbaren Trainingsdaten zu maximieren und vertrauenswürdige, auf NNPs basierende MD-Simulationen zu ermöglichen: Der erste Forschungsartikel stellt die Differentiable Trajectory Reweighting (DiffTRe) Methode vor, die das Training von NNPs anhand experimenteller Daten erleichtert. DiffTRe ermöglicht insbesondere die Kombination von experimentellen und Quantenmechaniksimulationsdaten, was besonders für größere Systeme von Bedeutung ist, für die genaue computergestützte quantenmechanische Simulationen nicht durchführbar sind.

Im zweiten Artikel wird gezeigt, dass das Training von NNPs durch Minimierung der relativen Entropie (RE) ein äußerst dateneffizientes Trainingsverfahren ist, das außerdem numerische Fehler korrigiert, indem während des Trainings das NNP für MD-Simulationen verwendet wird. Somit ermöglicht die RE-Minimierung genauere grobskalige MD-Simulationen bei gleichzeitiger Reduzierung des Rechenaufwands für die Datenerzeugung.

Das dritte Paper zeigt, dass skalierbare Verfahren zur Unsicherheitsquantifizierung (UQ) für NNPs die Prädiktion zuverlässiger Glaubwürdigkeitsintervalle für MD-Observables ermöglichen. Insbesondere die Deep-Ensemble-Methode und die Stochastische-Gradienten-Markov-Ketten-Monte-Carlo-Methode (SG-MCMC) mit mehreren Markov-Ketten erweisen sich in diesem Zusammenhang als zuverlässige UQ-Verfahren. Es wurde jedoch auch gezeigt, dass weitere Forschung zu SG-MCMC Methoden erforderlich ist, um den theoretischen Vorteil der zusätzlichen Auflösung des Posterior-Volumens zu nutzen. Zu diesem Zweck wurde im vierten Artikel die Softwarebibliothek *JaxSGMC* vorgestellt, die die Entwicklung neuartiger SG-MCMC-Sampler durch wiederverwendbare algorithmische Bausteine beschleunigen soll. Darüber hinaus fördert *JaxSGMC* die bayessche UQ von neuronalen Netzwerken, indem es eine einfach zu bedienende Schnittstelle für moderne SG-MCMC-Sampler bietet.

Im fünften und letzten Artikel wird die Energy Minimized Atomistic Insertion (EMATI) Methode vorgestellt, die Artefakte an der Schnittstelle der Auflösungen im Adaptive Resolution Scheme reduziert. EMATI erreicht dies, indem es die Atome nicht zufällig sondern basierend auf der lokalen chemischen Umgebung einfügt. Die Minimierung von Artefakten an der Auflösungsschnittstelle ist eine Voraussetzung für die Verwendung von präzisen NNPs in Mehrskalen-MD-Simulationen, sodass große Fehler an der Auflösungsschnittstelle die erhöhte Genauigkeit der NNPs nicht überlagern.

Zusammenfassend lässt sich sagen, dass die in dieser Dissertation vorgestellten methodischen Fortschritte den Weg hin zu genauen und zuverlässigen, auf NNPs basierenden MD-Simulationen ebnen, die reale Entscheidungsprozesse in einem breiten Spektrum an materialwissenschaftlichen Anwendungen unterstützen können.

Preface

This cumulative dissertation builds on the following original journal articles:

Journal articles (peer-reviewed)

[1] **Thaler, S.**, Praprotnik, M. & Zavadlav, J. Back-mapping augmented adaptive resolution simulation. *J. Chem. Phys.* **153**, 16, 164118 (2020), DOI: 10.1063/5.0025728.

[2] **Thaler, S.** & Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. *Nat. Commun.* **12**, 6884 (2021), DOI: 10.1038/s41467-021-27241-4.

[3] **Thaler, S.**, Stupp, M. & Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *J. Chem. Phys.* **157**, 24, 244103 (2022), DOI: 10.1063/5.0124538.

[4] **Thaler, S.**, Doehner, G. & Zavadlav, J. Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls. *J. Chem. Theory Comput.* **19**, 14, 4520–4532 (2023), DOI: 10.1021/acs.jctc.2c01267.

Journal articles (submitted)

[5] **Thaler, S.***, Fuchs, P.*, Cukarska, A. & Zavadlav, J. JaxSGMC: Modular stochastic gradient MCMC in JAX. (2023), URL: https://github.com/tummfm/jax-sgmc (in review process *SoftwareX*).

* denotes shared first authorship

Additional topic-related conference contributions:

Conference contributions (peer-reviewed)

[6] **Thaler, S.** & Zavadlav, J. Uncertainty Quantification for Molecular Models via Stochastic Gradient MCMC. *In 10th Vienna Conference on Mathematical Modelling*, 19–20 (Vienna, Austria, July 27–29, 2022), DOI: 10.11128/arep.17.a17046.

Contents

Acronyms and Abbreviations

ACE	atomic cluster expansion
AD	automatic differentiation
AdResS	Adaptive Resolution Scheme
AIMD	ab initio molecular dynamics
AL	active learning
AT	atomistic
CG	coarse-grained
COM	center of mass
CQM	computational quantum mechanics
DFT	density-functional theory
DiffTRe	Differentiable Trajectory Reweighting
EAM	Embedded Atom Model
EMATI	Energy Minimized Atomistic Insertion
FC	force capping
FES	free energy surface
FM	force matching
GNN	graph neural network
GP	Gaussian process
HMC	Hamiltonian Monte Carlo
KL	Kullback-Leibler
LJ	Lennard Jones
MCMC	Markov chain Monte Carlo
MD	molecular dynamics
MH	Metropolis-Hastings
ML	machine learning
MLP	multilayer perceptron
MOF	metal-organic framework
MSE	mean squared error
NN	neural network
NUTS	No-U-Turn Sampler
PMF	potential of mean force
pSGLD	preconditioned Stochastic Gradient Langevin Dynamics
RE	relative entropy
SG-MCMC	stochastic gradient Markov chain Monte Carlo
UQ	uncertainty quantification

1. Introduction

Designing materials tailored to specific applications is a long-standing vision in material science. For instance, highly customizable materials such as Metal-organic frameworks (MOFs) [7], Perovskites [8, 9] and Zeolites [10] promise significant progress in applications including hydrogen storage, high-efficiency photovoltaics and catalysis. Due to the vast chemical design space of these materials, finding the optimal compound via exhaustive synthesis and experimentation is intractable. Hence, compounds need to be investigated in-silico, which allows screening a large database of candidate materials to maximize target properties while satisfying application-specific constraints [11–15]. If successful, only a small number of the most promising candidates need to be investigated experimentally, resulting in substantial savings in development time and cost [16, 17]. The screening of a large database of potential small-molecule inhibitors against SARS-CoV-2 represents a recent high-impact application example [18]. However, the success of this in-silico screening approach hinges on the quality of the employed computational models predicting material properties [17]. Hence, accurate and reliable molecular modeling is critical for decision-making in practice.

1.1. Molecular Modeling

Materials can be modeled at various resolutions, where the specific modeling technique determines the accessible time and length scales of the simulation (fig. 1.1).

Molecular Modeling Scales

Small systems can be represented by their sub-atomic components and treated quantum-mechanically. The approximate solution of the Schrödinger equation via computational quantum mechanics (CQM) enables computation of the potential energy of the system. This potential energy can be used in a molecular dynamics (MD) simulation to compute the time evolution of the modeled system. As the potential energy is computed from first principles, this approach is referred to as ab initio molecular dynamics (AIMD) [21, 22]. AIMD enables the prediction of material properties from first principles only [23], but the size of the investigated systems is significantly limited by the costly CQM simulation at each MD time step [24].

To reduce the large computational effort from CQM, atomistic (AT) models approximate the potential energy of the system via a so-called semi-empirical potential energy function. This potential energy function depends on atomic properties only, i.e. there is no dependence on sub-atomic particles. Given that the evaluation of the semi-empirical potential energy function requires orders of magnitude less computational effort than a CQM simulation, AT MD simulations can access much larger time and length scales (fig. 1.1), rendering

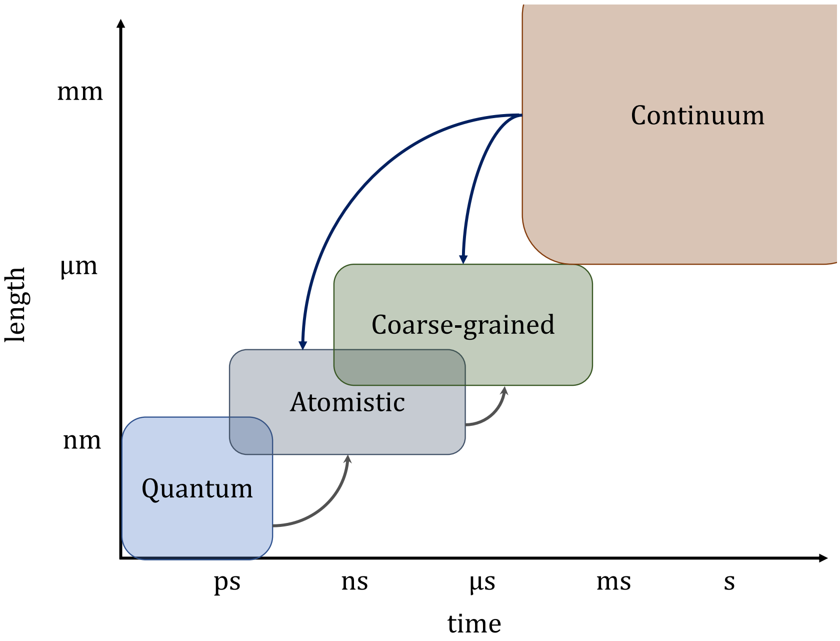


Figure 1.1.: Typical time and length scales for different modeling techniques [19, 20]. The typical flow of information for molecular modeling is visualized via blue (top-down) and gray arrows (bottom-up).

it a method of choice to investigate structural, thermodynamic, mechanic and dynamic properties of materials [25].

To model even larger systems and longer time scales, e.g. biophysical systems [20], groups of atoms can be coarse-grained (CG) into effective interaction beads [26]. The speed-up over AT models results from the smaller number of CG particles reducing the computational cost per potential energy evaluation and a larger admissible MD time step due to smoother dynamics [27].

The potential energy function is at the core of AT and CG MD simulations because it encodes the way particles interact. Hence, designing accurate and computationally efficient potentials is at the center of molecular modeling. The two main building blocks that define the potential energy function are the training data and the employed functional form.

Potential: Training Data

In principle, there are two sources of information available to optimize the parameters of the potential energy function: the potential energy computed from first principles as well as experimental data. To optimize the potential based on the former, a large data set of molecular structures needs to be generated, ideally containing molecular states at different thermodynamic state points [26, 28]. Due to a favourable trade-off between accuracy and computational cost, density-functional theory (DFT) [29, 30] is typically employed to label the data set, i.e., to approximate the potential energy and forces of the molecular state. For CG modeling, it is common to consider a well-tested AT force field as the ground truth model. Then, given the labelled data set, the most widespread scheme to optimize the parameters of the potential is energy [28] and/or force matching (FM) [31–33], where the optimal potential minimizes the difference between the predicted energy/forces and the target energy/forces for each molecular state in the data set.

In contrast to matching the predictions of a data-generating high-fidelity simulation (bottom-up learning), the goal of top-down learning is to parametrize the potential such that when used in an MD simulation, the result matches the reference experiments (typically conducted at the continuum scale) [26, 34]. Hence, leveraging information gathered at different scales is the heart of AT and CG molecular modeling (fig. 1.1).

Potential: Functional Form

The functional form of the potential determines the computational cost as well as the maximum achievable accuracy of an MD simulation by defining which molecular interactions can be represented. Classical force fields such as AMBER [35] and GROMOS [36] at the AT scale as well as MARTINI [37] at the CG scale are composed of physics-inspired terms such as the Lennard-Jones potential for non-bonded interactions as well as harmonic bonds, angles and dihedrals for intramolecular interactions. Classical force fields are computationally efficient and have been applied successfully to model a wide range of systems [38]. However, in some cases, obtained results differ significantly from experiments, in particular for chemically complex systems with strong polarization effects, bond breaking, etc. [38–40]. Even through reactive [41] and polarizable [42, 43] force fields address some of these shortcomings, classical force fields remain inherently limited by their rather simple functional form, which limits its ability to represent the higher body-order terms of solutions of the Schrödinger equation [44, 45].

Machine learning (ML) potentials promise much higher accuracy by replacing functional forms based on physical heuristics by highly flexible many-body function approximators. The most common ML potentials are based on Gaussian Processes (GP) [46–50] and neural networks (NN) [51, 52]. While GPs have the advantage of providing an uncertainty estimate of the predicted potential energy at no additional cost, they do not scale well to large data sets [46, 49] common in molecular modeling [53, 54]. Hence, this thesis focuses on NN potentials, which have found great success within the last few years, both in terms of method development [55–61] and in terms of enabling novel computational studies [44, 62–65]. In particular, NN potentials have delivered on the promise of providing functional forms with very high model capacity, with recent NN potentials achieving two orders of magnitude smaller test errors within the training data distribution compared to the expected DFT error [66, 67].

1.2. Research Objectives

In principle, the large model capacity of NN potentials enables modeling of molecular systems at unprecedented accuracy. However, there are several issues in practical application of NN potentials that hamper more widespread adoption: First, NN potentials are data-driven black-box models. Their flexible functional form comes at the cost of losing physically-reasonable constraints of classical models. Thus, when applied outside their training domain, NN potential predictions can be highly inaccurate or even qualitatively unphysical [2, 68, 69]. As a consequence, practitioners may prefer less accurate, but more constrained potentials in practice [70].

Second, generating sufficiently large data sets with broad coverage is challenging due to the high dimensionality of chemical space and the computational cost of labelling molecular states via DFT [61, 69, 71]. This problem is exacerbated by the property of MD simulations to evolve the simulation state along the adversarial direction of the potential, preferentially sampling chemical space where the NN is least accurate [57]. Accordingly, NN potentials are typically data-constrained, i.e. the availability of accurate data sets relevant for the molecular system at hand tends to be more important for model performance than the specific NN architecture.

Third, the vast majority of NN potentials are trained bottom-up. For bottom-up training, the maximum accuracy of NN potentials is limited to the accuracy of the data-generating high-fidelity simulation. Given that DFT only provides approximate solutions to Schrödinger's equation, some deviation from experimental results is to be expected, even in the limit of infinite training data and NN model capacity. While more accurate CQM methods such as coupled cluster CCSD(T) [72], quantum Monte Carlo [73, 74] and NN-based methods [75, 76] can be expected to reduce the deviation from experiments, they are significantly more expensive, exacerbating the data-coverage issue [64, 77]. Hence, the goal of this thesis is to address the following research questions:

- **Training on Experimental Data**

  Top-down learning circumvents the problem of the limited accuracy of the data-generating simulation by directly matching experimental observations. The importance of top-down training has been recognized in the development of classical MD force fields, which typically optimize both bottom-up and top-down objective functions [35–38]. While classical force fields can be optimized via gradient-free optimization schemes [78, 79], NN potential training requires gradient information due to the curse of dimensionality associated with the large amount of learnable NN parameters. To leverage the power of top-down training in the context of NN potentials, it is therefore imperative to develop an efficient gradient-based training scheme that also integrates well with auto-differentiable NN frameworks such as PyTorch [80], Tensorflow [81] or JAX [82].

- **Leveraging Alternative Training Schemes**

  For bottom-up training, the majority of NN potentials are trained via energy matching and/or FM [28]. This training scheme is computationally efficient and straightforward to implement in NN frameworks, but it features drawbacks when applying the learned NN potential in MD simulations. Given that data sets for bottom-up training are often generated via MD simulations, they usually contain predominantly states close to energy minima, but less high-energy states and no unphysical configurations. Consequently, phase-space regions further away from energy minima may be insufficiently represented in the data set. As FM training relies on sufficient sampling of all relevant phase-space regions, this renders FM comparatively data inefficient [83]. Additionally, FM training cannot constrain the potential in unphysical phase-space regions that are not included in the training data. To avoid entering unphysical phase-space regions during an MD simulation, NN potentials trained via FM heavily rely on physics-inspired prior potentials that assign high potential energy to unphysical configurations [68, 69]. To achieve higher data efficiency and less dependence on prior potentials compared to FM, alternative training schemes that leverage MD simulations during training need to be investigated.

- **Evaluating Predictive Uncertainty**

  Active learning (AL) [84, 85] promises efficient generation of more diverse data sets. Instead of labelling every state along an MD trajectory, AL is an iterative method that only labels molecular states for which the estimated uncertainty of the model exceeds a predefined threshold. Afterwards, these high uncertainty states are added to the existing data set, the potential is retrained and a new MD trajectory is generated using the retrained model. The AL loop ends once the uncertainties of all generated states are below the uncertainty threshold. As a side effect, AL guarantees that the generated MD trajectory only sampled low-uncertainty phase-space regions, increasing the trustworthiness of obtained MD results. AL has been applied successfully in practice [65, 86, 87], but the efficiency of AL critically depends on the quality of uncertainty estimates. In particular, the most common uncertainty quantification (UQ) scheme for NN potentials, Deep Ensembles [88, 89], was attributed rather poor uncertainty estimates for active learning applications [90].

  While AL is an efficient means to generate data sets and reliable MD results when the application is known at training time [65], practitioners may be interested in directly applying pre-trained NN potentials without any re-training. Given that NN potentials are unreliable outside the training data distribution and estimating whether the conducted MD simulation visited out-of-distribution phase-space regions is non-trivial, obtaining uncertainty estimates of MD observables is essential to establish trust in NN potential-based MD results. Hence, investigating the quality of UQ schemes for NN potentials is essential, both for efficient AL as well as for reliable UQ of MD observables. The latter are a prerequisite for using NN potential-based MD simulation results in industrial decision-making.

1.3. Structure

This cumulative dissertation is structured as follows: Chapter 2 outlines main components of parametrizing molecular force fields with corresponding theory and a focus on NN potentials. This molecular modeling framework provides the foundation for the contributions of the research papers summarized and embedded in chapter 3. Finally, chapter 4 summarizes and discusses the contributions of this thesis and provides an outlook on further research directions.

2. Methods

This chapter introduces the main components of the molecular modeling framework (fig. 2.1), which form the basis of the contributions in chap. 3.

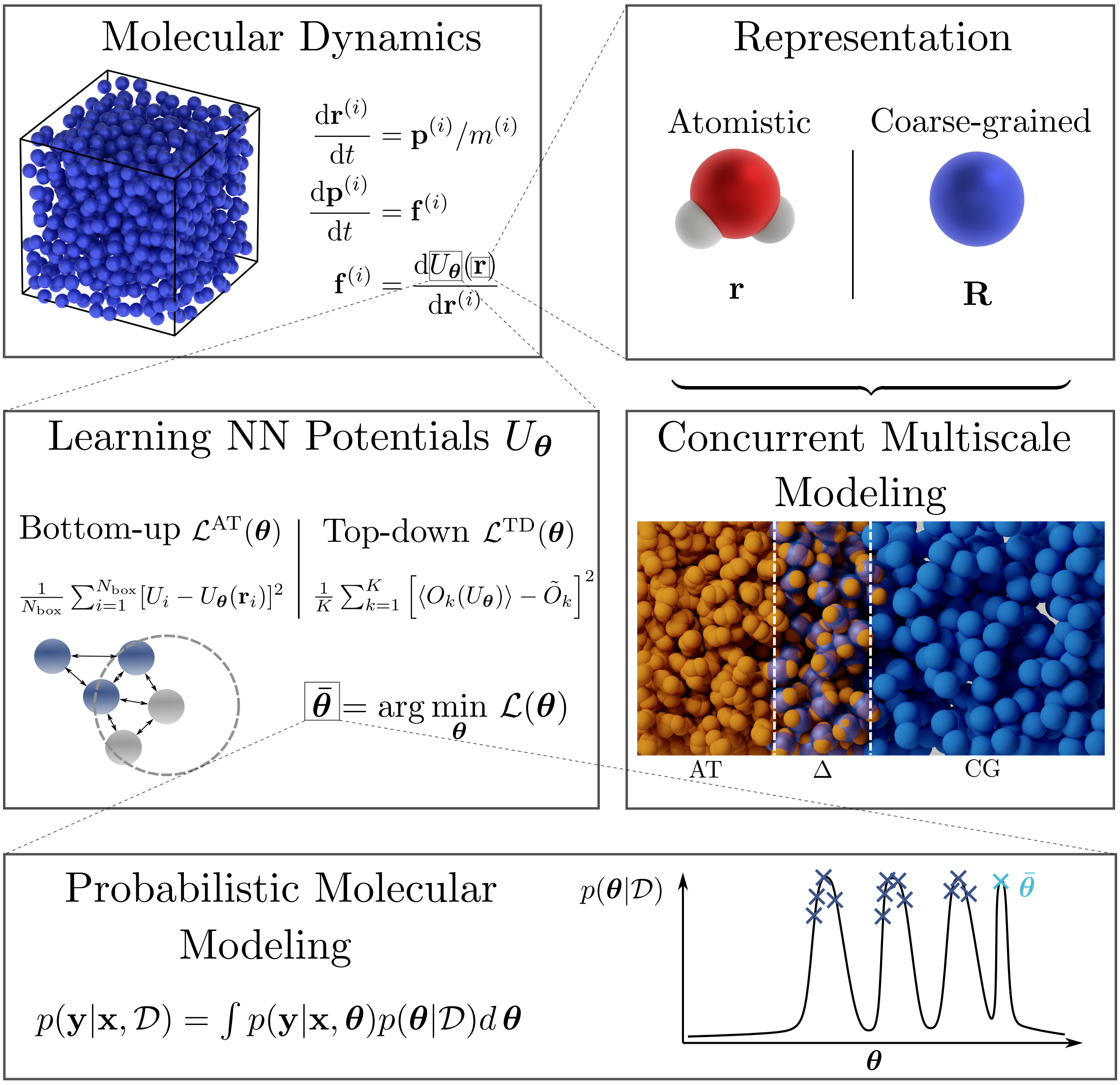

Molecular Dynamics
$\frac{\mathrm{d}\mathbf{r}^{(i)}}{\mathrm{d}t} = \mathbf{p}^{(i)}/m^{(i)}$
$\frac{\mathrm{d}\mathbf{p}^{(i)}}{\mathrm{d}t} = \mathbf{f}^{(i)}$
$\mathbf{f}^{(i)} = \frac{\mathrm{d}\boxed{U_{\boldsymbol{\theta}}}(\boxed{\mathbf{r}})}{\mathrm{d}\mathbf{r}^{(i)}}$
Representation
Atomistic
Coarse-grained
$\mathbf{r}$
$\mathbf{R}$
Learning NN Potentials $U_{\boldsymbol{\theta}}$
Bottom-up $\mathcal{L}^{\mathrm{AT}}(\boldsymbol{\theta})$
Top-down $\mathcal{L}^{\mathrm{TD}}(\boldsymbol{\theta})$
$\frac{1}{N_{\mathrm{box}}}\sum_{i=1}^{N_{\mathrm{box}}}[U_i - U_{\boldsymbol{\theta}}(\mathbf{r}_i)]^2$
$\frac{1}{K}\sum_{k=1}^{K}\left[\langle O_k(U_{\boldsymbol{\theta}})\rangle - \tilde{O}_k\right]^2$
$\boxed{\bar{\boldsymbol{\theta}}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$
Concurrent Multiscale Modeling
AT
Δ
CG
Probabilistic Molecular Modeling
$p(\mathbf{y}|\mathbf{x},\mathcal{D}) = \int p(\mathbf{y}|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$
$p(\boldsymbol{\theta}|\mathcal{D})$
$\bar{\theta}$
$\boldsymbol{\theta}$

Figure 2.1.: Molecular modeling framework. Overview of the relationships between the main molecular modeling components discussed in this thesis.

MD simulations are at the center of this framework. An MD simulation numerically integrates Newton's equation of motion in time (sec. 2.1). The potential energy function $U_{\boldsymbol{\theta}}$ defines the dynamics of the system via the forces $\mathbf{f}^{(i)}$. In the context of this thesis, graph NN potentials are typically used to model $U_{\boldsymbol{\theta}}$ (sec. 2.2). Training the NN potential, i.e.

finding an approximation to the optimal parameter set $\bar{\boldsymbol{\theta}}$, can be achieved by minimizing a loss function $\mathcal{L}(\boldsymbol{\theta})$ that results from the bottom-up or top-down training scheme (sec. 2.2.1). Rather than betting on a single parameter set, Bayesian modeling approximates the posterior predictive distribution, which promises more accurate predictions and enables UQ of MD observables (sec. 2.4). Instead of operating on an AT representation of the system, NN potentials can also be trained for CG systems (sec. 2.3). If both an AT and a CG model are available for the system of interest, concurrent multiscale simulations allow to combine the different resolutions (sec. 2.5).

2.1. Molecular Modeling

In this section, I review the basics of MD simulations with an emphasis on the potential energy function to introduce the notation for the rest of the thesis. For a more detailed outline of statistical mechanics and MD, refer to refs. [25, 91].

2.1.1. Molecular Dynamics Simulation

Fundamentally, an MD simulation integrates Newton's second law to compute the time evolution of the system under investigation:

$$m^{(i)} \frac{\mathrm{d}^2 \mathbf{r}^{(i)}}{\mathrm{d}t^2} = \mathbf{f}^{(i)} \quad \text{with} \quad \mathbf{f}^{(i)} = \frac{\mathrm{d}U_{\boldsymbol{\theta}}(\mathbf{r})}{\mathrm{d}\mathbf{r}^{(i)}} \ , \tag{2.1}$$

where $\mathbf{r} \in \mathbb{R}^{n \times 3}$ is a matrix containing the position vectors of all atoms and n is the total number of atoms in the system. For each atom i, $m^{(i)}$ is its constant mass, $\mathbf{r}^{(i)}$ is its position and $\mathbf{f}^{(i)}$ is the force acting on its center of mass (COM). The forces are typically computed as the gradient of the potential energy function $U_{\boldsymbol{\theta}}(\mathbf{r})$, which ensures that the resulting force field is conservative and the sum of the forces is $\mathbf{0}$ [91]. $U_{\boldsymbol{\theta}}(\mathbf{r})$ is parametrized by a vector of model parameters $\boldsymbol{\theta}$.

Starting from an initial state, the time evolution of the system can be computed by numerical integration. To this end, eq. (2.1) can be re-written in terms of first order ordinary differential equations (ODEs) by introducing $\mathbf{p}^{(i)}$, the momentum of particle i :

$$\frac{\mathrm{d}\mathbf{r}^{(i)}}{\mathrm{d}t} = \mathbf{p}^{(i)}/m^{(i)} \quad ; \quad \frac{\mathrm{d}\mathbf{p}^{(i)}}{\mathrm{d}t} = \mathbf{f}^{(i)} \ . \tag{2.2}$$

Then, the Velocity Verlet [92] time integration scheme approximates the state of the system $\mathbf{S} = \{\mathbf{r}, \mathbf{p}\}$ after a time step Δt, where $\mathbf{p} \in \mathbb{R}^{n \times 3}$ is a matrix containing the momentum vectors of all atoms in the system:

$$\begin{aligned} \mathbf{p}^{(i)}(t + 0.5\Delta t) &= \mathbf{p}^{(i)}(t) + 0.5\Delta t \mathbf{f}^{(i)}(t) \\ \mathbf{r}^{(i)}(t + \Delta t) &= \mathbf{r}^{(i)}(t) + \Delta t \mathbf{p}^{(i)}(t + 0.5\Delta t)/m^{(i)} \\ \mathbf{p}^{(i)}(t + \Delta t) &= \mathbf{p}^{(i)}(t + 0.5\Delta t) + 0.5\Delta t \mathbf{f}^{(i)}(t + \Delta t) \ . \end{aligned} \tag{2.3}$$

The Velocity Verlet algorithm is one of the most popular time integration schemes in MD because it is exactly time reversible, symplectic (sec. 2.1.2) and features a second order global error, while only requiring a single computationally expensive force computation per time step.

In principle, an MD simulation consists of the following steps: Starting from an initial state, the time integration scheme is applied iteratively to generate the trajectory of the system $\{\mathbf{S}^{(i)}\}_{i=1}^{N}$. From this trajectory, any observable of interest can be computed, including structural (e.g. distribution of bond lengths and dihedral angles [93, 94], radial distribution function [95] and angular distribution function), thermodynamic (e.g. temperature, pressure [96]), mechanic (e.g. stress [97] and stiffness tensors [98]) and dynamic quantities (e.g. diffusion coefficients). The obtained observables can then be compared to experimental observations [99–101] to evaluate the quality of the MD simulation [26, 38, 79].

2.1.2. Statistical Mechanics Ensembles

Statistical mechanics links macroscopic equilibrium observables $\langle O \rangle$ to microscopic states $\mathbf{S}$ via the ensemble average [91]

$$\langle O \rangle = \int_{\mathbf{S}} O(\mathbf{S}, U_{\boldsymbol{\theta}}) p(\mathbf{S}) \mathrm{d}\mathbf{S} \simeq \frac{1}{N} \sum_{i=1}^{N} O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) \; ; \mathbf{S}_i \sim p(\mathbf{S}) \; , \tag{2.4}$$

where $O(\mathbf{S}, U_{\boldsymbol{\theta}})$ is the instantaneous value of the observable function. Given that the integral in eq. (2.4) is typically very high-dimensional, it is approximated by a Monte Carlo estimate. Hence, computing observables requires sampling of microscopic states $\mathbf{S}$ according to the distribution $p(\mathbf{S})$, which depends on the chosen ensemble [91].

The remainder of this section introduces the microcanonical and canonical equilibrium ensembles. The isothermal-isobaric equilibrium ensemble is also of great importance, as it most closely corresponds to typical experimental setups. While the choice of the ensemble impacts MD results, this difference vanishes in the thermodynamic limit ($n \to \infty$) [91].

Microcanonical Ensemble

The microcanonical ensemble subsumes all states $\mathbf{S}$ of an isolated system, which cannot exchange mass or energy with its surroundings. Consequently, the total energy of the system E is constant [91]. The microcanonical ensemble is also referred to as NVE ensemble because the isolated system features constant number of particles n, constant volume V, and constant total energy E.

The total energy (or Hamiltonian) of a state $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$ is the sum of the potential energy $U_{\boldsymbol{\theta}}(\mathbf{r})$ and the kinetic energy $\mathcal{K}(\mathbf{p})$:

$$\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S}) = U_{\boldsymbol{\theta}}(\mathbf{r}) + \mathcal{K}(\mathbf{p}) \quad \text{with} \quad \mathcal{K}(\mathbf{p}) = \sum_{i=1}^{n} \frac{||\mathbf{p}^{(i)}||^2}{2m^{(i)}} \; . \tag{2.5}$$

In the NVE ensemble, all states have the same Hamiltonian $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S}) = E$. All states accessible to the isolated system are assumed to be equally probable [91]. Thus, states in the NVE ensemble are uniformly distributed with probability

$$p(\mathbf{S}) = \frac{1}{\Omega(N,V,E)} \ , \tag{2.6}$$

where $\Omega(N,V,E)$ is the microcanonical partition function, which measures the amount of phase-space accessible to the system [91].

Given that exact time integration of the ODEs in eq. (2.12) conserves $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$, an MD simulation samples states from the NVE ensemble with $E = \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S}_{\text{init}})$. Under the assumption of ergodicity, i.e. the property of a system to visit all of its accessible states in the limit of an infinite amount of time, the ensemble average in eq. (2.4) can be substituted by a time average over the MD trajectory:

$$\langle O \rangle \simeq \frac{1}{N} \sum_{i=1}^{N} O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) \ . \tag{2.7}$$

Due to small values of Δt necessary to sustain numerical stability of the MD simulation, subsequent states are highly correlated. To save memory and improve the statistical efficiency in eq. (2.7), states are typically only saved in the order of magnitude of every 1000 time steps.

Canonical Ensemble

The canonical ensemble, also referred to as NVT ensemble, corresponds to a system with constant density and temperature T. Hence, the NVT ensemble includes all microscopic states $\mathbf{S}$ of a closed system within a heat-bath, which cannot exchange mass with its surroundings, but exchanges thermal energy with the heat bath [91]. Due to the heat transfer, the Hamiltonian $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$ is not constant but depends on the state $\mathbf{S}$.

To be able to use an MD simulation to sample from the NVT ensemble, a thermostat must be added to the ODEs in eq. (2.12) such that the instantaneous temperature

$$\bar{T}(\mathbf{S}) = \frac{2\mathcal{K}(\mathbf{p})}{N^{\text{DOF}} k_{\text{B}}} \tag{2.8}$$

is kept constant rather than $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$, where k_{B} is the Boltzmann constant and N^{DOF} is the number of degrees of freedom. The Langevin and Nose-Hoover chain thermostats [102] are two popular options. The former controls the temperature by adding a friction term and random forces, while the latter is a deterministic thermostat.

States in the NVT ensemble follow the Boltzmann distribution

$$p(\mathbf{S}) = \frac{e^{-\beta \mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})}}{\Omega(N,V,T)} \ , \tag{2.9}$$

where $\Omega(N,V,T)$ is the canonical partition function and $\beta = 1/(k_{\text{B}}T)$. An MD simulation with a well-chosen thermostat can be used to sample the Boltzmann distribution under the assumptions of ergodicity and thermodynamic equilibrium. The latter can be satisfied by retaining states after an initial equilibration phase only.

As an alternative to MD, the Boltzmann distribution can be directly sampled via Metropolis-Hastings (MH) [103] Markov chain Monte Carlo (MCMC). However, while MD also predicts dynamical properties of the system, MCMC only yields equilibrium properties [91]. Both approaches can be combined by running short MD simulations to generate proposal states with high acceptance probability (Hamiltonian or hybrid Monte Carlo [104]). Hamiltonian Monte Carlo has received renewed attention in the last decade in the context of efficient sampling of high-dimensional posterior distributions for Bayesian modeling [105, 106] (chapter 2.4.1).

2.1.3. Classical Potentials

So far, the potential energy function $U_{\boldsymbol{\theta}}(\mathbf{r})$ was assumed to be given. However, the selection of an appropriate $U_{\boldsymbol{\theta}}(\mathbf{r})$ is the main modeling choice in MD, since $U_{\boldsymbol{\theta}}(\mathbf{r})$ defines the dynamics of the system. Additionally, computation of the forces is the computationally most demanding part of the time integration scheme (eq. (2.3)). Therefore, a good potential $U_{\boldsymbol{\theta}}(\mathbf{r})$ must be accurate enough to model all relevant atomic interactions, while being computationally inexpensive to generate sufficiently long trajectories.

To improve physical interpretability, $U_{\boldsymbol{\theta}}(\mathbf{r})$ can be decomposed into intra-molecular and non-bonded interactions as well as into a series of increasing body-order interactions:

$$\begin{aligned} U_{\boldsymbol{\theta}}(\mathbf{r}) &= U_{\boldsymbol{\theta}}^{\mathrm{intra}}(\mathbf{r}) + U_{\boldsymbol{\theta}}^{\mathrm{non-bonded}}(\mathbf{r}) \\ &= \sum_{i} u_{\boldsymbol{\theta}}(\mathbf{r}_i) + \sum_{i,j>i} u_{\boldsymbol{\theta}}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{i,j>i,k>j} u_{\boldsymbol{\theta}}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + ... \end{aligned} \tag{2.10}$$

The rationale of the body-order decomposition is that the magnitude of interactions tends to decrease for increasing body-order. Hence, classical potentials usually truncate the series beyond 2-body non-bonded interactions and 4-body intra-molecular interactions [35–37].

The remainder of this subsection introduces the main building blocks used by many classical force fields to build $U_{\boldsymbol{\theta}}(\mathbf{r})$. Classical intra-molecular potentials typically include harmonic bonds, angles and dihedrals, representing 2, 3 and 4-body interactions:

$$U_{\boldsymbol{\theta}}^{\mathrm{intra}}(\mathbf{r}) = \sum_{i=1}^{N_{\mathrm{bonds}}} K_i^b[b_i - \hat{b}_i]^2 + \sum_{j=1}^{N_{\mathrm{angles}}} K_i^{\alpha}[\alpha_i - \hat{\alpha}_i]^2 + \sum_{k=1}^{N_{\mathrm{dihedrals}}} K_i^{\omega}[1 + \cos(n_{\omega,i}\omega_i - \hat{\omega}_i)] \ , \tag{2.11}$$

where b_i are the bond lengths, α_i are enclosed angles of triplets of bonded atoms, and ω_i are dihedral angles. The energy scale of the interactions K_i, the equilibrium bond lengths $\hat{b}_i$, the equilibrium angles $\hat{\alpha}_i$, the dihedral multiplicities $n_{\omega,i}$, and the dihedral phase shifts $\hat{\omega}_i$ depend on the atom types involved and represent parameters in $\boldsymbol{\theta}$.

Non-bonded interactions are computationally more expensive given that the number of 2-body pairs in the system scales as $\mathcal{O}(n^2)$, 3-body triplets as $\mathcal{O}(n^3)$, etc. Hence, the non-bonded part of classical potentials often consists of 2-body Lennard Jones (LJ) and Coulomb potentials only [35–37], modeling steric repulsion, Van-der-Waals attraction and electrostatic interactions:

$$U_{\boldsymbol{\theta}}^{\text{non-bonded}}(\mathbf{r}) = \sum_{i,j>i} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{d_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{d_{ij}} \right)^{6} \right] + \sum_{i,j>i} \frac{q_i q_j}{4\pi\epsilon_0 d_{ij}} \ , \tag{2.12}$$

where d_{ij} is the Euclidean distance between a pair of atoms and ϵ_0 is the vacuum permittivity. The partial charge of an atom type q_i and LJ parameters ϵ_{ij} and σ_{ij} are degrees of freedom of $\boldsymbol{\theta}$. The LJ parameters depend on the atom types of atoms i and j via geometric combination rules [107].

Given that the LJ potential is a short-range interaction, a cut-off radius r_{cut} can be introduced beyond which the potential energy of the pair interaction is set to 0. Electrostatic interactions can be cut-off analogously using the generalized reaction field method [108]. Restricting the potential to local interactions significantly reduces the computational effort as only neighbors within the cut-off need to be considered. Introducing cell and neighbor lists then reduces the computational complexity of force computations to $\mathcal{O}(n \log n)$ [25, 91]. Sophisticated classical force fields introduce further modifications to the components outlined above, such as 1-4 LJ interactions as well as mixed atom type pairs [36], but these are beyond the scope of this methodological outline.

2.1.4. Reweighting

Knowledge of the probability distributions of different ensembles (sec. 2.1.2) allows the reuse of a previously generated MD trajectory to estimate the resulting observables for slightly perturbed simulation parameters. To this end, thermodynamic perturbation theory [109] replaces the time average in eq. (2.7) with a weighted average over the trajectory

$$\langle O \rangle \simeq \sum_{i=1}^{N} w_i O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) \quad \text{with} \quad w_i = \frac{p(\mathbf{S}_i)/\hat{p}(\mathbf{S}_i)}{\sum_{j=1}^{N} p(\mathbf{S}_j)/\hat{p}(\mathbf{S}_j)} \ , \tag{2.13}$$

where the weight w_i depends on the probability of $\mathbf{S}_i$ in the perturbed simulation $p(\mathbf{S}_i)$ relative to the probability in the existing reference system $\hat{p}(\mathbf{S}_i)$. Without perturbation ($p(\mathbf{S}) = \hat{p}(\mathbf{S})$), eq. (2.13) is identical to eq. (2.7). In the NVT ensemble, the Boltzmann distribution (eq. (2.9)) defines $p(\mathbf{S})$.

An example application of reweighting is changing the temperature of the system. Assuming an NVT ensemble, the weights w_i evaluate to

$$w_i = \frac{e^{-(\beta - \hat{\beta}) U_{\boldsymbol{\theta}}(\mathbf{S}_i)}}{\sum_{j=1}^{N} e^{-(\beta - \hat{\beta}) U_{\boldsymbol{\theta}}(\mathbf{S}_j)}} \ , \tag{2.14}$$

where $\hat{\beta} = 1/(k_{\mathrm{B}}\hat{T})$ is the inverse temperature of the reference trajectory and β is the target inverse temperature.

Another important application of reweighting is the optimization of $U_{\boldsymbol{\theta}}(\mathbf{S})$. For this application, it is important to obtain observables that correspond to a potential $U_{\boldsymbol{\theta}}$ given a trajectory generated from the reference potential $U_{\hat{\boldsymbol{\theta}}}$. Assuming the NVT ensemble and inserting the Boltzmann distribution into eq. (2.13) yields the weights

$$w_i = \frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{\sum_{i=j}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}} \ , \tag{2.15}$$

which can be used to reduce the number of necessary trajectory generations during optimization [110–112] (sec. 2.3.3).

Due to the generality of the reweighting principle, it is useful in a wide range of applications including estimation of free energy differences [113, 114], switching between different ensembles [115], simulating phase equilibria [116–118], and unbiasing trajectories generated from enhanced sampling schemes [119]. The main caveat of reweighting is the reduction of the effective sample size [120]

$$N_{\text{eff}} \approx e^{-\sum_{i=1}^{N} w_i \ln(w_i)} \tag{2.16}$$

due to significantly reduced contribution of states that are unlikely given the target distribution. This increases the statistical estimation error in eq. (2.13) and restricts the applicability of reweighting to small perturbations such that the expected statistical error remains below a certain threshold.

2.1.5. Numerical Integration Errors

Finally, sampling the NVE ensemble via an MD simulation relies on the fact that exact time integration of eq. (2.12) with a conservative force field conserves $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$ along the trajectory. However, due to truncation errors of the numerical integration scheme, $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$ is not conserved for finite Δt. In this case, a trajectory obtained from the numerical integration scheme is the solution of a modified differential equation [121], which includes time step-dependent terms that vanish for $\Delta t \to 0$. In the case of a symplectic integration scheme (e.g. eq. (2.3)), the integrator exactly conserves a shadow Hamiltonian $\tilde{\mathcal{H}}_{\boldsymbol{\theta}}(\mathbf{S}, \Delta t)$ [122, 123]. For sufficiently small Δt, $\tilde{\mathcal{H}}_{\boldsymbol{\theta}}(\mathbf{S}, \Delta t)$ remains close to $\mathcal{H}_{\boldsymbol{\theta}}(\mathbf{S})$ along the trajectory, bounding the numerical error [91].

In the case of the NVT ensemble (with Nose-Hoover chain thermostat [102]), a symplectic integrators exactly conserves a shadow temperature $\tilde{\mathcal{T}}(\mathbf{S}, \Delta t)$ [124]. Analogous to the NVE ensemble, the deviation of $\tilde{\mathcal{T}}(\mathbf{S}, \Delta t)$ from $T(\mathbf{S})$ is bounded for sufficiently small Δt. For both ensembles, the numerical error scales as $(\Delta t)^2$ for second order accurate integration schemes such as the Velocity Verlet integrator [124].

2.2. Neural Network Potentials

NN potentials differ from classical potentials by featuring a flexible functional form and a high-dimensional set of parameters $\boldsymbol{\theta}$. Due to this difference, there are several considerations to account for when designing and training NN potentials, which will be outlined in this section.

2.2.1. Training

Once the functional form of the potential $U_{\boldsymbol{\theta}}(\mathbf{S})$ has been fixed (for the functional form of classical potentials, see section 2.1.3; for NN potentials, section 2.2.5), the goal of training is to obtain an optimal parametrization

$$\bar{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \ , \tag{2.17}$$

such that the obtained model minimizes a loss function $\mathcal{L}(\boldsymbol{\theta})$. The training approaches discussed below are suitable for both classical and NN potentials as the minimization problem in eq. (2.17) will be solved by gradient-based optimization. While classical potentials can be optimized via trial-and-error [125] or gradient-free optimizers [126–128], gradients are a necessity to train larger NNs due to the curse of dimensionality associated with the large NN parameter set.

Bottom-Up Training

Most commonly, AT NN potentials are optimized to match the potential energy U_i and forces $\mathbf{f}_i \in \mathbb{R}^{n\times 3}$ of a training data set $\{\mathbf{r}_i, U_i, \mathbf{f}_i\}_{i=1}^{N_\text{box}}$ containing N_box molecular states [28]. The labels U_i and $\mathbf{f}_i$ are computed from a high-fidelity model, typically a CQM scheme [53, 129, 130], and are considered the ground truth. Matching the predictions of the high-fidelity model can be achieved by minimizing the mean squared error (MSE) loss

$$\mathcal{L}^\text{AT}(\boldsymbol{\theta}) = \frac{1}{N_\text{box}} \sum_{i=1}^{N_\text{box}} [U_i - U_{\boldsymbol{\theta}}(\mathbf{r}_i)]^2 + \frac{\gamma}{N_\text{box}} \sum_{i=1}^{N_\text{box}} \left|\left| \mathbf{f}_i + \frac{\mathrm{d}U_{\boldsymbol{\theta}}(\mathbf{r}_i)}{\mathrm{d}\mathbf{r}_i} \right|\right|^2 \ , \tag{2.18}$$

where $||...||$ is the Frobenius norm and γ is a hyperparameter controlling the relative contribution of errors in the energies and forces. If force targets are available, it is advantageous to include them in eq. (2.18) because of the much larger information content per data point and their importance to the dynamics of subsequent MD simulations [67, 131].

Given the loss function $\mathcal{L}^\text{AT}(\boldsymbol{\theta})$, the potential $U_{\boldsymbol{\theta}}$ can be trained using stochastic gradient descent optimization: The gradient $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ is approximated by a stochastic estimator based on a subset (called mini-batch) of the data set and computed via automatic differentiation (AD) [132]. This estimate of the gradient is then passed to a stochastic optimizer such as RMSProp [133] or Adam [134] to update $\boldsymbol{\theta}$. The computation of the gradient estimates and the updating of the parameters is then repeated until convergence. Finally, it is important to test the trained model on a held-out test data set as an estimate for performance on unseen data.

Top-Down Training

For top-down learning, the following MSE loss can be minimized to match a set of K experimental observables $\tilde{O}_k$:

$$\mathcal{L}^\text{TD}(\boldsymbol{\theta}) = \frac{1}{K} \sum_{k=1}^{K} \left[\langle O_k(U_{\boldsymbol{\theta}}) \rangle - \tilde{O}_k \right]^2 \ . \tag{2.19}$$

However, unlike the potential energy and forces, $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ is connected to $U_{\boldsymbol{\theta}}$ only indirectly via an MD simulation, as reflected in the following equation:

$$\mathcal{L}^{\mathrm{TD}}(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K}\left[O_k(\mathrm{Simulation}(U_{\boldsymbol{\theta}})) - \tilde{O}_k\right]^2 \ . \tag{2.20}$$

Straightforward computation of $\nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ via AD requires backpropagation through the MD simulation (eq. (2.20), fig. 2.2), which is enabled by recently developed differentiable MD simulation codes such as JAX, M.D. [135] and TorchMD [136].

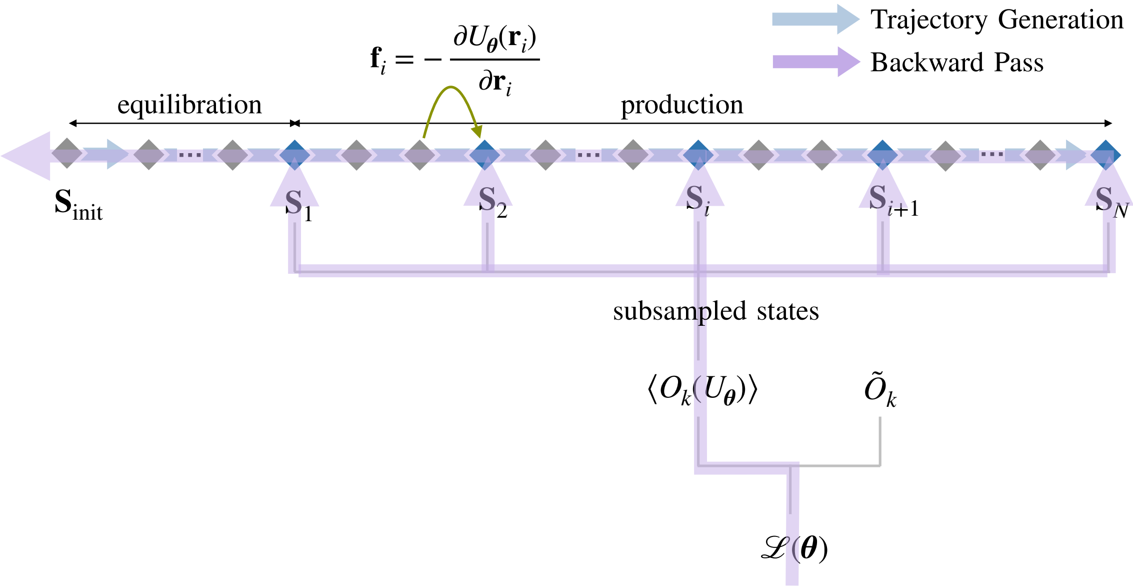

Trajectory Generation
Backward Pass
$\mathbf{f}_i = -\dfrac{\partial U_{\boldsymbol{\theta}}(\mathbf{r}_i)}{\partial \mathbf{r}_i}$
equilibration
production
$\mathbf{S}_{\mathrm{init}}$
$\mathbf{S}_1$
$\mathbf{S}_2$
$\mathbf{S}_i$
$\mathbf{S}_{i+1}$
$\mathbf{S}_N$
subsampled states
$\langle O_k(U_{\boldsymbol{\theta}})\rangle$
$\tilde{O}_k$
$\mathscr{L}(\boldsymbol{\theta})$

Figure 2.2.: Direct backpropagation schematic. Visualization of the forward and backward pass for computing the gradient of the top-down loss $\mathcal{L}^{\mathrm{TD}}(\boldsymbol{\theta})$ via direct backpropagation through the simulation.

Learning $U_{\boldsymbol{\theta}}$ by backpropagation through a differentiable MD simulation has been achieved in recent works [135–139]. However, this approach is restricted to small trajectory lengths [139] due to the curse of chaos [140]: The gradient of eq. (2.20) indicates the direction in which $U_{\boldsymbol{\theta}}$ must be updated to direct the MD trajectory such that the resulting $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ improves. This is an ill-posed problem in chaotic systems such as MD [141]. The exponential increase of the sensitivity of the final simulation state on the initial conditions in time, in particular beyond the Lyaponov time scale [142], results in exploding and uninformative gradients [137].

Instead of capturing the dependence of $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ on $U_{\boldsymbol{\theta}}$ via the dynamics of the system (eq. (2.3), fig. 2.2), the gradient can also be computed via thermodynamic fluctuation formulas from ensemble averages of the gradient of observables [143–145], circumventing the curse of chaos. However, this approach requires significant implementation effort to combine gradients from different observables, especially for observables that are not mere averages of instantaneous quantities.

2.2.2. Prior Potential

When training NN potentials in a bottom-up manner via eq. (2.18), the data set is often generated by sampling states from the ground truth model, e.g. via a MD simulation. In this case, the obtained data set contains few high potential energy states and no unphysical states. As a result of the flexible functional form and data-driven nature of NNs, the NN potential $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r})$ is only constrained in phase-space regions resolved by the training data. However, MD simulations using the trained $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r})$ might be able to unphysically cross an energy barrier and sample from unphysical phase-space regions. This results in highly inaccurate simulation outcomes [68, 69] or even numerical instability [50, 61].

In order to counteract this phenomenon, $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r})$ can be combined with a fixed, "prior" potential $U^{\mathrm{prior}}(\mathbf{r})$ with a physics-inspired functional form [68, 69, 146]:

$$U_{\boldsymbol{\theta}}(\mathbf{r}) = U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r}) + U^{\mathrm{prior}}(\mathbf{r}) \ . \tag{2.21}$$

Suitable choices for U^{prior} are usually available from the literature, e.g. the Embedded Atom Model (EAM) [147] for metals or AMBER [35] for bio-molecules. The ansatz in eq. (2.21) can be interpreted as Δ-learning [148], where the goal of U^{prior} is to yield qualitatively correct predictions outside the training data distribution and $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r})$ can correct U^{prior} in phase-space regions where training data are available. Hence, U^{prior} can be considered a physics-based initialization of the combined potential $U_{\boldsymbol{\theta}}$. In particular, $U^{\mathrm{prior}}(\mathbf{r})$ is distinct from the prior in Bayesian modeling (sec. 2.4.1).

2.2.3. Physical Requirements

There are several requirements for $U_{\boldsymbol{\theta}}$ entailed by physics: First, energy conservation requires $U_{\boldsymbol{\theta}}$ to be continuously differentiable at r_{cut} such that the resulting force is $\mathbf{0}$. For gradient-based learning of $U_{\boldsymbol{\theta}}$ with force targets (eq. (2.18)), $U_{\boldsymbol{\theta}}$ should be $\mathcal{C}^2$ everywhere such that gradients $\nabla_{\boldsymbol{\theta}}$ are continuous. Second, $U_{\boldsymbol{\theta}}$ needs to be invariant to translation and rotation of the coordinate system as well as to permutation of the numbering of particles given that the potential energy of the system remains unaffected by these changes (fig. 2.3). While classical potentials fulfil these requirements by design of the considered functional

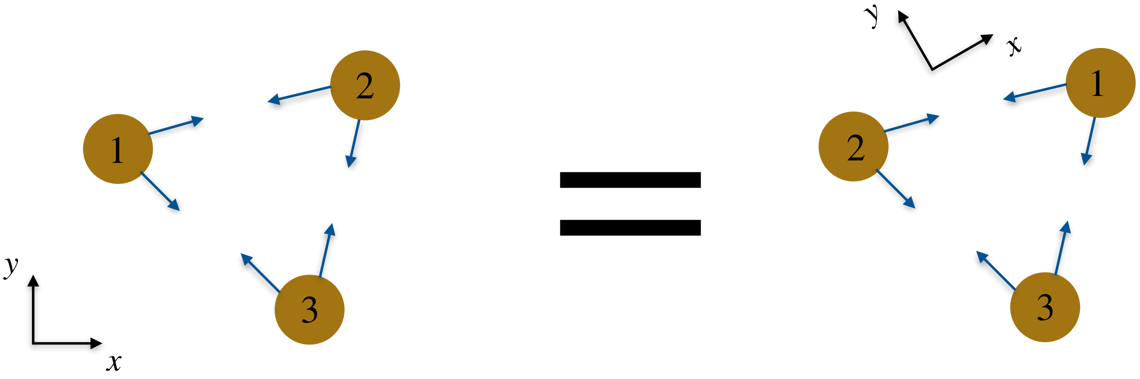


Figure 2.3.: Physical invariances. Invariance of the potential energy of a molecular system to translation, rotation and permutation.

forms, more care needs to be taken to build a NN potential architecture with analogous properties.

In principle, these invariances can also be learned by data augmentation [149]. However, this decreases data efficiency by 3 orders of magnitude [150] and the predictions of $U_{\boldsymbol{\theta}}$ will only approximately lie on the manifold defined by the physical invariances. Therefore, explicitly encoding these invariances has been found to be key to data-efficient learning [51].

2.2.4. Feature Extraction

The most straightforward approach to encoding the physical invariances into NN potentials is given by the Behler-Parinello architecture [51]. In a first step, invariant features of the local chemical environment of each atom are extracting via fixed descriptor functions such as the atomic cluster expansion (ACE) [151, 152] or atom-centred symmetry functions [153–155]. In a second step, for each atom, the extracted features are the input of a multilayer perceptron (MLP) predicting the potential energy contribution of the atom $U_{\boldsymbol{\theta}}^{(i),\mathrm{NN}}(\mathbf{r})$. Finally, the potential energy of the system

$$U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{r}) = \sum_{i=1}^{n} U_{\boldsymbol{\theta}}^{(i),\mathrm{NN}}(\mathbf{r}) \tag{2.22}$$

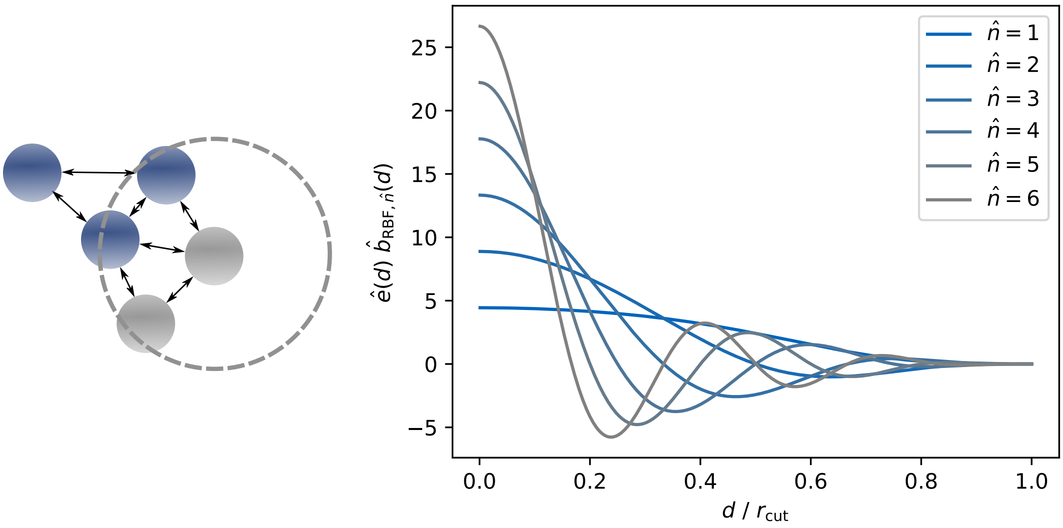
is the sum of all atom-wise potential energy contributions [51].

Analogous to computer vision [156], hand-crafted feature extractors are the accuracy bottleneck of the Behler-Parinello architecture. Accordingly, replacing the descriptors by continuous convolution layers [55] that learn to extract expressive features in an end-to-end fashion improves performance considerably [64, 157]. In this case, the physcial invarances are obeyed by representing the molecular configuration as a graph with invariant features (fig. 2.4 (a)), typically nodes (atoms) and edges (pairwise distances of atoms within r_{cut} of each other). The resulting graph neural network (GNN) architecture [158] operates on the molecular graph by iterative pairwise message-passing [159], extracting higher-body features with each consecutive message-passing step.

2.2.5. Message-Passing Neural Network Architecture

This sections outlines main building blocks of invariant GNNs that are used across multiple NN potential architectures. The concrete implementation of these components and their combination with other building blocks strongly depends on the respective architecture, which has led to a large number of proposed GNN potentials [55, 59, 67–69, 159–164].

The input of invariant GNN potentials is the molecular graph, which typically consists of the atomic numbers $Z^{(i)}$ and pairwise distances $d^{(ij)}$ for particles within r_{cut}. GNN potentials usually consist of four main components: basis functions, embedding layer, message-passing layers and an output layer. In order to reduce the correlation between different convolutional filters and to speed up training [55], scalar properties are usually embedded into vector-values quantities. To this end, the embedding layer builds a vector-representation for $Z^{(i)}$ and different basis functions represent $d^{(ij)}$. Given this vector-valued graph representation, repeated application of the message-passing layer extracts features



(a) (b)

Figure 2.4.: Graph neural network. (a) Sketch of a graph neural network with cut-off around a central atom (adapted from [2]). (b) $\hat{n}^{\text{th}}$ radial Bessel basis function $\hat{b}_{\mathrm{RBF},\hat{n}}(d)$ times envelope function $e(d)$ as a function of the pairwise distance d.

of the chemical environment of each atom. The output layer then further processes these features and predicts per-atom potential energies, which are summed to obtain the potential energy of the system.

Basis Functions

Schütt et al. [55] proposed Gaussian radial basis functions to represent each $d^{(ij)}$. Inspired by the functional form of solutions to Schrödinger's equation, Klicpera et al. [67] later proposed a Fourier-Bessel basis to expand $d^{(ij)}$ (and triplet angles $\alpha^{(ijk)}$). The $\hat{n} \in [1, .., N_{\mathrm{RBF}}]$ radial Bessel functions (RBF) $\hat{b}_{\mathrm{RBF},\hat{n}}(d)$ take the following form:

$$\hat{b}_{\mathrm{RBF},\hat{n}}(d) = \sqrt{\frac{2}{r_{\mathrm{cut}}}}\frac{\sin(\hat{n}\pi d/r_{\mathrm{cut}})}{d}\ . \tag{2.23}$$

To enforce that the contribution to the potential energy by an atom pair converges to 0 at r_{cut} in a continuously differentiable manner, each RBF is multiplied by an envelope function

$$\hat{e}(d) = 1 - \frac{(\hat{p}+1)(\hat{p}+2)}{2}\left(\frac{d}{r_{\mathrm{cut}}}\right)^{\hat{p}} + \hat{p}(\hat{p}+2)\left(\frac{d}{r_{\mathrm{cut}}}\right)^{\hat{p}+1} - \frac{\hat{p}(\hat{p}+1)}{2}\left(\frac{d}{r_{\mathrm{cut}}}\right)^{\hat{p}+2}\ , \tag{2.24}$$

where $\hat{p} = 6$ is the default choice [67]. The resulting representation of $d^{(ij)}$ is a N_{RBF} dimensional vector $\mathbf{e}^{(ij)} = \hat{e}(d^{(ij)})\hat{\mathbf{b}}_{\mathrm{RBF}}(d^{(ij)})$ (fig. 2.4 (b)). The radial Bessel basis has been shown to be more efficient than the Gaussian basis because it requires a smaller number of basis functions while increasing accuracy [67].

Embedding Layer

A common embedding method for $Z^{(i)}$ employs a randomly initialized and learnable look-up-table $\mathbf{T}_{\boldsymbol{\theta}}$ [55, 67, 161], which stores for each atom type a N_{embed}-dimensional vector

$$\mathbf{h}_0^{(i)} = \mathbf{T}_{\boldsymbol{\theta}}(Z^{(i)}) \ . \tag{2.25}$$

Alternatively, the node embedding can be predicted by a linear layer with the one-hot encoded $Z^{(i)}$ as input [165].

Message Passing Layer

The standard message-passing formalism can be described via two update functions [166]: First, each atom accumulates messages from all connected atoms in the graph $\mathfrak{N}$ (fig. 2.4 (a)) to generate the total message

$$\mathbf{m}_k^{(i)} = \sum_{j \in \mathfrak{N}(i)} \mathbf{M}_{k,\boldsymbol{\theta}}(\mathbf{h}_k^{(i)}, \mathbf{h}_k^{(j)}, \mathbf{e}^{(ij)}) \ , \tag{2.26}$$

which is used to update the hidden node state

$$\mathbf{h}_{k+1}^{(i)} = \mathbf{H}_{k,\boldsymbol{\theta}}(\mathbf{h}_k^{(i)}, \mathbf{m}_k^{(i)}) \ . \tag{2.27}$$

$\mathbf{M}_{k,\boldsymbol{\theta}}$ and $\mathbf{H}_{k,\boldsymbol{\theta}}$ are two arbitrary learnable functions for each message-passing step k, which typically consist of MLPs and ResNet-type [167] skip connections [55, 67, 161, 165].

Output Layer

Afterwards, the hidden node states $\mathbf{h}_k^{(i)}$ are used to predict the per-atom potential energy contributions

$$U_{\boldsymbol{\theta}}^{(i),\text{NN}} = \sum_{k=1}^{K} \mathbf{F}_{k,\boldsymbol{\theta}}(\mathbf{h}_k^{(i)}) \ , \tag{2.28}$$

where $\mathbf{F}_{k,\boldsymbol{\theta}}$ are learnable output functions. Eq. (2.28) corresponds to the general case in which all intermediate hidden node states are used [67, 161]. Finally, the predicted $U_{\boldsymbol{\theta}}^{(i),\text{NN}}$ are summed analogous to eq. (2.22) to obtain the potential energy prediction of the system $U_{\boldsymbol{\theta}}^{\text{NN}}(\mathbf{r})$.

2.3. Coarse-Graining

So far, the methodological review has focused on AT systems. However, for many biophysical systems of interest, relevant time and lengths scales cannot be reached by AT MD simulations; thus, CG models are required [26]. Hence, this section covers the basics of CG modeling as well as relevant bottom-up training schemes for CG NN potentials.

At the core of CG modeling is the mapping function $\mathbf{M}$:

$$\mathbf{R} = \mathbf{M}(\mathbf{r}) \ . \tag{2.29}$$

$\mathbf{M}$ maps the coordinates of an AT molecular configuration $\mathbf{r}$ onto CG coordinates $\mathbf{R} \in \mathbb{R}^{N_{\mathrm{CG}} \times 3}$ ($N_{\mathrm{CG}} < n$, fig. 2.5). The remainder of this section assumes that $\mathbf{M}$ is a linear function and that both AT and CG systems are NVT equilibrium ensembles. For generalizations to a non-linear $\mathbf{M}(\mathbf{r})$ and non-equilibrium systems, refer to [168, 169].

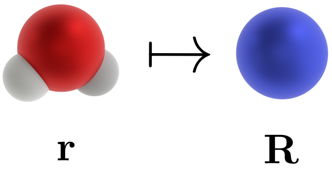


Figure 2.5.: Coarse-graining mapping function. Visualization of a coarse-graining mapping function from an atomistic ($\mathbf{r}$) to a coarse-grained configuration of water ($\mathbf{R}$). Adapted from [2].

2.3.1. Training Coarse-Grained Potentials

CG models can be trained top-down [170] analogous to AT models (eq. (2.19)). However, the bottom-up training scheme (eq. (2.18)) needs to be adjusted to train CG models given that the target CG potential energy is not available. Ideally, the obtained CG model should be consistent with the data-generating AT model. In this case, the configurational distribution of the CG model $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ equals the configurational distribution of the AT model when mapped to CG coordinates $p^{\mathrm{AT}}(\mathbf{R})$ [32]. If the CG potential $U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ equals the many-body potential of mean force (PMF)

$$U^{\mathrm{PMF}}(\mathbf{R}) = -\frac{1}{\beta} \ln p^{\mathrm{AT}}(\mathbf{R}) + C \ , \tag{2.30}$$

where C as an arbitrary additive constant, then $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R}) = p^{\mathrm{AT}}(\mathbf{R})$ [32, 171]. The most popular training schemes that approximate the PMF are force matching (FM) [32] and relative entropy (RE) minimization [171], which are introduced in the next two subsections.

2.3.2. Force Matching

FM [31–33, 172] is closely related to the AT energy-matching scheme in eq. (2.18). Given that no potential energy targets are available, only target forces $\mathbf{F}^{\mathrm{AT}} \in \mathbb{R}^{N_{\mathrm{CG}} \times 3}$ can be matched:

$$\mathcal{L}^{FM}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{box}}} \sum_{i=1}^{N_{\mathrm{box}}} \left|\left| \mathbf{F}_i^{\mathrm{AT}} + \frac{\mathrm{d}U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R}_i)}{\mathrm{d}\mathbf{R}_i} \right|\right|^2 \ . \tag{2.31}$$

$\mathbf{F}^{\mathrm{AT}}$ is computed by summing the instantaneous AT forces of all atoms corresponding to the same CG particle [32].

Note that in the limit of infinite data, the loss in eq. (2.31) can be decomposed into the loss of the PMF $\mathcal{L}^{FM}(\boldsymbol{\theta}_{\mathrm{PMF}})$ plus the deviation of $U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ from $U^{\mathrm{PMF}}(\mathbf{R})$ [32]. Unlike for $\mathcal{L}^{AT}(\boldsymbol{\theta})$ (eq. (2.18)), there exists an unknown lower bound of the loss $\mathcal{L}^{FM}(\boldsymbol{\theta}_{\mathrm{PMF}}) > 0$ as a result of the information loss due to the non-injective CG mapping $\mathbf{M}(\mathbf{r})$. $\mathcal{L}^{FM}(\boldsymbol{\theta}_{\mathrm{PMF}})$

only depends on $\mathbf{M}(\mathbf{r})$ and is not a function of $\boldsymbol{\theta}$. Hence, in the limit of infinite data and model capacity, minimization of $\mathcal{L}^{FM}(\boldsymbol{\theta})$ causes $U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ to converge to $U^{\mathrm{PMF}}(\mathbf{R})$.

2.3.3. Relative Entropy Minimization

The Kullback-Leibler (KL) divergence [173]

$$D_{\mathrm{KL}}(p||q) = \int_{\mathbf{x}} p(\mathbf{x}) \ln\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \mathrm{d}\mathbf{x} \tag{2.32}$$

is a measure of the distance between two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$. Due to Gibbs' inequality, $D_{\mathrm{KL}}(p||q) \geq 0$. Hence, $D_{\mathrm{KL}}(p||q) = 0$ is the global minimum of the KL divergence, which is obtained if and only if $p(\mathbf{x}) = q(\mathbf{x})$. In the context of CG modeling, the KL divergence is referred to as relative entropy and inserting $p^{\mathrm{AT}}(\mathbf{R})$ and $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ for $p(\mathbf{x})$ and $q(\mathbf{x})$ leads to

$$S_{\mathrm{rel}}(\boldsymbol{\theta}) = \int_{\mathbf{R}} p^{\mathrm{AT}}(\mathbf{R}) \underbrace{\ln\left(\frac{p^{\mathrm{AT}}(\mathbf{R})}{p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})}\right)}_{\Phi_{\boldsymbol{\theta}}(\mathbf{R})} \mathrm{d}\mathbf{R} \ . \tag{2.33}$$

Hence, assuming infinite data and model capacity, minimizing $S_{\mathrm{rel}}(\boldsymbol{\theta})$ [171, 174–176] yields a CG model that is consistent with the underlying AT model.

Inserting the configurational probabilities corresponding to the NVT ensemble into eq. (2.33) yields [171, 177]

$$\mathcal{L}^{RE}(\boldsymbol{\theta}) = S_{\mathrm{rel}}(\boldsymbol{\theta}) = \beta\langle U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{M}(\mathbf{r})) - U^{\mathrm{AT}}(\mathbf{r})\rangle_{\mathrm{AT}} - \beta(A_{\boldsymbol{\theta}}^{\mathrm{CG}} - A^{\mathrm{AT}}) + S_{\mathrm{map}} \ , \tag{2.34}$$

where $\langle ... \rangle_{\mathrm{AT}}$ denotes an ensemble average with respect to the AT ensemble. S_{map} is independent of $\boldsymbol{\theta}$ and only depends on $M(\mathbf{r})$ [177]. Due to the difficulty of computing the difference in Helmholtz free energies $A_{\boldsymbol{\theta}}^{\mathrm{CG}} - A^{\mathrm{AT}}$, evaluating $S_{\mathrm{rel}}(U_{\boldsymbol{\theta}}^{\mathrm{CG}})$ is non-trivial. However, gradient descent optimization only requires the gradient $\nabla_{\boldsymbol{\theta}} S_{\mathrm{rel}}(\boldsymbol{\theta})$, which can be estimated by time averages over AT and CG trajectories [177]:

$$\nabla_{\boldsymbol{\theta}} S_{\mathrm{rel}}(\boldsymbol{\theta}) = \frac{\beta}{N_{\mathrm{box}}} \sum_{i=1}^{N_{\mathrm{box}}} \nabla_{\boldsymbol{\theta}} U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{M}(\mathbf{r}_i)) - \frac{\beta}{N} \sum_{j=1}^{N} \nabla_{\boldsymbol{\theta}} U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R}_j) \ . \tag{2.35}$$

The first term in eq. (2.35) is an average over the AT data set. The computationally expensive part of computing $\nabla_{\boldsymbol{\theta}} S_{\mathrm{rel}}(\boldsymbol{\theta})$ is the second term, which averages over a trajectory generated by the current potential $U_{\boldsymbol{\theta}}^{\mathrm{CG}}$. Hence, after every gradient descent update, the CG trajectory needs to be re-generated. In order to reduce the computational effort, reweighting (eq. (2.15)) can be employed to re-use CG trajectories [171, 177, 178].

The relationship between FM and RE minimization has been studied extensively [168, 177, 179, 180]. In particular, RE and FM minimize a different functional of $\Phi_{\boldsymbol{\theta}}(\mathbf{R})$ (eq. (2.33)), which causes convergence to different loss minima for finite model capacity. As a result, the converged RE model reproduces all structural correlations of $p^{\mathrm{AT}}(\mathbf{R})$ conjugate to the basis functions of the potential [177], while no similar guarantees exist for FM [179].

2.4. Probabilistic Molecular Modeling

The goal of the training schemes introduced so far was to approximate the optimal parameter set $\bar{\boldsymbol{\theta}}$ (eq. (2.17)). However, by considering multiple parameter sets $\{\boldsymbol{\theta}_i\}_{i=1}^{N_{\mathrm{models}}}$, the mean prediction from different models may be more robust and the distribution of predictions can be used for UQ. This section first outlines the Bayesian modeling framework in a generic learning problem with training data set $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N_{\mathrm{obs}}}$ of size N_{obs}. Then, the likelihood and the prior distributions will be tailored to the case of molecular energy matching with a NN potential.

2.4.1. Bayesian Uncertainty Quantification

Bayesian statistics provides the mathematical foundation of UQ. In this framework, a probabilistic model $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ predicts the distribution of the output $\mathbf{y}$ as a result of aleatoric uncertainty, i.e. stochastic noise inherent to the modeled process that is irreducible. For example, if the aleatoric uncertainty is independent, homoscedastic and Gaussian with variance σ^2, then $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}, \sigma^2\mathbf{I})$, where the mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ is predicted by the (ML) model. Additionally, Bayesian modeling quantifies the epistemic uncertainty [70, 181], i.e. uncertainty due to a finite data set, which can be reduced by gathering more data. To this end, Bayesian UQ marginalizes over $\boldsymbol{\theta}$ [182], weighting the prediction of each model $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i)$ by its posterior probability $p(\boldsymbol{\theta}_i|\mathcal{D})$. Thus, Bayesian inference aims to estimate the posterior predictive distribution

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \ , \tag{2.36}$$

which needs to be approximated by the Monte Carlo method due to the high dimensionality of $\boldsymbol{\theta}$:

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \approx \frac{1}{N_{\mathrm{models}}} \sum_{i=1}^{N_{\mathrm{models}}} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_i) \ ; \ \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\mathcal{D}) \ . \tag{2.37}$$

The Monte Carlo method requires sampling from $p(\boldsymbol{\theta}|\mathcal{D})$, which is given by Bayes' theorem

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{D})} \propto \exp\left(-\mathcal{U}(\boldsymbol{\theta})\right) \ , \tag{2.38}$$

where $p(\mathcal{D}|\boldsymbol{\theta})$ is referred to as likelihood and $p(\boldsymbol{\theta})$ is referred to as prior. In the second expression, $p(\boldsymbol{\theta}|\mathcal{D})$ is re-written as a Boltzmann-type distribution [105] (eq. (2.9)), with posterior potential energy

$$\mathcal{U}(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \ . \tag{2.39}$$

Due to the assumption of independently distributed data points, the likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ is the product of the probability of each observation of the data set given $\boldsymbol{\theta}$. In analogy to statistical mechanics, the Boltzmann-type form of the posterior enables sampling via Hamiltonian (also called hybrid [104]) Monte Carlo (HMC) [105] techniques. HMC leverages the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta})$ to perform short Hamiltonian simulations in parameter space to generate a decorrelated proposal with high MH [103] acceptance probability, which is necessary in order to scale to high-dimensional $\boldsymbol{\theta}$ [105].

2.4.2. Bayesian Molecular Modeling

In the case of an AT energy matching task, the probabilistic model is $p(U|\mathbf{r},\boldsymbol{\theta}) \sim \mathcal{N}(U_{\boldsymbol{\theta}}(\mathbf{r}), \sigma_{\mathrm{H}}^2)$, under the assumption of independent Gaussian homoscedastic aleatoric uncertainty with variance σ_{H}^2. In this case, the likelihood evaluates to [90]

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{i=1}^{N_{\mathrm{box}}} \frac{1}{\sqrt{2\pi\sigma_{\mathrm{H}}^2}} \exp\left(-\frac{[U_i - U_{\boldsymbol{\theta}}(\mathbf{r}_i)]^2}{2\sigma_{\mathrm{H}}^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma_{\mathrm{H}}^2}}\right)^{N_{\mathrm{box}}} \exp\left(-\frac{\sum_{i=1}^{N_{\mathrm{box}}}[U_i - U_{\boldsymbol{\theta}}(\mathbf{r}_i)]^2}{2\sigma_{\mathrm{H}}^2}\right) . \end{aligned} \tag{2.40}$$

The aleatoric uncertainty is usually small for energy matching given that it corresponds to uncertainty of the energy labels as predicted by the data-generating model. However, the scale σ_{H} is unknown a priori and is typically modeled as a learnable parameter. Consequently, the prior $p(\boldsymbol{\theta}) = p(\mathbf{w})p(\sigma_{\mathrm{H}})$ is composed of a prior for the potential energy model, e.g. NN potential weights and biases $\mathbf{w}$, and a prior for σ_{H}.

In molecular modeling, the data set is usually large and NN potentials are comparatively expensive to evaluate. Given that the computation of $\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta})$ for a single step of the Hamiltonian simulation of HMC requires the evaluation of the NN potential over the whole data set, this renders HMC computationally intractable for learning NN potentials [90]. Hence, more scalable UQ schemes are required in this case: Stochastic gradient MCMC (SG-MCMC) [183–187] and Stochastic Variational Inference [188, 189] operate on mini-batches of data, analogous to stochastic gradient descent, enabling scalable Bayesian UQ. However, also the non-Bayesian [70, 89, 181, 190] Deep Ensemble [88, 89] method is applicable to NN potentials [90, 191]

2.5. Adaptive Resolution Simulation

In the case where both an AT and a CG model are available for a system of interest, concurrent multiscale modeling provides an attractive method to combine the accuracy and high resolution of the AT model with the computational efficiency of the CG model [192]. This approach can be interpreted as a computational magnifying glass [193], where a local region of interest, e.g. near a binding site [194] or near an interaction with a lipid membrane [195], is fully resolved while only large scale behaviour of the rest of the system is modeled. Hence, concurrent multiscale modeling can be useful to identify the necessary components of a system essential to the physical process under investigation [193].

Concurrent multiscale simulations can be distinguished into constant [196–199] and adaptive [200–202] resolution methods. This section focuses on the latter, as they have the advantage of allowing free diffusion of particles and, accordingly, online change of resolution.

In particular, the Adaptive Resolution Scheme [200, 203–206] (AdResS) will be discussed, which divides the simulation domain into AT and CG regions with a hybrid (Δ) region in between (fig. 2.6).

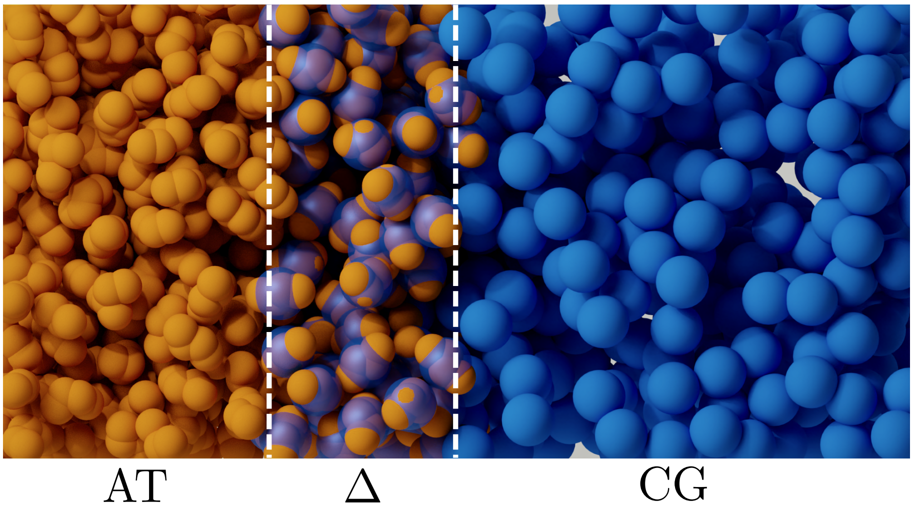


Figure 2.6.: Adaptive Resolution Scheme. Visualization of the hybrid region (Δ) between the atomistic (AT) and coarse-grained (CG) regions in the Adaptive Resolution Scheme.

In an AdResS simulation, the following relation describes the force $\mathbf{F}_i$ acting on molecule i [200]:

$$\mathbf{F}_i = \sum_{j \in \mathfrak{N}(i)} \lambda(\mathbf{R}_i)\lambda(\mathbf{R}_j)\mathbf{F}_{ij}^{AT} + \sum_{j \in \mathfrak{N}(i)} [1 - \lambda(\mathbf{R}_i)\lambda(\mathbf{R}_i)]\mathbf{F}_{ij}^{CG} + \mathbf{F}^{TD}(\mathbf{R}_i)$$
$$\text{with} \quad \lambda(\mathbf{R}) = \begin{cases} 1 \text{ if } \mathbf{R} \in \text{AT} \\ 0 \text{ if } \mathbf{R} \in \text{CG} \\ 0 < \lambda(\mathbf{R}) < 1 \text{ if } \mathbf{R} \in \Delta \end{cases} . \tag{2.41}$$

The interaction between a pair of molecules is defined by the weight $\lambda(\mathbf{R})$. If both molecules are within the CG (AT) region, they interact exclusively via the CG force field $\mathbf{F}_{ij}^{CG}$ (AT force field $\mathbf{F}_{ij}^{AT}$). In all other cases, $\lambda(\mathbf{R})$ smoothly interpolates between the AT and CG forces. Additionally, the external thermodynamic force $\mathbf{F}^{TD}(\mathbf{R})$ compensates the difference in chemical potential between the AT and CG force fields. It is obtained in an a-priori iterative optimization to achieve a constant density profile across the resolution interface.

The Δ region enables a smooth transition from one resolution to another. However, this comes at the cost of incorrect predictions within the Δ region [193] as well as increased computational effort as both AT and CG models need to be evaluated [207, 208].

3. Publications

This chapter introduces the peer-reviewed research papers this cumulative dissertation builds upon.

3.1. Training Neural Network Potentials

First, research articles are presented that improve training of NN potentials. The common denominator of these articles is the incorporation of MD simulations into the training pipeline, which allows to match experimental data (sec. 3.1.1) and improve the reliability and accuracy of obtained NN potentials (sec. 3.1.2).

3.1.1. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting

Thaler, S. & Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. *Nat. Commun.* **12**, 6884 (2021), DOI: 10.1038/s41467-021-27241-4.

Summary

The large model capacity of recently developed NN potential architectures enables molecular modeling at quantum chemical accuracy. With model capacity becoming an increasingly less restrictive factor, the performance of NN potentials hinges on the quality of the training data. Even though, in principle, NN potentials can be optimized via bottom-up or top-down learning, the vast majority of works relies on bottom-up training. End-to-end differentiation of MD observables with respect to potential energy parameters by backpropagating through the dynamics of MD simulations enables top-down training of NN potentials. However, this direct backpropagation approach results in excessive memory usage and is prone to exploding gradients.

This article addresses the need for NN potentials trained on experimental data by introducing the Differentiable Trajectory Reweighting (DiffTRe) method. DiffTRe offers end-to-end automatic gradient computation and circumvents the need to differentiate through the MD simulation by re-formulating the learning problem. At the core of DiffTRe is the reweighting scheme, which establishes a direct functional relation between MD observables and the NN potential parameters. By differentiating through the reweighting scheme rather than through the dynamics of the MD simulation, DiffTRe avoids exploding gradients and reduces the computational effort of gradient computations by around two orders of magnitude. Due to the comparatively large computational cost of generating MD trajectories during training, DiffTRe has to converge within a few hundred parameter updates compared to typically millions of updates in bottom-up learning. In order to speed-up convergence, we augment the NN potential by a prior potential, which encodes a-priori known physical principles – lowering the burden to learn these relations from scratch. This is a different purpose compared to energy and/or force matching, where the prior potential should enforce reasonable predictions outside the training data distribution.

We empirically verify the theoretically expected computational speed-up, memory savings and superior gradient stability compared to backpropagation through the simulation based on a toy example of ideal gas particles in a double-well potential. We showcase the broad applicability of DiffTRe by training a DimeNet++ graph NN potential for two real-world systems of water and diamond. The learned models yield MD simulation results in unprecedented agreement with a diverse set of target experimental observables, including thermodynamic, structural and mechanical properties.

Main advantages of DiffTRe include its general applicability and simplicity: DiffTRe enables training CG and AT models of arbitrary functional form. Additionally, practitioners are only required to provide a MD simulation and target observables, while DiffTRe computes gradients conveniently in an end-to-end fashion. Furthermore, DiffTRe generalizes structural CG methods such as iterative Boltzmann inversion to higher body-order correlations, extending their applicability to many-body potentials. Finally, the demonstrated computational efficiency of DiffTRe promotes its application to larger systems in the future. DiffTRe provides a means to systematically enhance NN potentials with experimental data, which is particularly relevant for larger systems where CQM data are unavailable.

CRediT author statement

Stephan Thaler: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing

Julija Zavadlav: Conceptualization, Supervision, Visualization, Writing – original draft, Writing – review & editing

Copyright notice

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-27241-4 OPEN

Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting

Stephan Thaler [1]✉ & Julija Zavadlav [1,2]✉

In molecular dynamics (MD), neural network (NN) potentials trained bottom-up on quantum mechanical data have seen tremendous success recently. Top-down approaches that learn NN potentials directly from experimental data have received less attention, typically facing numerical and computational challenges when backpropagating through MD simulations. We present the Differentiable Trajectory Reweighting (DiffTRe) method, which bypasses differentiation through the MD simulation for time-independent observables. Leveraging thermodynamic perturbation theory, we avoid exploding gradients and achieve around 2 orders of magnitude speed-up in gradient computation for top-down learning. We show effectiveness of DiffTRe in learning NN potentials for an atomistic model of diamond and a coarse-grained model of water based on diverse experimental observables including thermodynamic, structural and mechanical properties. Importantly, DiffTRe also generalizes bottom-up structural coarse-graining methods such as iterative Boltzmann inversion to arbitrary potentials. The presented method constitutes an important milestone towards enriching NN potentials with experimental data, particularly when accurate bottom-up data is unavailable.

[1] Professorship of Multiscale Modeling of Fluid Materials, TUM School of Engineering and Design, Technical University of Munich, Munich, Germany. [2] Munich Data Science Institute, Technical University of Munich, Munich, Germany. ✉email: stephan.thaler@tum.de; julija.zavadlav@tum.de

Molecular modeling has become a cornerstone of many disciplines, including computational chemistry, soft matter physics, and material science. However, simulation quality critically depends on the employed potential energy model that defines particle interactions. There are two distinct approaches for model parametrization[1,2]: Bottom-up approaches aim at matching data from high-fidelity simulations, providing labeled data of atomistic configurations with corresponding target outputs. Labeled data allow straightforward differentiation for gradient-based optimization, at the expense of inherently limiting model accuracy to the quality imposed by the underlying data-generating simulation. On the other hand, top-down approaches optimize the potential energy model such that simulations match experimental data. From experiments, however, labeled data on the atomistic scale are not available. Experimental observables are linked only indirectly to the potential model via an expensive molecular mechanics simulation, complicating optimization.

A class of potentials with tremendous success in recent years are neural network (NN) potentials due to their flexibility and capacity of learning many-body interactions[3,4]. The vast majority of NN potentials are trained via bottom-up methods[5–16]. The objective is to match energies and/or forces from a data set, most commonly generated via density functional theory (DFT) for small molecules in vacuum[17]. Within the data set distribution, state-of-the-art NN potentials have already reached the accuracy limit imposed by DFT, with the test error in predicting potential energy being around two orders of magnitude smaller than the corresponding expected DFT accuracy[11,18]. In the limit of a sufficiently large data set without a distribution shift[19,20] with respect to the application domain (potentially generated via active learning approaches[21]), remaining deviations of predicted observables from experiments are attributable to uncertainty in DFT simulations[11]—in line with literature reporting DFT being sensitive to employed functionals[22]. More precise computational quantum mechanics models, e.g., the coupled cluster CCSD(T) method, improve DFT accuracy at the expense of significantly increased computational effort for data set generation[23,24]. However, for larger systems such as macromolecules, quantum mechanics computations will remain intractable in the foreseeable future, preventing ab initio dataset generation altogether. Thus, the main obstacle in bottom-up learning of NN potentials is the currently limited availability of highly precise and sufficiently broad data sets.

Top-down approaches circumvent the need for reliable data-generating simulations. Leveraging experimental data in the potential optimization process has contributed greatly to the success of classical atomistic[25,26] and coarse-grained[27] (CG) force fields[1]. Training difficulties have so far impeded a similar approach for NN potentials: Only recent advances in automatic differentiation (AD)[28] software have enabled end-to-end differentiation of molecular dynamics (MD) observables with respect to potential energy parameters[29,30], by applying AD through the dynamics of a MD simulation[29–32]. This direct reverse-mode AD approach saves all simulator operations on the forward pass to be used during gradient computation on the backward pass, resulting in excessive memory usage. Thus, direct reverse-mode AD for systems with more than hundred particles and a few hundred time steps is typically intractable[29–32]. Numerical integration of the adjoint equations[33,34] represents a memory-efficient alternative that requires to save only those atomic configurations that directly contribute to the loss. However, both approaches backpropagate the gradient through the entire simulation, which dominates computational effort and is prone to exploding gradients, as stated by Ingraham et al.[31] and shown below.

Addressing the call for NN potentials trained on experimental data[1], we propose the Differentiable Trajectory Reweighting (DiffTRe) method. DiffTRe offers end-to-end gradient computation and circumvents the need to differentiate through the simulation by combining AD with previous work on MD reweighting schemes[35–38]. For the common use case of time-independent observables, DiffTRe avoids exploding gradients and reduces the computational effort of gradient computations by around two orders of magnitude compared to backpropagation through the simulation. Memory requirements are comparable to the adjoint method. We showcase the broad applicability of DiffTRe on three numerical test cases: First, we provide insight into the training process on a toy example of ideal gas particles inside a double-well potential. Second, we train the state-of-the-art graph neural network potential DimeNet++[11,12] for an atomistic model of the diamond from its experimental stiffness tensor. Finally, we learn a DimeNet++ model for CG water based on pressure, as well as radial and angular distribution functions. The last example shows how DiffTRe also generalizes bottom-up structural coarse-graining methods such as the iterative Boltzmann inversion[39] or inverse Monte Carlo[40] to many-body correlation functions and arbitrary potentials. DiffTRe allows to enhance NN potentials with experimental data, which is particularly relevant for systems where bottom-up data are unavailable or not sufficiently accurate.

Results

Differentiable Trajectory Reweighting. Top-down potential optimization aims to match the K outputs of a molecular mechanics simulation $\boldsymbol{O}$ to experimental observables $\tilde{\boldsymbol{O}}$. Therefore, the objective is to minimize a loss function $L(\boldsymbol{\theta})$, e.g., a mean-squared error (MSE)

$$L(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K}\left[\langle O_k(U_{\boldsymbol{\theta}})\rangle - \tilde{O}_k\right]^2, \tag{1}$$

where $\langle\rangle$ denotes the ensemble average, and $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ depends on the potential energy $U_{\boldsymbol{\theta}}$ parametrized by $\boldsymbol{\theta}$. We will focus on the case where a MD simulation approximates $\langle O_k(U_{\boldsymbol{\theta}})\rangle$—with Monte Carlo[41] being a usable alternative. With standard assumptions on ergodicity and thermodynamic equilibrium, the ensemble average $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ is approximated via a time average

$$\langle O_k(U_{\boldsymbol{\theta}})\rangle \simeq \frac{1}{N}\sum_{i=1}^{N} O_k(\mathbf{S}_i, U_{\boldsymbol{\theta}})\,, \tag{2}$$

where $\{\mathbf{S}_i\}_{i=1}^{N}$ is the trajectory of the system, i.e., a sequence of N states consisting of particle positions and momenta. Due to the small time step size necessary to maintain numerical stability in MD simulations, states are highly correlated. Subsampling, i.e., only averaging over every 100th or 1000th state, reduces this correlation in Eq. (2).

As the generated trajectory depends on $\boldsymbol{\theta}$, every update of $\boldsymbol{\theta}$ during training would require a re-computation of the entire trajectory. However, by leveraging thermodynamic perturbation theory[42], it is possible to re-use decorrelated states obtained via a reference potential $\hat{\boldsymbol{\theta}}$. Specifically, the time average is reweighted to account for the altered state probabilities $p_{\boldsymbol{\theta}}(\mathbf{S}_i)$ from the perturbed potential $\boldsymbol{\theta}$[35,36,42]:

$$\langle O_k(U_{\boldsymbol{\theta}})\rangle \simeq \sum_{i=1}^{N} w_i O_k(\mathbf{S}_i, U_{\boldsymbol{\theta}}) \quad \text{with} \quad w_i = \frac{p_{\boldsymbol{\theta}}(\mathbf{S}_i)/p_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i)}{\sum_{j=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{S}_j)/p_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j)}. \tag{3}$$

Assuming a canonical ensemble, state probabilities follow the Boltzmann distribution $p_{\boldsymbol{\theta}}(\mathbf{S}_i) \sim e^{-\beta H(\mathbf{S}_i)}$, where $H(\mathbf{S}_i)$ is the Hamiltonian of the state (sum of potential and kinetic energy), $\beta = 1/(k_B T)$, k_B Boltzmann constant, T temperature. Inserting $p_{\boldsymbol{\theta}}(\mathbf{S}_i)$ into Eq. (3) allows computing weights as a function of $\boldsymbol{\theta}$

(the kinetic energy cancels)

$$w_i = \frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{\sum_{i=j}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}}. \qquad (4)$$

For the special case of $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$, $w_i = 1/N$, recovering Eq. (2). Note that similar expressions to Eq. (4) could be derived for other ensembles, e.g., the isothermal–isobaric ensemble, via respective state probabilities $p_{\boldsymbol{\theta}}(\mathbf{S}_i)$. In practice, the reweighting ansatz is only applicable given small potential energy differences. For large differences between $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$, by contrast, few states dominate the average. In this case, the effective sample size[37]

$$N_{\text{eff}} \approx e^{-\sum_{i=1}^{N} w_i \ln(w_i)} \qquad (5)$$

is reduced and the statistical error in $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ increases (Eq. (3)).

Reweighting can be exploited for two purposes that are linked to speedups in the forward and backward pass, respectively: first, reweighting reduces computational effort as decorrelated states from previous trajectories can often be re-used[37]. Second, and most importantly, reweighting establishes a direct functional relation between $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ and $\boldsymbol{\theta}$. This relation via $\mathbf{w}$ provides an alternative end-to-end differentiable path for computing the gradient of the loss $\nabla_{\boldsymbol{\theta}} L$: differentiating through the reweighting scheme replaces the backward pass through the simulation. Leveraging this alternative differentiation path, while managing the effective sample size N_{eff}, are the central ideas behind the DiffTRe method.

The workflow of the DiffTRe algorithm consists of the following steps: first, an initial reference trajectory is generated from the canonical ensemble, e.g., via a stochastic or deterministic thermostat, from an initial state $\mathbf{S}_{\text{init}}$ and reference potential $\hat{\boldsymbol{\theta}}$ (Fig. 1a). Initial equilibration states are disregarded and the following states are subsampled yielding decorrelated states $\{\mathbf{S}_i\}_{i=1}^{N}$. Together with their reference potential energies $\{U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i)\}_{i=1}^{N}$, these states are saved for re-use during reweighting. In the next step, the reweighting scheme is employed to compute $\nabla_{\boldsymbol{\theta}} L$ with respect to current parameters $\boldsymbol{\theta}$, where initially $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$. An optimizer subsequently uses $\nabla_{\boldsymbol{\theta}} L$ to improve $\boldsymbol{\theta}$. This procedure of reweighting, gradient computation and updating is repeated as long as the statistical error from reweighting is acceptably small, i.e., N_{eff} is larger than a predefined $\bar{N}_{\text{eff}}$. As soon as $N_{\text{eff}} < \bar{N}_{\text{eff}}$, a new reference trajectory needs to be sampled using the current $\boldsymbol{\theta}$ as the new $\hat{\boldsymbol{\theta}}$. At least one $\boldsymbol{\theta}$ update per reference trajectory is ensured because initially $N_{\text{eff}} = N$. Using the last generated state $\mathbf{S}_N$ as $\mathbf{S}_{\text{init}}$ for the next trajectory counteracts overfitting to a specific initial configuration. In addition, $p_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_{\text{init}})$ is reasonably high when assuming small update steps, reducing necessary equilibration time for trajectory generation. Saving only $\{\mathbf{S}_i\}_{i=1}^{N}$ and $\{U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i)\}_{i=1}^{N}$ from the simulation entails low-memory requirements similar to the adjoint method. DiffTRe assumes that deviations in predicted observables are attributable to an inaccurate potential $U_{\boldsymbol{\theta}}$ rather than a statistical sampling error. Accordingly, N and the subsampling ratio n need to be chosen to yield a sufficiently small statistical error. Optimal values for N and n depend on the specific system, target observables, and the thermodynamic-state point.

Computation of $\nabla_{\boldsymbol{\theta}} L$ via reverse-mode AD through the reweighting scheme comprises a forward pass starting with computation of the potential $U_{\boldsymbol{\theta}}(\mathbf{S}_i)$ and weight w_i for each $\mathbf{S}_i$ (Eq. (4); Fig. 1a). Afterward, reweighted observables $\langle O_k(U_{\boldsymbol{\theta}})\rangle$ (Eq. (3)) and the resulting loss $L(\boldsymbol{\theta})$ (Eq. (1)) are calculated. The corresponding backward pass starts at $L(\boldsymbol{\theta})$ and stops at parameters $\boldsymbol{\theta}$ in the potential energy computation $U_{\boldsymbol{\theta}}(\mathbf{S}_i)$. The differentiation path defined by the reweighting ansatz is therefore independent of the trajectory generation.

Evaluation of $U_{\boldsymbol{\theta}}(\mathbf{S}_i)$ (Fig. 1b) involves computing the pairwise distance matrix $\mathbf{D}$ from atom positions of $\mathbf{S}_i$, that are fed into a learnable potential $U_{\boldsymbol{\theta}}^{\text{model}}$ and a prior potential U^{prior}. Both potential components are combined by adding the predicted potential energies

$$U_{\boldsymbol{\theta}}(\mathbf{S}_i) = U_{\boldsymbol{\theta}}^{\text{model}}(\mathbf{D}) + U^{\text{prior}}(\mathbf{D}). \qquad (6)$$

In subsequent examples of diamond and CG water, $U_{\boldsymbol{\theta}}^{\text{model}}$ is a graph neural network operating iteratively on the atomic graph defined by $\mathbf{D}$. U^{prior} is a constant potential approximating a priori-known properties of the system, such as the Pauli exclusion principle (e.g., Eq. (12)). Augmenting NN potentials with a prior

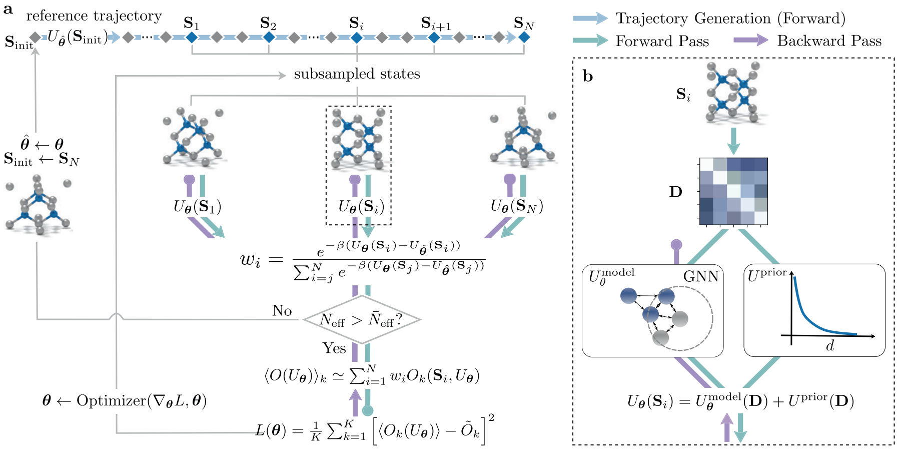

a
reference trajectory
$\mathbf{S}_{\text{init}}$
$U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_{\text{init}})$
$\mathbf{S}_1$
$\mathbf{S}_2$
$\mathbf{S}_i$
$\mathbf{S}_{i+1}$
$\mathbf{S}_N$
Trajectory Generation (Forward)
Forward Pass
Backward Pass
subsampled states
$\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}$
$\mathbf{S}_{\text{init}} \leftarrow \mathbf{S}_N$
$U_{\boldsymbol{\theta}}(\mathbf{S}_1)$
$U_{\boldsymbol{\theta}}(\mathbf{S}_i)$
$U_{\boldsymbol{\theta}}(\mathbf{S}_N)$
$w_i = \frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{\sum_{i=j}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}}$
No
$N_{\text{eff}} > \bar{N}_{\text{eff}}$?
Yes
$\langle O(U_{\boldsymbol{\theta}})\rangle_k \simeq \sum_{i=1}^{N} w_i O_k(\mathbf{S}_i, U_{\boldsymbol{\theta}})$
$\boldsymbol{\theta} \leftarrow \text{Optimizer}(\nabla_{\boldsymbol{\theta}} L, \boldsymbol{\theta})$
$L(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K}\left[\langle O_k(U_{\boldsymbol{\theta}})\rangle - \tilde{O}_k\right]^2$
b
$\mathbf{S}_i$
$\mathbf{D}$
$U_{\boldsymbol{\theta}}^{\text{model}}$
GNN
U^{prior}
d
$U_{\boldsymbol{\theta}}(\mathbf{S}_i) = U_{\boldsymbol{\theta}}^{\text{model}}(\mathbf{D}) + U^{\text{prior}}(\mathbf{D})$

Fig. 1 Differentiable Trajectory Reweighting (DiffTRe). **a** Based on an initial state $\mathbf{S}_{\text{init}}$ and reference potential parameters $\hat{\boldsymbol{\theta}}$, a reference trajectory is generated, of which only subsampled states are retained (blue diamonds), while the majority of visited states are discarded (gray diamonds). For each retained state $\mathbf{S}_i$ (represented by a generic molecular system), the potential energy $U_{\boldsymbol{\theta}}(\mathbf{S}_i)$ and weight w_i are computed under the current potential parameters $\boldsymbol{\theta}$. w_i allow computation of reweighted observables $\langle O_k(U_{\boldsymbol{\theta}})\rangle$, the loss $L(\boldsymbol{\theta})$, its gradient $\nabla_{\boldsymbol{\theta}} L$ and subsequently, updating $\boldsymbol{\theta}$ via the optimizer. The updating procedure is repeated until the effective sample size $N_{\text{eff}} < \bar{N}_{\text{eff}}$, at which point a new reference trajectory needs to be generated starting from the last sampled state $\mathbf{S}_N$. **b** Computation of $U_{\boldsymbol{\theta}}(\mathbf{S}_i)$ from the pairwise distance matrix $\mathbf{D}$, which is fed into the learnable potential $U_{\boldsymbol{\theta}}^{\text{model}}$ (e.g., a graph neural network—GNN) and U^{prior} (e.g., a pairwise repulsive potential).

is common in the bottom-up coarse-graining literature[8,10] to provide qualitatively correct behavior in regions of the potential energy surface (PES) not contained in the dataset, but reachable by the CG model. By contrast, DiffTRe does not rely on precomputed data sets. Rather, the prior serves to control the data (trajectory) generation in the beginning of the optimization. In addition, U^{prior} reformulates the problem from learning $U_{\boldsymbol{\theta}}^{\mathrm{model}}$ directly to learning the difference between U^{prior} and the optimal potential given the data[10]. A well-chosen U^{prior} therefore represents a physics-informed initialization accelerating training convergence. Suitable U^{prior} can often be found in the literature: Classical force fields such as AMBER[25] and MARTINI[27] define reasonable interactions for bio-molecules and variants of the Embedded Atom Model[43] (EAM) provide potentials for metals and alloys. Note that U^{prior} is not a prior in the Bayesian sense providing a pervasive bias on learnable parameters in the small data regime. If U^{prior} is in contradiction with the data, $U_{\boldsymbol{\theta}}^{\mathrm{model}}$ will correct for U^{prior} as a result of the optimization. In the next section, we further illustrate for a toy problem the interplay between prior, gradients and the learning process in DiffTRe, and provide a comparison to direct reverse-mode AD through the simulation.

Double-well toy example. We consider ideal gas particles at a temperature $k_BT = 1$ trapped inside a one-dimensional double-well potential (Fig. 2a) parametrized by

$$U(x) = k_BT * \left[2500(x-0.5)^6 - 10(x-0.55)^2\right]. \tag{7}$$

The goal is to learn $\boldsymbol{\theta}$ such that $U_{\boldsymbol{\theta}}(x) = U_{\boldsymbol{\theta}}^{\mathrm{model}}(x) + U^{\mathrm{prior}}(x)$ matches $U(x)$. We select a cubic spline as $U_{\boldsymbol{\theta}}^{\mathrm{model}}$, which acts as a flexible approximator for twice continuously differentiable functions. The cubic spline is parametrized via the potential energy values of 50 control points $\{x_j, U_j\}_{j=1}^{50}$ evenly distributed over $x \in [0, 1]$. Analogous to NN potentials in subsequent problems, we randomly initialize $U_j \sim \mathcal{N}(0, 0.01^2 k_BT)$. Initializing $U_j = 0$ leads to largely identical results in this toy problem. The harmonic single-well potential $U^{\mathrm{prior}}(x) = \lambda(x - 0.5)^2$, with scale $\lambda = 75$, encodes the prior knowledge that particles cannot escape the double-well. We choose the normalized density profile $\rho(x)/\rho_0$ of ideal gas particles as the target observable. The resulting loss function is

$$L = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\langle\rho(x_k)\rangle}{\rho_0} - \frac{\tilde{\rho}(x_k)}{\rho_0}\right)^2, \tag{8}$$

where $\rho(x)$ is discretized via K bins. $\langle\rho(x_k)\rangle$ are approximated based on $N = 10{,}000$ states after skipping 1000 states for equilibration, where a state is retained every 100 time steps. We minimize Eq. (8) via an Adam[44] optimizer with learning rate decay. For additional DiffTRe and simulation parameters, see Supplementary Method 1.1.

Initially, ρ/ρ_0 resulting from $U^{\mathrm{prior}}(x)$ deviates strongly from the target double-well density (Fig. 2b). The loss curve illustrates successful optimization over 200 update steps (Fig. 2c). The wall-clock time per parameter update Δt clearly shows two distinct levels: at the start of the optimization, update steps are rather large, significantly reducing N_{eff}. Hence a new reference trajectory generation is triggered with each update (average $\Delta t \approx 39.2$ s). Over the course of the simulation, updates of $U_{\boldsymbol{\theta}}^{\mathrm{model}}(x)$ become smaller and reference trajectories are occasionally re-used (average $\Delta t \approx 2.76$ s). After optimization, the target density is matched well. The learned potential energy function $U_{\boldsymbol{\theta}}(x)$ recovers the data-generating potential $U(x)$ (Supplementary Fig. 1a); thus, other thermodynamic and kinetic observables will match reference values closely. However, this conclusion does not

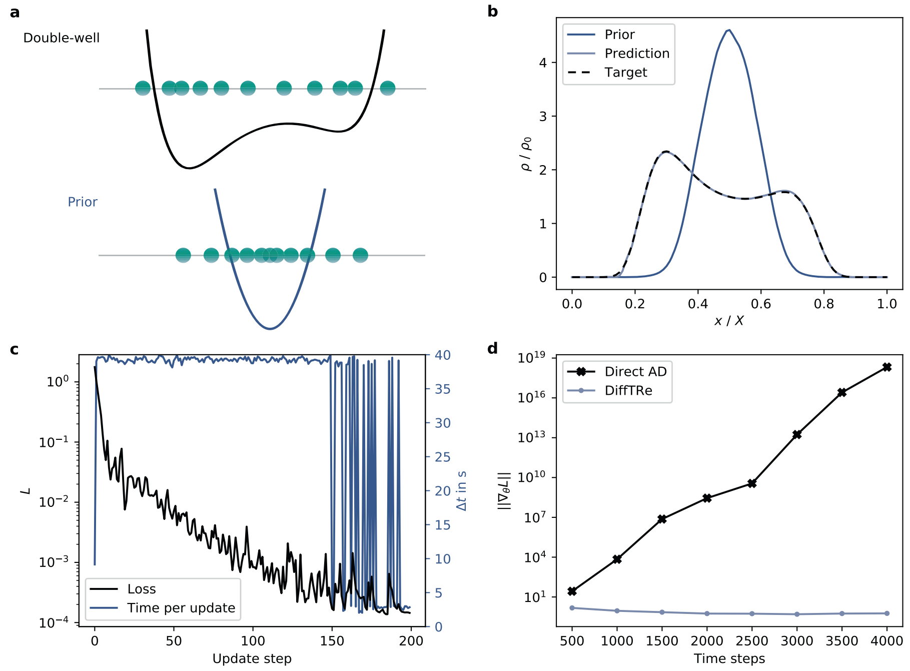


Fig. 2 Double-well toy example. a Sketch of the double-well and prior potential with corresponding example states of ideal gas particles (green circles). The learned potential results in a normalized density ρ/ρ_0 (over the normalized position x/X) that matches the target closely (**b**). Successful learning is reflected in the loss curve L, where a significant reduction in wall-clock time per parameter update Δt towards the end of the optimization is achieved through re-using previously generated trajectories (**c**). Gradients computed via DiffTRe have constant magnitudes while gradients obtained from direct reverse-mode automatic differentiation through the simulation suffer from exploding gradients for longer trajectories (**d**).

apply in realistic applications, where learned potentials are in general not unique[2] due to the limited number of target observables that can be considered in practice.

The effect of U^{prior} on the training process is twofold: First, by encoding prior knowledge, it simplifies convergence, as $U_{\boldsymbol{\theta}}^{\text{model}}(x)$ only needs to adapt the single-well prior instead of learning large energy barriers from scratch. Second, U^{prior} also impacts the information content of the gradient by controlling the generation of trajectories in the beginning of the optimization (Eq. (15)). The local support of the cubic spline allows analyzing this relation empirically (Supplementary Fig. 2): The gradient is nonzero only in regions of the PES that are included in the reference trajectory. Hence, other regions of the PES are not optimized despite delivering a nonzero contribution to the loss. A well-chosen prior potential should therefore yield trajectories that are as close as possible to trajectories sampled from the true potential. However, satisfactory learning results can be obtained for a sensible range of prior scales (Supplementary Fig. 3).

We study the robustness of our results by varying the random seed that controls the initialization of the spline as well as the initial particle positions and velocities. Results from the variation study in Supplementary Fig. 4 demonstrate that the predicted $\rho(x)/\rho_0$ is robust to the random initialization. The corresponding $U_{\boldsymbol{\theta}}(x)$ exhibits some variance at the left well boundary, mirroring difficult training in this region due to vanishing gradients for vanishing predicted densities (Supplementary Fig. 2) and minor influence of the exact wall position on the resulting density profile (Supplementary Fig. 4a).

For comparison, we have implemented gradient computation via direct reverse-mode AD through the simulation. This approach clearly suffers from the exploding gradients problem (Fig. 2d): The gradient magnitude increases exponentially as a function of the simulation length. Without additional modifications (e.g., as implemented by Ingraham et al.[31]), these gradients are impractical for longer trajectories. By contrast, gradients computed via DiffTRe show constant magnitudes irrespective of the simulation length.

To measure the speed-up over direct reverse-mode AD empirically, we simulate the realistic case of an expensive potential by substituting the numerically inexpensive spline with a fully connected neural network with two hidden layers and 100 neurons each. We measure speedups of $s_g = 486$ for gradient computations and $s = 3.7$ as overall speed-up per update when a new reference trajectory is sampled. However, these values are rather sensitive to the exact computational and simulation setup. Memory overflow in the direct AD method constrained trajectory lengths to ten retained states and a single state for equilibration (a total of 1100 time steps). Measuring speed-up for one of the real-world problems below would be desirable, but is prevented by the memory requirements of direct AD.

The measured speed-up values are in line with theoretical considerations: While direct AD backpropagates through the whole trajectory generation, DiffTRe only differentiates through the potential energy computation of decorrelated states $\{\mathbf{S}_i\}_{i=1}^{N}$ (Fig. 1). From this algorithmic difference, we expect speed-up values that depend on the subsampling ratio n, the number of skipped states during equilibration N_{equilib} and the cost multiple of backward passes with respect to forward passes G (details in Supplementary Method 2)

$$s_g \sim Gn\left(1 + N_{\text{equilib}}/N\right)\,; \quad s \sim G + 1\,. \tag{9}$$

For this toy example setup, the rule-of-thumb estimates in Eq. (9) yield $s_g = 330$ and $s = 4$, agreeing with the measured values. In the next sections, we showcase the effectiveness of DiffTRe in real-world, top-down learning of NN potentials.

Atomistic model of diamond. To demonstrate the applicability of DiffTRe to solids on the atomistic scale, we learn a DimeNet++[12] potential for diamond from its experimental elastic stiffness tensor **C**. Due to symmetries in the diamond cubic crystal, **C** only consists of three distinct stiffness moduli $\tilde{C}_{11} = 1079$ GPa, $\tilde{C}_{12} = 124$ GPa and $\tilde{C}_{44} = 578$ GPa[45] (in Voigt notation). In addition, we assume the crystal to be in a stress-free state $\boldsymbol{\sigma} = \mathbf{0}$ for vanishing infinitesimal strain $\boldsymbol{\epsilon} = \mathbf{0}$. These experimental data define the loss

$$L = \frac{\gamma_{\boldsymbol{\sigma}}}{9}\sum_{i=1,j=1}^{i=3,j=3}\sigma_{ij}^2 + \frac{\gamma_{\mathbf{C}}}{3}\left(\left(C_{11} - \tilde{C}_{11}\right)^2 + \left(C_{12} - \tilde{C}_{12}\right)^2 + \left(C_{44} - \tilde{C}_{44}\right)^2\right), \tag{10}$$

where loss weights $\gamma_{\boldsymbol{\sigma}}$ and $\gamma_{\mathbf{C}}$ counteract the effect of different orders of magnitude of observables. To demonstrate learning, we select the original Stillinger–Weber potential[46] parametrized for silicon as U^{prior}. We have adjusted the length and energy scales to $\sigma_{\text{SW}} = 0.14$ nm and $\epsilon_{\text{SW}} = 200$ kJ/mol, reflecting the smaller size of carbon atoms. We found learning to be somewhat sensitive to U^{prior} in this example because weak prior choices can lead to unstable MD simulations. Simulations are run with a cubic box of size $L \approx 1.784$ nm containing 1000 carbon atoms (Fig. 3a) to match the experimental density ($\rho = 3512\,\text{kg/m}^3$)[45] exactly. The temperature in the experiment ($T = 298.15$ K[45]) determines the simulation temperature. Each trajectory generation starts with 10 ps of equilibration followed by 60 ps of production, where a decorrelated state is saved every 25 fs. We found these trajectories to yield observables with acceptably small statistical noise. The stress tensor $\boldsymbol{\sigma}$ is computed via Eq. (13) and the stiffness tensor **C** via the stress fluctuation method (Eq. (14)). Further details are summarized in Supplementary Method 4.

Figure 3 visualizes convergence of the stress (b) and stiffness components (c). Given that the model is only trained on rather short trajectories, we test the trained model on a trajectory of 10 ns length to ensure that the model neither overfitted to initial conditions nor drifts away from the targets. The resulting stress and stiffness values $\sigma_1 = 0.29$ GPa, $\sigma_4 = 0.005$ GPa, $C_{11} = 1070$ GPa, $C_{12} = 114$ GPa, and $C_{44} = 560$ GPa are in good agreement with respective targets. These results could be improved by increasing the trajectory length, which reduces statistical sampling errors. The corresponding inverse stress–strain relation is given by the compliance tensor $\mathbf{S} = \mathbf{C}^{-1}$, which can be constructed from by Young's modulus $E = 1047$ GPa, shear modulus $G = 560$ GPa, and Poisson's ratio $\nu = 0.097$. The training loss curve and wall-clock time per update Δt are displayed in Supplementary Fig. 5a.

Computing the stress–strain curve (Supplementary Fig. 5b) from the trained model in the linear regime ($\epsilon_i < 0.005$) verifies that computing **C** via Eq. (14) yields the same result as explicitly straining the box and measuring stresses. In addition, this demonstrates that the DimeNet++ potential generalizes from the training box ($\boldsymbol{\epsilon} = 0$) to boxes under small strain. We also strained the box beyond the linear regime, creating a distribution shift[19,20], to test generalization to unobserved state points. The predicted stress–strain curve in Fig. 3d shows good agreement with DFT data[47] for medium-sized natural strains $e_1 = \log(1 + \epsilon_1) < 0.02$. For large strains, the deviation quickly increases, including an early fracture. These incorrect predictions of the learned potential are due to limited extrapolation capacities of NN potentials: states under large strain are never encountered during training, leading to large uncertainty in predicted forces. Incorporating additional observables linked to states of large strain into the optimization, such as the point of maximum stress, should improve predictions.

To test the trained DimeNet++ potential on held-out observables, we compute the phonon density of states (PDOS).

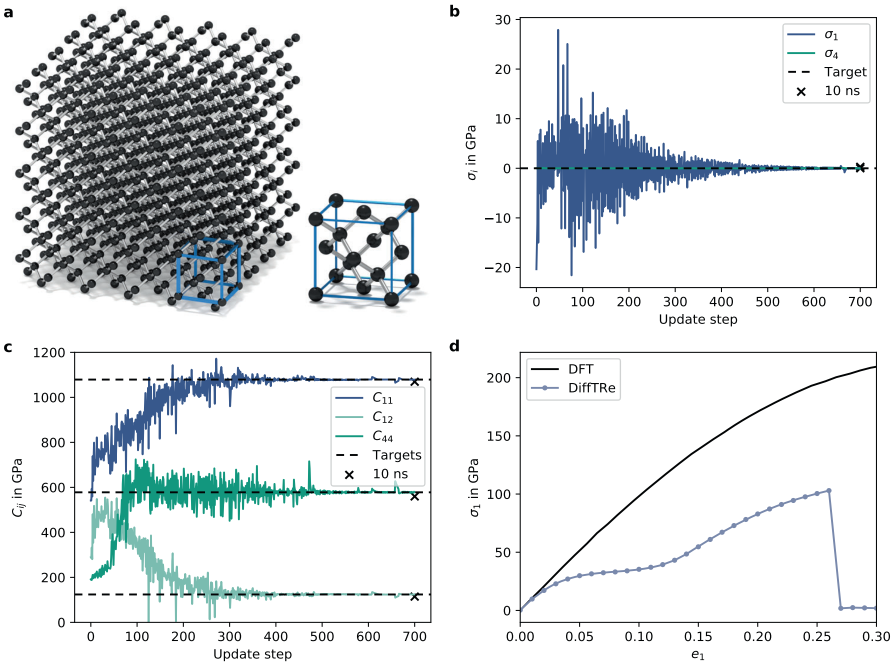


Fig. 3 Atomistic model of the diamond. The simulation box consists of five diamond unit cells in each direction, whose primary crystallographic directions [1, 0, 0], [0, 1, 0] and [0, 0, 1] are aligned with the x, y, and z axes of the simulation box (**a**). Stress σ_i (**b**) and stiffness values C_{ij} (**c**) converge to their respective targets during the optimization. These results are robust to long simulation runs of 10 ns (marked with crosses). The stress–strain curve over normal natural strains e_1 agrees with density functional theory (DFT) data[47] for medium-sized strains ($e_1 <= 0.02$), but deviates for large strains due to limited extrapolation capabilities of neural network potentials (**d**).

The predicted PDOS deviates from the experiment[48], analogous to a Stillinger–Weber potential optimized for diamond[49] (Supplementary Fig. 5c). The evolution of the predicted PDOS over the course of the optimization is shown in Supplementary Fig. 5d. Deviations of held-out observables are expected given that top-down approaches allow learning potentials that are consistent with target experimental observables but lack theoretical convergence guarantees of bottom-up schemes (in the limit of a sufficiently large data set and a sufficiently expressive model)[2]. In principle, we expect sufficiently expressive top-down models to converge to the true potential in the limit of an infinite number of matched target observables. In practice, however, many different potentials can reproduce a sparse set of considered target observables, rendering the learned potential non-unique[2]. In this particular example, we show that many different potentials can reproduce the target stress and stiffness, but predict different PDOSs: While predicted stress and stiffness values are robust to random initialization of NN weights and initial particle velocities within the statistical sampling error, the corresponding predicted PDOSs vary to a great extent (Supplementary Fig. 6). Incorporating additional observables more closely connected to phonon properties into the loss function could improve the predicted PDOS.

Coarse-grained water model. Finally, we learn a DimeNet++ potential for CG water. Water is a common benchmark problem due to its relevance in bio-physics simulations and its pronounced 3-body interactions, which are challenging for classical potentials[50]. We select a CG-mapping, where each CG particle is centered at the oxygen atom of the corresponding atomistic water molecule (Fig. 4a). This allows using experimental oxygen–oxygen radial (RDF) and angular distribution functions (ADF) as target observables. Given that the reference experiment[51] was carried out at ambient conditions ($T = 296.15$ K), we can additionally target a pressure $\tilde{p} = 1$ bar. Hence, we minimize

$$L = \frac{1}{G}\sum_{g=1}^{G}\left(RDF(d_g) - \tilde{RDF}(d_g)\right)^2 + \frac{1}{M}\sum_{m=1}^{M}\left(ADF(\alpha_m) - \tilde{ADF}(\alpha_m)\right)^2 + \gamma_p(p - \tilde{p})^2. \tag{11}$$

As the prior potential, we select the repulsive term of the Lennard–Jones potential

$$U^{\text{prior}}(d) = \epsilon_R\left(\frac{\sigma_R}{d}\right)^{12}. \tag{12}$$

Drawing inspiration from atomistic water models, we have chosen the length scale of the SPC[52] water model as $\sigma_R = 0.3165$ nm as well as a reduced energy scale of $\epsilon_R = 1$ kJ/mol to counteract the missing Lennard–Jones attraction term in Eq. (12). We build a cubic box of length 3 nm with 901 CG particles, implying a density of $\rho = 998.28$ g/l, to match the experimental water density of $\rho = 997.87$ g/l at 1 bar. Trajectory generation consists of 10 ps of equilibration and 60 ps of subsequent production, where a decorrelated state is saved every 0.1 ps. For additional details, see Supplementary Method 2.3.

Figure 4b–d displays properties predicted by the final trained model during a 10 ns production run: DiffTRe is able to train a DimeNet++ potential that simultaneously matches experimental oxygen RDF, ADF, and pressure to the line thickness. The evolution of predicted RDFs and ADFs as well as the loss and wall-clock times per update are displayed in Supplementary Fig. 7a–c. The learning process is robust to weak choices of U^{prior}: DiffTRe is able to converge to the same prediction quality as with the reference prior even if σ_R is misestimated by ±0.1 nm (approximately ±30%) compared to the classical SPC water model (Supplementary Fig. 8a, b). This represents a large variation given

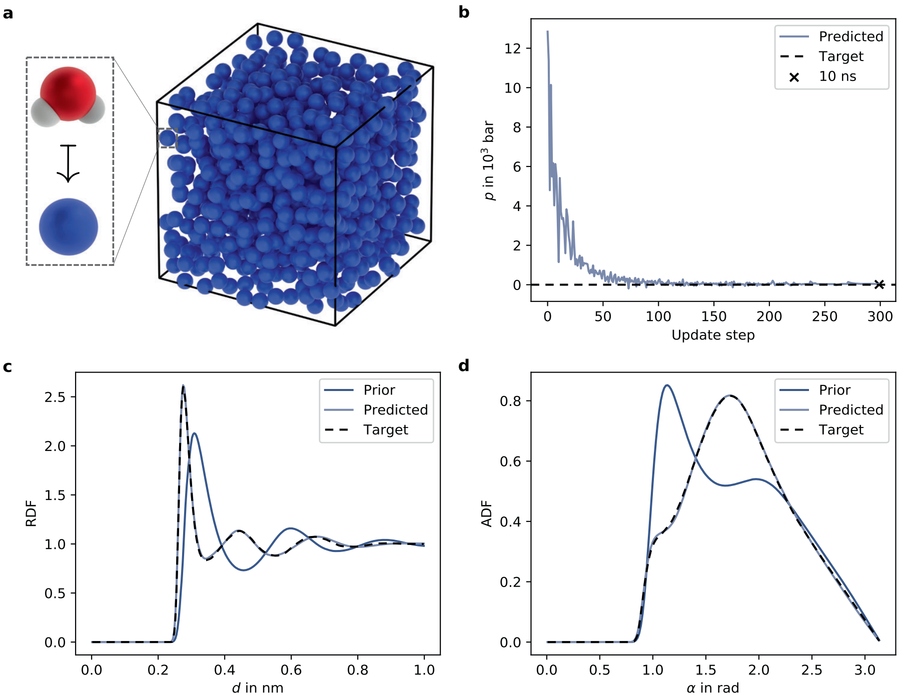


Fig. 4 Coarse-grained model of water. Coarse-grained particles representing water molecules are visualized as blue balls in the simulation box (**a**). The pressure p converges quickly toward its target of 1 bar during optimization and the subsequent 10 ns simulation (black cross; $p \approx 12.9$ bar) verifies the result (**b**). Over a 10 ns simulation, the learned potential reconstructs the experimental radial distribution function (RDF) and angular distribution function (ADF) well (**c**, **d**).

that within common atomistic water models, σ_R varies by <0.5%[53].

To test the learned potential on held-out observables, we compute the tetrahedral order parameter q[54] and the self-diffusion coefficient D. $q \approx 0.569$ matches the experimental value of $\tilde{q} = 0.576$ closely. This is expected as q considers the structure of four nearest neighbor particles, which is closely related to the ADF. The learned CG water model predicts a larger self-diffusion coefficient than were experimentally measured ($D = 10.91\ \mu\text{m}^2/\text{ms}$ vs. $\tilde{D} = 2.2\ \mu\text{m}^2/\text{ms}$)[55]. With the same simulation setup, a single-site tabulated potential parametrized via iterative Boltzmann inversion[39] with pressure correction[39,56] predicts $D = 14.15\ \mu\text{m}^2/\text{ms}$. These results are in line with the literature: Due to smoother PESs, CG models exhibit accelerated dynamical processes compared to atomistic models[2]. For CG water models specifically, diffusion coefficients decrease with increasing number of interaction sites[57]. In this context, the decreasing diffusion coefficients over the course of the optimization (Supplementary Fig. 7d) could indicate that $U_{\boldsymbol{\theta}}$ acts effectively as a single-site model in the beginning, while learning 3-body interactions during the optimization casts $U_{\boldsymbol{\theta}}$ more similar to multisite CG models. Obtained results are robust to random initialization of NN weights and initial particle velocities, both for predicted target (Supplementary Fig. 8c, d) and held-out observables ($D = 10.93 \pm 0.20\ \mu\text{m}^2/\text{ms}$).

The accuracy of predicted 2 and 3-body interactions (Fig. 4c, d) showcases the potency of graph neural network potentials in top-down molecular modeling: capturing 3-body interactions is essential for modeling water given that pair potentials trained via force matching fail to reproduce both RDF and ADF of the underlying high-fidelity model[50]. Other top-down CG water models with simple functional form tend to deviate from the experimental RDF[58,59]. Deviations from experimental structural properties, albeit smaller in size, also arise in DFT simulations[22,60], limiting the accuracy of bottom-up trained NN potentials[8].

Discussion

In this work, we demonstrate numerically efficient learning of NN potentials from experimental data. The main advantages of our proposed DiffTRe method are its flexibility and simplicity: DiffTRe is applicable to solid and fluid materials, coarse-grained and atomistic models, thermodynamic, structural and mechanical properties, as well as potentials of arbitrary functional form. To apply DiffTRe, practitioners only need to set up a MD simulation with corresponding observables and a loss function, while gradients are computed conveniently in an end-to-end fashion via AD. The demonstrated speedups and limited memory requirements promote application to larger systems.

Without further adaptations, DiffTRe can also be applied as a bottom-up model parametrization scheme. In this case, a high-fidelity simulation, rather than an experiment, provides target observables. For CG models, DiffTRe generalizes structural coarse-graining schemes such as iterative Boltzmann inversion[39] or Inverse Monte Carlo[40]. DiffTRe overcomes the main limitations of these approaches: First, structural coarse-graining is no longer restricted to one-dimensional potentials, and matching many-body correlation functions (e.g., ADFs) is therefore feasible. Second, the user can integrate additional observables into the optimization without relying on hand-crafted iterative update rules, for instance for pressure-matching[39,56]. This is particularly useful if an observable needs to be matched precisely (e.g., pressure in certain multiscale simulations[61]). Matching many-body correlation functions will likely allow structural bottom-up coarse-graining to take on significance within the new paradigm of many-body CG potentials[8–10].

For the practical application of DiffTRe, a few limitations need to be considered. The reweighting scheme renders DiffTRe invariant to the sequence of states in the trajectory. Hence, dynamical properties cannot be employed as target observables. In addition, the NN potential test cases considered in this work required a reasonably chosen prior potential. Lastly, two distinct sources of overfitting when learning from experimental data for a single system need to be accounted for[1]: To avoid overfitting to a specific initial state, DiffTRe uses a different initial state for each reference trajectory. Moreover, increasing the system size and trajectory length ensures representative reference trajectories. Irrespective of overfitting, generalization to different systems, observables, and thermodynamic-state points remains to be addressed, for instance via training on multi-systemic experimental data sets. To this end, an in-depth assessment of out-of-sample properties of top-down learned NN potentials is required.

From a machine learning (ML) perspective, DiffTRe belongs to the class of end-to-end differentiable physics approaches[62–64]. These approaches are similar to reinforcement learning in that the target outcome of a process (here a MD simulation) represents the data. A key difference is the availability of gradients through the process, allowing for efficient training. Differentiable physics approaches, increasingly popular in control applications[34,65–67], enable direct training of the ML model via the physics simulator, advancing the ongoing synthesis of ML and physics-based methods.

Finally, the combination of bottom-up and top-down approaches for learning NN potentials, i.e., considering information from both the quantum and macroscopic scale, represents an exciting avenue for future research. For top-down approaches, pre-training NN potentials on bottom-up data sets can serve as a sensible extrapolation for the PES in areas unconstrained by the experimental data. In DiffTRe, a pre-trained model could also circumvent the need for a prior potential. Bottom-up trained NN potentials, on the other hand, can be enriched with experimental data, which enables targeted refinement of the potential. This is particularly helpful for systems in which DFT accuracy is insufficient or the generation of a quantum mechanical data set is computationally intractable.

Methods

Differentiable histogram binning. To obtain an informative gradient $\frac{\partial L}{\partial \boldsymbol{\theta}}$, predicted observables need to be continuously differentiable. However, many common observables in MD, including density and structural correlation functions, are computed by discrete histogram binning. To obtain a differentiable observable, the (discrete) Dirac function used in binning can be approximated by a narrow Gaussian probability density function (PDF)[34]. Similarly, we smooth the non-differentiable cutoff in the definition of ADFs via a Gaussian cumulative distribution function (CDF) centered at the cutoff (details on differentiable density, RDF, and ADF in Supplementary Method 3).

Stress-strain relations. Computing the virial stress tensor $\boldsymbol{\sigma}^V$ for many-body potentials, e.g., NN potentials, under periodic boundary conditions requires special attention. This is due to the fact that most commonly used formulas are only valid for non-periodic boundary conditions or pairwise potentials[68]. Therefore, we resort to the formulation proposed by Chen et al.[69], which is well suited for vectorized computations in NN potentials.

$$\boldsymbol{\sigma}^V = \frac{1}{\Omega}\left[-\sum_{k=1}^{N_p} m_k \mathbf{v}_k \otimes \mathbf{v}_k - \mathbf{F}^T\mathbf{R} + \left(\frac{\partial U}{\partial \mathbf{h}}\right)^T \mathbf{h}\right], \tag{13}$$

where N_p is the number of particles, $\otimes$ represents the dyadic or outer product, m_k and $\mathbf{v}_k$ are mass and thermal excitation velocity of particle k, $\mathbf{R}$ and $\mathbf{F}$ are $(N_\mathrm{p} \times 3)$ matrices containing all particle positions and corresponding forces, $\mathbf{h}$ is the (3×3) lattice tensor spanning the simulation box, and $\Omega = \det(\mathbf{h})$ is the box volume.

Due to the equivalence of the ensemble-averaged virial stress tensor $\langle \boldsymbol{\sigma}^V \rangle$ and the Cauchy stress tensor $\boldsymbol{\sigma}$[70], we can compute the elastic stiffness tensor from MD simulations and compare it to continuum mechanical experimental data (details in Supplementary Method 5). In the canonical ensemble, the isothermal elastic stiffness tensor $\mathbf{C}$ can be calculated at constant strain $\boldsymbol{\epsilon}$ via the stress fluctuation method[71]:

$$C_{ijkl} = \frac{\partial \langle \sigma_{ij}^V \rangle}{\partial \epsilon_{kl}} = \langle C_{ijkl}^B \rangle - \Omega\beta\Big(\langle \sigma_{ij}^B \sigma_{kl}^B \rangle - \langle \sigma_{ij}^B \rangle \langle \sigma_{kl}^B \rangle\Big) + \frac{N_p}{\Omega\beta}\Big(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}\Big), \tag{14}$$

with the Born contribution to the stress tensor $\sigma_{ij}^B = \frac{1}{\Omega}\frac{\partial U}{\partial \epsilon_{ij}}$, the Born contribution to the stiffness tensor $C_{ijkl}^B = \frac{1}{\Omega}\frac{\partial^2 U}{\partial \epsilon_{ij} \partial \epsilon_{kl}}$ and Kronecker delta δ_{ij}. Eq. (14) integrates well into DiffTRe by reweighting individual ensemble average terms (Eq. (3)) and combining the reweighted averages afterwards. Implementing the stress fluctuation method in differentiable MD simulations is straightforward: AD circumvents manual derivation of strain-derivatives, which is non-trivial for many-body potentials[72].

Statistical mechanics foundations. Thermodynamic fluctuation formulas allow to compute the gradient $\frac{\partial L}{\partial \boldsymbol{\theta}}$ from ensemble averages[73–75]. Specifically, considering a MSE loss for a single observable $O(U_{\boldsymbol{\theta}})$ in the canonical ensemble[73],

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = 2(\langle O(U_{\boldsymbol{\theta}})\rangle - \tilde{O})\left[\left\langle \frac{\partial O(U_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} \right\rangle - \beta\left(\left\langle O(U_{\boldsymbol{\theta}}) \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right\rangle - \langle O(U_{\boldsymbol{\theta}})\rangle \left\langle \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \right\rangle\right)\right]. \tag{15}$$

It can be seen that the AD routine in DiffTRe estimates $\frac{\partial L}{\partial \boldsymbol{\theta}}$ by approximating ensemble averages in Eq. (15) via reweighting averages (Derivation in Supplementary Method 5). End-to-end differentiation through the reweighting scheme simplifies optimization by combining obtained gradients from multiple observables. This is particularly convenient for observables that are not merely averages of instantaneous quantities, e.g., the stiffness tensor $\mathbf{C}$ (Eq. (14)).

DimeNet++. We employ a custom implementation of DimeNet++[11,12] that fully integrates into Jax MD[29]. Our implementation takes advantage of neighbor lists for efficient computation of the sparse atomic graph. We select the same NN hyperparameters as in the original publication[12] except for the embedding sizes, which we reduced by factor 4. This modification allowed for a significant speed-up while retaining sufficient capacity for the problems considered in this work. For diamond, we have reduced the cutoff to 0.2 nm yielding an atomic graph, where each carbon atom is connected to its four covalently bonded neighbors. A comprehensive list of employed DimeNet++ hyperparameters is provided in Supplementary Method 6.

Data availability

Simulation setups and trained DimeNet++ models have been deposited in https://github.com/tummfm/difftre. The data generated in this study are provided in the paper or in the Supplementary information file.

Code availability

The code for DiffTRe and its application to the three test cases is available at https://github.com/tummfm/difftre[76].

Received: 2 June 2021; Accepted: 9 November 2021;
Published online: 25 November 2021

References

1. Fröhlking, T., Bernetti, M., Calonaci, N. & Bussi, G. Toward empirical force fields that match experimental observables. *J. Chem. Phys.* **152**, 230902 (2020).
2. Noid, W. G. Perspective: coarse-grained models for biomolecular systems. *J. Chem. Phys.* **139**, 090901 (2013).
3. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R. & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).
4. Noé, F., Tkatchenko, A., Müller, K. R. & Clementi, C. Machine learning for molecular simulation. *Annu. Rev. Phys. Chem.* **71**, 361–390 (2020).
5. Behler, J. & Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **98**, 146401 (2007).
6. Schütt, K. T. et al. SchNet: a continuous-filter convolutional neural network for modeling quantum interactions. in *Advances in Neural Information Processing Systems* Vol. 30, 992–1002 (Curran Associates, Inc., 2017).
7. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. in *Proceedings of the 34th International Conference on Machine Learning*. 1263–1272 (PMLR, 2017).
8. Zhang, L., Han, J., Wang, H., Car, R. & Weinan, W. E. DeePCG: constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **149**, 034101 (2018).

9. Wang, J. et al. Machine learning of coarse-grained molecular dynamics force fields. *ACS Cent. Sci.* **5**, 755–767 (2019).
10. Husic, B. E. et al. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **153**, 194101 (2020).
11. Klicpera, J., Groß, J. & Günnemann, S. Directional message passing for molecular graphs. In *8th International Conference on Learning Representations, ICLR* (2020).
12. Klicpera, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. in *Machine Learning for Molecules Workshop at NeurIPS* (2020).
13. Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R. & Miller, T. F. OrbNet: deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **153**, 124111 (2020).
14. Vlachas, P. R., Zavadlav, J., Praprotnik, M. & Koumoutsakos, P. Accelerated simulations of molecular systems through learning of their effective dynamics. Preprint at https://arxiv.org/abs/2011.14115 (2021).
15. Jain, A. C. P., Marchand, D., Glensk, A., Ceriotti, M. & Curtin, W. A. Machine learning for metallurgy III: a neural network potential for Al-Mg-Si. *Phys. Rev. Mater.* **5**, 053805 (2021).
16. Ko, T. W., Finkler, J. A., Goedecker, S. & Behler, J. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **12**, 398 (2021).
17. Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **1**, 1–7 (2014).
18. Faber, F. A. et al. Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).
19. Cubuk, E. D. & Schoenholz, S. S. Adversarial forces of physical models. in *3rd NeurIPS workshop on Machine Learning and the Physical Sciences* (2020).
20. Schwalbe-Koda, D., Tan, A. R. & Gómez-Bombarelli, R. Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nat. Commun.* **12**, 5104 (2021).
21. Zhang, L., Lin, D.-Y., Wang, H., Car, R. & Weinan, E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Mater.* **3**, 023804 (2019).
22. Gillan, M. J., Alfè, D. & Michaelides, A. Perspective: how good is DFT for water? *J. Chem. Phys.* **144**, 130901 (2016).
23. Smith, J. S. et al. Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **10**, 2930 (2019).
24. Sauceda, H. E., Vassilev-Galindo, V., Chmiela, S., Müller, K. R. & Tkatchenko, A. Dynamical strengthening of covalent and non-covalent molecular interactions by nuclear quantum effects at finite temperature. *Nat. Commun.* **12**, 442 (2021).
25. Cornell, W. D. et al. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *J. Am. Chem. Soc.* **117**, 5179–5197 (1995).
26. Oostenbrink, C., Villa, A., Mark, A. E. & Van Gunsteren, W. F. A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameter sets 53A5 and 53A6. *J. Comput. Chem.* **25**, 1656–1676 (2004).
27. Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P. & De Vries, A. H. The MARTINI force field: coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **111**, 7812–7824 (2007).
28. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 1–43 (2018).
29. Schoenholz, S. S. & Cubuk, E. D. JAX MD: A Framework for Differentiable Physics. in *Advances in Neural Information Processing Systems* Vol. 33, 11428–11441 (Curran Associates, Inc., 2020).
30. Doerr, S. et al. Torchmd: a deep learning framework for molecular simulations. *J. Chem. Theory Comput.* **17**, 2355–2363 (2021).
31. Ingraham, J., Riesselman, A., Sander, C. & Marks, D. Learning protein structure with a differentiable simulator. in *7th International Conference on Learning Representations, ICLR* (2019).
32. Goodrich, C. P., King, E. M., Schoenholz, S. S., Cubuk, E. D. & Brenner, M. P. Designing self-assembling kinetics with differentiable statistical physics models. *Proc. Natl Acad. Sci. USA* **118**, e2024083118 (2021).
33. Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. Neural ordinary differential equations. in *Advances in Neural Information Processing Systems* Vol. 31 (Curran Associates, Inc., 2018).
34. Wang, W., Axelrod, S. & Gómez-Bombarelli, R. Differentiable molecular simulations for control and learning. in *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*. (2020).
35. Norgaard, A. B., Ferkinghoff-Borg, J. & Lindorff-Larsen, K. Experimental parameterization of an energy function for the simulation of unfolded proteins. *Biophys. J.* **94**, 182–192 (2008).
36. Li, D. W. & Brüschweiler, R. Iterative optimization of molecular mechanics force fields from NMR data of full-length proteins. *J. Chem. Theory Comput.* **7**, 1773–1782 (2011).
37. Carmichael, S. P. & Shell, M. S. A new multiscale algorithm and its application to coarse-grained peptide models for self-assembly. *J. Phys. Chem. B* **116**, 8383–8393 (2012).
38. Wang, L. P., Chen, J. & Van Voorhis, T. Systematic parametrization of polarizable force fields from quantum chemistry data. *J. Chem. Theory Comput.* **9**, 452–460 (2013).
39. Reith, D., Pütz, M. & Müller-Plathe, F. Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.* **24**, 1624–1636 (2003).
40. Lyubartsev, A. P. & Laaksonen, A. Calculation of effective interaction potentials from radial distribution functions: a reverse Monte Carlo approach. *Phys. Rev. E* **52**, 3730–3737 (1995).
41. Binder, K., Heermann, D., Roelofs, L., Mallinckrodt, A. J. & McKay, S. Monte Carlo simulation in statistical physics. *Comput. Phys.* **7**, 156 (1993).
42. Zwanzig, R. W. High-temperature equation of state by a perturbation method. I. Nonpolar gases. *J. Chem. Phys.* **22**, 1420–1426 (1954).
43. Daw, M. S. & Baskes, M. I. Embedded-atom method: derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* **29**, 6443 (1984).
44. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. in *3rd International Conference on Learning Representations, ICLR* (2015).
45. McSkimin, H. J., Andreatch, P. & Glynn, P. The elastic stiffness moduli of diamond. *J. Appl. Phys.* **43**, 985–987 (1972).
46. Stillinger, F. H. & Weber, T. A. Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B* **31**, 5262–5271 (1985).
47. Jensen, B. D., Wise, K. E. & Odegard, G. M. Simulation of the elastic and ultimate tensile properties of diamond, graphene, carbon nanotubes, and amorphous carbon using a revised reaxFF parametrization. *J. Phys. Chem. A* **119**, 9710–9721 (2015).
48. Dolling, G. & Cowley, R. A. The thermodynamic and optical properties of germanium, silicon, diamond and gallium arsenide. *Proc. Phys. Soc.* **88**, 463 (1966).
49. Barnard, A. S., Russo, S. P. & Leach, G. I. Nearest neighbour considerations in stillinger-weber type potentials for diamond. *Mol. Simul.* **28**, 761–771 (2002).
50. Scherer, C. & Andrienko, D. Understanding three-body contributions to coarse-grained force fields. *Phys. Chem. Chem. Phys.* **20**, 22387–22394 (2018).
51. Soper, A. K. & Benmore, C. J. Quantum differences between heavy and light water. *Phys. Rev. Lett.* **101**, 065502 (2008).
52. Berendsen, H. J. C., Postma, J. P. M., van Gunsteren, W. F. & Hermans, J. Interaction models for water in relation to protein hydration. in *Intermolecular forces*. (ed. Pullman, B.) 331–342 (Springer, 1981).
53. Wu, Y., Tepper, H. L. & Voth, G. A. Flexible simple point-charge water model with improved liquid-state properties. *J. Chem. Phys.* **124**, 024503 (2006).
54. Errington, J. R. & Debenedetti, P. G. Relationship between structural order and the anomalies of liquid water. *Nature* **409**, 318–321 (2001).
55. Mills, R. Self-diffusion in normal and heavy water in the range 1-45°. *J. Phys. Chem.* **77**, 685–688 (1973).
56. Wang, H., Junghans, C. & Kremer, K. Comparative atomistic and coarse-grained study of water: what do we lose by coarse-graining? *Eur. Phys. J. E* **28**, 221–229 (2009).
57. Matysiak, S., Clementi, C., Praprotnik, M., Kremer, K. & Delle Site, L. Modeling diffusive dynamics in adaptive resolution simulation of liquid water. *J. Chem. Phys.* **128**, 024503 (2008).
58. Molinero, V. & Moore, E. B. Water modeled as an intermediate element between carbon and silicon. *J. Phys. Chem. B* **113**, 4008–4016 (2009).
59. Chan, H. et al. Machine learning coarse grained models for water. *Nat. Commun.* **10**, 379 (2019).
60. Distasio, R. A., Santra, B., Li, Z., Wu, X. & Car, R. The individual and collective effects of exact exchange and dispersion interactions on the ab initio structure of liquid water. *J. Chem. Phys.* **141**, 084502 (2014).
61. Thaler, S., Praprotnik, M. & Zavadlav, J. Back-mapping augmented adaptive resolution simulation. *J. Chem. Phys.* **153**, 164118 (2020).
62. Belbute-Peres, F. D. A., Smith, K. A., Allen, K. R., Tenenbaum, J. B. & Kolter, J. Z. End-to-end differentiable physics for learning and control. in *Advances in Neural Information Processing Systems* Vol. 31 (Curran Associates, Inc., 2018).
63. Innes, M. et al. A differentiable programming system to bridge machine learning and scientific computing. Preprint at https://arxiv.org/abs/1907.07587 (2019).
64. Hu, Y. et al. DiffTaichi: differentiable programming for physical simulation. in *8th International Conference on Learning Representations, ICLR* (2020).
65. Degrave, J., Hermans, M., Dambre, J. & Wyffels, F. A differentiable physics engine for deep learning in robotics. *Front. Neurorobot.* **13**, 6 (2019).
66. Holl, P., Koltun, V. & Thuerey, N. Learning to control PDEs with differentiable physics. in *8th International Conference on Learning Representations, ICLR* (2020).

67. Schäfer, F., Kloc, M., Bruder, C. & Lörch, N. A differentiable programming method for quantum control. *Mach. Learn. Sci. Technol.* **1**, 35009 (2020).
68. Thompson, A. P., Plimpton, S. J. & Mattson, W. General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *J. Chem. Phys.* **131**, 154107 (2009).
69. Chen, X. et al. TensorAlloy: an automatic atomistic neural network program for alloys. *Comput. Phys. Commun.* **250**, 107057 (2020).
70. Subramaniyan, A. K. & Sun, C. T. Continuum interpretation of virial stress in molecular simulations. *Int. J. Solids Struct.* **45**, 4340–4346 (2008).
71. Van Workum, K., Yoshimoto, K., De Pablo, J. J. & Douglas, J. F. Isothermal stress and elasticity tensors for ions and point dipoles using Ewald summations. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **71**, 061102 (2005).
72. Van Workum, K., Gao, G., Schall, J. D. & Harrison, J. A. Expressions for the stress and elasticity tensors for angle-dependent potentials. *J. Chem. Phys.* **125**, 144506 (2006).
73. Di Pierro, M. & Elber, R. Automated optimization of potential parameters. *J. Chem. Theory Comput.* **9**, 3311–3320 (2013).
74. Wang, L. P. et al. Systematic improvement of a classical molecular model of water. *J. Phys. Chem. B* **117**, 9956–9972 (2013).
75. Wang, L. P., Martinez, T. J. & Pande, V. S. Building force fields: an automatic, systematic, and reproducible approach. *J. Phys. Chem. Lett.* **5**, 1885–1891 (2014).
76. Thaler, S. & Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. https://github.com/tummfm/difftre, https://doi.org/10.5281/zenodo.5643099 (2021).

Author contributions

S.T. conceptualized, implemented, and applied the DiffTRe method and conducted MD simulations as well as postprocessing. S.T. and J.Z. planned the study, analyzed and interpreted the results, and wrote the paper.

Funding

Open Access funding enabled and organized by Projekt DEAL.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-27241-4.

Correspondence and requests for materials should be addressed to Stephan Thaler or Julija Zavadlav.

Peer review information *Nature Communications* thanks Ekin Cubuk and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

3.1.2. Deep coarse-grained potentials via relative entropy minimization

Thaler, S., Stupp, M. & Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *J. Chem. Phys.* **157**, 244103 (2022), DOI: 10.1063/5.0124538.

Summary

For CG systems, employing NN potentials promises highly accurate MD simulation results due their ability to approximate the many-body terms of the PMF. Even though RE minimization is the standard alternative to FM for the parametrization of classical CG potentials, bottom-up trained CG NN potentials have so far almost exclusively been trained via the latter scheme. Unlike classical CG potentials, limited capacity of the functional form is not the main accuracy bottleneck for CG NN potentials. Consequently, other error sources become dominant, especially finite data size effects. CG NN potentials trained via FM have been found to be prone to these effects: Given that FM NN potentials are unconstrained in phase-space regions that are unresolved by the training data, they rely heavily on prior potentials to avoid unphysical predictions and numerical instability. Additionally, FM models have been found to be weak in reconstructing the reference free energy surface (FES) as given by the AT training data.

This article demonstrates the applicability of RE minimization to train CG NN potentials and showcases that RE training is more data efficient than FM, reducing errors from finite data size significantly. First, a thought experiment shows that sensitivity with respect to rarely sampled transition regions can be the reason for suboptimal FES predictions of FM NN potentials and that the potential energy-based loss function of RE minimization is not susceptible to this error mechanism.

Second, the benchmark problem of CG liquid water represents the case when errors from limited model capacity and finite data size are small. In this case, both FM and RE yield highly accurate CG NN potentials, in particular compared to classical 2-body CG potentials. This is expected given that the obtained potential from both training schemes converges to the PMF for infinite data and model capacity. Interestingly, given that the two main error sources – limited model capacity and finite data size effects – are less pronounced in this problem, numerical errors can become the dominant error component. Unlike FM, RE minimization optimizes the potential with respect to the resulting distribution of the CG MD simulation. Hence, RE minimization can correct for numerical errors of the simulation, enabling larger time steps in CG MD simulations without compromising accuracy.

Finally, the benchmark problem of CG alanine dipeptide tests both training schemes in a finite data size setting. Here, unlike FM, RE is able to accurately reproduce the reference FES. This outperformance can be partially attributed to higher data efficiency of RE minimization, which achieves its accuracy maximum already at a fraction of the data required for FM. Additionally, RE is less dependent on prior potentials given that sampling of unphysical phase-space results in a large S_{rel}, which can be corrected by subsequent gradient descent updates.

These results highlight the importance of including MD simulations into the training pipeline of NN potentials for improved accuracy and robustness. Developing novel CG NN potential training schemes beyond FM is a promising direction for future research.

CRediT author statement

Stephan Thaler: Conceptualization, Data curation, Formal analysis, Methodology, Visualization, Writing – original draft, Writing – review & editing

Maximilian Stupp: Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft

Julija Zavadlav: Conceptualization, Data curation, Supervision, Writing – original draft, Writing – review & editing

Copyright notice

The Journal of Chemical Physics ARTICLE scitation.org/journal/jcp

Deep coarse-grained potentials via relative entropy minimization

Cite as: J. Chem. Phys. 157, 244103 (2022); doi: 10.1063/5.0124538
Submitted: 6 September 2022 • Accepted: 23 November 2022 •
Published Online: 28 December 2022

Stephan Thaler, Maximilian Stupp, and Julija Zavadlav[a)]

AFFILIATIONS

Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, Munich, Germany

[a)] Author to whom correspondence should be addressed: julija.zavadlav@tum.de. Also at: Munich Data Science Institute and Munich Institute for Integrated Materials, Energy and Process Engineering, Technical University of Munich, Munich, Germany.

ABSTRACT

Neural network (NN) potentials are a natural choice for coarse-grained (CG) models. Their many-body capacity allows highly accurate approximations of the potential of mean force, promising CG simulations of unprecedented accuracy. CG NN potentials trained bottom-up via force matching (FM), however, suffer from finite data effects: They rely on prior potentials for physically sound predictions outside the training data domain, and the corresponding free energy surface is sensitive to errors in the transition regions. The standard alternative to FM for classical potentials is relative entropy (RE) minimization, which has not yet been applied to NN potentials. In this work, we demonstrate, for benchmark problems of liquid water and alanine dipeptide, that RE training is more data efficient, due to accessing the CG distribution during training, resulting in improved free energy surfaces and reduced sensitivity to prior potentials. In addition, RE learns to correct time integration errors, allowing larger time steps in CG molecular dynamics simulation, while maintaining accuracy. Thus, our findings support the use of training objectives beyond FM, as a promising direction for improving CG NN potential's accuracy and reliability.

I. INTRODUCTION

Molecular dynamics (MD) simulations are a popular tool for studying bio-physical processes at the nanoscale. For atomistic (AT) simulations, time and length scales of many processes of interest are still out of reach in currently available computational hardware. Coarse-graining[1–8] (CG) - grouping of AT particles into effective interaction beads is a common approach to model these systems as larger spatiotemporal scales can be reached due to a reduced number of interactions and an increased time step size.[7]

The fidelity of CG simulations strongly depends on the employed CG potential energy function that defines particle interactions. In the classical CG literature, potentials follow simple functional forms.[3,7] Recent years have seen an increased use of neural network (NN) potentials[9–19] for CG models, both—for bottom-up learning, to match properties of AT models,[20–26] and for top-down learning, to match experimental data.[27] In the following, we focus on the bottom-up learning case, with the aim of obtaining a CG model that is consistent with an existing AT model. Consistency is achieved if the distribution of CG states sampled from the CG model equals the distribution generated by the AT model when mapping the AT states to CG coordinates.[4] In this case, the CG potential equals the many-body potential of mean force (PMF).[4] Consequently, NN potentials are a natural choice for CG potential energy functions: Their many-body nature[28] allows for a more accurate approximation of the PMF than classical CG models, promising CG simulations of unprecedented accuracy.

So far, most bottom-up CG NN potentials have been trained via force matching (FM).[20–24] FM minimizes the difference between CG force predictions and corresponding target AT forces for a given dataset,[4,21] typically generated by an AT MD simulation. FM training, while computationally inexpensive and straightforward to implement, suffers from two problems caused by the low availability of high energy states.[29] First, reproducing the ratio of different metastable states proves difficult for CG NN potentials trained via FM.[24,26] The CG potential is thought to be susceptible to errors in the rarely sampled transition regions, which affects the global accuracy of the free energy surface (FES).[26] Second, NN potentials are physics-free universal function approximators. Thus, they rely heavily on prior potentials that enforce qualitatively correct force predictions outside the training data distribution, to avoid unphysical states, e.g., particle overlaps.[21,23] Both problems can cause erroneous results in subsequent CG MD simulations, but critically, their extent is not reflected in the FM validation error during training.[26]

Given these drawbacks of FM in practice, recent efforts focused on training schemes beyond FM, including noise-contrastive estimation[25,30] and flow-matching.[26] Another alternative to FM is relative entropy (RE) minimization,[5] which has frequently been used to optimize classical CG models,[31–34] but has not yet been applied to CG NN potentials. The main conceptual difference between RE and FM lies in the availability of molecular states during training: While FM trains exclusively on states provided by the AT model, RE additionally samples states from the CG model.[5,35] Sampling the CG model at each update step is computationally expensive, but it gives direct access to the CG distribution during training. Thus, deviations from the AT distribution can be accessed and subsequently corrected via gradient descent optimization—subject to the functional form of the CG model and the statistical sampling error. Thus, in the context of CG NN potentials, RE counters suboptimal global FESs and sensitivity to prior potentials.

In this work, we demonstrate the effectiveness of optimizing CG NN potentials via RE minimization. To this end, we train the CG DimeNet++[13,14] graph NN potential for the benchmark problems of liquid water and alanine dipeptide. For liquid water, both FM and RE yield highly accurate CG potentials, but RE allows larger time steps in subsequent CG MD simulations, without compromising on accuracy. For alanine dipeptide, the RE method results in a more accurate FES and is more robust to the choice of prior potential compared to FM. Finally, we showcase that pre-training via FM allows us to reduce the computational cost of RE training. Hence, the exploitation of training targets beyond FM is a promising path toward next generation CG NN potentials.

II. METHODS

We reiterate the fundamentals of FM[4,36–38] and RE minimization[5,35,39–42] theory, based on which we discuss specific properties of both methods in the context of CG NN potentials. The starting point for CG modeling is the selection of a mapping function $\mathbf{M}$,

$$\mathbf{R} = \mathbf{M}(\mathbf{r}), \tag{1}$$

that maps AT coordinates $\mathbf{r} \in \mathbb{R}^{3n}$ onto a lower-dimensional set of CG coordinates $\mathbf{R} \in \mathbb{R}^{3N}$, with $N < n$. In the following, we assume canonical (NVT) ensembles in equilibrium and $\mathbf{M}$ to be a linear function, even though generalizations to non-equilibrium systems[43] and non-linear mappings[44] exist.

The CG model is consistent with the underlying AT model, if the configurational equilibrium distribution of the CG model $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ equals $p^{\mathrm{AT}}(\mathbf{R})$; the configurational equilibrium distribution of the AT model $p^{\mathrm{AT}}(\mathbf{r})$, when mapped to CG coordinates[4] is

$$p^{\mathrm{AT}}(\mathbf{R}) = \langle \delta[\mathbf{R} - \mathbf{M}(\mathbf{r})] \rangle_{\mathrm{AT}}, \tag{2}$$

where $\langle \cdots \rangle_{\mathrm{AT}}$ indicates an AT ensemble average. $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ depends on the model parameters $\boldsymbol{\theta}$ via the CG potential $U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$. The CG model is consistent with the AT model if the CG potential equals the many-body potential of mean force (PMF)[4,5]

$$U^{\mathrm{PMF}}(\mathbf{R}) = -\frac{1}{\beta} \ln p^{\mathrm{AT}}(\mathbf{R}) + C, \tag{3}$$

where $\beta = 1/(k_{\mathrm{B}} T)$, with Boltzmann's constant k_{B} and temperature T. C is an arbitrary constant that we omit in the following. To approximate the PMF, the most popular methods are the FM[4,36,37] and the RE minimization[5] methods.

A. Force matching

FM—also known as multiscale coarse-graining[4,36,37]—aims to match the CG forces $-\nabla_{\mathbf{R}} U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ to the instantaneous forces acting on CG particles $\mathbf{F}^{AT}$, computed from the AT system. Thus, FM minimizes the mean squared error (MSE) loss function

$$\chi^2(U_{\boldsymbol{\theta}}^{\mathrm{CG}}) = \left\langle \|\mathbf{F}^{\mathrm{AT}} + \nabla_{\mathbf{R}} U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{M}(\mathbf{r}))\|^2 \right\rangle_{\mathrm{AT}}, \tag{4}$$

where $\| \cdots \|$ is the Frobenius norm. In practice, the AT ensemble average is approximated by the mean over a reference dataset of AT configurations, typically generated by an AT MD simulation.[37] Minimizing the loss in Eq. (4) represents a standard supervised learning problem, which is solved by computing the gradient $\nabla_{\boldsymbol{\theta}} \chi^2(U_{\boldsymbol{\theta}}^{\mathrm{CG}})$ via automatic differentiation for a mini-batch of AT configurations and updating $\boldsymbol{\theta}$ via a stochastic optimizer.[21]

To connect $U_{\boldsymbol{\theta}}^{\mathrm{CG}}$ to the PMF, Eq. (4) can be reformulated[4] as

$$\begin{aligned} \chi^2(U_{\boldsymbol{\theta}}^{\mathrm{CG}}) = {} & \langle \|\nabla_{\mathbf{R}} U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{M}(\mathbf{r})) - \nabla_{\mathbf{R}} U^{\mathrm{PMF}}(\mathbf{M}(\mathbf{r}))\|^2 \rangle_{\mathrm{AT}} \\ & + \underbrace{\langle \|\mathbf{F}^{\mathrm{AT}} + \nabla_{\mathbf{R}} U^{\mathrm{PMF}}(\mathbf{M}(\mathbf{r}))\|^2 \rangle_{\mathrm{AT}}}_{\equiv \chi^2(U^{\mathrm{PMF}})}. \end{aligned} \tag{5}$$

Note that $\chi^2(U^{\mathrm{PMF}})$ depends on the CG mapping M, but cannot be optimized via $\boldsymbol{\theta}$. Thus, FM minimizes the first term, resulting in the force predictions of the CG potential to approximate the forces of the PMF. For infinite data and model capacity, $U_{\boldsymbol{\theta}}^{\mathrm{CG}}$, therefore, converges to U^{PMF} (up to an additive constant). From an ML perspective, the second term in Eq. (5) corresponds to the noise term in a regression problem.[21] Physically, the noise term results from the fact that multiple AT states with different $\mathbf{F}^{AT}$ map to the same CG configuration. Hence, the noise term is irreducible and constitutes the lower bound of the loss.

B. Relative entropy minimization

The relative entropy—known as the Kullback–Leibler divergence[45] in information theory—is commonly used to quantify the distance between two distributions. In the context of CG modeling, these two distributions are $p^{\mathrm{AT}}(\mathbf{R})$ and $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$, defining the relative entropy S_{rel} as[41,42,46]

$$S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}}) = \int p^{\text{AT}}(\mathbf{R}) \ln\left(\frac{p^{\text{AT}}(\mathbf{R})}{p_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})}\right) \mathrm{d}\mathbf{R}. \tag{6}$$

Due to Gibbs's inequality, $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}}) \geq 0$. Consequently, $S_{\text{rel}}(U^{\text{PMF}}) = 0$ is the global minimum, reached if $p^{\text{AT}}(\mathbf{R}) = p_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})$ and $U_{\boldsymbol{\theta}}^{\text{CG}} = U^{\text{PMF}}$.[46] Thus, minimization of $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}})$ provides a means to approximate U^{PMF}.

Inserting the configurational probabilities of the canonical ensemble into Eq. (6) yields[5,35,42]

$$S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}}) = \beta\langle U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{M}(\mathbf{r})) - U^{\text{AT}}(\mathbf{r})\rangle_{\text{AT}} - \beta(A_{\boldsymbol{\theta}}^{\text{CG}} - A^{\text{AT}}) + S_{\text{map}}, \tag{7}$$

where A is the Helmholtz free energy, and S_{map} depends on the mapping function M, but is independent of $\boldsymbol{\theta}$.[35,42] For classical CG potentials, $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}})$ is typically minimized via the Newton–Raphson scheme.[5,32,35,39] In this work, we follow standard deep learning practice and optimize the NN potential via a first order optimizer. This approach avoids the high memory cost of computing the Hessian of the NN parameter set. While computing $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}})$ is non-trivial due to the difference in the Helmholtz free energies $A_{\boldsymbol{\theta}}^{\text{CG}} - A^{\text{AT}}$ [Eq. (7)], minimizing $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}})$ via first order optimizers only requires the computation of its gradient[35]

$$\nabla_{\boldsymbol{\theta}} S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}}) = \beta\langle \nabla_{\boldsymbol{\theta}} U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{M}(\mathbf{r}))\rangle_{\text{AT}} - \beta\langle \nabla_{\boldsymbol{\theta}} U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})\rangle_{\text{CG}}. \tag{8}$$

In practice, the first term in Eq. (8) is approximated by an average over the AT reference dataset. The second term is computationally more expensive, as the distribution corresponding to the CG potential needs to be sampled on-the-fly, typically via a CG MD simulation. The gradient $\nabla_{\boldsymbol{\theta}} U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})$ can be computed conveniently via automatic differentiation.

C. Linking force matching and relative entropy minimization

A large body of literature has studied the relationship between FM and RE,[35,42,44,46] which we reiterate in parts in the following. Defining the quantity[45]

$$\Phi_{\boldsymbol{\theta}}(\mathbf{R}) = \ln\left(\frac{p^{\text{AT}}(\mathbf{R})}{p_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})}\right) \tag{9}$$

allows reformulating the optimization objectives of RE [Eq. (6)] and FM [Eq. (5)][46] to

$$\begin{aligned} S_{\text{rel}} &= \int p^{\text{AT}}(\mathbf{R})\Phi_{\boldsymbol{\theta}}(\mathbf{R})\mathrm{d}\mathbf{R}, \\ \chi^2(U_{\boldsymbol{\theta}}^{\text{CG}}) &= \frac{(k_{\text{B}}T)^2}{3n}\int p^{\text{AT}}(\mathbf{R})\|\nabla_{\mathbf{R}}\Phi_{\boldsymbol{\theta}}(\mathbf{R})\|^2\mathrm{d}\mathbf{R} + \chi^2(U^{\text{PMF}}). \end{aligned} \tag{10}$$

Hence, both FM and RE minimize a functional of $\Phi_{\boldsymbol{\theta}}(\mathbf{R})$. Differences in the learned CG potential result from minimizing an average of $\Phi_{\boldsymbol{\theta}}(\mathbf{R})$ in RE, compared to an average of $\|\nabla_{\mathbf{R}}\Phi_{\boldsymbol{\theta}}(\mathbf{R})\|^2$ in FM.

Thus far, the comparison of FM and RE has generally focused on the case of finite model capacity and infinite AT data: FM reaches the minimum of $\chi^2(U_{\boldsymbol{\theta}}^{\text{CG}})$ [Eq. (5)] if $U_{\boldsymbol{\theta}}^{\text{CG}}$ is the projection of U^{PMF} onto the function space spanned by the CG potential basis set.[37,38] However, the resulting $U_{\boldsymbol{\theta}}^{\text{CG}}$ is not guaranteed to reproduce any AT correlation function mapped to CG coordinates.[46] In contrast, the CG potential that minimizes $S_{\text{rel}}(U_{\boldsymbol{\theta}}^{\text{CG}})$ is guaranteed to reproduce all mapped AT structural correlation functions that are conjugate to basis functions of the CG potential.[35] For example, if the CG potential includes a flexible parameterization of pairwise interactions, the radial distribution function (RDF) of the mapped AT system will be matched.

D. Finite data size effects

In the following, we compare FM and RE in the context of NN potentials, where we expect a reduced impact of the finite functional basis set, but a larger contribution from finite data size effects. We assume the common case of a dataset generated by an equilibrium AT MD simulation. Consequently, the dataset primarily contains states in potential energy minima, but rarely high energy states.[29] This gives rise to two issues in training CG NN potentials via FM: the difficulty of obtaining a globally accurate FES[24,26] and the reliance on prior potentials[21,23] (discussed in Sec. II E).

Inaccurate FESs can be caused by sensitivity of the learned potential to errors in sparsely sampled transition regions, as recently hypothesized.[26] We illustrate this idea through a thought experiment, where a system is coarse-grained to a 1D CG coordinate X (Fig. 1). We consider a specific CG potential $U_{\hat{\boldsymbol{\theta}}}^{\text{CG}}$ that differs from U^{PMF} within the transition region $\mathcal{T}$. Outside $\mathcal{T}$, $U_{\hat{\boldsymbol{\theta}}}^{\text{CG}}$ is only shifted with respect to U^{PMF}. If we assume that the AT dataset does not contain any states within $\mathcal{T}$, the validation FM loss of $U_{\hat{\boldsymbol{\theta}}}^{\text{CG}}$ is identical to the validation FM loss of U^{PMF}, given that the forces outside $\mathcal{T}$ are identical. However, the probabilities of samples generated by

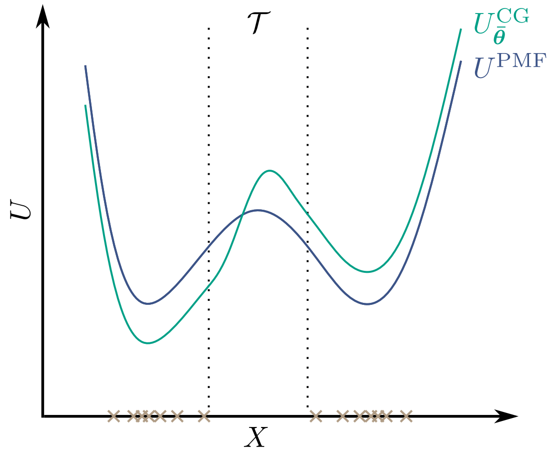


FIG. 1. Coarse-graining thought experiment. If the atomistic dataset (brown crosses) contains no states within the transition region $\mathcal{T}$, a candidate potential $U_{\hat{\boldsymbol{\theta}}}^{\text{CG}}$, whose shape only differs from the potential of mean force U^{PMF} within $\mathcal{T}$, yields the same force matching validation loss as U^{PMF}, despite resulting in a different coarse-grained distribution $p_{\hat{\boldsymbol{\theta}}}^{\text{CG}}(\mathbf{R}) \neq p^{\text{AT}}(\mathbf{R})$.

both potentials differ, e.g., $U_{\bar{\boldsymbol{\theta}}}^{\mathrm{CG}}$ preferentially samples the left minimum. FM needs to infer the free energy difference between minima by integrating the mean-force along the transition path, which is unavailable in this thought experiment. In real-world applications, this mean-force integral is determined by few and noisy[21] transition states in the AT dataset, which explains the reported difficulty in reproducing the correct relative sampling probabilities of different metastable states.[24,26] Since the transition states are comparatively rare in the training data, they only have a small impact on the FM validation loss.[29] Therefore, the FM validation loss is not a useful metric to assess the global quality of the FES.[24]

In contrast, the incorrect CG distribution $p_{\bar{\boldsymbol{\theta}}}^{\mathrm{CG}}(\mathbf{R})$ generated by $U_{\bar{\boldsymbol{\theta}}}^{\mathrm{CG}}$ results in a large S_{rel} [Eq. (6)]. Thus, RE minimization will adjust the potential, such that both metastable configurations are sampled equally, matching U^{PMF}, where AT data are available. This is consistent with the interpretation that optimizing $S_{\mathrm{rel}}(U_{\boldsymbol{\theta}}^{\mathrm{CG}})$ minimizes the difference between the potential energy surfaces of the AT and CG models,[40] i.e.,

$$S_{\mathrm{rel}}(U_{\boldsymbol{\theta}}^{\mathrm{CG}}) = \ln \left\langle e^{\Delta_{\boldsymbol{\theta}}(\mathbf{r}) - \langle \Delta_{\boldsymbol{\theta}}(\mathbf{r}) \rangle_{\mathrm{AT}}} \right\rangle_{\mathrm{AT}},$$

with

$$\Delta_{\boldsymbol{\theta}}(\mathbf{r}) \equiv \beta[U^{\mathrm{AT}}(\mathbf{r}) - U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{M}(\mathbf{r}))], \tag{11}$$

where a constant offset between the potential energy surfaces is captured by $\langle \Delta_{\boldsymbol{\theta}}(\mathbf{r}) \rangle_{\mathrm{AT}}$. In sum, RE is better suited to reproduce the global FES, especially if the phase-space is resolved inhomogeneously by the training data, making RE minimization more data efficient.[26]

E. Prior potentials

Classical CG potentials typically use physics-based functional forms,[3,7] which enforce qualitatively correct behavior, irrespective of the specific parameter values at hand; for example, Lennard-Jones interactions encode the Pauli exclusion principle at short distances and van der Waals forces at longer distances. To encode physically meaningful behavior in a similar way, the flexible functional form of NN potentials can be combined with a physics-informed prior potential $U^{\mathrm{prior}}(\mathbf{R})$:[21,23,24,26,27]

$$U_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R}) = U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{R}) + U^{\mathrm{prior}}(\mathbf{R}). \tag{12}$$

In this formulation, training the NN potential $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{R})$ can be interpreted as Δ-learning[47–49] with respect to $U^{\mathrm{prior}}(\mathbf{R})$.[21]

Note that the role of $U^{\mathrm{prior}}(\mathbf{R})$ differs significantly for FM compared to RE minimization: Since the dataset is obtained via physically sound principles in an AT MD simulation, it does not contain any unphysical configurations, such as overlapping particles. In such unphysical regions of phase-space, the CG NN potential, therefore, operates in the extrapolation regime, and can easily predict short-range attraction instead of physically sound repulsion. For FM, a well-chosen $U^{\mathrm{prior}}(\mathbf{R})$, therefore, enforces qualitative correct predictions outside the training data, to drive the CG MD simulation back into the AT data distribution where $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{R})$ is accurate. Hence, FM requires careful selection of $U^{\mathrm{prior}}(\mathbf{R})$, given that weak choices can lead to unphysical CG MD simulation results.[50]

In contrast, a strong deviation of $p_{\boldsymbol{\theta}}^{\mathrm{CG}}(\mathbf{R})$ from $p^{\mathrm{AT}}(\mathbf{R})$ caused by an unphysical trajectory leads to a large $S_{\mathrm{rel}}(U_{\boldsymbol{\theta}}^{\mathrm{CG}})$, which can be corrected by the optimizer during training. Rather than stabilizing the CG MD simulation, the role of $U^{\mathrm{prior}}(\mathbf{R})$ in RE minimization is to speed up training convergence. Without a prior, physical principles need to be learned from the AT reference data, which significantly increases the number of update steps until convergence.[27]

F. Finite time step effects

So far, we have implicitly assumed ideal sampling of the Boltzmann distribution corresponding to a specific potential, i.e., assuming an infinitesimal MD simulation time step Δt. However, in practice, the AT distribution $p_{\Delta t_{\mathrm{AT}}}^{AT}(\mathbf{r})$ results from the time step-dependent shadow Hamiltonian[51] of the reference AT simulation[26] (with the shadow temperature[52] representing the conserved quantity in the NVT ensemble). Thus, RE learns a CG potential whose shadow Hamiltonian yields a CG distribution $p_{\boldsymbol{\theta},\Delta t_{\mathrm{CG}}}^{\mathrm{CG}}(\mathbf{R})$ that approximates $p_{\Delta t_{\mathrm{AT}}}^{AT}(\mathbf{R})$. Consequently, assuming infinite data and model capacity, the optimal RE potential differs from the true PMF as a function of Δt_{AT} and Δt_{CG}. On the other hand, FM models train on a dataset of forces computed from the true AT potential. Hence, the ideal FM potential equals the true PMF, but the resulting CG distribution $p_{\boldsymbol{\theta},\Delta t_{\mathrm{CG}}}^{\mathrm{CG}}(\mathbf{R})$ will differ from the analytic $p^{AT}(\mathbf{R})$ as a function of the production time step Δt_{CG}.

G. Neural network potential

We use our previously published implementation[27] of the graph NN DimeNet++[13,14] as $U_{\boldsymbol{\theta}}^{\mathrm{NN}}$, with graph cut-off radius $r_{\mathrm{cut}} = 0.5$ nm. We set all hyperparameters to the default values of the original implementation,[14] except for embedding sizes, which we reduce by a factor of 4. With the default of four interaction blocks, DimeNet++ captures up to eight-body correlations,[28] as angles are a direct input quantity that already captures three-body properties. This high-body interaction capacity promises highly accurate approximations to the PMF.

III. RESULTS

A. Liquid water

We choose the classical benchmark problem of CG liquid water to test FM and RE in a setting where AT reference data are abundantly available. We generate a 10 ns AT trajectory of the TIP4P/2005[53] water model at a temperature $T_{\mathrm{ref}} = 298$ K, which we subsample, to retain a state every 1 ps. Each state consists of a cubical simulation box of length $l = 3.129$ nm, containing 1000 water molecules. The first 8 ns are used for training, the subsequent 0.8 ns for validation, and the last 1.2 ns are retained as a test set.

We select a CG mapping, where each water molecule is mapped to a CG particle located at its center of mass. To the DimeNet++ $U_{\boldsymbol{\theta}}^{\mathrm{NN}}(\mathbf{R})$, we add the pairwise repulsive part of the Lennard-Jones potential as prior

$$U^{\mathrm{prior}}(\mathbf{R}) = \sum_{i=1}^{N_{\mathrm{pair}}} \varepsilon \left(\frac{\sigma}{d_i} \right)^{12}, \tag{13}$$

where we sum over all N_{pair} pairs with distance $d_i < r_{\text{cut}}$, [Eq. (12)]. Analogous to our previous work,[27] we choose $\varepsilon = 1$ kJ/mol and $\sigma = 0.3165$ nm, which is the length scale of the SPC[54] water model.

The FM model is trained for 100 epochs, with a batch size of ten states (for loss curves, see supplementary material Fig. 1). We select the model with the smallest validation loss, which is computed after each training epoch. The validation set is exclusively used in FM for this purpose, giving FM a small advantage over RE in terms of data usage. We train the RE model for 300 update steps. To sample $p_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})$ during training, we run 70 ps simulations, including 5 ps of equilibration, with a time step of 2 fs. Each trajectory starts from the last state of the previous trajectory, to reduce equilibration time. For more technical details, see the supplementary material, Method 1.

We evaluate the quality of force predictions of both trained models, based on the held-out test dataset (Fig. 2). Compared to AT NN potentials, predicted forces exhibit larger errors, due to the noise resulting from the non-injective CG mapping [Eq. (5)]. The FM model yields slightly better force predictions ($R^2 = 0.695$) than the RE model ($R^2 = 0.670$). Apart from possible overfitting of the FM model onto forces, this result likely stems from the finite time step effects discussed above:[26] The reference forces are computed from the true AT potential, which is consistent with the optimization objective of the FM method. In contrast, the RE potential needs to account for the shadow Hamiltonians of the AT and CG simulations.

To test the capabilities of the models in an application context, we perform CG MD simulations with a time step of $\Delta t_{\text{CG}} = 2$ fs and a trajectory length of 1.1 ns, where the first 0.1 ns are discarded for equilibration. We compute the RDF and angular distribution function (ADF),[55] as well as the equilateral triplet correlation function (TCF),[56–58] to assess different structural correlations in the generated CG distribution. The RE model matches the AT references to the line thickness (Fig. 3). This is in line with theoretical expectations that RE reproduces all structural correlation functions for which conjugate terms in the CG potential exist.[35] FM is in better agreement with the AT reference pressure $p_{\text{ref}} = -6.2$ MPa ($p_{\text{FM}} = 212.2$ MPa, $p_{\text{RE}} = 311.0$ MPa) at the expense of slightly larger errors in the structural correlation functions. These results are insensitive to the specific choice of prior potential, which we tested by selecting a softer prior $\left(\frac{\sigma}{d}\right)^6$ [Eq. (13); supplementary material Fig. 2].

Additionally, we compare the DimeNet++ model to a classical two-body cubic spline model. The spline model is computationally inexpensive compared to the DimeNet++ model (184.5 vs 7.3 ps/min), at the expense of reduced accuracy: In contrast to the FM spline model, the RE spline model matches the target RDF (supplementary material Fig. 3),reproducing literature results.[39,59] However, as expected, both models fail to match three-body correlations. Adding three-body terms improves those classical models,[60] but the accuracy still remains limited, compared to the DimeNet ++ model. Overall, these results suggest that the difference between models obtained via FM and RE tends to increase for decreasing adequacy of the functional basis set for a given system—in line with previous computational studies.[61]

Given that computational speed-up is the principal motivation for CG modeling, we evaluate trained FM and RE models for the real-world case of larger production CG MD time step sizes Δt_{CG} (Fig. 4). The resulting MSE values of FM models increase significantly for larger Δt_{CG}, which we attribute to increased time integration errors. Presumably, the impact of the CG shadow Hamiltonian becomes noticeable for FM CG NN potentials, as errors from an incomplete basis set and finite data size effects are small in this problem. By contrast, the MSE values of RE models increase only slightly for larger Δt_{CG}, when using the same time step during training. Given that RE optimizes the empirical CG distribution $p_{\boldsymbol{\theta},\Delta t_{\text{CG}}}^{\text{CG}}(\mathbf{R})$, we presume that the RE potential learns to correct for the time step-dependent terms in the CG shadow Hamiltonian. To test this hypothesis, we apply the RE models trained with 10 fs in CG MD simulations with $\Delta t_{\text{CG}} = 2$ fs. In line with our hypothesis,

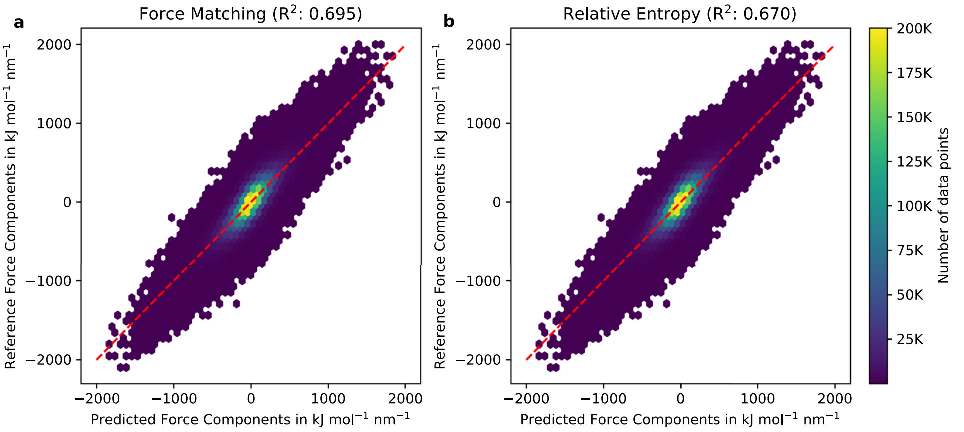


FIG. 2. Liquid water force predictions on test data. Each data point corresponds to a predicted force component for a coarse-grained particle in the test dataset, compared to its atomistic reference for models trained via (a) force matching and (b) relative entropy minimization.

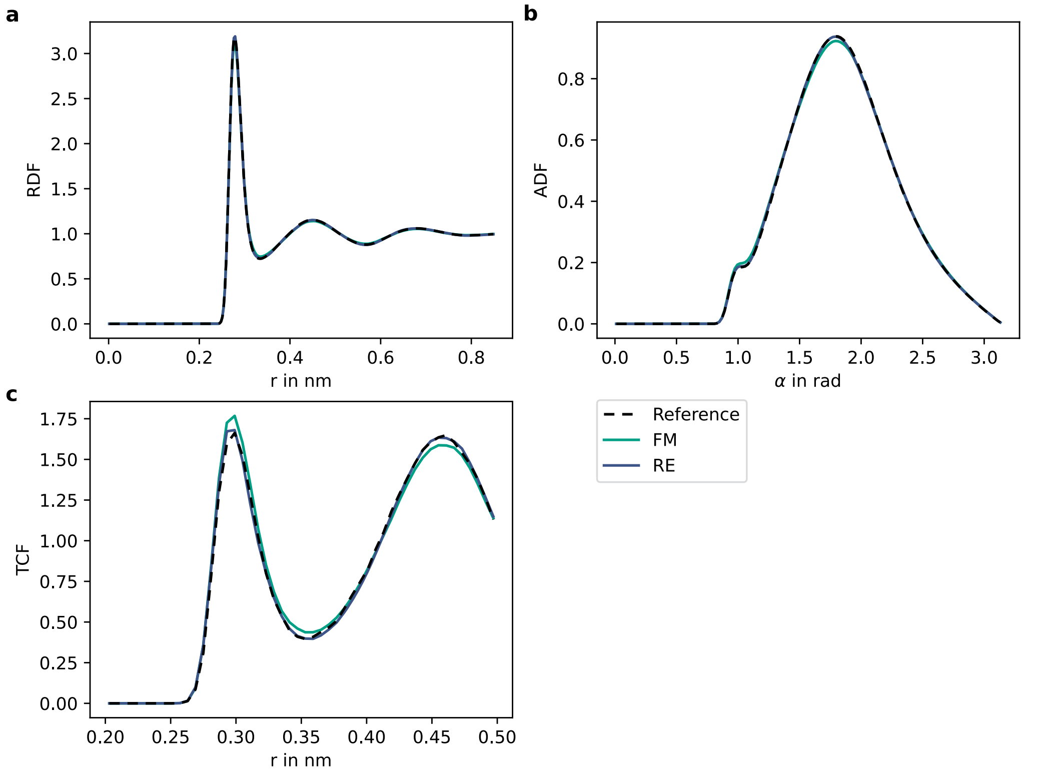


FIG. 3. Structural correlation functions in liquid water. Resulting (a) radial distribution function (RDF) and (b) angular distribution function (ADF),[55] as well as (c) equilateral triplet correlation function (TCF)[57,58] of models trained via force matching (FM) and relative entropy (RE) minimization, compared to the atomistic reference.

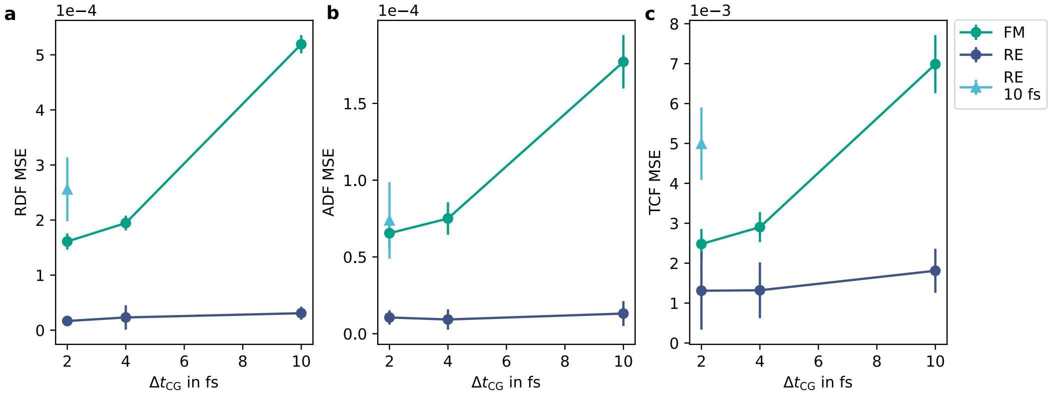


FIG. 4. Time step variation. Mean squared error (MSE) of resulting (a) radial (RDF) and (b) angular distribution functions (ADF), as well as (c) equilateral distribution function (TCF) for different time step sizes Δt_{CG} in subsequent molecular dynamics simulations. The plotted mean and standard deviation values are computed from five models, with different random seeds for neural network parameter initialization and velocity distribution of the initial simulation state. For force matching (FM), the same five models are used for different simulation time steps. For relative entropy (RE) minimization, the models are retrained such that the training time step matches Δt_{CG}. An exception is models trained with 10 fs, which are additionally run with Δt_{CG} = 2 fs (light blue).

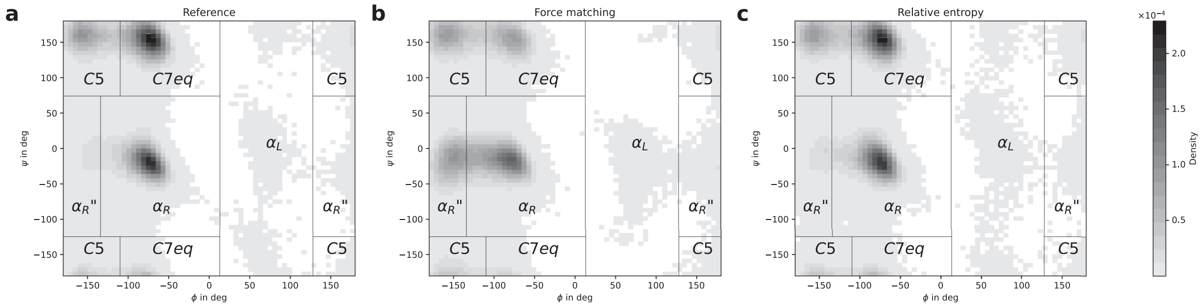


FIG. 5. Ramachandran diagrams. Resulting density histograms of the dihedral angles ϕ and ψ from (a) the AT reference simulation and from the CG models trained via (b) force matching and (c) relative entropy minimization.

this combination yields larger MSE values than RE models with consistent time steps (Fig. 4): If the RE model learns to correct large time step-dependent terms in the shadow Hamiltonian, this biases CG simulations that exhibit only small time integration errors. Hence, RE minimization provides a means to mitigate the accuracy degradation of larger production time steps Δt_{CG}.

B. Alanine dipeptide

Alanine dipeptide[62,63] is a standard problem to benchmark CG methods in reconstructing an FES with multiple metastable states. We generate a 100 ns AT reference trajectory at $T_{\mathrm{ref}} = 300$ K, from which a state is retained every 0.2 ps, resulting in $5 \cdot 10^5$ data points. The training dataset consists of the first 80 ns, the FM validation set of the subsequent 8 ns, and the final 12 ns forms the test set. We select a CG mapping that retains all ten heavy atoms of alanine dipeptide, but drops hydrogen atoms and water molecules. The CG particles representing CH_3, CH, and C are encoded as different particle types. Following the Δ-learning ansatz in Eq. (12), we select a prior potential

$$U^{\mathrm{prior}}(\mathbf{R}) = \sum_{i=1}^{N_{\mathrm{bonds}}} U^{\mathrm{harmonic}}(b_i) + \sum_{j=1}^{N_{\mathrm{angles}}} U^{\mathrm{harmonic}}(\alpha_j) + \sum_{k=1}^{N_{\mathrm{dihedrals}}} U^{\mathrm{proper}}(\omega_k),$$

$$U^{\mathrm{harmonic}}(x_i) = \frac{k_{\mathrm{B}}T}{2\mathrm{Var}[x_i]}(x_i - \langle x_i \rangle_{\mathrm{AT}})^2,$$

with

$$\mathrm{Var}[x_i] = \langle (x_i - \langle x_i \rangle_{\mathrm{AT}})^2 \rangle_{\mathrm{AT}}, \qquad U^{\mathrm{proper}}(\omega_i) = k_\omega(1 + \cos n\omega_i - \omega_0), \tag{14}$$

where we sum over all N_{bonds} harmonic bonds with bond lengths b_i, all N_{angles} harmonic angles with triplet angles α_j and all $N_{\mathrm{dihedrals}}$ proper dihedral angles ω_k. The dihedral force constant k_ω, the multiplicity n, and the phase constant ω_0 are taken from the AMBER03[64] force field.

We train the FM model for 100 epochs, with a batch size of 500 states, and select the model that yields the smallest validation loss

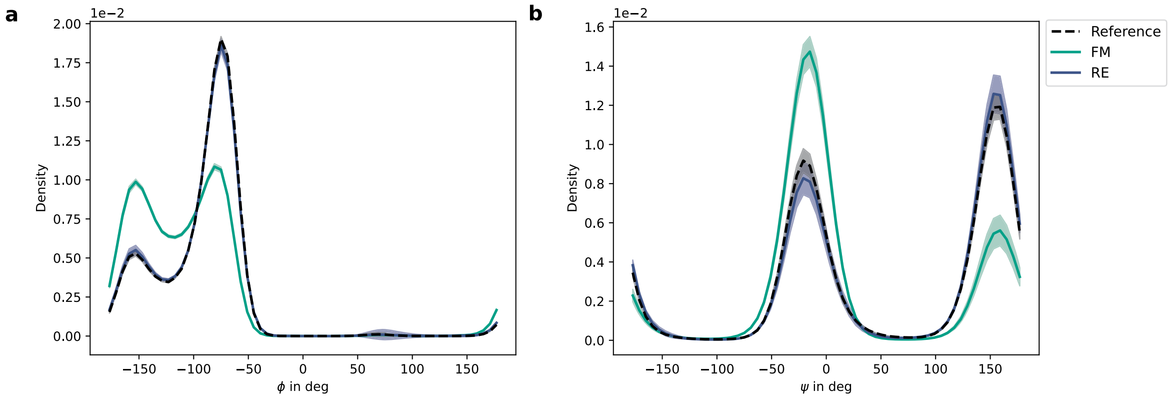


FIG. 6. Dihedral angle density. Distribution of dihedral angles (a) ϕ and (b) ψ, as predicted from the CG models trained via force matching (FM) and relative entropy (RE) minimization, compared to the atomistic reference. The mean and standard deviation (shaded area) are computed from 50 trajectories of 100 ns lengths.

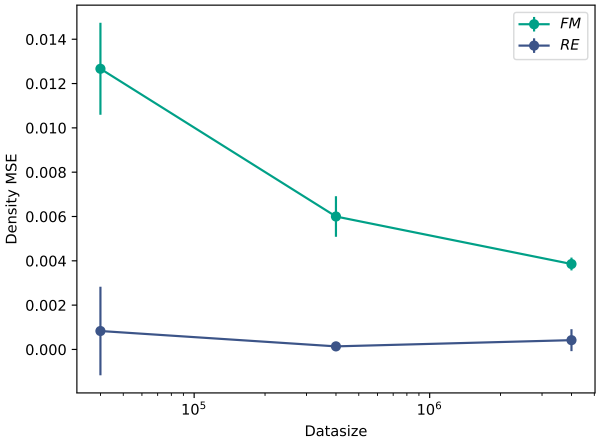


FIG. 7. Training data variation. Mean squared error (MSE) of the $\phi - \psi$ dihedral density histograms of force matching (FM) and relative entropy (RE) minimization models for varying training data sizes. The mean and standard deviation values are computed from 50 trajectories of 100 ns lengths.

(for loss curves, see supplementary material Fig. 4). For RE training, we sample the CG distribution through 50 vectorized CG MD simulations starting from different initial states, which improves the computational efficiency of GPUs. Each parallel simulation generates a 1 ns trajectory, of which 5 ps are discarded for equilibration. The simulations restart from the last obtained state of the previous trajectory, and we train the RE model for 300 updates. Additional technical details are available in the supplementary material, Method 2.

First, we compare the force prediction quality of the FM and RE models on the test dataset (scatter plots in supplementary material Fig. 5). Analogous to the liquid water example, the FM model yields better force predictions ($R^2 = 0.780$) than the RE model ($R^2 = 0.717$). However, in the case of alanine dipeptide, we are mostly interested in an accurate reproduction of the FES. Hence, we sample 100 ns CG trajectories with the trained RE and FM models, such that the number of generated states equals the AT reference data.

The resulting 2D density histograms of the dihedral angles ϕ and ψ[62,63] are shown in Fig. 5 (corresponding FESs in supplementary material Fig. 6). The FESs obtained via the DimeNet++ potential compare favorably to previously reported results with a classical, generalized Born[65] implicit solvent model of alanine dipeptide.[24] The Ramachandran diagram of the RE model matches the AT reference well, but the FM model oversamples the α_R'' configuration. The 1D projections of the dihedral density are shown in Fig. 6. Despite the smaller test set error with respect to forces, the FM model fails to accurately reproduce the ratio of metastable states. This result supports the notion that the FM validation error is not a useful metric to judge the global quality of the learned FES.[26]

Assuming that insufficient resolution of transition regions caused the suboptimal FES of the FM model, increasing the amount of training data should improve the FES: We generate a 1 μs AT trajectory, increasing the amount of training data by a factor of 10. Using this dataset, the error of the FM model decreases as expected, but is still inferior to the RE model (Fig. 7). Note that numerical errors of the CG simulations also contribute to the FM MSE, which cannot be reduced by enlarging the training dataset. To test the data requirement limits of RE, we reduce the 100 ns training dataset by a factor of 10. In this case, RE still results in a smaller error than FM with the largest dataset, despite using only 1% of the training data, which highlights the data efficiency of RE for reproducing the FES.

By combining FM and RE, we aim to exploit their respective strengths—data efficiency of RE and computational inexpensiveness of FM. Based on the 100 ns dataset, we optimize the FM potential by additional RE updates. As depicted in Fig. 8, few RE updates are sufficient to significantly improve the FES. With 30 updates, the obtained dihedral densities are comparable to the randomly initialized 300 update RE model (Fig. 6), and significantly better than a

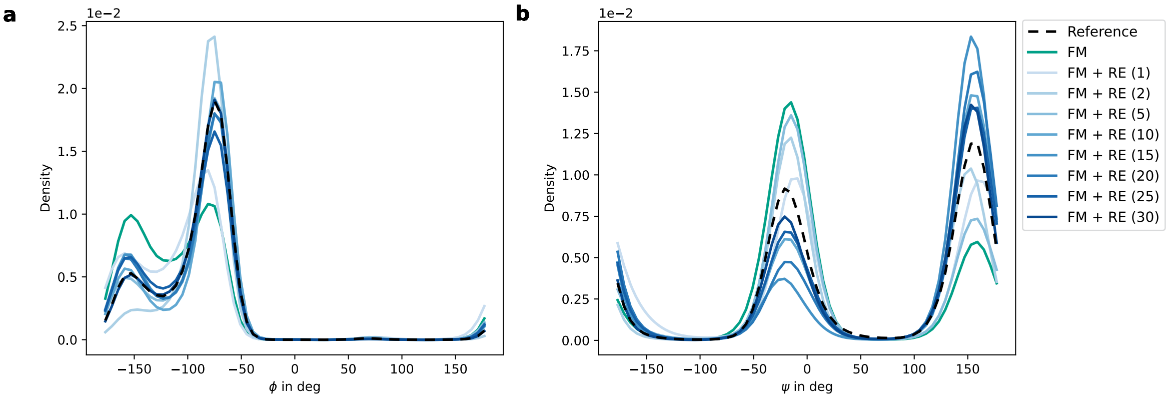


FIG. 8. Convergence of relative entropy correction steps. (a) ϕ and (b) ψ dihedral angle distributions corresponding to potentials obtained by different numbers of relative entropy (RE) updates, when being initialized to the force matching (FM) potential. The lines represent the mean computed from 50 trajectories of 100 ns lengths.

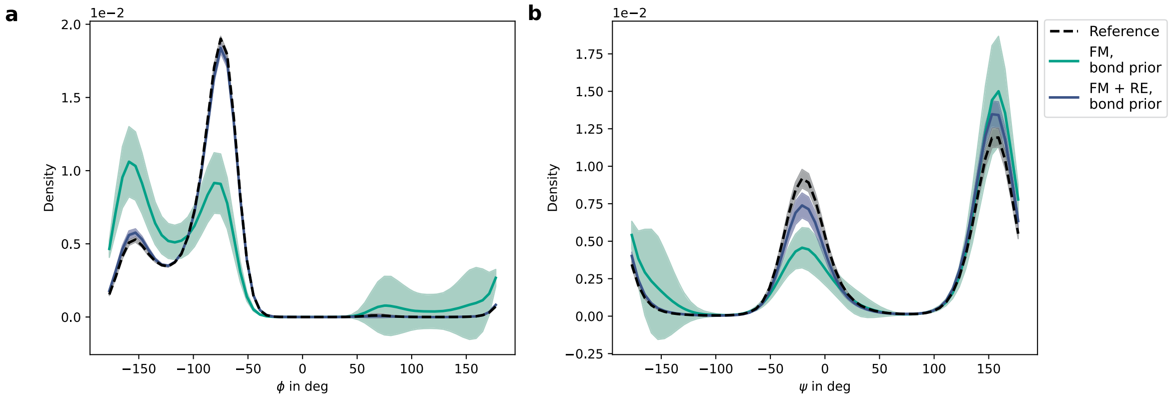


FIG. 9. Prior potential ablation. Distribution of dihedral angles (a) ϕ and (b) ψ resulting from the force matching (FM) model, when only considering bonds in the prior potential. These are compared to the atomistic reference and to a model that optimizes the FM potential via 300 relative entropy (RE) update steps. The mean and standard deviation (shaded area) are computed from 50 trajectories of 100 ns lengths.

randomly initialized 30 update RE model (supplementary material Fig. 7). Consequently, initializing RE minimization with the FM model allows us to reduce the number of necessary RE updates significantly.

Finally, we test the robustness of both methods with respect to prior potentials, by considering only harmonic bonds in $U^{\text{prior}}(\mathbf{R})$ [Eq. (14)]. In this case, the FM potential significantly oversamples α_L configurations (Fig. 9, supplementary material Fig. 8), despite superior validation force predictions ($R^2 = 0.762$), compared to the reference RE model above. Conversely, when optimizing the resulting FM model via 300 additional RE update steps, the dihedral density is in close agreement with the AT reference. Hence, RE minimization also helps correcting weak choices of prior potentials.

IV. DISCUSSION AND CONCLUSION

In this work, we have demonstrated the effectiveness of training CG NN potentials via the RE minimization scheme: For water, the difference between FM and RE minimization is significantly reduced when training CG NN potentials, compared to classical two-body CG potentials.[60] This is expected, as the learned potential converges toward the PMF for increasing model capacity with both methods, given sufficient amount of data.[4,35,46] For alanine dipeptide, RE results in a more accurate FES than FM. The discrepancy in the FES increases for decreasing quality of the prior, to the point that the FM CG NN potential is no longer competitive with classical CG models. Sampling the CG model during training probes its robustness with respect to data generated on-the-fly. As a consequence, the RE training scheme can recognize and correct undesired model properties, which reduces the sensitivity in the prior potential.

For liquid water, RE allows larger time steps in subsequent CG MD simulations, without compromising on accuracy. We presume that RE is able to learn to correct the time integration error of the underlying CG MD simulation, by training directly on the sampled CG distribution. Given that computational speed-up is the primary objective of CG modeling, a larger simulation time step is as important as an accurate approximation of the PMF. Consequently, CG NN potentials trained via RE may reach larger time scales in production CG simulations, with less impact from time integration errors.

The advantages of RE minimization come at the cost of increased computational effort during training, which can, however, be reduced: As demonstrated in this work, pre-training via FM is a computationally efficient way to reduce the number of necessary RE updates through a better parameter initialization. Furthermore, histogram reweighting,[31,42,66–69] frequently used in RE minimization,[5,32,35] is also applicable to NN potentials.[27] Reweighting allows previously generated trajectories to be reused, increasing the number of gradient descent steps per trajectory computation. Additionally, it seems reasonable to increase the trajectory length during the course of the optimization. In the beginning of training, where the model error significantly exceeds the statistical error of trajectories, short trajectories can save computational time. Toward the end of training, longer trajectories with reduced statistical noise allow fine-tuning of the model. This scheme matches well with reweighting: Initial trajectories cannot be used for reweighting, irrespective of their length, due to large changes in the potential, while expensive trajectories toward the end of training may be reused for multiple updates. Moreover, we argue that in the realistic scenario of an expensive AT model with a, by design, orders of magnitude cheaper CG model, the computational bottleneck is AT training data generation, rather than CG model optimization. Finally, NN potential architectures optimized for computational efficiency, such as the ultra-fast force fields,[70] are well-suited for CG applications. These architectures allow to capture many-body features of the PMF, while reducing the computational overhead of the more expensive DimeNet++[14] model used in this work. Evaluating the computational cost–accuracy trade-off between different computationally efficient CG NN potentials and classical CG models is an interesting avenue for future research. Additionally, the merits of CG NN potentials should be examined for more complex systems than those considered in this work.

The presented results can also be interpreted in terms of data efficiency. In CG applications, an accurate representation of the FES is usually of higher interest than accurate force predictions in energy minima. RE is well suited to reproduce the FES, in practice, by directly minimizing the difference between the potential energy surfaces of the AT and CG models [Eq. (11)]. By contrast, FM requires a sufficient resolution of transition areas, to learn a globally accurate FES.[26] This requires a large amount of AT training data, which is expensive to obtain. Long AT MD trajectories do not seem efficient in this regard, due to the repetitive sampling of energy minima and sparse sampling of high energy states.[29] Accordingly, enhanced sampling schemes, such as metadynamics[29,71,72] or normal mode sampling,[73,74] may improve data efficiency of FM, by spreading the sampling more evenly across the phase space.

In line with the literature on simulation-based optimization schemes for classical CG models,[2,6] our results suggest that including MD simulations in the training process can be considered as a means to improve the reliability and accuracy of NN potentials, allowing to address recent concerns about their stability.[75,76] Active learning NN potentials,[77,78] recognized as a major building block in achieving stable and transferable models,[22,79,80] can be similarly interpreted as an incorporation of MD simulations into the FM training scheme: Performing MD simulations and screening visited molecular states for high-uncertainty configurations allows us to augment the dataset iteratively in phase space regions that are reachable by the NN potential, but still sparsely represented in the dataset. Alternatively, MD simulations can also be inserted directly into the training pipeline[27,81–84] using auto-differentiable MD codes.[82,84] We expect that the benefits of using ML in simulations and, inversely, simulations for ML training will continue to drive the ongoing synthesis of ML and physical simulations in molecular modeling and beyond.

SUPPLEMENTARY MATERIAL

See the supplementary material for additional computational details, force matching loss curves, liquid-water prior variation and two-body cubic spline models, alanine dipeptide force predictions, free energy surface, and additional dihedral density histograms.

ACKNOWLEDGMENTS

The authors thank Dominik Blechschmidt for the contributions to initial feasibility studies.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Stephan Thaler: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Methodology (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Maximilian Stupp**: Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Julija Zavadlav**: Conceptualization (equal); Data curation (equal); Project administration (lead); Supervision (lead); Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request. The code to train the presented DimeNet++ models via FM and RE is open-sourced at https://github.com/tummfm/relative-entropy.

REFERENCES

[1] J. D. McCoy and J. G. Curro, "Mapping of explicit atom onto united atom potentials," Macromolecules **31**, 9362–9368 (1998).

[2] D. Reith, M. Pütz, and F. Müller-Plathe, "Deriving effective mesoscale potentials from atomistic simulations," J. Comput. Chem. **24**, 1624–1636 (2003).

[3] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. de Vries, "The MARTINI force field: Coarse grained model for biomolecular simulations," J. Phys. Chem. B **111**, 7812–7824 (2007).

[4] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models," J. Chem. Phys. **128**, 244114 (2008).

[5] M. S. Shell, "The relative entropy is fundamental to multiscale and inverse thermodynamic problems," J. Chem. Phys. **129**, 144108 (2008).

[6] W. G. Noid, "Perspective: Coarse-grained models for biomolecular systems," J. Chem. Phys. **139**, 090901 (2013).

[7] H. I. Ingólfsson, C. A. Lopez, J. J. Uusitalo, D. H. de Jong, S. M. Gopal, X. Periole, and S. J. Marrink, "The power of coarse graining in biomolecular simulations," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **4**, 225–248 (2014).

[8] N. Singh and W. Li, "Recent advances in coarse-grained models for biomolecules and their applications," Int. J. Mol. Sci. **20**, 3774 (2019).

[9] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," Phys. Rev. Lett. **98**, 146401 (2007).

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

[11] K. T. Schütt, P. J. Kindermans, H. E. Sauceda, S. Chmiela, A. Tkatchenko, and K. R. Müller, "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2017), pp. 992–1002.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning* (PMLR, 2017), pp. 1263–1272.

[13] J. Klicpera, J. Groß, and S. Günnemann, "Directional message passing for molecular graphs," in 8th International Conference on Learning Representations, ICLR, 2020.

[14] J. Klicpera, S. Giri, J. T. Margraf, and S. Günnemann, "Fast and uncertainty-aware directional message passing for non-equilibrium molecules," in Machine Learning for Molecules Workshop at NeurIPS (2020).

[15] Z. Qiao, M. Welborn, A. Anandkumar, F. R. Manby, and T. F. Miller, "OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features," J. Chem. Phys. **153**, 124111 (2020).

[16] A. C. P. Jain, D. Marchand, A. Glensk, M. Ceriotti, and W. A. Curtin, "Machine learning for metallurgy III: A neural network potential for Al-Mg-Si," Phys. Rev. Mater. **5**, 053805 (2021).

[17] T. W. Ko, J. A. Finkler, S. Goedecker, and J. Behler, "A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer," Nat. Commun. **12**, 398 (2021).

[18] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, and B. Kozinsky, "E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials," Nat. Commun. **13**, 2453 (2022).

[19] I. Batatia, D. P. Kovács, G. N. Simm, C. Ortner, and G. Csányi, "MACE: Higher order equivariant message passing neural networks for fast and accurate force fields," arXiv:2206.07697 (2022).

[20] L. Zhang, J. Han, H. Wang, R. Car, and W. E, "DeePCG: Constructing coarse-grained models via deep neural networks," J. Chem. Phys. **149**, 034101 (2018).

[21] J. Wang, S. Olsson, C. Wehmeyer, A. Pérez, N. E. Charron, G. De Fabritiis, F. Noé, and C. Clementi, "Machine learning of coarse-grained molecular dynamics force fields," ACS Cent. Sci. **5**, 755–767 (2019).

[22] T. D. Loeffler, T. K. Patra, H. Chan, and S. K. R. S. Sankaranarayanan, "Active learning a coarse-grained neural network model for bulk water from sparse training data," Mol. Syst. Des. Eng. **5**, 902–910 (2020).

[23] B. E. Husic, N. E. Charron, D. Lemm, J. Wang, A. Pérez, M. Majewski, A. Krämer, Y. Chen, S. Olsson, G. de Fabritiis, F. Noé, and C. Clementi, "Coarse graining molecular dynamics with graph neural networks," J. Chem. Phys. **153**, 194101 (2020).

[24] Y. Chen, A. Krämer, N. E. Charron, B. E. Husic, C. Clementi, and F. Noé, "Machine learning implicit solvation for molecular dynamics," J. Chem. Phys. **155**, 084101 (2021).

[25] X. Ding and B. Zhang, "Contrastive learning of coarse-grained force fields," J. Chem. Theory Comput. **18**(10), 6334–6344 (2022).

[26] J. Köhler, Y. Chen, A. Krämer, C. Clementi, and F. Noé, "Force-matching coarse-graining without forces," arXiv:2203.11167 (2022).

[27] S. Thaler and J. Zavadlav, "Learning neural network potentials from experimental data via differentiable trajectory reweighting," Nat. Commun. **12**, 6884 (2021).

[28] I. Batatia, S. Batzner, D. P. Kovács, A. Musaelian, G. N. Simm, R. Drautz, C. Ortner, B. Kozinsky, and G. Csányi, "The design space of E(3)-equivariant atom-centered interatomic potentials," arXiv:2205.06643 (2022).

[29] J. E. Herr, K. Yao, R. McIntyre, D. W. Toth, and J. Parkhill, "Metadynamics for training neural network model chemistries: A competitive assessment," J. Chem. Phys. **148**, 241710 (2018).

[30] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, JMLR Workshop and Conference Proceedings (PMLR, 2010), pp. 297–304.

[31] S. P. Carmichael and M. S. Shell, "A new multiscale algorithm and its application to coarse-grained peptide models for self-assembly," J. Phys. Chem. B **116**, 8383–8393 (2012).

[32] S. Bottaro, K. Lindorff-Larsen, and R. B. Best, "Variational optimization of an all-atom implicit solvent force field to match explicit solvent simulation data," J. Chem. Theory Comput. **9**, 5641–5652 (2013).

[33] S. Y. Mashayak, M. N. Jochum, K. Koschke, N. R. Aluru, V. Rühle, and C. Junghans, "Relative entropy and optimization-driven coarse-graining methods in VOTCA," PLoS One **10**, e0131754 (2015).

[34] T. Sanyal and M. S. Shell, "Transferable coarse-grained models of liquid–liquid equilibrium using local density potentials optimized with the relative entropy," J. Phys. Chem. B **122**, 5678–5693 (2018).

[35] A. Chaimovich and M. S. Shell, "Coarse-graining errors and numerical optimization using a relative entropy framework," J. Chem. Phys. **134**, 094112 (2011).

[36] S. Izvekov and G. A. Voth, "Multiscale coarse graining of liquid-state systems," J. Chem. Phys. **123**, 134105 (2005).

[37] W. G. Noid, P. Liu, Y. Wang, J.-W. Chu, G. S. Ayton, S. Izvekov, H. C. Andersen, and G. A. Voth, "The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models," J. Chem. Phys. **128**, 244115 (2008).

[38] J. W. Mullinax and W. G. Noid, "Generalized Yvon-Born-Green theory for molecular systems," Phys. Rev. Lett. **103**, 198104 (2009).

[39] A. Chaimovich and M. S. Shell, "Anomalous waterlike behavior in spherically-symmetric water models optimized with the relative entropy," Phys. Chem. Chem. Phys. **11**, 1901–1915 (2009).

[40] A. Chaimovich and M. S. Shell, "Relative entropy as a universal metric for multiscale errors," Phys. Rev. E **81**, 060104 (2010).

[41] P. Español and I. Zúñiga, "Obtaining fully dynamic coarse-grained models from MD," Phys. Chem. Chem. Phys. **13**, 10538–10545 (2011).

[42] M. S. Shell, "Coarse-graining with the relative entropy," Adv. Chem. Phys. **161**, 395–441 (2016).

[43] V. Harmandaris, E. Kalligiannaki, M. Katsoulakis, and P. Plecháč, "Path-space variational inference for non-equilibrium coarse-grained systems," J. Comput. Phys. **314**, 355–383 (2016).

[44] E. Kalligiannaki, V. Harmandaris, M. A. Katsoulakis, and P. Plecháč, "The geometry of generalized force matching and related information metrics in coarse-graining of molecular systems," J. Chem. Phys. **143**, 084105 (2015).

[45] S. Kullback and R. A. Leibler, "On information and sufficiency," Ann. Math. Stat. **22**, 79–86 (1951).

[46] J. F. Rudzinski and W. G. Noid, "Coarse-graining entropy, forces, and structures," J. Chem. Phys. **135**, 214101 (2011).

[47] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. Von Lilienfeld, "Big data meets quantum chemistry approximations: The Δ-machine learning approach," J. Chem. Theory Comput. **11**, 2087–2096 (2015).

[48] L. Shen and W. Yang, "Molecular dynamics simulations with quantum mechanics/molecular mechanics and adaptive neural networks," J. Chem. Theory Comput. **14**, 1442–1455 (2018).

[49] L. Böselt, M. Thürlemann, and S. Riniker, "Machine learning in QM/MM molecular dynamics simulations of condensed-phase systems," J. Chem. Theory Comput. **17**, 2641–2658 (2021).

[50] H. Wang, C. Junghans, and K. Kremer, "Comparative atomistic and coarse-grained study of water: What do we lose by coarse-graining?," Eur. Phys. J. E **28**, 221–229 (2009).

[51] S. Toxvaerd, "Hamiltonians for discrete dynamics," Phys. Rev. E **50**, 2271 (1994).

[52] S. Toxvaerd, "Ensemble simulations with discrete classical dynamics," J. Chem. Phys. **139**, 224106 (2013).

[53] J. L. F. Abascal and C. Vega, "A general purpose model for the condensed phases of water: TIP4P/2005," J. Chem. Phys. **123**, 234505 (2005).

[54] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, and J. Hermans, "Interaction models for water in relation to protein hydration," in *Intermolecular Forces* (Springer, 1981), pp. 331–342.

[55] A. K. Soper and C. J. Benmore, "Quantum differences between heavy and light water," Phys. Rev. Lett. **101**, 065502 (2008).

[56] A. Baranyai and D. J. Evans, "Three-particle contribution to the configurational entropy of simple fluids," Phys. Rev. A **42**, 849 (1990).

[57] B. Bildstein and G. Kahl, "Triplet correlation functions for hard-spheres: Computer simulation results," J. Chem. Phys. **100**, 5882 (1994).

[58] D. Dhabal, M. Singh, K. T. Wikfeldt, and C. Chakravarty, "Triplet correlation functions in liquid water," J. Chem. Phys. **141**, 174504 (2014).

[59] V. Rühle, C. Junghans, A. Lukyanov, K. Kremer, and D. Andrienko, "Versatile object-oriented toolkit for coarse-graining applications," J. Chem. Theory Comput. **5**, 3211–3223 (2009).

[60] C. Scherer and D. Andrienko, "Understanding three-body contributions to coarse-grained force fields," Phys. Chem. Chem. Phys. **20**, 22387–22394 (2018).

[61] E. Kalligiannaki, A. Chazirakis, A. Tsourtis, M. A. Katsoulakis, P. Plecháč, and V. Harmandaris, "Parametrizing coarse grained models for molecular systems at equilibrium," Eur. Phys. J.: Spec. Top. **225**, 1347–1372 (2016).

[62] B. Montgomery Pettitt and M. Karplus, "The potential of mean force surface for the alanine dipeptide in aqueous solution: A theoretical approach," Chem. Phys. Lett. **121**, 194–201 (1985).

[63] D. J. Tobias and C. L. Brooks III, "Conformational equilibrium in the alanine dipeptide in the gas phase and aqueous solution: A comparison of theoretical results," J. Phys. Chem. **96**, 3864–3870 (1992).

[64] Y. Duan, C. Wu, S. Chowdhury, M. C. Lee, G. Xiong, W. Zhang, R. Yang, P. Cieplak, R. Luo, T. Lee, J. Caldwell, J. Wang, and P. Kollman, "A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations," J. Comput. Chem. **24**, 1999–2012 (2003).

[65] J. Chen, C. L. Brooks III, and J. Khandogin, "Recent advances in implicit solvent-based methods for biomolecular simulations," Curr. Opin. Struct. Biol. **18**, 140–148 (2008).

[66]R. W. Zwanzig, "High-temperature equation of state by a perturbation method. I. Nonpolar gases," J. Chem. Phys. **22**, 1420–1426 (1954).

[67]C. Chipot and A. Pohorille, *Free Energy Calculations* (Springer, 2007), Vol. 86.

[68]A. B. Norgaard, J. Ferkinghoff-Borg, and K. Lindorff-Larsen, "Experimental parameterization of an energy function for the simulation of unfolded proteins," Biophys. J. **94**, 182–192 (2008).

[69]D.-W. Li and R. Brüschweiler, "Iterative optimization of molecular mechanics force fields from NMR data of full-length proteins," J. Chem. Theory Comput. **7**, 1773–1782 (2011).

[70]S. R. Xie, M. Rupp, and R. G. Hennig, "Ultra-fast interpretable machine-learning potentials," arXiv:2110.00624 (2021).

[71] A. Barducci, M. Bonomi, and M. Parrinello, "Metadynamics," Wiley Interdiscip. Rev.: Comput. Mol. Sci. **1**, 826–843 (2011).

[72]L. Bonati and M. Parrinello, "Silicon liquid structure and crystal nucleation from *ab initio* deep metadynamics," Phys. Rev. Lett. **121**, 265701 (2018).

[73]W. Schneider and W. Thiel, "Anharmonic force fields from analytic second derivatives: Method and application to methyl bromide," Chem. Phys. Lett. **157**, 367–373 (1989).

[74]M. Rupp, R. Ramakrishnan, and O. A. Von Lilienfeld, "Machine learning for quantum mechanical properties of atoms in molecules," J. Phys. Chem. Lett. **6**, 3309–3313 (2015).

[75]S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, and J. T. Margraf, "How robust are modern graph neural network potentials in long and hot molecular dynamics simulations?," Mach. Learn.: Sci. Technol. **3**, 045010 (2022).

[76]X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, "Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations," arXiv:2210.07237 (2022).

[77]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**, 241733 (2018).

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

[79]R. Jinnouchi, K. Miwa, F. Karsai, G. Kresse, and R. Asahi, "On-the-fly active learning of interatomic potentials for large-scale atomistic simulations," J. Phys. Chem. Lett. **11**, 6946–6955 (2020).

[80]J. S. Smith, B. Nebgen, N. Mathew, J. Chen, N. Lubbers, L. Burakovsky, S. Tretiak, H. A. Nam, T. Germann, S. Fensin *et al.*, "Automated discovery of a robust interatomic potential for aluminum," Nat. Commun. **12**, 1257 (2021).

[81] J. Ingraham, A. Riesselman, C. Sander, and D. Marks, "Learning protein structure with a differentiable simulator," in 7th International Conference on Learning Representations, ICLR, 2019.

[82]S. S. Schoenholz and E. D. Cubuk, "JAX, M.D.: A framework for differentiable physics," in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2020), Vol. 33.

[83]C. P. Goodrich, E. M. King, S. S. Schoenholz, E. D. Cubuk, and M. P. Brenner, "Designing self-assembling kinetics with differentiable statistical physics models," Proc. Natl. Acad. Sci. U. S. A. **118**, e2024083118 (2021).

[84]S. Doerr, M. Majewski, A. Pérez, A. Krämer, C. Clementi, F. Noe, T. Giorgino, and G. De Fabritiis, "TorchMD: A deep learning framework for molecular simulations," J. Chem. Theory Comput. **17**, 2355–2363 (2021).

3.2. Uncertainty Quantification for Neural Network Potentials

This section presents research articles advancing UQ for MD potentials via SG-MCMC methods. This includes a comparative study of SG-MCMC for NN potentials (sec. 3.2.1) and a modular library that simplifies application and development of novel SG-MCMC schemes (sec. 3.2.2).

3.2.1. Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls

Thaler, S., Doehner, G. & Zavadlav, J. Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls. *J. Chem. Theory Comput.* **19**, 4520–4532 (2023), DOI: 10.1021/acs.jctc.2c01267.

Summary

The flexible functional form of NN potentials allows to train highly accurate potential energy models. At the same time, this flexibility casts NN potentials unreliable outside the training data distribution, where potential energy predictions can be highly inaccurate. For practical application of these models in MD simulations, it is imperative to quantify whether the obtained results are trustworthy. The Bayesian posterior predictive distribution allows to quantify the aleatoric and epistemic uncertainty of predictions. Its evaluation involves sampling from the posterior distribution, which is infeasible for real-world NN potentials using classical MCMC schemes such as HMC due to the required full-dataset likelihood evaluations.

This paper investigates the quality of scalable UQ schemes for NN potentials, specifically SG-MCMC and the non-Bayesian Deep Ensemble method. As a representative of SG-MCMC schemes, we selected the preconditioned Stochastic Gradient Langevin Dynamics (pSGLD) method with RMSprop. Additionally, this work compares the impact of using a single Markov chain (S-pSGLD) versus multiple Markov chains (M-pSGLD).

In a toy example of a LJ potential with training data points within the potential energy well, both M-pSGLD and the Deep Enesmble method yield high quality approximations of the posterior predictive distribution as predicted by the No-U-Turn Sampler (NUTS), a gold-standard HMC scheme. In contrast, S-pSGLD significantly underestimates the epistemic uncertainty outside the training data distribution because it only samples a single posterior mode. Interestingly, also the NUTS with a single Markov chain fails to sample multiple posterior modes. This indicates that sampling multiple posterior modes with a singe Markov chain is difficult in the molecular energy-matching task, even for sophisticated posterior exploration schemes.

Next, we used the scalable UQ schemes to train the DimeNet++ model for CG liquid water via FM. In this case, M-pSGLD and the Deep Ensemble method quantify the epistemic uncertainty accurately. The true values are contained within the MD observable credible intervals, both within the training data distribution as well as under temperature shift. However, the latter is only possible if the NN potential can account for the state-point dependency of the PMF. Otherwise, the error from the state-point change cannot be captured due to model misspecification, resulting in unquantified systematic uncertainty.

Lastly, the problem of CG alanine dipeptide demonstrates the value of UQ in the case of insufficient training data. All considered UQ schemes predict very large epistemic uncertainty when models sample unphysical phase-space regions. This signals to practitioners that the obtained results are not yet trustworthy and that more data needs to be gathered in order to sufficiently constrain the NN potential. Hence, scalable UQ also proves valuable during the NN training process.

In all of the considered test cases, S-pSGLD yielded overconfident uncertainty estimates. This highlights the importance of sampling multiple posterior modes, which can be achieved most reliably by de-correlation of the NN potentials via different random initializations. Additionally, cold posteriors proved beneficial to Bayesian training of NN potentials, reducing the required amount of training data significantly. However, uncovering the source of this cold posterior effect requires further research. Furthermore, the results show that the Deep Ensemble method quantifies epistemic uncertainty as well as M-pSGLD, while requiring less training and hyperparameter tuning. This is in contrast to prior research that suggests that the Deep Ensemble method was prone to overconfidence and less reliable than Bayesian methods. Rather, the Deep Ensemble method can be interpreted as Bayesian model averaging, where the posterior predictive distribution is approximated via multiple approximate maximum a-posteriori points. Finally, NN potentials are well suited for UQ given that many-body interactions and state-point dependency can be included by default. Consequently, systematic uncertainties can be minimized, which allows for the prediction of accurate credible intervals. Hence, quantifying the uncertainty of MD observables is an important building block for trustworthy NN potential-based MD simulations.

CRediT author statement

Stephan Thaler: Conceptualization, Data curation, Formal analysis, Funding aquisition, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing

Gregor Döhner: Formal analysis, Investigation, Visualization, Writing – original draft

Julija Zavadlav: Conceptualization, Supervision, Writing – original draft, Writing – review & editing

Copyright notice

JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

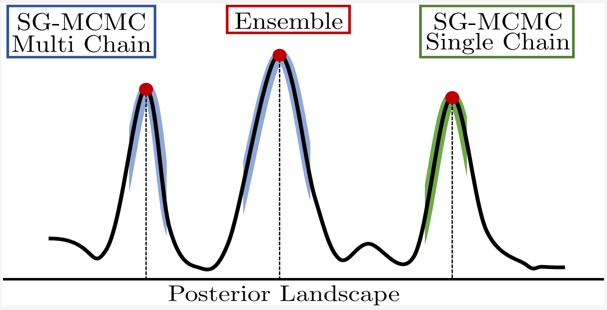
Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls

Stephan Thaler,* Gregor Doehner, and Julija Zavadlav*

Cite This: *J. Chem. Theory Comput.* 2023, 19, 4520−4532 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Neural network (NN) potentials promise highly accurate molecular dynamics (MD) simulations within the computational complexity of classical MD force fields. However, when applied outside their training domain, NN potential predictions can be inaccurate, increasing the need for Uncertainty Quantification (UQ). Bayesian modeling provides the mathematical framework for UQ, but classical Bayesian methods based on Markov chain Monte Carlo (MCMC) are computationally intractable for NN potentials. By training graph NN potentials for coarse-grained systems of liquid water and alanine dipeptide, we demonstrate here that scalable Bayesian UQ via stochastic gradient MCMC (SG-MCMC) yields reliable uncertainty estimates for MD observables. We show that cold posteriors can reduce the required training data size and that for reliable UQ, multiple Markov chains are needed. Additionally, we find that SG-MCMC and the Deep Ensemble method achieve comparable results, despite shorter training and less hyperparameter tuning of the latter. We show that both methods can capture aleatoric and epistemic uncertainty reliably, but not systematic uncertainty, which needs to be minimized by adequate modeling to obtain accurate credible intervals for MD observables. Our results represent a step toward accurate UQ that is of vital importance for trustworthy NN potential-based MD simulations required for decision-making in practice.



1. INTRODUCTION

Molecular dynamics (MD) simulations are the computational tool of choice to describe complex molecular phenomena. Their computational effort and accuracy depend on the chosen potential energy model. Neural network (NN) potentials,[1−7] which model many-body interactions,[8,9] promise MD simulations at ab initio accuracy[4,10] within the computational complexity of classical molecular mechanics force fields.

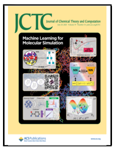
The quality of NN potentials is limited by the scarcity of suitable training data,[11] given that data generation via computational quantum mechanics simulations and/or experiments is resource intensive. Hence, potentials are commonly applied outside their training domain due to the high-dimensional chemical space. As NN potentials are data-driven black box models, predictions outside the training domain may be inaccurate or even unphysical.[12−14] This may hinder more widespread adoption of NN potentials in practical applications where less powerful but physically more constrained models are preferred.[15]

Uncertainty quantification (UQ) can provide a remedy, as it enables practitioners to quantify the trustworthiness of MD simulation predictions.[16−18] Additionally, the availability of a UQ metric enables more efficient training data generation via active learning,[14,19−23] as well as an adaptive combination of NN potentials with established classical force fields.[24] Bayesian statistics provides a mathematically rigorous approach to UQ. However, classical Bayesian inference schemes based on Markov Chain Monte Carlo (MCMC), such as Hamiltonian (or hybrid[25]) Monte Carlo (HMC),[26] require an evaluation of the likelihood over the whole data set for each parameter update. Frequent full likelihood evaluations are prohibitively expensive for computationally demanding NNs and large data sets.[27] Stochastic gradient MCMC (SG-MCMC) schemes[28−32] enable scalable Bayesian UQ of NNs by computing stochastic estimates of the gradient of the likelihood on a mini-batch of data. Stochastic variational inference[33,34] represents another scalable Bayesian UQ method, while the Deep Ensemble[35,36] method is a popular non-Bayesian[15,36−38] alternative.

In the context of NN potentials, the Deep Ensemble method is in fact the most common UQ scheme,[5,24,27] but Dropout Monte Carlo[39] and last-layer Gaussian Mixture Models[40] have also been applied. In view of the poor performance of the Deep Ensemble method in an active learning context, Kahle and Zipoli[27] recently hypothesized that Bayesian approaches may

Special Issue: Machine Learning for Molecular Simulation

Received: December 12, 2022
Published: April 4, 2023



provide more reliable uncertainty estimates for NN potentials. However, a comprehensive assessment of Bayesian UQ in the context of NN potentials is still outstanding.

In this work, we investigate scalable Bayesian UQ of MD observables for simulations utilizing NN molecular models. To this end, we first compare the UQ quality of a SG-MCMC method to the popular Deep Ensemble method and a gold-standard[15,31,32,41] HMC sampler based on a Lennard-Jones (LJ) system with a 2-body toy NN potential. We then extend the comparison by learning graph NN potentials for coarse-grained (CG) systems of water and alanine dipeptide, demonstrating the practical applicability of SG-MCMC methods to fully-Bayesian modeling of graph NN potentials. Additionally, we investigate the influence of so-called cold posteriors[38] and the number of MCMC chains on the quality of Bayesian UQ. Finally, we advocate distinguishing between different sources of uncertainty; in particular, we highlight the importance of minimizing systematic uncertainties to obtain reliable credible intervals of MD observables.

2. METHODS

In the following, we briefly summarize the employed SG-MCMC sampler as well as the Deep Ensemble method and continue with an outline of the Bayesian molecular modeling problem considered in this work.

2.1. Sources of Uncertainty. The uncertainty in physical modeling can be divided into aleatoric, epistemic and systematic uncertainty.[15] Aleatoric uncertainty refers to the inherent stochastic nature of the modeled process, which can be interpreted as randomness in the labels $\mathbf{y}$ for a given input $\mathbf{x}$.[41,42] Epistemic uncertainty refers to the uncertainty about the true hypothesis (model) within the considered hypothesis space (model family). In contrast to aleatoric uncertainty, epistemic uncertainty can be reduced by gathering more data. Finally, systematic uncertainty is caused by model misspecification, i.e., when the true data-generating process is not contained within the hypothesis space. Systematic uncertainty manifests itself in an inconsistency between the data and the hypothesis space.[42]

2.2. Bayesian Modeling. A probabilistic model $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ predicts the distribution of $\mathbf{y}$ reflecting the aleatoric uncertainty, given a training data set $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$ of size N. Bayesian UQ additionally aims to quantify the epistemic uncertainty resulting from the model fit to a finite amount of data.[15,37] Instead of selecting a single set of model parameters $\boldsymbol{\theta}$, the Bayesian approach promises more robust predictions by marginalizing over $\boldsymbol{\theta}$.[43] Hence, the goal of Bayesian UQ is to compute the posterior predictive distribution

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \tag{1}$$

where $p(\boldsymbol{\theta}|\mathcal{D})$ is the posterior distribution. The integral in eq 1 is typically approximated by the Monte Carlo method:

$$p(\mathbf{y}|\mathbf{x}, \mathcal{D}) \approx \frac{1}{N}\sum_{n=1}^{N} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_n) \tag{2}$$

where $\boldsymbol{\theta}_n$ represents the n^{th} model parameter set drawn from the posterior. Evaluating eq 2 requires sampling from the posterior distribution

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \propto \exp\left(\frac{-\mathcal{U}(\boldsymbol{\theta})}{\mathcal{T}}\right) \tag{3}$$

with likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ and prior $p(\boldsymbol{\theta})$. In analogy to statistical mechanics, the posterior can be rewritten to allow sampling from a Boltzmann-type distribution,[26] with posterior potential energy

$$\mathcal{U}(\boldsymbol{\theta}) = -\sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \tag{4}$$

and posterior temperature $\mathcal{T}$, which is introduced as an additional hyperparameter. $\mathcal{T} = 1$ corresponds to the Bayesian posterior, while $\mathcal{T} < 1$ are sharper[44] cold posteriors.[38]

The gold-standard HMC[25,26] method leverages the gradient $\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta})$ to simulate Hamiltonian dynamics to generate parameter proposals for the Metropolis Hastings[45] (MH) acceptance step, which guarantees that the equilibrium distribution of the Markov chain corresponds to $p(\boldsymbol{\theta}|\mathcal{D})$. The computation of $\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta})$[26] requires an evaluation of the (NN) model for the whole training data set $\mathcal{D}$ (eq 4), rendering HMC computationally intractable for training NN potentials.[27,46]

2.3. Stochastic Gradient MCMC. Stochastic gradient MCMC (SG-MCMC) methods[28−31] achieve enormous computational speed-ups by replacing $\nabla_{\boldsymbol{\theta}}\mathcal{U}(\boldsymbol{\theta})$ by a stochastic estimate over a mini-batch of data

$$\nabla_{\boldsymbol{\theta}}\tilde{\mathcal{U}}(\boldsymbol{\theta}) = -\frac{N}{B}\sum_{i=1}^{B} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}_i|\mathbf{x}_i, \boldsymbol{\theta}) - \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}) \tag{5}$$

where B is the mini-batch size.

The simplest SG-MCMC scheme is the Stochastic Gradient Langevin Dynamics (SGLD) method,[28] which updates parameters according to

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \frac{\lambda_k}{2}\nabla_{\boldsymbol{\theta}}\tilde{\mathcal{U}}(\boldsymbol{\theta}) + \boldsymbol{\eta}_t; \qquad \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \lambda_k\mathbf{I}) \tag{6}$$

The learning rate λ_k is decreasing as a function of update step k, and $\boldsymbol{\eta}_t$ is a learning rate-dependent Gaussian noise vector. λ_k typically follows a polynomial schedule:[28,47]

$$\lambda_k = a(k+1)^{-\gamma} \tag{7}$$

where a is the initial learning rate and γ is the decay rate. To reduce the bias due to the omitted MH acceptance step, it is necessary to sample only below a certain learning rate threshold, given that the acceptance probability asymptotically converges to 1 for $\lambda \to 0$. Hence, SGLD smoothly transitions from stochastic posterior maximization to asymptotically unbiased sampling from $p(\boldsymbol{\theta}|\mathcal{D})$ during training.[28,47] In our experiments, we employ a preconditioned version of SGLD (pSGLD),[30] which uses a RMSProp[48] preconditioner to simplify sampling the highly nonconvex posterior of NNs,[30,49] as implemented in jax-sgmc.[50]

2.4. Deep Ensemble Method. Analogous to the Monte Carlo approximation in Bayesian UQ (eq 2), the Deep Ensemble method[35,36] estimates epistemic uncertainty from the statistics of predictions from an ensemble of NNs. However, instead of sampling models from the posterior, the ensemble of NNs is generated by minimizing a loss function via stochastic gradient descent, starting from different random

https://doi.org/10.1021/acs.jctc.2c01267
J. Chem. Theory Comput. 2023, 19, 4520−4532

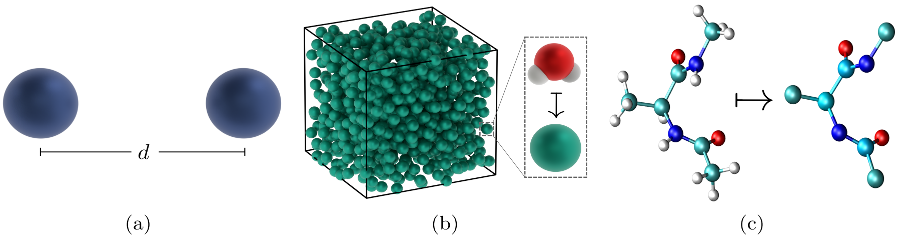


Figure 1. Visualization of numerical test case systems: (*a*) Lennard-Jones potential, (*b*) coarse-grained liquid water (Adapted with permission from ref 13. Copyright The Authors 2021.), (*c*) coarse-grained alanine dipeptide.

NN weight initializations. If desired, aleatoric uncertainty can be quantified by additionally predicting standard deviations and minimizing a negative log-likelihood loss.[36] While most authors consider the Deep Ensemble method non-Bayesian,[15,36−38] Wilson and Izmailov[43] compellingly argue that it can also be interpreted as Bayesian model averaging.

2.5. Neural Network Posterior Landscape. The posterior distribution of NNs is high-dimensional, nonconvex and multimodal.[43,44,49,51] The NNs of the Deep Ensemble typically converge to different posterior modes due to the strong decorrelation effect of different random weight initializations.[43,51] Hence, the Deep Ensemble method performs a Bayesian model average of NNs corresponding to different approximate maximum a-posteriori (MAP) points on the NN posterior (assuming regularization terms that mimic the prior).[43] The Deep Ensemble method therefore exploits the NN posterior multimodality to estimate the uncertainty. By contrast, most scalable Bayesian methods, including single-chain SG-MCMC and stochastic variational inference, have been found to typically approximate a single posterior mode only.[37,41,43] However, sampling multiple posterior modes is essential for robust UQ.[43]

2.6. Multichain SG-MCMC. Sampling the posterior with multiple randomly initialized SG-MCMC chains appears to be a promising approach. It combines Bayesian posterior exploration along the Markov chain with strong decorrelation from different random initializations, the benefits of which have been shown to be complementary.[44,51] Multichain SG-MCMC can be interpreted as a custom trade-off between the number of approximated posterior modes and the amount of Bayesian exploration per mode, with single-mode Bayesian methods and the Deep Ensemble method representing the two extreme cases.

The computational training cost of the Deep Ensemble method and SG-MCMC can be estimated as $C * n_{\text{steps}} * n_{\text{chains}}$, where n_{chains} is the number of ensemble members (chains), n_{steps} is the number of parameter updates per ensemble member (chain), and C is the cost per update. Training the different ensemble members (chains) can be parallelized trivially, if desired.

2.7. Probabilistic Molecular Modeling. *2.7.1. Maximum Likelihood Molecular Modeling.* The most common training scheme for atomistic (AT) NN potentials is to match the potential energy (possibly also forces and virial) of an underlying high-fidelity model, usually a computational quantum mechanics scheme,[52] given a training data set of N_{box} molecular states.[8] This can be achieved by minimizing the mean-squared error loss function

$$L(\boldsymbol{\theta}) = \frac{1}{N_{\text{box}}} \sum_{i=1}^{N_{\text{box}}} [U_i - U_{i,\boldsymbol{\theta}}]^2 \tag{8}$$

where U_i and $U_{i,\boldsymbol{\theta}}$ are the target and predicted potential energies of molecular state i, respectively. The predicted potential energy $U_{\boldsymbol{\theta}}(\mathbf{r})$ depends on atom positions $\mathbf{r}$.

Similarly, for CG systems, the NN potential can be trained via force matching (FM),[53−56] i.e., matching the instantaneous force components F_j acting on each CG particle as computed from the AT force field:

$$L(\boldsymbol{\theta}) = \frac{1}{N_{\text{F}}} \sum_{j=1}^{N_{\text{F}}} [F_j - F_{j,\boldsymbol{\theta}}]^2 \tag{9}$$

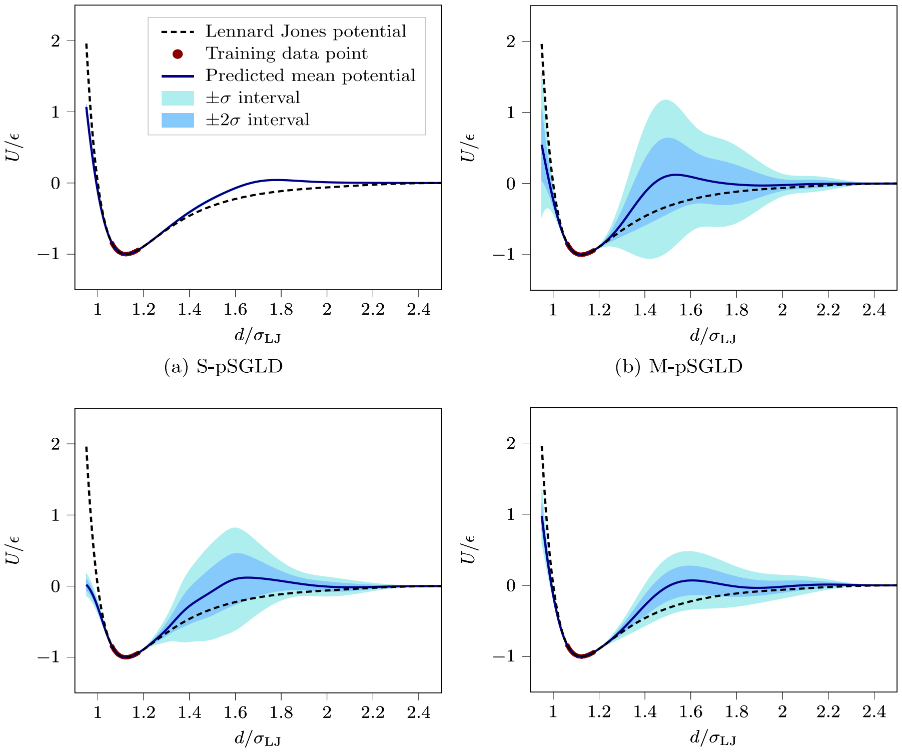
where N_{F} is 3 times the number of CG particles in the training data set. The predicted force components are computed from the CG NN potential $\mathbf{F}_{\boldsymbol{\theta}} = -\nabla_{\mathbf{R}} U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})$, which acts on CG coordinates $\mathbf{R} = \mathbf{M}(\mathbf{r})$. $\mathbf{M}$ is a linear function that maps from AT to CG coordinates. For infinite data and model capacity, $U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R})$ converges to the potential of mean force (PMF). Given that multiple AT configurations map to the same CG configuration, there exists a lower bound of the loss in eq 9, which corresponds to the loss of the PMF.[12,54]

2.7.2. Bayesian Molecular Modeling. Assuming independent Gaussian homoscedastic aleatoric uncertainty with variance σ_{H}^2, the probabilistic model of the energy matching task is $p(U|\mathbf{r}, \boldsymbol{\theta}) \sim \mathcal{N}(U_{\boldsymbol{\theta}}(\mathbf{r}), \sigma_{\text{H}}^2)$. In this case, the likelihood can be written similar to eq 8 as[27]

$$\begin{aligned} p(\mathcal{D}|\boldsymbol{\theta}) &= \prod_{i=1}^{N_{\text{box}}} \frac{1}{\sqrt{2\pi\sigma_{\text{H}}^2}} \exp\left(-\frac{[U_i - U_{i,\boldsymbol{\theta}}]^2}{2\sigma_{\text{H}}^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma_{\text{H}}^2}}\right)^{N_{\text{box}}} \exp\left(-\frac{\sum_{i=1}^{N_{\text{box}}} [U_i - U_{i,\boldsymbol{\theta}}]^2}{2\sigma_{\text{H}}^2}\right) \end{aligned} \tag{10}$$

The probabilistic model and the likelihood for the FM task follow analogously. The loss minima in eqs 8 and 9 correspond to the likelihood maxima in eq 10.

The aleatoric uncertainty is uncertainty inherent to the data. When learning atomistic models from simulation data, the aleatoric uncertainty stems from the data-generating simulation and is typically small. For CG systems, the noninjective CG mapping contributes significantly to the aleatoric uncertainty. The variance of the aleatoric uncertainty σ_{H}^2 is typically unknown a priori, and we model it as a learnable parameter. Thus, the prior $p(\boldsymbol{\theta}) = p(\mathbf{w})p(\sigma_{\text{H}})$ is the product of a prior for the NN potential weights and biases $\mathbf{w}$ and a prior for the aleatoric uncertainty scale.



(a) S-pSGLD (b) M-pSGLD

(c) Deep Ensemble (d) NUTS

Figure 2. Distribution of NN potentials. Predicted mean potential with ±σ and ±2σ intervals of the single chain pSGLD (*a*), the multichain pSGLD (*b*), the Deep Ensemble method (*c*) and the multichain No-U-Turn Sampler (NUTS, *d*), compared to the Lennard-Jones reference.

2.8. Neural Network Potential. We choose a graph NN potential, which is a state-of-the-art NN architecture that learns to extract predictive features from the molecular configuration in an end-to-end manner instead of relying on hand-crafted descriptors.[2,3] Specifically, we select our previously published implementation[13] of the DimeNet++[4,5] potential. We set all hyperparameters to their default values, including the graph cutoff radius of $r_{\text{cut}} = 0.5$ nm, except for embedding sizes, which we reduce by factor 4 for computational speed-up. We select a Gaussian prior over all learnable weights and biases $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, 10^2\mathbf{I})$, except for the radial Bessel frequencies,[4] which we model by a uniform distribution.

Given that DimeNet++ trained via FM tends to yield unstable MD simulations,[57] we augment the NN potential with a fixed, physics-informed "prior" potential $U^{\text{prior}}(\mathbf{R})$:[12,58,59]

$$U_{\boldsymbol{\theta}}^{\text{CG}}(\mathbf{R}) = U_{\boldsymbol{\theta}}^{\text{NN}}(\mathbf{R}) + U^{\text{prior}}(\mathbf{R}) \tag{11}$$

Note that $U^{\text{prior}}(\mathbf{R})$ is not a prior in the Bayesian sense but rather a physics-informed initialization that enforces physically reasonable predictions in phase-space regions unconstrained by the training data[12,13,60] (see Supplementary Methods 1 for more details).

3. RESULTS

We present three examples (Figure 1) to distinguish between different sources of uncertainty: A LJ toy example features epistemic uncertainty only, while the following two CG systems include a significant amount of aleatoric uncertainty. We additionally show the effects of systematic uncertainty for liquid water and for alanine dipeptide.

3.1. Lennard-Jones Potential. We learn a LJ potential (σ_{LJ}, ϵ) with a pairwise additive NN potential to benchmark the scalable UQ methods against a HMC scheme. As the reference method, we select the No-U-Turn Sampler (NUTS),[61] which selects the number of HMC integration steps on-the-fly. Additionally, a window adaption warm-up scheme[62,63] automatically selects an appropriate mass matrix and step size, such that no hyperparameter tuning is required for the NUTS. The NN potential predicts the pairwise potential energy $U(d)$ given pairwise particle distance d and consists of a single hidden layer with 64 neurons and swish activation, where d is represented by six radial Bessel functions[4] with a cutoff $r_{\text{cut}} = 2.5\sigma_{\text{LJ}}$. We choose a Gaussian prior for weights and biases $p(\mathbf{w}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and an exponential prior with scale 1 for the aleatoric uncertainty $p(\sigma_{\text{H}})$. For all Bayesian methods, we sample 100 models from the Bayesian posterior ($\mathcal{T} = 1$), evenly distributed over all considered Markov chains. The training data consists of 100 randomly drawn training data points from the well of the LJ potential $d/\sigma_{\text{LJ}} \in [1.0615, 1.1800]$ (Supplementary Figure 1). Additional technical details are provided in Supplementary Methods 2.

In the following, we benchmark pSGLD with a single chain (S-pSGLD), pSGLD with 10 chains (M-pSGLD), and a Deep Ensemble consisting of 10 NNs against a 10 chain NUTS. The obtained mean potentials and corresponding standard deviation intervals are visualized in Figure 2. The mean potentials of all considered methods fit the LJ potential very

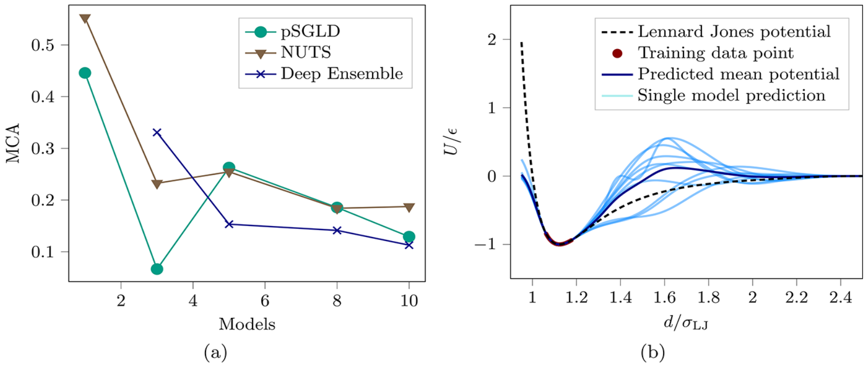


Figure 3. Posterior mode analysis. (*a*) Miscalibration area (MCA) of the No-U-Turn Sampler (NUTS), the pSGLD, and the Deep Ensemble methods as a function of the number of randomly initialized models. The MCA includes both within and out-of-distribution test data. (*b*) All predicted potentials of the Deep Ensemble method with resulting mean compared to the Lennard-Jones reference.

well where training data were generated: On held-out data within the training interval, we obtained low root-mean-squared errors (RMSE/ϵ) of 0.011 (S-pSGLD), 0.014 (NUTS), 0.023 (M-pSGLD), and 0.025 (Deep Ensemble).

Bayesian methods estimate the scale of the aleatoric uncertainty to $\sigma_{\mathrm{H}}/\epsilon \approx 10^{-3}$. Such low estimates are expected given that the aleatoric uncertainty of the LJ data set is zero and the NN potential has sufficient capacity to interpolate the training data. Hence, we neglect the contribution of the aleatoric uncertainty in the following uncertainty predictions and only show the epistemic uncertainty. The S-pSGLD method samples a single posterior mode only, yielding highly overconfident potential energy predictions outside the training interval. By contrast, the other methods using multiple randomly initialized models sample multiple posterior modes, such that they can capture a significant amount of epistemic uncertainty. Accordingly, the obtained credible intervals include the reference potential across a broad range of distances. Both M-pSGLD and the Deep Ensemble method provide good approximations to the NUTS reference distribution. However, compared to the NUTS reference, the Deep Ensemble method underestimates uncertainty at short distances and M-pSGLD overestimates uncertainty at medium distances.

All UQ methods sampling multiple modes exhibit a similar shape of the predicted epistemic uncertainty. Local uncertainty maxima are located between $1.4\sigma_{\mathrm{LJ}} < d < 1.8\sigma_{\mathrm{LJ}}$, and the uncertainty significantly increases for $d < 0.9\sigma_{\mathrm{LJ}}$. This uncertainty shape is the result of the NN potential architecture and the location of the training data set: On the one hand, the radial Bessel representation[4] of d smoothly shrinks the NN potential toward 0 at r_{cut}. On the other hand, the training data constrains the potential for $1.06\sigma_{\mathrm{LJ}} < d < 1.18\sigma_{\mathrm{LJ}}$. Hence, these results are consistent with the expectation that the epistemic uncertainty should increase with the distance from points that constrain the potential.

We further investigate the effect of the number of randomly initialized models (number of Markov chains for MCMC) on UQ quality. The miscalibration area (MCA)[64−66] quantifies the agreement between the predicted standard deviation and the true error. For all methods, the MCA shows a decreasing trend with increasing number of randomly initialized models (Figure 3*a*). This reflects the importance of sampling multiple posterior modes for robust UQ,[43] which can be achieved comparatively easily by exploiting the strong decorrelation effect of random NN initializations.[43,51] Different posterior modes represent different potentials, all of which are consistent with the training data but differ significantly where there is no data available, thus capturing the epistemic uncertainty (Figure 3*b*).

The inability to sample multiple posterior modes using a single Markov chain is not unique to pSGLD. A single chain of the NUTS also samples a single posterior mode only, and the captured epistemic uncertainty increases with additional chains (see also Supplementary Figure 3). This suggests that sampling multiple posterior modes with a single Markov chain is difficult to achieve when training NN potentials, even for sophisticated posterior exploration schemes. Finally, we note that by artificially fixing σ_{H} to a large value as in ref 27, the single chain NUTS predicts large epistemic uncertainty outside the training interval (Supplementary Figure 3). However, this comes at the cost of a larger error within the training interval (RMSE/ϵ = 0.044 for $\sigma_{\mathrm{H}}/\epsilon$ = 0.05) given that models with poorer fit also appear probable due to the allegedly large aleatoric noise in the data.

3.2. Coarse-Grained Liquid Water.

We apply pSGLD and the Deep Ensemble method to CG liquid water, a classic benchmark problem, to test their respective performance both within the training distribution as well as under distribution shift. The reference data consists of 100 cubic boxes of length l = 3.129 nm containing 1000 water molecules each, sampled every 1 ps from the TIP4P/2005[67] model at a temperature T_{ref} = 298 K, resulting in a pressure p_{ref} = −6.2 MPa. We divide the data into training, validation and test with a 80%-8%-12% split. Each water molecule is modeled by a CG particle positioned at its center of mass.

We select the repulsive part of the LJ potential as prior potential

$$U^{\mathrm{prior}}(\mathbf{R}) = \sum_{i=1}^{N_{\mathrm{pair}}} \epsilon_{\mathrm{w}} \left(\frac{\sigma_{\mathrm{w}}}{d_i} \right)^{12} \tag{12}$$

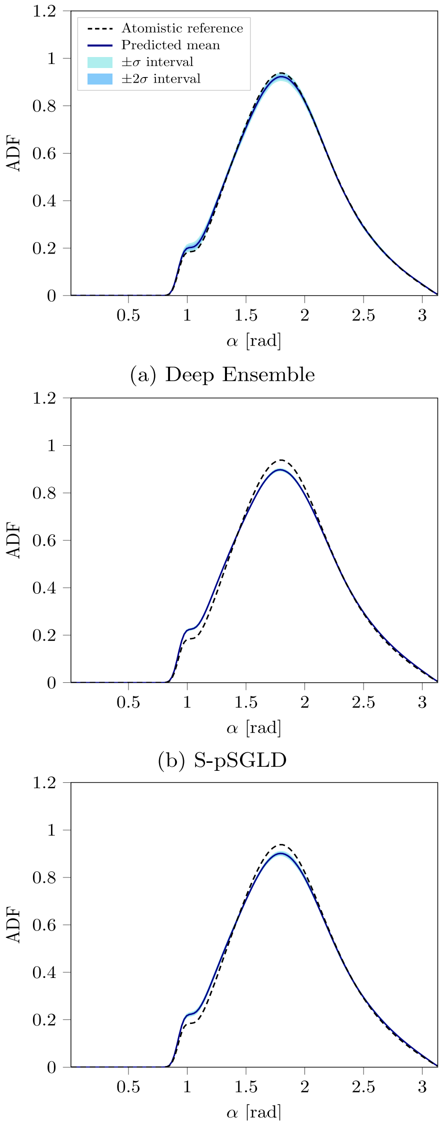
with ϵ_{w} = 1 kJ/mol and σ_{w} = 0.3165 nm, where σ_{w} corresponds to the length scale of the SPC water model.[68] This corresponds to the U^{prior} used in our previous works, where we found DimeNet++ results to be insensitive to the specific prior potential chosen.[13,60] We account for the thermodynamic state point dependency of the PMF[54,69,70] by augmenting the edge embedding of DimeNet++ by two learnable 16 dimensional vectors, one multiplied and one divided by $k_{\mathrm{B}}T$. This dual embedding ensures that the temperature dependency effect vanishes for neither high nor low k_BT.

The models are trained with a batch size of 5 boxes, an initial learning rate $a = 5 \times 10^{-4}$ and a polynomial learning rate decay schedule (eq 7) with $\gamma = 0.55$. We generate a Deep Ensemble of 8 models and train each for 100 epochs with the Adam[71] optimizer with default parameters. For each training trajectory, we select the model parameters with the smallest validation loss, giving the Deep Ensemble method a slight advantage in terms of data usage over the Bayesian methods. For Bayesian modeling, we select a prior distribution $p(\sigma_H) \sim \Gamma(5, 27)$, incorporating the prior knowledge that $\sigma_H > 0$ due to the noise from the noninjective CG mapping.[56] By default, we select a posterior temperature $\mathcal{T} = 0.01$. Each pSGLD chain is run for 10000 epochs, 8000 of which are discarded as burn-in. We randomly subsample the remaining models such that a total of 40 models are selected, evenly distributed over all available chains (8 chains for M-pSGLD, 1 chain for S-pSGLD). One chain of the M-pSGLD method yielded poor potentials, and we omitted it for a more balanced comparison.

First, we evaluate the mean force predictions on the test data. The Deep Ensemble method with a RMSE of 135.8 kJ/(mol nm) is more accurate than S-pSGLD and M-pSGLD with RMSE = 137.2 kJ/(mol nm) and RMSE = 136.6 kJ/(mol nm), respectively. We were unable to find a set of pSGLD hyperparameters that closed the error differential to the Deep Ensemble method. The force error is dominated by the large aleatoric uncertainty, which is estimated as $\sigma_H = 136.6$ kJ/(mol nm) by S-pSGLD.

We run CG MD simulations at a temperature $T = T_{\text{ref}}$ to investigate the resulting observables without a distribution shift. All CG MD simulations use a time step of 2 fs and are equilibrated for 10 ps, followed by 100 ps of production, where a state is retained every 0.1 ps. The CG MD simulation averages over the aleatoric uncertainty resulting from the CG mapping. Consequently, the predicted standard deviation of observables σ includes the epistemic uncertainty as well as a small amount of MD sampling uncertainty due to finite trajectory lengths. Figure 4 shows the resulting distributions of angular distribution functions (ADFs). The mean prediction of the Deep Ensemble method matches the AT reference well and is slightly more accurate than the S-pSGLD and M-pSGLD schemes, reflecting the lower test set RMSE. Additionally, the 2σ credible interval of the Deep Ensemble method covers the AT reference, and areas with higher uncertainty correspond to areas with larger error. M-pSGLD captures slightly more variance than S-pSGLD, but both schemes are overconfident. The overconfidence of M-pSGLD seems to be primarily attributable to a larger deviation of the predicted mean ADF from the reference curve (in line with the larger test set RMSE) and only secondarily to less captured epistemic uncertainty compared to the Deep Ensemble method. The conclusions drawn from the radial distribution function (RDF) predictions are identical (Supplementary Figure 4).

We investigate the impact of the Gaussian prior for weights and biases by retraining the NN potential with an improper uniform distribution. The obtained results for S-pSGLD are largely identical to the Gaussian prior case (Supplementary Figure 5). However, the Gaussian prior appears to improve the learning robustness: With the uniform distribution, a total of 4 M-pSGLD Markov chains yielded models with large errors in the mean predictions, compared to only a single Markov chain with the Gaussian prior. Having verified that neither of the priors $p(\mathbf{w})$ and $p(\sigma_H)$ are too restrictive, we hypothesize that the higher RMSEs of the pSGLD schemes may be the result of the training, where the coupling of learning rate and additive random noise might impede convergence to models with the highest likelihood.



(a) Deep Ensemble

(b) S-pSGLD

(c) M-pSGLD

Figure 4. Angular distribution functions (ADFs) at $T = T_{\text{ref}}$. Resulting mean ADFs with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the Deep Ensemble method (*a*), the single chain pSGLD (*b*) and the multichain pSGLD (*c*) schemes at a temperature $T = T_{\text{ref}}$, compared to the atomistic reference.

Next, we investigate the impact of the posterior temperature $\mathcal{T}$ on pSGLD models (Figure 5). With the Bayesian posterior $\mathcal{T} = 1$, S-pSGLD requires a large data set (1000 boxes) to sample accurate models. For a medium data set size (300 boxes), the obtained models are highly inaccurate compared to using the cold posterior $\mathcal{T} = 0.01$. For smaller data set sizes, models sampled with the Bayesian posterior result in unstable

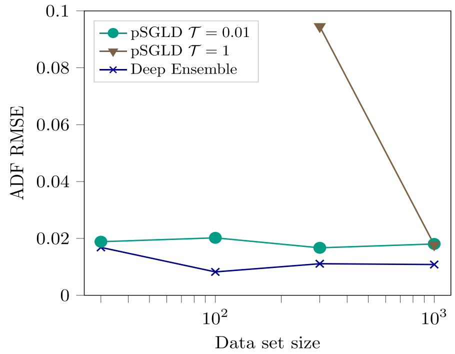
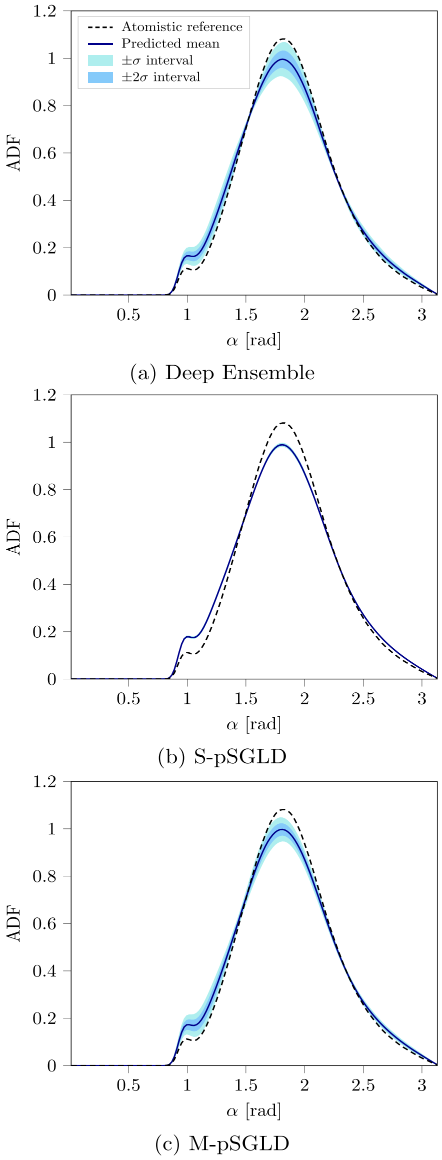


Figure 5. Cold posterior effect. Root-mean-squared error (RMSE) of the mean predicted angular distribution function (ADF) at $T = T_{\text{ref}}$ of the Deep Ensemble method and the single chain pSGLD scheme with $\mathcal{T} = 1$ and $\mathcal{T} = 0.01$ for different data sizes. Note that pSGLD $\mathcal{T} = 1$ yields unstable MD simulations for data sizes of 30 and 100 boxes.

CG MD simulations. By contrast, the cold posterior allows us to sample accurate models with only a fraction of the data (30 boxes). Moreover, the accuracy of pSGLD models hinges on a sufficient amount of burn-in epochs to reduce the learning rate (Supplementary Figure 6). Consequently, the pSGLD schemes require significantly more computational training effort in this example than the Deep Ensemble method. Still, the Deep Ensemble method yields more accurate models for all data set sizes considered in Figure 5.

To test the quality of UQ under distribution shift, we apply the obtained models at a temperature $T = 260$ K. The mean predictions of the considered UQ schemes are very similar to each other and, as expected, deviate from the respective TIP4P results (Figure 6). While S-pSGLD results in highly overconfident predictions, both M-pSGLD and the Deep Ensemble method provide accurate credible intervals, with a slight advantage for the latter. The predicted RDFs allow for identical conclusions (Supplementary Figure 7). The accurate ADF credible intervals we obtained with the M-pSGLD and the Deep Ensemble method stand in contrast to previous findings with a 2-body cubic spline model:[72] Given that the 2-body spline cannot model many-body effects, uncertainty with respect to 3-body interactions cannot be captured in the epistemic uncertainty. This model misspecification results in systematic uncertainty not included in the credible interval. This highlights the advantage of the many-body capabilities of NN potentials in a UQ context.

Finally, we study the impact of the k_BT-dependent edge embedding. In the first step, we match the AT reference pressure at T_{ref} during the FM training (details in Supplementary Methods 3). Using the Deep Ensemble method, we then compute the density ρ as a function of temperature with and without the k_BT-dependent embeddings (Figure 7). As desired, the credible interval includes the AT reference and the uncertainty increases with the distance from the training temperature T_{ref} for the k_BT-dependent model. By contrast, without k_BT-dependent embedding, the predicted uncertainty barely increases with the distance from T_{ref}, resulting in overconfident predictions due to model misspecification. Given that the k_BT dependence enables a broader range of outcomes at $T \neq T_{\text{ref}}$, the mean predictions also change significantly and yield smaller errors further away from T_{ref}. These results highlight the potency of scalable UQ methods to quantify errors resulting from applying CG models at different thermodynamic state points than during training.



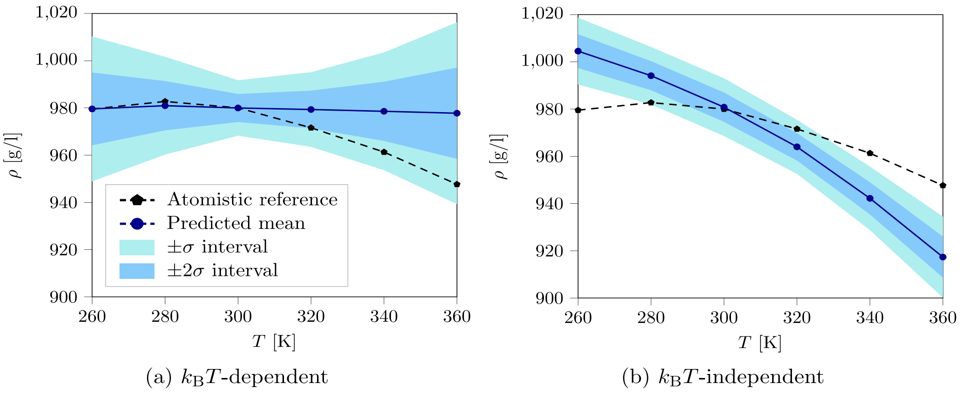
(a) Deep Ensemble

(b) S-pSGLD

(c) M-pSGLD

Figure 6. Out-of-distribution angular distribution functions (ADF) at $T = 260$ K. Resulting mean ADFs with ±σ and ±2σ intervals as predicted by the Deep Ensemble (*a*), the single chain pSGLD (*b*) and the multichain pSGLD (*c*) schemes at a temperature $T = 260$ K, compared to the atomistic reference.

3.3. Coarse-Grained Alanine Dipeptide. We consider the benchmark problem of learning the free energy surface (FES) of alanine dipeptide,[73,74] which has recently been shown to be a challenging task for NN potentials trained via FM.[57,60] Here, we investigate the sources of these challenges using the scalable UQ toolbox. We build on the computational setup of our previous study:[60] The CG map retains all 10 heavy atoms of alanine dipeptide, dropping hydrogen atoms and water molecules. The CG particles modeling CH_3, CH and C are encoded as different particle types. The training data set consists of a 100 ns AT trajectory at $T_{\text{ref}} = 300$ K, which is



(a) k_BT-dependent

(b) k_BT-independent

Figure 7. Water density profile. Resulting mean density ρ at pressure p = 1 bar with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the Deep Ensemble method using the k_BT-dependent reference model (a) and the same model without the k_BT-dependent edge embedding (b), compared to the atomistic reference.

subsampled to 5 × 10^5 data points by retaining a state every 0.2 ps. The first 80 ns form the training set, and the subsequent 8 ns, the validation set. To counteract the instability of DimeNet++ in CG MD simulations of alanine dipeptide,[57] we add a prior potential $U^{\mathrm{prior}}(\mathbf{R})$ (eq 11) that consists of harmonic bonds and angles, as well as proper dihedrals. For more technical details on U^{prior} and the AT reference data, we refer to our previous work.[60]

We train all models with an initial learning rate $a = 10^{-3}$ and a polynomial learning rate decay schedule (eq 7) with $\gamma = 0.55$, as well as a batch size of 512 configurations. The 8 models of the Deep Ensemble method are trained for 1000 epochs, using the Adam[71] optimizer with default parameters. For each training trajectory, we select the model parameters with the smallest validation loss. pSGLD chains are run for 3000 epochs, with the first 2500 epochs discarded as burn-in. We randomly subsample the remaining models such that a total of 40 models are selected, evenly distributed over all available chains (8 chains for M-pSGLD, 1 chain for S-pSGLD). We select a prior distribution $p(\sigma_H) \sim \Gamma(10, 40)$ and a posterior temperature $\mathcal{T} = 0.05$.

We initially evaluate the performance of the considered methods on the test set: The S-pSGLD (RMSE = 414.12 kJ/(mol nm)), M-pSGLD (RMSE = 414.01 kJ/(mol nm)) and Deep Ensemble methods (RMSE = 413.84 kJ/(mol nm)) yield very similar accuracy, with a slight advantage for the Deep Ensemble method. This is in line with the aleatoric uncertainty scale σ_H = 414.69, estimated by S-pSGLD.

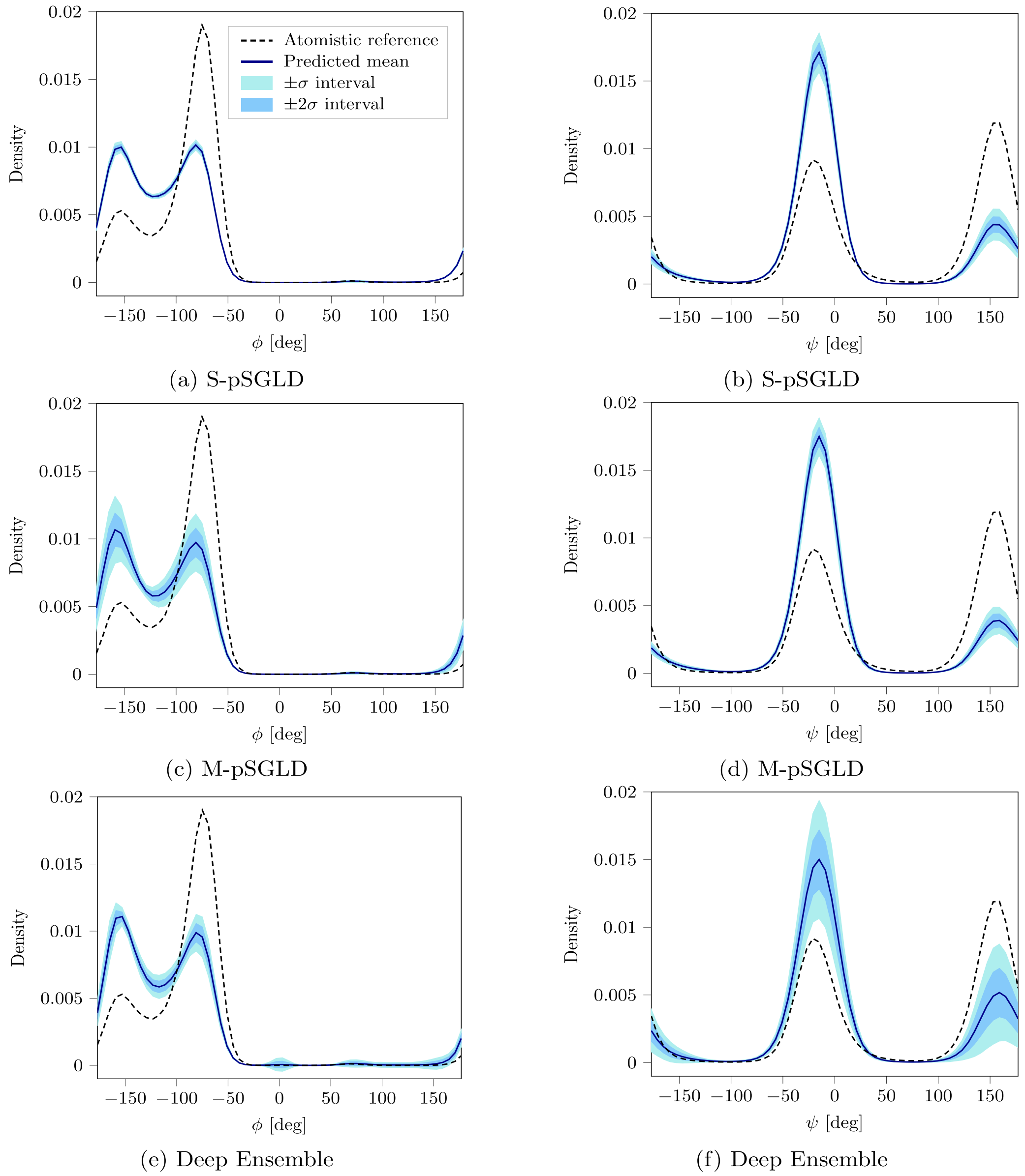
We perform a 100 ns CG MD production simulation for all sampled models in order to compute the FES. To obtain the same number of trajectories as with the pSGLD schemes, each of the 8 Deep Ensemble models generates 5 trajectories, all starting from different initial states. Despite using a prior potential, some models became stuck in unphysical potential energy "holes",[14] i.e. deep potential energy minima in rarely sampled phase-space regions, which also led to instability in some cases. These potential energy holes might be avoided by employing better prior potentials or by incorporating MD simulations into training, e.g. via active learning[14,19] or alternative training schemes such as relative entropy (RE) minimization.[60,75,76] We note that using the Bayesian posterior $\mathcal{T} = 1$ significantly increased the number of unphysical trajectories (tested for S-pSGLD). For $\mathcal{T} = 1$, we observed results of comparable quality to $\mathcal{T} = 0.05$ only when increasing the data set to 1 μs.

First, we investigate UQ results after removing unphysical trajectories that mainly sampled configurations in a potential energy hole. To this end, we removed 1, 7, and 13 trajectories from the S-pSGLD, the M-pSGLD and the Deep Ensemble methods, respectively. The resulting means and standard deviations of the dihedral angles ϕ and ψ[73,74] are shown in Figure 8. The mean predictions of S-pSGLD and M-pSGLD are very similar, but—consistent with the examples above—S-pSGLD significantly underestimates the epistemic uncertainty. The Deep Ensemble method yields similar mean predictions in ϕ and a slightly improved mean prediction in ψ. The M-pSGLD predicts larger epistemic uncertainty in ϕ, while the Deep Ensemble method predicts larger uncertainty in ψ. However, all considered methods show that the epistemic uncertainty is not sufficiently large to fully account for the deviation from the AT reference in this case.

To contextualize this result, we replace the FM training by RE minimization. Given that the RE model trained on the same data set can match the AT FES accurately,[60] insufficient model capacity is not the main limiting factor. Additionally, a poor approximation of the posterior by the considered UQ methods, which would result in an incorrect size of the predicted credible interval, can also be ruled out: The posterior probability ratio of the RE model and the last model sampled by the S-pSGLD FM scheme is $p(\theta_{\mathrm{RE}}|\mathcal{D})/p(\theta_{\mathrm{FM}}|\mathcal{D}) = e^{-25872}$. This is in line with previous findings showing that the error on held-out force data is smaller for FM than for RE for alanine dipeptide.[60] Given that the posterior probability ratio of the RE model is numerically zero, UQ schemes with a FM-based posterior cannot sample the RE model.

Multiple mechanisms may contribute to the comparatively weak FES prediction of FM models. First, the FES of a FM model is sensitive to predictions in sparsely resolved transition regions.[60] Second, if a CG MD simulation is able to reach unphysical phase-space regions, sampling such configurations yields an erroneous FES. Both of these mechanisms result in very large epistemic uncertainty. We empirically show this effect when we include trajectories that sampled potential energy holes in the evaluation of the FES distribution (Supplementary Figure 8). In particular, the predicted credible intervals of M-pSGLD and the Deep Ensemble method mostly

https://doi.org/10.1021/acs.jctc.2c01267
J. Chem. Theory Comput. 2023, 19, 4520−4532



(a) S-pSGLD (b) S-pSGLD

(c) M-pSGLD (d) M-pSGLD

(e) Deep Ensemble (f) Deep Ensemble

Figure 8. Dihedral angle density histograms. Resulting mean distribution of dihedral angles ϕ (left column) and ψ (right column) with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the single chain pSGLD (*a*, *b*), the multichain pSGLD (*c*, *d*) and the Deep Ensemble (*e*, *f*) methods, compared to the atomistic reference.

cover the reference FES in this case. Hence, these UQ methods can signal to practitioners that the obtained results are not yet trustworthy.

Rather, more data needs to be generated to further constrain the learned models. We increased the data set size by a factor of 10 by generating an AT trajectory of 1 μs length. Interestingly, simply generating more Boltzmann-distributed training data did not solve the potential energy hole problem, nor did it significantly reduce the deviation of the mean prediction when neglecting trajectories stuck in potential energy holes (Supplementary Figure 9). Hence, generating more diverse, non-Boltzmann distributed data sets, e.g. via enhanced sampling schemes[77,78] or active learning,[14,19] seems to be a more promising approach.

The remaining deviation of the mean prediction from the reference FES that is not captured by the predicted epistemic uncertainty (Figure 8) suggests that other, likely systematic sources of error exist. For finite model capacity, a systematic difference between FM and RE minimization is the objective function: RE training minimizes the difference between the potential energy surfaces of the AT and CG models,[79] which is directly related to the FES. In contrast, the optimum of a force-based training objective might trade off accuracy in the FES for improved accuracy in other (e.g., thermodynamic[60]) observables, resulting in systematic uncertainty in the predicted FES.

Additionally, numerical errors introduced by the CG MD simulation, similar to the shadow Hamiltonian effect,[80,81] can be corrected for by RE minimization.[60] In FM models, these numerical errors manifest as unquantifiable systematic uncertainty. However, for a comprehensive analysis of the relative impact of each error mechanism, further research is needed.

4. DISCUSSION AND CONCLUSION

Our results show that M-pSGLD is well suited to estimate the epistemic uncertainty of MD observables. This method enables fully-Bayesian UQ for NN potentials. All experiments highlight the importance of sampling multiple posterior modes. Exploiting the strong decorrelation effect of multiple random NN initializations via multiple Markov chains is an effective means to this end. In the graph NN examples, cold posteriors proved beneficial to sample both stable and accurate models, reducing the required amount of training data significantly. Hence, we found the number of Markov chains to be the most important additional M-pSGLD hyperparameter, followed by the posterior temperature, the prior distributions and the number of samples per chain.

Both the Deep Ensemble and the M-pSGLD methods provided good approximations to the epistemic uncertainty estimated by the NUTS[61] in the LJ example. In addition, the Deep Ensemble method yielded similar UQ quality to M-pSGLD, although it required less training and hyperparameter tuning effort. We found no evidence that the Deep Ensemble method was prone to overconfident predictions, contrasting prior research in an active learning setting.[27] Instead, our results suggest that the Deep Ensemble method quantifies epistemic uncertainty effectively, both within and out of the training distribution.

M-pSGLD promises accurate UQ by leveraging the complementary benefits from sampling multiple posterior modes and additional Bayesian exploration of each mode.[43,44,51] However, further research into SG-MCMC schemes is required before routine application in practice: In our experiments, a single MCMC chain (both pSGLD[28] and NUTS[61]) sampled a single posterior mode only. Hence, the development of methods that sample multiple posterior modes with a single chain, e.g., by leveraging cyclical step size schedules[82] or parallel tempering,[83] is important. Additionally, automatic hyperparameter tuning with a computationally efficient metric could improve the SG-MCMC efficiency; e.g., the popular Stein's discrepancy[84] scales quadratically with the data set size.[31] Finally, recent SG-MCMC samplers such as AMAGOLD[82] or SGGMC[85] include infrequent Metropolis-Hastings acceptance steps to avoid the bias of SGLD.[28,30] Consequently, these samplers use constant learning rates, which may counteract the increased training time of SGLD that results from its small learning rate requirement.[47,82]

We observed a clear cold posterior effect[38] in our experiments with the graph NN potential. For image classification tasks, cold posteriors have demonstrated superior performance in practice.[38,86,87] However, this performance increase is mainly attributed to data augmentation,[44] which increases the effective data size without increasing the data size considered in the likelihood. Analogously, the effective data size might be underestimated by the likelihood in eq 10, which the cold posterior may correct for: When learning the potential energy of a molecular state, the effective data size is clearly larger than a single data point. For instance, in the case of a pairwise additive potential, the effective data size corresponds to the total number of particle pairs within a cutoff. For FM, the data size per box considered in the likelihood in eq 10 equals 3 times the number of CG particles, but whether the effective data size exceeds this value is less clear. More research into the nature of the cold posterior effect is required—ideally resulting in likelihood formulations that better consider the effective data size.

Our results corroborate that for successful UQ, a sufficiently large hypothesis space is necessary: Effects describable by the model can be quantified reliably as epistemic uncertainty, but effects beyond the model capacity become hard to quantify systematic uncertainties.[15,42] For instance, if a potential lacks important many-body interactions or a CG model lacks state point dependency, the resulting uncertainty estimates are overconfident. Consequently, NN potentials are attractive models in a UQ context, given that they model many-body interactions inherently.

To obtain uncertainty estimates for MD observables, we performed a dedicated MD simulation for every sampled NN potential. This approach is rigorous, as both epistemic uncertainty and MD sampling uncertainty are captured,[18] but also computationally expensive. The computational effort for MD simulations scales linearly with the number of sampled potentials, but the simulations can be parallelized. Distilling the mean potential energy prediction into a single model via student−teacher[30,88,89] learning could improve computational efficiency. With this approach, one could obtain uncertainty estimates for time-averaged observables using a single MD simulation and a reweighting scheme.[24] Concerning the development of computationally more efficient UQ schemes for NN potentials,[40] we have demonstrated that both M-pSGLD and the Deep Ensemble method can serve as reliable baseline schemes. Efficient UQ schemes may pave the way for more reliable MD simulations based on NN potentials to support simulation-based decision-making in health care and material science industries.[18]

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.2c01267.

> Background information on the prior potential, training details of the Lennard-Jones example and training data visualization; pressure correction scheme; NUTS predictions for different numbers of Markov chains and fixed σ_H; predicted CG water RDFs; ADFs for S-pSGLD with uniform prior over weights and biases; ADF RMSEs for different pSGLD Markov chain lengths; and dihedral angle density histograms without removing potential energy holes as well as for the 1 μs data set. (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Stephan Thaler − *Professorship of Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, 85748 Garching near Munich, Germany;* orcid.org/0000-0001-5383-1615; Email: stephan.thaler@tum.de

Julija Zavadlav − *Professorship of Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, 85748 Garching near Munich, Germany; Munich Data Science Institute, Technical University of Munich, 85748 Garching near Munich, Germany;* orcid.org/0000-0002-4495-9956; Email: julija.zavadlav@tum.de

Author

Gregor Doehner − *Professorship of Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, 85748 Garching near Munich, Germany;* orcid.org/0000-0003-0247-5125

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.2c01267

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Paul Fuchs for addressing jax-sgmc feature requests. S.T. acknowledges financial support from the Munich Data Science Institute Seed Fund.

REFERENCES

(1) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(2) Schütt, K. T.; Kindermans, P. J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K. R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, Dec. 4−9, 2017, pp 992−1002.

(3) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, Aug. 6−11, 2017, pp 1263−1272.

(4) Klicpera, J.; Groß, J.; Günnemann, S. Directional Message Passing for Molecular Graphs. In *8th International Conference on Learning Representations*, Online, Apr. 26−May 1, 2020.

(5) Klicpera, J.; Giri, S.; Margraf, J. T.; Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. In *Machine Learning for Molecules Workshop at NeurIPS*, Online, Dec. 12, 2020.

(6) Ko, T. W.; Finkler, J. A.; Goedecker, S.; Behler, J. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **2021**, *12*, 398.

(7) Batzner, S.; Musaelian, A.; Sun, L.; Geiger, M.; Mailoa, J. P.; Kornbluth, M.; Molinari, N.; Smidt, T. E.; Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **2022**, *13*, 2453.

(8) Noé, F.; Tkatchenko, A.; Müller, K. R.; Clementi, C. Machine Learning for Molecular Simulation. *Annu. Rev. Phys. Chem.* **2020**, *71*, 361−390.

(9) 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.

(10) Faber, F. A.; Hutchison, L.; Huang, B.; Gilmer, J.; Schoenholz, S. S.; Dahl, G. E.; Vinyals, O.; Kearnes, S.; Riley, P. F.; Von Lilienfeld, O. A. Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theory Comput.* **2017**, *13*, 5255−5264.

(11) Stocker, S.; Gasteiger, J.; Becker, F.; Günnemann, S.; Margraf, J. T. How robust are modern graph neural network potentials in long and hot molecular dynamics simulations? *Mach. Learn.: Sci. Technol.* **2022**, *3*, 045010.

(12) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **2019**, *5*, 755−767.

(13) Thaler, S.; Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. *Nat. Commun.* **2021**, *12*, 6884.

(14) van der Oord, C.; Sachs, M.; Kovács, D. P.; Ortner, C.; Csányi, G. Hyperactive Learning (HAL) for Data-Driven Interatomic Potentials. *arXiv* **2022**, DOI: 10.48550/arXiv.2210.04225.

(15) Gal, Y.; Koumoutsakos, P.; Lanusse, F.; Louppe, G.; Papadimitriou, C. Bayesian uncertainty quantification for machine-learned models in physics. *Nat. Rev. Phys.* **2022**, *4*, 573−577.

(16) Angelikopoulos, P.; Papadimitriou, C.; Koumoutsakos, P. Bayesian uncertainty quantification and propagation in molecular dynamics simulations: a high performance computing framework. *J. Chem. Phys.* **2012**, *137*, 144103.

(17) Zavadlav, J.; Arampatzis, G.; Koumoutsakos, P. Bayesian selection for coarse-grained models of liquid water. *Sci. Rep.* **2019**, *9*, 99.

(18) Wan, S.; Sinclair, R. C.; Coveney, P. V. Uncertainty quantification in classical molecular dynamics. *Philos. Trans. Royal Soc. A* **2021**, *379*, 20200082.

(19) 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.

(20) Zhang, L.; Lin, D.-Y.; Wang, H.; Car, R.; Weinan, E. Active learning of uniformly accurate interatomic potentials for materials simulation. *Phys. Rev. Mater.* **2019**, *3*, 023804.

(21) Loeffler, T. D.; Patra, T. K.; Chan, H.; Sankaranarayanan, S. K. Active learning a coarse-grained neural network model for bulk water from sparse training data. *Mol. Syst. Des. Eng.* **2020**, *5*, 902−910.

(22) Smith, J. S.; Nebgen, B.; Mathew, N.; Chen, J.; Lubbers, N.; Burakovsky, L.; Tretiak, S.; Nam, H. A.; Germann, T.; Fensin, S.; et al. Automated discovery of a robust interatomic potential for aluminum. *Nat. Commun.* **2021**, *12*, 1257.

(23) Xie, S. R.; Rupp, M.; Hennig, R. G. Ultra-fast Force Fields (UF3) framework for machine-learning interatomic potentials. In *American Physical Society March Meeting*, Chicago, IL, USA, Mar. 14−18, 2021.

(24) Imbalzano, G.; Zhuang, Y.; Kapil, V.; Rossi, K.; Engel, E. A.; Grasselli, F.; Ceriotti, M. Uncertainty estimation for molecular dynamics and sampling. *J. Chem. Phys.* **2021**, *154*, 074102.

(25) Duane, S.; Kennedy, A. D.; Pendleton, B. J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195*, 216−222.

(26) Neal, R. M. In *Handbook of Markov Chain Monte Carlo*, 1st ed.; Brooks, S., Gelman, A., Jones, G. L., Meng, X.-L., Eds.; Chapman and Hall/CRC: New York, USA, 2011; Chapter MCMC using Hamiltonian Dynamics, pp 139−188.

(27) Kahle, L.; Zipoli, F. Quality of uncertainty estimates from neural network potential ensembles. *Phys. Rev. E* **2022**, *105*, 015311.

(28) Welling, M.; Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, Jun. 28−Jul. 2, 2011, pp 681−688.

(29) Chen, T.; Fox, E.; Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, Jun. 21−26, 2014, pp 1683−1691.

(30) Li, C.; Chen, C.; Carlson, D. E.; Carin, L. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Phoenix, AZ, USA, February 12−17, 2016, pp 1788−1794.

(31) Nemeth, C.; Fearnhead, P. Stochastic gradient Markov chain Monte Carlo. *J. Am. Stat. Assoc.* **2021**, *116*, 433−450.

(32) Lamb, G.; Paige, B. Bayesian Graph Neural Networks for Molecular Property Prediction. In *Machine Learning for Molecules Workshop at NeurIPS*, Online, Dec. 12, 2020.

(33) Graves, A. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, Granada, Spain, Dec. 12−14, 2011.

(34) Hoffman, M. D.; Blei, D. M.; Wang, C.; Paisley, J. Stochastic Variational Inference. *J. Mach. Learn. Res.* **2013**, *14*, 1303−1347.

(35) Hansen, L.; Salamon, P. Neural Network Ensembles. *IEEE Trans. Pattern Anal. Machine Intell.* **1990**, *12*, 993−1001.

(36) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, Dec. 4−9, 2017.

(37) Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; Snoek, J. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, Dec. 8−14, 2019.

(38) Wenzel, F.; Roth, K.; Veeling, B. S.; Swikatkowski, J.; Tran, L.; Mandt, S.; Snoek, J.; Salimans, T.; Jenatton, R.; Nowozin, S. How Good is the Bayes Posterior in Deep Neural Networks Really?. In *Proceedings of the 37th International Conference on Machine Learning*, Online, Jul. 13−18, 2020, pp 10248−10259.

(39) Wen, M.; Tadmor, E. B. Uncertainty quantification in molecular simulations with dropout neural network potentials. *npj Comput. Mater.* **2020**, *6*, 124.

(40) Zhu, A.; Batzner, S.; Musaelian, A.; Kozinsky, B. Fast Uncertainty Estimates in Deep Learning Interatomic Potentials. *arXiv* **2022**, DOI: 10.48550/arXiv.2211.09866.

(41) Gustafsson, F. K.; Danelljan, M.; Schon, T. B. Evaluating Scalable Bayesian Deep Learning Methods for Robust Computer Vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Seattle, WA, USA, Jun. 14−19, 2020, 1289−1298.

(42) Hüllermeier, E.; Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Mach. Learn.* **2021**, *110*, 457−506.

(43) Wilson, A. G.; Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*, Online, Dec. 6−12, 2020.

(44) Izmailov, P.; Vikram, S.; Hoffman, M. D.; Wilson, A. G. G. What Are Bayesian Neural Network Posteriors Really Like?. In *Proceedings of the 38th International Conference on Machine Learning*, Online, Jul. 18−44, 2021, pp 4629−4640.

(45) Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970**, *57*, 97−109.

(46) Li, D. W.; Brüschweiler, R. Iterative Optimization of Molecular Mechanics Force Fields from NMR Data of Full-Length Proteins. *J. Chem. Theory Comput.* **2011**, *7*, 1773−1782.

(47) Teh, Y. W.; Thiery, A. H.; Vollmer, S. J. Consistency and Fluctuations For Stochastic Gradient Langevin Dynamics. *J. Mach. Learn. Res.* **2016**, *17*, 1−33.

(48) Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* **2012**, *4*, 26−31.

(49) Dauphin, Y. N.; Pascanu, R.; Gulcehre, C.; Cho, K.; Ganguli, S.; Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 8−13, 2014.

(50) Thaler, S.; Fuchs, P.; Cukarska, A.; Zavadlav, J. *jax-sgmc: Modular Stochastic Gradient MCMC for JAX*. 2020; https://github.com/tummfm/jax-sgmc.

(51) Fort, S.; Hu, H.; Lakshminarayanan, B. Deep ensembles: A loss landscape perspective*arXiv*. *arXiv* **2019**, DOI: 10.48550/arXiv.1912.02757.

(52) Smith, J. S.; Zubatyuk, R.; Nebgen, B.; Lubbers, N.; Barros, K.; Roitberg, A. E.; Isayev, O.; Tretiak, S. The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Sci. Data* **2020**, *7*, 134.

(53) Izvekov, S.; Voth, G. A. Multiscale coarse graining of liquid-state systems. *J. Chem. Phys.* **2005**, *123*, 134105.

(54) Noid, W. G.; Chu, J.-W.; Ayton, G. S.; Krishna, V.; Izvekov, S.; Voth, G. A.; Das, A.; Andersen, H. C. The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **2008**, *128*, 244114.

(55) Noid, W.; Liu, P.; Wang, Y.; Chu, J.-W.; Ayton, G. S.; Izvekov, S.; Andersen, H. C.; Voth, G. A. The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *J. Chem. Phys.* **2008**, *128*, 244115.

(56) Wang, H.; Junghans, C.; Kremer, K. Comparative atomistic and coarse-grained study of water: What do we lose by coarse-graining? *Eur. Phys. J. E* **2009**, *28*, 221−229.

(57) Fu, X.; Wu, Z.; Wang, W.; Xie, T.; Keten, S.; Gomez-Bombarelli, R.; Jaakkola, T. Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations. In *AI for Science: Progress and Promises Workshop at NeurIPS*, New Orleans, LA, USA, Dec. 2, 2022.

(58) Das, A.; Andersen, H. C. The multiscale coarse-graining method. III. A test of pairwise additivity of the coarse-grained potential and of new basis functions for the variational calculation. *J. Chem. Phys.* **2009**, *131*, 034102.

(59) Husic, B. E.; Charron, N. E.; Lemm, D.; Wang, J.; Pérez, A.; Krämer, A.; Chen, Y.; Olsson, S.; de Fabritiis, G.; Noé, F.; Clementi, C. Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **2020**, *153*, 194101.

(60) Thaler, S.; Stupp, M.; Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *J. Chem. Phys.* **2022**, *157*, 244103.

(61) Hoffman, M. D.; Gelman, A. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **2014**, *15*, 1593−1623.

(62) Gelman, A.; Lee, D.; Guo, J. Stan: A probabilistic programming language for Bayesian inference and optimization. *J. Educ. Behav. Stat.* **2015**, *40*, 530−543.

(63) Lao, J.; Louf, R. *Blackjax: A sampling library for JAX*. 2020; http://github.com/blackjax-devs/blackjax.

(64) Tran, K.; Neiswanger, W.; Yoon, J.; Zhang, Q.; Xing, E.; Ulissi, Z. W. Methods for comparing uncertainty quantifications for material property predictions. *Mach. Learn. Sci. Technol.* **2020**, *1*, 025006.

(65) Chung, Y.; Char, I.; Guo, H.; Schneider, J.; Neiswanger, W. Uncertainty Toolbox: an Open-Source Library for Assessing, Visualizing, and Improving Uncertainty Quantification. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, Online, July 23, **2021**.

(66) Hirschfeld, L.; Swanson, K.; Yang, K.; Barzilay, R.; Coley, C. W. Uncertainty quantification using neural networks for molecular property prediction. *J. Chem. Inf. Model.* **2020**, *60*, 3770−3780.

(67) Abascal, J. L. F.; Vega, C. A general purpose model for the condensed phases of water: TIP4P/2005. *J. Chem. Phys.* **2005**, *123*, 234505.

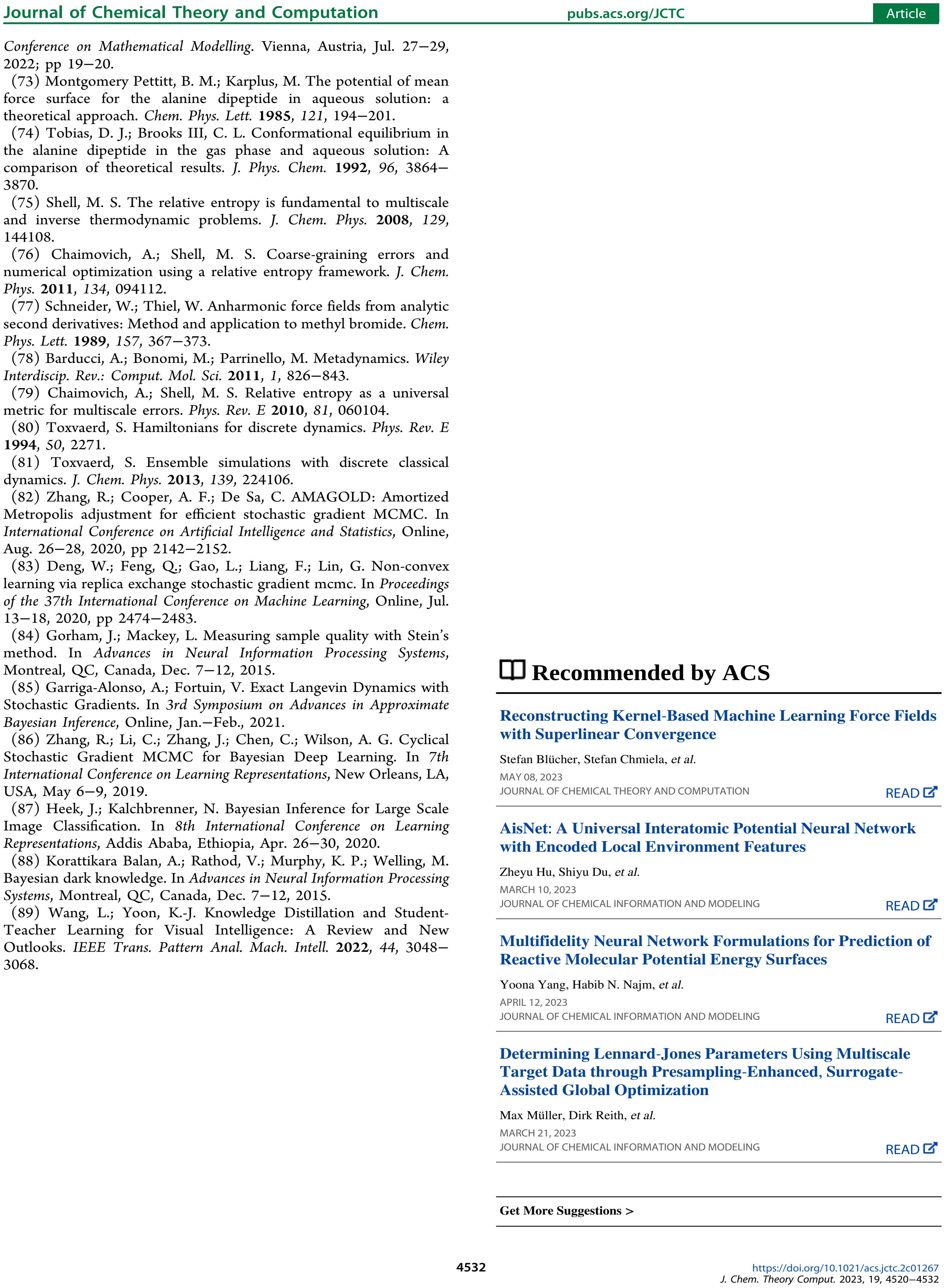
(68) Berendsen, H. J. C.; Postma, J. P. M.; van Gunsteren, W. F.; Hermans, J. Interaction models for water in relation to protein hydration. In *Intermolecular forces: Proceedings of the Fourteenth Jerusalem Symposium on Quantum Chemistry and Biochemistry*, Jerusalem, Israel, Apr. 13−16, 1981; Vol. *14*. pp 331−342.

(69) Chaimovich, A.; Shell, M. S. Anomalous waterlike behavior in spherically-symmetric water models optimized with the relative entropy. *Phys. Chem. Chem. Phys.* **2009**, *11*, 1901−1915.

(70) Potestio, R.; Peter, C.; Kremer, K. Computer Simulations of Soft matter: Linking the Scales. *Entropy* **2014**, *16*, 4199−4245.

(71) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*, San Diego, CA, USA, May 7−9, 2015.

(72) Thaler, S.; Zavadlav, J. Uncertainty Quantification for Molecular Models via Stochastic Gradient MCMC. In *10th Vienna*



Conference on Mathematical Modelling. Vienna, Austria, Jul. 27−29, 2022; pp 19−20.

(73) Montgomery Pettitt, B. M.; Karplus, M. The potential of mean force surface for the alanine dipeptide in aqueous solution: a theoretical approach. *Chem. Phys. Lett.* **1985**, *121*, 194−201.

(74) Tobias, D. J.; Brooks III, C. L. Conformational equilibrium in the alanine dipeptide in the gas phase and aqueous solution: A comparison of theoretical results. *J. Phys. Chem.* **1992**, *96*, 3864−3870.

(75) Shell, M. S. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.* **2008**, *129*, 144108.

(76) Chaimovich, A.; Shell, M. S. Coarse-graining errors and numerical optimization using a relative entropy framework. *J. Chem. Phys.* **2011**, *134*, 094112.

(77) Schneider, W.; Thiel, W. Anharmonic force fields from analytic second derivatives: Method and application to methyl bromide. *Chem. Phys. Lett.* **1989**, *157*, 367−373.

(78) Barducci, A.; Bonomi, M.; Parrinello, M. Metadynamics. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 826−843.

(79) Chaimovich, A.; Shell, M. S. Relative entropy as a universal metric for multiscale errors. *Phys. Rev. E* **2010**, *81*, 060104.

(80) Toxvaerd, S. Hamiltonians for discrete dynamics. *Phys. Rev. E* **1994**, *50*, 2271.

(81) Toxvaerd, S. Ensemble simulations with discrete classical dynamics. *J. Chem. Phys.* **2013**, *139*, 224106.

(82) Zhang, R.; Cooper, A. F.; De Sa, C. AMAGOLD: Amortized Metropolis adjustment for efficient stochastic gradient MCMC. In *International Conference on Artificial Intelligence and Statistics*, Online, Aug. 26−28, 2020, pp 2142−2152.

(83) Deng, W.; Feng, Q.; Gao, L.; Liang, F.; Lin, G. Non-convex learning via replica exchange stochastic gradient mcmc. In *Proceedings of the 37th International Conference on Machine Learning*, Online, Jul. 13−18, 2020, pp 2474−2483.

(84) Gorham, J.; Mackey, L. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 7−12, 2015.

(85) Garriga-Alonso, A.; Fortuin, V. Exact Langevin Dynamics with Stochastic Gradients. In *3rd Symposium on Advances in Approximate Bayesian Inference*, Online, Jan.−Feb., 2021.

(86) Zhang, R.; Li, C.; Zhang, J.; Chen, C.; Wilson, A. G. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *7th International Conference on Learning Representations*, New Orleans, LA, USA, May 6−9, 2019.

(87) Heek, J.; Kalchbrenner, N. Bayesian Inference for Large Scale Image Classification. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia, Apr. 26−30, 2020.

(88) Korattikara Balan, A.; Rathod, V.; Murphy, K. P.; Welling, M. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, Montreal, QC, Canada, Dec. 7−12, 2015.

(89) Wang, L.; Yoon, K.-J. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3048−3068.

https://doi.org/10.1021/acs.jctc.2c01267

3.2.2. JaxSGMC: Modular stochastic gradient MCMC in JAX

Thaler, S., Fuchs, P., Cukarska, A. & Zavadlav, J. JaxSGMC: Modular stochastic gradient MCMC in JAX (2023). URL `https://github.com/tummfm/jax-sgmc`. (in review process *SoftwareX*)

Summary

Trustworthy predictions with uncertainty-aware MD simulations and increased data efficiency via active learning are promising avenues for progress in NN potentials. These approaches address main weaknesses of NN potentials, but their effectiveness critically depends on the quality of UQ estimates. As shown in sec. 3.2.1, for reliable UQ of NN potentials, it is insufficient to sample a single NN posterior mode only, as is typically the case with the Stochastic Variation Inference or Dropout Monte Carlo methods. The popular Deep Ensemble scheme captures different posterior modes, but neglects the uncertainty contribution from the volume of the posterior. In contrast, SG-MCMC, especially with multiple Markov chains, can sample multiple modes as well as the volume of the posterior. Despite this theoretical advantage, SG-MCMC schemes are still underused in practice - in part due to two main problems: First, SG-MCMC schemes lack a comprehensive and easy-to-use library that implements state-of-the-art SG-MCMC samplers to promote their application to ML problems in practice. Second, sec. 3.2.1 has shown that simple SG-MCMC schemes such as pSGLD cannot fully take advantage of the additional exploration of the posterior volume. Hence, further research into SG-MCMC samplers with more sophisticated posterior exploration capabilities is needed.

To address these issues, this paper presents *JaxSGMC*, a library for SG-MCMC in JAX. *JaxSGMC* reduces the barriers for switching from stochastic optimization to SG-MCMC sampling by providing a common API (`alias.py`) for several state-of-the-art SG-MCMC samplers. The implemented samplers can replace stochastic optimizers without modifications to the JAX NN model, making recently developed SG-MCMC schemes available to a broader user base. The software architecture of *JaxSGMC* focuses on modularity to accelerate research into novel SG-MCMC schemes by increasing the re-usability of SG-MCMC building blocks as well as boilerplate Bayesian modeling code. Additionally, inspired by *optax*, this modular structure allows users to combine these SG-MCMC building blocks to construct custom samplers tailored to the ML problem at hand.

Two standard ML problems showcase the use of the library: First, a linear regression problem illustrates data loading and building a custom sampler. Second, an image classification task for CIFAR-10 demonstrates the ease of use of pre-built SG-MCMC samplers in deep learning applications in practice. In sum, *JaxSGMC* aims to promote uncertainty-aware ML via SG-MCMC in molecular modeling and beyond.

CRediT author statement

Stephan Thaler: Conceptualization, Funding aquisition, Methodology, Software, Writing – original draft, Writing – review & editing

Paul Fuchs: Conceptualization, Methodology, Software, Validation, Visualization, Writing – original draft

Ana Cukarska: Formal analysis, Visualization, Writing – original draft

Julija Zavadlav: Supervision, Writing – review & editing

JaxSGMC: Modular stochastic gradient MCMC in JAX

Stephan Thaler[a,1], Paul Fuchs[a,1], Ana Cukarska[a], Julija Zavadlav[a,b]

[a] *Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, Germany*
[b] *Munich Data Science Institute, Technical University of Munich, Germany*

Abstract

We present *JaxSGMC*, an application-agnostic library for stochastic gradient Markov chain Monte Carlo (SG-MCMC) in JAX. SG-MCMC schemes are uncertainty quantification (UQ) methods that scale to large datasets and high-dimensional models, enabling trustworthy neural network predictions via Bayesian deep learning. *JaxSGMC* implements several state-of-the-art SG-MCMC samplers to promote UQ in deep learning by reducing the barriers of entry for switching from stochastic optimization to SG-MCMC sampling. Additionally, *JaxSGMC* allows users to build custom samplers from standard SG-MCMC building blocks. Due to this modular structure, we anticipate that *JaxSGMC* will accelerate research into novel SG-MCMC schemes and facilitate their application across a broad range of domains.

Keywords: SGMCMC, Bayesian Inference, Machine Learning

Email address: `julija.zavadlav@tum.de` (Julija Zavadlav)

[1] contributed equally

Preprint submitted to SoftwareX *June 15, 2023*

Code Metadata

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v0.1.3
C2	Permanent link to code/repository used for this code version	https://github.com/tummfm/jax-sgmc
C3	Code Ocean compute capsule	None
C4	Legal Code License	Apache-2.0
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python
C7	Compilation requirements, operating environments & dependencies	JAX
C8	If available Link to developer documentation/manual	https://jax-sgmc.readthedocs.io
C9	Support email for questions	stephan.thaler@tum.de

Table 1: Code metadata

1. Motivation and significance

Deep learning models have seen enormous success in many scientific fields over the last decade, including disciplines as diverse as natural language processing [1], autonomous driving [2], health care [3] and physics-based modeling [4, 5, 6]. However, neural networks (NNs) are data-driven black-box models – their predictions can be highly inaccurate when applied outside their training distribution. Uncertainty Quantification (UQ) provides a means to evaluate the trustworthiness of predictions, which is imperative for applying NNs in practice, in particular for safety-critical applications.

Bayesian statistics is the mathematical foundation of UQ, but classical Bayesian UQ methods based on Markov chain Monte Carlo (MCMC) [7, 8] are intractable for computationally expensive NNs and large datasets [9]. Stochastic gradient (SG) MCMC schemes [9, 10, 11, 12, 13] circumvent the need for a full evaluation of the likelihood per parameter update of classical MCMC by leveraging a stochastic estimate of the gradient of the likelihood over a mini-batch of data. This results in a large computational speed-up, enabling Bayesian deep learning.

The landscape of UQ libraries is fragmented: There are domain-dependent libraries such as NeuralUQ [14] on the one hand and domain-independent libraries on the other hand. TensorFlow Probability [15] and Pyro [16] are the most popular domain-independent UQ libraries for Tensorflow and PyTorch, respectively. Both focus on classical Hamiltonian Monte Carlo [7] schemes and Stochastic Variational Inference [17], while Stochastic Gradient Langevin Dynamics (SGLD) is the only implemented SG-MCMC sampler. Thus, dedicated SG-MCMC libraries have been developed for Tensorflow [18], Theano [19] and JAX [20]. However, the structure of these libraries currently does not allow for recently proposed SG-MCMC building blocks such as parallel tempering [21] and amortized Metropolis Hastings (MH) acceptance steps [22, 23]. Hence, many newly developed SG-MCMC samplers are published as stand-alone code [24, 21, 22] and do not take advantage of these existing libraries, which slows adoption of novel SG-MCMC samplers in practice.

In this work, we introduce the domain-independent *JaxSGMC* library. *JaxSGMC* implements several state-of-the-art SG-MCMC samplers such as replica exchange SG-MCMC [21] and AMAGOLD [22]. The implemented SG-MCMC schemes follow a common application programming interface (API), which simplifies switching between samplers and reduces the barriers of entry to UQ for practitioners. The SG-MCMC samplers are designed in a modular fashion, which allows re-using standard SG-MCMC building blocks. Additionally, the samplers can be compiled end-to-end just-in-time (jit), which improves their computational efficiency.

2. Software description

2.1. Bayesian Modeling

A model consists of an architecture $\mathcal{M}$ and parameters $\boldsymbol{\theta}$. In deep learning, these models are NNs, such as ResNet [25], with millions of parameters. The frequentist machine learning (ML) approach selects a good model via stochastic optimization of a loss function to obtain an optimal set of parameters $\bar{\boldsymbol{\theta}}$ that best fits a dataset $\mathcal{D}$ (e.g. CIFAR-10 [26]). The selected $\bar{\boldsymbol{\theta}}$ critically determines the model performance and reliability.

In contrast, the Bayesian approach studies the posterior predictive distribution

$$p(\mathrm{y}|\mathbf{x}, \mathcal{D}, \mathcal{M}) = \int p(\mathrm{y}|\mathbf{x}, \boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) \mathrm{d}\boldsymbol{\theta} \ , \tag{1}$$

which encodes the uncertainty in the model prediction y given an input $\mathbf{x}$. Instead of betting on a single parameter set $\bar{\boldsymbol{\theta}}$, eq. (1) considers an infinite number of models weighted according to their agreement with the dataset given by the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$. This integral is analytically intractable, but can be estimated via Monte Carlo integration employing a finite number of models

$$p(\mathrm{y}|\mathbf{x}, \mathcal{D}, \mathcal{M}) \approx \frac{1}{N_{\text{models}}} \sum_{i=1}^{N_{\text{models}}} p(\mathrm{y}|\mathbf{x}, \boldsymbol{\theta}_i, \mathcal{M}); \qquad \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M}) \tag{2}$$

drawn from the posterior distribution. Bayes formula relates the posterior

$$p(\boldsymbol{\theta}|\mathcal{D},\mathcal{M}) = \frac{p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M})p(\boldsymbol{\theta}|\mathcal{M})}{p(\mathcal{D}|\mathcal{M})} \propto \exp\left(-\mathcal{U}(\boldsymbol{\theta})\right) \tag{3}$$

to the likelihood $p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M})$, prior $p(\boldsymbol{\theta}|\mathcal{M})$ and model evidence $p(\mathcal{D}|\mathcal{M})$, which normalizes the distribution. The likelihood is the probability that a model could have generated the data. Complementary, the prior encodes beliefs about the model, independent of the data. Likelihood and prior are closely connected to loss functions and regularization techniques commonly used in the frequentist ML approach. However, no known method exists that can generate independent samples from arbitrary distributions.

Instead, modern MCMC algorithms propose sequences of samples by simulating physical processes such as Hamiltonian or Langevin dynamics, which are driven by the gradient of the potential $\mathcal{U}(\boldsymbol{\theta})$ (eq. (3)). An additional MH step accepts or rejects each proposal such that the equilibrium distribution of the Markov chain agrees with the posterior distribution [27]. Extending gradient-based MCMC approaches to big data applications is computationally infeasible due to the dependence of the gradient $\nabla\mathcal{U}(\boldsymbol{\theta})$ on all data points, which needs to be computed at each timestep of the simulation.

Similar to stochastic gradient descent (SGD) schemes, SG-MCMC methods resort to a noisy estimate of the potential

$$\mathcal{U}(\boldsymbol{\theta}) \approx -\frac{N}{n}\sum_{i=1}^{n} \log p(\mathrm{y}_i|\mathbf{x}_i,\boldsymbol{\theta},\mathcal{M}) - \log p(\boldsymbol{\theta}|\mathcal{M}) \tag{4}$$

based on a random mini-batch of n data points [9]. However, these approximate dynamics bias the equilibrium distribution and render the conventional MH corrections inapplicable [9, 23]. Nevertheless, classical SG-MCMC schemes achieve an asymptotically correct equilibrium distribution by adequately annealing the simulation timestep $\Delta t \to 0$ and adding the right amount of noise to the stochastic gradient [9, 28].

More recent SG-MCMC schemes offer enhanced MH steps to sample from the unbiased distribution at finite Δt. These MH steps accept or reject multiple consecutive proposals, while requiring a full potential evaluation only once [22, 23]. Other schemes aim to improve the mixing behavior of the Markov chain on highly curved NN posteriors by extending stochastic optimization algorithms, such as Adam and preconditioned SGD, to SG-MCMC [29, 11]. Additionally, tempered and multi-chain algorithms enhance exploratory capabilities to address high posterior multi-modality, e.g. by cyclically annealing temperature and timestep size [30] or swapping samples between tempered and non-tempered chains [21].

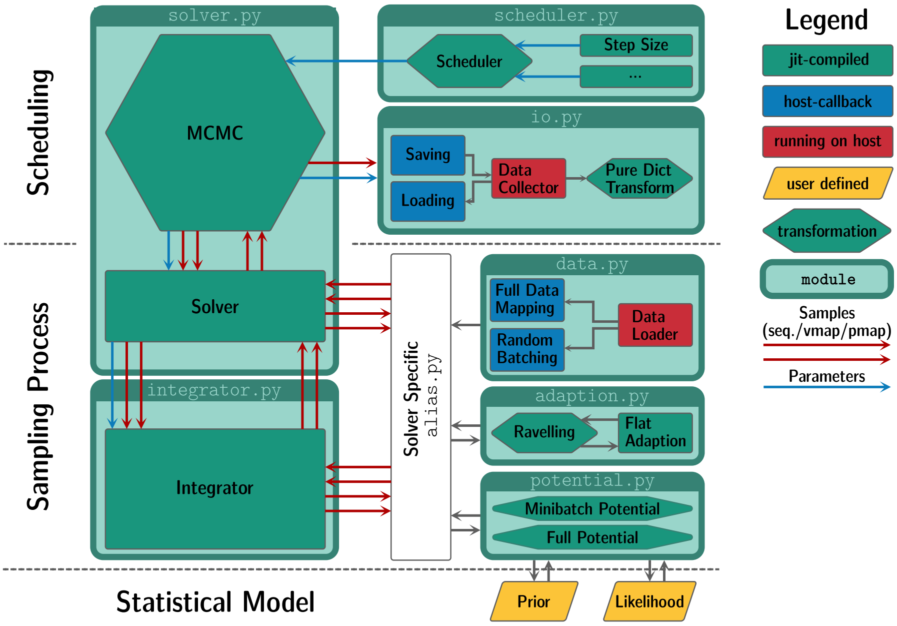


Figure 1: Software architecture of *JaxSGMC*. The modules are ordered hierarchically from left to right, and the specificity of a module to the machine learning problem increases from top to bottom.

2.2. Software Architecture

Fig. 1 visualizes the relationship between the software modules in *JaxSGMC*. Each module contains algorithmic building blocks that form a SG-MCMC sampler. Concisely, an SG-MCMC sampler repeats the following steps: (a) Retrieve the current process parameters (Δt, temperature, ...), (b) simulate the process via the potential $\mathcal{U}$ starting from the current sample $\boldsymbol{\theta}_i$, (c) process the obtained proposal and (d) save the new sample $\boldsymbol{\theta}_{i+1}$. The specific implementation of these steps defines the SG-MCMC sampler.

Each sampler builds on a user-provided model, which can be any JAX-transformable function, e.g. a NN. The model is part of the log-likelihood, which – together with the log-prior – forms the potential $\mathcal{U}(\boldsymbol{\theta})$ (eq. (4)). $\mathcal{U}(\boldsymbol{\theta})$ represents the statistical model and links the model and the sampler. Although sometimes plainly referred to as likelihood and prior, all computations rely on log-transformed values for computational reasons. Due to the (assumed) independence of observations in the dataset, the user-provided log-likelihood function computes the log-probability $p(\mathrm{y}_k|\mathbf{x}_k, \boldsymbol{\theta}_i, \mathcal{M})$ of the current model $\boldsymbol{\theta}_i$ for a single observation $(\mathbf{x}_k, \mathrm{y}_k) \in \mathcal{D}$ drawn from the dataset. The functions in the `potential.py` module then apply the log-probability to mini-batches of data to compute $\mathcal{U}(\boldsymbol{\theta}_i)$ or its stochastic estimate (eq. (4)). By designing *JaxSGMC* in a functional programming style, we leverage JAX's function transformations and automatic differentiation throughout the framework to perform these batched evaluations of the log-likelihood, to calculate the (stochastic) gradients $\nabla\mathcal{U}(\boldsymbol{\theta})$ and run multiple Markov chains.

JaxSGMC focuses on the big data case, where storing the whole dataset on the device (GPU) is inefficient or impossible due to memory limitations. Yet, functions relying on data should support JAX transformations. To this end, the `data.py` module offers an API around the host-callback module of JAX to efficiently insert data from so-called DataLoaders into jit-compiled computations. By providing different DataLoaders, we support multiple mini-batch assembly methods as well as straightforward data integration from different sources. Similarly, already a small number of parameter samples $\boldsymbol{\theta}_i$ of deep NNs can fill up the device memory. Accordingly, we designed the `io.py` module analogously to the `data.py` module to efficiently store the gathered samples. Hence, *JaxSGMC* enables collecting many samples from within the jit-compiled SG-MCMC algorithm and saving them to different file formats.

The `adaption.py` module, which supports the adaption of process quantities to the learning problem, together with the `data.py` and `potential.py` modules, provide all relevant components for the core of a SG-MCMC algorithm. This core lies in the sampling section simulating the physical process (step (b)) and processing the proposals (step (c)). In line with the modularity design principle, *JaxSGMC* subsequently separates the former into the `integrator.py` module and the latter into the `solver.py` module.

The scheduling part of the algorithm builds on top of the sampling process (fig. 1). The scheduling section includes boilerplate code, which provides a general entry point to run a constructed algorithm. In particular, it interfaces the schedules of the process parameters (step (a)) in `scheduler.py` with the Markov chain (steps (b) and (c)) and saves the current solver state or the collected samples (step (d)). Typically, the schedules are static and thus independent of the sampling process, but *JaxSGMC* also supports a feedback loop to enable adaptive step-size schemes.

2.3. Software Functionalities

We implemented two API levels: First, we built a high-level interface to popular SG-MCMC samplers in `alias.py`, including (preconditioned) SGLD [9, 11], stochastic gradient Hamiltonian Monte Carlo (SGHMC) [10], replica exchange SG-MCMC (reSGLD) [21], AMAGOLD [22], and stochastic gradient guided Monte Carlo (SGGMC) [23]. This interface aims to enable users

Module	Content
`adaption.py`	**Algorithms:** RMSProp [32], online covariance estimation, Fisher Information estimation [33]
`integrator.py`	**Process simulators:** OBABO [23], time-reversible leapfrog [22], leapfrog with friction [10], Langevin diffusion [9]
`scheduler.py`	**Schedules:** adaptive step size [8], polynomial step size with optimal decay [34], constant temperature, initial burn-in, random thinning
`potential.py`	**Potentials:** stochastic potential, true potential
`data.py`	**Data sources:** numpy/JAX arrays, TensorFlow dataset, HDF5 **Data batching:** mini-batching (drawing / shuffling / shuffling in epochs), batched mapping across full dataset
`io.py`	**Output formats:** numpy/JAX arrays, JSON, HDF5
`solvers.py`	*Lower-level interface to solvers in* `alias.py`

Table 2: Overview of algorithmic building blocks in each *JaxSGMC module.*

with an existing dataset and a JAX model to easily switch from stochastic optimizers to SG-MCMC samplers, while still providing flexibility in the dataset format and stochastic potential evaluation strategy.

A second API level is inspired by the stochastic optimization library Optax [31]. It allows more advanced users to combine SG-MCMC building blocks to create custom samplers tailored to the individual problem. Table 2 gives an overview of the currently implemented algorithmic building blocks and supported data formats. In addition, the implemented DataLoaders enable end-to-end jit-compilation of learning algorithms for maximum computational efficiency, even beyond the scope of SG-MCMC.

3. Illustrative Examples

We provide two ML examples, each illustrating a different use-case of *JaxSGMC*: building a custom sampler in a linear regression problem and using a pre-built sampler in an image classification problem. The interested reader may refer to the examples in the GitHub repository for more details.

3.1. Linear Regression

Many of the functionalities of *JaxSGMC* can be introduced with a simple linear regression model. Dataset arrays can be passed as keyword arguments to a DataLoader (`data.py`). The DataLoader stores the dataset on the host (CPU) and can orchestrate sending mini-batches to the device (e.g. GPU) as they are requested (listing 1).

```python
from jax_sgmc.data.numpy_loader import NumpyDataLoader
from jax_sgmc.data import random_reference_data

data_loader = NumpyDataLoader(x=training_data_x,
                              y=training_data_y)

data_fn = random_reference_data(data_loader,
                                mb_size=batch_size,
                                cached_batches_count=100)
```

Listing 1: Loading a dataset with *JaxSGMC*.

The next step is to define the linear model with weights $\mathbf{w}$, as well as log-likelihood and log-prior. The log-likelihood follows from the assumption that the data includes Gaussian-distributed noise with mean 0 and a (learned) homoscedastic standard deviation σ. The log-prior consists of an (improper) uniform distribution for $\mathbf{w}$ and an exponential distribution for the σ parameter. Log-likelihood and log-prior define the (mini-batch) potential (`potential.py`, eq. (4), listing 2).

```python
from jax_sgmc import potential

def model(sample, observations):
    weights = sample["w"]
    predictors = observations["x"]
    return jnp.dot(predictors, weights)

def log_likelihood(sample, observations):
    sigma = jnp.exp(sample["log_sigma"])
    y = observations["y"]
    y_pred = model(sample, observations)
    return jax.scipy.stats.norm.logpdf(y - y_pred, loc=0, scale=sigma)

def log_prior(sample):
    return 1 / jnp.exp(sample["log_sigma"])

potential_fn = potential.minibatch_potential(prior=log_prior,
                                             likelihood=log_likelihood)
```

Listing 2: Defining the stochastic potential function from the log-likelihood and log-prior.

The MemoryCollector (`io.py`) stores the sampled models in the host's working memory. We implement the RMSProp [32] preconditioned Stochastic Gradient Langevin Dynamics (pSGLD) method [11], which can be defined using RMSProp from the `adaption.py` module, a Langevin diffusion simulator (`integrator.py`) and a solver that accepts each sample unconditionally (`solver.py`). The schedulers in the `scheduler.py` module operate independently from the solver and manage the stepsize, burn-in and thinning along the Markov chain. The combination of these building blocks to obtain the pSGLD sampler is shown in listing 3.

```
from jax_sgmc import io, adaption, integrator, solver
from jax_sgmc.scheduler import polynomial_step_size_first_last,
    initial_burn_in, random_thinning, init_scheduler

my_data_collector = io.MemoryCollector()
save_fn = io.save(data_collector=my_data_collector)

rms_prop_adaption = adaption.rms_prop()

ld_integrator = integrator.langevin_diffusion(potential_fn=potential_fn,
                                              batch_fn=data_fn,
                                              adaption=rms_prop_adaption)

rms_prop_solver = solver.sgmc(ld_integrator)


#Initialize the solver by providing initial values for the latent variables
init_sample = {"log_sigma": jnp.array(0.0), "w": jnp.zeros(N)}

init_state = rms_prop_solver[0](init_sample)

step_size_schedule = polynomial_step_size_first_last(first=0.05,
                                                     last=0.001,
                                                     gamma=0.33)
burn_in_schedule = initial_burn_in(2000)
thinning_schedule = random_thinning(step_size_schedule=step_size_schedule,
                                    burn_in_schedule=burn_in_schedule,
                                    selections=1000)

schedule = init_scheduler(step_size=step_size_schedule,
                          burn_in=burn_in_schedule,
                          thinning=thinning_schedule)

mcmc = solver.mcmc(solver=rms_prop_solver,
                   scheduler=schedule,
                   saving=save_fn)
```

Listing 3: Building the preconditioned Stochastic Gradient Langevin Dynamics sampler from its building blocks.

Now the SG-MCMC sampling procedure can be performed. Afterwards, the saved samples can be accessed for postprocessing (listing 4).

```
# Take the result of the first chain
results = mcmc(init_state, iterations=10000)[0]

print(f"Collected {results['sample_count']} samples")

sigma_rms = onp.exp(results["samples"]["variables"]["log_sigma"]
w_rms = results["samples"]["variables"]["w"]
```

Listing 4: Sampling and accessing the results.

We visualize the sampled parameters and compare the resulting distribution to a gold-standard Hamiltonian Monte Carlo scheme implemented in the NumPyro library [35, 16] (fig. 2). The obtained distributions of both methods agree reasonably well, in line with expectations.

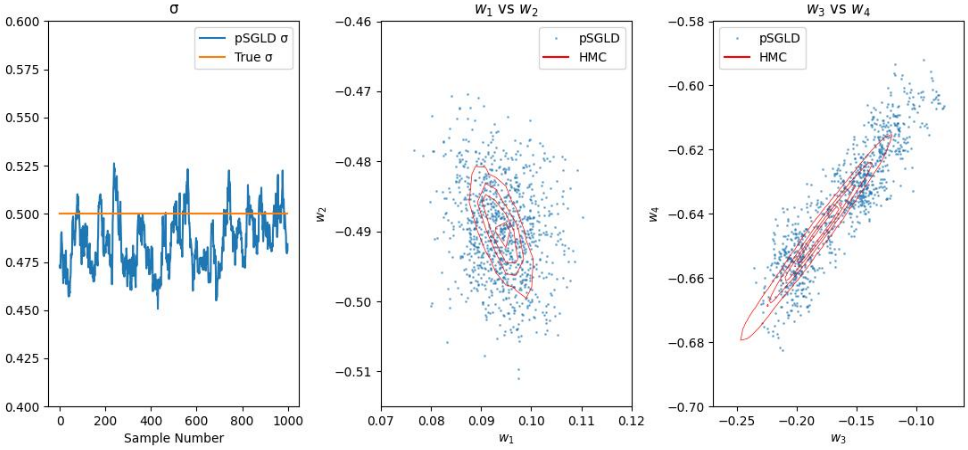


Figure 2: Left: Sampled standard deviation σ parameters (blue) compared to the data-generating value (orange). Middle and right: First and second (Middle) and third and fourth (right) components of weights w_i sampled with the pSGLD scheme (blue scatter plot) compared to contour plots of Gaussians obtained from the Hamiltonian Monte Carlo (HMC) method (red) implemented in NumPyro [35, 16].

3.2. Image Classification on CIFAR-10

Next, we provide an example more typical for recent deep learning models. In particular, we consider an image classification task with the CIFAR-10 dataset [26], which we split into a training, validation and test set containing 50000, 5000 and 5000 images, respectively. For the architecture of the NN, we use the 2.1 million parameter Haiku [36] implementation of MobileNet version 1 [37] without batch normalization. Given that the MobileNet architecture shows superior performance with larger images, we resized the images from 32x32 to 112x112 pixels using bilinear interpolation. The log-likelihood for a multiclass classification problem corresponds to the negative cross entropy. As prior distribution over NN weights and biases **w**, we select a Gaussian centered at 0 with standard deviation of 10. With these components, the potential function can be defined (listing 5).

```
import haiku as hk, optax, tree_math
from jax import tree_map
from functools import partial
from jax_sgmc import potential

def init_mobilenet():
    @hk.transform
    def mobilenetv1(batch, is_training=True):
        images = batch["image"].astype(jnp.float32)
        mobilenet = hk.nets.MobileNetV1(num_classes=num_classes,
                                        use_bn=False)
        logits = mobilenet(images, is_training=is_training)
        return logits
    return mobilenetv1.init, mobilenetv1.apply

init_mobilenet, apply_mobilenet = init_mobilenet()

def log_likelihood(sample, observations):
    logits = apply_mobilenet(sample["w"], None, observations)
    log_likelihood = -optax.softmax_cross_entropy_with_integer_labels(
        logits, observations["label"])
    return log_likelihood

def log_gaussian_prior(sample):
    gaussian = partial(jscipy.stats.norm.logpdf, loc=0, scale=10)
    priors = tree_map(gaussian, sample["w"])
    return tree_math.Vector(priors).sum()

potential_fn = potential.minibatch_potential(prior=log_gaussian_prior,
                                             likelihood=log_likelihood,
                                             is_batched=True,
                                             strategy='vmap')
```

Listing 5: Creating a MobileNet version 1 [37] NN model using Haiku [36] and defining log-likelihood, log-prior, and the (mini-batch) potential functions.

We use a pSGLD sampler with RMSProp preconditioner [11], which can be set up with the ready-to-use sampler interface of the `alias.py` module (listing 6). To initialize the sampler, the potential function, the DataLoader, and a set of hyperparameters need to be passed. We cache 10 batches of data in the device memory and set the batch size to 256 images. The learning rate is initially set to 0.001 and controlled by a polynomial step size scheduler.

```
from jax_sgmc import alias

sampler = alias.sgld(potential_fn=potential_fn,
                     data_loader=train_loader,
                     cache_size=cached_batches,
                     batch_size=batch_size,
                     first_step_size=lr_first,
                     last_step_size=lr_last,
                     burn_in=burn_in_size,
                     accepted_samples=accepted_samples,
                     rms_prop=True,
                     progress_bar=True)

results = sampler(sample, iterations=39000)
results = results[0]['samples']['variables']
```

Listing 6: Defining a SG-MCMC sampler via the `alias.py` API with subsequent sampling.

We sample for 39000 iterations - corresponding to 200 epochs - and retain 20 NN models via random thinning after a burn-in period of 35100 iterations. This results in a training accuracy of 65.25%, a validation accuracy of 55.72%

and a test accuracy of 57.32%, which is evaluated via soft voting of the ensemble [38].

We validate this result by comparing it to a deterministic model with the same model architecture and hyperparameters, and find a comparable performance. Furthermore, the runtimes of the SG-MCMC sampling and the stochastic optimization are comparable.

An advantage of using SG-MCMC is that the distribution of predictions from the sampled models can readily be used for UQ. As an example, we take five random images from the testset and visualize the distribution of the logits of each class in a box plot (fig. 3).

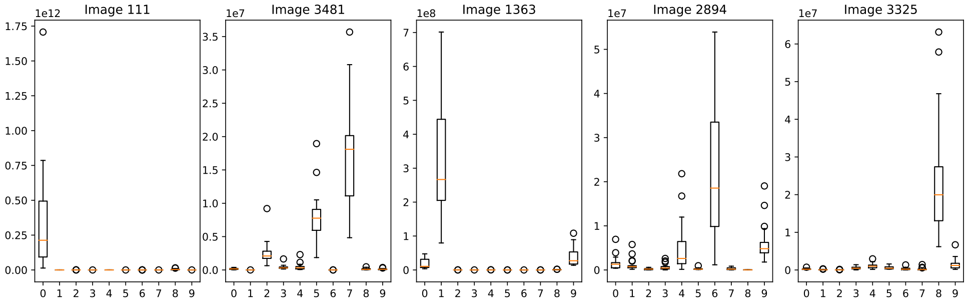


Figure 3: Obtained distributions of logits for five randomly selected images from the testset visualized as box plots. The true labels for the images in the order above are: 0, 7, 1, 6, 6.

Finally, we can assess the quality of the obtained uncertainty estimates by computing the testset accuracy for images, where the certainty of the prediction exceeds a specific threshold. We employ a hard voting-based [38] estimate of the prediction certainty, i.e. the percentage of all sampled models that predicted the majority class. As expected, the accuracy increases with increasing prediction certainty (table 3). However, for higher certainty thresholds, the obtained models are overconfident.

certainty	$\geq 50\%$	$\geq 60\%$	$\geq 70\%$	$\geq 80\%$	$\geq 90\%$	$= 100\%$
validation accuracy	58.89	61.72	64.61	67.60	71.64	77.95
test accuracy	60.11	63.06	66.59	70.20	74.65	80.81

Table 3: Accuracy depending on the certainty of the ensemble.

In this example, we opted for pSGLD [11], a comparatively simple SG-MCMC scheme. The accuracy and the quality of UQ could probably be increased by leveraging more advanced SG-MCMC components provided by *JaxSGMC*, including running multiple Markov chains [39]. However, this is beyond the scope of this illustrative example.

4. Impact and Conclusion

Trustworthy predictions via uncertainty-aware ML [40] and increasing data efficiency via active learning [41] represent highly promising avenues in deep learning. However, the effectiveness of these approaches critically depends on the quality of UQ estimates. To this end, it is insufficient to sample only a single NN posterior mode [42], e.g. when using Stochastic Variation Inference [17] or Dropout Monte Carlo [43]. While the popular Deep Ensemble [44, 45] scheme captures different posterior modes, it neglects the uncertainty contribution from the volume of the posterior. In contrast, SG-MCMC, especially when using multiple Markov chains, can sample multiple modes as well as the volume of the posterior.

Despite this theoretical advantage, SG-MCMC schemes are still underused in practice, in part due to a lack of easy-to-use libraries that implement state-of-the-art SG-MCMC samplers. *JaxSGMC* simplifies switching from stochastic optimization to Bayesian sampling by providing a common API (`alias.py`) for SG-MCMC samplers, which can replace stochastic optimizers [31] without modifications to the JAX NN model. Hence, *JaxSGMC* makes recently developed SG-MCMC samplers available to a broader user base. Additionally, by building custom samplers, the SG-MCMC schemes can be tailored to the ML problem at hand.

JaxSGMC is an domain-independent library. As such, we selected classical ML benchmark problems for the examples presented in this paper, but the provided code can also be used for other applications such as physics-based modeling. A recent example is the training of NN potentials [46], where the jit-compatible DataLoaders of *JaxSGMC* are used to improve computational performance and simplify implementation by integrating data loading into the jit-compiled parameter update function. Furthermore, the pre-implemented SG-MCMC samplers of *JaxSGMC* have been used to switch from stochastic optimization to Bayesian inference for cases of molecular modeling with classical [47] and NN potentials [39]. These studies have shown that pSGLD does not yet fully exploit the theoretical advantage of additional exploration of the posterior volume [39]. Thus, further research into SG-MCMC samplers with more sophisticated posterior exploration capabilities is required, which can be accelerated with *JaxSGMC*. We envision that *JaxSGMC* will promote uncertainty-aware ML and active learning applications in physical modeling and beyond.

5. Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

S.T. acknowledges financial support from the Munich Data Science Institute Seed Fund.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv preprint arXiv:1810.04805 (2018).

[2] S. Grigorescu, B. Trasnea, T. Cocias, G. Macesanu, A survey of deep learning techniques for autonomous driving, J. Field Robot. 37 (3) (2020) 362–386.

[3] R. Miotto, F. Wang, S. Wang, X. Jiang, J. T. Dudley, Deep learning for healthcare: review, opportunities and challenges, Brief. Bioinformatics 19 (6) (2018) 1236–1246.

[4] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[5] F. Noé, A. Tkatchenko, K. R. Müller, C. Clementi, Machine Learning for Molecular Simulation, Annu. Rev. Phys. Chem. 71 (2020) 361–390.

[6] S. Thaler, J. Zavadlav, Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting, Nat. Commun. 12 (2021) 6884.

[7] R. M. Neal, Handbook of Markov Chain Monte Carlo, 1st Edition, Chapman and Hall/CRC, New York, USA, 2011, Ch. MCMC using Hamiltonian Dynamics, pp. 139–188.

[8] M. D. Hoffman, A. Gelman, The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, J. Mach. Learn. Res. 15 (2014) 1593–1623.

[9] M. Welling, Y. W. Teh, Bayesian learning via stochastic gradient Langevin dynamics, in: Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, Jun. 28 – Jul. 2, 2011, pp. 681–688.

[10] T. Chen, E. Fox, C. Guestrin, Stochastic gradient Hamiltonian Monte Carlo, in: Proceedings of the 31st International Conference on Machine Learning, Beijing, China, Jun. 21–26, 2014, pp. 1683–1691.

[11] C. Li, C. Chen, D. E. Carlson, L. Carin, Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, February 12–17, 2016, pp. 1788–1794.

[12] C. Nemeth, P. Fearnhead, Stochastic gradient Markov chain Monte Carlo, J. Am. Stat. Assoc. 116 (533) (2021) 433–450.

[13] G. Lamb, B. Paige, Bayesian Graph Neural Networks for Molecular Property Prediction, in: Machine Learning for Molecules Workshop at NeurIPS, MIT Press, Online, Dec. 12, 2020.

[14] Z. Zou, X. Meng, A. F. Psaros, G. E. Karniadakis, NeuralUQ: A comprehensive library for uncertainty quantification in neural differential equations and operators, arXiv preprint arXiv:2208.11866 (2022).

[15] J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R. A. Saurous, Tensorflow distributions, arXiv preprint arXiv:1711.10604 (2017).

[16] E. Bingham, J. P. Chen, M. Jankowiak, F. Obermeyer, N. Pradhan, T. Karaletsos, R. Singh, P. A. Szerlip, P. Horsfall, N. D. Goodman, Pyro: Deep Universal Probabilistic Programming, J. Mach. Learn. Res. 20 (2019) 1–6.

[17] M. D. Hoffman, D. M. Blei, C. Wang, J. Paisley, Stochastic Variational Inference, J. Mach. Learn. Res. 14 (2013) 1303–1347.

[18] J. Baker, P. Fearnhead, E. B. Fox, C. Nemeth, sgmcmc: An R Package for Stochastic Gradient Markov Chain Monte Carlo, J. Stat. Softw. 91 (3) (2019) 1–27.

[19] A. K. Gupta, SG-MCMC (2016).
URL https://github.com/akshaykgupta/SG_MCMC

[20] J. Coullon, C. Nemeth, SGMCMCJax: a lightweight JAX library for stochastic gradient Markov chain Monte Carlo algorithms, J. Open Source Softw. 7 (72) (2022) 4113.

[21] W. Deng, Q. Feng, L. Gao, F. Liang, G. Lin, Non-convex Learning via Replica Exchange Stochastic Gradient MCMC, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, Online, Jul. 13–18, 2020, pp. 2474–2483.

[22] R. Zhang, A. F. Cooper, C. De Sa, AMAGOLD: Amortized Metropolis adjustment for efficient stochastic gradient MCMC, in: International Conference on Artificial Intelligence and Statistics, PMLR, Online, Aug. 26–28, 2020, pp. 2142–2152.

[23] A. Garriga-Alonso, V. Fortuin, Exact Langevin Dynamics with Stochastic Gradients, in: 3rd Symposium on Advances in Approximate Bayesian Inference, Online, Jan. – Feb., 2021.

[24] V. Gallego, D. R. Insua, Stochastic Gradient MCMC with Repulsive Forces, in: Bayesian Deep Learning Workshop at NeurIPS, MIT Press, Montreal, Canada, Dec. 7, 2018.

[25] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 770–778.

[26] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Tech. rep., University of Toronto (2009).

[27] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.

[28] Y.-A. Ma, T. Chen, E. B. Fox, A Complete Recipe for Stochastic Gradient MCMC, in: Advances in Neural Information Processing Systems, Vol. 28, MIT Press, Montreal, Canada, 2015, p. 2917–2925.

[29] S. Kim, Q. Song, F. Liang, Stochastic gradient Langevin dynamics with adaptive drifts, J. Stat. Comput. Simul. 92 (2) (2022) 318–336.

[30] R. Zhang, C. Li, J. Zhang, C. Chen, A. G. Wilson, Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning, in: 7th International Conference on Learning Representations, New Orleans, LA, USA, May 6–9, 2019.

[31] I. Babuschkin, K. Baumli, A. Bell, S. Bhupatiraju, J. Bruce, P. Buchlovsky, D. Budden, T. Cai, A. Clark, I. Danihelka, C. Fantacci, J. Godwin, C. Jones, R. Hemsley, T. Hennigan, M. Hessel, S. Hou,

S. Kapturowski, T. Keck, I. Kemaev, M. King, M. Kunesch, L. Martens, H. Merzic, V. Mikulik, T. Norman, J. Quan, G. Papamakarios, R. Ring, F. Ruiz, A. Sanchez, R. Schneider, E. Sezener, S. Spencer, S. Srinivasan, L. Wang, W. Stokowiec, F. Viola, The DeepMind JAX Ecosystem (2020).
URL `http://github.com/deepmind`

[32] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA: Neural networks for machine learning 4 (2) (2012) 26–31.

[33] S. Ahn, A. Korattikara, M. Welling, Bayesian Posterior Sampling via Stochastic Gradient Fisher Scoring, in: Proceedings of the 29th International Conference on Machine Learning, Omnipress, Madison, WI, USA, Jun. 26 – Jul. 1, 2012, pp. 1771–1778.

[34] Y. W. Teh, A. H. Thiery, S. J. Vollmer, Consistency and Fluctuations For Stochastic Gradient Langevin Dynamics, J. Mach. Learn. Res. 17 (2016) 1–33.

[35] D. Phan, N. Pradhan, M. Jankowiak, Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro, in: Program Transformations for ML at NeurIPS, MIT Press, Vancouver, Canada, Dec. 14, 2019.

[36] T. Hennigan, T. Cai, T. Norman, I. Babuschkin, Haiku: Sonnet for JAX (2020).
URL `http://github.com/deepmind/dm-haiku`

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv preprint arXiv:1704.04861 (2017).

[38] J. Kim, S. Choi, Automated machine learning for soft voting in an ensemble of tree-based classifiers, in: International Workshop on Automatic Machine Learning at ICML, Stockholm, Sweden, Jul. 14, 2018.

[39] S. Thaler, G. Doehner, J. Zavadlav, Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls, J. Chem. Theory Comput. (2023).

[40] H. Wang, D.-Y. Yeung, A survey on bayesian deep learning, ACM Comput. Surv. 53 (5) (2020) 1–37.

[41] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, X. Wang, A survey of deep active learning, ACM Comput. Surv. 54 (9) (2021) 1–40.

[42] A. G. Wilson, P. Izmailov, Bayesian Deep Learning and a Probabilistic Perspective of Generalization, in: Advances in Neural Information Processing Systems, Vol. 33, Online, Dec. 6–12, 2020, pp. 4697–4708.

[43] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.

[44] L. Hansen, P. Salamon, Neural Network Ensembles, IEEE Trans. Pattern Anal. Machine Intell. 12 (10) (1990) 993–1001.

[45] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in Neural Information Processing Systems, Vol. 30, Long Beach, CA, USA, Dec. 4–9, 2017, pp. 6405–6416.

[46] S. Thaler, M. Stupp, J. Zavadlav, Deep coarse-grained potentials via relative entropy minimization, J. Chem. Phys. 157 (2022) 244103.

[47] S. Thaler, J. Zavadlav, Uncertainty Quantification for Molecular Models via Stochastic Gradient MCMC, in: 10th Vienna Conference on Mathematical Modelling, Vienna, Austria, Jul. 27–29, 2022, pp. 19–20.

3.3. Augmented Adaptive Resolution Scheme

This section presents a research article that improves the AT-CG interface in an AdResS simulation, which paves the way for leveraging more accurate AT and CG NN potentials in AdResS with reduced interface artefacts.

3.3.1. Back-mapping augmented adaptive resolution simulation

Thaler, S., Praprotnik, M. & Zavadlav, J. Back-mapping augmented adaptive resolution simulation. *J. Chem. Phys.* **153**, 164118 (2020), DOI: 10.1063/5.0025728.

Summary

AdResS enables accurate and at the same time computationally efficient MD simulations by concurrent application of AT and CG models in different regions of the simulation box. In the standard AdResS formulation, the Δ region is necessary to maintain numerical stability at the interface. When a molecule diffuses from the CG region into the AT region, AdResS inserts AT DOFs randomly. The Δ region enables a gradual increase of the contribution from the AT force field, avoiding numerical instability due to a possible overlap of AT DOFs with neighboring atoms. However, this comes at the cost of unphysical results within the Δ region as well as increased computational effort. Consequently, previous work proposed to shrink the size of the Δ region to 0, switching resolutions abruptly. To maintain numerical stability, force capping (FC) at the interface was required.

This article proposes to tackle the numerical instability issue at its root-cause by reinserting AT DOFs in a manner that respects the chemical environment of the molecule. To this end, we introduce the Energy Minimized AT (DOF) Insertion method (EMATI). Instead of inserting AT particles randomly, EMATI selects the AT configuration such that the potential energy of the AT molecule is minimal while obeying the COM constraint given by the CG model. The EMATI algorithm consists of the following steps: All molecules are screened whether they diffused from the CG into the AT region during the last time step. If AT DOFs need to be inserted, the positions of AT particles are optimized iteratively by gradient descent minimization of the potential energy. After each gradient descent update, the position of the molecule is shifted such that its COM coincides with the COM given by the CG model. Convergence of the energy minimization is achieved if the variance of the forces in a moving window falls below a threshold value. The AT configuration is accepted if the maximum force on the AT molecule is below a target value. Otherwise, the AT molecule is rotated randomly and the minimization is repeated with the goal of obtaining a better local minimum.

The paper introduces a metric called overlap severity to estimate for which systems EMATI is most beneficial. In particular, the severity of overlaps increases for less spherical molecules as single site CG models of solvents enforce a spherically symmetric minimum distance of the COM to the COM of other molecules. Consequently, liquid butane is chosen as a system with high overlap severity, which would yield numerical instability for an AdResS simulation without the Δ region and FC. By employing the EMATI algorithm, the simulation remains stable even without FC. Compared to AdResS with FC, the temperature artifact at the interface is smaller. Consequently, EMATI allows using standard thermostat friction values while FC requires a stronger thermostat to remove the heat introduced by large, capped forces from overlapping atoms.

Instead of modifying the dynamics of all molecules inside the Δ region, EMATI only modifies single molecules during the AT DOF insertion. Consequently, EMATI reduces the computational effort compared to standard AdREsS with the Δ region. Additionally, structural properties at the interface are not affected by EMATI. Hence, EMATI extends the applicability of direct-coupling AdResS to systems for which FC is inadequate.

CRediT author statement

Stephan Thaler: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing

Matej Praprotnik: Writing – review & editing

Julija Zavadlav: Conceptualization, Supervision, Writing – original draft, Writing – review & editing

Copyright notice

The Journal of Chemical Physics

ARTICLE

scitation.org/journal/jcp

Back-mapping augmented adaptive resolution simulation

Cite as: J. Chem. Phys. **153**, 164118 (2020); doi: 10.1063/5.0025728
Submitted: 18 August 2020 • Accepted: 9 October 2020 •
Published Online: 28 October 2020

View Online Export Citation CrossMark

S. Thaler,[1] M. Praprotnik,[2,a)] and J. Zavadlav[1,b)]

AFFILIATIONS

[1] Professorship of Multiscale Modeling of Fluid Materials, Department of Mechanical Engineering, Technical University of Munich, Munich, Germany
[2] Laboratory for Molecular Modeling, National Institute of Chemistry, SI-1001 Ljubljana, Slovenia

[a)] Also at: Department of Physics, Faculty of Mathematics and Physics, University of Ljubljana, SI-1000 Ljubljana, Slovenia.
[b)] Author to whom correspondence should be addressed: julija.zavadlav@tum.de

ABSTRACT

Concurrent multiscale techniques such as Adaptive Resolution Scheme (AdResS) can offer ample computational advantages over conventional atomistic (AT) molecular dynamics simulations. However, they typically rely on aphysical hybrid regions to maintain numerical stability when high-resolution degrees of freedom (DOFs) are randomly re-inserted at the resolution interface. We propose an Energy Minimized AT (DOF) Insertion (EMATI) method that uses an informed rather than random AT DOF insertion to tackle the root cause of the issue, i.e., overlapping AT potentials. EMATI enables us to directly couple AT and coarse-grained resolutions without any modifications of the interaction potentials. We exemplify AdResS-EMATI in a system of liquid butane and show that it yields improved structural and thermodynamic properties at the interface compared to competing AdResS approaches. Furthermore, our approach extends the applicability of the AdResS without a hybrid region to systems for which force capping is inadequate.

I. INTRODUCTION

Biomolecular processes are challenging for computational modeling as they involve a vast span of time and length scales as the macroscopic properties of interest emerge from a molecular origin. While coarse-grained (CG) representations can reach larger spatiotemporal scales,[1,2] these models lack the accuracy and detail of atomistic (AT) models. Several multiscale approaches aim at resolving these conflicting objectives including back-mapping methods,[3–7] resolution replica exchange methods,[8,9] and concurrent multiscale simulations. The latter approach incorporates an intriguing idea of a computational magnifying glass: preserving atomistic accuracy and detail around a region of interest while reducing the remainder of the system to its essential degrees of freedom (DOFs).[10] Prototypical applications are processes where atomistic details are of interest only in a localized region, e.g., binding processes[11] and interactions of antimicrobial peptides with lipid membranes.[12] The coupling of AT and CG representations can be utilized either with constant resolution methods[13–16] or adaptive resolution methods.[10,17–26]

In the Adaptive Resolution Scheme[17] (AdResS), the simulation domain is separated into an AT and a CG region. Particles can diffuse freely between both regions, changing their DOFs on the fly. To allow for a smooth change in the resolution of the transitioning particles, a hybrid (HY) region is introduced at the interface of the AT and CG regions. However, the inclusion of the HY region is computationally demanding as it requires the computation of forces from both AT and CG potentials.[27,28] Furthermore, structural properties deviate in the HY region even if they match in the AT and CG regions.[29] First attempts to overcome these drawbacks were made in a recent paper by Krekeler *et al.*,[28] where the AdResS was employed in the limiting case of no HY region and sizable computational speed-ups over the standard AdResS were reported.

However, direct coupling of different resolutions results in interface difficulties that are in the standard AdResS alleviated by the HY region. A central hindrance is due to potentially overlapping AT particles when CG sites migrate from the CG into the AT region.[28] The current implementations of the AdResS in GROMACS[28,30] and Espresso++[31] insert AT DOFs randomly. Due to this random scheme, atoms are sometimes inserted unphysically close to existing atoms in the AT region, resulting in fatally high forces from steric repulsion that ultimately make the simulation numerically unstable.

To avoid these fatally large forces, Krekeler *et al.*[28] simply capped forces above a specified threshold, even though the authors noted that a method yielding proper AT DOFs might be necessary in certain cases. Up to now, the method was only applied to small and/or rather spherical solvent molecules.[21,28,32] Spherical molecules do not tend to yield severe overlaps as a spherical CG potential can be a good approximation to the AT molecule.[25] By contrast, non-spherical AT molecules can extend significantly beyond the van der Waals (VdW) volume enforced by the CG potential as we discuss in this paper. For these systems, random placement of DOFs at the interface can be detrimental to the AdResS even with the HY region.[25] For example, in an AdResS simulation of alkane systems,[25] the Lennard-Jones (LJ) potential needed to be substituted by a soft-core potential in the HY region that gradually blends back to the original LJ potential toward the AT region. Such modifications for the sake of avoiding fatally large forces, however, alter the AT force field and hence also the properties close to the AT-CG interface.

In this paper, we propose the Energy Minimized AT (DOF) Insertion (EMATI) method that avoids fatally overlapping potentials by using an informed rather than random AT DOF insertion. We, therefore, tackle the root cause of the numerical instability instead of inserting DOFs randomly and *ad hoc* mitigating the consequences of occasional overlaps. To solve the problem of finding valid AT DOFs based on the center of mass (COM) and the surrounding chemical environment, we transfer methods from the closely related back-mapping multiscale approach[3–6] to the AdResS. We demonstrate that direct coupling of AT and CG resolutions without the HY region with AdResS-EMATI eliminates numerical instability in a system of liquid butane without requiring force capping or AT potential modifications. By contrast, simulations with force capping become numerically unstable because occasionally more energy is introduced into the system at the interface than can be dissipated by the thermostat for common friction values. Additionally, we showcase improved interface properties with our method compared to both the standard AdResS with the HY region (subsequently referred to as AdResS) and AdResS without the HY region using force capping[28] (subsequently referred to as AdResS-FC).

II. METHODS

A. Adaptive resolution simulation

The AdResS[17] divides the simulation domain in an AT, a CG, and a HY region. The total force $\mathbf{F}_\alpha$ acting on a molecule α is

$$\mathbf{F}_\alpha = \sum_{\beta\neq\alpha} w(\mathbf{R}_\alpha)w(\mathbf{R}_\beta)\mathbf{F}^{AT}_{\alpha\beta} + \sum_{\beta\neq\alpha}[1 - w(\mathbf{R}_\alpha)w(\mathbf{R}_\beta)]\mathbf{F}^{CG}_{\alpha\beta} + \mathbf{F}^{TD}_\alpha,$$
$$\text{with} \quad w(\mathbf{R}) = \begin{cases} 1 \text{ if } \mathbf{R} \in \text{AT} \\ 0 \text{ if } \mathbf{R} \in \text{CG} \\ 0 < w(\mathbf{R}) < 1 \text{ if } \mathbf{R} \in \text{HY}, \end{cases} \tag{1}$$

where $\mathbf{R}$ is a molecule's COM, $\mathbf{F}^{AT}_{\alpha\beta}$ and $\mathbf{F}^{CG}_{\alpha\beta}$ are forces acting between molecules α and β via the AT and CG force fields, respectively, and $w(\mathbf{R})$ is a smooth resolution weighting function. $\mathbf{F}^{TD}$ is a thermodynamic force, which compensates differences in chemical potentials between both resolutions[22,29,33] and is typically applied in a close neighborhood of the resolution interface.

Shrinking the size of the HY region to 0 (Fig. 1) conceptually transforms $w(\mathbf{R})$ into the Heaviside step function,[28] reducing its purpose to a switching function that sorts molecules into AT or CG resolution. In this case, Eq. (1) implies that two molecules with the same resolution interact via the force field of the respective resolution, whereas molecules with different resolutions interact via the CG force field. Such a coupling definition is reminiscent of constant resolution multiscale methods,[13–16] whose common feature is direct interaction of molecules at different resolutions. In particular, the virtual sites approach[14,34] models the AT-CG interaction via the unaltered CG force field, equivalently to the AdResS without the HY region, albeit not allowing molecules to change their resolution. Furthermore, without the HY region, the AdResS[17] becomes a Hamiltonian method.[24] As already mentioned, omitting the HY region can cause the simulation to become numerically unstable unless the AT DOFs are inserted in a proper way. Section II B describes the EMATI scheme.

B. EMATI scheme

The aim of the presented EMATI method is to insert AT DOFs at sensible locations such that no fatally large forces occur in an AdResS simulation without the HY region. For each molecule entering the AT region, EMATI therefore needs to propose valid AT DOFs based on neighboring atoms in the AT region and given a

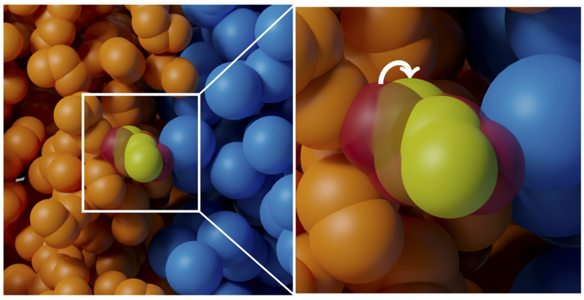


FIG. 1. Visualization of the interface of the AdResS for liquid butane without a HY region. The AT region resolves butane molecules atomistically (orange), while the CG region only resolves the COM of each molecule (blue). Random DOF insertion may yield potential overlaps (red molecule). EMATI proposes DOFs based on surrounding AT molecules avoiding severe overlaps (yellow molecule).

fixed COM position defined by the CG site. A well-known algorithm for on-the-fly insertion of molecules into dense fluids for open system simulations is USHER.[35] A generalization of USHER[36] achieves molecule insertion at prescribed potential energy values by simultaneously moving the COM and rigidly rotating the molecule using a steepest descent iterator. However, this approach relies on adjusting the COM of molecules to be inserted to find an appropriate insertion position. USHER is therefore not applicable to the AdResS, where the COM of the molecule is fixed. Furthermore, in its original formulation,[36] it does not generalize to non-rigid molecules. The insertion task in the AdResS resembles much more closely the central problem of the back-mapping multiscale approach,[3,4] where the AT detail needs to be back-inserted into a given skeleton of CG sites. Two of the most common ingredients of back-mapping algorithms are random initial insertion of atoms[5,6] and energy minimization to avoid overlapping AT potentials,[3,4,6,7] which serve as the main components of the EMATI method.

Our AdResS-EMATI approach tracks when a molecule has migrated from the CG into the AT region and applies EMATI (Fig. 2) to each of those CG sites. The first step of EMATI is to retrieve all neighboring AT atoms within a cutoff r_{cut} around the CG site position $\mathbf{R}$. This retrieval of neighbors is only required once as all particle positions are fixed, except for the molecule whose AT DOFs we are inserting (subsequently referred to as the central molecule). Note that r_{cut} can be chosen smaller than the cutoff of the AT potential to increase computational efficiency because a small number of nearest atoms dominate repulsive forces.

```
Algorithm 1: Energy Minimized AT DOF Insertion (EMATI)
input  : A CG site position $\mathbf{R}$ and initial AT positions $\{\mathbf{r}_i^0\}_{i=1}^N$
output: Valid AT positions $\{\mathbf{r}_i^{\text{final}}\}_{i=1}^N$
$\{\mathbf{r}_i^{\text{Nei}}\}_{i=1}^{N_{\text{Nei}}}$ = gather_AT_neighbors($\mathbf{R}$, $r_{\text{cut}}$);
for resets = 1 to $n_{\text{resets}}$ do
    while $\sigma > \sigma_{\text{target}}$ do
        $\{\mathbf{F}_i\}_{i=1}^N$ =sum_forces($\{\mathbf{r}_i^{\text{Nei}}\}_{i=1}^{N_{\text{Nei}}}$, $\{\mathbf{r}_i^k\}_{i=1}^N$);
        $F^{\text{COM}} = \|\sum_{i=1}^N \mathbf{F}_i\|$;
        $F^{\max} = \max(\{\|\mathbf{F}_i\|\}_{i=1}^N)$;
        $\sigma$ = compute_SD($F^{\text{COM}}, n_{\text{window}}$) ;
        foreach $\mathbf{r}_i^k$ in $\{\mathbf{r}_i^k\}_{i=1}^N$ do $\mathbf{r}_i^{k+1} = \mathbf{r}_i^k + \frac{\alpha \mathbf{F}_i}{m_i \ \max(1, F^{\max}/F^{\text{thresh}})}$;
        $\mathbf{R}^{\text{new}}$ = compute_COM($\{\mathbf{r}_i^k\}_{i=1}^N$) ;
        foreach $\mathbf{r}_i^k$ in $\{\mathbf{r}_i^k\}_{i=1}^N$ do $\mathbf{r}_i^{k+1} = \mathbf{r}_i^{k+1} - \mathbf{R}^{\text{new}} + \mathbf{R}$;
        $k = k + 1$;
    end
    if $F^{\text{COM}} < F^{\text{target}}$ then
        $\{\mathbf{r}_i^{\text{final}}\}_{i=1}^N = \{\mathbf{r}_i^k\}_{i=1}^N$;
        break;
    else if resets = $n_{\text{resets}}$ then
        $\{\mathbf{r}_i^{\text{final}}\}_{i=1}^N = \{\mathbf{r}_i^{\text{cur\_best}}\}_{i=1}^N$;
    else
        if $F^{\text{COM}} < F^{\text{cur\_best}}$ then
            $\{\mathbf{r}_i^{\text{cur\_best}}\}_{i=1}^N = \{\mathbf{r}_i^k\}_{i=1}^N$;
            $F^{\text{cur\_best}} = F^{\text{COM}}$;
        end
        $\{\mathbf{r}_i^0\}_{i=1}^N$ = random_reset($\{\mathbf{r}_i^k\}_{i=1}^N$) ;
    end
end
```

FIG. 2. Energy minimized AT DOF insertion (EMATI) algorithm.

The inner gradient-based constrained potential energy minimization loop is the core of EMATI. It serves to find a local minimum of the potential energy for the central molecule while satisfying the CG site position constraint. As the starting configuration for the energy minimization $\{\mathbf{r}_i^0\}_{i=1}^N$, where N is the number of AT particles per molecule, we choose the pseudo-random insertion provided by the AdResS implementation. The pseudo-randomness emerges from AdResS implementations that do not delete AT DOFs upon leaving the AT region. Inside the CG region, AT particles travel along with the CG site and simply re-appear at their current positions upon re-entering the AT region. We additionally implemented a truly random initial insertion but found no measurable effect on results. We opt for a steepest descent energy minimization scheme with step size α. The potential energy gradient is computed from forces acting on the central molecule from AT inter- and intra-molecular interactions. Displacing the current atom positions of the central molecule $\{\mathbf{r}_i^k\}_{i=1}^N$ along the steepest descent direction yields the configuration of the next iteration step,

$$\mathbf{r}_i^{k+1} = \mathbf{r}_i^k + \frac{\alpha \mathbf{F}_i}{m_i \ \max(1, F^{\max}/F^{\text{thresh}})}, \tag{2}$$

where m_i is the mass of the particle i and $\mathbf{F}_i$ is the force on particle i exerted by the neighboring atoms. To avoid overshooting local minima, we re-scale all forces if the magnitude of the maximum force $F^{\max} = \max(\{\|\mathbf{F}_i\|\}_{i=1}^N)$ exceeds a prescribed threshold force $F^{\max} > F^{\text{thresh}}$. This gradient re-scaling guarantees a constant maximum atomic displacement per iteration step. Updating AT positions according to Eq. (2) changes the COM of the central molecule to $\mathbf{R}^{\text{new}}$. To fulfill the "mapping condition,"[4] i.e., that the central molecule COM coincides with the CG site position, we move the central molecule back to the original $\mathbf{R}$ in each iteration, exactly fulfilling this constraint.

We define the convergence criterion of the constrained energy minimization based on the variance of the gradient: A local minimum is obtained when the standard deviation of the magnitude of the COM force $F^{\text{COM}} = \|\sum_{i=1}^N \mathbf{F}_i\|$ of the last n_{window} steps is smaller than the target σ_{target}. This variance-based convergence criterion is more suitable than directly dictating a maximum F^{COM} because it allows detection of local minima where further energy minimization would not yield a significantly better configuration, thus saving computational effort. A convergence criterion similar to Ref. 35 based on the potential could also be formulated. However, we opt for the above-mentioned criterion based on forces to avoid the additional computation of potential values. A convergence criterion based on the displacement of the central molecule might be a reasonable alternative.

The outer resetting loop checks if the obtained local minimum is acceptable, i.e., if $F^{\text{COM}} < F^{\text{target}}$. Otherwise, the molecule is reset randomly to yield new initial atom positions $\{\mathbf{r}_i^0\}_{i=1}^N$ to search for a better local minimum. We save $\{\mathbf{r}_i\}_{i=1}^N$ that yielded the smallest F^{COM} to continue the simulation with the best obtained configuration in case none of the obtained configurations yields a $F^{\text{COM}} < F^{\text{target}}$. In our simulations, we found this resetting scheme to be necessary for numerical stability as it avoids being stuck in unacceptable local minima, a known phenomenon in USHER[35,36] and back-mapping problems.[6] Theoretically, a series of n_{resets} unfortunate random resets that all lead to local minima with fatally large forces would be possible. However, increasing n_{resets} significantly reduces the probability of this rare event. Note that this situation did not occur in our implementation of EMATI, even for simulations of 10 ns length with more than 300 000 EMATI executions.

The presented method shows similarities to the rejection criterion in Monte Carlo Simulations in the sense that the unlikely configuration from random insertion is rejected and the higher likelihood configuration output from EMATI is accepted.[28]

C. Simulation setup

We choose liquid butane as an exemplary solvent to demonstrate the effectiveness of AdResS-EMATI. We use the GROMOS53A5[37] force field with flexible bond lengths to represent AT butane. The CG potential was derived via Iterative Boltzmann Inversion (IBI) including pressure correction[38] with the STOCK coarse-graining kit.[39] We performed a 10 ns AT reference simulation in a 5 × 5 × 5 nm^3 box to compute the target AT radial distribution function (RDF) and pressure. The obtained CG potential is shown in Fig. 3. Both AT and CG potentials are cut off at 1.4 nm.

We perform all simulations in the NVT ensemble in an orthorhombic simulation box with periodic boundary conditions using the Espresso++ 2.0.2.[31] package. We use a velocity Verlet time integration scheme with a 2 fs time step in accordance with the GROMOS[37] force field and a Langevin thermostat to maintain a target temperature of 323 K. We choose a friction coefficient of $\gamma = 1/\text{ps}$ unless stated otherwise.

For multiscale simulations, we use the box size of 20 × 5 × 5 nm^3 containing 3174 molecules, which corresponds to a density of 612.7 kg/m^3. For AdResS-EMATI and AdResS-FC, the simulation box is split along the x axis with an AT region of length 10 nm in the center and two connected CG regions (due to periodicity) of 5 nm (Fig. 1). For AdResS simulations, we choose an AT region width of 7.2 nm and a HY region length of 1.4 nm such that the explicit HY region coincides with the interface region of AdResS-EMATI, where AT molecules are influenced by AT and CG force fields. All simulations are run for 10 ns to generate the data.

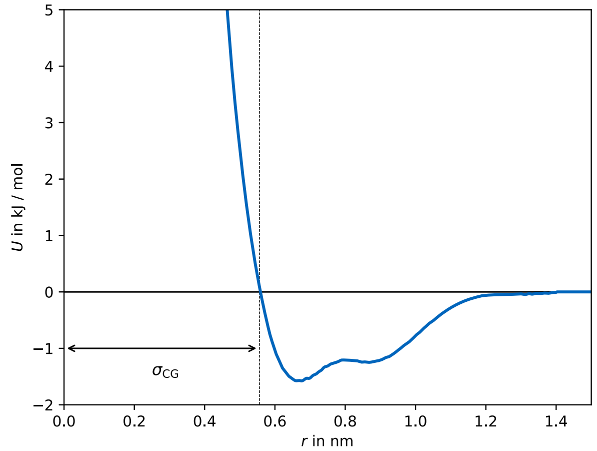


FIG. 3. CG potential of liquid butane obtained by iterative Boltzmann inversion. We defined σ_{CG} analogous to the LJ potential.

$\mathbf{F}^{\text{TD}}$ acts in a close neighborhood of the resolution interface and guarantees a uniform particle density distribution by construction. It is obtained iteratively before the production run $\mathbf{F}^{TD}_{i+1}(x) = \mathbf{F}^{TD}_{i}(x) - C\nabla\rho_i(x)$, where C is a convergence-driven, tunable constant.[29,40] We performed 30 iterations of length 1 ns each to reach a converged $\mathbf{F}^{\text{TD}}$ for AdResS-EMATI and AdResS-FC, while 50 iterations were necessary for the AdResS. An iteration constant $C = 2.2 \cdot 10^{-3}$ (kJ m^3)/(mol kg) was selected.

To maintain numerical stability with the AdResS and AdResS-FC, we cap forces component-wise at $F^{\text{cap}} = 5000$ kJ/(mol nm). For the AdResS-EMATI, we track the migration of molecules across the resolution regions via the Heaviside switching function $w(\mathbf{R})$. The EMATI scheme is triggered for all molecules whose w switches from 0 to 1. The scheme is applied after the velocity Verlet position update that caused the resolution change but before the force re-computation, where valid AT DOFs are necessary. We implement EMATI with the following parameters: $\alpha = 28.125$ fs^2, $r_{\text{cut}} = r_{\text{cut,AT}}/2 = 0.7$ nm, $F^{\text{thresh}} = 2000$ kJ/(mol nm), $\sigma_{\text{target}} = 5$ kJ/(mol nm), $n_{\text{window}} = 5$, $F^{\text{target}} = 1000$ kJ/(mol nm), and $n_{\text{resets}} = 10$. The random resetting function rotates the central molecule rigidly around a randomly drawn axis by a random angle between 45° and 135°. This range of angles ensures a large rotation to avoid staying inside the same insufficient local minimum.

III. RESULTS AND DISCUSSION

A. AdResS-EMATI

We demonstrate that AdResS-EMATI fulfills three central requirements: AT and CG regions are in equilibrium, it reproduces structural properties of a reference AT simulation, and the resolution interface does not act as an artificial diffusion barrier. Figure 4 visualizes the normalized density profile (NDP) of AdResS-EMATI with $\mathbf{F}^{\text{TD}}$ as a function of the distance from the AdResS center d. For the employed $\mathbf{F}^{\text{TD}}$, see Fig. 8. A maximum error of less than 2.5% in the NDP confirms good convergence of $\mathbf{F}^{\text{TD}}$. This minor density deviation together with a close to homogeneous temperature profile (discussed below) shows that AT and CG regions are in equilibrium.

We analyze the quality of reproducing structural properties by computing the COM–COM RDF (Fig. 5). The RDF in the AT region matches the reference full AT RDF perfectly, and the very well fit of the RDF in the CG region confirms convergence of the IBI. Note that both AT and CG RDFs are computed based on structural data from their whole respective domain, without ignoring particles in a neighborhood around the resolution interface. Shrinking the HY region to 0, therefore, significantly increases the size of the domain usable for structural analyses. For a more detailed discussion of the structural quality in the interface region, see Fig. 9.

Figure 6 demonstrates the absence of an artificial diffusion barrier in AdResS-EMATI. Particles in the AT region up to 1 nm left of

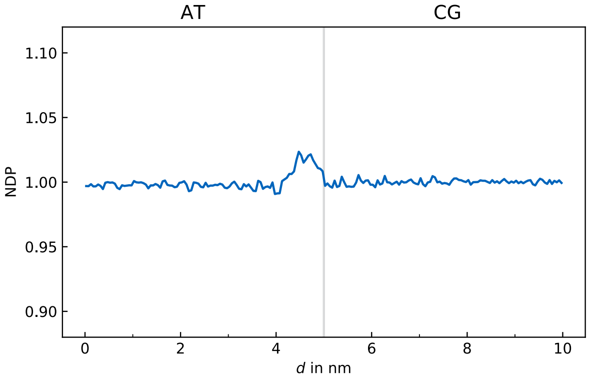


FIG. 4. Normalized density profile (NDP) of AdResS-EMATI with $\mathbf{F}^{\mathrm{TD}}$. The gray line visualizes the AT-CG interface.

the interface are marked at $t = 0$, and the probability density functions (PDFs) of marked particles over time are visualized. The larger PDF tails in the CG region reflect the higher diffusion constant in the CG region due to smoother dynamics from a lack of fluctuating forces that are missing AT DOFs.[41] The PDFs clearly show undisturbed Brownian motion, confirming that EMATI does not hinder diffusion.

B. Comparison to AdResS-FC and AdResS

AdResS-FC does not yield numerically stable simulations for $\gamma \le 5$ because, occasionally, more energy is introduced into the system at the interface from capped overlap forces than the thermostat can dissipate. This monotonically rising temperature makes the simulation numerically unstable. We therefore show results for $\gamma = 6$ in order to compare to AdResS-FC, even though such high values can affect the dynamics and viscosity of the system.[42,43] The common friction values for butane multiscale simulations in the literature are, e.g., $\gamma = 0.1$/ps[44] or $\gamma = 1$/ps.[14] Note that we additionally tested AdResS-FC with the MARTINI CG potential and $\gamma = 1$, resulting in the same type of fatal temperature rise. By contrast, AdResS-EMATI yields numerically stable simulations with both CG potentials even for $\gamma = 0.1$/ps, showing no evidence of these fatally rising temperatures.

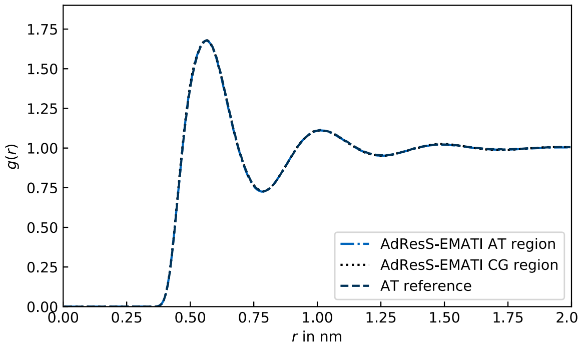


FIG. 5. COM–COM RDFs of AdResS-EMATI in the AT and CG region compared to a reference full AT simulation.

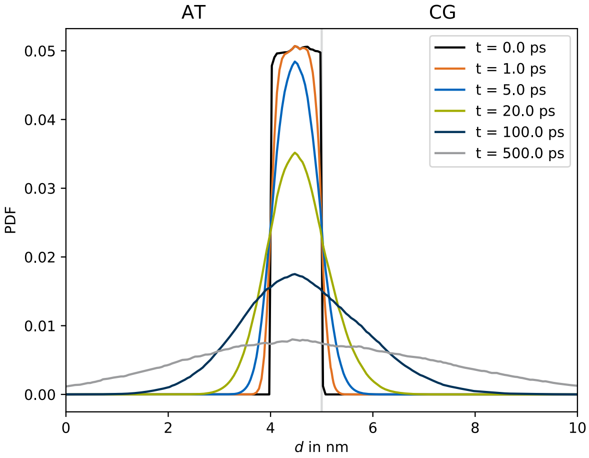


FIG. 6. Particle diffusion across the resolution interface for AdResS-EMATI. The AT-CG interface is visualized by a gray line. Particles in the AT region next to the interface are marked at $t = 0$, and the PDFs of marked particles over time are visualized.

A constant temperature profile is as important as a constant density profile for AT and CG regions to be in equilibrium.[22,29,33,45] We compare the normalized temperature profile (NTP) of AdResS-EMATI and AdResS-FC in Fig. 7 for $\gamma = 6$ to allow a comparison on equal terms. For the AdResS-EMATI, $\gamma = 6$ simulation, we reuse $\mathbf{F}^{\mathrm{TD}}$ derived for $\gamma = 1$. Despite the large friction coefficient of $\gamma = 6$, AdResS-FC yields a peak temperature deviation of 10.8%, while AdResS-EMATI only yields a maximum deviation of 0.6% for the same friction value. For $\gamma = 1$, AdResS-EMATI yields a temperature error of 3.6%, which is on the same order as the density deviation of 2.4%.

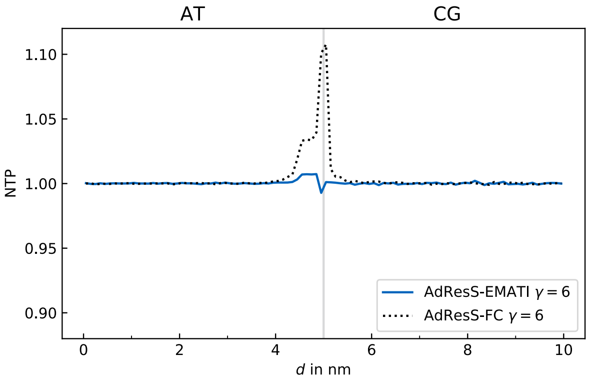


FIG. 7. Normalized temperature profile (NTP) of AdResS-EMATI and AdResS-FC. The temperature is computed based on CG DOFs in the CG region and AT DOFs in the AT region. The gray line visualizes the AT-CG interface.

As the temperature deviation is primarily controlled by the Langevin thermostat, a dedicated γ for the region around the resolution interface might improve AdResS-FC: The effect of a necessarily larger γ would be limited to this region, while the AT and CG regions could be simulated with a more desirable, smaller γ. Nonetheless, the above conclusions from our simulations with a constant γ still hold in the sense that AdResS-FC always requires a much larger γ in the interface region than AdResS-EMATI for the same maximum deviation threshold. This larger than desired impact of the thermostat might eventually alter structural and/or thermodynamic properties around the interface.

Figure 8 displays $\mathbf{F}^{\mathrm{TD}}$ and the NDP without $\mathbf{F}^{\mathrm{TD}}$ for AdResS-EMATI and AdResS-FC. NDP and $\mathbf{F}^{\mathrm{TD}}$ are very similar for both methods except for a very narrow region around the interface, hinting at a strictly local impact of EMATI and force capping. The most striking difference in NDP is a larger density well directly at the interface for AdResS-FC, presumingly due to large forces from overlaps (capped at F^{cap}) causing both overlapping molecules to quickly diffuse away from the interface. The same phenomenon emerges in $\mathbf{F}^{\mathrm{TD}}$ as it needs to compensate for this density well, resulting in larger force extrema close to the interface.

While achieving a constant NTP is not problematic in the AdResS, the smoothing inside the HY region comes at the cost of structural deviations in this region. Figure 9 compares the COM–COM RDF of AdResS-EMATI at the region that coincides with the HY region of the AdResS. The agreement with the AT reference RDF is very good for AdResS-EMATI (and for AdResS-FC, not shown). By contrast, the AdResS yields an RDF in the HY region that significantly deviates from the AT reference.

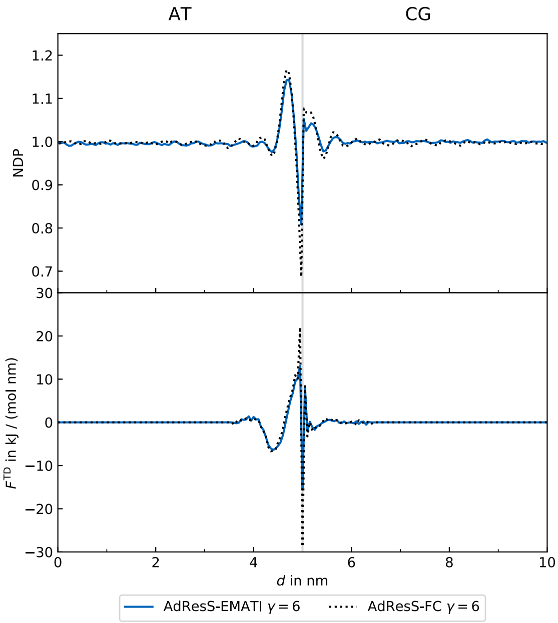


FIG. 8. Normalized density profile (NDP) and $\mathbf{F}^{\mathrm{TD}}$ of AdResS-EMATI and AdResS-FC. The gray line visualizes the AT-CG interface.

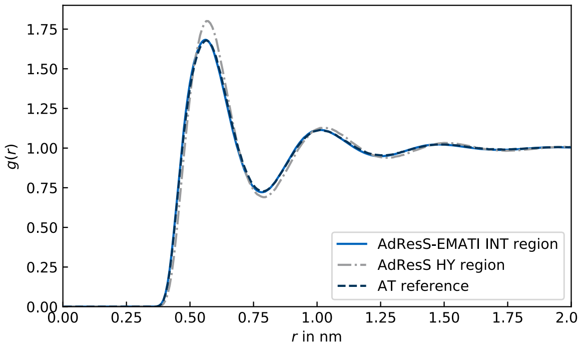


FIG. 9. COM–COM RDFs of AdResS-EMATI in the interface (INT) region compared to the AdResS in the HY region and to the reference full AT simulation.

Our results demonstrate that AdResS-EMATI yields well matching thermodynamic as well as structural properties at the interface. Matching additional properties beyond density and temperature at the interface increases numerical accuracy in AT and CG regions by reducing coupling artifacts.[29] For example, matching the COM–COM RDF in the interface implies that the PDF in the AT region matches the PDF of a full AT simulation at least up to second order,[22,29] while even third-order accuracy has been shown empirically.[22] This characteristic of reduced AT-CG coupling artifacts makes AdResS-EMATI a prime choice in a computational magnifying glass setting where the CG region serves to deliver a coarse but informative representation of the system.

C. Estimation of overlap severity

We propose a coarse measure based on the solvent geometry and both the AT and CG force fields to *a priori* estimate the possible severity of overlapping potentials at the interface. This measure might be helpful in identifying systems for which augmenting the AdResS with EMATI (or some other similar back-mapping method) is necessary. Note that we neglect electrostatic forces in this discussion as steric repulsion forces from the LJ potential are dominant for small atom distances.

The superposition of the VdW volumes of all likely occurring AT configurations that correspond to a given COM position yields a sphere of radius r_{AT} that determines the maximum extent of inserted AT molecules [Fig. 10(a)]. We approximate r_{AT} as a sum of the VdW radius of the outermost AT atom $\sigma_{\mathrm{AT}}/2$ and its distance from the COM position in the equilibrium configuration[46] r_{COM}, i.e., $r_{\mathrm{AT}} \approx r_{\mathrm{COM}} + \sigma_{\mathrm{AT}}/2$. The difference between the radius of this AT sphere and the VdW radius of the CG potential ($r_\delta = r_{\mathrm{AT}} - \sigma_{\mathrm{CG}}/2$) determines the likelihood and severity of overlaps for random insertion, where we estimate σ_{CG} analogously to the LJ potential (Fig. 3).

We define a measure for overlap severity v by considering the worst possible scenario [Fig. 10(b)]. Suppose two neighboring CG

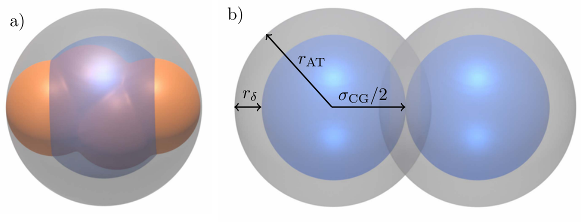


FIG. 10. Sketch of the severity of overlap estimation. Panel (a) visualizes the origin of the sphere of maximum AT extent for butane. The VdW volume of a sample butane molecule in orange significantly extends beyond the VdW volume of the CG potential in blue. Superposition of all possible AT configurations yields the gray sphere of maximum AT extent. Panel (b) sketches the configuration of the worst case overlap estimation with two molecules being σ_{CG} apart. This configuration yields overlapping AT spheres shaded dark gray. If both AT molecules happen to point exactly toward each other, their most outward VdW spheres overlap by $2r_\delta$.

sites are separated by a distance σ_{CG}. Coincidentally, the AT DOFs are inserted such that the outermost AT atoms of the two CG sites are facing each other, i.e., they are at the minimal mutual distance equal to $r = \sigma_{\mathrm{CG}} - 2r_{\mathrm{COM}} = \sigma_{\mathrm{AT}} - 2r_\delta$. The overlap severity ν is

$$\nu = \frac{F^{\mathrm{AT}}\,|_{r=\sigma_{\mathrm{AT}}-2r_\delta}}{F^{\mathrm{AT}}\,|_{r=\sigma_{\mathrm{AT}}}}, \tag{3}$$

where $F^{\mathrm{AT}} = \|\mathbf{F}^{\mathrm{AT}}\|$ is the force magnitude computed from the AT potential of the molecule's outermost AT atom.

In the case of SPC water, where ν is not excessively large ($r_\delta = 0.026$ nm, $\nu = 17.1$), even this worst case insertion scenario does not create fatally large forces. Therefore, force capping is not necessary for SPC water in the AdResS without the HY region,[47] which we confirmed with test simulations. Larger, non-spherical molecules are more challenging for the AdResS, given that a spherical CG potential is not a good approximation[25] and r_{AT} increases with the maximum distance of the outermost atoms from the COM position r_{COM}. Thus, alkane chains, including butane,[44] require a method to handle overlapping AT potentials, even with the HY region.[25] The issue is even worse without the HY region with ν of butane ($r_\delta = 0.10$ nm, $\nu = 5.4 \cdot 10^4$) being more than three orders of magnitude larger than in the case of SPC water.

Any back-mapping method that augments the AdResS implicitly relies on the assumption that the COM of the central molecule is sensible, given the surrounding AT neighbors. Examples that clearly violate this assumption are the recent applications of the AdResS without the HY region modeling CG molecules as ideal gas particles:[21,32] Ideal gas particles in the CG region do not enforce a minimum distance between CG molecules ($\sigma_{\mathrm{CG}} = 0$); hence, the COM of the central molecule can be arbitrarily close to neighboring AT atoms. Consequently, there does not always exist an AT configuration of the central molecule that is consistent with the given COM and simultaneously yields non-fatal forces, necessitating force capping, even for water.[32]

The main drawback of EMATI is its reliance on energy minimization. Energy minimization is prone to yield over-stabilized structures in back-mapping problems such that additional molecular dynamics steps are often required to obtain structures at a target state point.[4] However, we did not experience large over-stabilization effects in our simulations. If, however, application of EMATI would result in unacceptably large over-stabilization, increasing σ_{target} should shift the potential energy distribution toward larger energies counteracting this effect.

Energy minimization steps also increase computational effort, generating a computational overhead over AdResS-FC. The extra computational effort is proportional to the number of CG sites migrating into the AT region, hence scaling with the resolution interface area. In our simulations, EMATI is applied approximately once every 17 time steps. Computation per migrated CG site, i.e., per EMATI execution, is limited to a small hemisphere of radius r_{cut} around the central molecule and is mainly controlled by the EMATI parameters r_{cut}, σ_{target}, F^{target}, and n_{resets}. On the other hand, AdResS-EMATI reduces the overhead due to the exclusion of the HY region. The related speed-up scales with the volume of the HY region and depends on the employed force fields. For example, Ref. 28 reported a speed-up of 1.4 for a system of two micelles in water. We refer the reader to Refs. 27 and 28 for a detailed discussion of the computational implications of the HY region. In our numerical experiments, the overhead of EMATI, with the parameters given in Sec. II C, was approximately the same as the overhead from the HY region. For this proof of concept study, we neither optimized EMATI nor its parameters for numerical efficiency. As a test, we also changed EMATI's parameters toward numerical efficiency (e.g., reducing r_{cut}) and found a significant reduction in overhead, hence outperforming the AdResS. The numerical efficiency could also be improved by considering more advanced methods, e.g., the Fast Inertial Relaxation Engine (FIRE).[48]

IV. CONCLUSION

In this work, we proposed EMATI, an interface DOF insertion method for the AdResS without the need of a HY region sandwiched in between the AT and CG regions. AdResS-EMATI is conceptually similar to the method introduced in Ref. 28. However, it largely extends the applicability of the direct AT/CG coupling. In particular, it enables the direct resolution coupling to systems with non-spherical molecules, e.g., butane, for which simple force capping does not suffice. Numerical stability in AdResS-EMATI is achieved without requiring force capping or changing the AT potential at the interface. AdResS-EMATI directly tackles the root cause of the numerical instability issue, i.e., overlapping AT potentials, by inserting AT DOFs based on the minimized interacting energy with surrounding atoms. We demonstrated the applicability of our method in a system of liquid butane, for which AdResS-FC results in a fatal temperature rise for common thermostat friction values. We further showcased reduced temperature artifacts over AdResS-FC while also removing the structural discrepancies observed for the standard AdResS in the HY region.

We chose the IBI CG potential to showcase excellent structural properties in the interface region (Fig. 9). However, there are many CG potentials that would work with AdResS-EMATI as well, e.g., a potential derived from the multiscale coarse-graining method[49–51] would be a reasonable alternative. We additionally implemented the MARTINI[52] CG force field and found the same numerical stability preserving properties of AdResS-EMATI, even though the RDFs of the MARTINI and AT force fields do not match. AdResS-EMATI is in principle applicable to polar and apolar solvents and arbitrary CG force fields, as long as the CG potential enforces a sufficiently large VdW volume such that an acceptable AT molecule configuration exists. It could also be applied to coarse-grained models of molecules with several CG particles, i.e., coarse-grained models of polymers. In this case, the calculation of the forces used for the EMATI scheme needs to also include the bonded interactions (bonds, angles, and dihedrals) between the atoms belonging to different CG sites in the AT region of the same molecule.

EMATI might also prove valuable to the standard AdResS with the HY region, e.g., for macromolecular systems,[25] as an alternative to the soft-core potential substitution method. Augmentation of the AdResS without the HY region by a back-mapping method represents one step toward its applicability as a computational magnifying glass by improving numerical stability and reducing artifacts from overlaps in the interface. To improve the method further, replacing the energy minimization in EMATI by a more advanced back-mapping scheme[53–57] might be a next step worth investigating.

ACKNOWLEDGMENTS

M.P. acknowledges financial support through Grant No. P1-0002 from the Slovenian Research Agency.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] W. G. Noid, "Perspective: Coarse-grained models for biomolecular systems," J. Chem. Phys. **139**, 090901 (2013).

[2] H. I. Ingólfsson, C. A. Lopez, J. J. Uusitalo, D. H. de Jong, S. M. Gopal, X. Periole, and S. J. Marrink, "The power of coarse graining in biomolecular simulations," Wiley Interdiscip. Rev. Comput. Mol. Sci. **4**, 225–248 (2014).

[3] A. P. Heath, L. E. Kavraki, and C. Clementi, "From coarse-grain to all-atom: Toward multiscale analysis of protein landscapes," Proteins Struct. Funct. Genet. **68**, 646–661 (2007).

[4] C. Peter and K. Kremer, "Multiscale simulation of soft matter systems—From the atomistic to the coarse-grained level and back," Soft Matter **5**, 4357–4366 (2009).

[5] A. J. Rzepiela, L. V. Schäfer, N. Goga, H. Jelger Risselada, A. H. De Vries, and S. J. Marrink, "Software news and update reconstruction of atomistic details from coarse-grained structures," J. Comput. Chem. **31**, 1333–1343 (2010).

[6] T. A. Wassenaar, K. Pluhackova, R. A. Böckmann, S. J. Marrink, and D. P. Tieleman, "Going backward: A flexible geometric approach to reverse transformation from coarse grained to atomistic models," J. Chem. Theory Comput. **10**, 676–690 (2014).

[7] M. Shimizu and S. Takada, "Reconstruction of atomistic structures from coarse-grained models for protein-DNA complexes," J. Chem. Theory Comput. **14**, 1682–1694 (2018).

[8] E. Lyman, F. M. Ytreberg, and D. M. Zuckerman, "Resolution exchange simulation," Phys. Rev. Lett. **96**, 028105 (2006).

[9] P. Liu and G. A. Voth, "Smart resolution replica exchange: An efficient algorithm for exploring complex energy landscapes," J. Chem. Phys. **126**, 045106 (2007).

[10] M. Praprotnik, L. Delle Site, and K. Kremer, "Multiscale simulation of soft matter: From scale bridging to adaptive resolution," Annu. Rev. Phys. Chem. **59**, 545–571 (2008).

[11] Y. Shan, E. T. Kim, M. P. Eastwood, R. O. Dror, M. A. Seeliger, and D. E. Shaw, "How does a drug molecule find its target binding site?," J. Am. Chem. Soc. **133**, 9181–9183 (2011).

[12] Y. Wang, T. Zhao, D. Wei, E. Strandberg, A. S. Ulrich, and J. P. Ulmschneider, "How reliable are molecular dynamics simulations of membrane active antimicrobial peptides?," Biochim. Biophys. Acta **1838**, 2280–2288 (2014).

[13] Q. Shi, S. Izvekov, and G. A. Voth, "Mixed atomistic and coarse-grained molecular dynamics: Simulation of a membrane-bound ion channel," J. Phys. Chem. B **110**, 15045–15048 (2006).

[14] A. J. Rzepiela, M. Louhivuori, C. Peter, and S. J. Marrink, "Hybrid simulations: Combining atomistic and coarse-grained force fields using virtual sites," Phys. Chem. Chem. Phys. **13**, 10437–10448 (2011).

[15] P. Sokkar, S. M. Choi, and Y. M. Rhee, "Simple method for simulating the mixture of atomistic and coarse-grained molecular systems," J. Chem. Theory Comput. **9**, 3728–3739 (2013).

[16] A. B. Kuhn, S. M. Gopal, and L. V. Schäfer, "On using atomistic solvent layers in hybrid all-atom/coarse-grained molecular dynamics simulations," J. Chem. Theory Comput. **11**, 4460–4472 (2015).

[17] M. Praprotnik, L. Delle Site, and K. Kremer, "Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly," J. Chem. Phys. **123**, 224106 (2005); arXiv:0510223 [cond-mat].

[18] J. Zavadlav, M. N. Melo, S. J. Marrink, and M. Praprotnik, "Adaptive resolution simulation of an atomistic protein in martini water," J. Chem. Phys. **140**, 054114 (2014).

[19] J. Zavadlav, S. J. Marrink, and M. Praprotnik, "Multiscale simulation of protein hydration using the SWINGER dynamical clustering algorithm," J. Chem. Theory Comput. **14**, 1754–1761 (2018).

[20] J. Zavadlav, J. Sablić, R. Podgornik, and M. Praprotnik, "Open-boundary molecular dynamics of a DNA molecule in a hybrid explicit/implicit salt solution," Biophys. J. **114**, 2352–2362 (2018).

[21] J. Whittaker and L. Delle Site, "Investigation of the hydration shell of a membrane in an open system molecular dynamics simulation," Phys. Rev. Res. **1**, 033099 (2019).

[22] H. Wang, C. Hartmann, C. Schütte, and L. Delle Site, "Grand-canonical-like molecular-dynamics simulations by using an adaptive-resolution technique," Phys. Rev. X **3**, 011018 (2013).

[23] A. Agarwal, H. Wang, C. Schütte, and L. Delle Site, "Chemical potential of liquids and mixtures via adaptive resolution simulation," J. Chem. Phys. **141**, 034102 (2014); arXiv:1311.6982.

[24] R. Potestio, S. Fritsch, P. Español, R. Delgado-Buscalioni, K. Kremer, R. Everaers, and D. Donadio, "Hamiltonian adaptive resolution simulation for molecular liquids," Phys. Rev. Lett. **110**, 108301 (2013).

[25] J. H. Peters, R. Klein, and L. Delle Site, "Simulation of macromolecular liquids with the adaptive resolution molecular dynamics technique," Phys. Rev. E **94**, 023309 (2016).

[26] A. Chaimovich, C. Peter, and K. Kremer, "Relative resolution: A hybrid formalism for fluid mixtures," J. Chem. Phys. **143**, 243107 (2015); arXiv:1903.04755.

[27] C. Junghans, A. Agarwal, and L. Delle Site, "Computational efficiency and Amdahl's law for the adaptive resolution simulation technique," Comput. Phys. Commun. **215**, 20–25 (2017).

[28] C. Krekeler, A. Agarwal, C. Junghans, M. Praprotnik, and L. Delle Site, "Adaptive resolution molecular dynamics technique: Down to the essential," J. Chem. Phys. **149**, 024104 (2018); arXiv:1806.09870.

[29] H. Wang, C. Schütte, and L. Delle Site, "Adaptive resolution simulation (AdResS): A smooth thermodynamic and structural transition from atomistic to coarse grained resolution and *vice versa* in a grand canonical fashion," J. Chem. Theory Comput. **8**, 2878–2887 (2012).

[30] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, "GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers," SoftwareX **1-2**, 19–25 (2015).

[31] H. V. Guzman, N. Tretyakov, H. Kobayashi, A. C. Fogarty, K. Kreis, J. Krajniak, C. Junghans, K. Kremer, and T. Stuehn, "ESPResSo++ 2.0: Advanced methods for multiscale molecular simulation," Comput. Phys. Commun. **238**, 66–76 (2019); arXiv:1806.10841.

[32] L. Delle Site, C. Krekeler, J. Whittaker, A. Agarwal, R. Klein, and F. Höfling, "Molecular dynamics of open systems: Construction of a mean-field particle reservoir," Adv. Theory Simul. **2**, 1900014 (2019).

[33] S. Poblete, M. Praprotnik, K. Kremer, and L. Delle Site, "Coupling different levels of resolution in molecular simulations," J. Chem. Phys. **132**, 114101 (2010).

[34] Y. Liu, A. H. De Vries, J. Barnoud, W. Pezeshkian, J. Melcr, and S. J. Marrink, "Dual resolution membrane simulations using virtual sites," J. Phys. Chem. B **124**, 3944–3953 (2020).

[35] R. Delgado-Buscalioni and P. V. Coveney, "USHER: An algorithm for particle insertion in dense fluids," J. Chem. Phys. **119**, 978–987 (2003); arXiv:0303366 [cond-mat].

[36] G. De Fabritiis, R. Delgado-Buscalioni, and P. V. Coveney, "Energy controlled insertion of polar molecules in dense fluids," J. Chem. Phys. **121**, 12139 (2004).

[37] C. Oostenbrink, A. Villa, A. E. Mark, and W. F. Van Gunsteren, "A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6," J. Comput. Chem. **25**, 1656–1676 (2004).

[38] D. Reith, M. Pütz, and F. Müller-Plathe, "Deriving effective mesoscale potentials from atomistic simulations," J. Comput. Chem. **24**, 1624–1636 (2003).

[39] S. Bevc, C. Junghans, and M. Praprotnik, "STOCK: Structure mapper and online coarse-graining kit for molecular simulations," J. Comput. Chem. **36**, 467–477 (2015).

[40] S. Fritsch, S. Poblete, C. Junghans, G. Ciccotti, L. Delle Site, and K. Kremer, "Adaptive resolution molecular dynamics simulation through coupling to an internal particle reservoir," Phys. Rev. Lett. **108**, 170602 (2012); arXiv:1112.3151.

[41] S. Matysiak, C. Clementi, M. Praprotnik, K. Kremer, and L. Delle Site, "Modeling diffusive dynamics in adaptive resolution simulation of liquid water," J. Chem. Phys. **128**, 024503 (2008).

[42] K. Kremer and G. S. Grest, "Dynamics of entangled linear polymer melts: A molecular-dynamics simulation," J. Chem. Phys. **92**, 5057–5086 (1990).

[43] C. Junghans, M. Praprotnik, and K. Kremer, "Transport properties controlled by a thermostat: An extended dissipative particle dynamics thermostat," Soft Matter **4**, 156–161 (2008).

[44] J. Zavadlav, M. N. Melo, A. V. Cunha, A. H. de Vries, S. J. Marrink, and M. Praprotnik, "Adaptive resolution simulation of MARTINI solvents," J. Chem. Theory Comput. **10**, 2591–2598 (2014).

[45] M. Praprotnik, K. Kremer, and L. Delle Site, "Adaptive molecular resolution via a continuous change of the phase space dimensionality," Phys. Rev. E **75**, 017701 (2007).

[46] Allowing for flexible bonds and angles will increase r_{AT} in practice.

[47] B. Duenweg, J. Castagna, S. Chiacchiera, H. Kobayashi, and C. Krekeler (2018). "Meso- and multi-scale modelling E-CAM modules II," Zenodo. https://doi.org/10.5281/zenodo.1210075 (2018).

[48] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," Phys. Rev. Lett. **97**, 170201 (2006).

[49] S. Izvekov and G. A. Voth, "A multiscale coarse-graining method for biomolecular systems," J. Phys. Chem. B **109**, 2469–2473 (2005).

[50] W. G. Noid, J.-W. Chu, G. S. Ayton, V. Krishna, S. Izvekov, G. A. Voth, A. Das, and H. C. Andersen, "The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models," J. Chem. Phys. **128**, 244114 (2008).

[51] W. G. Noid, P. Liu, Y. Wang, J.-W. Chu, G. S. Ayton, S. Izvekov, H. C. Andersen, and G. A. Voth, "The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models," J. Chem. Phys. **128**, 244115 (2008).

[52] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. De Vries, "The MARTINI force field: Coarse grained model for biomolecular simulations," J. Phys. Chem. B **111**, 7812–7824 (2007).

[53] L.-J. Chen, H.-J. Qian, Z.-Y. Lu, Z.-S. Li, and C.-C. Sun, "An automatic coarse-graining and fine-graining simulation method: Application on polyethylene," J. Phys. Chem. B **110**, 24093–24100 (2006).

[54] J. Krajniak, Z. Zhang, S. Pandiyan, E. Nies, and G. Samaey, "Reverse mapping method for complex polymer systems," J. Comput. Chem. **39**, 648–664 (2018).

[55] J. Peng, C. Yuan, R. Ma, and Z. Zhang, "Backmapping from multiresolution coarse-grained models to atomic structures of large biomolecules by restrained molecular dynamics simulations using Bayesian inference," J. Chem. Theory Comput. **15**, 3344–3353 (2019).

[56] G. Zhang, A. Chazirakis, V. A. Harmandaris, T. Stuehn, K. C. Daoulas, and K. Kremer, "Hierarchical modelling of polystyrene melts: From soft blobs to atomistic resolution," Soft Matter **15**, 289–302 (2019).

[57] W. Li, C. Burkhart, P. Polińska, V. Harmandaris, and M. Doxastakis, "Backmapping coarse-grained macromolecules: An efficient and versatile machine learning approach," J. Chem. Phys. **153**, 041101 (2020).

4. Summary and Discussion

Section 4.1 summarizes the main results of the thesis. Subsequently, section 4.2 discusses important findings and provides an outlook on further research questions.

4.1. Summary

This thesis presents a body of work that promotes the more widespread adoption of NN potential-based MD simulations in support of decision-making in practice. First, the proposed DiffTRe algorithm facilitates training NN potential on experimental data. This represents a major step towards highly accurate NN potentials necessary for reliable in-silico analyses. DiffTRe also integrates well into existing ML frameworks as it only requires to encode the forward pass of the computation of MD observables, whereas the resulting gradient can be obtained via AD.

Second, the thesis demonstrated that NN potentials can be trained effectively via RE minimization. This CG NN potential training scheme proved to be more data efficient and less sensitive to the choice of prior potential. In contrast to FM, the RE objective function considers phase-space regions not included in the data set as improbable and assigns them a high potential energy if visited during training. In addition, RE minimization enables the correction of numerical errors in CG MD simulations and thus a speed-up due to larger simulation time steps without sacrificing accuracy.

Third, results showed that both M-pSGLD as well as the Deep Ensemble method enable reliable UQ of MD observables. To achieve this goal, it is central to sample multiple posterior modes and minimize difficult-to-quantify systematic uncertainties. However, while Bayesian UQ methods have the theoretical advantage of being able to sample the volume of the posterior, the employed pSGLD method was unable to capitalize on this. To address this, the *jax-sgmc* library aims to foster research on more effective SG-MCMC samplers that are capable of harnessing the power of fully-Bayesian UQ.

Last, the introduced EMATI method reduces the disturbance of the interface of AdResS simulations without Δ region since the AT DOFs are inserted based on the chemical environment of the molecule rather than randomly. These reduced interface artefacts are a prerequisite for using NN potentials in an AdResS simulation. Otherwise, the benefit of more accurate potentials may be outweighed by errors in the interface region, either due to incorrect structural properties within a Δ region or excess energy introduced by capped potential overlaps without a Δ region.

4.2. Discussion and Outlook

The results of this thesis show that NN potentials allow to model molecular systems at unprecedented accuracy: The DimeNet++ models of water and diamond trained via DiffTRe are in excellent agreement with all target experimental quantities. In addition, the bottom-up trained models of CG water approximate the PMF sufficiently well such that numerical errors become the dominant error component for larger time steps. This represents a remarkable level of accuracy given that for classical potentials, numerical errors are usually considered to be small enough to be neglected. This large model capacity is made possible by the flexible functional form of NN potentials, which, however, involves the loss of physical constraints built into classical potentials. Therefore, NN potential predictions can be highly inaccurate and even numerically unstable when applied outside the training data distribution, which hinders the application of NN potentials in MD simulations in practice and limits the trustworthiness of the obtained MD results.

This thesis proposed several means to address this issue: First, all NN potentials considered in this body of work were augmented by prior potentials, which significantly improved their numerical stability and, in the case of FM training, the quality of the obtained MD results. Second, the inclusion of MD simulations into the training pipeline, e.g. by incorporating top-down optimization objectives via DiffTRe or a RE-based loss function, reduced the tendency of NN potentials to enter unphysical phase-space regions. In this case, the optimization can correct the sampling of unphyscial configurations, as this results in a large increase in the loss. In this context, the prior potential serves as a well-chosen initialization of the NN potential to accelerate training. Lastly, scalable UQ techniques were shown to be well suited to provide reliable credible intervals for the obtained MD observables, which increases trustworthiness. Scalable UQ schemes also signal the practitioners cases where the model has been applied outside the training data domain and more data needs to be collected.

In sum, these molecular modeling techniques highlight the advantage of combining ML and physics-based methods: NN potentials provide unprecedented model accuracy, while physics-based methods can enforce first-principle knowledge and improve extrapolation capabilities. Therefore, the results of this thesis represent a major step towards accurate and reliable application of NN potentials in MD simulations in practice, benefiting a broad range of applications that profit from expedited material design via in-silico experimentation.

The remainder of this section discusses specific results of the thesis through the lens of the following topics:

4.2.1. Neural Network Potentials: Architectural Considerations

The articles outlined in this thesis mainly build on the DimeNet++ [67, 161] potential. Hence, the obtained results should be interpreted with the following architectural considerations in mind:

Equivariance

DimeNet++ [161] and its successor GemNet [164] are invariant graph NN potentials. Invariant NN potentials rely on the explicit computation of triplet angles as well as dihedral angles of all quadruplets in order to achieve universal approximation of functions that are invariant to translation and rotation as well as equivariant to permutation [164]. However, explicitly computing and storing all of these angles is computationally inefficient. Instead of operating on invariant graph features only, equivariant graph NNs [165, 209–211] achieve universality by using higher order internal representations such as vectors, tensors, etc. [212]. Consequently, equivariant NNs are more memory efficient. Additionally, they have been found to be more data efficient [150, 165].

Scalability

Most graph NN architectures exhibit a large receptive field given that the maximum interaction distance is given by the graph cut-off r_{cut} times the number of message-passing iterations [166], e.g. 4×0.5nm in the case of DimeNet++. This large effective cut-off is disadvantageous for the parallelizability of NN potential-based MD simulations due to the large amount of ghost atoms stored across compute nodes, which hampers the scaling to large molecular systems and longer simulation times. Consequently, more recent graph NN potentials reduce the effective interaction radius by restricting message-passing to the local neighborhood of each central atom [166] or by significantly reducing the number of message-passing iterations by passing high body-order messages [213].

Stability

Fu et al. [61] define stability of NN potentials as MD simulation time elapsed before the potential samples unphysical configurations. Stability of NN potentials is essential for their applicability in MD simulations, but this property has only recently gained attention as an important metric when designing new NN architectures. Interestingly, the functional form of the potential defined by the graph NN architecture impacts stability significantly: While NequIP [165] outperforms other architectures consistently, DimeNet becomes unstable already after 30 ps of an AT water simulation [61]. Given that CG water simulations in this thesis have been shown to be stable for up to 10 ns with the DimeNet++ potential [2, 3], this indicates that a well-chosen prior potential may increase stability by orders of magnitude. However, this thesis shows that using a prior potential is no guarantee for stability, as evidenced by some unstable pSGLD and Deep Ensemble trajectories (sec. 3.2.1).

In the case of DimeNet++, sampling unphysical phase-space regions is connected to numerical instability via the Bessel basis [67]: If two atoms approach a pairwise distance $d \approx 0$, the numerical implementation of the Bessel functions diverges, even though the analytic Bessel basis is finite at $d = 0$. In this case, the predicted potential energy diverges and the trajectory becomes NaN from this point forward. This is particularly problematic for simulation-based training such as DiffTRe or RE, where untrained models inevitably sample configurations with atom overlaps. Consequently, another important objective of the prior potential is to avoid numerical instabilities during simulation-based training. Hence, solving the problem of the numerical instability of the Bessel basis or developing stable basis sets are important directions for future research.

Short-Range Interactions

The majority of ML potentials, both descriptor- and message passing-based, only model short-range interactions. For the systems considered in this thesis, short-range interactions modeled by DimeNet++ were shown to be sufficient for an accurate description of the potential energy. However, for other systems, modeling long-range interactions can be imperative, e.g. long-range electrostatics for ionic systems [214]. To this end, environment-dependent partial charges can be predicted for each atom [214–216] and the electrostatic contribution to the potential energy can be obtained via standard Ewald summation [217]. However, given that this approach uses local features to predict partial charges, it cannot account for long-range charge transfer [59]. Recent works address this issue either by improving locally predicted partial charges via a global charge equilibration scheme [59] or by directly learning long-range interactions in Fourier space via Ewald message passing [218].

Cost-Accuracy Trade-Off

DimeNet++ with 4 message-passing iterations models up to 8-body interactions [219]. This large model capacity comes at the cost of significant computational effort for force computations in MD simulations, especially compared to classical force fields. For many systems, these high body-order interactions may not be necessary to obtain a sufficiently accurate potential energy model [220]. Consequently, the availability of different model families along the cost-accuracy Pareto front is beneficial for practitioners, who can select the cheapest model that is sufficiently accurate for the problem at hand.

Ultra-Fast Potentials [220] are an example for computationally efficient potentials, which model 2- and 3-body non-bonded interactions via B-splines. Such ML potentials are promising models for CG systems, where the high capacity of NN potentials developed for training on CQM data may be unnecessary. In addition, cheaper NN potentials can promote the use of simulation-based training schemes by further shifting the bottleneck of the ML pipeline from training to data generation.

4.2.2. Machine Learning and Concurrent Multiscale Modeling

Highly accurate CG NN potentials such as DimeNet++ are computationally more demanding than classical AT force fields. Consequently, combining expensive CG NN potentials with classical AT force fields via AdResS seems unattractive. However, as AT NN potentials become more widespread, the same relative speed-up considerations as in the classical case argue for the concurrent combination of AT NN potentials with efficient CG ML potentials in the future.

ML methods could also improve the interface accuracy in AdResS simulations: After minimizing interface artefacts caused by AT DOF insertion via back-mapping approaches such as EMATI (assuming no Δ region), the remaining error at the interface can be linked to clustering effects due to the difference in solvation free energy of the CG and AT models [197]. Consequently, by integrating the interface density error into the loss function, DiffTRe could optimize the AT-CG interaction in a differentiable AdResS simulation. Alternatively, when using AT and CG NN potentials, the interface errors may be reduced by augmenting the FM training with randomly drawn AT-CG interfaces. In this case, the NN potential can learn to approximate the target AT forces acting on AT atoms and CG particles given a box of heterogeneously resolved particles as input.

4.2.3. DiffTRe as a Structural Coarse-Graining Method

Iterative Boltzmann inversion [221] and Inverse Monte Carlo [222] are popular CG training methods that optimize pairwise potentials to match structural correlation functions, typically the radial distribution function [223]. Analogously, DiffTRe can be used to match pair, triplet and higher body-order correlation functions. Thus, DiffTRe enables structural CG training of NN potentials, which require higher body-order correlations to constrain their high body-order interactions. However, explicitly computing non-bonded, high body-order correlation functions at each optimization step is computationally demanding. Hence, RE minimization appears to be a more suitable approach for bottom-up structural coarse-graining of NN potentials because it matches all structural correlation functions conjugate to the basis functions of the potential without computing the correlations explicitly [177]. Nonetheless, combining RE and DiffTRe can be useful in order to match important thermodynamic quantities such as the pressure or the solvation free energy exactly. Furthermore, DiffTRe is needed to match experimentally obtained structural correlations.

4.2.4. Non-Uniqueness of Top-Down Neural Network Potentials

Experimental measurements are typically sparsely available only, leaving NN potentials trained via DiffTRe under-determined. This results in large epistemic uncertainty with respect to held-out observables, as evidenced with the Deep Ensemble [88, 89] method for the case of the phonon density of state in the diamond example in the original DiffTRe paper [2]. Given that many NN parameter sets $\boldsymbol{\theta}$ can match a sparse set of target observables, the maximum entropy principle [224] suggests to select the potential that yields the distribution with the maximum entropy [225]. Whether this maximum entropy model yields improved predictions for held-out observables is an open research question.

4.2.5. Uncertainty-Aware Molecular Dynamics Simulations

As demonstrated in sec. 3.2.1, M-pSGLD and the Deep Ensemble method enable reliable quantification of the epistemic uncertainty of MD observables. In addition to assessing the trustworthiness of ML potential-based MD simulation results, UQ for MD observables also facilitates top-down active learning: When the uncertainty of an observable of interest exceeds a pre-specified threshold, a new experiment can be conducted and included into the top-down training data set, maximizing the information-gain per experiment [17].

The paper presented in sec. 3.2.1 generated a dedicated MD trajectory for each parameter set $\boldsymbol{\theta}_i$. Running an ensemble of simulations in parallel is recommended to ensure the reproducibility of MD results [226]. This approach allows to estimate the uncertainty resulting from using the MD time average as an approximation to the ensemble average for finite trajectory lengths. This uncertainty can be significant given that stimulation times for systems of interest are often much shorter than the Poincaré recurrence time [226]. In practice, running multiple parallel simulations often exceeds the available computational budget. In this case, UQ for MD observables can be achieved via reweighting based on a single MD trajectory generated by the mean potential energy function [191].

Estimating the uncertainty of the potential energy at each MD time step without a large computational overhead is useful for bottom-up active learning [50, 84, 85, 130] as well as on-the-fly combination of ML potentials with classical force fields [191]. For these applications, the Monte Carlo-based UQ schemes discussed in this thesis are impractical due to their computational cost, but they can serve as a reliable baseline for benchmarking more computationally efficient UQ methods: GPs [46] are a natural choice given that they provide an uncertainty estimate for potential energy as a by-product [49]. However, GPs with hand-crafted kernels do not achieve the same predictive accuracy as state-of-the-art GNNs [164, 165]. Consequently, future research should aim at integrating GNN backbones into sampling-free UQ methods [227], e.g., via Deep Kernel Learning [228].

4.2.6. Synthesis of Machine Learning and Molecular Dynamics

Up until the late 2010s, the relationship between ML and MD was primarily characterized by ML improving the accuracy of MD simulations through powerful ML potentials, typically trained via energy or force matching as a preprocessing step [44, 51, 59, 68, 69, 153, 154, 216]. More recently, this feed-forward ML pipeline, ranging from data generation to training and subsequent application of the learned potential in an MD simulation, has been increasingly extended to include multiple feedback loops (fig. 4.1).

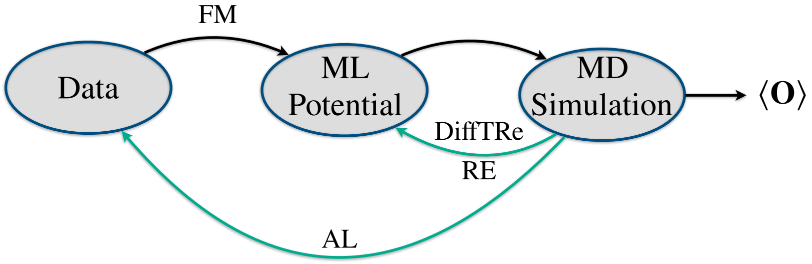


Figure 4.1.: Molecular machine learning pipeline. The standard feed-forward pipeline (black arrows) trains the machine learning (ML) potential via energy or force matching (FM). The ML potential is the input to the molecular dynamics (MD) simulation that predicts the observables $\langle \mathbf{O} \rangle$. Additional feedback loops (turquoise arrows) include data generation via active learning (AL) and simulation-based training schemes such as Differentiable Trajectory Reweighting (DiffTRe) or relative entropy (RE) minimization.

Simulation-based training schemes such as DiffTRe and RE integrate MD simulations into NN potential training. Additionally, AL [84, 85] leverages MD simulations for efficient data generation. Recent works extend this ML pipeline with additional components by introducing coarse CQM simulations for on-the-fly feature generation, [229], GANs for training CG potentials [56, 230] and accelerated active learning [231], autoencoders for learning non-linear CG mappings [232, 233] as well as normalising flows [234–236] for enhanced sampling from the Boltzmann distribution [237], NN potential training [83, 238] and free energy computations [239]. As a result, ML and physics simulations are becoming increasingly intertwined, aided by differentiable simulators [135, 136] that allow seamless integration into gradient-based training and inference pipelines [2, 95, 137, 240].

4.2.7. Towards Next-Generation Neural Network Potentials

Significant advances in ML tasks have often involved the use of large computational resources to train high-capacity models that can learn from large amounts of high quality data [156, 241]. Within the training data distribution, current GNN architectures already achieve test set errors that are significantly smaller than the expected DFT error [66, 67]. Additionally, recently developed GNN components improve upon these models by providing high-order equivariant activations [165], scalability [166, 213], efficient long-range interactions [218] and higher model capacity [242]. Thus, architectures that enable next-generation NN potentials are within reach.

The quantity and quality of available training data is therefore increasingly emerging as the main limiting factor for the accuracy and transferability of NN potentials. Thus, the development of a platform that curates available [53, 54, 243] and newly generated CQM data represents a major opportunity in this context. For maximal utility, such a novel QCM data platform could be integrated with a platform for experimental data [244], which would also simplify access to available experimental databases [245–247]. As an additional benefit, custom queries to open access databases could address the inevitable trade-off between model generality and computational cost for inference: Training smaller, application-specific models that are accurate within a small part of chemical space, e.g. by only considering atom types present in the physical process of interest, could speed up NN potential-based MD simulations. In this context, uncertainty-aware MD simulations are particularly important to ensure that a special-purpose potential is not applied outside its range of validity.

Considering both experimental and CQM data allows to combine the advantages of both sources of information: Bottom-up training improves transferability by providing broad phase-space coverage, while top-down training fine-tunes the model using available experimental data to correct for CQM errors. In addition, top-down optimization, e.g. via DiffTRe, enables training on large molecular systems, which is important given that CQM will remain limited to rather small system sizes in the foreseeable future. This hybrid modeling approach is facilitated by recently developed NN potential training libraries that support reweighting-based top-down optimization combined with bottom-up training [248].

Leveraging CQM data at different modeling levels (DFT [29, 30], coupled cluster CCSD(T) [72], quantum Monte Carlo [73, 74]) requires training schemes that account for the different levels of uncertainty associated with the different data sources. Specifically, accelerated bottom-up active learning methods [50, 231, 249] can play an important role in training high-accuracy models using CCSD(T) data while maintaining sufficient data-coverage. Furthermore, bottom-up active learning could be combined with top-down active learning:

Depending on the current state of the model, the combined active learning scheme could select which type of data to add (experimental, CCSD(T) or DFT) to maximize model accuracy and optimally utilize available experimental and computational resources. The development of such an algorithm represents a task of major importance for future research.

Bibliography

[1] Thaler, S., Praprotnik, M. & Zavadlav, J. Back-mapping augmented adaptive resolution simulation. *J. Chem. Phys.* **153**, 164118 (2020).

[2] Thaler, S. & Zavadlav, J. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting. *Nat. Commun.* **12**, 6884 (2021).

[3] Thaler, S., Stupp, M. & Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *J. Chem. Phys.* **157**, 244103 (2022).

[4] Thaler, S., Doehner, G. & Zavadlav, J. Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls. *J. Chem. Theory Comput.* **19**, 4520–4532 (2023).

[5] Thaler, S., Fuchs, P., Cukarska, A. & Zavadlav, J. JaxSGMC: Modular stochastic gradient MCMC in JAX (2023). URL `https://github.com/tummfm/jax-sgmc`. (in review process *SoftwareX*).

[6] Thaler, S. & Zavadlav, J. Uncertainty Quantification for Molecular Models via Stochastic Gradient MCMC. In *10th Vienna Conference on Mathematical Modelling*, 19–20 (Vienna, Austria, Jul. 27–29, 2022).

[7] James, S. L. Metal-organic frameworks. *Chem. Soc. Rev.* **32**, 276–288 (2003).

[8] Bhalla, A., Guo, R. & Roy, R. The perovskite structure—a review of its role in ceramic science and technology. *Mater. Res. Innov.* **4**, 3–26 (2000).

[9] Green, M. A., Ho-Baillie, A. & Snaith, H. J. The emergence of perovskite solar cells. *Nat. Photon* **8**, 506–514 (2014).

[10] Weitkamp, J. Zeolites and catalysis. *Solid State Ion.* **131**, 175–188 (2000).

[11] Wilmer, C. E. *et al.* Large-scale screening of hypothetical metal-organic frameworks. *Nat. Chem.* **4**, 83–89 (2012).

[12] Xie, T. & Grossman, J. C. Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. *Phys. Rev. Lett.* **120**, 145301 (2018).

[13] Chung, Y. G. *et al.* Advances, updates, and analytics for the computation-ready, experimental metal–organic framework database: CoRE MOF 2019. *J. Chem. Eng. Data* **64**, 5985–5998 (2019).

[14] Raza, A., Sturluson, A., Simon, C. M. & Fern, X. Message Passing Neural Networks for Partial Charge Assignment to Metal-Organic Frameworks. *J. Phys. Chem. C* **124**, 19070–19082 (2020).

[15] Rosen, A. S. *et al.* Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter* **4**, 1578–1597 (2021).

[16] Lookman, T., Balachandran, P. V., Xue, D. & Yuan, R. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Comput. Mater.* **5**, 21 (2019).

[17] Koscher, B. *et al.* Autonomous, multi-property-driven molecular discovery: from predictions to measurements and back. *ChemRxiv* (2023).

[18] Schimunek, J. *et al.* A community effort to discover small molecule SARS-CoV-2 inhibitors. *ChemRxiv* (2023).

[19] Suter, J. L., Anderson, R. L., Greenwell, H. C. & Coveney, P. V. Recent advances in large-scale atomistic and coarse-grained molecular dynamics simulation of clay minerals. *J. Mater. Chem.* **19**, 2482–2493 (2009).

[20] Ozboyaci, M., Kokh, D. B., Corni, S. & Wade, R. C. Modeling and simulation of protein-surface interactions: achievements and challenges. *Q. Rev. Biophys.* **49** (2016).

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

[22] Marx, D. & Hutter, J. *Ab initio molecular dynamics: basic theory and advanced methods* (Cambridge University Press, 2009).

[23] Kirchner, B., di Dio, P. J. & Hutter, J. *Multiscale Molecular Methods in Applied Chemistry*, chap. Real-world predictions from ab initio molecular dynamics simulations, 109–153 (Springer, 2012).

[24] Mahmoudi, M. *et al.* Protein-Nanoparticle Interactions: Opportunities and Challenges. *Chem. Rev.* **111**, 5610–5637 (2011).

[25] Allen, M. P. & Tildesley, D. J. *Computer simulation of liquids* (Oxford university press, 2017).

[26] Noid, W. G. Perspective: Coarse-grained models for biomolecular systems. *J. Chem. Phys.* **139**, 090901 (2013).

[27] Ingólfsson, H. I. *et al.* The power of coarse graining in biomolecular simulations. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **4**, 225–248 (2014).

[28] Noé, F., Tkatchenko, A., Müller, K. R. & Clementi, C. Machine Learning for Molecular Simulation. *Annu. Rev. Phys. Chem.* **71**, 361–390 (2020).

[29] Bickelhaupt, F. M. & Baerends, E. J. Kohn-Sham Density Functional Theory: Predicting and Understanding Chemistry. *Rev. Comput. Chem.* 1–86 (2000).

[30] Becke, A. D. Perspective: Fifty years of density-functional theory in chemical physics. *J. Chem. Phys.* **140**, 18A301 (2014).

[31] Izvekov, S. & Voth, G. A. Multiscale coarse graining of liquid-state systems. *J. Chem. Phys.* **123**, 134105 (2005).

[32] Noid, W. G. *et al.* The multiscale coarse-graining method. I. A rigorous bridge between atomistic and coarse-grained models. *J. Chem. Phys.* **128**, 244114 (2008).

[33] Noid, W. *et al.* The multiscale coarse-graining method. II. Numerical implementation for coarse-grained molecular models. *J. Chem. Phys.* **128**, 244115 (2008).

[34] Potter, T. D., Tasche, J. & Wilson, M. R. Assessing the transferability of common top-down and bottom-up coarse-grained molecular models for molecular mixtures. *Phys. Chem. Chem. Phys.* **21**, 1912–1927 (2019).

[35] Cornell, W. D. *et al.* A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **117**, 5179–5197 (1995).

[36] Oostenbrink, C., Villa, A., Mark, A. E. & Van Gunsteren, W. F. A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6. *J. Comput. Chem.* **25**, 1656–1676 (2004).

[37] Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P. & De Vries, A. H. The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **111**, 7812–7824 (2007).

[38] Fröhlking, T., Bernetti, M., Calonaci, N. & Bussi, G. Toward empirical force fields that match experimental observables. *J. Chem. Phys.* **152**, 230902 (2020).

[39] Stillinger, F. H. & Weber, T. A. Computer simulation of local order in condensed phases of silicon. *Phys. Rev. B* **31**, 5262–5271 (1985).

[40] Bianchini, F., Kermode, J. & De Vita, A. Modelling defects in Ni–Al with EAM and DFT calculations. *Model. Simul. Mater. Sci. Eng.* **24**, 045012 (2016).

[41] Van Duin, A. C., Dasgupta, S., Lorant, F. & Goddard, W. A. ReaxFF: a reactive force field for hydrocarbons. *J. Phys. Chem. C* **105**, 9396–9409 (2001).

[42] Wang, L.-P., Chen, J. & Van Voorhis, T. Systematic Parametrization of Polarizable Force Fields from Quantum Chemistry Data. *J. Chem. Theory Comput.* **9**, 452–460 (2013).

[43] Jing, Z. *et al.* Polarizable Force Fields for Biomolecular Simulations: Recent Advances and Applications. *Annu. Rev. Biophys.* **48**, 371–394 (2019).

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

[45] 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.* **120**, 143001 (2018).

[46] Williams, C. K. & Rasmussen, C. E. *Gaussian processes for machine learning*, vol. 2 (MIT press Cambridge, MA, 2006).

[47] John, S. & Csányi, G. Many-body coarse-grained interactions using Gaussian approximation potentials. *J. Phys. Chem. B* **121**, 10934–10949 (2017).

[48] Scherer, C., Scheid, R., Andrienko, D. & Bereau, T. Kernel-Based Machine Learning for Efficient Simulations of Molecular Liquids. *J. Chem. Theory Comput.* **16**, 3194–3204 (2020).

[49] Deringer, V. L. *et al.* Gaussian Process Regression for Materials and Molecules. *Chem. Rev.* **121**, 10073–10141 (2021).

[50] van der Oord, C., Sachs, M., Kovács, D. P., Ortner, C. & Csányi, G. Hyperactive Learning (HAL) for Data-Driven Interatomic Potentials. *arXiv:2210.04225* (2022).

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

[52] Behler, J. Constructing high-dimensional neural network potentials: A tutorial review. *Int. J. Quantum Chem.* **115**, 1032–1050 (2015).

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

[54] Smith, J. S. *et al.* The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Sci. Data* **7**, 134 (2020).

[55] Schütt, K. T. *et al.* SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, vol. 30 (Long Beach, CA, USA, Dec. 4–9, 2017).

[56] Durumeric, A. E. & Voth, G. A. Adversarial-residual-coarse-graining: Applying machine learning theory to systematic molecular coarse-graining. *J. Chem. Phys.* **151**, 124110 (2019).

[57] Cubuk, E. D. & Schoenholz, S. S. Adversarial forces of physical models. In *Machine Learning for the Physical Sciences Workshop at NeurIPS* (Online, Dec. 11, 2020).

[58] Ruza, J. *et al.* Temperature-transferable coarse-graining of ionic liquids with dual graph convolutional neural networks. *J. Chem. Phys.* **153**, 164501 (2020).

[59] Ko, T. W., Finkler, J. A., Goedecker, S. & Behler, J. A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **12**, 398 (2021).

[60] Schwalbe-Koda, D., Tan, A. R. & Gómez-Bombarelli, R. Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nat. Commun.* **12**, 5104 (2021).

[61] Fu, X. *et al.* Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations. In *AI for Science: Progress and Promises Workshop at NeurIPS* (New Orleans, LA, USA, Dec. 2, 2022).

[62] Lemke, T. & Peter, C. Neural Network Based Prediction of Conformational Free Energies – A New Route toward Coarse-Grained Simulation Models. *J. Chem. Theory Comput.* **13**, 6213–6221 (2017).

[63] Bonati, L. & Parrinello, M. Silicon Liquid Structure and Crystal Nucleation from Ab Initio Deep Metadynamics. *Phys. Rev. Lett.* **121**, 265701 (2018).

[64] Smith, J. S. *et al.* Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning. *Nat. Commun.* **10**, 2930 (2019).

[65] Smith, J. S. *et al.* Automated discovery of a robust interatomic potential for aluminum. *Nat. Commun.* **12**, 1257 (2021).

[66] Faber, F. A. *et al.* Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theory Comput.* **13**, 5255–5264 (2017).

[67] Klicpera, J., Groß, J. & Günnemann, S. Directional Message Passing for Molecular Graphs. In *8th International Conference on Learning Representations* (Online, Apr. 26 – May 1, 2020).

[68] Wang, J. *et al.* Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **5**, 755–767 (2019).

[69] Husic, B. E. *et al.* Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **153**, 194101 (2020).

[70] Gal, Y., Koumoutsakos, P., Lanusse, F., Louppe, G. & Papadimitriou, C. Bayesian uncertainty quantification for machine-learned models in physics. *Nat. Rev. Phys.* **4**, 573–577 (2022).

[71] Stocker, S., Gasteiger, J., Becker, F., Günnemann, S. & Margraf, J. T. How robust are modern graph neural network potentials in long and hot molecular dynamics simulations? *Mach. Learn.: Sci. Technol.* **3**, 045010 (2022).

[72] Purvis III, G. D. & Bartlett, R. J. A full coupled-cluster singles and doubles model: The inclusion of disconnected triples. *J. Chem. Phys.* **76**, 1910–1918 (1982).

[73] Foulkes, W., Mitas, L., Needs, R. & Rajagopal, G. Quantum Monte Carlo simulations of solids. *Rev. Mod. Phys.* **73**, 33 (2001).

[74] Carlson, J. *et al.* Quantum Monte Carlo methods for nuclear physics. *Rev. Mod. Phys.* **87**, 1067 (2015).

[75] Hermann, J., Schätzle, Z. & Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.* **12**, 891–897 (2020).

[76] Pfau, D., Spencer, J. S., Matthews, A. G. & Foulkes, W. M. C. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Research* **2**, 033429 (2020).

[77] Sauceda, H. E., Vassilev-Galindo, V., Chmiela, S., Müller, K. R. & Tkatchenko, A. Dynamical strengthening of covalent and non-covalent molecular interactions by nuclear quantum effects at finite temperature. *Nat. Commun.* **12**, 442 (2021).

[78] Bejagam, K. K., Singh, S., An, Y. & Deshmukh, S. A. Machine-Learned Coarse-Grained Models. *J. Phys. Chem. Lett.* **9**, 4667–4672 (2018).

[79] Ye, H., Xian, W. & Li, Y. Machine Learning of Coarse-Grained Models for Organic Molecules and Polymers: Progress, Opportunities, and Challenges. *ACS Omega* **6**, 1758–1772 (2021).

[80] Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, vol. 32 (Vancouver, Canada, Dec. 8–14, 2019).

[81] Abadi, M. *et al.* Tensorflow: a system for large-scale machine learning. In *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, vol. 16, 265–283 (Savannah, GA, USA, Nov. 2–4, 2016).

[82] Bradbury, J. *et al.* JAX: composable transformations of Python+NumPy programs (2018). URL `http://github.com/google/jax`.

[83] Köhler, J., Chen, Y., Krämer, A., Clementi, C. & Noé, F. Flow-Matching: Efficient Coarse-Graining of Molecular Dynamics without Forces. *J. Chem. Theory Comput.* 942–952 (2022).

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

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

[86] Loeffler, T. D., Patra, T. K., Chan, H. & Sankaranarayanan, S. K. Active learning a coarse-grained neural network model for bulk water from sparse training data. *Mol. Syst. Des. Eng.* **5**, 902–910 (2020).

[87] Jinnouchi, R., Miwa, K., Karsai, F., Kresse, G. & Asahi, R. On-the-Fly Active Learning of Interatomic Potentials for Large-Scale Atomistic Simulations. *J. Phys. Chem. Lett.* **11**, 6946–6955 (2020).

[88] Hansen, L. & Salamon, P. Neural Network Ensembles. *IEEE Trans. Pattern Anal. Machine Intell.* **12**, 993–1001 (1990).

[89] Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, vol. 30 (Long Beach, CA, USA, Dec. 4–9, 2017).

[90] Kahle, L. & Zipoli, F. Quality of uncertainty estimates from neural network potential ensembles. *Phys. Rev. E* **105**, 015311 (2022).

[91] Tuckerman, M. *Statistical Mechanics: Theory and Molecular Simulation* (Oxford University Press, 2010), second edn.

[92] Swope, W. C., Andersen, H. C., Berens, P. H. & Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *J. Chem. Phys.* **76**, 637–649 (1982).

[93] Pettitt, B. M. & Karplus, M. The potential of mean force surface for the alanine dipeptide in aqueous solution: a theoretical approach. *Chem. Phys. Lett.* **121**, 194–201 (1985).

[94] Tobias, D. J. & Brooks III, C. L. Conformational Equilibrium in the Alanine Dipeptide in the Gas Phase and Aqueous Solution: A Comparison of Theoretical Results. *J. Phys. Chem.* **96**, 3864–3870 (1992).

[95] Wang, W., Axelrod, S. & Gómez-Bombarelli, R. Differentiable Molecular Simulations for Control and Learning. In *Workshop on Integration of Deep Neural Models and Differential Equations at ICLR* (Online, Apr. 26, 2020).

[96] Thompson, A. P., Plimpton, S. J. & Mattson, W. General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *J. Chem. Phys.* **131**, 154107 (2009).

[97] Chen, X. *et al.* TensorAlloy: An automatic atomistic neural network program for alloys. *Comput. Phys. Commun.* **250**, 107057 (2020).

[98] Van Workum, K., Yoshimoto, K., De Pablo, J. J. & Douglas, J. F. Isothermal stress and elasticity tensors for ions and point dipoles using Ewald summations. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **71**, 061102 (2005).

[99] Soper, A. K. & Benmore, C. J. Quantum Differences between Heavy and Light Water. *Phys. Rev. Lett.* **101**, 065502 (2008).

[100] Mills, R. Self-diffusion in normal and heavy water in the range 1-45°. *J. Phys. Chem.* **77**, 685–688 (1973).

[101] McSkimin, H. J., Andreatch, P. & Glynn, P. The Elastic Stiffness Moduli of Diamond. *J. Appl. Phys.* **43**, 985–987 (1972).

[102] Martyna, G. J., Klein, M. L. & Tuckerman, M. Nosé-Hoover chains: The canonical ensemble via continuous dynamics. *J. Chem. Phys.* **97**, 2635–2643 (1992).

[103] Hastings, W. K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109 (1970).

[104] Duane, S., Kennedy, A. D., Pendleton, B. J. & Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **195**, 216–222 (1987).

[105] Neal, R. M. *Handbook of Markov Chain Monte Carlo*, chap. MCMC using Hamiltonian Dynamics, 139–188 (Chapman and Hall/CRC, New York, USA, 2011), first edn.

[106] Hoffman, M. D. & Gelman, A. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15**, 1593–1623 (2014).

[107] Hermans, J., Berendsen, H. J., Van Gunsteren, W. F. & Postma, J. P. A consistent empirical potential for water–protein interactions. *Biopolymers* **23**, 1513–1518 (1984).

[108] Tironi, I. G., Sperb, R., Smith, P. E. & van Gunsteren, W. F. A generalized reaction field method for molecular dynamics simulations. *J. Chem. Phys.* **102**, 5451–5459 (1995).

[109] Zwanzig, R. W. High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. *J. Chem. Phys.* **22**, 1420–1426 (1954).

[110] Norgaard, A. B., Ferkinghoff-Borg, J. & Lindorff-Larsen, K. Experimental Parameterization of an Energy Function for the Simulation of Unfolded Proteins. *Biophys. J.* **94**, 182–192 (2008).

[111] Li, D. W. & Brüschweiler, R. Iterative Optimization of Molecular Mechanics Force Fields from NMR Data of Full-Length Proteins. *J. Chem. Theory Comput.* **7**, 1773–1782 (2011).

[112] Wang, L. P., Chen, J. & Van Voorhis, T. Systematic Parametrization of Polarizable Force Fields from Quantum Chemistry Data. *J. Chem. Theory Comput.* **9**, 452–460 (2013).

[113] Bennett, C. H. Efficient estimation of free energy differences from Monte Carlo data. *J. Comput. Phys.* **22**, 245–268 (1976).

[114] Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H. & Kollman, P. A. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comput. Chem.* **13**, 1011–1021 (1992).

[115] Kadoura, A., Salama, A. & Sun, S. Switching Between the NVT and NpT Ensembles Using the Reweighting and Reconstruction Scheme. *Procedia Comput. Sci.* **51**, 1259–1268 (2015).

[116] Conrad, P. & De Pablo, J. Comparison of histogram reweighting techniques for a flexible water model. *Fluid Phase Equilib.* **150**, 51–61 (1998).

[117] Fenwick, M. K. A direct multiple histogram reweighting method for optimal computation of the density of states. *J. Chem. Phys.* **129**, 09B619 (2008).

[118] Messerly, R. A., Soroush Barhaghi, M., Potoff, J. J. & Shirts, M. R. Histogram-Free Reweighting with Grand Canonical Monte Carlo: Post-simulation Optimization of Non-bonded Potentials for Phase Equilibria. *J. Chem. Eng. Data* **64**, 3701–3717 (2019).

[119] Torrie, G. M. & Valleau, J. P. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **23**, 187–199 (1977).

[120] Carmichael, S. P. & Shell, M. S. A New Multiscale Algorithm and Its Application to Coarse-Grained Peptide Models for Self-Assembly. *J. Phys. Chem. B* **116**, 8383–8393 (2012).

[121] Richtmyer, R. D. & Morton, K. W. *Difference Methods for Initial-Value Problems* (Krieger Pub. Co., 1967), second edn.

[122] Yoshida, H. Construction of higher order symplectic integrators. *Phys. Lett. A* **150**, 262–268 (1990).

[123] Toxvaerd, S. Hamiltonians for discrete dynamics. *Phys. Rev. E* **50**, 2271 (1994).

[124] Toxvaerd, S. Ensemble simulations with discrete classical dynamics. *J. Chem. Phys.* **139**, 224106 (2013).

[125] McAliley, J. H. & Bruce, D. A. Development of Force Field Parameters for Molecular Simulation of Polylactide. *J. Chem. Theory Comput.* **7**, 3756–3767 (2011).

[126] Mostaghim, S., Hoffmann, M., Konig, P. H., Frauenheim, T. & Teich, J. Molecular force field parametrization using multi-objective evolutionary algorithms. In *Proceedings of the Congress on Evolutionary Computation*, 212–219 (Portland, OR, USA, Jun. 19–23, 2004).

[127] Betz, R. M. & Walker, R. C. Paramfit: Automated optimization of force field parameters for molecular dynamics simulations. *J. Comput. Chem.* **36**, 79–87 (2015).

[128] Bejagam, K. K., Singh, S., An, Y., Berry, C. & Deshmukh, S. A. PSO-Assisted Development of New Transferable Coarse-Grained Water Models. *J. Phys. Chem. B* **122**, 1958–1971 (2018).

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

[130] Smith, J. S. *et al.* The ANI-1ccx and ANI-1x data sets, coupled-cluster and density functional theory properties for molecules. *Sci. Data* **7**, 134 (2020).

[131] Pukrittayakamee, A. *et al.* Simultaneous fitting of a potential-energy surface and its corresponding force fields using feedforward neural networks. *J. Chem. Phys.* **130**, 134101 (2009).

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

[133] Tieleman, T. & Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning* **4**, 26–31 (2012).

[134] Kingma, D. P. & Ba, J. L. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations* (San Diego, CA, USA, May 7-9, 2015).

[135] Schoenholz, S. S. & Cubuk, E. D. JAX, M.D.: A Framework for Differentiable Physics. In *Advances in Neural Information Processing Systems*, vol. 33 (Online, Dec. 6–12, 2020).

[136] Doerr, S. *et al.* TorchMD: A Deep Learning Framework for Molecular Simulations. *J. Chem. Theory Comput.* **17**, 2355–2363 (2021).

[137] Ingraham, J., Riesselman, A., Sander, C. & Marks, D. Learning Protein Structure with a Differentiable Simulator. In *7th International Conference on Learning Representations* (New Orleans, LA, USA, May 6–9, 2019).

[138] Goodrich, C. P., King, E. M., Schoenholz, S. S., Cubuk, E. D. & Brenner, M. P. Designing self-assembling kinetics with differentiable statistical physics models. *Proc. Natl. Acad. Sci. U.S.A.* **118**, e2024083118 (2021).

[139] Wang, W., Wu, Z., Dietschreit, J. C. & Gómez-Bombarelli, R. Learning pair potentials using differentiable simulations. *J. Chem. Phys.* **158**, 044113 (2023).

[140] Metz, L., Freeman, C. D., Schoenholz, S. S. & Kachman, T. Gradients are not all you need. *arXiv preprint arXiv:2111.05803* (2021).

[141] Evans, D. J., Cohen, E. & Morriss, G. P. Viscosity of a simple fluid from its maximal Lyapunov exponents. *Phys. Rev. A* **42**, 5990 (1990).

[142] Wolf, A., Swift, J. B., Swinney, H. L. & Vastano, J. A. Determining Lyapunov exponents from a time series. *Phys. D: Nonlinear Phenom.* **16**, 285–317 (1985).

[143] Di Pierro, M. & Elber, R. Automated Optimization of Potential Parameters. *J. Chem. Theory Comput.* **9**, 3311–3320 (2013).

[144] Wang, L. P. *et al.* Systematic Improvement of a Classical Molecular Model of Water. *J. Phys. Chem. B* **117**, 9956–9972 (2013).

[145] Wang, L. P., Martinez, T. J. & Pande, V. S. Building Force Fields: An Automatic, Systematic, and Reproducible Approach. *J. Phys. Chem. Lett.* **5**, 1885–1891 (2014).

[146] Das, A. & Andersen, H. C. The multiscale coarse-graining method. III. A test of pairwise additivity of the coarse-grained potential and of new basis functions for the variational calculation. *J. Chem. Phys.* **131**, 034102 (2009).

[147] Daw, M. S. & Baskes, M. I. Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals. *Phys. Rev. B* **29**, 6443 (1984).

[148] Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *J. Chem. Theory Comput.* **11**, 2087–2096 (2015).

[149] Shorten, C. & Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **6**, 60 (2019).

[150] Geiger, M. & Smidt, T. e3nn: Euclidean Neural Networks. *arXiv preprint arXiv:2207.09453* (2022).

[151] Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Phys. Rev. B* **99**, 014104 (2019).

[152] Dusson, G. *et al.* Atomic cluster expansion: Completeness, efficiency and stability. *J. Comput. Phys.* **454**, 110946 (2022).

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

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

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

[156] Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, vol. 25 (Lake Tahoe, NV, USA, Dec. 3–8, 2012).

[157] Bogojeski, M., Vogt-Maranto, L., Tuckerman, M. E., Müller, K.-R. & Burke, K. Quantum chemical accuracy from density functional approximations via machine learning. *Nat. Commun.* **11**, 5223 (2020).

[158] Battaglia, P. W. *et al.* Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261* (2018).

[159] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, 1263–1272 (Sydney, Australia, Aug. 6–11, 2017).

[160] Zhang, L., Han, J., Wang, H., Car, R. & Weinan, W. E. DeePCG: Constructing coarse-grained models via deep neural networks. *J. Chem. Phys.* **149**, 034101 (2018).

[161] Klicpera, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules. In *Machine Learning for Molecules Workshop at NeurIPS* (Online, Dec. 12, 2020).

[162] Qiao, Z., Welborn, M., Anandkumar, A., Manby, F. R. & Miller, T. F. OrbNet: Deep learning for quantum chemistry using symmetry-adapted atomic-orbital features. *J. Chem. Phys.* **153**, 124111 (2020).

[163] Chen, Y. *et al.* Machine learning implicit solvation for molecular dynamics. *J. Chem. Phys.* **155**, 084101 (2021).

[164] Gasteiger, J., Becker, F. & Günnemann, S. GemNet: Universal Directional Graph Neural Networks for Molecules. In *Advances in Neural Information Processing Systems*, vol. 34 (Online, Dec. 6–14, 2021).

[165] Batzner, S. *et al.* E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).

[166] Musaelian, A. *et al.* Learning local equivariant representations for large-scale atomistic dynamics. *Nat. Commun.* **14**, 579 (2023).

[167] He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (Las Vegas, NV, USA, Jun. 27–30, 2016).

[168] Kalligiannaki, E., Harmandaris, V., Katsoulakis, M. A. & Plecháč, P. The geometry of generalized force matching and related information metrics in coarse-graining of molecular systems. *J. Chem. Phys.* **143**, 084105 (2015).

[169] Harmandaris, V., Kalligiannaki, E., Katsoulakis, M. & Plecháč, P. Path-space variational inference for non-equilibrium coarse-grained systems. *J. Comput. Phys.* **314**, 355–383 (2016).

[170] Chan, H. *et al.* Machine learning coarse grained models for water. *Nat. Commun.* **10**, 379 (2019).

[171] Shell, M. S. The relative entropy is fundamental to multiscale and inverse thermodynamic problems. *J. Chem. Phys.* **129**, 144108 (2008).

[172] Mullinax, J. & Noid, W. Generalized Yvon-Born-Green theory for molecular systems. *Phys. Rev. Lett.* **103**, 198104 (2009).

[173] Kullback, S. & Leibler, R. A. On Information and Sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951).

[174] Chaimovich, A. & Shell, M. S. Anomalous waterlike behavior in spherically-symmetric water models optimized with the relative entropy. *Phys. Chem. Chem. Phys.* **11**, 1901–1915 (2009).

[175] Chaimovich, A. & Shell, M. S. Relative entropy as a universal metric for multiscale errors. *Physical Review E* **81**, 060104 (2010).

[176] Espanol, P. & Zuniga, I. Obtaining fully dynamic coarse-grained models from MD. *Phys. Chem. Chem. Phys.* **13**, 10538–10545 (2011).

[177] Chaimovich, A. & Shell, M. S. Coarse-graining errors and numerical optimization using a relative entropy framework. *J. Chem. Phys.* **134**, 094112 (2011).

[178] Bottaro, S., Lindorff-Larsen, K. & Best, R. B. Variational Optimization of an All-Atom Implicit Solvent Force Field To Match Explicit Solvent Simulation Data. *J. Chem. Theory Comput.* **9**, 5641–5652 (2013).

[179] Rudzinski, J. F. & Noid, W. Coarse-graining entropy, forces, and structures. *J. Chem. Phys.* **135**, 214101 (2011).

[180] Shell, M. S. *Advances in Chemical Physics*, vol. 161, chap. Coarse-graining with the relative entropy, 395–441 (John Wiley & Sons, Inc., 2016), first edn.

[181] Ovadia, Y. *et al.* Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. In *Advances in Neural Information Processing Systems*, vol. 32 (Vancouver, BC, Canada, Dec. 8–14, 2019).

[182] Wilson, A. G. & Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In *Advances in Neural Information Processing Systems*, vol. 33 (Online, Dec. 6–12, 2020).

[183] Welling, M. & Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, 681–688 (Bellevue, WA, USA, Jun. 28 – Jul. 2, 2011).

[184] Chen, T., Fox, E. & Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, 1683–1691 (Beijing, China, Jun. 21–26, 2014).

[185] Li, C., Chen, C., Carlson, D. E. & Carin, L. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 1788–1794 (Phoenix, AZ, USA, February 12–17, 2016).

[186] Nemeth, C. & Fearnhead, P. Stochastic gradient Markov chain Monte Carlo. *J. Am. Stat. Assoc.* **116**, 433–450 (2021).

[187] Lamb, G. & Paige, B. Bayesian Graph Neural Networks for Molecular Property Prediction. In *Machine Learning for Molecules Workshop at NeurIPS* (Online, Dec. 12, 2020).

[188] Graves, A. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, vol. 24 (Granada, Spain, Dec. 12 – 14, 2011).

[189] Hoffman, M. D., Blei, D. M., Wang, C. & Paisley, J. Stochastic Variational Inference. *J. Mach. Learn. Res.* **14**, 1303–1347 (2013).

[190] Wenzel, F. *et al.* How Good is the Bayes Posterior in Deep Neural Networks Really? In *Proceedings of the 37th International Conference on Machine Learning*, 10248–10259 (Online, Jul. 13–18, 2020).

[191] Imbalzano, G. *et al.* Uncertainty estimation for molecular dynamics and sampling. *J. Chem. Phys.* **154**, 074102 (2021).

[192] Potestio, R., Peter, C. & Kremer, K. Computer Simulations of Soft matter: Linking the Scales. *Entropy* **16**, 4199–4245 (2014).

[193] Wang, H., Schütte, C. & Delle Site, L. Adaptive Resolution Simulation (AdResS): A Smooth Thermodynamic and Structural Transition from Atomistic to Coarse Grained Resolution and Vice Versa in a Grand Canonical Fashion. *J. Chem. Theory Comput.* **8**, 2878–2887 (2012).

[194] Shan, Y. *et al.* How does a drug molecule find its target binding site? *J. Am. Chem. Soc.* **133**, 9181–9183 (2011).

[195] Wang, Y. *et al.* How reliable are molecular dynamics simulations of membrane active antimicrobial peptides? *Biochim. Biophys. Acta - Biomembr.* **1838**, 2280–2288 (2014).

[196] Shi, Q., Izvekov, S. & Voth, G. A. Mixed Atomistic and Coarse-Grained Molecular Dynamics: Simulation of a Membrane-Bound Ion Channel. *J. Phys. Chem. B* **110**, 15045–15048 (2006).

[197] Rzepiela, A. J., Louhivuori, M., Peter, C. & Marrink, S. J. Hybrid simulations: combining atomistic and coarse-grained force fields using virtual sites. *Phys. Chem. Chem. Phys.* **13**, 10437–10448 (2011).

[198] Sokkar, P., Choi, S. M. & Rhee, Y. M. Simple Method for Simulating the Mixture of Atomistic and Coarse-Grained Molecular Systems. *J. Chem. Theory Comput.* **9**, 3728–3739 (2013).

[199] Kuhn, A. B., Gopal, S. M. & Schäfer, L. V. On Using Atomistic Solvent Layers in Hybrid All-Atom/Coarse-Grained Molecular Dynamics Simulations. *J. Chem. Theory Comput.* **11**, 4460–4472 (2015).

[200] Praprotnik, M., Delle Site, L. & Kremer, K. Adaptive resolution molecular-dynamics simulation: Changing the degrees of freedom on the fly. *J. Chem. Phys.* **123**, 224106 (2005).

[201] Potestio, R. *et al.* Hamiltonian Adaptive Resolution Simulation for Molecular Liquids. *Phys. Rev. Lett.* **110**, 108301 (2013).

[202] Chaimovich, A., Peter, C. & Kremer, K. Relative resolution: A hybrid formalism for fluid mixtures. *J. Chem. Phys.* **143**, 243107 (2015).

[203] Praprotnik, M., Site, L. D. & Kremer, K. Multiscale Simulation of Soft Matter: From Scale Bridging to Adaptive Resolution. *Annu. Rev. Phys. Chem.* **59**, 545–71 (2008).

[204] Wang, H., Hartmann, C., Schütte, C. & Site, L. D. Grand-Canonical-like Molecular-Dynamics Simulations by Using an Adaptive-Resolution Technique. *Phys. Rev. X* **3**, 011018 (2013).

[205] Zavadlav, J., Melo, M. N., Marrink, S. J. & Praprotnik, M. Adaptive resolution simulation of an atomistic protein in MARTINI water. *J. Chem. Phys.* **140**, 054114 (2014).

[206] Peters, J. H., Klein, R. & Delle Site, L. Simulation of macromolecular liquids with the adaptive resolution molecular dynamics technique. *Phys. Rev. E* **94**, 023309 (2016).

[207] Junghans, C., Agarwal, A. & Delle Site, L. Computational efficiency and Amdahl's law for the adaptive resolution simulation technique. *Comput. Phys. Commun.* **215**, 20–25 (2017).

[208] Krekeler, C., Agarwal, A., Junghans, C., Praprotnik, M. & Delle Site, L. Adaptive resolution molecular dynamics technique: Down to the essential. *J. Chem. Phys.* **149**, 024104 (2018).

[209] Weiler, M., Geiger, M., Welling, M., Boomsma, W. & Cohen, T. S. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *Advances in Neural Information Processing Systems*, vol. 31 (Montreal, Canada, Dec. 2–8, 2018).

[210] Thomas, N. *et al.* Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219* (2018).

[211] Anderson, B., Hy, T.-S. & Kondor, R. Cormorant: Covariant Molecular Neural Networks. In *Advances in Neural Information Processing Systems*, vol. 32 (Vancouver, Canada, Dec. 8–14, 2019).

[212] Dym, N. & Maron, H. On the Universality of Rotation Equivariant Point Cloud Networks. In *9th International Conference on Learning Representations* (Vienna, Austria, May 3–7, 2021).

[213] Batatia, I., Kovács, D. P., Simm, G. N., Ortner, C. & Csányi, G. MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields. In *Advances in Neural Information Processing Systems* (New Orleans, LA, USA, Nov. 28 – Dec. 9, 2022).

[214] Deng, Z., Chen, C., Li, X.-G. & Ong, S. P. An electrostatic spectral neighbor analysis potential for lithium nitride. *npj Comput. Mater.* **5**, 75 (2019).

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

[216] Artrith, N., Morawietz, T. & Behler, J. High-dimensional neural-network potentials for multicomponent systems: Applications to zinc oxide. *Phya. Rev. B* **83**, 153101 (2011).

[217] Darden, T., York, D. & Pedersen, L. Particle mesh Ewald: An N log(N) method for Ewald sums in large systems. *J. Chem Phys.* **98**, 10089–10092 (1993).

[218] Kosmala, A., Gasteiger, J., Gao, N. & Günnemann, S. Ewald-based Long-Range Message Passing for Molecular Graphs. *arXiv preprint arXiv:2303.04791* (2023).

[219] Batatia, I. *et al.* The Design Space of E(3)-Equivariant Atom-Centered Interatomic Potentials. *arXiv preprint arXiv:2205.06643* (2022).

[220] Xie, S. R., Rupp, M. & Hennig, R. G. Ultra-fast Force Fields (UF3) framework for machine-learning interatomic potentials. In *American Physical Society March Meeting* (Chicago, IL, USA, Mar. 14–18, 2021).

[221] Reith, D., Pütz, M. & Müller-Plathe, F. Deriving effective mesoscale potentials from atomistic simulations. *J. Comput. Chem.* **24**, 1624–1636 (2003).

[222] Lyubartsev, A. P. & Laaksonen, A. Calculation of effective interaction potentials from radial distribution functions: A reverse Monte Carlo approach. *Phys. Rev. E* **52**, 3730–3737 (1995).

[223] Bevc, S., Junghans, C. & Praprotnik, M. STOCK: Structure mapper and online coarse-graining kit for molecular simulations. *J. Comput. Chem.* **36**, 467–477 (2015).

[224] Jaynes, E. T. Information Theory and Statistical Mechanics. *Phys. Rev.* **106**, 620–630 (1957).

[225] Zhu, S. C., Wu, Y. N. & Mumford, D. Minimax Entropy Principle and Its Application to Texture Modeling. *Neural Comput.* **9**, 1627–1660 (1997).

[226] Wan, S., Sinclair, R. C. & Coveney, P. V. Uncertainty quantification in classical molecular dynamics. *Philos. Trans. Royal Soc. A* **379**, 20200082 (2021).

[227] Zhu, A., Batzner, S., Musaelian, A. & Kozinsky, B. Fast Uncertainty Estimates in Deep Learning Interatomic Potentials. *arXiv preprint arXiv:2211.09866* (2022).

[228] Wilson, A. G., Hu, Z., Salakhutdinov, R. R. & Xing, E. P. Stochastic Variational Deep Kernel Learning. *Advances in Neural Information Processing Systems* **29** (Dec. 5–10, 2016).

[229] Qiao, Z. *et al.* Informing geometric deep learning with electronic interactions to accelerate quantum chemistry. *Proc. Natl. Acad. Sci. U.S.A.* **119**, e2205221119 (2022).

[230] Ding, X. & Zhang, B. Contrastive Learning of Coarse-Grained Force Fields. *J. Chem. Theory Comput.* **18**, 6334–6344 (2022).

[231] Schwalbe-Koda, D., Tan, A. R. & Gómez-Bombarelli, R. Differentiable sampling of molecular geometries with uncertainty-based adversarial attacks. *Nat. Commun.* **12**, 5104 (2021).

[232] Mardt, A., Pasquali, L., Wu, H. & Noé, F. VAMPnets for deep learning of molecular kinetics. *Nat. Commun.* **9**, 5 (2018).

[233] Wang, W. & Gómez-Bombarelli, R. Coarse-graining auto-encoders for molecular dynamics. *npj Comput. Mater.* **5**, 125 (2019).

[234] Noé, F., Olsson, S., Köhler, J. & Wu, H. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147 (2019).

[235] Dibak, M., Klein, L., Krämer, A. & Noé, F. Temperature steerable flows and Boltzmann generators. *Phys. Rev. Res.* **4**, L042005 (2022).

[236] Köhler, J., Invernizzi, M., de Haan, P. & Noé, F. Rigid body flows for sampling molecular crystal structures. *arXiv preprint arXiv:2301.11355* (2023).

[237] Klein, L. *et al.* Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics. *arXiv preprint arXiv:2302.01170* (2023).

[238] Wirnsberger, P. *et al.* Normalizing flows for atomic solids. *Mach. Learn.: Sci. Technol.* **3**, 025009 (2022).

[239] Wirnsberger, P. *et al.* Targeted free energy estimation via learned mappings. *J. Chem. Phys.* **153**, 144112 (2020).

[240] Šípka, M., Dietschreit, J. C. & Gómez-Bombarelli, R. Differentiable Simulations for Enhanced Sampling of Rare Events. *arXiv preprint arXiv:2301.03480* (2023).

[241] Brown, T. *et al.* Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, vol. 33 (Online, Dec. 6–12, 2020).

[242] Thölke, P. & Fabritiis, G. D. Equivariant Transformers for Neural Network based Molecular Potentials. In *10th International Conference on Learning Representations* (Online, Apr. 25–29, 2022).

[243] Eastman, P. *et al.* SPICE, A Dataset of Drug-like Molecules and Peptides for Training Machine Learning Potentials. *Sci. Data* **10**, 11 (2023).

[244] Boothroyd, S., Wang, L.-P., Mobley, D. L., Chodera, J. D. & Shirts, M. R. Open Force Field Evaluator: An Automated, Efficient, and Scalable Framework for the Estimation of Physical Properties from Molecular Simulation. *J. Chem. Theory Comput.* **18**, 3566–3576 (2022).

[245] Frenkel, M. *et al.* ThermoML – An XML-Based Approach for Storage and Exchange of Experimental and Critically Evaluated Thermophysical and Thermochemical Property Data. 1. Experimental Data. *J. Chem. Eng. Data* **48**, 2–13 (2003).

[246] Mobley, D. L. & Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Des.* **28**, 711–720 (2014).

[247] Liu, T., Lin, Y., Wen, X., Jorissen, R. N. & Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res.* **35**, D198–D201 (2007).

[248] Wang, X. *et al.* DMFF: An Open-Source Automatic Differentiable Platform for Molecular Force Field Development and Molecular Dynamics Simulation. *ChemRxiv* (2022).

[249] Kulichenko, M. *et al.* Uncertainty-driven dynamics for active learning of interatomic potentials. *Nat. Comput. Sci.* **3**, 230–239 (2023).

Acknowledgments

Finally, I would like to express my gratitude to all those who contributed in various ways to the success of this thesis.

First and foremost, I want to thank my supervisor Prof. Dr. Julija Zavadlav for giving me the opportunity to pursue my doctoral research in this exciting field and for providing me with the academic freedom to follow my research interests. Your guidance, unwavering support and our regular discussions have been instrumental in the accomplishment of this work.

I would also like to extend my sincere appreciation to the members of the examination board: Prof. Dr. Matej Praprotnik and Prof. Dr. Phaedon-Stelios Koutsourelakis for their participation as examiners, as well as Prof. Dr.-Ing. Nikolaus A. Adams for his mentorship and for chairing the examination committee.

Additionally, I want to thank the co-authors of the articles included in this thesis, Prof. Dr. Julija Zavadlav, Prof. Dr. Matej Praprotnik, Maximilian Stupp, Gregor Döhner, Paul Fuchs, and Ana Cukarska, for their valuable contributions to the published works and our numerous productive discussions.

I would also like to extend my gratitude to my former and current colleagues at the Multiscale Modeling of Fluid Materials Group, namely Christoph Näger, Maximilian Stupp, Sebastien Röcken, and Elena Humphries. Our time together in the office and at group events, along with our collaborative and harmonious working environment, have been truly enjoyable. I particularly want to acknowledge Elena Humphries, probably TUM's best Team Assistant, for her support in reducing my administrative workload.

Moreover, I want to thank my academic collaborators outside the MFM group: First, the Analytics and Machine Learning Group with Prof. Dr. Stephan Günnemann, Bertrand Charpentier, Nicholas Gao, and Tom Wollschläger, for our ambitious work on uncertainty quantification and the dedication in securing the MDSI Seed Funds. Second, the Simulation of Nanosystems for Energy Conversion group, including Prof. Dr. Alessio Gagliardi, Dr. Siby Thomas, and Felix Mayr, for our joint deep-dive into metal-organic frameworks and the perseverance in dealing with challenging data sets. Last, the whole Computational Science & Engineering Laboratory group headed by Prof. Dr. Petros Koumoutsakos, for their warm hospitality, engaging on- and off-topic conversations, and their contribution to the stimulating environment I experienced during my stay at the Harvard School of Engineering and Applied Sciences.

I would also like to acknowledge all the students whom I had the pleasure of supervising during my doctorate. Our collaborative work served as an excellent platform for testing new research ideas and allowed me to further develop my supervision and project management skills.

I also thank my entire family for their unwavering support during my academic studies.

Last, but certainly not least, I want to extend a heartfelt thank you to my partner Linda: Your support, understanding, and companionship have been invaluable during the stressful times of this doctorate journey. You have made the victories along the way all the more meaningful, and I am truly grateful to have you in my life – thank you for everything!

A. Supplementary Information

A.1. Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting

The following section contains the supporting information for ref. 2 embedded in section 3.1.1.

Learning neural network potentials from experimental data via Differentiable Trajectory Reweighting: Supplementary Information

Stephan Thaler[1] and Julija Zavadlav[1,2]

[1] *Professorship of Multiscale Modeling of Fluid Materials, TUM School of Engineering and Design, Technical University of Munich, Germany*
[2] *Munich Data Science Institute, Technical University of Munich, Germany*

Contents

Supplementary Methods

1. DiffTRe and simulation parameters

First, we summarize DiffTRe parameters relevant to all examples before we list problem-specific parameters below. We have set $\bar{N}_{\text{eff}} = 0.9N$ as the threshold above which re-using a trajectory is allowed. We employ an Adam optimizer [1] with exponentially decaying learning rate. Adam hyperparameters $\beta_1 = 0.1$ and $\beta_2 = 0.4$ are chosen to account for training with rather large step sizes and only few parameter updates. All examples are initialized with a global random seed 0, which controls the random initialization of $\boldsymbol{\theta}$ and the initial simulation state. We observed that despite setting random seeds, results are not matched exactly across different re-runs – even when running JAX on reproducibility configuration. We tackle this issue by reporting results for varying random seeds that also capture variability from non-deterministic operations. All computations are run on a single Nvidia RTX 3090 GPU with the exception of computations with the cubic spline potential in the double-well toy example. As the numerically inexpensive spline cannot saturate the GPU, computations were faster on an AMD Ryzen Threadripper 3070X CPU.

1.1. Double-well toy example

Simulations consist of $N_p = 2000$ ideal gas particles of mass $m = 1$ within a box of size $X = 1$ and time step $\delta t = 0.001$. The constant temperature of $k_B T = 1$ in the canonical ensemble is enforced by a Nose-Hoover chain thermostat [2] with 5 chains and time scale $\tau = 0.02$. The initial state $\mathbf{S}_{\text{init}}$ is constructed by randomly drawing particles uniformly from $x \in [0, 1]$. $\mathbf{S}_{\text{init}}$ for the final production run consists of particles drawn uniformly from $x \in [0.5, 0.51]$ to test convergence to the target density distribution, even from a state far from equilibrium. Density distributions are computed via the differentiable density function in Supplementary Eq. (4) with bin width

$\Delta x = 0.01$. During optimization, the initial learning rate $\eta = 0.5$ of Adam [1] is decayed exponentially by a factor of 0.01 over 200 update steps. The target and final predicted densities $\tilde{\rho}(x)/\rho_0$ and $\rho(x)/\rho_0$ are computed based on a production run of 100000 states following 10000 skipped states for equilibration.

1.2. Atomistic model of diamond

Simulations are run with a time step size of $\delta t = 0.5$ fs. The temperature is controlled by a Langevin thermostat with friction coefficient $\gamma = 4/$ ps, which corresponds to a coupling time scale of $250 fs$. These values are common in simulations of diamond in the literature [3]. Carbon atoms have a mass $m = 12.011$ u. The loss weights $\gamma_{\boldsymbol{\sigma}} = 5 \cdot 10^{-8} (\frac{\text{kJ}}{\text{mol nm}^3})^{-2}$ and $\gamma_{\mathbf{C}} = 10^{-10} (\frac{\text{kJ}}{\text{mol nm}^3})^{-2}$ balance the impact of both observables, i.e. stress $\boldsymbol{\sigma}$ and stiffness values C_{ij}. Optimization starts with an initial Adam learning rate $\eta = 0.002$ that is exponentially decayed by a factor of 0.01 over 500 steps.

In computation of phonon density of states (PDOS), we minimize the potential energy via 500 steps of the Fast Inertial Relaxation Engine (FIRE) [4]. PDOS is computed afterwards via the finite displacement method as implemented in Phonopy [5] with displacement length 0.001 nm.

1.3. Coarse-grained water model

Coarse-grained water is simulated with a time step size of $\delta t = 2$ fs. Water molecules (and CG water particles correspondingly) have a mass $m = 18.0154$ u. A Nose-Hoover chain thermostat [2] with chain length 5 and time scale $\tau = 200$ fs enforces the target temperature. We approximate radial (RDF) and angular distribution functions (ADF) with the differentiable versions presented in Supplementary Eq. (5) and (6). The RDF is discretized by 300 bins of width $\Delta x = \frac{1}{300}$ nm. The ADF is discretized by 200 bins of width $\Delta \alpha = \frac{\pi}{200}$ rad and triplets are cut off at $r_c = 0.318$ nm analogous to the experimental evaluation [6]. The loss weight $\gamma_p = 10^{-7} (\frac{\text{kJ}}{\text{mol nm}^3})^{-2}$ accounts for the larger magnitude of pressure versus the RDF and ADF. The initial Adam learning rate $\eta = 0.003$ is decayed exponentially by a factor of 0.01 over 200 steps.

The tetrahedral order parameter q [7] is computed via the triplet angles α_{ijk} spanned by neighboring particles i and k of a central particle j. i and k are indices running over the 4 nearest neighbors of particle i and

$$q = 1 - \frac{3}{8} \sum_{i=1}^{3} \sum_{k=i+1}^{4} \left(\cos \alpha_{ijk} + \frac{1}{3} \right)^2 . \tag{1}$$

We compute the self-diffusion coefficient D via the Green-Kubo relation from the velocity auto-correlation function (VACF)

$$D = \frac{1}{3} \int_0^{t_{\text{cut}}} \left\langle \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{v}_i(t_0) \cdot \mathbf{v}_i(t_0 + \tau) \right\rangle_{t_0} \mathrm{d}\tau , \tag{2}$$

where we cut the VACF at $t_{\text{cut}} = 1$ ps to reduce the effect of spurious long-term non-zero correlations. N_p is the number of particles in the box.

2. Speed-up considerations

Assuming a numerically expensive (NN) potential dominating computational effort, s_g is determined by the cost of necessary force evaluations during trajectory generation per retained state energy computation: As forces for NN potentials are computed by backpropagating potential energy values, they are approximately G times as expensive as energy computations. The provided rule-of-thumb formula in the main text overestimates s_g for expensive observables, but systematically underestimating s_g by ignoring the cost of backpropagating through time integrator operations. Recognizing that gradient computation costs with DiffTRe are negligible compared to reference trajectory generation costs (under the same assumption of numerically cheap observables), $s \sim G + 1$ reflects the cost of trajectory generation plus backward pass versus only the trajectory generation in the case of DiffTRe. We presumed a value of $G \approx 3$ for the given estimates in the toy example, which mirrors that gradient computation in the adjoint method requires integrating 3 ordinary differential equations backwards in time [8].

3. Continuously differentiable binning

The (discrete) Dirac function used in binning can be substituted by a Gaussian probability density function (PDF) centered at position x_k of binned entity k. The value of bin $b_k(x)$ centered at x can be computed as

$$b_k(x) = \Delta x * s_k(x) \quad \text{with} \quad s_k(x) = \frac{1}{\sqrt{2\pi\delta^2}} e^{-\frac{(x-x_k)^2}{2\delta^2}} \ , \tag{3}$$

where Δx is the bin width. The implied discrete integral over a PDF guarantees an overall contribution of unity for each binned entity. We set the Gaussian standard deviation $\delta = \Delta x$. For a fine grid $\delta \to 0$, the Dirac function is recovered.

Eq. (3) allows defining a normalized differentiable density function

$$\rho(x) \simeq \frac{1}{N_p} \sum_{k=1}^{N_p} b_k(x) \ , \tag{4}$$

where x_k is the position of each particle in the simulation and N_p is the number of particles in the box. Analogously, we can define

$$RDF(d) \simeq \frac{\Omega}{V(d) N_p^2} \sum_{k=1}^{N_{\text{pair}}} b_k(d) \ , \tag{5}$$

where $V(d)$ is the volume of the sphere shell of $b_k(d)$ and Ω is the simulation box volume.

The ADF is a probability density function (PDF) over triplet angles α_{ijk} for all particle triplets ijk within a cut-off radius r_c of central particle j. We smooth the radial cut-off via a Gaussian cumulative distribution function (CDF) $\Phi(r; r_c, \sigma^2)$ centered at r_c with variance σ^2.

$$ADF(\alpha) \simeq \frac{\overline{ADF}(\alpha)}{\int_0^\pi \overline{ADF}(\alpha) \mathrm{d}\alpha} \quad \text{with} \quad \overline{ADF}(\alpha) = \sum_{k=1}^{N_{\text{triplet}}} (1 - \Phi(r_{k,\max}; r_c, \sigma^2)) b_k(\alpha) \ , \tag{6}$$

where $r_{k,\max} = \max(r_{ij}, r_{kj})$.

4. Stress-strain relations

Voigt notation provides a convenient way to describe the stress-strain relation by reducing pairs of indices to single digits: 11 ↦ 1, 22 ↦ 2, 33 ↦ 3, 23 ↦ 4, 13 ↦ 5, and 12 ↦ 6. Generalized Hooke's law can then be written as

$$\sigma_i = \mathbf{C}_{ij} \epsilon_j \quad \text{with} \quad \boldsymbol{\sigma} = \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{pmatrix} \ ; \ \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix} = \begin{pmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ 2\epsilon_{23} \\ 2\epsilon_{13} \\ 2\epsilon_{12} \end{pmatrix} \ , \tag{7}$$

assuming $\boldsymbol{\sigma} = \mathbf{0}$ for $\boldsymbol{\epsilon} = \mathbf{0}$. Due to the symmetry in the diamond cubic crystal system, Eq. (7) simplifies to only 3 distinct values in $\mathbf{C}$

$$\begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} = \begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ & C_{11} & C_{12} & 0 & 0 & 0 \\ & & C_{11} & 0 & 0 & 0 \\ & & & C_{44} & 0 & 0 \\ & \text{sym} & & & C_{44} & 0 \\ & & & & & C_{44} \end{pmatrix} \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix} . \tag{8}$$

The inverse relation is defined by the compliance tensor $\mathbf{S} = \mathbf{C}^{-1}$, which is usually given in terms of Young's modulus E, shear modulus G and Poisson's ratio ν

$$\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{pmatrix} = \begin{pmatrix} \frac{1}{E} & \frac{-\nu}{E} & \frac{-\nu}{E} & 0 & 0 & 0 \\ & \frac{1}{E} & \frac{-\nu}{E} & 0 & 0 & 0 \\ & & \frac{1}{E} & 0 & 0 & 0 \\ & & & \frac{1}{G} & 0 & 0 \\ & \text{sym} & & & \frac{1}{G} & 0 \\ & & & & & \frac{1}{G} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{pmatrix} . \tag{9}$$

The stress-strain curves are computed by deforming the box in two separate modes that yield states of pure normal and shear strain, respectively [9]. In the normal mode, we transform the box according to

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X(1+\xi) \\ Y \\ Z \end{pmatrix}, \tag{10}$$

which yields the non-zero strain $\epsilon_1 = \epsilon_{11} = \xi$ in the strain vector $\boldsymbol{\epsilon} = (\epsilon_1, 0, 0, 0, 0, 0)$. A pure shear mode is given by the transformation

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \rightarrow \begin{pmatrix} X \\ Y + Z\xi \\ Z \end{pmatrix}, \tag{11}$$

which yields $\epsilon_4 = 2\epsilon_{23} = \xi$ in the strain vector $\boldsymbol{\epsilon} = (0, 0, 0, \epsilon_4, 0, 0)$. These elementary deformations [9] allow probing $\mathbf{C}$ such that a single component of $\mathbf{C}$ describes the relation between ϵ_i and measured σ_j (Eq. (8))

$$\sigma_1 = C_{11}\epsilon_1 \ ; \quad \sigma_2 = C_{12}\epsilon_1 \ ; \quad \sigma_4 = C_{44}\epsilon_4 \ . \tag{12}$$

5. Derivation of the gradient

$$L(\boldsymbol{\theta}) = (\langle O(U_{\boldsymbol{\theta}})\rangle - \tilde{O})^2 \simeq \left(\sum_{i=1}^{N} w_i O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) - \tilde{O}\right)^2 = \bar{L}(\boldsymbol{\theta}) \tag{13}$$

$$\frac{\partial \bar{L}}{\partial \boldsymbol{\theta}} = 2\Bigg(\underbrace{\sum_{i=1}^{N} w_i O(\mathbf{S}_i, U_{\boldsymbol{\theta}})}_{\simeq\langle O(U_{\boldsymbol{\theta}})\rangle} - \tilde{O}\Bigg) \frac{\partial}{\partial \boldsymbol{\theta}} \sum_{i=1}^{N} w_i O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) \tag{14}$$

$$\sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} (w_i O(\mathbf{S}_i, U_{\boldsymbol{\theta}})) = \sum_{i=1}^{N} \frac{\partial w_i}{\partial \boldsymbol{\theta}} O(\mathbf{S}_i, U_{\boldsymbol{\theta}}) + \underbrace{\sum_{i=1}^{N} w_i \frac{\partial O(\mathbf{S}_i, U_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}}_{\simeq\langle \frac{\partial O(U_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}}\rangle} \tag{15}$$

$$\begin{aligned}
\frac{\partial w_i}{\partial \boldsymbol{\theta}} &= \frac{\partial}{\partial \boldsymbol{\theta}} \left(\frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{\sum_{j=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}} \right) \\
&= \underbrace{\frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{\sum_{j=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}}}_{w_i} \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_i)}{\partial \boldsymbol{\theta}} \right) + \frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{-\left(\sum_{j=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}\right)^2} \\
&* \sum_{j=1}^{N} \left[e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))} \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_j)}{\partial \boldsymbol{\theta}} \right) \frac{\sum_{k=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_k) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_k))}}{\sum_{k=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_k) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_k))}} \right] \\
&= w_i \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_i)}{\partial \boldsymbol{\theta}} \right) + \frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{-\left(\sum_{j=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}\right)^2} \\
&* \sum_{k=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_k) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_k))} \sum_{j=1}^{N} \Bigg[\left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_j)}{\partial \boldsymbol{\theta}} \right) \underbrace{\frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}}{\sum_{k=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_k) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_k))}}}_{w_j} \Bigg] \\
&= w_i \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_i)}{\partial \boldsymbol{\theta}} \right) + \underbrace{\frac{e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_i) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_i))}}{-\sum_{j=1}^{N} e^{-\beta(U_{\boldsymbol{\theta}}(\mathbf{S}_j) - U_{\hat{\boldsymbol{\theta}}}(\mathbf{S}_j))}}}_{-w_i} * \sum_{j=1}^{N} w_j \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_j)}{\partial \boldsymbol{\theta}} \right)
\end{aligned} \tag{16}$$

$$\sum_{i=1}^{N} \frac{\partial w_i}{\partial \boldsymbol{\theta}} O(\mathbf{S}_i, \boldsymbol{\theta}) = \sum_{i=1}^{N} w_i \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_i)}{\partial \boldsymbol{\theta}} \right) O(\mathbf{S}_i, \boldsymbol{\theta}) + \sum_{i=1}^{N} -w_i O(\mathbf{S}_i, \boldsymbol{\theta}) \sum_{j=1}^{N} w_j \left(-\beta \frac{\partial U_{\boldsymbol{\theta}}(\mathbf{S}_j)}{\partial \boldsymbol{\theta}} \right) \tag{17}$$

$$\simeq \langle -\beta \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} O(U_{\boldsymbol{\theta}}) \rangle + \langle -O(U_{\boldsymbol{\theta}}) \rangle \langle -\beta \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \rangle \tag{18}$$

$$\Rightarrow \frac{\partial \bar{L}}{\partial \boldsymbol{\theta}} \simeq 2(\langle O(U_{\boldsymbol{\theta}}) \rangle - \tilde{O}) \left[\langle \frac{\partial O(U_{\boldsymbol{\theta}})}{\partial \boldsymbol{\theta}} \rangle - \beta \left(\langle O(U_{\boldsymbol{\theta}}) \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \rangle - \langle O(U_{\boldsymbol{\theta}}) \rangle \langle \frac{\partial U_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}} \rangle \right) \right] = \frac{\partial L}{\partial \boldsymbol{\theta}} \tag{19}$$

6. DimeNet++ hyperparameters

We refer the reader to the original DimeNet / DimeNet++ publications [10, 11] for a detailed description of the neural network architecture. We reduced embedding sizes by factor 4: The standard embedding size then becomes 32, the output embedding size 64, the triplet and atom-type embedding size becomes 16 and the Bessel-basis embedding remains at a size of 8. All other hyperparameters are unchanged: A cut-off length of 0.5 nm (0.2 nm for diamond), 4 interaction layers, 3 fully-connected output layers, 1 residual block before and 2 residual blocks after the skip connection, 6 radial and 7 angular Bessel embedding function with a continuously differentiable envelope function of order 6 and a swish [12] activation function. Weights are initialized via an orthogonal Glorot[13, 10] scheme.

Supplementary Figures

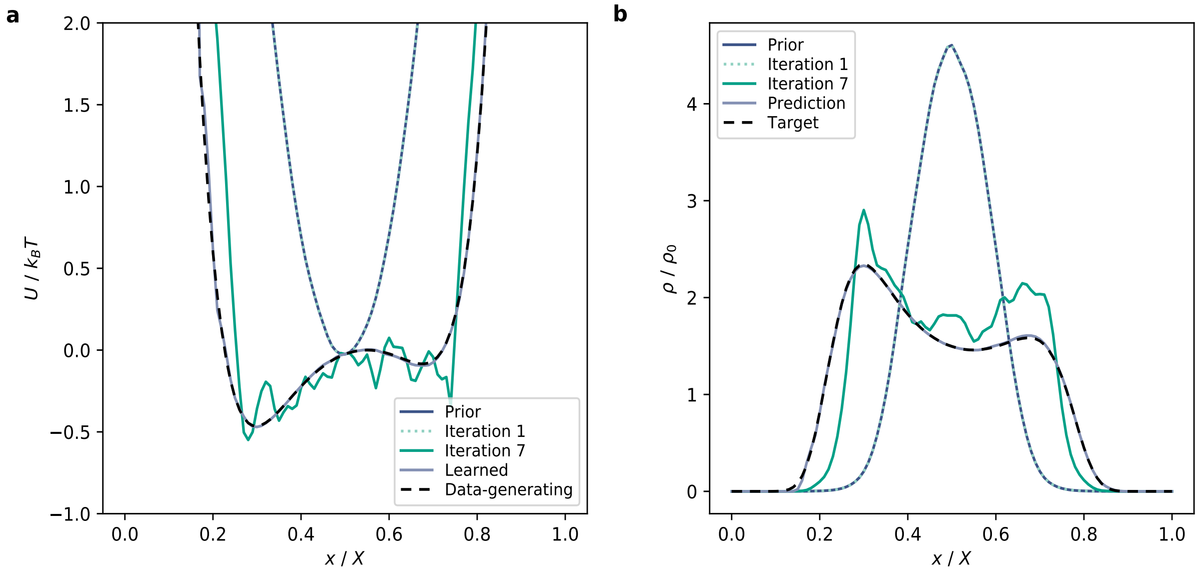


Figure 1: Double-well toy example across optimization. By learning the normalized density, DiffTRe adjusts $U_{\boldsymbol{\theta}}^{\text{model}}$ such that $U^{\text{prior}} + U_{\boldsymbol{\theta}}^{\text{model}}$ eventually recovers the data-generating potential (**a**). Accordingly, the corresponding predicted normalized densities converge to the target (**b**). Potentials in panel **a** are shifted vertically for visualization purposes such that all potentials coincide at $x/X = 0.5$.

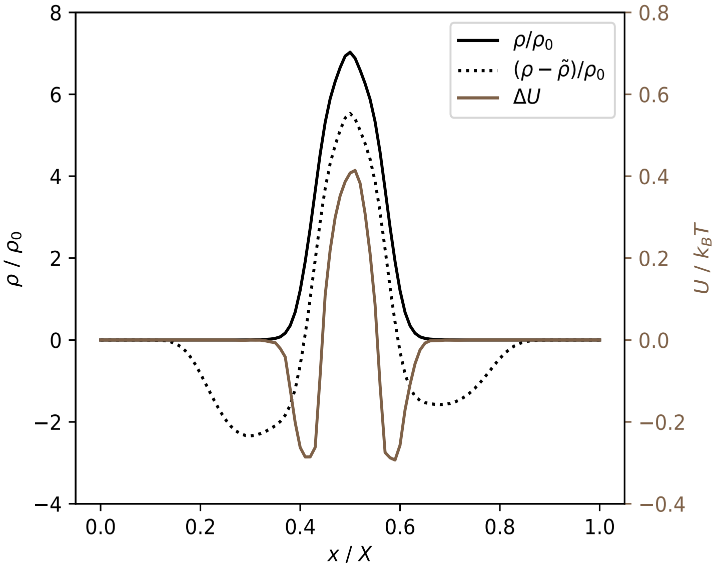


Figure 2: Double-well toy example vanishing gradients. There are areas on the potential energy surface (PES) where the effect of the gradient on the PES $\Delta U = U(\boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} L) - U(\boldsymbol{\theta}) = 0$, even though these areas contribute to the loss ($\rho - \tilde{\rho} \neq 0$). This is due to the reference trajectory that contains no states in these areas of the PES ($\rho = 0$ and $\nabla_{\boldsymbol{\theta}} \rho = 0$; compare Supplementary Eq. (19)).

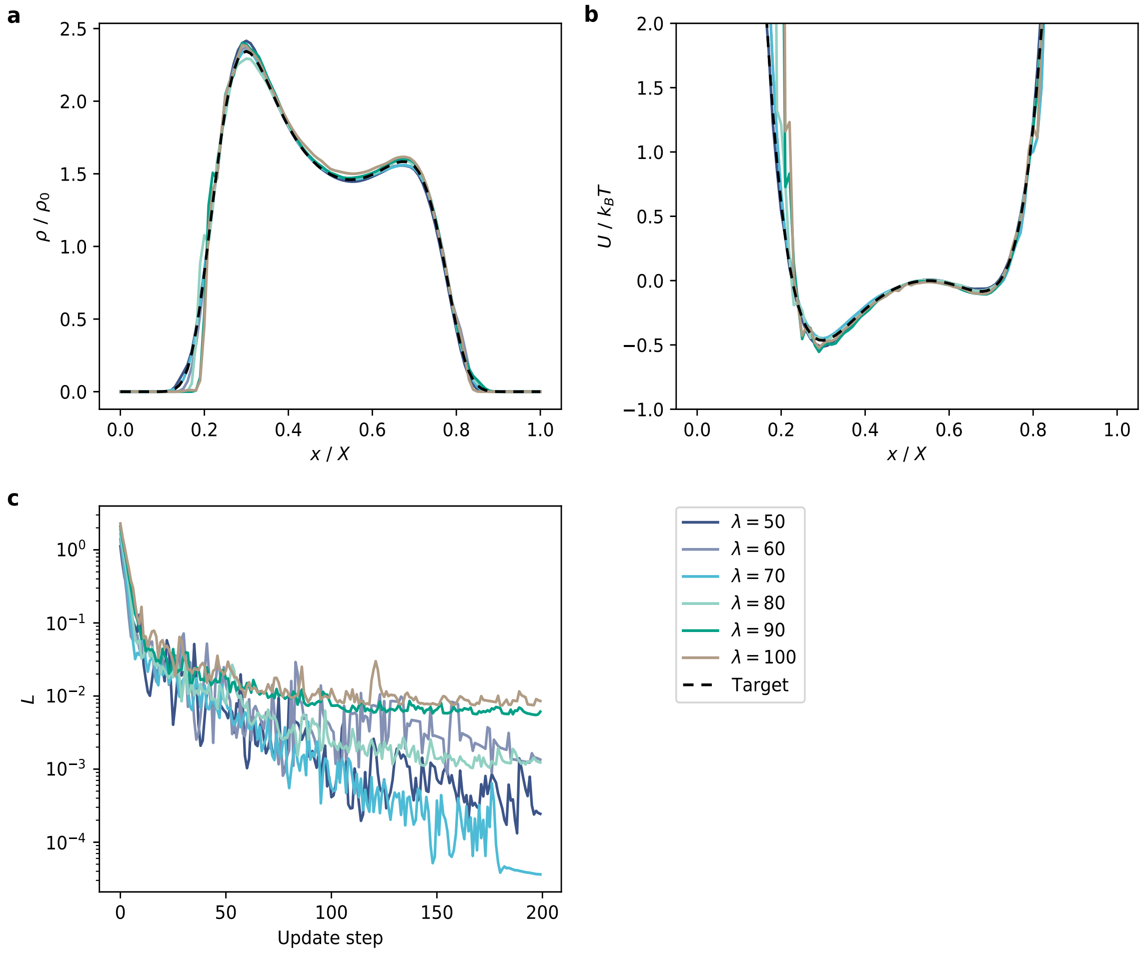


Figure 3: Double-well prior variation study. Resulting density (**a**) and learned potential (**b**) with respective targets for varying prior scales λ. The prior loss curves (**c**) show the impact of the prior on the initial loss value $L(0)$ and optimization convergence. Many possible λ lead to satisfactory learning outcomes ($\mathbf{a} - \mathbf{b}$).

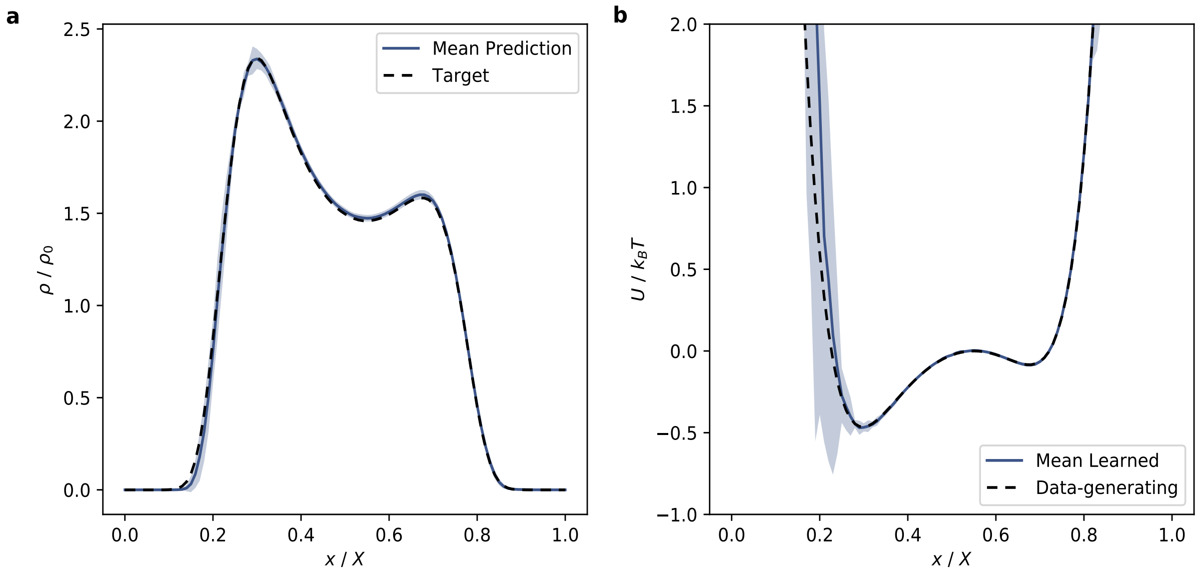


Figure 4: Random initialization study for the double-well toy example. A mean matching the target and small standard deviations (shaded area) when re-starting the optimization with random seeds from $0 - 99$ demonstrates that the learned normalized density profile is robust with respect to initialization of the spline and the initial simulation state (**a**). The corresponding learned potential exhibits larger standard deviations at the left well boundary due to difficult training in this region (**b**). Potentials are shifted vertically for visualization purposes such that all potentials coincide at $x/X = 0.5$.

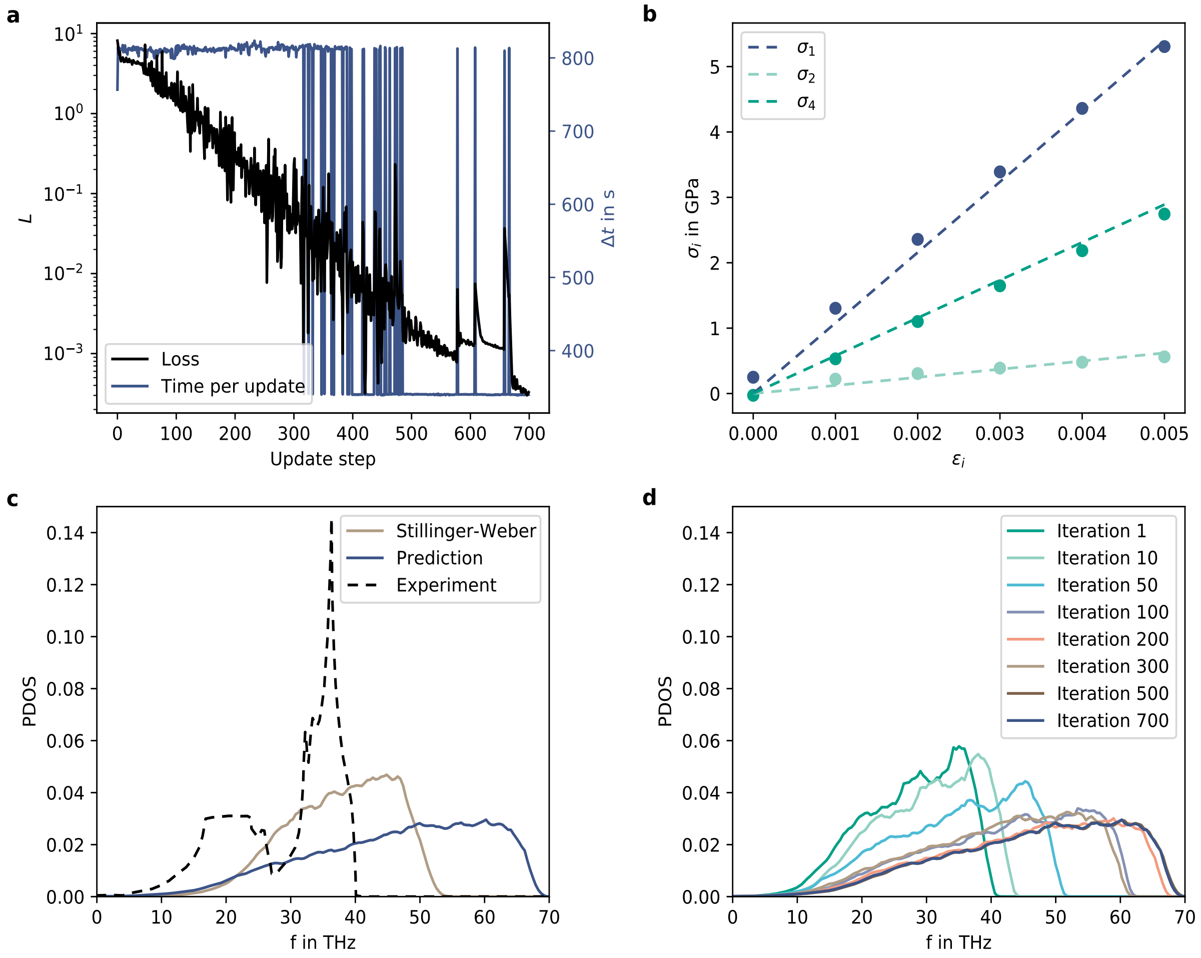


Figure 5: Supplementary results for the diamond model. The large reduction in the loss L confirms successful learning (**a**). Reduction in wall-clock time per parameter update Δt in the second half of the optimization is achieved through re-using previously generated trajectories. Panel **b** displays an alternative stiffness computation method, explicit box deformation. Assuming a linear stress-strain relationship for small $\boldsymbol{\epsilon}$ and a perfect alignment of the learned potential with experimental $\tilde{\boldsymbol{\sigma}} = \mathbf{0}$ and $\tilde{C}_{ij}$, all measured σ_i lie on the respective dashed lines. Hence, both methods for computing stiffness tensor $\mathbf{C}$ give equivalent results and the neural network potential generalizes from the un-strained training box to boxes under small strain. Panel **c** compares the predicted phonon density of states (PDOS) with the experiment [14] and a Stillinger-Weber potential optimized for diamond [15]. The evolution of predicted PDOS over the course of the training is shown in panel **d**.

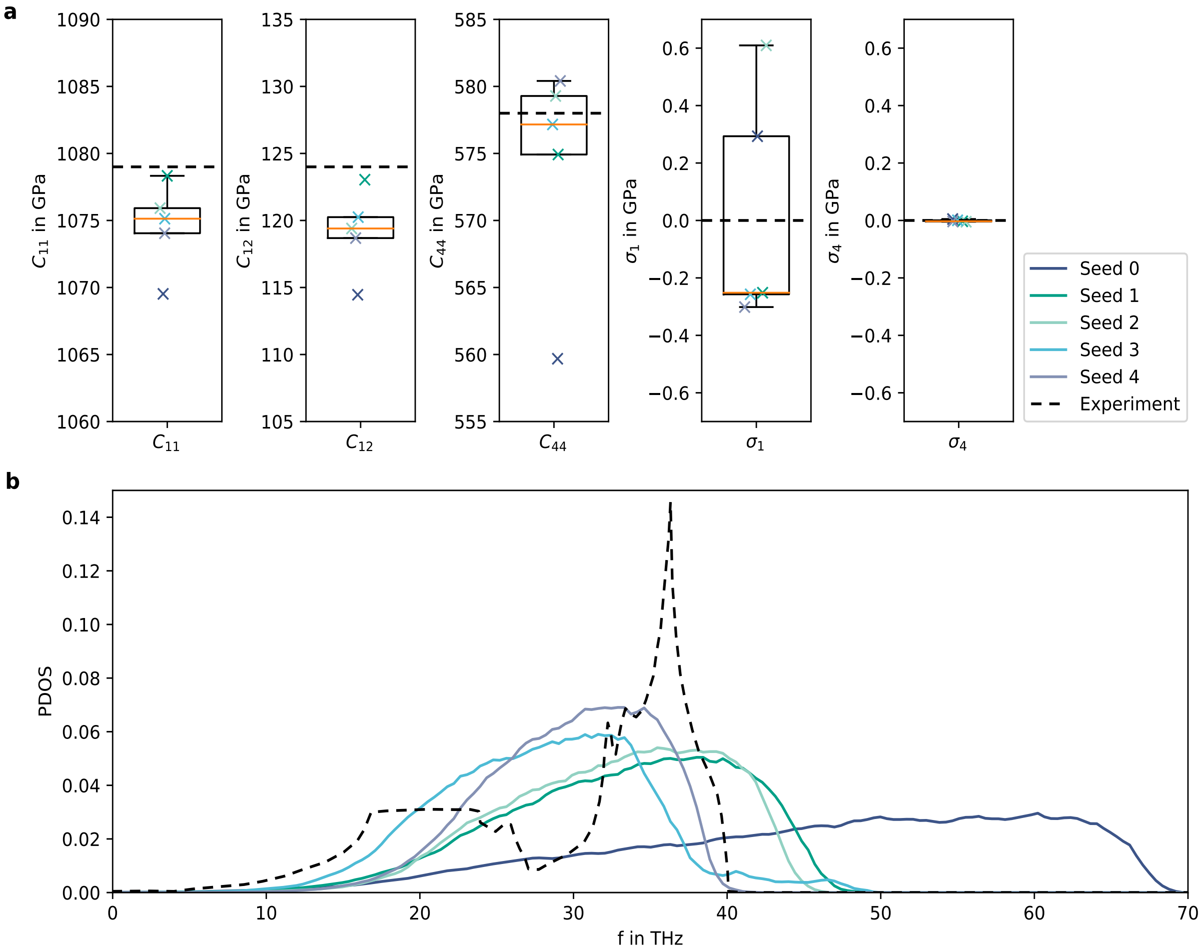


Figure 6: Random initialization study for diamond. For random seeds from 0 to 4 (controlling random initialization of neural network weights as well as initial particle velocities), the predicted observables are distributed closely around their respective targets (**a**). Corresponding predicted phonon densities of states (PDOSs) vary largely across different random seeds (**b**), confirming that different PDOSs are consistent with the target stress and stiffness values. The boxplots in **a** with median (orange line), interquartile range as box limits and whiskers representing 1.5 times interquartile range are added for visualization purposes.

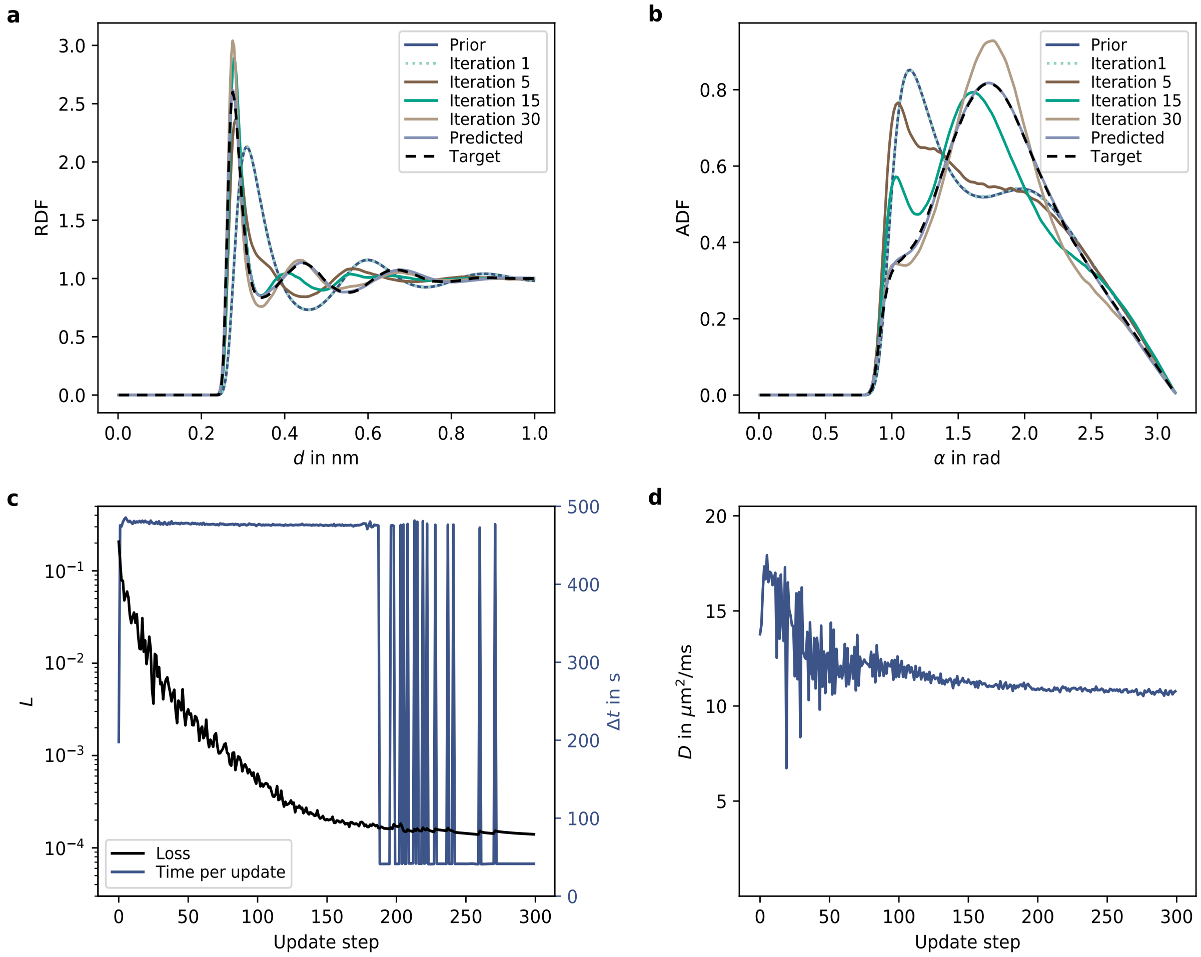


Figure 7: Supplementary results for the coarse-grained water model. Predicted radial distribution functions (RDFs) and angular distribution functions (ADFs) converge from predictions close to the prior to the respective targets (**a** - **b**). Quick reduction in the loss L confirms the learning success (**c**). Significant reduction in wall-clock time per parameter update Δt towards the end of the optimization is achieved through re-using previously generated trajectories. The predicted self-diffusion coefficient D decreases over the course of the optimization (**d**).

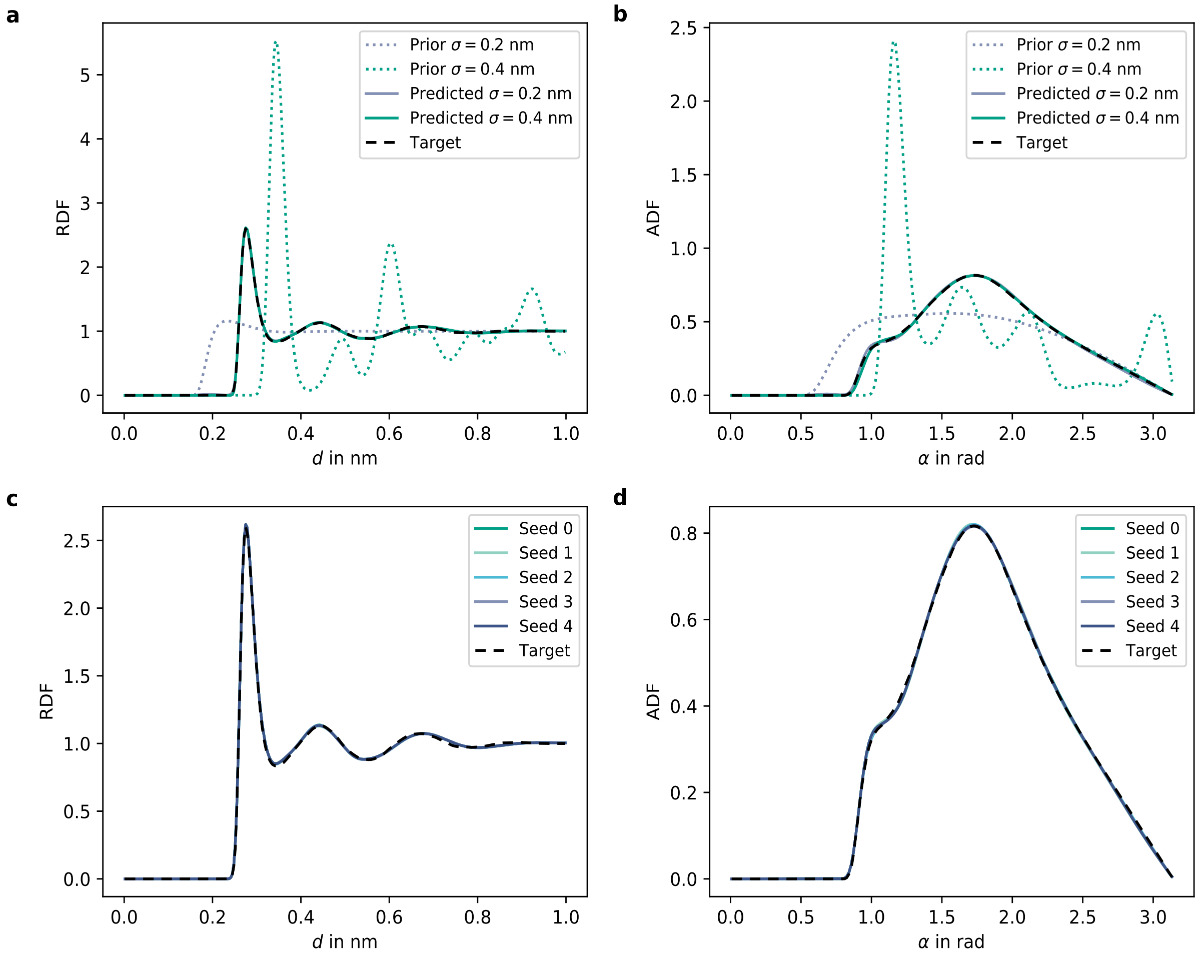


Figure 8: Robustness analysis for coarse-grained water. Predicted target observables are robust to weak choices of U^{prior} ($\mathbf{a} - \mathbf{b}$). These results are obtained using the same hyperparameters as in the reference case $\sigma_R = 0.3165$, except for longer training (1000 steps) with increased learning rate decay factor (0.25) in the case of $\sigma_R = 0.4$ nm. Additionally, predicted target observables are robust to random initialization of NN weights and initial particle velocities ($\mathbf{c} - \mathbf{d}$, $p = 68 \pm 32$ bar).

Supplementary References

[1] Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR* (2015).

[2] Martyna, G. J., Klein, M. L. & Tuckerman, M. Nosé-Hoover chains: The canonical ensemble via continuous dynamics. *J. Chem. Phys.* **97**, 2635–2643 (1992).

[3] Jensen, B. D., Wise, K. E. & Odegard, G. M. The effect of time step, thermostat, and strain rate on reaxff simulations of mechanical failure in diamond, graphene, and carbon nanotube. *J. Comput. Chem.* **36**, 1587–1596 (2015).

[4] Bitzek, E., Koskinen, P., Gähler, F., Moseler, M. & Gumbsch, P. Structural relaxation made simple. *Phys. Rev. Lett.* **97**, 170201 (2006).

[5] Togo, A. & Tanaka, I. First principles phonon calculations in materials science. *Scr. Mater.* **108**, 1–5 (2015).

[6] Soper, A. K. & Benmore, C. J. Quantum differences between heavy and light water. *Phys. Rev. Lett.* **101**, 065502 (2008).

[7] Errington, J. R. & Debenedetti, P. G. Relationship between structural order and the anomalies of liquid water. *Nature* **409**, 318–321 (2001).

[8] Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. Neural Ordinary Differential Equations. In *Advances in Neural Information Processing Systems*, vol. 31 (2018).

[9] Clavier, G. *et al.* Computation of elastic constants of solids using molecular simulation: comparison of constant volume and constant pressure ensemble methods. *Mol. Simul.* **43**, 1413–1422 (2017).

[10] Klicpera, J., Groß, J. & Günnemann, S. Directional Message Passing for Molecular Graphs. In *8th International Conference on Learning Representations, ICLR* (2020).

[11] Klicpera, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and Uncertainty-Aware Directional Message Passing for Non-Equilibrium Molecules (2020). Preprint at `https://arxiv.org/abs/2011.14115`.

[12] Ramachandran, P., Zoph, B. & Le, Q. V. Searching for Activation Functions (2017). Preprint at `https://arxiv.org/abs/1710.05941`.

[13] Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, AISTATS*, 249–256 (2010).

[14] Dolling, G. & Cowley, R. A. The thermodynamic and optical properties of germanium, silicon, diamond and gallium arsenide. *Proc. Phys. Soc.* **88**, 463 (1966).

[15] Barnard, A. S., Russo, S. P. & Leach, G. I. Nearest neighbour considerations in stillinger-weber type potentials for diamond. *Mol. Simul.* **28**, 761–771 (2002).

A.2. Deep coarse-grained potentials via relative entropy minimization

The following section contains the supporting information for ref. 3 embedded in section 3.1.2.

Deep Coarse-grained Potentials via Relative Entropy Minimization: Supporting Information

Stephan Thaler[1], Maximilian Stupp[1], and Julija Zavadlav[1,2]

[1] *Professorship of Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, Germany*
[2] *Munich Data Science Institute & Munich Institute for Integrated Materials, Energy and Process Engineering, Technical University of Munich, Germany*

Contents

Supplementary Methods

We train all models via the Adam optimizer [1] with default parameters, except for the learning rate. We decay the initial learning rate η_0 by an exponential decay schedule such that the η_0 is reduced by a factor 0.01 at the end of the training. For force matching (FM), we set $\eta_0 = 0.001$, the default value of the Adam optimizer. For relative entropy (RE) minimization, we choose a larger step size $\eta_0 = 0.003$ as the model needs to converge within a significantly smaller number of updates. All CG simulations are run in JAX, M.D. [2] on a single Nvidia RTX 3090 GPU.

1. Liquid Water

To generate the atomistic (AT) reference data, we run a LAMMPS [3] simulation with time step 2 fs. Initially, the box size is set by a 1 ns NPT simulation with target pressure of 1 atm. After equilibrating the system for 1 ns in the NVT ensemble, we generate the 10 ns data trajectory.

All CG simulations use a Nose-Hoover [4] thermostat with a chain length of 3, 3 Suzuki-Yoshida steps and a coupling time of 200 fs. We subsample generated trajectories such that a state is retained every 0.1 ps.

For RE training, we approximate the AT average (first term) in eq. 8 via an average over random batches consisting of 700 states from the AT data set. This reduces the overhead from averaging over the whole AT data set significantly. Given that we average over the same number of states as generated by the CG MD simulation, we do not expect a meaningful increase in the statistical error of the computed gradient. Increasing the number of states considered in both averages offers a systematic way to decrease the statistical noise, if necessary.

We discretize the radial distribution function (RDF) with 300 bins, the triplet correlation function (TCF) with 50 bins in each direction and the ADF with 150 bins. For the ADF, we select a triplet cut-off value of 0.318 nm, which is consistent with experimental evaluations [5].

2. Alanin Dipeptide

We generate the AT reference trajectory via a NVT simulation in GROMACS [6] using the AMBER03 [7] force field, which resolves hydrogen atoms. The protein is solvated in TIP3P water. The simulation employs a velociiy-rescaling thermostat [8] with a time constant $\tau = 0.1$ ps and a time step of 2 fs. We equilibrate the system for 1 ns in the NVT ensemble and 1 ns in the NPT ensemble with a barostat pressure of 1 bar, before generating the 100 ns reference trajectory in the NVT ensemble.

All CG simulations use a Langevin thermostat with $\gamma = 100$ ps^{-1} and a time step $\Delta t = 2$ fs. Generated trajectories are subsampled such that a state is retained every 0.2 ps. We discretize the ϕ and ψ density histograms and free energy surfaces via 60 bins each. The data set variation study in fig. 7 follows the default training scheme detailed above with 100 training epochs, except for the small 10 ns data set, which we train for 1000 epochs to have the same number of updates as with the 100 ns reference data set. For the convergence analysis in fig. 8, we increase the learning rate decay factor to 0.1 due to the even smaller number of updates. We note that in practice, a smaller initial learning rate than $\eta_0 = 0.003$ might be appropriate to avoid changing the FM potential beyond the necessary.

Supplementary Figures

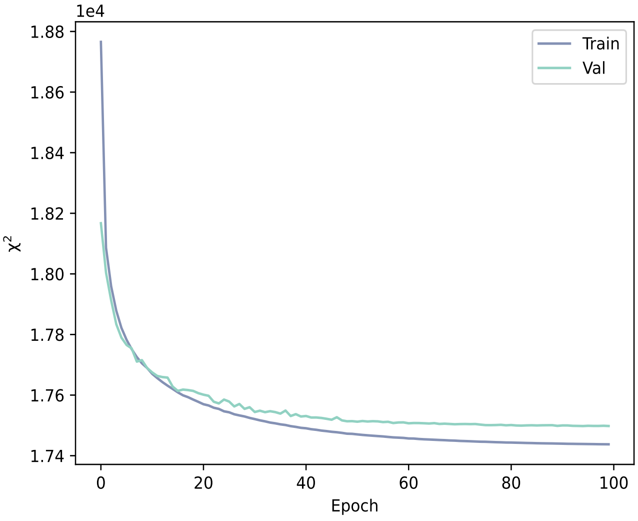


Figure 1: Liquid water loss curves. Per-epoch training and validation loss χ^2 for force matching training of the liquid water model.

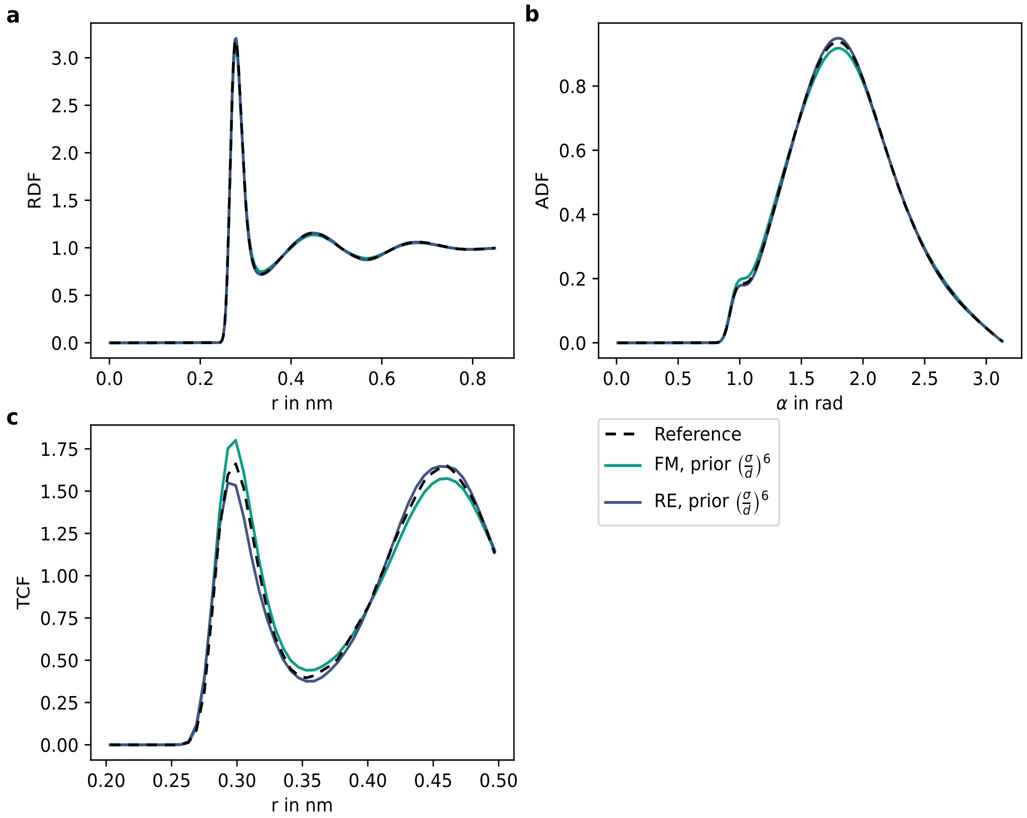


Figure 2: Liquid water prior variation. Resulting (**a**) radial (RDF) and (**b**) angular distribution function (ADF) [5] as well as (**c**) equilateral triplet correlation function (TCF) [9, 10] of models with a prior exponent of 6 trained via force matching (FM) and relative entropy (RE) minimization compared to the atomistic reference.

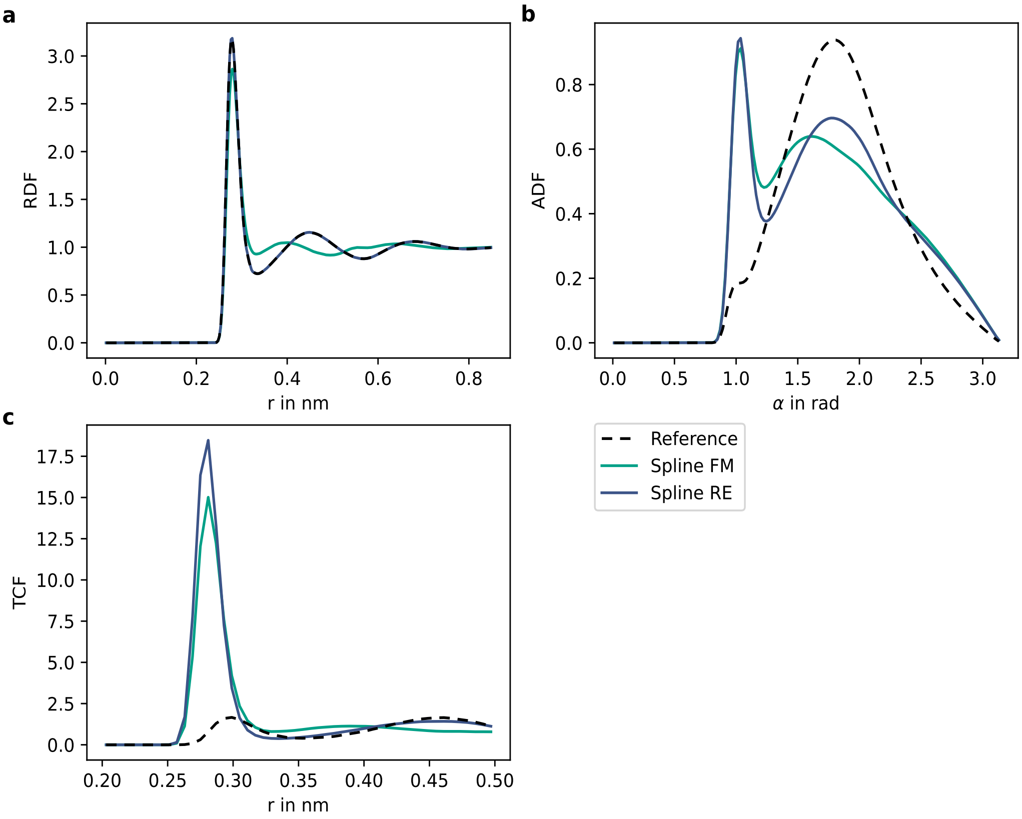


Figure 3: Liquid water spline models. Resulting (**a**) radial (RDF) and (**b**) angular distribution function (ADF) [5] as well as (**c**) equilateral triplet correlation function (TCF) [9, 10] of 2-body cubic spline models trained via force matching (FM) and relative entropy (RE) minimization compared to the atomistic reference.

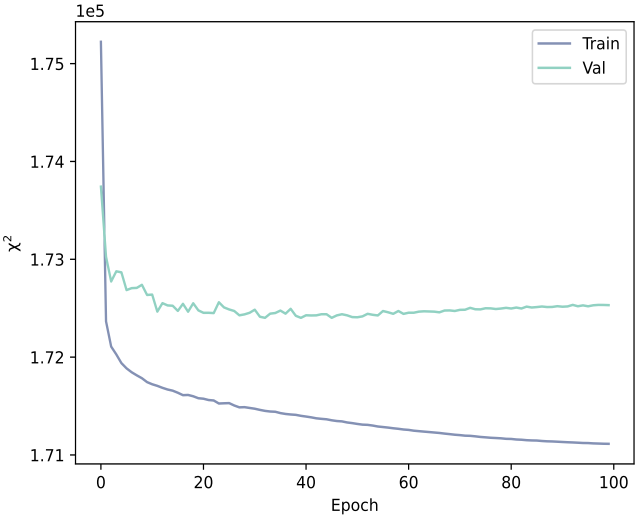


Figure 4: Alanine dipeptide loss curves. Per-epoch training and validation loss χ^2 for force matching training of the alanine dipeptide model.

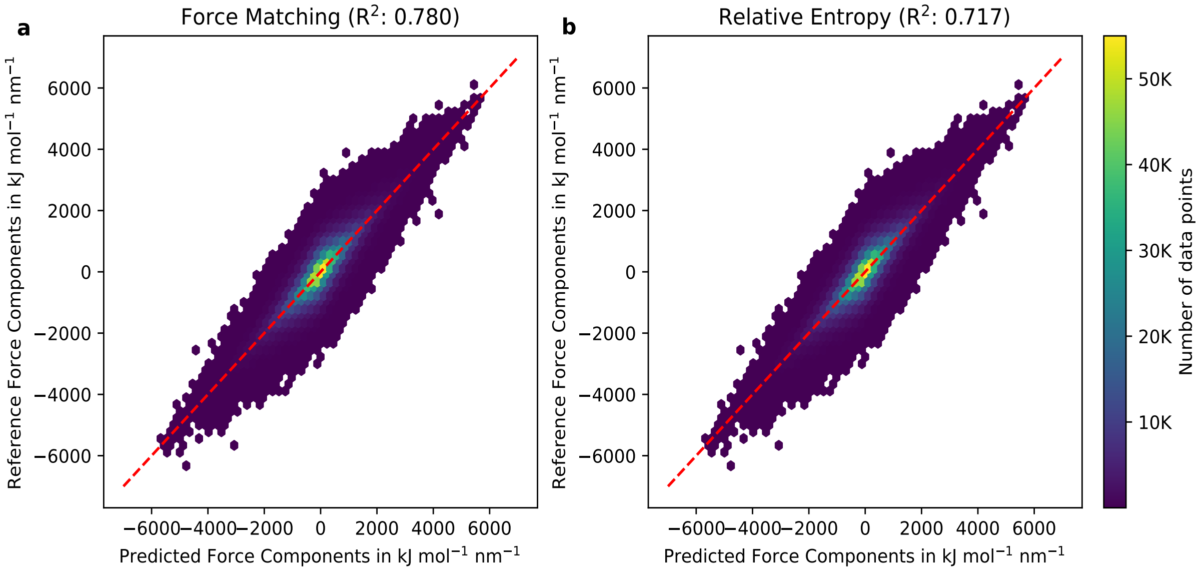


Figure 5: Alanin dipeptide force predictions on test data. Each data point corresponds to a predicted force component for a coarse-grained particle in the test data set compared to its atomistic reference for models trained via (**a**) force matching and (**b**) relative entropy minimization.

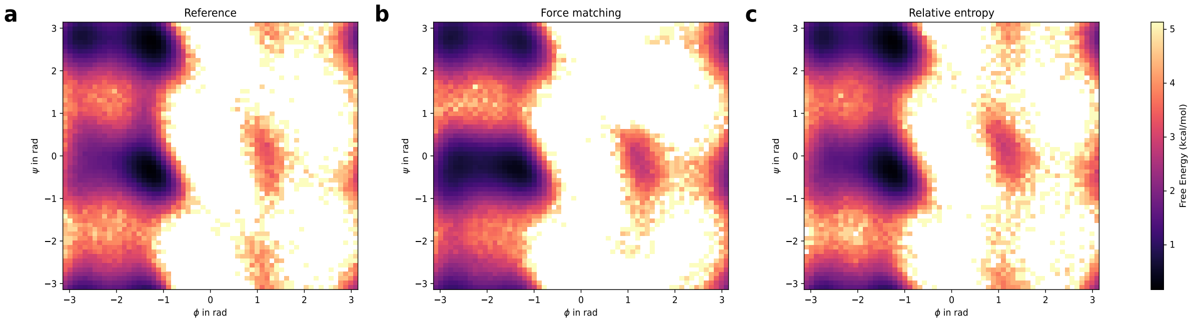


Figure 6: Free energy surface. Resulting free energy surface of the dihedral angles ϕ and ψ from (**a**) the AT reference simulation and from the CG models trained via (**b**) force matching and (**c**) relative entropy minimization.

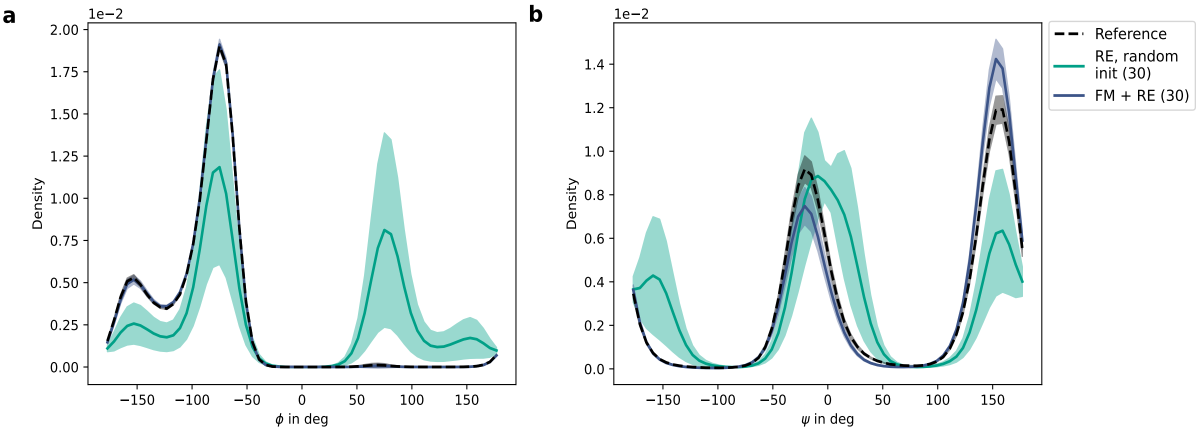


Figure 7: Dihedral angle density. Distribution of dihedral angles (**a**) ϕ and (**b**) ψ as predicted from CG models trained via 30 relative entropy (RE) updates, compared to the atomistic reference. One model is initialized with random parameters and the other one is initialized to the parameters obtained from force matching (FM) pre-training. The mean and standard deviation (shaded area) are computed from 50 trajectories of 100 ns length.

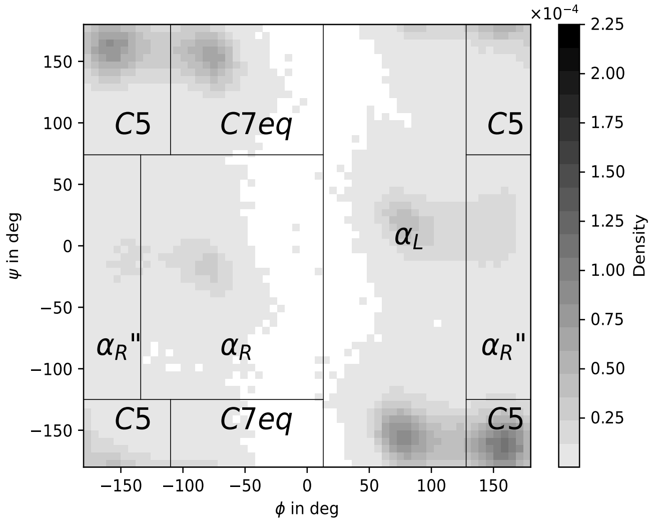


Figure 8: Force matching bond prior Ramachandran diagram. Resulting density histogram of the dihedral angles ϕ and ψ of a single 100 ns trajectory as predicted from the CG model trained via force matching with only bonds as prior potential. This specific trajectory was selected to showcase the undesired behaviour in the α_L region.

Supplementary References

[1] Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR* (2015).

[2] Schoenholz, S. S. & Cubuk, E. D. JAX, M.D.: A Framework for Differentiable Physics. In *Advances in Neural Information Processing Systems*, vol. 33 (2020).

[3] Thompson, A. P. *et al.* Lammps-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **271**, 108171 (2022).

[4] Martyna, G. J., Klein, M. L. & Tuckerman, M. Nosé-Hoover chains: The canonical ensemble via continuous dynamics. *J. Chem. Phys.* **97**, 2635–2643 (1992).

[5] Soper, A. K. & Benmore, C. J. Quantum differences between heavy and light water. *Phys. Rev. Lett.* **101**, 065502 (2008).

[6] Abraham, M. J. *et al.* Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1**, 19–25 (2015).

[7] Duan, C., Y.and Wu *et al.* A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *J. Comput. Chem.* **24**, 1999–2012 (2003).

[8] Bussi, G., Donadio, D. & Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **126**, 014101 (2007).

[9] Bildstein, B. & Kahl, G. Triplet correlation functions for hard-spheres: Computer simulation results. *J. Chem. Phys.* **100**, 5882 (1994).

[10] Dhabal, D., Singh, M., Wikfeldt, K. T. & Chakravarty, C. Triplet correlation functions in liquid water. *J. Chem. Phys.* **141**, 174504 (2014).

A.3. Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls

The following section contains the supporting information for ref. 4 embedded in section 3.2.1.

Scalable Bayesian Uncertainty Quantification for Neural Network Potentials: Promise and Pitfalls - Supporting Information

Stephan Thaler[1], Gregor Doehner[1], and Julija Zavadlav[1,2]

[1] *Professorship of Multiscale Modeling of Fluid Materials, Department of Engineering Physics and Computation, TUM School of Engineering and Design, Technical University of Munich, Germany*
[2] *Munich Data Science Institute, Technical University of Munich, Germany*

Contents

Supplementary Methods

1. Prior Potential

In the common case of generating a data set by sampling from the Boltzmann distribution defined by the underlying high-fidelity model, e.g. via a molecular dynamics (MD) simulation, the resulting data set contains mostly states near energy minima, few high-energy states and no unphysical states such as overlapping atoms. Due to their data-driven nature, NN potentials are unconstrained in phase-space regions not contained in the training data set. When using such a NN potential in MD simulations, the simulation state may reach these unconstrained phase-space regions, resulting in unphysical or highly inaccurate potential energy predictions [1, 2] that often result in numerical instability [3]. This is exacerbated for CG potentials, which are designed to reach time and length scales inaccessible to the data-generating AT model.

Classical potentials avoid this problem by using physics-inspired functional forms [4, 5] that encode a priori known physical principles. For example, the Lennard Jones potential encodes repulsion due to the Pauli exclusion principle at close distances and Van-der-Walls attraction at larger distances. Eq. (11) in the main text casts training the NN potential as Δ-learning [6], where the NN potential corrects an a priori chosen classical potential in phase-space regions where training data are available. With this ansatz, the goal of the prior potential is to enforce qualitatively correct predictions where the NN potential is unconstrained, especially for (unphysical) high-energy states, in order to drive the MD simulation back into the training data distribution, where the NN potential is accurate [7].

For the considered examples with the DimeNet++ [8, 9] potential, the prior potential increases simulation stability significantly compared to the case without it [3]. We found that the specific parameters of the prior potential typically have a minor effect on simulation results [10, 7], assuming the chosen prior successfully restricted the MD simulation from entering unphysical phase-space regions.

2. Lennard Jones Training

We generate 7500 samples using the No-U-Turn Sampler (NUTS) [11], of which 4100 are discarded during warm-up. For the Deep Ensemble [12, 13] and the preconditioned Stochastic Gradient Langevin Dynamics (pSGLD) methods, we employ a batch size of 1 as well as an initial learning rate of 0.01 and a final learning rate of $5 \cdot 10^{-5}$, with all intermediate learning rates being set via a polynomial step size schedule [14]. We train models of the former for 10000 epochs and run chains for 10000 epochs for the latter, where 8000 were discarded as burn-in.

3. Pressure Matching

The pressure P can be computed from the following relation [15]:

$$P = \frac{N_{\mathrm{DOF}} k_{\mathrm{B}} T}{3V} + \frac{\langle W \rangle}{3V} \ , \tag{1}$$

with temperature T, Boltzmann constant k_{B}, volume V, ensemble averaged internal virial $\langle W \rangle$ and number of degrees of freedom in the system N_{DOF}. We augment the FM loss with a virial-matching term [16], which we weight by $w_{\mathrm{P}} = 0.1$

$$L(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{F}}} \sum_{j=1}^{N_{\mathrm{F}}} [F_j - F_{j,\boldsymbol{\theta}}]^2 + \frac{w_{\mathrm{P}}}{N_{\mathrm{box}}} \sum_{k=1}^{N_{\mathrm{box}}} \left[\frac{W_k^i}{3V} - \frac{W_{k,\boldsymbol{\theta}}}{3V} \right]^2 \ , \tag{2}$$

where $W_k^i/(3V)$ is the target instantaneous internal virial term, $W_{k,\boldsymbol{\theta}}/(3V)$ is the internal virial term of state k predicted by the NN potential with parameters $\boldsymbol{\theta}$ and the definition of the first term is given in eq. (9) in the main text.

Similar to approaches in the literature [17], we employ an iterative pressure matching scheme. We adjust the internal virial values of the atomistic (AT) trajectory W_k^{AT} to account for the smaller kinetic energy of the CG system to obtain the initial targets:

$$\frac{W_k^0}{3V} = \frac{W_k^{\mathrm{AT}}}{3V} + \frac{k_{\mathrm{B}} T}{3V} (N_{\mathrm{DOF}}^{\mathrm{AT}} - N_{\mathrm{DOF}}^{\mathrm{CG}}) \ . \tag{3}$$

The iterative scheme then accounts for differences in the pressure due to the distribution shift between the mapped AT trajectory and the trajectory sampled by the CG model:

$$\frac{W_k^{i+1}}{3V} = \frac{W_k^i}{3V} + P^{\mathrm{AT}} - P_{\boldsymbol{\theta}}^{i,\mathrm{CG}} \ , \tag{4}$$

where P^{AT} is the AT reference pressure and $P_{\boldsymbol{\theta}}^{i,\mathrm{CG}}$ is the pressure obtained from a CG MD simulation with model parameters $\boldsymbol{\theta}$ at iteration i. We obtained acceptable models after the third iteration.

Supplementary Figures

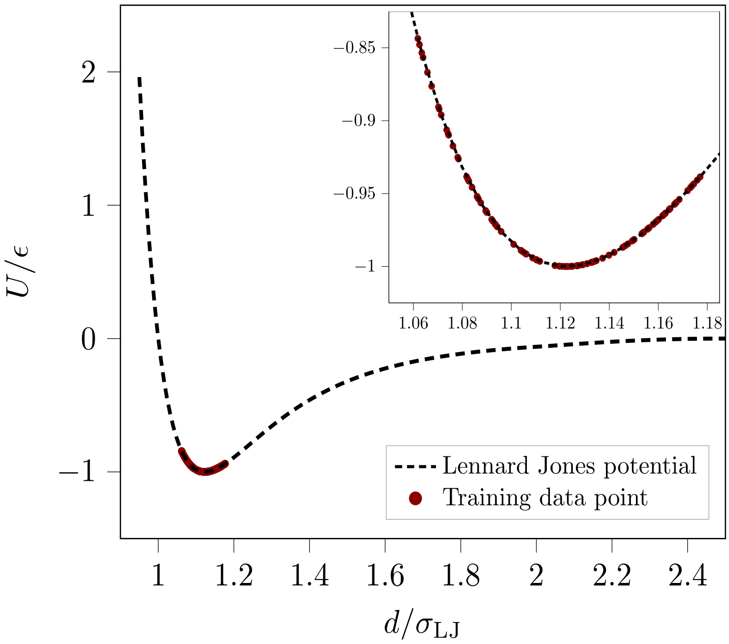
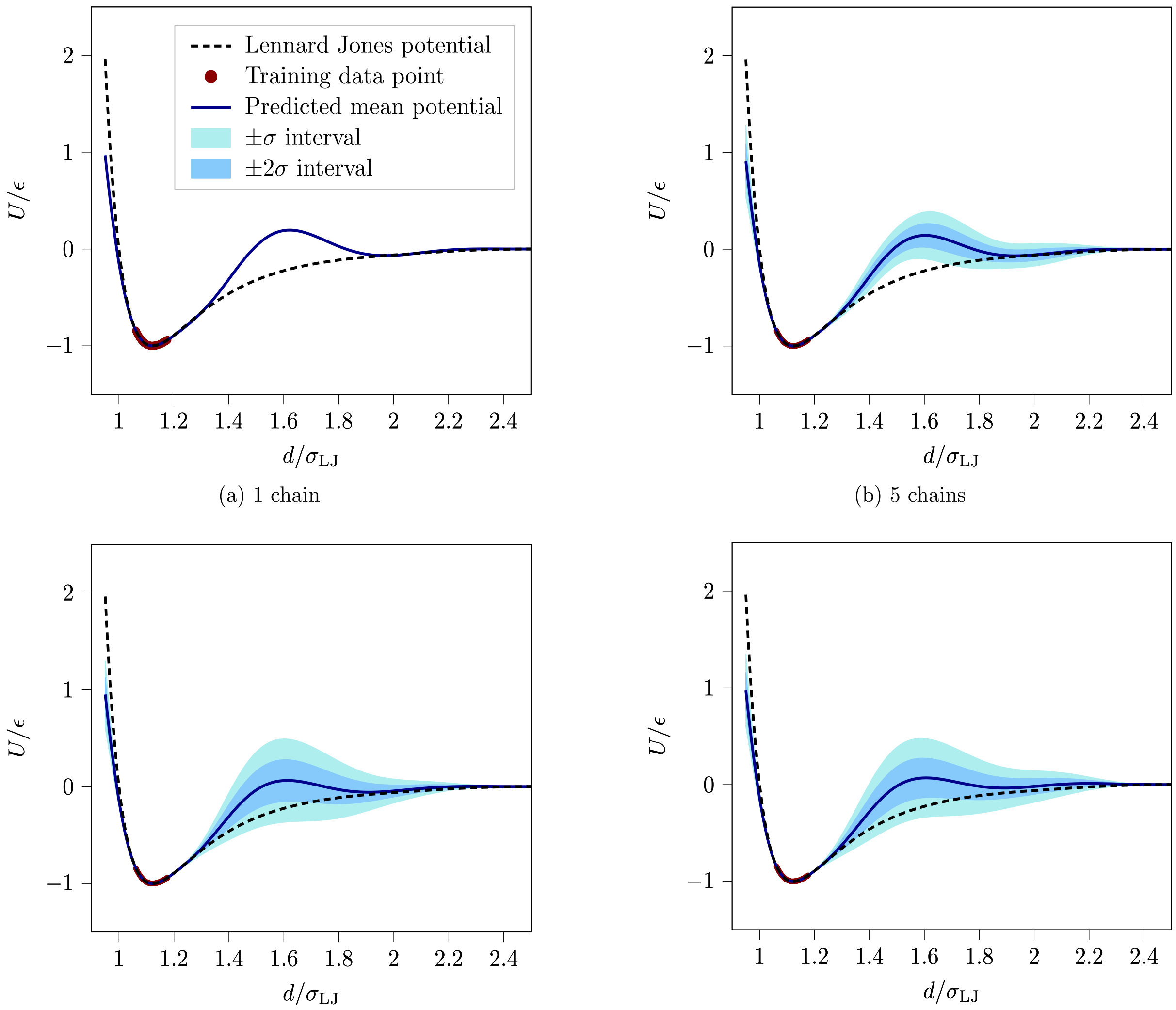


Figure 1: Lennard Jones potential. Data-generating Lennard Jones potential with sampled training data points and zoom on the training interval.



(a) 1 chain

(b) 5 chains

(c) 8 chains

(d) 10 chains

Figure 2: Chain convergence of the No-U-Turn Sampler. Predicted mean potential with ±σ and ±2σ intervals of the No-U-Turn Sampler (NUTS) with samples collected from 1 (a), 5 (b), 8 (c) and 10 chains (d) – compared to the Lennard Jones reference.

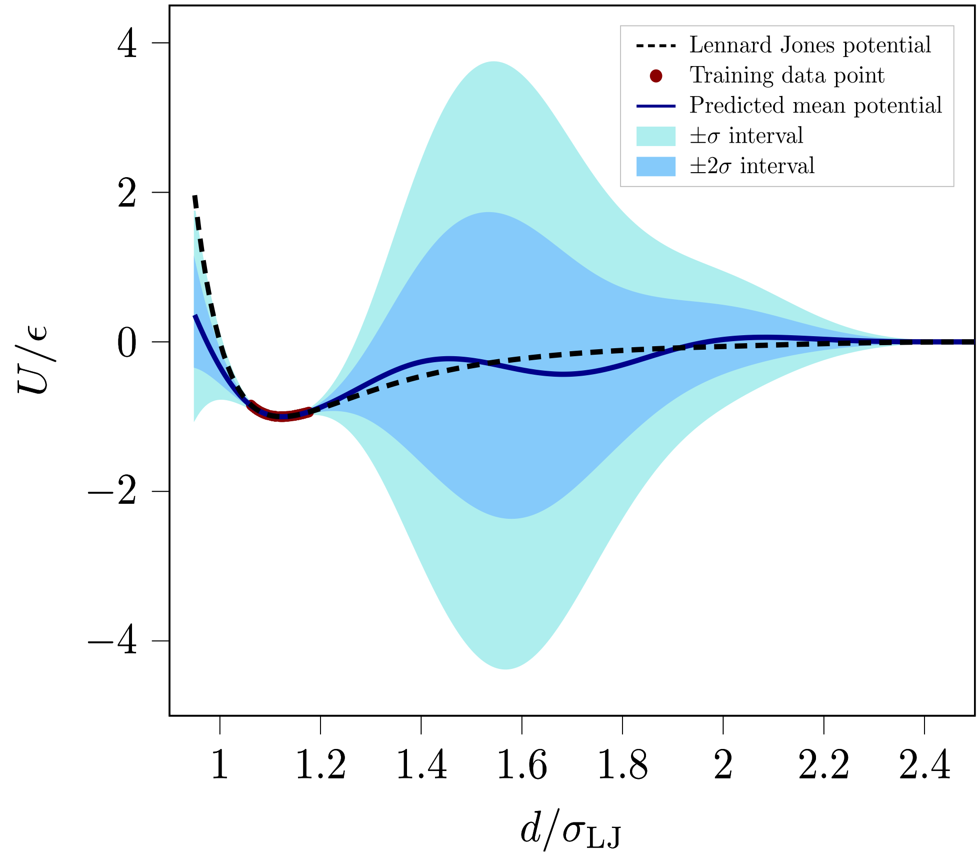
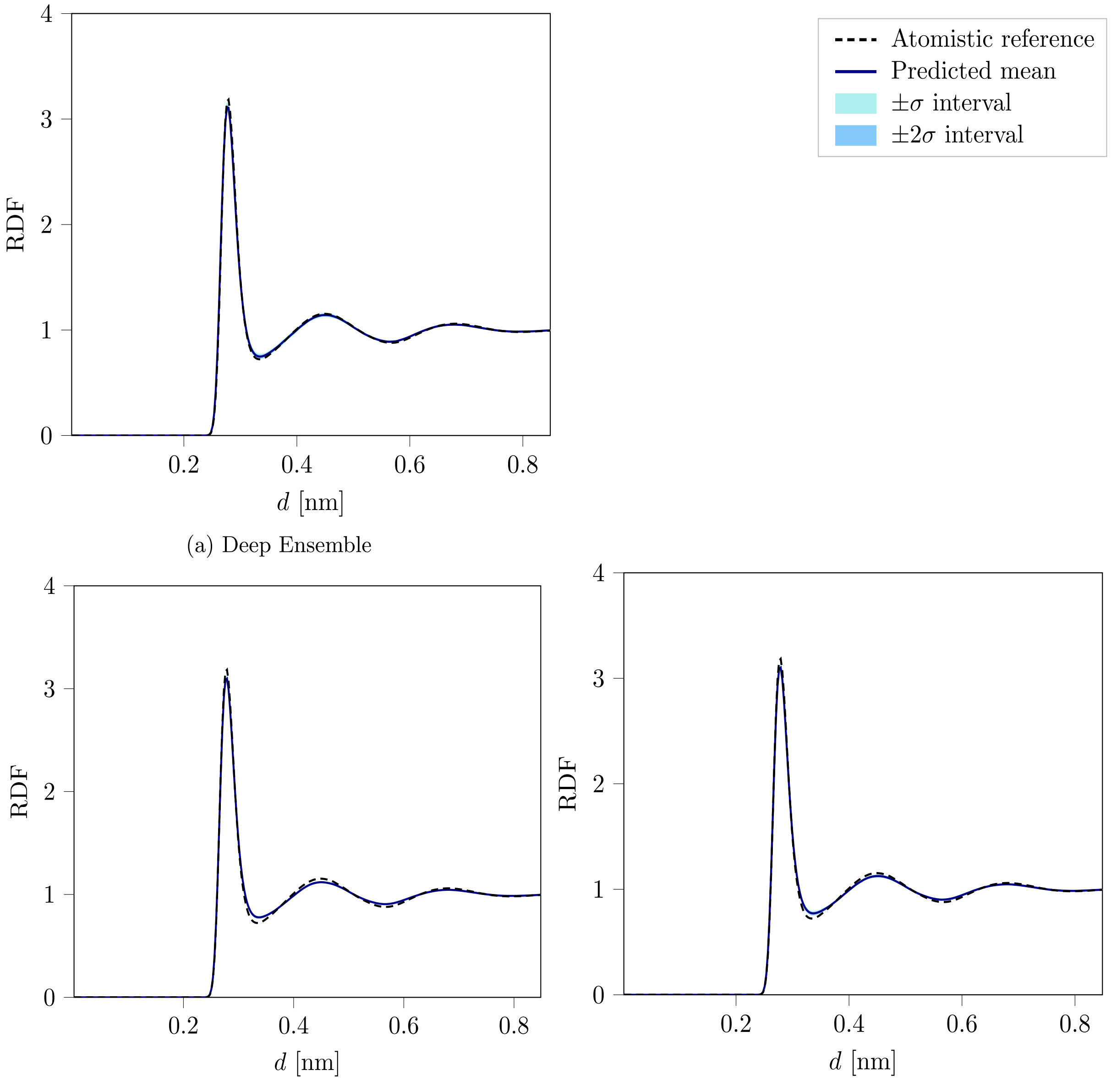


Figure 3: Single chain No-U-Turn Sampler (NUTS) with fixed $\sigma_{\mathrm{H}} = 0.05$. Predicted mean potential with $\pm\sigma$ and $\pm 2\sigma$ intervals compared to the Lennard Jones reference.

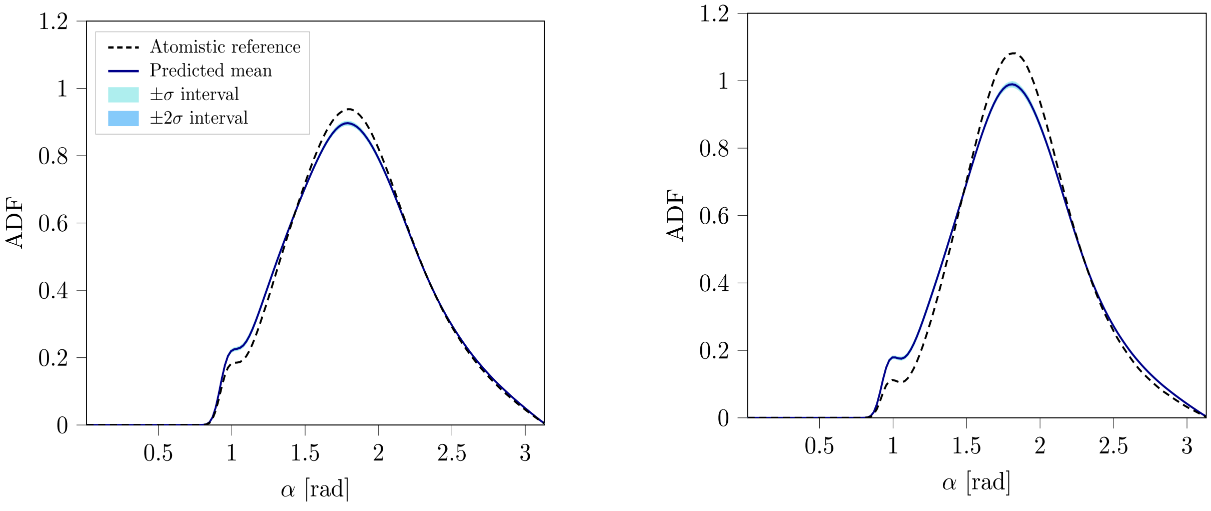


(a) Deep Ensemble

(b) S-pSGLD

(c) M-pSGLD

Figure 4: Radial distribution functions (RDF) at $T = T_{\text{ref}}$. Resulting mean RDF with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the Deep Ensemble (a), the single chain pSGLD (b) and the multi-chain pSGLD (c) schemes at a temperature $T = T_{\text{ref}}$, compared to the atomistic reference.



(a) $T = T_{\text{ref}}$

(b) $T = 260$K

Figure 5: Angular distribution functions (ADF) with uniform weight prior for S-pSGLD. Resulting mean ADF with ±σ and ±2σ intervals as predicted by the S-pSGLD method with uniform prior over weights and biases at a temperature $T = T_{\text{ref}}$ (*a*) and $T = 260$ K (*b*), compared to the atomistic reference.

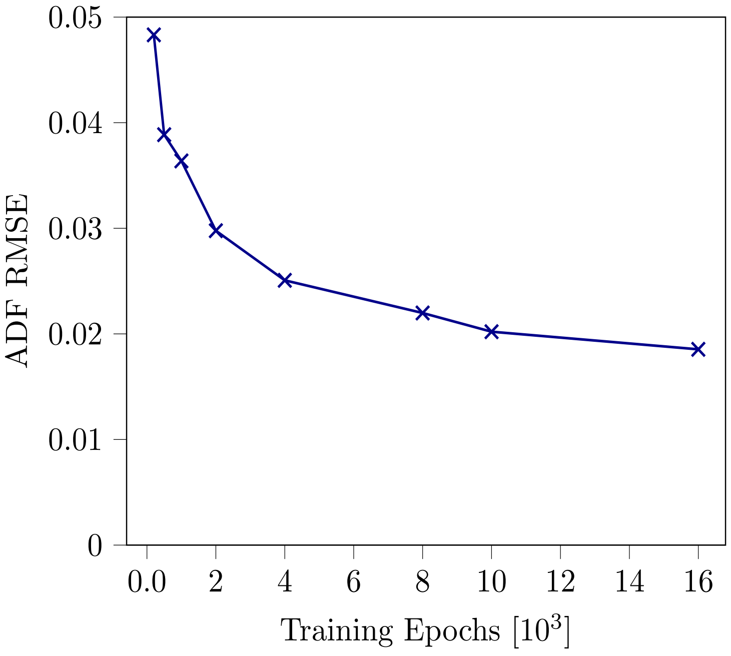
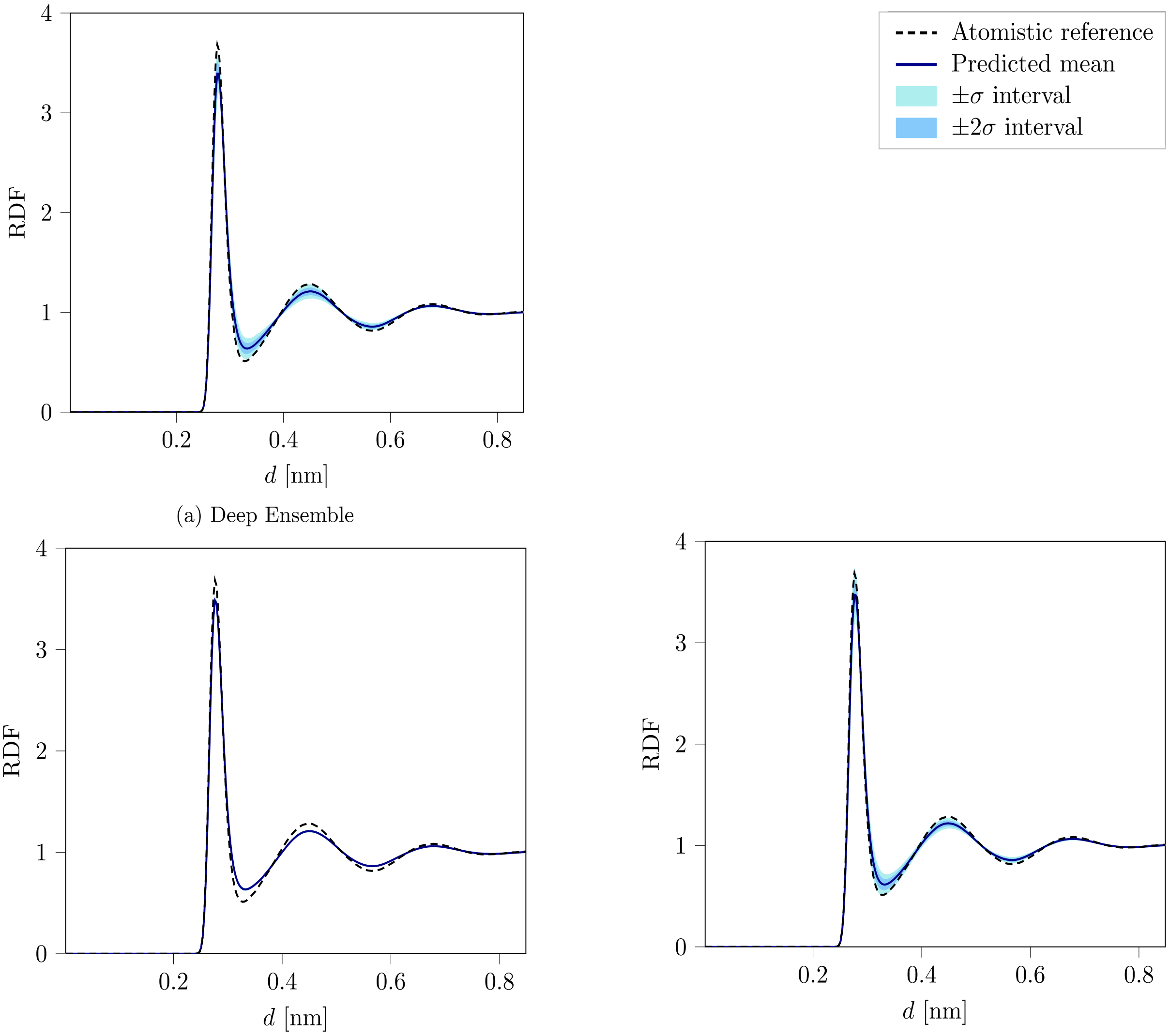


Figure 6: pSGLD chain length variation. Root mean squared error (RMSE) of the mean predicted angular distribution function (ADF) of models sampled by the pSGLD scheme with a single chain of different total lengths. The models are randomly retained after a burn-in period, which is 1000 epochs shorter than the total chain length. Exceptions are chains with lengths of 200, 500 and 1000 epochs, which feature a burn-in period of 200, 400 and 500 epochs, respectively.

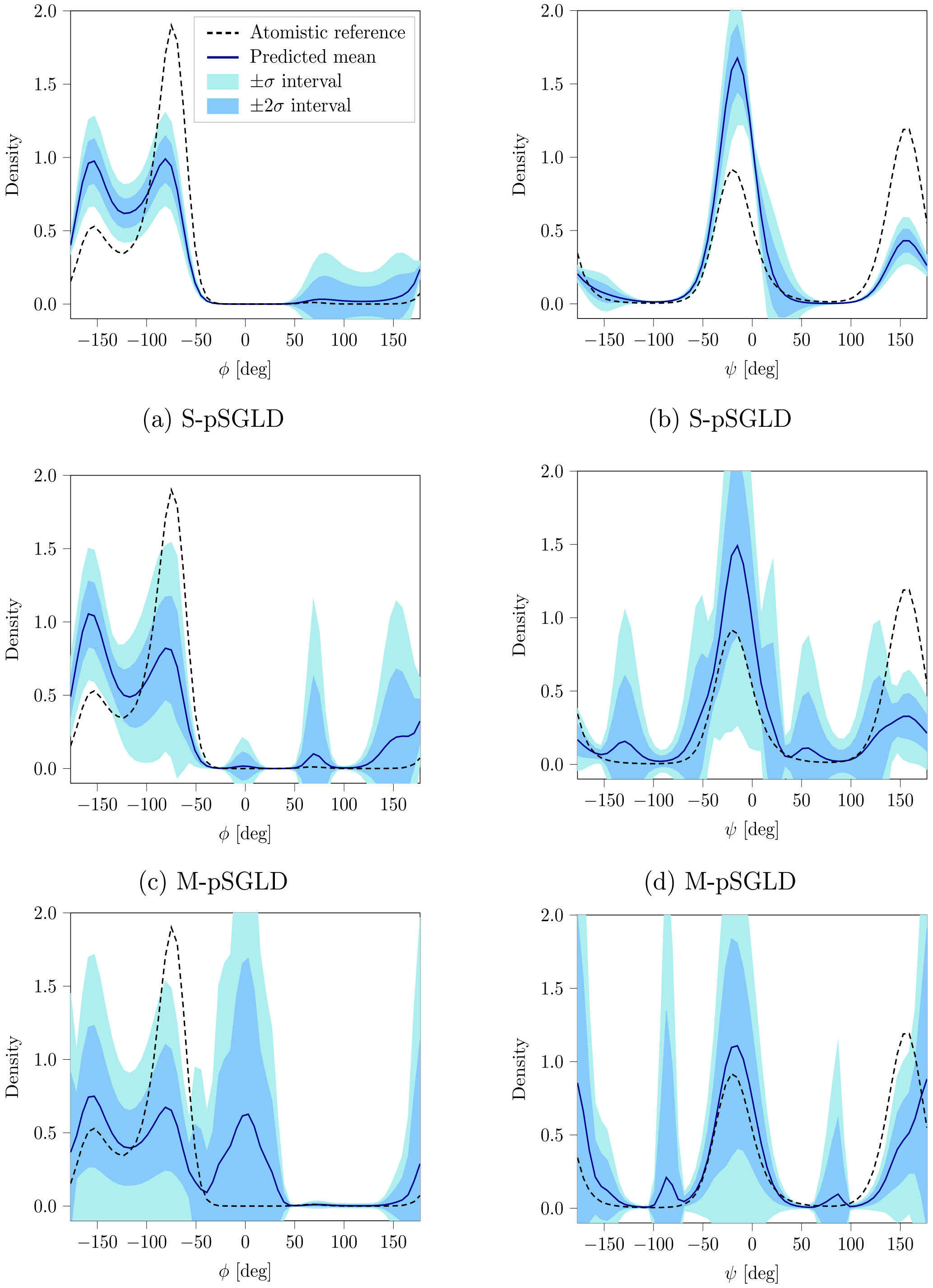


(a) Deep Ensemble

(b) S-pSGLD

(c) M-pSGLD

Figure 7: Radial distribution functions (RDF) at $T = 260$ K. Resulting mean RDF with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the Deep Ensemble (a), the single chain pSGLD (b) and the multi-chain pSGLD (c) schemes at a temperature $T = 260$ K, compared to the atomistic reference.



(a) S-pSGLD

(b) S-pSGLD

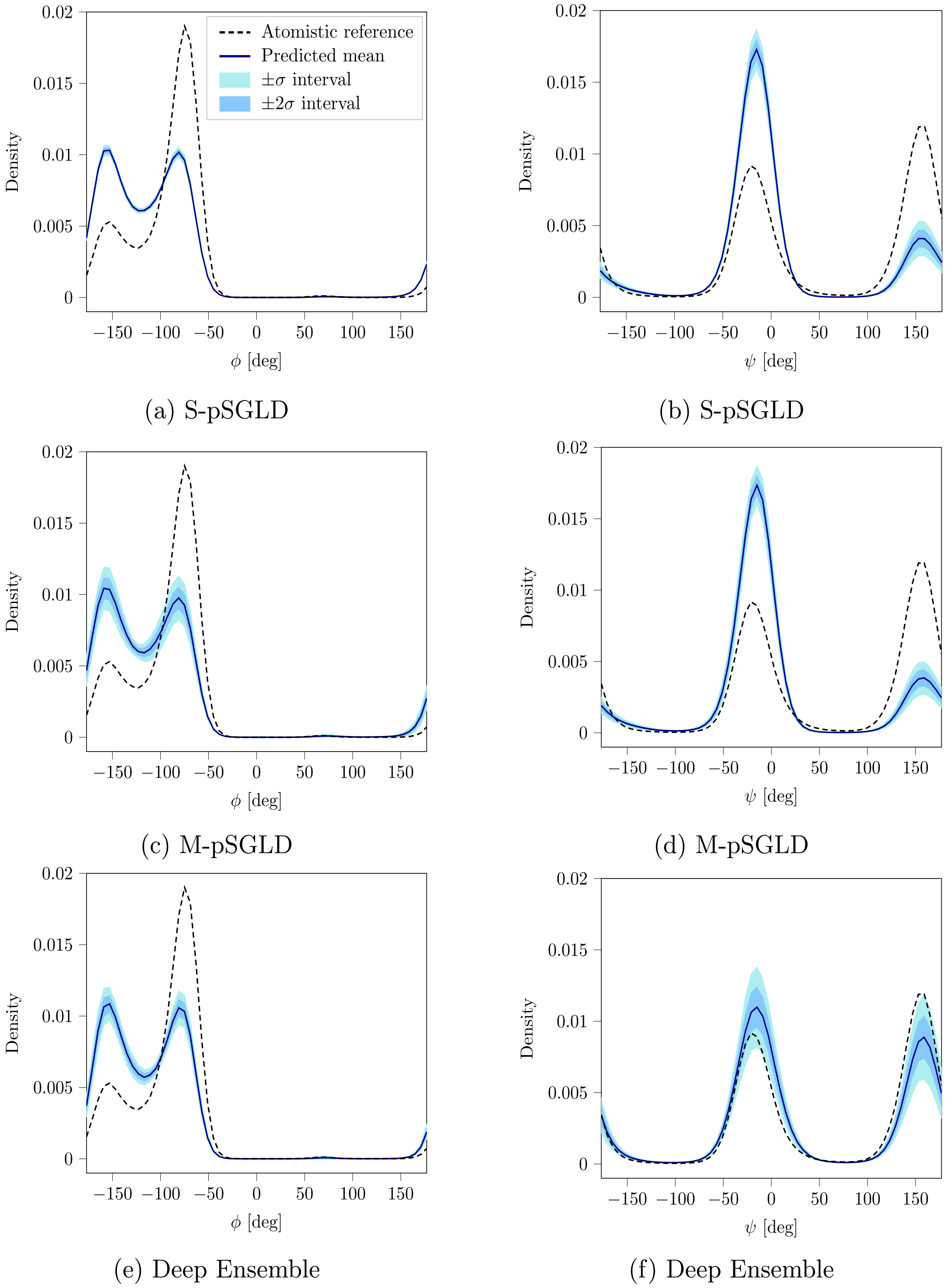
(c) M-pSGLD

(d) M-pSGLD

(e) Deep Ensemble

(f) Deep Ensemble

Figure 8: Dihedral angle density histograms including potential energy holes. Resulting mean distribution of dihedral angles ϕ (left column) and ψ (right column) with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the single chain pSGLD (a, b), the multi-chain pSGLD (c, d) and the Deep Ensemble (e, f) methods based on the 100 ns reference data set, compared to the atomistic reference. No trajectories were removed, except when a potential energy hole resulted in a divergent trajectory. The number of diverged trajectories are 0, 6 and 7 for single chain pSGLD, multi-chain pSGLD, and the Deep Ensemble method, respectively.



(a) S-pSGLD

(b) S-pSGLD

(c) M-pSGLD

(d) M-pSGLD

(e) Deep Ensemble

(f) Deep Ensemble

Figure 9: Resulting dihedral angle density histograms from $1\mu s$ training data set. Resulting mean distribution of dihedral angles ϕ (left column) and ψ (right column) with $\pm\sigma$ and $\pm 2\sigma$ intervals as predicted by the single chain pSGLD (a, b), the multi-chain pSGLD (c, d) and the Deep Ensemble (e, f) methods, compared to the atomistic reference. Analogous to the results in the main text, we removed 2, 10 and 28 trajectories due to potential energy holes from S-pSGLD, M-pSGLD and the Deep Ensemble method, respectively.

Supplementary References

[1] Wang, J. *et al.* Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **5**, 755–767 (2019).

[2] Husic, B. E. *et al.* Coarse graining molecular dynamics with graph neural networks. *J. Chem. Phys.* **153**, 194101 (2020).

[3] Fu, X. *et al.* Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. In *AI for Science: Progress and Promises Workshop at NeurIPS* (New Orleans, LA, USA, Dec. 2, 2022).

[4] Marrink, S. J., Risselada, H. J., Yefimov, S., Tieleman, D. P. & De Vries, A. H. The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B* **111**, 7812–7824 (2007).

[5] Ingólfsson, H. I. *et al.* The power of coarse graining in biomolecular simulations. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **4**, 225–248 (2014).

[6] Ramakrishnan, R., Dral, P. O., Rupp, M. & Von Lilienfeld, O. A. Big data meets quantum chemistry approximations: The Δ-machine learning approach. *J. Chem. Theory Comput.* **11**, 2087–2096 (2015).

[7] Thaler, S., Stupp, M. & Zavadlav, J. Deep coarse-grained potentials via relative entropy minimization. *J. Chem. Phys.* **157**, 244103 (2022).

[8] Klicpera, J., Groß, J. & Günnemann, S. Directional Message Passing for Molecular Graphs. In *8th International Conference on Learning Representations* (Online, Apr. 26 – May 1, 2020).

[9] Klicpera, J., Giri, S., Margraf, J. T. & Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In *Machine Learning for Molecules Workshop at NeurIPS* (Online, Dec. 12, 2020).

[10] Thaler, S. & Zavadlav, J. Learning neural network potentials from experimental data via differentiable trajectory reweighting. *Nat. Commun.* **12**, 6884 (2021).

[11] Hoffman, M. D., Gelman, A. *et al.* The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.* **15**, 1593–1623 (2014).

[12] Hansen, L. & Salamon, P. Neural network ensembles. *IEEE Trans. Pattern Anal. Machine Intell.* **12**, 993–1001 (1990).

[13] Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., Long Beach, CA, USA, Dec. 4–9, 2017).

[14] Welling, M. & Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, 681–688 (Omnipress, 2600 Anderson St, Madison, WI, USA, Bellevue, WA, USA, Jun. 28 – Jul. 2, 2011).

[15] Thompson, A. P., Plimpton, S. J. & Mattson, W. General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions. *J. Chem. Phys.* **131**, 154107 (2009).

[16] Das, A. & Andersen, H. C. The multiscale coarse-graining method. iii. a test of pairwise additivity of the coarse-grained potential and of new basis functions for the variational calculation. *J. Chem. Phys.* **131**, 034102 (2009).

[17] Dunn, N. J. & Noid, W. Bottom-up coarse-grained models that accurately describe the structure, pressure, and compressibility of molecular liquids. *J. Chem. Phys.* **143**, 243148 (2015).