ENHANCING THE PREDICTIVE POWER OF MOLECULAR DYNAMICS SIMULATIONS TO FURTHER THE MATERIALS GENOME INITIATIVE

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Saaketh Desai

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2020

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Alejandro Strachan, Chair
School of Materials Engineering
Dr. Anter El-Azab
School of Materials Engineering
Dr. Marisol Koslowski
School of Mechanical Engineering
Dr. Ilias Bilionis
School of Mechanical Engineering

Approved by:

Dr. David Bahr
Head of the School Graduate Program

To my mother and father, for maintaining balance in my life

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Alejandro Strachan, for guiding me along every step of my PhD. I am extremely fortunate to have you as one of my first professional mentors, and you have made me a better researcher and a person.

I would also like to thank my committee members Prof. Marisol Koslowski, Prof. Anter El-Azab, and Prof. Ilias Bilionis for their time and advice. A huge thanks to Prof. Rodney Trice for being an amazing mentor and supporting me in my various endeavors, and to Prof. Byron Pipes for helping me in the early years of my PhD. Thank you to Dr. James Belak and Sam Reeve for a wonderful internship experience at LLNL. I would like to thank funding from US-DOE BES, IACMI, and NCN to support this work.

Thank you to my early mentors Mitch Wood, Kiettipong Banlusan, Sam Reeve, and Aniruddha Jana, for holding my hand through a difficult transition from student to researcher, and instilling a core set of principles that have proven invaluable to me over time. I am thankful to Shivam Tripathi and Karthik Guda Vishnu for keeping me rooted and providing perspective to my work. Thank you to Strachan group members Tongtong Shen, Michael Sakano, Zachary McClure, and Juan Carlos Verduzco for their constant support and help. I hope I can continue to work with you all throughout my professional career.

A big thank you to my friends Aaditya Candadai, Prabhu Kumar, Radhakrishna T.G, Abhas Deva, Rishav Roy, Kshitiz Gupta, Rajath Kantharaj, Sri Tapaswi Nori, and Revati Chowgule. You all were there through the good times and bad, and I could not have asked for a better peer group and support system. Thank you to the folks at Purdue Cricket for many fun games and friendships beyond borders.

Lastly, thank you to my parents, who were a pillar of strength to me and whose unending love and support helped me maintain balance.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

ABSTRACT

Desai, Saaketh Ph.D., Purdue University, December 2020. Enhancing the predictive power of molecular dynamics simulations to further the Materials Genome Initiative. Major Professor: Alejandro Strachan.

Accelerating the development of novel materials is one of the central goals of the Materials Genome Initiative and improving the predictive power of computational material science methods is critical to attain this goal. Molecular dynamics (MD) is one such computational technique that has been used to study a wide range of materials since its invention in the 1950s. In this work we explore some examples of using and increasing the predictive power of MD simulations to understand materials phenomena and provide guidelines to design tailored materials.

We first demonstrate the use of MD simulations as a tool to explore the design space of shape memory alloys, using simple interatomic models to identify characteristics of an integrated coherent second phase that will modify the transformation characteristics of the base shape memory alloy to our desire. Our approach provides guidelines to identify potential coherent phases that will achieve tailored transformation temperatures and hysteresis.

We subsequently explore ideas to enhance the length and time scales accessible via MD simulations. We first discuss the use of kinetic Monte Carlo methods in MD simulations to predict the microstructure evolution of carbon fibers. We find our approach to accurately predict the transverse microstructures of carbon fibers, additionally predicting the transverse modulus of these fibers, a quantity difficult to measure via experiments. Another avenue to increase length and time scales accessible via MD simulations is to explore novel implementations of algorithms involved in machine-learned interatomic models to extract performance portability. Our ap-

proach here results in significant speedups and an efficient utilization of increasingly common CPU-GPU hybrid architectures.

We finally explore the use of machine learning methods in molecular dynamics, specifically developing machine learning methods to discover interpretable laws directly from data. As examples, we demonstrate the discovery of integration schemes for MD simulations, and the discovery of melting laws for perovskites and single elements. Overall, this work attempts to illustrate how improving the predictive capabilities of molecular dynamics simulations and incorporating machine learning ideas can help us design novel materials, in line with the goals of the Materials Genome Initiative.

1. INTRODUCTION

From the Stone Age and the Bronze Age, to the Industrial Revolution and the Information Age, most of human history has been defined with the discovery of novel materials. These materials defined the tools and technology of the time, and the development of new materials has historically resulted in significant improvements in these tools, improving human life. The quest to develop new materials has led to the invention of increasingly complex materials designed for specific applications. One example is the invention of steel by combining iron with carbon to improve on the strength of iron. More recently, some of the greatest technological advances of the twentieth century were also realized via novel materials, whose development was aided by new experimental and computational techniques. One example is the development of transistors in the 1940s, leading to the advent of electronic devices and heralding the start of the Information Age. Yet another powerful example is the development of lithium ion batteries in the 1980s and 1990s, revolutionizing energy technology beyond fossil-fuels. Looking ahead, the 21st century will also be defined by materials innovations that solve some of the grand challenges of humanity such as capturing excess carbon dioxide, and developing sustainable, renewable energy sources.

Despite these successes in developing new materials, one of the current challenges in developing new materials is to accelerate the time required to transition from discovery/invention to deployment, a process that currently takes 15-20 years for most novel materials [1]. For instance, Li-ion batteries were first discovered in 1985, but saw widespread use only in the 2000s. Recognizing the need to develop new materials faster, the Materials Genome Initiative was launched in 2011 with the goal of accelerating the development of advanced materials for energy storage, next generation electronics and various other applications by a factor of two [1]. The MGI has identified that accelerating the development of novel materials requires synergy

between computational modeling approaches, experimental tools, and increasingly, data-driven approaches to material design. The role of computational approaches in this synergy is to develop robust models that guide or replace expensive and time consuming experiments, while also providing insight to processing-structure-property relationships that are difficult or time consuming to obtain via experiments.

Computational material science methods span various length and time scales, ranging from quantum mechanical methods such Density Functional Theory (DFT) and molecular dynamics (MD) at the atomic scale, to mesoscale methods such as phase field simulations, all the way to macroscale, component level methods such as Finite Element Methods. Each method offers unique insights into materials phenomena, for instance DFT calculations provide a detailed picture of the electronic structure of material, critical to designing advanced semiconductors. However, they are limited in the length and time scales accessible and cannot be used to predict properties that require system sizes bigger than a few hundred atoms. Molecular dynamics simulations also offer atomistic insight into materials phenomena but can predict high temperature properties unlike DFT calculations and can be used to inform mesoscale and component level models, which then provide macroscale material insights.

The computational design and understanding of materials thus requires integration of these various modeling techniques. This need was identified by the Integrated Computational Materials Engineering (ICME) initiative. ICME was launched to streamline materials development by linking modeling protocols across various length and time scales to provide comprehensive predictions and guidance in developing new materials. The ICME approach attempts to seamlessly link manufacturing processes to materials models, integrating process-property and structure relationships in materials. This approach has led to many success stories such as the design of high-performance alloys for turbines [2] and the virtual aluminum casting process at Ford [3].

An emerging arm of materials development approaches is the use of machine learning and data-driven methods in discovering and designing novel materials. The role of data-driven methods is to develop surrogate models that offer rapid predictions in situations that are difficult or expensive to understand via experiments and computational methods. Recent success stories demonstrate the use of data-driven approaches in conjunction with computational methods to develop new materials, significantly reducing the time required to discover new perovskite materials [4], low hysteresis shape memory alloys [5], and solid Li-ion conductors [6]. However, these approaches are still limited to combining quantum mechanical DFT calculations with machine learning methods, leveraging an increasing number of curated DFT calcualation databases such as Materials Project [7], OQMD [8], and AFLOW [9]. Utilizing insights from higher scale computational materials models can further aid these data-driven approaches, and is an active area of research. The existence of this 'Materials Data Infrastructure' will allow streamlined access to this multi scale materials data, in addition to standardized protocols to work with such data [10].

The central theme of this thesis is molecular dynamics simulations and this collection of work will explore ways to use molecular dynamics simulations as well as enhance them to design novel materials in an effort to achieve the goals outlined by the MGI and ICME efforts. This thesis is organized as follows: We will first introduce molecular dynamics simulations in Chapter 2, identifying limitations and sources of improvement in the predictive capabilities of this technique. We will also briefly explore machine learning methods in material science and molecular dynamics simulations, again identifying limitations in the current state-of-the-art. Subsequent chapters will discuss case studies of fundamentally improving on these limitations for specific applications. Chapter 3 demonstrates the use of molecular dynamics simulations with simple interatomic models to understand trends in martensitic material behavior, guiding the design of novel, room-temperature operable, light-weight shape memory alloys. Chapter 4 discusses the coupling of molecular dynamics simulations with other computational material science techniques, such as kinetic Monte

Carlo methods, to describe microstructure evolution in carbon fibers, providing a first step towards designing next generation high-strength, high-stiffness fibers. Chapter 5 explores novel implementations of molecular dynamics algorithms to access greater length and time scales. Chapter 6 focuses on some novel machine learning approaches to discover integration schemes for molecular dynamics simulations, exemplifying the use of interpretable machine learning models as tools to discover underlying physics directly from data.

2. METHODS

In this chapter we will explore classical molecular dynamics (MD) simulations as a method to go beyond highly accurate quantum mechanical calculations and connect these calculations to higher scale models at the microstructural and component level. We will also explore machine learning methods in material science and introduce some methods that are used in this thesis, either to improve the accuracy of MD simulations, or to discover laws relevant to MD simulations. We will also look at limitations in the current state of the art and identify areas for improvement that will be addressed in subsequent chapters.

2.1 Molecular Dynamics

Molecular dynamics simulations allow the study of complex systems at an atomic scale by following the trajectory of a set of atoms whose interactions are defined by Newton's laws of motion. Classical molecular dynamics, here after referred to as molecular dynamics, was first developed by Alder and Wainwright in the 1950s [11,12] for a system of hard sphere particles, with realistic materials simulations soon following in the 1960s and 1970s [13,14].

Since its inception, molecular dynamics simulations have been critical to provide atomistic insight into materials phenomena that involve length and time scales larger than Density Functional Theory calculations. MD simulations are crucial to connect information from ab-initio calculations to higher scale models. For instance, MD simulations, with parameters fitted to DFT calculations, have provided discrete dislocation dynamics models with dislocation core energies [15], which are critical to predict dislocation interactions and crystal plasticity at macroscales. In recent times, large scale MD simulations have also identified deformation mechanisms in single crystals

beyond dislocation-based plasticity, further informing higher scale models of materials phenomena under low temperatures and high strain rates [16]. MD simulations can also inform dislocation behaviors concurrently, as is the case in coupled atomistic discrete dislocation (CADD) simulations [17]. A similar approach has proven successful to study radiation damage in materials [18].

Another example of molecular dynamics simulations being used to propagate quantum mechanical information to higher scale models is the use of multi-scale models to simulate crack propagation [19]. In this case, the molecular dynamics simulations simulate atomic stresses in environments where continuum elastic theory fails, and is informed by bond-breaking events from the quantum mechanical calculations.

The goal of this section is to provide an understanding of MD simulations, to understand how these simulations can be more accurate and access higher length and time scales, and couple more effectively with higher scale models. The typical workflow for a molecular dynamics simulation is shown in Figure 2.1, and each subsequent subsection will expand on each aspect of this workflow.

2.1.1 Initial structure

To begin the simulation, we start with a set of atoms whose positions and velocities are defined. For simple crystal structures, the atom positions can be assigned manually, or by using standard packages such as Atomsk [20] or LAMMPS [21]. For instance, we use the LAMMPS package [21] to define a body centered cubic structure as the initial structure that will subsequently undergo a martensitic transformation to a monoclinic phase, see Chapter 3 for additional details. Software packages such as Atomsk are also used to generate atomistic structures of polycrystalline materials. For complex amorphous polymer structures, one can use Packmol [22] or the Polymer Modeler tool [23]. These tools allow users to specify a single monomer or a set of monomers, which can then be used to generate amorphous polymer structures.

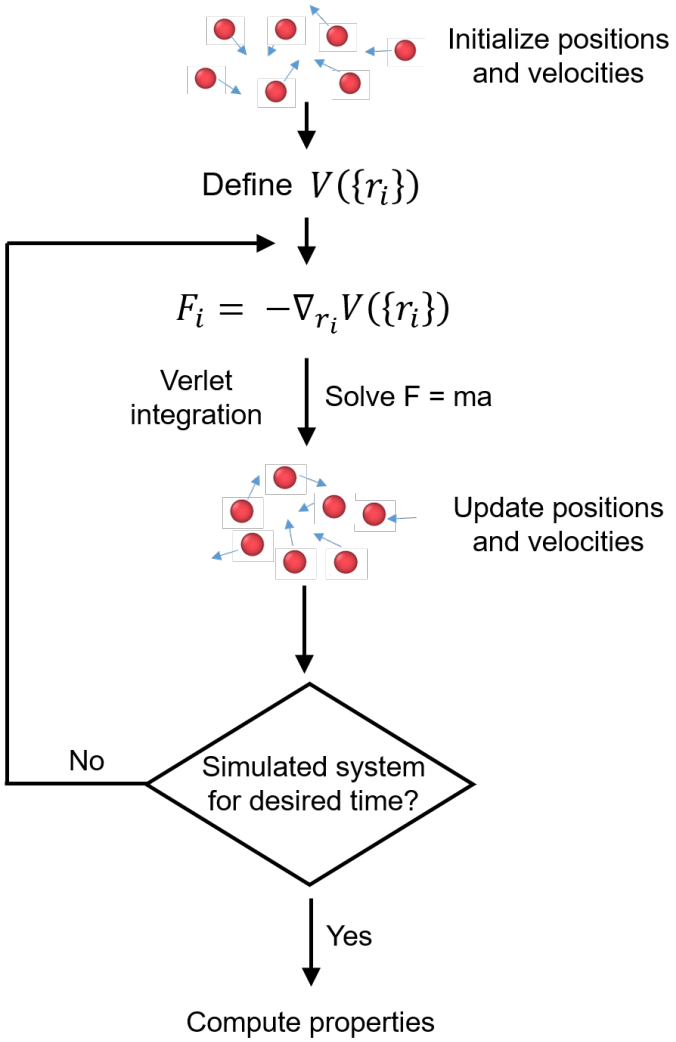

Fig. 2.1. Typical molecular dynamics workflow

To generate structures with complex interfaces, one can use the Virtual NanoLab toolkit from Synopsys-Quantum ATK [24]. Software packages like these showcase how publicly available software can be beneficial to a large audience

However, certain specialized initial structures still require custom codes and scripts. One example of this is the customized ladder-like structures built to represent stabilized carbon fiber precursor structures, used as initial structures for the MD-CF

carbonization algorithm presented in Chapter 4. Even for such specialized cases, the availability of the codes used to generate these structures aids reproducibility.

2.1.2 Interactions between atoms

Having defined a set of positions and velocities, we now define the interactions between atoms. The forces between atoms could be obtained by solving the time-independent Schrodinger's equation for the energy of the system, the negative gradient of which (with respect to the atomic coordinate) will give us the force on each atom. This approach is known as $ab-initio$ molecular dynamics (AIMD) and results in highly accurate forces but at great computational cost. AIMD simulations are thus usually limited to a few thousand atoms for hundreds of picoseconds. AIMD simulations are most often used when accuracy is of prime concern, and/or a computationally cheaper interaction model is unavailable. In classical MD, we compute the forces on each atom using an approximate 'interatomic potential', also known as an 'interatomic model', or a 'force field'. This approximation removes the need to solve the Schrodinger's equation and reduces the cost of computing the forces, allowing simulations of larger system sizes (up to billions of atoms) and longer timescales (up to nanoseconds). We will now briefly introduce various forms of interatomic models.

The central idea of an interatomic model is to encode the environment around each atom into a set of 'descriptors', subsequently mapping the descriptors to the energy of the system and the forces acting on each atom. The earliest interatomic models were designed to simulate the behavior of simple materials such as liquid argon [13], where a two-body interatomic model could accurately describe the properties of interest such as the self-diffusion coefficient. In a two-body model, the environment around each atom is described by the set of distances $\{r_{ij}\}$ between a central atom i and each of its neighbors j. The mapping from the set of distances (descriptors) to the potential energy is given by various functional forms, such as a power law form for the Lennard-Jones model

$$V(\{r_i\}) = \sum_{i=1}^{N} \sum_{j=1}^{i} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \tag{2.1}$$

or an exponential function for the Morse model.

$$V(\{r_i\}) = \sum_{i=1}^{N} \sum_{j=1}^{i} D_0 \left[e^{-2\alpha(r_{ij}-r_0)} - e^{\alpha(r_{ij}-r_0)} \right] \tag{2.2}$$

where ϵ, σ, D_0, α, r_0 are model parameters.

Two-body interatomic models are computationally cheap as they only consider pairwise considerations, and can provide useful trends in materials properties if the model is parametrized accordingly. As discussed in detail in Chapter 2, we use a Morse interatomic model to obtain trends in martensitic phase transformations and guide the design of lightweight shape memory alloys. In addition, these models are the computationally cheapest. However, for all these advantages, a pairwise interaction model has a few pitfalls. For instance, two-body models always predict the vacancy formation energy and cohesive energy to be equal, which is not true for most materials.

To address these limitations, many body, higher order interatomic models were developed such as the Embedded Atom Model (EAM) [25], Stillinger-Weber [26], and Dreiding [27]. In the EAM formalism, the atomic environment is encoded in a pairwise term, containing the set of distances $\{r_{ij}\}$, as well as an embedding term, with the functional mapping for these terms varying for different models.

$$V(\{r_i\}) = \sum_{i=1}^{N} F_i(\rho_i) + \sum_{i=1}^{N} \sum_{j=1}^{i} \phi(r_{ij}) \tag{2.3}$$

where F is the embedding function, ρ_i is the electron density, approximated as a function $f(r_{ij})$ that depends on pairwise distances.

In the Stillinger-Weber and Dreiding models, the descriptors for the environment around each atom are defined to be the set of pairwise distances $\{r_{ij}\}$ as well as three-body and four-body descriptors such as angles and torsions [26, 27]. For bonded and organic systems, the ReaxFF interatomic models offer accurate manybody descriptions of atomic interactions [28]. The ReaxFF model maps each atomic environment

in terms of bond-orders, which are then mapped to the energy using exponential functional forms, each functional form contributing one term of the total energy.

The interatomic models described so far have proven useful to model melting and surface properties in metals [29], thermal transport in semiconductors [30], glass transition phenomena in polymers [31], high energy materials [32] and a wide variety of other properties and materials. However, the accuracy of these models is still limited by the functional form chosen for the mapping, making these models fundamentally limited in the range of atomic environments they can accurately represent. In the past decade, machine learning methods have emerged as novel avenues to describe atomic interactions accurately across a wide range of atomic environments by leveraging the flexible nature of machine learned models and their thousands of tunable parameters. This aspect will be discussed in detail in Section 2.3.2.

2.1.3 Integration schemes

The previous subsection described various ways of defining an empirical interatomic model. We can now use any of those definitions to obtain the forces acting on each atom, subsequently solving Newton's second law of motion, $F = ma$, to obtain the new position and velocity of each particle. To do this, we discretize Newton's second law of motion, obtaining a numerical integration scheme.

We re-write $F = ma$ as

$$F = m\frac{dv}{dt} \tag{2.4}$$

$$v = \frac{dx}{dt} \tag{2.5}$$

Various discretization methods differ in the way the derivatives are approximated numerically, giving us different integration schemes with varying accuracy.

The simplest integration scheme is the Euler integration scheme, which numerically approximates each derivative using the forward difference scheme. Thus,

$$\begin{aligned} v(t) &= \frac{dx}{dt} = \frac{x(t+\Delta t) - x(t)}{\Delta t} \\ F(x(t)) &= m\frac{dv}{dt} = m\frac{v(t+\Delta t) - v(t)}{\Delta t} \end{aligned} \tag{2.6}$$

which can be re-written as:

$$\begin{aligned} x(t+\Delta t) &= x(t) + v(t)\Delta t \\ v(t+\Delta t) &= v(t) + \frac{F(x(t))}{m}\Delta t \end{aligned} \tag{2.7}$$

The Euler integration scheme can also be viewed as a Taylor series expansion of the position and velocity about Δt, truncated to the first-order.

To improve on this scheme, we can consider a central difference approximation for the derivatives:

$$v(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \tag{2.8}$$

including derivatives at a half-step, such as:

$$v(t + \frac{\Delta t}{2}) = \frac{x(t+\Delta t) - x(t)}{\Delta t} \tag{2.9}$$

Using the central difference scheme, we can derive the Verlet formulation [33]:

$$\begin{aligned} x(t+\Delta t) &= 2x(t) - x(t+\Delta t) + \frac{F(x(t))}{m}\Delta t^2 \\ v(t) &= \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \end{aligned} \tag{2.10}$$

Note that this integration scheme was discovered earlier under various other names [34]. While the Verlet formulation is not self-starting (we require both current and previous position to evaluate the next position), we can reformulate this as:

$$\begin{aligned} x(t+\Delta t) &= x(t) + v(t) + \frac{1}{2}\frac{F(x(t))}{m}\Delta t^2 \\ v(t+\Delta t) &= \frac{x(t+\Delta t) - x(t)}{\Delta t} + \frac{F(x(t+\Delta t)}{m}\frac{\Delta t}{2} \end{aligned} \tag{2.11}$$

making the integration scheme self-starting and requiring one force evaluation per integration step, $F(x(t+\Delta t))$.

We can again see that the Verlet scheme amounts to a Taylor series expansion of the position, truncated to the second-order. In practice, most MD codes will break this down into three steps as follows:

$$\begin{aligned} v(t+\frac{\Delta t}{2}) &= v(t) + \frac{F(x(t))}{m}\frac{\Delta t}{2} \\ x(t+\Delta t) &= x(t) + v(t+\frac{\Delta t}{2})\Delta t \\ v(t+\Delta t) &= v(t+\frac{\Delta t}{2}) + \frac{F(x(t+\Delta t))}{m}\frac{\Delta t}{2} \end{aligned} \tag{2.12}$$

This scheme is known as the Velocity Verlet scheme, and is the integration scheme used in the LAMMPS software package [21]. An alternative formulation, the Position Verlet scheme [35] can be obtained by using the Liouville operator, as discussed in Chapter 6. Additionally, further higher order schemes such as Runge-Kutta integration methods can be obtained by including higher-order terms in the Taylor series expansion, but require an increasing number of force evaluations at every step. In Chapter 6, we will explore a scheme to discover integration schemes from data without prior knowledge of the underlying physics.

2.2 Need for longer time scales and length scales

Newton's second law of motion is a second-order ODE that can be re-written as two first-order differential equations, as seen in the previous section. This system of differential equations is chaotic, meaning that small errors in numerically solving for the positions and velocities can quickly build up and generate inaccurate trajectories. To avoid these numerical errors, we require small timesteps in the integration schemes discussed before, which limits current molecular dynamics simulations to billions of atoms as of today [36], and microsecond timescales [37].

However, many key problems in material science are governed by material phenomena at higher length and time scales, while also requiring atomistic insight to accurately predict material response. Some examples of this are the time evolution of radiation damage in materials, low strain rate deformation response, and microstruc-

ture evolution processes. Increasing the length and time scales accessible via MD simulations are required to address these challenges. Accelerated MD methods such as Hyperdynamics [38] and Parallel replica dynamics [39] leverage Transition State Theory offer one way to access to microsecond - millisecond timescales, especially to simulate rare events such as diffusion processes [40] and radiation damage events [41], processes that are beyond the reach of conventional MD simulations.

In this thesis, we attempt to increase the time scales accessible via molecular dynamics simulations in two ways: (i) Chapter 4 discusses incorporation of kinetic Monte Carlo methods in MD to simulate the microstructure evolution in carbon fibers (ii) Chapter 5 explores novel ways to implement MD algorithms to extract performance across various platforms.

2.3 Machine learning methods for material science

Ever-increasing computing resources and the development of accelerated computational methods, as discussed in the previous section, continue to increase avenues for the computational design of materials from first principles. At the same time, the advent of machine learning methods has led to a novel avenue for accelerated materials design: the combination of computational methods with machine learning techniques, as mentioned in Chapter 1. This thesis briefly explores the use of machine learning methods in the context of molecular dynamics simulations. Specifically, Chapter 5 discusses ways to implement machine-learned interatomic models to extract cross-platform performance and leverage hybrid CPU-GPU architectures. Chapter 6 discusses the use of novel machine learning methods to discover molecular dynamics integration schemes directly from data. The goal of this section is to provide a brief background of machine learning methods in molecular dynamics and material science, providing context to the work in Chapter 5 and Chapter 6. This section will also highlight cyber-infrastructure developed to promote the use of machine learning and probabilistic methods in material science.

We first introduce machine learning methods in materials design efforts, and potential improvements in this area.

2.3.1 Machine learning for materials discovery and design

As mentioned in Chapter 1, the past decade has seen an increasing number of attempts at discovering new materials via a combination of machine learning and computational methods, specifically Density Functional Theory calculations. This has led to the discovery of novel perovskites [4], low hysteresis shape memory alloys [5], and solid Li-ion conductors [6]. The central thrust in these approaches is for the machine learning model to rapidly predict a material property of interest, bypassing time consuming simulations or experiments. This strategy leverages the flexibility in various ML models and their ability to learn mappings in various tasks ranging from image recognition [42] to sentence translation [43].

The inputs to these machine learning models in the case of materials design problems can be elemental information such as atomic radii, electronegativity, and other periodic table information readily accessible via databases such as Materials Project [7] and OQMD [8]. While these inputs offer little insight into the problem at hand, deep learning models can still learn mappings between these inputs and the material property of interest. For instance, a deep neural network has proven successful in predicting the stability of compounds directly from compositional data [44]. Another strategy for materials design is to inputs that capture the essential physics of the problem. This feature engineering allows the use of smaller ML models with lesser tunable parameters. A combination of feature engineering and transfer learning has shown great promise in predicting Li-ion conductors from small datasets [45]. While most materials design continue to rely on DFT calculations to design features for ML models, future work could focus on the use of higher scale models as input features to models, providing additional insight to the problem and providing increased confidence in model predictions.

One drawback of the use of ML models in materials design efforts is their lack of interpretability. Despite their high accuracy, ML models cannot 'explain' their predictions. The development of interepretable machine learning models is an active are of research [46, 47], as is the development of machine learning methods that satisfy pre-determined constraints [48, 49]. In Chapter 6, we will discuss the use of Parsimonious Neural Networks to discover interpretable models directly from data.

2.3.2 Machine learning in molecular dynamics simulations

The biggest use of machine learning methods in molecular dynamics simulations has been to increase the accuracy of interatomic models. Machine learned interatomic models differ from the traditional interatomic models discussed in Section 2.1.2 by using a wider set of descriptors for the atomic environment and by using flexible functional mappings such as neural networks or gaussian processes.

In the case of neural network potentials [50], the descriptors chosen are symmetry functions, which are a collection of gaussian functions describing the atomic environment. The symmetry functions are divided into radial and angular functions, which are defined below:

$$G^k(r_{ij}) = e^{-\eta_k(r_{ij}-r_{s_k})^2} f_c(r_{ij}) \tag{2.13}$$

$$G^k(r_{ijk}) = (1 + \lambda cos(\theta_{ijk}))e^{-\eta_k(r_{ij}^2+r_{ik}^2+r_{jk}^2)} f_c(r_{ij}) f_c(r_{ik}) f_c(r_{jk}) \tag{2.14}$$

These descriptors are then mapped to the energy of the system using a 'shallow' neural network, typically only two to three layers deep and consisting of ~30-50 neurons each.

Along similar lines is the Spectral Neighbor Analysis Potential [51]. Here the atomic environment is described in terms of Bispectrum coefficients, which are the coefficients of a four dimensional hyperspherical harmonic expansion of the neighbor density function. These bispectrum coefficients are then mapped to the energy and

forces using a linear regression model, with recent work exploring higher order mappings [52], highlighting the flexibility and the de-coupled nature of the descriptor and the mapping in machine learned interatomic models.

Recent efforts attempt to use machine learning methods to 'learn' the descriptors for an environment, as opposed to using physics-based intuition to design custom descriptors. These attempts show that an end-to-end machine learned interatomic model without human intuition can also prove to be an accurate description of the interactions between atoms in complex systems [53, 54].

We have recently developed neural network interatomic models for reactive systems containing carbon, oxygen, nitrogen and hydrogen, specifically for high-energy materials such as RDX. We develop an iterative scheme of training, where we train a 'first generation' interatomic model using an initial training set developed by scientific intuition. We then test the first generation model and collect any trajectories deemed incorrect to be passed back for training the subsequent generation. This iterative approach, in conjunction with small datasets periodically injected via scientific intuition, shows great promise in obtaining an accurate neural network interatomic model. We find that our interatomic model not only accurately the captures chemical kinetics of RDX, but also is an order-of-magnitude more accurate than the state-of-the-art ReaxFF models in predicting energies and forces for a wide variety of chemical configurations. This scheme is similar to other active learning based interatomic model fitting approaches [55].

Most machine learned interatomic models are at least an order of magnitude slower than traditional interatomic models [56, 57]. This largely stems from the extra computations involved in converting distances and angles between atoms in a neighborhood to multiple descriptors. While this may be rationalized as trading accuracy for computational cost, we will see in Chapter 5 that implementing the critical computational steps in a performance oriented manner can provide massive speedups and enable near quantum-accurate neural network interatomic model based molecular dynamics simulations for billions of atoms in the near future.

Other applications of ML methods in MD simulations are the use of recurrent neural networks to predict atomic trajectories, or the discovery of integrators directly from atomic data [58, 59]., as discussed in Chapter 6.

2.3.3 Cyber-infrastructure for machine learning methods in material science

The rapid advances in machine learning methods for material science outlined in the previous sections have relied on the availability of curated repositories of materials data. The rise of these databases has reduced the barrier to incorporate machine learning methods in material science research. However, one major barrier for more material scientists to adopt machine learning methods in their research is the lack of high-quality machine learning training resources focused on material science applications. To this end we develop Jupyter notebooks that introduce fundamental machine learning concepts [60] via nanoHUB, a cloud computing cyber-infrastructure that has provided easy access to materials simulation tools for over a decade (`http://nanohub.org/`). These notebooks introduce the ideas of data collection and curation via Numpy arrays and Pandas dataframes and fitting neural networks for regression and classification tasks. These notebook tools are accessible from a web browser without the need to install additional software, further increasing ease of access and widespread adoption.

The notebooks focus on problems relevant to material sciences, as introductory machine learning tutorials for a broad audience are widely available. For instance, instead of using neural networks to predict housing prices, the tool focuses on using neural networks to predict the Young's modulus of single elements given elemental information such as electronegativity, atomic radius, melting temperature etc.

Other tools on nanoHUB allow advanced users to familiarize themselves with advanced ML methods, such as training convolutional neural networks to predict the solubility of organic materials [61], using random forests to predict formation energies

of impurities in semiconductors [62], and and using state-of-the art active learning methods to guide further experiments [63].

The coupling of machine learning methods with computational and experimental methods poses additional challenges associated with information exchange across scales, physics, and models. Key among them is the quantification of uncertainties across these scales [15]. We introduce a Jupyter notebook for calibration [64] where users can easily upload data, determine state and calibration variables, match tool outputs with calibration data, and perform either standard or Bayesian calibration using the Dakota software package from Sandia National Laboratories [65]. One of the goals of this tool is to promote the use of uncertainty quantification approaches across material science. Here, we demonstrate this tool by performing a Bayesian calibration of an interatomic potential for molecular dynamics simulations.

The objective of this calibration is to determine posterior distributions for the parameters of an interatomic potential that match a given training dataset, including uncertainties. In this case, we are interested in calibrating a Sutton–Chen EAM potential, described by three parameters, ϵ, c and A, given a dataset consisting of four quantities of interest: lattice parameter, cohesive energy, unrelaxed vacancy formation energy and bulk modulus.

Uncertainties in parameters ϵ, c or A will thus be tied to uncertainties in each of the quantities of interest and the ability, or lack thereof, of the model to reproduce the training data. The uncertainties in the training data have various origins; in the case of DFT data, they originate from choice of the exchange and correlation functional, pseudopotential and numerical approximations, and in the case of experiments, originate from measurement errors or sample-to-sample variability. Uncertainties can also arise from the ability of the model to represent the training data. Figure 2.2 shows the calibrated parameters for a Sutton–Chen type EAM potential with three parameters [66].

Also shown in Figure 2.3 are the distributions for the output quantities of interest,

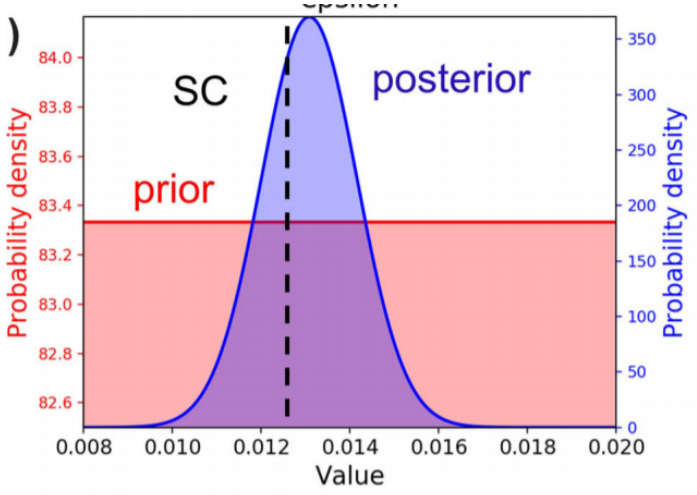

Fig. 2.2. Prior and posterior probability distributions for ϵ, one of the parameters of the interatomic model

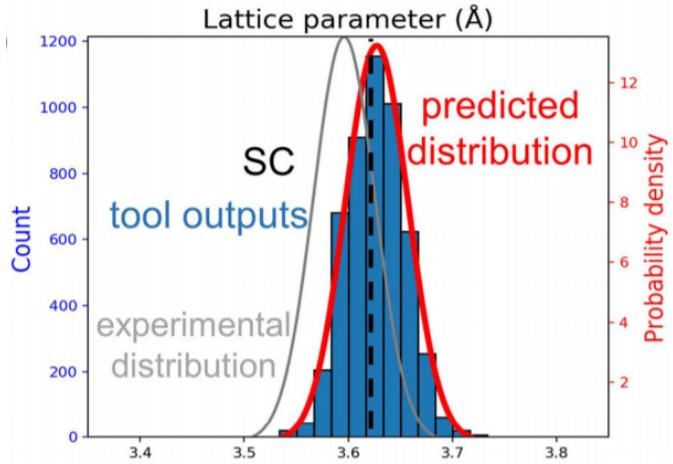

Fig. 2.3. Predicted probability distributions for each of the four output quantities of interest shown in red, with the blue histogram depicting brute force calculations for the chain of points explored by Dakota's Markov Chain Monte Carlo algorithm. The experimental data is shown in grey, while the predictions for the parameters in the Sutton Chen formulation are shown as a black vertical line

We find that our calibration results in potential parameters that aim to match the experimental distributions such that quantities with low uncertainty (lattice parameter and bulk modulus in our training set) are described better than quantities with higher uncertainty. This reflects the fact the posterior distributions strongly depend on the uncertainties in the training data, and the calibrated parameters will result in optimal output distributions for quantities with low variances, as opposed to quantities with high variances, for which the output distributions will not be as optimal, reflecting the uncertainty in the training data. We note that our parameter distributions are slightly different from the parameters proposed by Sutton and Chen, emanating from different training sets and approaches. Our new calibrated parameters tend to underestimate the ratio between vacancy formation energy and cohesive energy as compared to our ab initio-based training set and attempts to push the distributions into better agreement with the training set.

Additional details on this work can found in our publication: S. Desai, M. Hunt, and A. Strachan, 'Online tools for uncertainty quantification in nanoHUB', *JOM* [67].

The following chapters in this thesis will focus on the application and improvement of molecular dynamics simulations for specific applications, in addition to the development of novel, interpretable machine learning models. Chapter 3 focuses on the use of large-scale molecular dynamics simulations as a tool to explore trends in shape-memory alloy properties, guiding the design of room-temperature light-weight shape memory alloys.

3. MOLECULAR DYNAMICS SIMULATIONS TO OPTIMIZE SHAPE MEMORY ALLOYS

Having described the details of a molecular dynamics simulation, we will now consider an example of using molecular dynamics to explore the design space of shape memory alloys and provide guidelines on designing lightweight, room-temperature operable shape memory alloys. This example also serves as a demonstration of using simple but computationally cheap interatomic models to explore trends in material properties before developing specific interatomic models to predict properties with greater accuracy.

This chapter is organized as follows: Section 3.1 introduces shape memory alloy and the martensitic transformations governing these alloys, as well as the need to explore the design space of these materials with molecular dynamics. Section 3.2 describes the choice of interatomic potential and the procedure to build atomistic structures of the materials involved. Section 3.3 provides simulation details and Sections 3.4 and 3.5 describe the results of these simulations. We draw conclusions from our results in Section 3.6 and discuss the implications of our results in providing guidelines to design lightweight, room temperature shape memory alloys.

The work in this chapter has been published in the Journal of Applied Physics and can be found as: Saaketh Desai, Sam Reeve, K.G Vishnu, Alejandro Strachan, "Tuning martensitic transformations via coherent second phases in nanolaminates using free energy landscape engineering", Journal of Applied Physics 127, 125112 (2020).

3.1 Introduction

Shape memory alloys are materials which recover deformations applied to them at low temperatures upon heating back to high temperatures. Shape memory alloys also display superelasticity, where large deformations can be applied to the material without inducing plasticity, leading to complete strain recovery upon unloading. These alloys are governed by a transformation between a high-temperature, high-symmetry austenite phase and a low-temperature, low-symmetry martensite phase. Martensitic transformations can be either temperature-induced or stress-induced, and are the underlying cause for shape memory and superelasticity. Martensitic transformations are thus desirable for a range of applications from connectors and micro-actuators [68] to tires for Mars exploration rovers [69]. The effective design of shape memory alloys for these applications hinges on the ability to tune the underlying martensitic transformation for the specific application. For example, low hysteresis is desirable for actuation [68, 70], but the opposite is sought for mechanical damping [71]. Such optimizations have traditionally been pursued by modifying the composition of the alloy, using either physics-based approaches [72, 73] or via high-throughput experimental searches, which have identified ternary and quaternary alloys with ultra-low thermal hysteresis [74, 75]. More recently, machine learning principles coupled with high-throughput density functional theory calculations and experiments have also been used to discover alloy compositions with ultra-low hysteresis [76]. While these efforts have shown significant success, additional avenues to tune the properties of martensitic materials are desirable as they can open the design space and potentially result in significantly improved properties. An example of this need is the β-type family of Mg-Sc martensitic alloys, whose low density (about one third of NiTi based alloys) makes them attractive for aerospace and energy storage applications, yet their low operating temperatures currently make them impractical [77, 78]. Specifically, a Mg-20.5 at.% Sc alloy showed superelasticity at -150 °C, while a Mg-19.2 at.% Sc alloy showed a thermally induced martensitic transformation starting at -100 °C.

The incorporation of coherent second phases has emerged as a novel avenue to tune the thermo-mechanical response of shape memory alloys (SMAs). The first demonstration of this showed, via molecular dynamics simulations, that the incorporation of a second phase with desirable characteristics can reduce the hysteresis associated with the martensitic transformation in NiAl alloys [79]. Recent experiments have shown ultra-low fatigue in NiTi-Cu SMAs via the precipitation of coherent nanoscale Ti_2Cu [80]. Other experiments have also seen favorable changes in transformation characteristics in NiTi-Hf and NiTi-Pt SMAs due to the formation of coherent second phases [81,82]. Similarly, nanoscale phase separation via spinodal decomposition in a Ti-Nb gum metal creates a nanoscale composition variation, which in turn results in local confinement of the transformation and superelasticity over a wide range of temperatures [83]. In addition to second phases obtained through traditional metallurgical processing, epitaxial growth of 5 nm Mg-Nb nanolaminates suppressed the martensitic transformation in Mg, stabilizing the metastable bcc phase at ambient pressure [84]. Similar work has shown the ability to stabilize metastable phases in Cu-Mo thin films [85].

Our previous work with the concept of free energy landscape engineering (FELE) also demonstrated the ability to use coherent second phases to tune transformation characteristics in a controlled manner. Building on Ref. [79], MD simulations have demonstrated that adding a non-martensitic second phase to a martensitic base material, in the form of epitaxial nanolaminates, core-shell nanowires, or nanoprecipitates, can result in reduced thermal hysteresis, tunable transformation temperatures, and even ultra-low stiffness in a fully dense metal or second order martensitic transformations [86–88]. Ab initio simulations have also explored strain engineering to increase the martensitic transition temperature in Mg-Sc alloys [89].

While prior work has demonstrated the effect of a specific second phase on transformation characteristics and associated properties [85, 90], we lack a general understanding of how the properties of the non-martensitic second phase (relative to the martensitic alloy) map onto the properties of the overall material. Here, we use

MD simulations to characterize the tunability of martensitic transformation temperatures, thermal hysteresis, and transformation strain in a model system by adding a family of second phases with systematically changing free energy landscapes, with the aim of providing guidelines for choosing precipitates (or other nanostructures) that enable the discovery of novel lightweight SMAs that can operate at room (or elevated) temperature. Our choice of a nanolaminate configuration is partially motivated by the success of strain engineering to enhance semiconductor properties, as exemplified in the increased mobility of strained silicon grown epitaxially on a SiGe layer [91]. Our prior work has documented in detail the microstructure changes for more metallurgically relevant geometries such as precipitates [87].

3.2 Hyperparameter dependent Morse potential

While metallic alloys, including martensites, are typically described with embedded atom model (EAM) or modified EAM (MEAM) potentials, Guthikonda and Elliott developed a generic Morse potential to describe martensitic transformation in binary systems [92]. The potential parameters are a function of a hyperparameter denoted as θ, which enables a continuous change in the stability of the martensite and austenite and tuning of the transformation. The potential, accessible through the OpenKIM repository [93], was developed to describe an Au 47.5 at.% Cd SMA (for θ = 400), accurately describing the lattice parameters, thermal expansion coefficients, and bulk moduli for the B2 (austenite) and B19 (martensite) phases, in addition to the transformation between the B2 and B19 phases. Since our interest is in a model martensitic material and not in the details of the AuCd system, we will denote the two atom types A and B and treat the potential as one that describes a binary alloy with a high temperature cubic (austenite) phase and, for certain values of the hyperparameter, a transformation to a low temperature monoclinic phase (martensite) and potential transformation back to the cubic phase. The hyperparameter θ varies the three parameters describing all interactions: cohesive energy, stiffness, and lattice

parameter (D_0, α, and r_0) between different pairs of atom types. For given θ, varying r_0 allows us to simulate a family of second phases with various lattice parameters but otherwise similar behavior and phase stability. Similarly, slightly reducing the value of the hyperparameter θ results in a second phase with lower stiffness without substantially different phase stability; see Table 3.1 for more details.

Table 3.1.
Morse potential parameters for the base material and the stiff and soft set of second phases

Material	Interaction	D_0	α	r_0
Base ($\theta = 400$)	A - A	0.152716	1.46152	3.15313
	B - B	0.482113	1.53431	3.04440
	A - B	0.199790	1.76427	3.08713
2^{nd} phases (stiff set, $\theta = 1000$)	A - A	0.17777	1.25703	3.19045 - 3.37045
	B - B	0.437791	1.23394	3.12694 - 3.30694
	A - B	0.216752	1.61549	3.06538 - 3.24538
2^{nd} phases (soft set, $\theta = 800$)	A - A	0.16684	1.34838	3.20644 - 3.31644
	B - B	0.46330	1.36743	3.11380 - 3.22380
	A - B	0.20948	1.69309	3.11024 - 3.22024

3.3 Simulation Details

3.3.1 Simulating thermally induced martensitic transformations

We first built a disordered alloy of composition A 47.5 at.% B by replicating the B2 unit cell 100 times in the x, y, and z directions, resulting in a system that contains 2,000,000 atoms and with dimensions of 33.5 nm in each direction. Atom types were randomly swapped until the composition of each system was 47.5 at.% B. The simulation domain was chosen to be large enough to minimize size effects in the

predicted transformation temperatures. Finite size effects are observed in molecular dynamics simulations of phase transitions, as observed before in our prior work with NiAl alloys [94], as well as in other solidification simulations [95]. An important contributor to these size effects in this case is the disordered nature of the alloys as small simulation domains limit the composition heterogeneities present. To simulate thermally induced martensitic transformations, or lack thereof, in various systems of interest (base martensitic material, each of the second phases, and the epitaxial nanolaminates), each system was relaxed at 1600 K (above M_s for all systems) for 10 ps under constant stress and temperature (NPT) conditions, allowing all simulation cell angles to evolve independently, using damping constants of 10 fs and 100 fs for thermostat and barostat, respectively. We observed that all stress components were near zero and the potential energy and lattice parameters stabilized after a 10 ps equilibration. Each relaxed structure was then cooled to 200 K at a rate of 5 K/ps under the same NPT conditions through austenite–martensite transition and subsequently heated back to 2200 K, also at 5 K/ps.

Following this protocol for the base alloy ($\theta = 400$), we find that the M_s temperature varies strongly as the simulation domain increases from ~16,000 atoms to 1,000,000 atoms where doubling the size to 2,000,000 atoms results in minimal change, see Figure 3.1. We thus use a system size of 2 million atoms for all simulations in this work.

3.3.2 Free energy landscape calculations

The relationship between free energy and the lattice parameter of the simulation cell at various temperatures governs the thermodynamics and kinetics of the martensitic transformation [96]. This free energy landscape for each system is calculated by applying a biaxial strain on the austenite phase, in $[100]_A$ and $[010]_A$ directions (where A refers to austenite). A strain of up to 5% was applied in both tension and compression for second phases and nanolaminates, while a strain of 10% in tension is

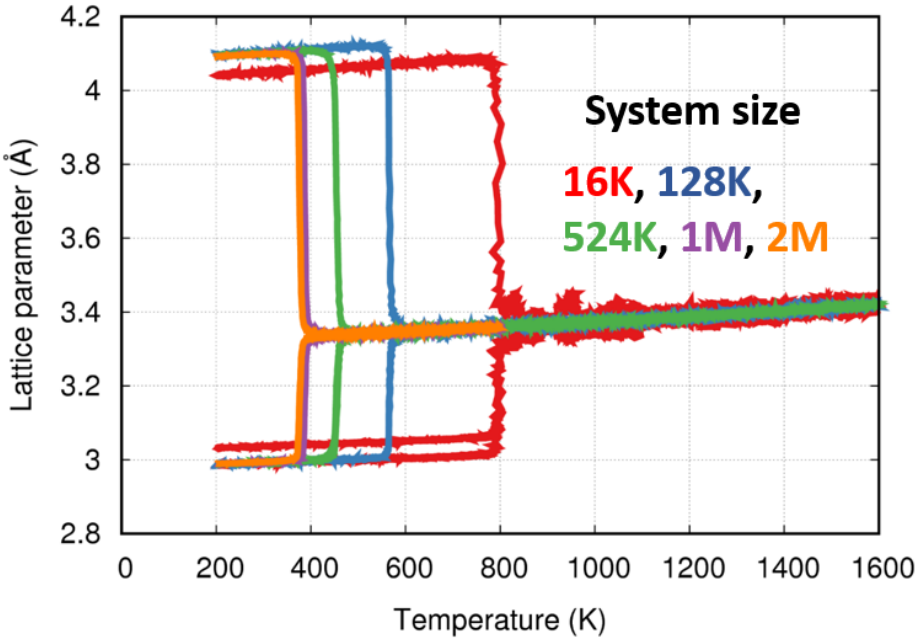

Fig. 3.1. Size effect on predicted M_s temperature shown via cooling simulations starting from the austenite phase at 1600 K. The labels indicate the number of atoms in the system (~16,000, ~128,000, ~524,000, ~1 million and 2 million atoms respectively). The M_s temperature, detected by the change in lattice parameter varies widely, becomes independent of system size for a system containing greater than 1 million atoms.

needed for the base material to cover the full transformation path. The components of stress and strain tensors are integrated to obtain the free energy along the path,

$$\Delta F = -\int \sigma_{xx} d\epsilon_x + \sigma_{yy} d\epsilon_y + \sigma_{zz} d\epsilon_z + \sigma_{yz} d\epsilon_{yz} + \sigma_{xz} d\epsilon_{xz} + \sigma_{xy} d\epsilon_{xy} \tag{3.1}$$

where the stress tensor is calculated by LAMMPS [21], as detailed elsewhere [97], and involves computing the virial, while the strains are computed using the conventional formulae based on changes in box lengths and angles. We note that the resulting energy landscapes are only approximate representations of the free energy as they depend on the strain rate applied and the path assumed for the transformation (in this case, uniform biaxial deformation). Computing a number of these landscapes

and applying Jarzynski's equality [98] can address these limitations of the calculation and relate our non-equilibrium free energy (work or potential of mean force) calculations to the equilibrium free energy landscape. While our approximations do not allow for quantitative predictions of transformation temperatures, they provide useful trends to understand how the properties of the family of second phases vary. All the landscapes shown in this work use a strain rate of $5\text{x}10^9$ s^{-1}.

3.3.3 Base phases and potential parameters

The hyperparameter θ of the interatomic model allows a description of both martensitic and non-martensitic materials. For $\theta = 400$, resulting in Morse parameters shown in 3.1, a martensitic transformation occurs with an M_s temperature of 390 K, as shown in Fig. 3.2(a), with the cubic austenite phase transforming to the monoclinic martensite phase. For $\theta = 1000$, the resultant parameters describe a non-martensitic alloy that does not transform thermally. The free energy landscapes, Figure 3.2 (b), also describe the martensitic and non-martensitic nature of the materials. At 1000 K, $\theta = 400$ displays a stress-induced transformation, resulting in a double-well landscape with equally stable martensite and austenite, while the landscape for $\theta = 1000$ phase is a single well for the austenite, with no transformation. Figure 3.2 (c) shows free energy landscapes for $\theta = 400$ across temperature, with the martensite phase increasing in stability as the temperature is decreased. To describe epitaxial nanolaminates consisting of both martensitic and non-martensitic phases, we use the random structure generated as described above and add Morse potential parameters for the cross terms, Figure 3.2 (d). The top half of the cell with the nanolaminate consists of the non-martensitic second phase (atom types C and D), while the bottom half describes the martensitic phase (atom types A and B). Interactions between cross-laminate atom types are then given by mixing rules described by the equations below, similar to mixing rules commonly used in other MD simulations [99]. Here, interactions between similar atom types (A and C or B and D) are

given by K = I and L = J and interactions between dissimilar atom types (B and C or A and D) are given by (K, L) $\notin$ (I, J).

We note that in some samples, we observed a transformation to a different martensite phase (with tetragonal symmetry), see Figure 3.3. Since these martensitic phases rarely occur in the laminate materials studied here, we refer to the monoclinic martensite in the remainder of this chapter.

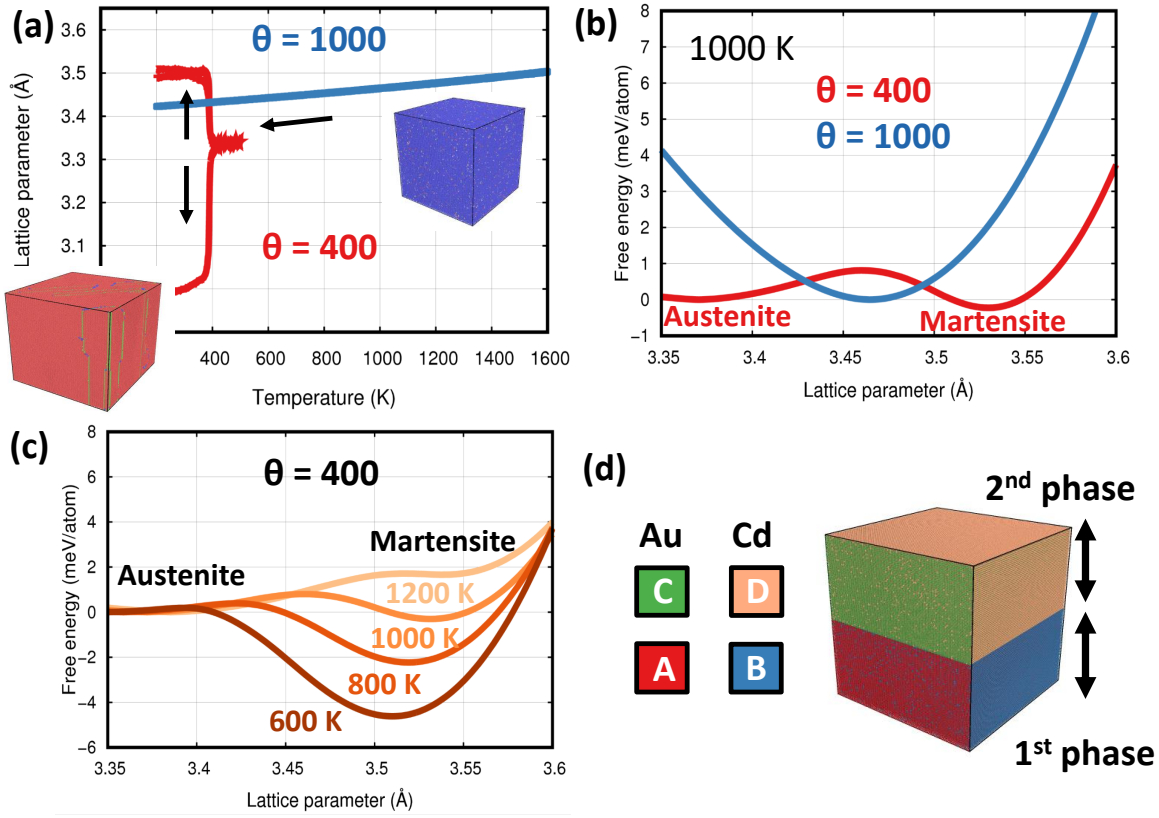

Fig. 3.2. (a) Cooling simulations showing the martensitic transformation for $\theta = 400$ at ~390 K, while $\theta = 1000$ does not transform. The arrows indicate the direction of change in the lattice parameter and the inset snapshots show initial and final (austenite and martensite) structures for $\theta = 400$. (b) Free energy landscapes for both phases. The double-well structure for $\theta = 400$ shows the stress-induced martensitic transformation, absent for $\theta = 1000$. (c) Free energy landscapes for $\theta = 400$ at various temperatures. (d) Initial structure illustrating the four atom types used to describe nanolaminates

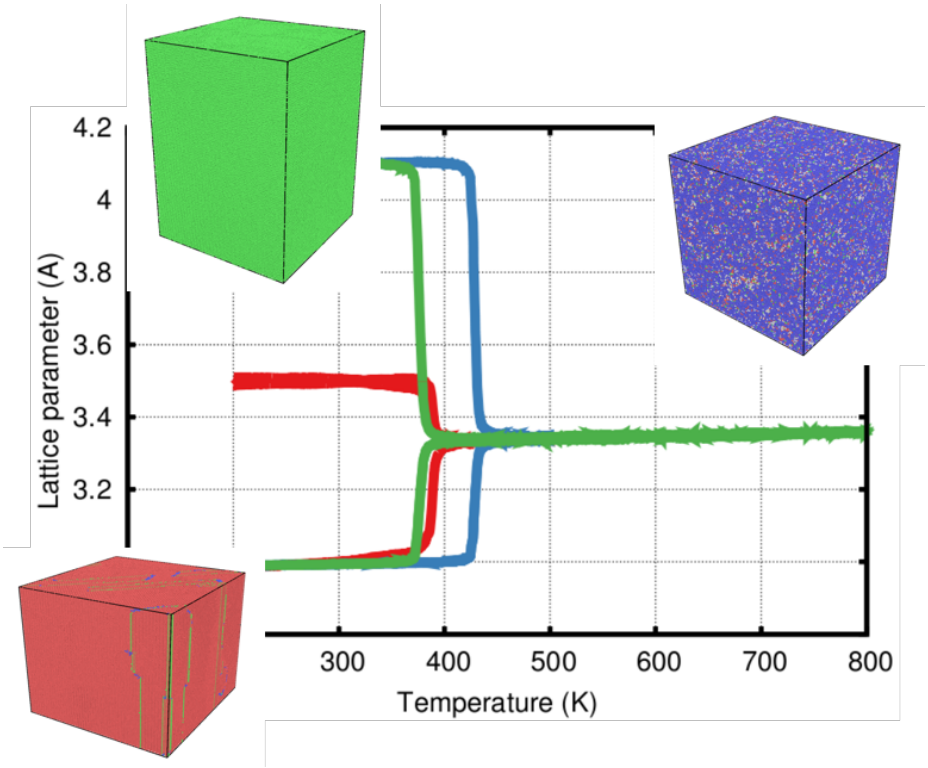

Fig. 3.3. Cooling simulations of the base material ($\theta = 400$) showing transformations of the austenite phase (blue) to the monoclinic (red) and the tetragonal (green) martensite phases

3.4 Effect of second phase lattice parameter on transformation characteristics

3.4.1 Second phase lattice parameter between the base material austenite and martensite

Effect on transformation temperatures and microstructures

To understand the change in transformation characteristics induced by the lattice parameter of the second phase (relative to the base martensitic material, i.e., misfit strain), we start with six candidate second phases whose lattice parameters span from

the austenite to the martensite and epitaxially combine them as nanolaminates with the base martensitic material, described by $\theta = 400$, with 50 at.% of the second phase. The family of second phase materials is described by $\theta = 1000$ in the model Morse potential (resulting in a single cubic phase), with individual second phases obtained by changing r_0 parameter to obtain the desired range of equilibrium lattice parameters, see Table 3.2 for the full parameter set. The free energy landscapes of each candidate second phase (denoted P1-P6) and the base material, at 600 K, are shown in Figure 3.4. The family of second phases ranges from having near zero lattice misfit to the austenite to having near zero misfit to $[100]_A$ and $[010]_A$ directions of the monoclinic martensite. Note that lattice parameters of the monoclinic martensitic phase and our nanolaminate arrangement allow for near zero in-plane misfit to the martensite despite the difference in symmetry between the two phases.

Table 3.2.
r_0 values for each individual phase from the first stiff set of second phases (P1-P6), with $\theta = 1000$, D_0 and α in Table 3.1

r_0	P1	$P1_1$	$P1_2$	P2	P3	P4	P5	$P5_1$	P6
A - A	3.26045	3.26295	3.26795	3.27045	3.29045	3.31045	3.33045	3.35045	3.37045
B - B	3.19694	3.19944	3.20444	3.20694	3.22694	3.24694	3.26694	3.28694	3.30694
A - B	3.13538	3.13788	3.14288	3.14538	3.16538	3.18538	3.20538	3.22538	3.24538

The cooling simulations, Figure 3.4 (b), indicate a significant ability to modify the transformation temperature via coherency stresses from the coherent second phase. For this model material, the simulations show that the addition of a second phase can decrease the Ms temperature by up to ~50% or increase it by up to ~200% depending on the lattice mismatch. The M_s temperatures for laminates constructed from candidate phases P1 and P2 (approximately 120 K and 375 K, respectively) are lower than the base material (~390 K). On the other hand, laminates constructed from phases P3-P5 (with lower misfit strain with the martensite phase) show Ms temperatures higher than the base material (approximately 730 K, 970 K, and 1130

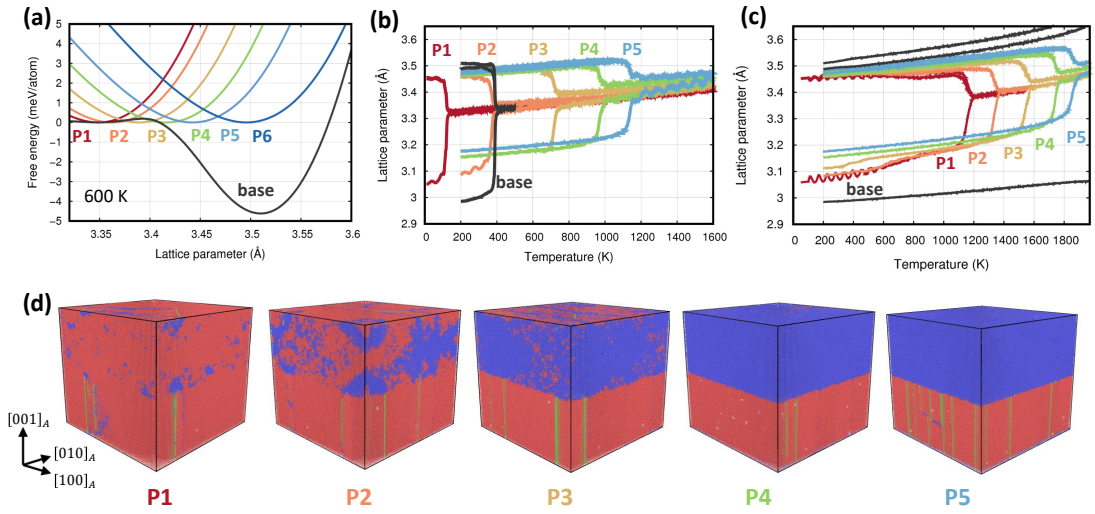

Fig. 3.4. Free energy landscapes of the six candidate phases, denoted P1 to P6, to be epitaxially integrated with the 'base' material, described by $\theta = 400$. Each second phase is non-martensitic, as indicated by the single well energy landscape (b) Cooling simulations showing Ms for nanolaminates with P1 to P5 (c) Heating simulations showing A_f for P1 to P5 nanolaminates and lack thereof for the 'base' material (d) Atomic snapshots (at 200 K), showing the transformed laminates (blue denotes the austenite phase, red martensite, and green defects)

K, respectively). Somewhat surprisingly, see Figure 3.4 (c), all laminates containing any of the P1-P5 phases result in a martensite to austenite transition upon heating, and this is not seen in the base material. Thus, adding any of the second phase studied reduces the A_f temperature, even when the austenite phase is stabilized. As expected, this reduction becomes more pronounced as the lattice parameter of the second phase approaches that of the austenite phase. The mechanisms behind this trend are discussed in Section 3.4.2.

Effect on microstructure: In all cases, despite misfit strains approaching 10%, the laminates remain coherent over this wide range of strains due to the non-convex energy landscape of the martensitic phase that results in significantly lower elastic strain than a linear elastic material and nanoscale dimensions of the laminate peri-

odicity. This is consistent with experimental observations in Fe–Pd magnetic shape memory alloys, where coherent epitaxial growth was achieved for laminates as thick as 50 nm, with the substrate applying strains as large as 8% [100]. We note that the boundary conditions used here make it difficult to lose coherency, where open lateral boundaries would be more appropriate to study coherency limits [101]. Snapshots of these systems at T = 200 K, Figure 3.4(d), show that we form only one martensitic domain whose close packed plane is oriented along $(110)_A$, with stacking faults observed on $(110)_A$ and $(\bar{1}10)_A$ planes. We also observe that the non-martensitic alloy (top half of the simulation cells) is driven to transform into the martensitic phase (atoms with local martensitic structures are colored red) due to epitaxial stress caused by the martensitic alloy. The laminate involving P2 second phase transforms partially, and both martensite and austenite phases coexist. For laminates P3-P5, the epitaxial stress from the martensite phase on the second phases is not enough to drive the transformation, and the snapshots in 3.4(d) indicate transformation of only the base martensitic phase, again with a single domain.

Figure 3.5 highlights the dependence of M_s, A_f, and thermal hysteresis on the lattice mismatch of the second phase. For reference, we include M_s of the base material (dashed red line) and its melting temperature (dashed blue line) since the base material does not have an A_f temperature. The M_s and A_f temperatures increase significantly as the lattice mismatch with the austenite phase increases. A second phase matching the martensite lattice parameter (P6) completely suppresses the martensite to austenite transformation. Under the conditions studied, the austenite to martensite transformation is never completely suppressed, even when the second phase matches the lattice parameter of the austenite phase. We attribute this to low stiffness of the austenite phase (as compared to the martensite), making it relatively easy to transform to the martensite phase. A larger volume fraction of the second phase or a second phase with higher stiffness would further stabilize the austenite phase and could suppress transformation. Intermediate misfit strains, corresponding to a second phase with a lattice parameter between austenite and martensitic phases, re-

sult in largest reduction in the activation barrier associated with transformation and, consequently, lead to lowest hysteresis. This is consistent with prior results in NiAl alloys [79,87]. We note that our hysteresis values are large compared to experiments; this can be attributed to the defect-free nature of our initial structures. We have previously observed large hysteresis for defect-free NiAl systems [79,87]. Our results, thus, indicate a potential avenue to increase the Ms temperature of a martensitic material, as desired for the case of lightweight Mg-Sc shape memory alloys [77, 78]. The incorporation of a relatively soft second phase with low misfit strain with the martensite, as demonstrated by second phases P3-P5, could increase the transformation temperature of these alloys to room temperature or above.

Underlying free energy landscapes of the nanolaminates

To understand the trends described in Section 3.4.1, we study the free energy landscapes of the family of nanolaminates. We approximate these landscapes by adding the landscapes of the base material and the candidate second phase in equal proportions (since the second phase constitutes 50 at.% of each laminate). Figures 3.6(a) and 3.6(b) show free energy landscapes of the P2 and P4 second phases, respectively, with thin colored lines, the landscape of the base alloy in black, and the analytically combined laminate landscapes with thick colored lines. Landscapes are computed at temperature (T_0) where the free energies of austenite and martensite phases are equal, i.e., the thermodynamic transformation temperature of the laminate (not of the base phase). The features of a landscape that affect the transformation temperature are the energy difference between the martensite and austenite (the thermodynamic driving force) and the barrier for transformation (kinetics).

We first focus on the changes in M_s, A_f, and hysteresis achieved by adding the second phase, relative to the base material. Figure 3.6(a) indicates that adding the P2 phase has the effect of stabilizing the austenite with respect to the martensite. The reduction of the driving force to transform to martensite and slightly increased

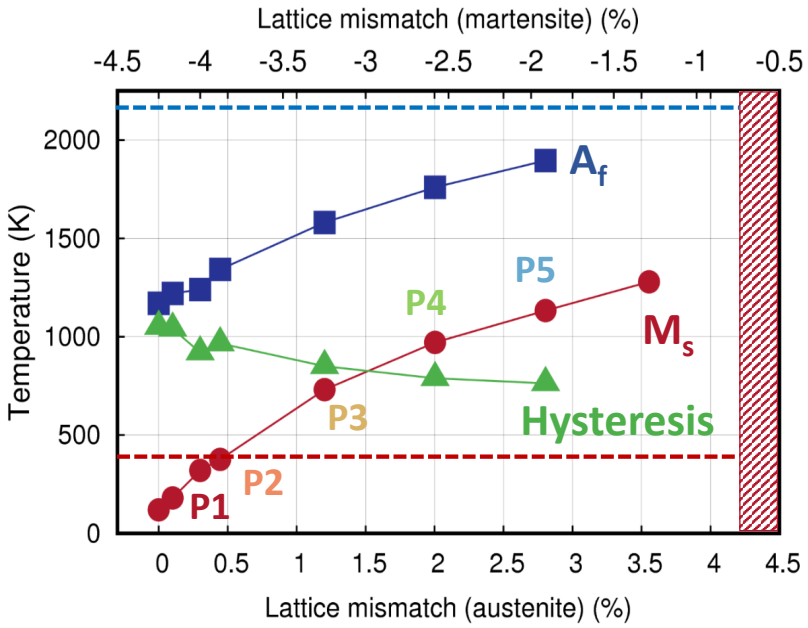

Fig. 3.5. Trends in M_s, A_f, and hysteresis as a function of misfit strain (or lattice mismatch) to the austenite and martensite phases of the base material. The red bar indicates a region (phase P6) where the martensite phase is fully stabilized. The dashed red and blue lines represent the Ms and melting temperature (due to lack of A_f) of the base material, respectively

energy barrier would be expected to result in a lower M_s temperature, which matches the direct cooling simulations. The P1 laminate shows similar behavior. In laminates P3-P5, with P4 as an example in Figure 3.6(b), the significant reduction in the transformation barrier can be expected to facilitate the martensitic transformation, even with a smaller driving force, increasing M_s as seen in Figure 3.5. Candidate phase P6 fully stabilizes the martensite, spontaneously transforming to martensite even near the melting temperature, and does not show martensite to austenite transformation on heating; correspondingly, it has a single well landscape. Regarding martensite

to austenite transition on heating, the reduction in the transformation barrier enables transformation to austenite that is suppressed in the base material; this is clear in Figs. 3.6(a) and 3.6(b). The hysteresis depends on the energy barrier between the austenite and the martensite phase at the thermodynamic transformation temperature, and Figure 3.6(c) compares the landscapes of the base material and each nanolaminate. We can confirm that the energy barrier between the austenite and the martensite phase is significantly reduced in the laminates as compared with the base alloy, explaining the reduced hysteresis in thermally induced transformations. Figure 3.6(d) compares free energy landscapes across phases P1-P6 at a single intermediate temperature, complementing the information presented above and allowing direct comparisons among the second phases themselves. This again confirms that the phase with the lowest thermodynamic transformation temperature (P1) and the highest transformation barrier (requiring large undercooling and overheating) will show the lowest M_s and the lowest A_f, and that the M_s and A_f temperatures would increase from P1 to P5, which is what we observe in Figure 3.5.

3.4.2 Second phase lattice parameter beyond the base material austenite

To characterize the limits of FELE in modifying the transformation temperature, we designed a second family of second phases, P1* to P5*, with lattice parameters smaller than those of the austenite phase, see Table 3.3 for potential parameters. The landscapes for these second phases, in comparison to the base material, are shown in Figure 3.7(a); direct heating and cooling simulations are shown in 3.8. One could naively expect these second phases to further stabilize the austenite phase relative to the martensite and reduce M_s and A_f further, continuing the trend described in Section 3.4.1. The cooling simulations, Figure 3.7(b), show that none of these second phases fully stabilize the austenite. Quite the opposite, phases P4* and P5* stabilize the martensite resulting in Ms temperatures of ~800 K and ~1000 K, comparable with phases P4 and P5. To explain this result, one must consider the difference in

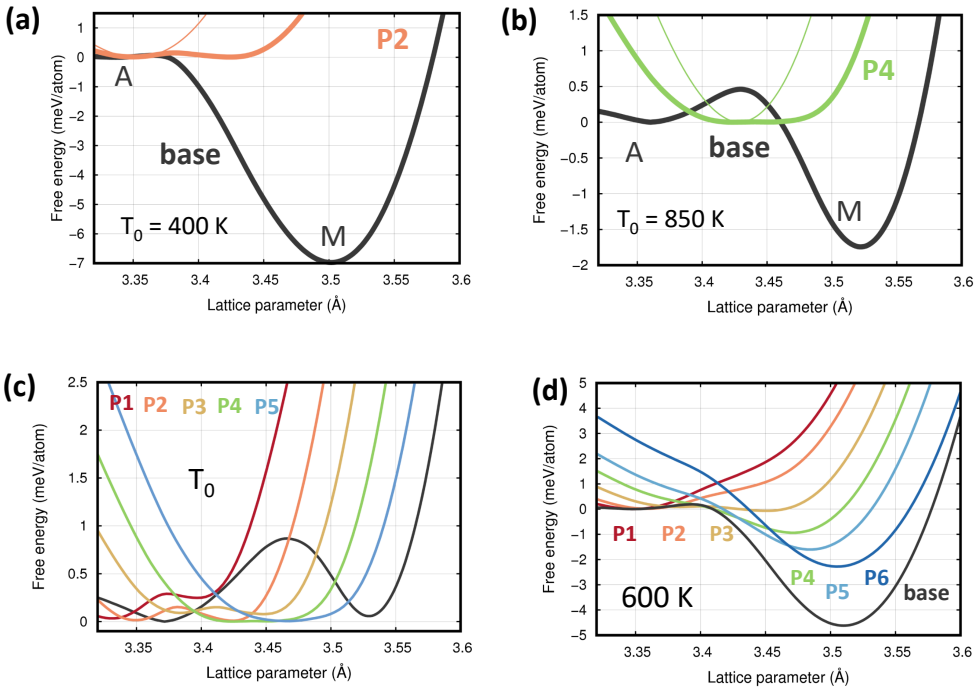

Fig. 3.6. (a and b) Comparisons of base material landscapes with numerically combined landscapes for laminates P2 and P4, shown as examples. 'A' indicates austenite and 'M' indicates martensite. (c) Free energy landscapes at the thermodynamic transformation temperature T_0 (d) Free energy landscapes for each nanolaminate at 600 K. In all landscapes, the horizontal axis is the lattice parameter in the $[100]_A$ and $[010]_A$ directions

symmetry between the phases. Reducing the lattice parameter of the cubic second phase increases the misfit strain with respect to the cubic austenite in both in-plane directions. However, one of the lattice parameters of the monoclinic martensite is significantly shorter than the other two. Thus, reducing the lattice parameter of the second phase creates an opportunity for a new martensite variant to form where the small lattice parameter accommodates the misfit strain imposed by the lattice mismatch instead of the alignment normal to the interface as is the case in P1-P5 simulations. The explicit cooling simulations show this, with second phases P1-P5 forming only one martensite variant, while second phases P1*-P5* result in two distinct vari-

ants coexisting in elongated domains to accommodate overall strain. Interestingly, we observed phases beyond P5* to fully stabilize the tetragonal martensite phase (see Figure 3.3) since the in-plane lattice parameter of the second phase matches the lattice parameter of the tetragonal martensite.

Table 3.3.
r_0 values for each individual phase from the second stiff set of second phases (P1*-P5*), with $\theta = 1000$, D_0 and α in Table 3.1

r_0	P5*	P4*	P3*	P2*	P1*
A - A	3.19045	3.21045	3.23045	3.24045	3.25045
B - B	3.12694	3.14694	3.16694	3.17694	3.18694
A - B	3.06538	3.08538	3.10538	3.11538	3.12538

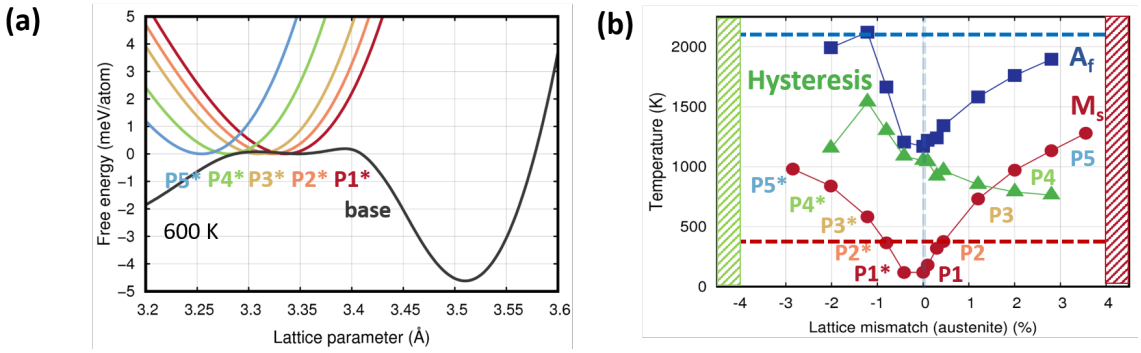

Fig. 3.7. (a) Free energy landscapes of the five candidate phases, denoted P1* to P5*, to be epitaxially integrated over the 'base' material, described by $\theta = 400$ (b) Trends in M_s, A_f, and hysteresis as a function of misfit strain. The vertical blue line demarcates phases which impose a tensile strain on the base material from phases P1* to P5* which impose a compressive strain. The red bar indicates a region where the monoclinic martensite is fully stable, while the green bar indicates a region where the tetragonal martensite is fully stable. The dashed red and blue lines represent the M_s and melting temperature (due to lack of A_f) of the base material

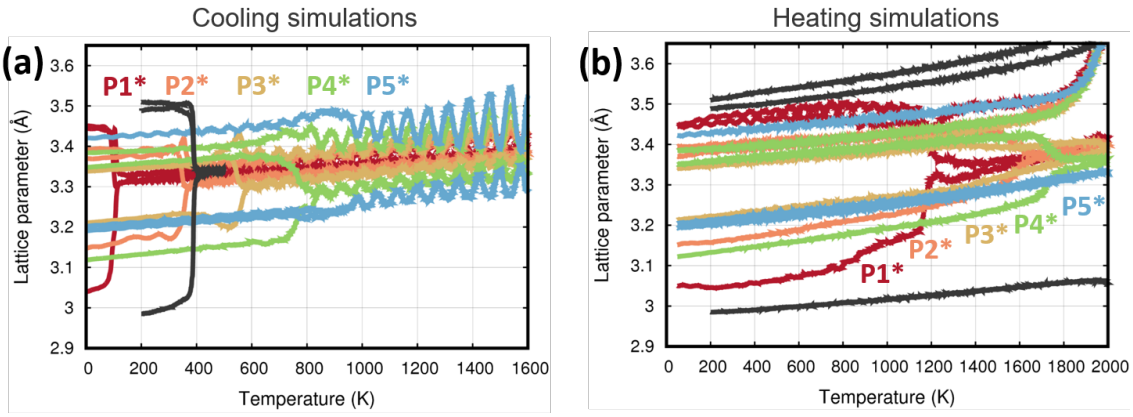

Fig. 3.8. (a) Cooling and (b) heating simulations for laminates P1* to P5*

To further understand the effect of intermediate and negative lattice strains on transformation characteristics, we estimate the strain energy added to the austenite and the martensite phase, imposed by the lattice mismatch. This is described by the equation below, where C_{ij}^{α} are the elastic constants of the α (austenite or martensite) phase and C_{ij}^{ϵ} are the strains with respect to that phase

$$E_{\alpha} = \frac{1}{2} C_{11}^{\alpha} (\epsilon_{11}^{\alpha})^2 + \frac{1}{2} C_{22}^{\alpha} (\epsilon_{22}^{\alpha})^2 \tag{3.2}$$

The strain energy of the austenite phase increases as the second phase varies from P1 to P6, see Figure 3.9, while the strain energy added to the martensite phase decreases, as expected from the landscapes in Figure 3.4. For second phases P1*–P5*, the rotated martensite variants accommodate the strain such that the strain energy added to the martensite again decreases from P1* to P5* although to a lesser degree than from P1 to P5. This implies that martensite phase stability with respect to the austenite phase increases from P1* to P5*; this corresponds to increases the M_s and A_f temperatures, as in Section 3.4.1. We note that this strain energy model only allows us to consider the in-plane lattice mismatch and its effect on the energy difference between the austenite and martensite and does not allow us to comment

on the transformation barriers discussed in Section 3.4.1. We also find that the strain energy added to austenite and martensite phases by second phases P1*–P5* is comparable in magnitude to phases P1–P5, see Figure 3.9. Thus, phases P1*–P5* stabilize the austenite and martensite phases in a similar manner as phases P1–P5, resulting in similar M_s temperatures.

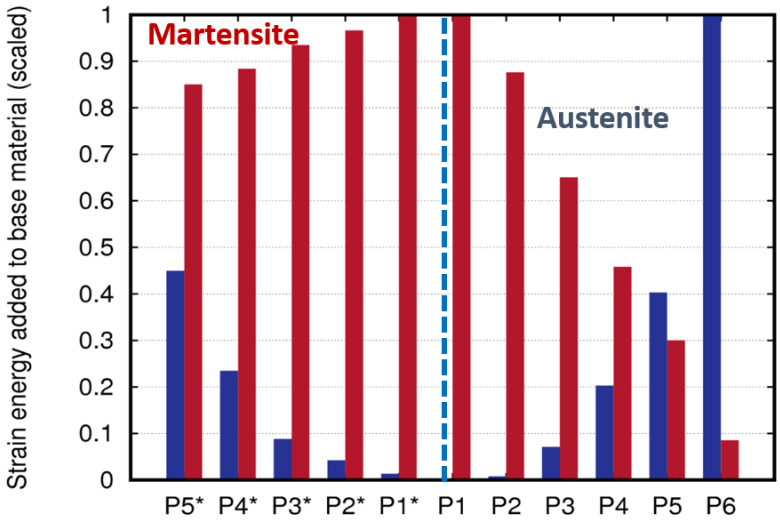

Fig. 3.9. Scaled strain energy added to the austenite and martensite phases of the base material due to the in-plane and out-of-plane lattice mismatch between the added second phase and the base material. The blue line demarcates phases P1 to P6, which impose a tensile strain (positive in-place lattice mismatch) on the base material, from phases P1* to P5* which impose a compressive strain (negative in-place lattice mismatch) on the base material

A consequence of the stabilization of new martensite variants is that phases P1*–P5* show distinct differences in the transformation, particularly in terms of defects generated and the variants of the martensite obtained. Most notably, we observe multiple domains in our microstructures despite the small simulation sizes, where one domain has its close packed plane along $(0\bar{1}\bar{1})_A$ and the other domain has its close packed plane along $(10\bar{1})_A$. The domain wall is oriented along $(\bar{1}\bar{1}0)_A$. Figure 3.10(a) compares phases P5 and P5*, where the stacking faults (green) are useful in

identifying martensite variants of different orientations. The P* family of phases contain combinations of compatible domains, creating a greater number of domains and stacking faults and therefore retained austenite upon cooling. Figure 3.10(b) shows the transition from multi-domain microstructures (P5*) to a single domain (P1*).

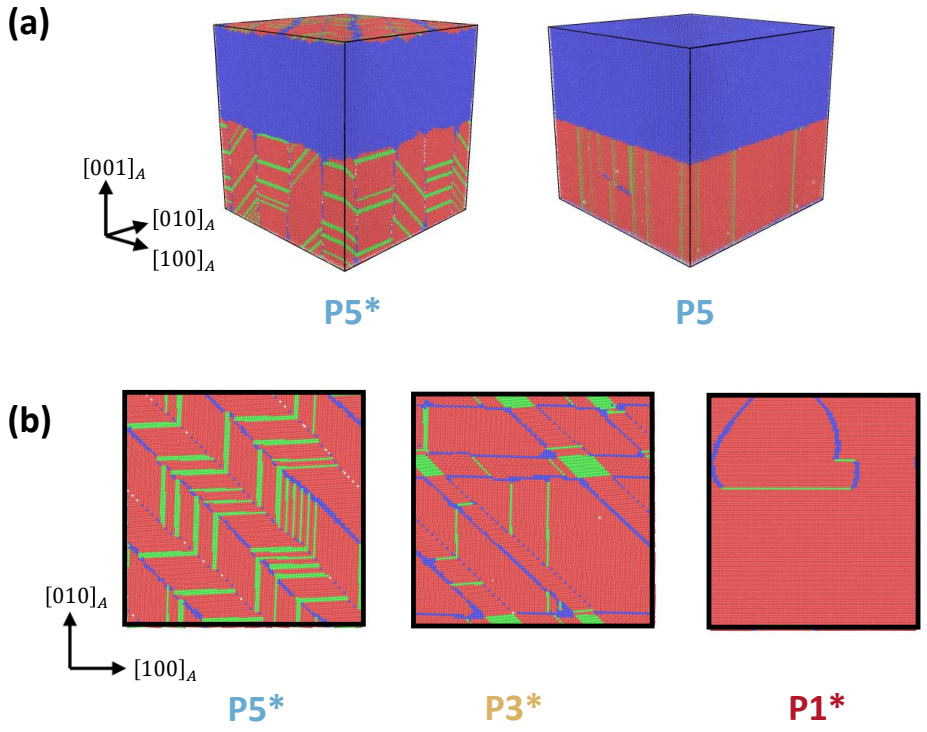

Fig. 3.10. (a) Atomic snapshots comparing the P5* laminate to the P5 laminate showing the increased defect formation (b) Slices showing the transition from multi-domain to single domain microstructures from P5* to P1*

3.5 Effect of stiffness of the second phase

To understand the effect of second phase stiffness, we now select six additional candidate phases for a third family of phases, $P1^S$-$P5^S$, using $\theta = 800$, and again

changing r_0 to systematically shift the stable lattice parameters; see Table 3.4 for potential parameters. The free energy landscapes of each of these phases, at 600 K, are shown in Figure 3.11, clearly visible to be much softer than the set of phases from Section 3.4.1 [Figure 3.4(a)], indicated by the decreased curvature of the free energy landscape of each of the second phases.

Table 3.4.
r_0 values for each individual phase from the soft set of second phases ($P1^S$-$P5^S$), with $\theta = 1000$, D_0 and α in Table 3.1

r_0	$P1^S$	$P1_1{}^S$	$P1_2{}^S$	$P2^S$	$P3^S$	$P4^S$	$P5^S$	$P5_1{}^S$
A - A	3.20644	3.20894	3.21394	3.21644	3.23644	3.25644	3.27644	3.29644
B - B	3.11380	3.11630	3.12130	3.12380	3.14380	3.16380	3.18380	3.20380
A - B	3.11024	3.11274	3.11774	3.12024	3.14024	3.16024	3.18024	3.20024

Figure 3.11(b) again indicates that laminates $P1^S$ and $P2^S$ have an M_s lower than the base material (~110 K and 330 K vs 390 K) while laminates $P3^S$-$P5^S$ have an M_s higher than the base material. The $P1^S$ and $P2^S$ laminates show an A_f of ~2000 K and 1900 K, while $P3^S$-$P5^S$ laminates have an A_f of approximately 1900 K, 1850 K, and 2000 K, respectively. Note again that the base material does not have a well-defined A_f. This trend is different from that observed for the stiff set of second phases and will be explored in detail below. The remainder of the trends and observations for this family of second phases follow from Section 3.4. Direct heating and cooling simulations are shown in 3.12.

3.5.1 Effect on microstructure

Using a softer set of second phases also allows us to tune the transformation strain, see Figure 3.11(c). As expected, the softer set of second phases show a greater transformation strain (and transformed volume) as the added second phase transforms from the austenite to the martensite (for all second phases), Figure 3.11(d). We note

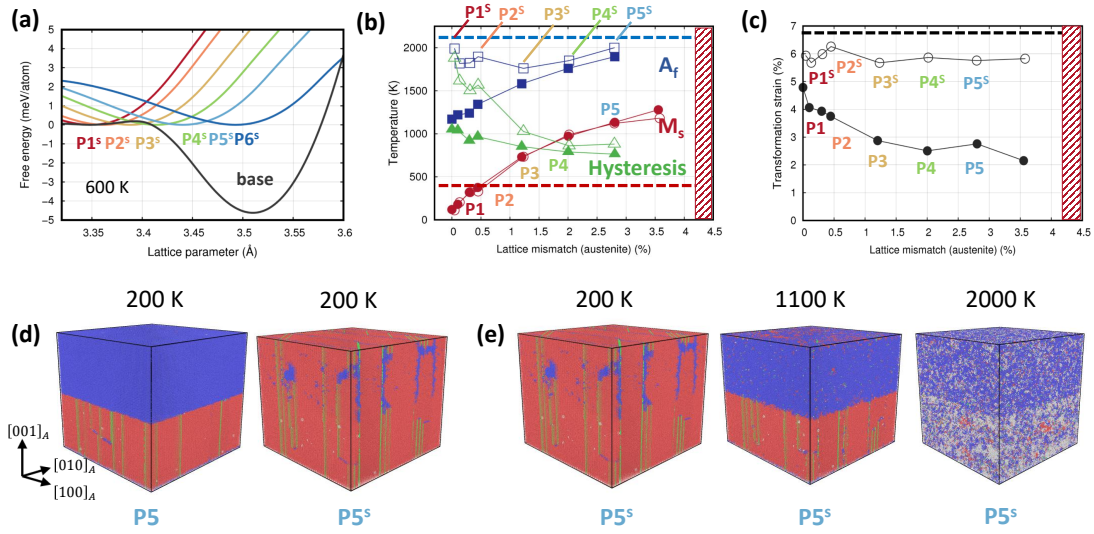

Fig. 3.11. (a) Free energy landscapes of the six candidate phases, denoted by $P1^S$ to $P6^S$, softer second phases (b) Trends in M_s, A_f and hysteresis as a function of lattice mismatch to the austenite and martensite phase. Filled symbols represent P1-P6, open symbols represent $P1^S$-$P5^S$. Refer to Figure 3.5 for more detail (c) Transformation strain for P1-P5 (closed circles) and $P1^S$-$P5^S$ soft second phases (open circles) as a function of lattice mismatch. Black dashed line represents the transformation strain of the base material. (d) Atomic snapshots showing the transformation for P5 and $P5^S$ (e) Atomic snapshots showing two-step transformation on heating for $P5^S$

that all the candidate second phases belonging to both the soft and stiff set have transformation strains lower than the base material, both because the added second phases are non-transforming on their own, and that all the candidate second phases are stiffer than the base material austenite (in tension). The microstructures observed here again show a single domain as observed in Section 3.4.1. In addition, we find that laminates P4* and P5* show a distinct two-step transformation while heating from martensite to austenite.

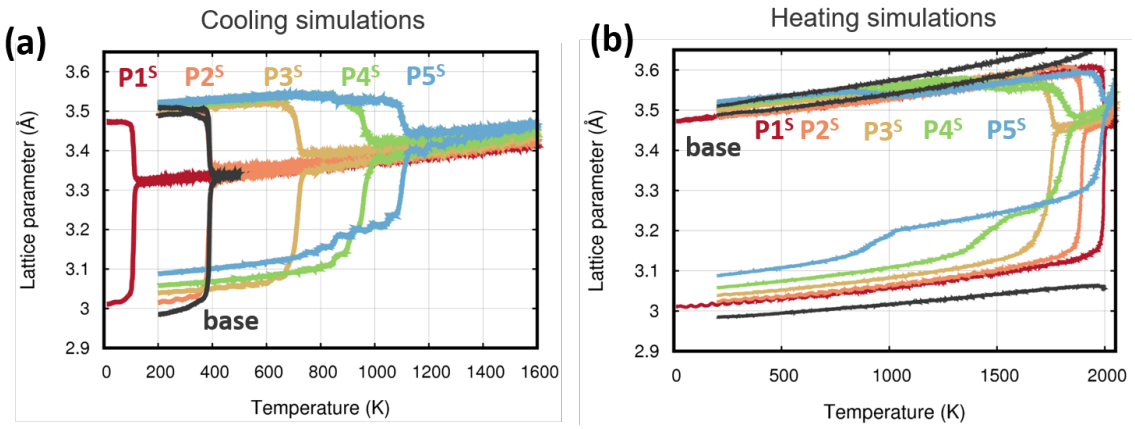

Fig. 3.12. (a) Cooling and (b) heating simulations for laminates $P1^S$ to $P5^S$

3.5.2 Distinct effects on transformation temperatures

While most trends were observed to be similar between P1–P5 and $P1^S$-$P5^S$, some key distinctions stand out. Most notably, we observe that the A_f temperature for P1 and P2 laminates is now significantly higher than the stiffer second phase laminates [see Figure 3.11(b)]. To explore this, we directly compare free energy landscapes for the P2–P5 laminates with the $P2^S$-$P5^S$ laminates, see Figure 3.13. We observe that in all cases, the austenite to martensite transformation barrier is comparable for both the soft and the stiff set of second phases; this matches the fact that the respective M_s temperatures do not differ significantly. However, the martensite to austenite transformation barrier for the P2 and P3 laminates is much higher for the soft set of phases. This similarly matches the $P2^S$ and $P3^S$ laminates' higher A_f compared to P2 and P3, respectively. The differences in landscapes become minor for P4 and P5, as do the differences in A_f. Finally, the free energy landscapes also reinforce the fact that using softer second phases results in a greater transformation strain between austenite and martensite.

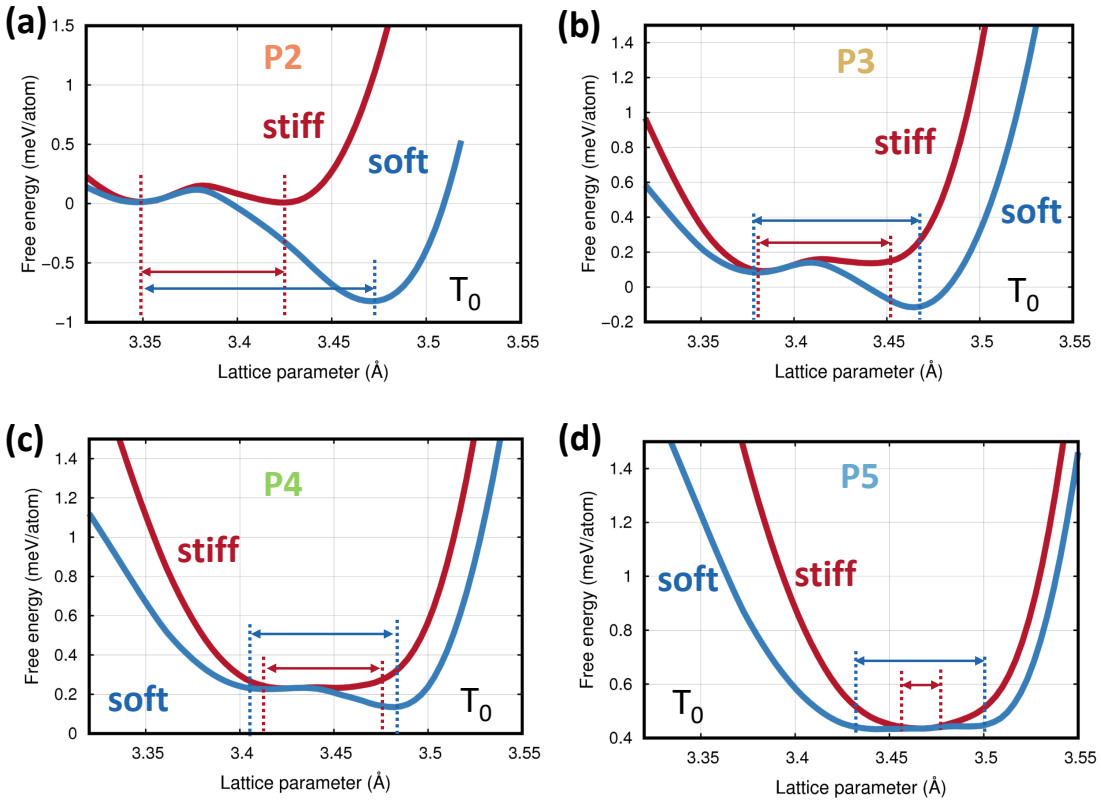

Fig. 3.13. Direct comparison of numerically combined landscapes for candidate phases (a) P2 and $P2^S$, (b) P3 and $P3^S$, (c) P4 and $P4^S$ and (d) P5 and $P5^S$. The arrows indicate the transformation strain, showing the increase in strain obtained when using softer second phases

3.6 Conclusions

We systematically investigated the effects of the coherent integration of non-martensitic second phase materials with a base martensitic alloy. We accomplished this by studying a model martensitic system, described by a generic Morse interatomic potential, varying the potential parameters that control lattice parameter and stiffness of the second phases independently. This work is an extension of our previous work in the NiAl family of SMAs, where we demonstrated tunability of M_s, A_f, and hysteresis for one second phase and multiple volume fractions [79, 87].

We find that the M_s temperature can be decreased up to 50% and increased up to 200% (with respect to the base material) by varying the lattice parameter of the second phase, with second phases having a lattice parameter close to the base martensite phase even fully stabilizing the martensite phase. We also observe a reverse transformation (martensite to austenite) in almost all nanolaminates, not seen in the base material, with each of the second phases lowering the martensite to austenite transformation barrier. The M_s and A_f temperatures increase as the lattice mismatch with respect to the austenite phase increases, with a minimum hysteresis observed for a second phase with intermediate lattice mismatch to both martensite and austenite phases, again due to a reduction in the transformation barrier. The addition of second phases results in a decrease in the transformation strain for actuation with respect to the base martensitic material due to mechanical constraints imposed by the non-transforming phase. However, this reduction in transformation strain can be minimized by using a softer second phase, with the caveat of larger hysteresis (as compared to a stiffer non-transforming phase).

This work, therefore, maps the tradeoffs between what is desired: an SMA with large transformation strain, minimal hysteresis, and transformation temperatures near the operating temperature. This can prove to be a guideline for defining and designing second phases that improve SMA characteristics, potentially the operating temperature of lightweight Mg–Sc SMAs [77,78], by incorporating a soft second phase with lattice mismatch to the martensite phase approaching zero. Future work could generalize these trends in a metallurgically relevant precipitate geometry and more specific alloys, providing further guidelines as well as exploring coherency limits.

In this chapter we explored the idea of a 'search' for a candidate material using molecular dynamics for a case where the material behavior is characterized by the diffusionless martensitic transformation. We will now turn our attention to a material design problem where the microstructure evolution is governed by longer timescale events.

4. ACCESSING GREATER TIMESCALES TO PREDICT CARBON FIBER MICROSTRUCTURE

Chapter 3 discussed an example of molecular dynamics simulations to design shape memory alloys with desired transformation characteristics. We saw how tailoring the properties of an added second phase can modify the microstructure of the alloy and thus, its properties. A thorough understanding of these structure-property relationships is critical to the design of tailored materials. While the microstructure of shape memory alloys is goverened by the diffusionless martensitic transformations, microstructure evolution in many other materials is governed by processes at a longer timescale. We now turn to one such material, carbon fibers, to understand structure-property relationships in this material with the goal of designing high-strength, high-stiffness carbon fibers. Specifically, we attempt to simulate the microstructure evolution of carbon fibers, relating the microstructure obtained to the mechanical properties displayed by the fibers. Simulating the microstructure evolution of these fibers requires accessing timescales greater than possible via conventional molecular dynamics. We thus use a combination of molecular dynamics and kinetic Monte Carlo methods to access greater timescales and simulate the microstructure evolution of carbon fibers.

This chapter is organized as follows: Section 4.1 highlights the need to simulate the microstructure evolution of carbon fibers and the need to understand the processing-structure-property relationships in these fibers to design next generation fibers. Section 4.2 describes the molecular model in the context of carbon fiber processing, providing details of the procedure used to simulate the chemical reactions and the subsequent structure relaxation. Section 4.3 specifies the details of the simulation workflow. This includes the procedures used to generate initial structures, crosslink the samples, and evaluate their mechanical properties. This section also provides a

justification of the choices employed in the model. The results obtained using our model are illustrated in Section 4.4. Our predictions for the final microstructure and its time evolution are discussed, along with quantitative experimental comparisons. Section 4.5 emphasizes the transverse moduli predicted for the simulated structures, along with a comparison to experimental results. Finally, we draw conclusions from our results in Section 4.6.

The work in this chapter has been published in the Journal of Chemical Physics and can be found as: Saaketh Desai, Chunyu Li, Tongtong Shen, Alejandro Strachan, "Molecular modeling of the microstructure evolution during carbon fiber processing", Journal of Chemical Physics 147, 224705 (2017).

4.1 Introduction

Carbon fibers (CFs) are the material of choice for a wide range of high performance composites due to their high stiffness and strength, combined with low density [102]. Early work in the field focused on pitch and cellulose (rayon) as precursors to carbon fibers; see, for example, Refs. [102] and [103]. However, polyacrylonitrile (PAN) is currently the precursor of choice for high-strength carbon fibers [104, 105]. The properties of CFs have improved significantly over the last 20 years; for example, high-strength fibers today reach tensile strengths of 12 GPa [106], significantly higher than the state of the art in the 1990s and early 2000s [102]. Despite this progress, there is significant room for improvement. While commercial carbon fibers today exhibit tensile moduli approaching that of ideal graphite (1000 GPa), the highest strength achieved to date [106] remains below 10% of the ideal value. Fiber strength is limited by the microstructure and defects; thus, the design of next-generation fibers with improved properties could be significantly accelerated with predictive models capable of relating the molecular structure of the precursor and processing conditions to the fiber microstructure and properties.

Due to the direct relationship between the fiber microstructure and properties, extensive experimental efforts have been devoted to microstructural characterization [107–110] using a range of techniques including electron microscopy [109] and Xray diffraction (XRD) [111]. Carbon fibers are polycrystalline in nature with long, graphitic sheets arranged in a turbostratic manner. These sheets often exhibit curvature along the fiber axis, with defects such as grain boundaries and voids between individual grains in the fiber. Electron microscopy studies of the cross-sectional microstructure indicate the presence of folded sheets and hairpin-type structures. As expected, the details of the texture, crystallite size, and defects are dictated by the processing conditions and the character of the initial precursor. This body of knowledge has led to several proposed schematics of the internal structure of these fibers [107, 109, 112]. Complementing this experimental work, molecular modeling is beginning to shed light into processing microstructure-property relationships of CFs. Recently, Schatz and co-workers used reactive molecular dynamics (MD) simulations to model the initial steps of the process of carbonization starting from stabilized PAN [113]. The authors were able to identify the mechanism for the formation of polycyclic rings as well as the evolution of gases such as N_2, H_2, and HCN. However, due to the computational intensity of these simulations, only the formation of a few graphitic rings was captured. Other efforts have focused on building characteristic isolated structures identified experimentally, such as defective D-Loops [114], basic structural units (BSUs) [115], or using simplified polycrystalline and multilayer graphite models [116, 117]. While these important studies provide insight into the molecular processes of failure and suggest possible mechanisms, the predicted strength values are one order of magnitude greater than experimental results. In summary, despite significant experimental and modeling work, we lack predictive models to relate the molecular structure of the precursor and processing conditions to the final microstructure and properties. In this chapter, we introduce the MD-based carbon fiber processing simulator, denoted MD-CF, to describe the carbonization and graphitization of stabilized PAN precursors and predict the transverse cross-sectional

microstructure of carbon fiber. The MD-CF simulator enables the prediction of the transverse stiffness of a fiber starting from atomic-scale information, with no input from experiments other that the insight used to develop the model. The processing of PAN-based CFs involves a series of complex steps, starting with the spinning of the precursor, followed by stabilization to transform it into a fiber capable of withstanding the high temperatures required for the key step of carbonization and graphitization. The model introduced here uses a combination of MD and kinetic Monte Carlo (kMC) to predict the graphitization of a structure initially containing stabilized ladder structures.

4.2 The MD-CF model for carbonization/graphitization

4.2.1 Scope of the model in the context of carbon fiber processing

The conversion of PAN to a carbon fiber involves three major steps: (i) spinning, (ii) stabilization, and (iii) carbonization and graphitization. A review of these manufacturing processes can be found in Refs. [103] and [118]. Stabilization involves heating in air at a temperature of 450-550 K, where a series of chemical reactions transform the PAN chains into structures that can withstand the high temperatures required for carbonization without decomposing. While several reactions are believed to take place during stabilization, there is consensus that the result is the conversion of PAN chains into ladder-like structures [119]. In the carbonization stage, the stabilized ladder-like structures are heated to a temperature of 1300-2000 K [105], converting them to a carbon fiber consisting of graphitic sheets. In the graphitization stage, the fibers are heated up to 2700-3000 K to improve sheet alignment and obtain high modulus fibers [118]. This work focuses on the process of carbonization and graphitization.

As mentioned above, a reactive MD study using the ReaxFF force field [113] has provided an atomic picture of the first steps of carbonization and suggested the elimination of gases such as N_2, H_2, NH_3, and HCN, along with cyclization reactions,

leading to formation of five-membered rings and eventually six- and seven-membered rings. These predictions are consistent with experimental observations. However, the use of reactive MD limits the time scales accessible and precludes the study of microstructure evolution. Our motivation to study microstructure evolution and predict the final CF microstructure dictates the use of a coarse-grained description and employing a combined kMC and MD approach. In this endeavor, we ignore the details of the reactions and chemical environments present during carbonization and graphitization. Instead, we describe a generic, averaged version of the individual processes and describe carbonization and graphitization as chemical reactions between carbon atoms in nearby ladder structures to create sp^2 bonds, resulting in graphitic sheets.

4.2.2 Simulating the carbonization/graphitization process

Initial molecular structure

MD-CF starts with a well-relaxed simulation cell containing several coarse-grained ladder structures representing the stabilized fiber. Our coarse-grained approach averages over the details of various chemical reactions responsible for graphitization . Thus, the initial ladder structure, see Figure 4.1(a), consists of a chain of carbon atoms arranged in a hexagonal manner [119]; we focus on the chain structure and not the specific heteroatoms present in the individual rings. The structure consists of two types of carbon atoms: saturated sp^2 carbon atoms bonded to three other carbon atoms, marked as C in Figure 4.1(a), and reactive atoms, marked as C*.

A set of such chains are packed into a simulation cell with periodic boundary conditions in all directions. All chains are infinitely long (with 8 atoms in the periodic unit cell) and perfectly aligned along the Z direction of the simulation cell; see Figs. 4.1(b) and 4.1(c). We stress that assuming that the chains are perfectly aligned is an approximation, and in this first effort to model microstructure evolution, we are interested in predicting the cross-sectional CF microstructure and properties. Before

cross-linking, this initial structure is relaxed, and the details of this procedure are described in Subsection 4.3.1.

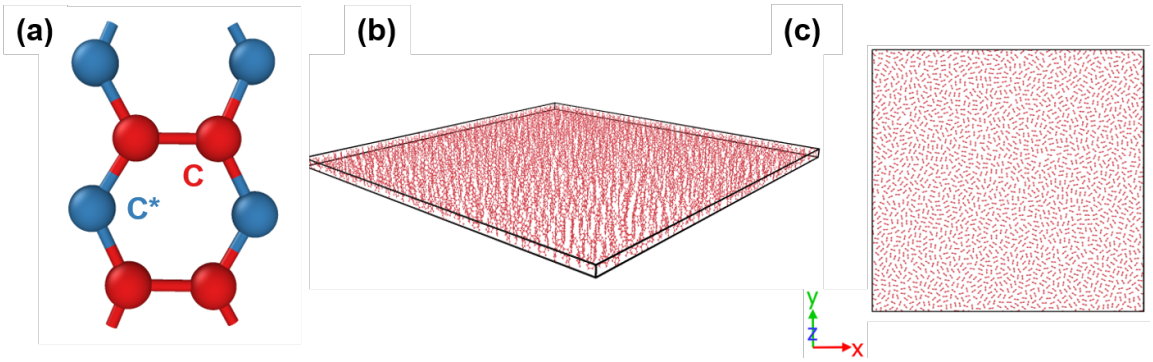

Fig. 4.1. (a) The initial chain configuration, where red atoms indicate the saturated sp^2 carbon atoms and blue atoms indicate the 'reactive atoms', with only 2 bonds. (b) Perspective view of the packed monomers. Note the small out-of-plane thickness of the simulation cell. (c) Top view of a representative relaxed structure

Graphitization model

The graphitization model involves cycles of bond creation followed by relaxation using MD; see Figure 4.3. A similar procedure has been applied previously to study cross-linking in epoxy-amine systems [120, 121] and has been successful in predicting the molecular structure and a wide range of properties such as Young's modulus and the yield strength [122]. The key inputs to the carbonization/graphitization model are the molecular structure of the initial ladder structures and the rate of the reactions (bond formation). In reality, carbonization involves multiple reactions, including the formation of polycyclic chains and the evolution of various gases, and the rates of each of these processes would affect the reactivity of the chains and hence, the final structure. The rate of these reactions depends on the activation energy associated with these processes and a prefactor that depends on the entropy of the reagents and transition state [123]. These rates can be calculated using electronic structure

calculations, as is the custom in kMC simulations. The activation energy (and thus the rate) in this case will depend on the local chemical environment (ignored in our reaction model), the separation distance, and the relative orientation of the molecules.

MD-CF takes into account that only nearby atoms are likely to react imposing a cutoff distance to identify possible reactive atom pairs; see Step 1 in Figure 4.3. We assume that the rate of bond formation is zero for pairs of atoms outside the cutoff. Given the set of carbon atom pairs within the cutoff, the energetics associated with bond formation and, consequently, reaction rates will be affected by the relative orientation and alignment between the two ladder molecules. To estimate the role of misalignment on formation rates, we computed the strain energy of two fully bonded ladder structures as a function of their angle. Figure 4.2 shows the molecular configuration and the energy as a function angle. As expected, the strain energy increases monotonously as a function of angle; a misalignment of 20°results in an increase of approximately 4 kcal/mol per bond formed (0.17 eV) and misalignment of 60°leads to an energy increase of approximately 20 kcal/mol (0.87 eV). Assuming similar increases in the activation energy, we can estimate the reduction in the bond formation rate. An increase in barrier of 0.87 eV would result in a decrease in the reaction rate by 15 orders of magnitude at 300 K and by 2 orders of magnitude at 2500 K, the carbonization temperature.

We thus impose an additional criterion to restrict reactions between pairs of atoms belonging to well-aligned molecules; see Step 2 in Figure 4.3. Consider a possible pair of reactive atoms I and J. Let atom I be bonded to atoms G and H in molecule 1 and atom J be bonded to atoms K and L in molecule 2. The additional constraint is based on a measure of the separation between the planes determined by I-G-H and J-K-L, in terms the relative orientation of the planes and their relative displacement. This is evaluated by calculating the angle between the planes I-J-K and I-K-L, as well as the angle between the planes G-I-J and G-H-J. Thus, given two reactive atoms, there are two improper angles possible and the atoms are eligible for bond creation only if

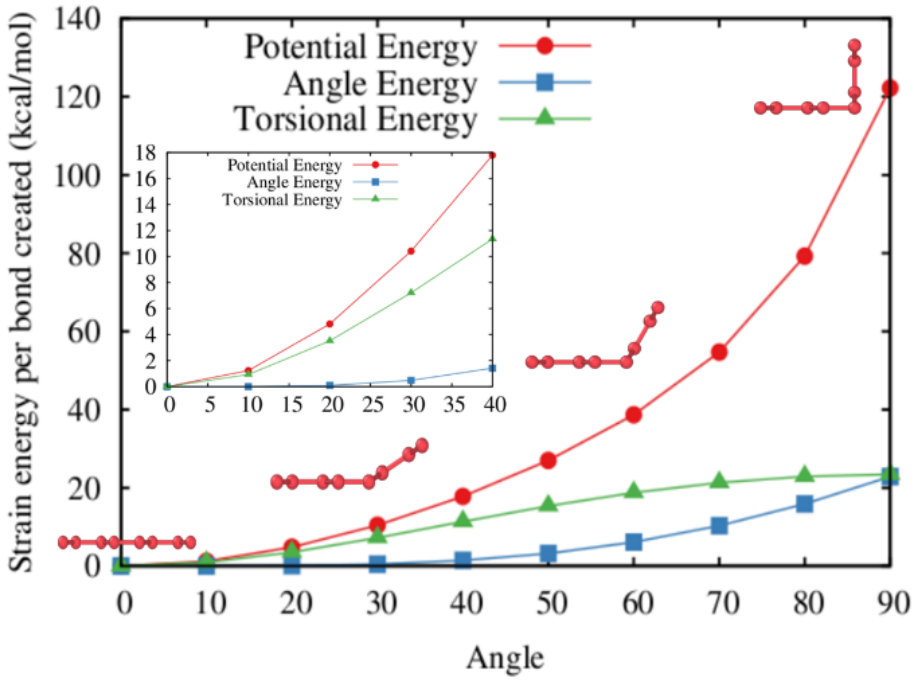

Fig. 4.2. Change in energy as a function of misalignment between two representative ladder structures. The increase in energy with misalignment shows the need to penalize bond formation between poorly oriented ladders

both these angles are within the imposed cutoff. That is, we assume a constant rate within the angle cutoff and zero otherwise.

Two-cutoff model

The dependence of reaction rate on the separation distance merits additional discussion. We find that two distinct processes, involving different separation distances, are important for graphitization. Nearest neighboring ladder chains with no covalent bonds between them are separated by typical van der Waals distances of approximately 3.5 Å; see Figure 4.4(a). Creating the first bond between them results in a structure with significant strain; see Figure 4.4(b). We estimated the strain energy associated with this process by comparing the energies of two chains before and after

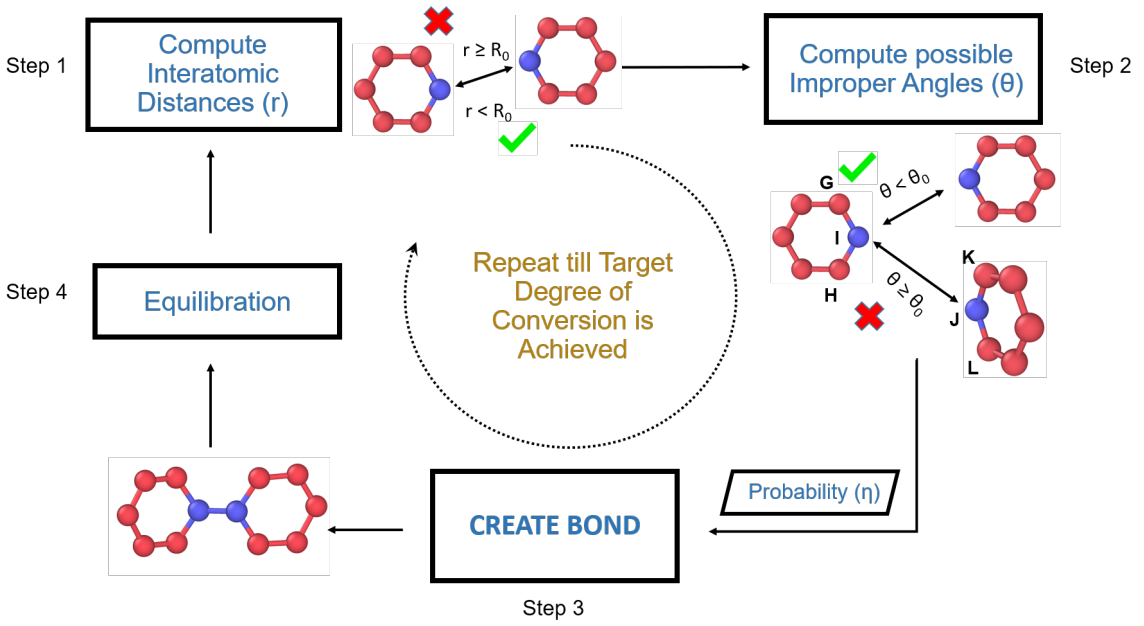

Fig. 4.3. Schematic of the crosslinking algorithm employed, where the blue atoms are the ones considered for bond creation. Here, R_0 indicates the distance cutoff used and 'θ_0' indicates the improper angle cutoff. The probability 'η' can range from 0 to 1.

the creation of the first bond and found a value of over 90 kcal/mol; Figure 4.4(c). However, once the first bond is created between two chains, the separation between nearby reactive carbon atoms is reduced to less than 3 Å, see Figure 4.4(b), and the strain energy associated with the creation of the subsequent bonds decreases significantly; see Figure 4.4(c). Thus, the activation energy associated with the chemical bonding between reactive atoms neighboring a previously created bond will be lower than that associated with creating the first bond between two ladder structures, resulting in a lower reaction rate. We consider this aspect in our method using a two-cutoff approach, as will be discussed next.

In this first effort to model the graphitization of the ladder chain structures, we define a simple set of rules to determine chemical reactions and study how they affect the resulting microstructures and properties. Given all pairs of reactive atoms (C*)

that fall within the capture distance (R_0) and whose ladder chains have an angle mismatch less than a threshold variable (θ_0), we select the possible reactants as pairs of atoms I and J such that J is the closest reactive atom to I and I is the closest reactive atom to J. All these possible reactions are considered to have equal reaction rates. At each bond formation cycle, a pre-determined fraction (η) of these reactive pairs is bonded; these are chosen stochastically. An extension of the current version of MD-CF to include bond creation rates that depend on the separation distance and relative angle between the two ladder structures involved would be straightforward. Following each cycle of bond creation with capture radius R_0 and the subsequent relaxation of the structure, we reduce the capture radius to 2.85 Å, denoted as R_1, and perform three cycles of bond creation with $\eta = 1$ to account for the higher rate of reaction for pairs of atoms with shorter separation distances shown in Fig. 3(b). The two-cutoff model thus has a large primary cutoff (R_0) and a smaller secondary cutoff (R_1).

Simulation time

The simulation time has contributions from both the kMC bond creation events and from the MD relaxation. Within the kMC formalism, given a set of possible events $\{i\}$ with rates $\{k_i\}$, the time is advanced by an amount selected stochastically from a probability density function given by $a_0 * exp(-a_0 t)$, where $a_0 = C * Pk_i$, t is the time, and C denotes the number of reactants molecules for each reactant species [124]. Thus, the time in each of the bond creation cycle involving n reactions is the sum of n stochastic numbers obtained by n samplings of this probability distribution. Each cycle of bond creation is followed by a relaxation using MD for 20 ps under isothermal and isobaric conditions. Note that this is significantly shorter that the kMC time as the relaxation of the structure (following bond creation) occurs relatively fast due to the high stiffness of the graphitic sheets. Reaction rates for the variety of chemical processes involved in carbonization can be performed with ab-initio simulations or

reactive force fields, but it is beyond the scope of this work. Such calculations would be necessary to estimate the kMC time scales in the simulations.

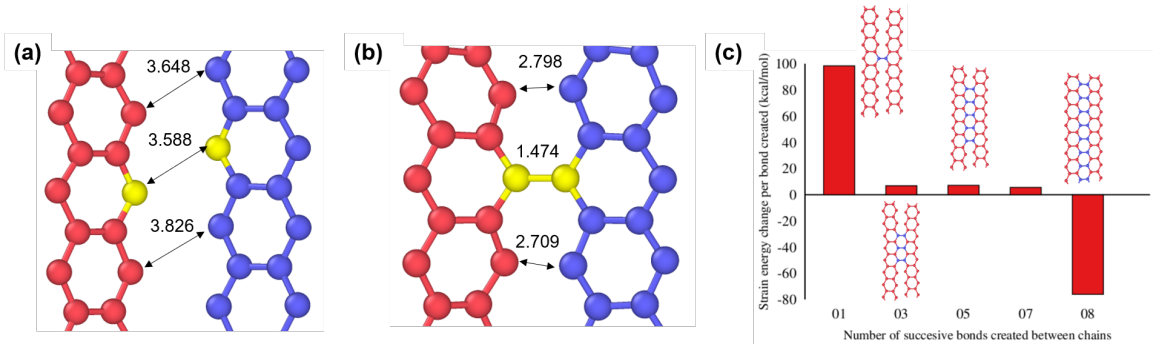

Fig. 4.4. (a) Simulation snapshot showing the change in chain structure after bond creation between two representative atoms (marked yellow). The atoms surrounding the bonds are observed to be close, allowing for a two-cutoff model (numbers indicate distances in Å).

4.3 Simulation Details

4.3.1 Initial structure relaxation

All simulations are performed using the LAMMPS software package [21], and the atomic interactions are described by the DREIDING force field [99]. We use a Lennard Jones form to describe the non-bonded (van der Waals) interactions. Both reactive and saturated atoms are treated as sp^2 carbon atoms, using the default DREIDING parameters. The time step used to integrate the equations is 1 fs unless otherwise specified. The temperature is controlled using the Nose-Hoover thermostat [125,126], with a damping constant of 0.1 ps. Similarly, the pressure is controlled by the Hoover barostat [127,128], with a damping constant of 1 ps.

The simulation begins by creating a ladder-like chain monomer, as shown in Figure 4.1(a). These infinite chains are packed into the simulation cell with random orientations in the XY plane, at a density of 0.5 g/cc, as shown in Figure 4.1(b).

After packing, we relax the structure via a series of steps; the first of which is an energy minimization, using the conjugate gradient method with an energy tolerance of 10^{-6}. The system is then relaxed at constant volume and at a temperature of 300 K (isothermal, isochoric NVT ensemble) for 50 ps. We then equilibrate the system at constant ambient pressure and temperature (isothermal, isobaric NPT ensemble) until the density achieves steady state (with a value of 1.38 g/cc); this step requiring 1.5 ns. We couple cell parameters along the X and Y (in plane) directions in the barostat to retain a square cross section.

The next step is to take the system to the temperature at which carbonization and graphitization will be modeled, which is chosen to be 2500 K to represent experimental conditions [118]. This is done in multiple steps to ensure a well-relaxed structure.We begin by heating the relaxed structure from 300 K to 2500 K under NVT conditions at a rate of 10 K/ps. The system is then equilibrated under NPT conditions with a compressive stress of 0.5 GPa in the transverse directions to ensure good packing of the chains, until the density achieves steady state (1 ns). This stress is then relaxed to 1 atm in 100 ps, and the system is finally relaxed under NPT conditions, at 1 atm for 4 ns, enough to fully equilibrate the system. Throughout the procedure, the barostat maintains a square cross section of the simulation cell. Given that our cross-linking algorithm is stochastic in nature, we generated statistically independent initial samples. This is done by using the relaxed structure from above and continuing to equilibrate it under NPT conditions for 120 ps. From this trajectory, we selected 6 samples, each spaced 20 ps apart.

4.3.2 Carbonization and Graphitization

The samples are then cross-linked using the scheme described in Sec. II. The method was implemented as an extension to the existing LAMMPS fix bond/create command, and the code is available at The code to simulate the crosslinking procedure is available at `https://github.rcac.purdue.edu/StrachanGroup`. The README

file associated with the GitHub repository provides instructions to download and use MD-CF. During the cross-linking, the time step is reduced to 0.25 fs to avoid large atomic displacements after bond creation, as was observed in Ref. [120]. After each bond creation cycle, an energy minimization is performed, using the conjugate gradient method until the change in energy between successive steps relative to the total energy is less than 10^{-6}. During the minimization, the atoms are not allowed to move more than 0.05 Å per step, allowing for a gradual descent in the energy of the system. After the minimization, the system is relaxed for 20 ps under NPT conditions at atmospheric pressure and the graphitization temperature. The NPT relaxation begins with optimized positions and velocities from the end of a minimization segment. The target temperature is achieved in approximately 1000 MD steps (1 fs), which is about 5% of the total relaxation time for each cycle.

4.3.3 Evaluating properties

Before computing mechanical properties, we replicate the carbon fiber structures in the fiber (Z) direction (5 times) in order to improve statistics. The structures are then cooled down to 300 K (under NPT conditions at rate of 10 K/ps) before equilibrating at 300 K for 200 ps, also under NPT conditions, at atmospheric pressure. At this stage, we uncouple the X and Y simulation cell parameters in the barostat to avoid residual strains. To evaluate the transverse moduli, the relaxed structures from above are strained up to 5% in the X and Y directions, at a rate of $5\text{x}10^{-9}\text{s}^{-1}$. A linear fit to the stress-strain curve gives Young's modulus in each direction.

4.3.4 Cross-sectional microstructure: Importance of two-cutoff model and angle control

Before presenting a systematic study of how the parameters in the graphitization model affect the microstructure and properties in Section 4.4, we illustrate MD-CF runs and discuss the importance of using a two-cutoff model to account for the high

reaction rates of reactive atoms at short distances (due to a nearby bond connecting two ladder chains). Similarly, we examine the effect of including a torsional angle constraint on the resulting microstructure. Figure 4.5 shows a series of structures obtained by crosslinking ladder structures at 300 K, employing different choices for the distance and angle cutoffs. Figure 4.5(a) shows a structure resulting from a single cutoff, $R_0 = 5$ Å, and no angle cutoff. The structure in Figure 4.5(b) was obtained with $R_0 = 5$ Å and an angle constraint $\theta_0 = 60°$. Figure 4.5(c) additionally employs the two-cutoff model, with $R_0 = 5$ Å and $R_1 = 2.85$ Å. In all cases, the probability (η) is 0.1.

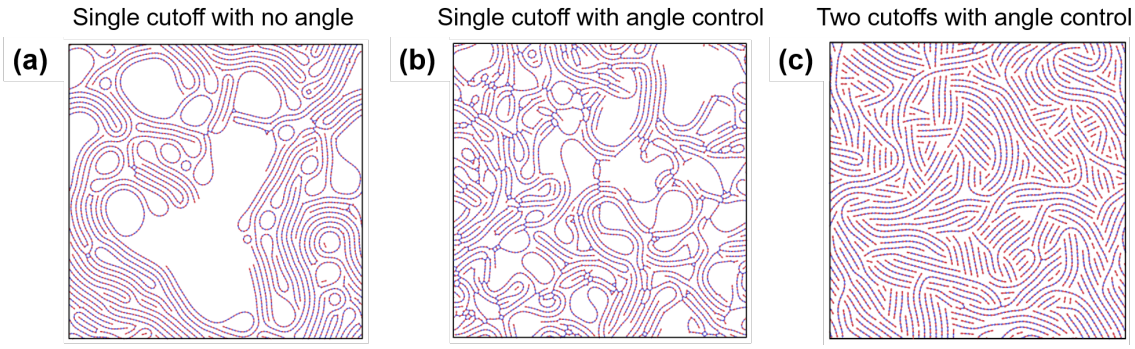

Fig. 4.5. (a) Structure obtained using a single cutoff has unrealistic nanotube structures (b) Implementing an additional angle control results in a branched and disordered structure (c) Using the two cutoff model results in structures similar to experimental PAN based fibers

Figures 4.5(a) and 4.5(b) highlight the importance of angle constraints. Not accounting for the decrease in reaction rates associated with ladder misalignment in Figure 4.5(a) enables bond formation between poorly oriented molecules resulting in a large number of loops (nanotubes) and open structures. The addition of angle control to a single cutoff distance approach restricts reactions to segments that are relatively well aligned, resulting in a more compact structure without nanotubes, as shown in Figure 4.5(b). However, this approach results in a disordered and open structure that does not contain large graphitic sheets. This is due to the fact that the

initial ladder structures have two sets of reactive atoms and using a single cutoff can result in each reactive atom bonding to different chains, leading to branched structures. Finally, Figure 4.5(c) shows that the combination of two-cutoff model with angle control which results in structures with high packing density and microstructures similar to experimental PAN-based fibers; a more quantitative comparison will be presented in Section 4.4.2.

4.4 Microstructure evolution

4.4.1 Evolution during carbonization/graphitization

Figure 4.6 shows the microstructure evolution during the cross-linking process for a representative sample with parameters $R_0 = 5$ Å, $R_1 = 2.85$ Å, $\theta_0 = 60°$, and $\eta = 0.1$. The snapshots show the process through which the ladder chains cross-link and form graphitic sheets that grow in length with conversion. A consistent feature of this process is the volume shrinkage, occurring due to the fact that unsaturated atoms that were previously at van der Waals separations (~3.5 Å) are brought together to ~1.42 Å (the equilibrium sp^2 bond distance).

Figure 4.7(a) shows the evolution of cure degree as a function of MD simulation time (ignoring the time associated with the kMC steps) for various bond creation rates (represented by the probability η). The degree of conversion is defined as the ratio between the number of bonds created and the total possible number of bonds that can be created. A smaller probability η represents a smaller number of reactions per kMC cycle and consequently, a shorter kMC time per cycle, i.e., a slower conversion rate; see Figure 4.7(b). We find that conversion degrees close to 90% can be achieved except for the fastest conversion rates, where the MD simulation time is not long enough to enable full relaxation. The total MD relaxation time varies between 200 ps and 1 ns and is comparable to the time scales used to cross-link polymers using atomistic simulations [120, 129]. At the start of the cross-linking, we see that the number of reactions is directly proportional to the reaction rate; see Figure 4.7(b). As

carbonization/graphitization occurs, the number of reactive carbon atoms decreases and so does the number of reactions. We stress that the total simulation time is the sum of the MD relaxation time and the kMC reaction (bond formation) time. An accurate evaluation of the kMC time would require evaluating a large number of possible chemical reactions and the rate constants associated with each of them, which is beyond the scope of this work.

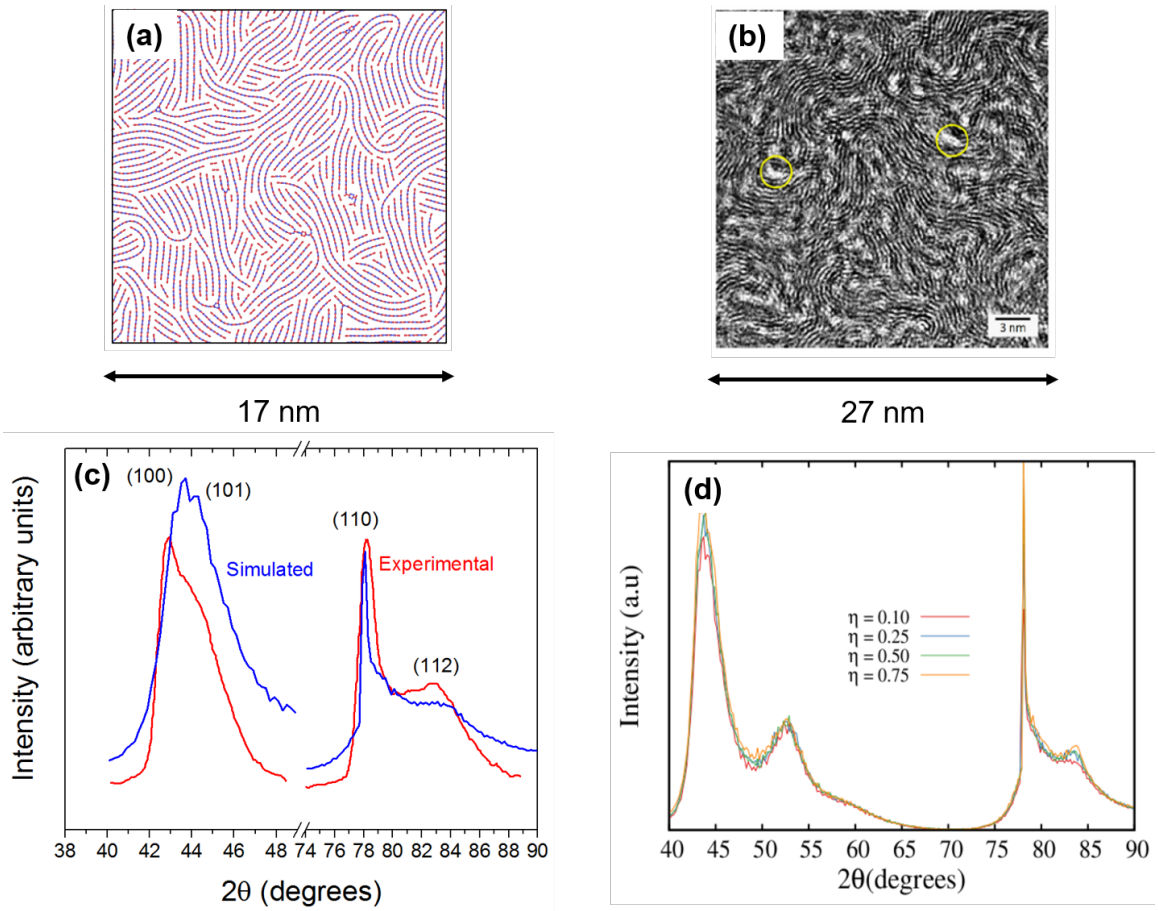

Fig. 4.6. Top view of the microstructure evolution during a sample crosslinking process. The parameters used for this process were: R_0=5Å, R_1=2.85Å, θ_0=60°and η=0.1

Figure 4.8(a) shows the evolution of density with the degree of conversion for various probabilities (η). We find that the density increases with the conversion

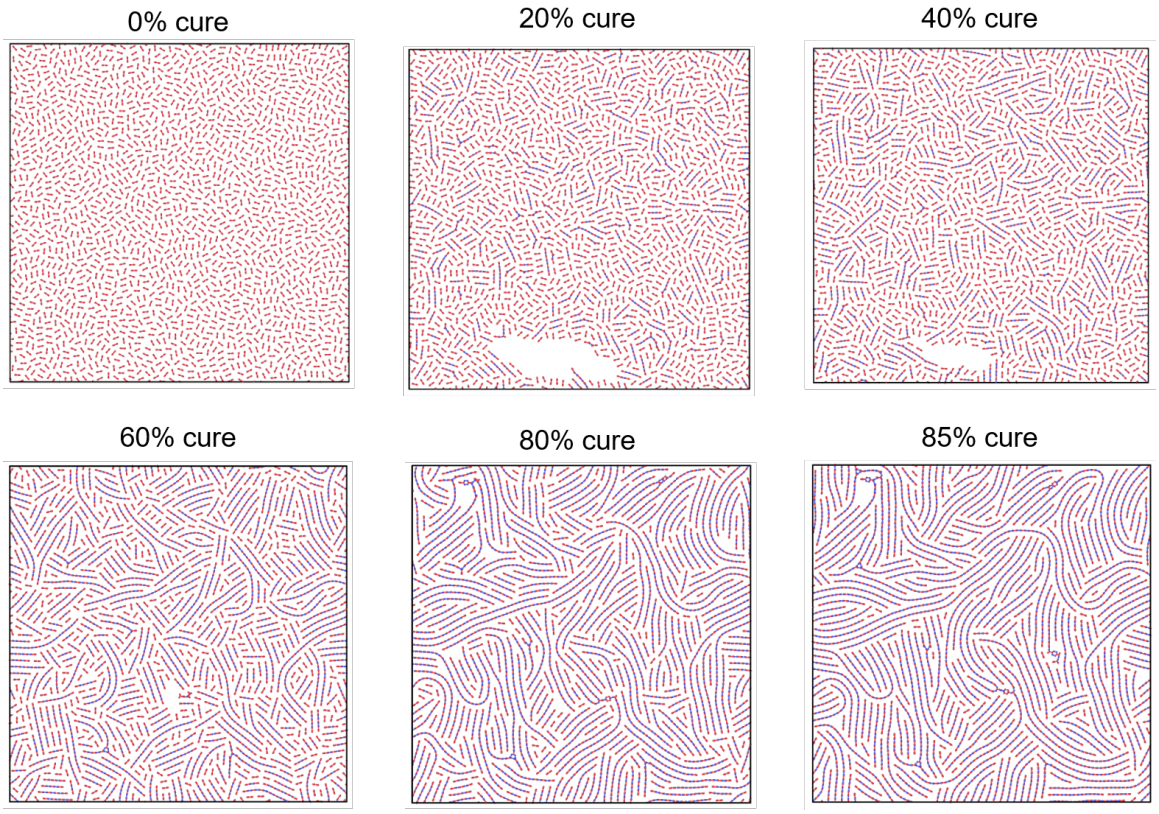

Fig. 4.7. (a) Time evolution of the degree of conversion for various probabilities (b) Time evolution of the number of reactions, for various probabilities, shown for the first 200 ps (MD time)

degree, again indicative of the fact that as more bonds are created, a greater number of unsaturated atoms move from a van der Waals separation to the equilibrium sp^2 bond distance. However, this trend is observed only until ~60% conversion, with the higher bond creation rates showing a subsequent drop in density. Achieving high conversion degrees at fast rates results in significant strain in the graphitic sheets that are unable to relax and pack efficiently, resulting in excluded volumes that remain as voids throughout the rest of the simulation. The shaded area in Figure 4.8(a) represents the typical range of experimental values for the density of PAN-based carbon fibers. Our predicted structures overestimate the density by approximately 10%. We attribute this observation to the fact that we use infinitely periodic and

perfectly aligned chains. This results in an unrealistically high degree of ordering in the fiber direction, resulting in a high degree of packing.

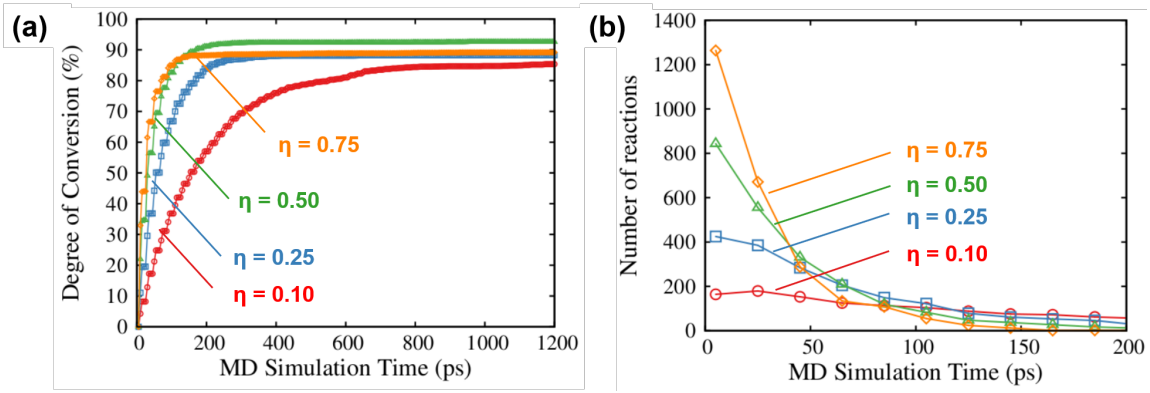

Fig. 4.8. (a) Evolution of density with degree of conversion (b) Evolution of the simulated XRD pattern with degree of conversion. The peaks at 26°, 42°and 78°correspond to the (002), (100) and (110) planes, respectively

Figure 4.8(b) shows the evolution of simulated XRD patterns with time. The predicted structures are compared to experimental data in Section 4.4.2; here we discuss its evolution during processing.We observe that the (002) peak, corresponding to the van der Waals separation in graphite, decreases in width, denoting the evolution of a graphitic structure as the simulation progresses. Also, the broad peak initially present between 40°and 60°transforms into a peak at ~43.5°, corresponding to the (100) plane and a second, weaker, peak at ~54°, denoting the (004) plane.

4.4.2 Microstructure validation

Figures 4.9(a) and 4.9(b) compare one of our predicted microstructures (R_0 = 5 Å, R_1 = 2.85 Å, θ_0 = 60°, and η = 0.1) with an experimental high resolution transmission electron microscope (HRTEM) image corresponding to high-strength and high-modulus gel-spun PAN copolymer fibers [5]. We find that the simulated

structure contains key microstructural features such as hairpins and curved graphitic sheets observed experimentally.

To further validate our structures, we use the LAMMPS software package to simulate a wide angle X-ray diffraction (WAXD) pattern, as shown in Figs. 4.9(c) and 4.9(d). We use the procedure detailed in Coleman et al. [130]. In this procedure, the simulation cell is divided into a grid in reciprocal space, the grid spacing determined either manually or using the cell lengths. At each point of this grid, the structure factor is evaluated, using the following equation

$$F(\mathbf{k}) = \sum_{j=1}^{N} f_j(\theta) exp(2\pi \mathbf{k} \bullet \mathbf{r_j}) \tag{4.1}$$

where $f_j(\theta)$ denotes the atomic form factors and r_j denotes the atomic coordinates. The structure factor, which gives the scattering due to the atom positions in all the unit cells, can then be used to calculate peak intensity, using the following equation,

$$I = LP(\theta)\frac{|F|^2}{N} \tag{4.2}$$

where $LP(\theta)$ denotes the polarization factor, used to account for unpolarized beams as well as finite crystal sizes.

We compare our predictions with an experimental measurement on high modulus, low strength, PAN-based, GY-70 fibers manufactured by BASF [111]. In Figure 4.9(c), we use the indexing notation followed in Ref. [111], where the (100) planes are stacked in the zigzag direction of the basal plane of the graphite sheet, while the (110) plane are stacked in the armchair direction. As seen in Figure 4.1(a), the initial ladder structure is oriented such that successive reactions will extend the graphitic sheet in the armchair direction, adding (110) planes. The absolute intensity depends on the sample size, and we thus scale intensities to match the (110) peaks between theory and experiments. The key features are the width of the (110) peak and the significant broadening of the (112) peak which the simulations capture. The width of the (110) peak is slightly underestimated in the predicted structure; this implies

longer graphitic crystallites in the simulated structure than the specific carbon fiber characterized in the experiment. This is likely associated with a high degree of order in our structures in the direction of the fiber axis.

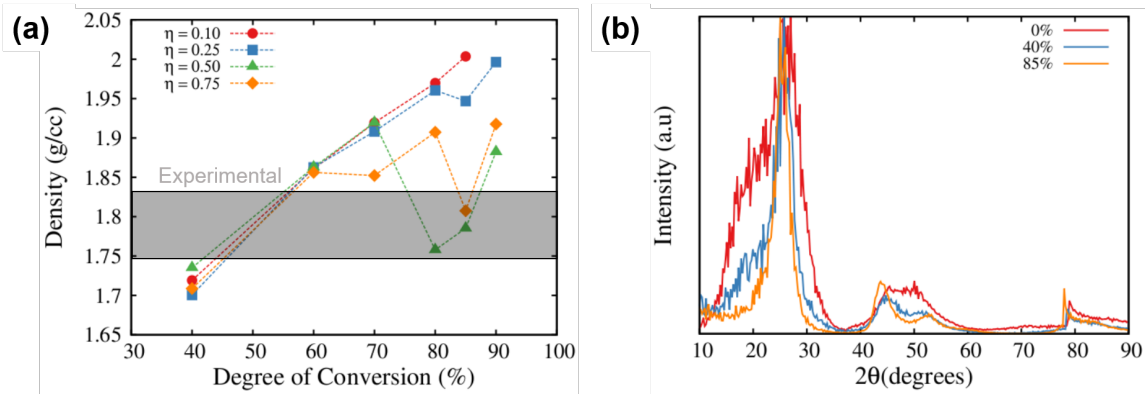

Fig. 4.9. (a) Top view of a representative simulated structure (b) An HRTEM image of a carbon fiber cross section, reprinted from Ref. [106], with permission from Elsevier (c) Simulated powder XRD pattern from the predicted structures (blue) and experimentally observed XRD pattern (red) taken from Ref. [111]. The (101) and (112) peaks indicate extent of 3D order in the carbon fiber, an aspect that the extended 3D model will attempt to capture. (d) XRD patterns averaged over all six samples, for various reaction rates

An important microstructural characteristic of carbonfibers is the average size of the graphitic crystallites; and these values can be obtained experimentally from microscopy and diffraction studies. Three values are often used to describe microstructural features size in carbon fibers: L_c is the average size in the direction normal to the graphene sheets and L_a is the characteristic size along the graphitic planes, which is subdivided into $L_{a_{par}}$, along the fiber axis, and $L_{a_{perp}}$, along the cross-sectional directions. Inspection of Figure 4.9(a) indicates that La perp ranges from 3 to 6 nm in our structures as well as in the gelspun fiber in Figure 4.9(b). Characteristic sizes can also be obtained from the XRD diffraction patterns. From the simulated peak width associated with 110 planes, we obtain an equivalent crystallite size of ~16 nm using the Scherrer equation. Typical PAN fiber crystallite sizes along the cross-sectional

direction, $L_{a_{perp}}$, range from 5 to 8 nm [102], although larger values have been reported. The experimental value is in agreement with the snapshot in Figure 4.9(a), and we attribute the overestimation of the crystallite size from the simulated XRD patterns to the fact that our fibers are perfectly aligned and infinitely long along the fiber axis. In reality, longer chains would deviate from the fiber axis by about 15°-25° [102], reducing the strong preferred orientation and decreasing the peak intensity, along with increasing the probability of random cross-linking between chains, decreasing the probability of the formation of long graphitic sheets.We further note that comparison between simulated and experiments should be performed with care. We simulate a powder X-ray diffraction where each peak has contributions from the entire family of the corresponding planes. In our case, this means that the 110 peak also has contributions from those planes aligned along the fiber axis are highly ordered. In contrast, the experimental pattern was obtained using a four-circle X-ray diffractometer, where the fiber sample is rotated at different angles to obtain specific diffraction peaks. For instance, the pattern shown in Figure 4.9(c) was obtained when the sample was tilted at 70°and the width of the (110) peak corresponds to the crystallite size in a specific direction. The contribution from other planes of the 110 family will appear in peaks from different scans [say, the equatorial (0°) or meridonial (90°) scans], and the effective crystallite size will thus be lower than observed in the simulated pattern.

The peak at ~43.5°is a mixture of the (100) and (101) peaks and may be attributed to the replication process employed in this study, which assumes greater order in three dimensions than might be revealed by using a full 3D model, beginning with an initial structure consisting of longer chains. We do not report the (002) peak as the 2D nature of the model ensures that the graphite planes in the fiber axis direction will be spaced at the equilibrium van der Waals separation. The (002) interplanar spacing will be of prime focus in the full 3D model as it determines Young's modulus along the fiber axis. Figure 4.9(d) shows that the XRD pattern varies little for various bond creation rates, indicating that the structure of the graphitic sheets, characterized by

the bonded C atoms in the basal plane and the van der Waals separation between planes, remains similar, with variations occurring only in density, as will be shown in Section 4.5.

4.5 Predicted properties

Figure 4.11 shows the average transverse modulus as a function of density for fully converted MD-CF fibers created with various bond creation probabilities (η), capture radius (R_0), and angle control threshold (θ_0). The predicted moduli range from 1.5 to 4.5 GPa. Before discussing how the model parameters affect the predictions, we discuss the obtained stiffness values. Figure 4.10 contains several stress strain curves (in tension and compression) up to a strain of 5%. The extracted modulus is sensitive to the strain range used in the linear fit. The stress-strain curves show slight softening after approximately 2.5% strain, and a linear fit up to 2.5% strain changes the range of the predicted moduli from 1.5-4.5 GPa to 2-5.5 GPa. In compression, the predicted moduli range from 2.5 to 5.5 GPa, similar to the range observed in tension.

It is useful to compare our predictions with the transverse modulus of graphite, which is significantly stiffer, approximately 36 GPa [131]. The MD trajectories of the uniaxial tension of converted fibers show that the prominent mode of CF deformation is via sliding of the chains across each other, whereas the modulus of graphite is a measure of the stiffness of the van der Waals attraction between the graphite layers. Given that the shear modulus for ideal graphite is ~4 GPa [132], it is clear that chain sliding is a low activation barrier process compared to increasing the van der Waals separation and can thus occur at lower stresses, explaining the order-of-magnitude difference in the moduli even after correcting for the different densities.

The transverse modulus of carbon fibers has been experimentally estimated using single fiber compression tests resulting in values in the 6-10 GPa range for high-strength fibers [133] and 1-3 GPa for high modulus PAN and pitch-based fibers [134]. In these tests, the modulus is obtained by fitting the experimental data to an ana-

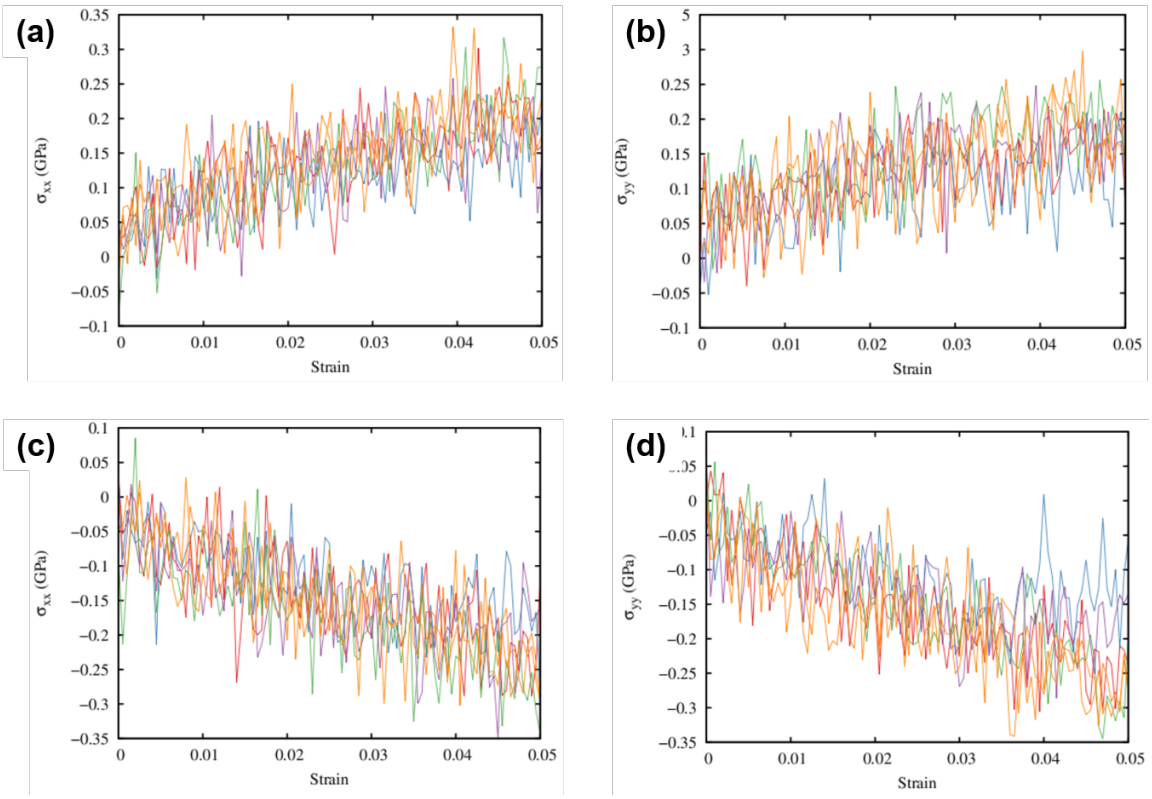

Fig. 4.10. Stress strain curves (in tension) for same representative MD-CF microstructures in the (a) X direction and (b) Y direction. (c) and (d) Stress strain curves in compression. The parameters used for these set of microstructures were: $R_0 = 5$Å, $R_1 = 2.85$ Å, $\theta_0 = 60°$and η=0.1

lytical equation relating the change in the fiber diameter to the applied load using anisotropic elasticity [135]. Our predicted average lies in the range of high modulus pitch- and PAN-based fibers and is lower than the value for high-strength fibers. This agreement is expected given the perfect axial order of our structures. Misoriented crystallites, sp^3 bonds between graphitic sheets and amorphous regions in the experimental fibers, particularly highstrength ones, will result in higher transverse moduli. We note that nanoindentation experiments [136] report higher values of 9-15 GPa for selected fibers from Toray and Mitsubishi. However, in the case of the nanoindentation measurements, the modulus is extracted from the load displacement curve by using the Oliver-Pharr method [137], which assumes that the material is

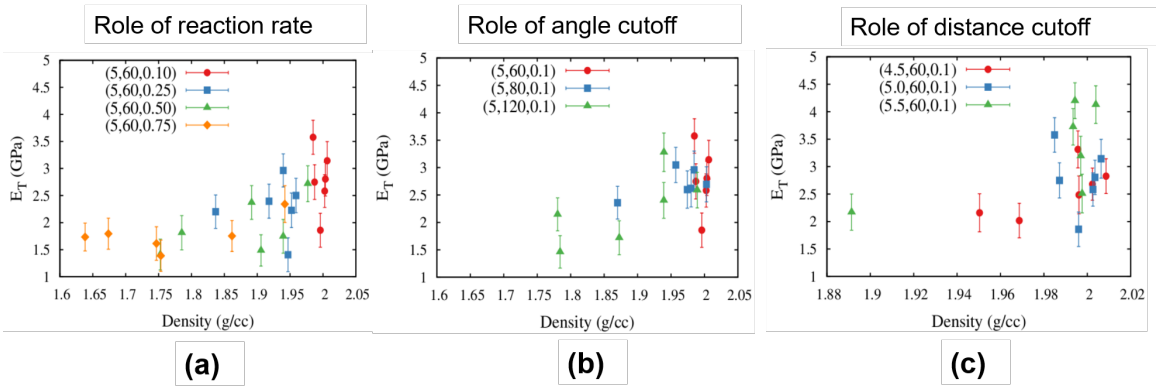

Fig. 4.11. Transverse moduli as a function of density for the predicted microstructures with conversion degree of 85%. For each panel, the legend is a set of three numbers (l,m,n) where 'l' represents the distance cutoff (in Å), 'm' represents the angle cutoff (in degrees), and 'n' represents the probability. For each set of parameters, the individual points represent the 6 samples, all at 85% conversion

isotropic. The anisotropic nature of fibers (high stiffness in the longitudinal direction) indicates that the extracted values (9-15 GPa) represent an upper bound to the transverse stiffness [138]. Figure 4.12 shows that the transverse modulus evaluated does not depend on the simulation cell.

We now discuss the effect of model parameters on the predicted properties. Figure 4.11(a) shows that increasing conversion rate results in a decrease in transverse modulus and density. This is because reducing the effective MD simulation time precludes the graphitic sheets to fully relax, resulting in significant internal strain and the formation of voids. Similarly, employing a loose angle constraint, see Figure 4.11(b), results in folded sheets and nanotube-like structures that enclose a volume that cannot be occupied by other chains, resulting in poor chain packing and lower densities and, consequently, lower modulus. Varying capture radius (or distance cutoff) from 4.5 to 5.5 Å does not result in significant changes in density or stiffness; see Figure 4.11(c). Interestingly, comparing predictions across all parameter sweeps, we observe that structures with nearly identical densities can have moduli varying

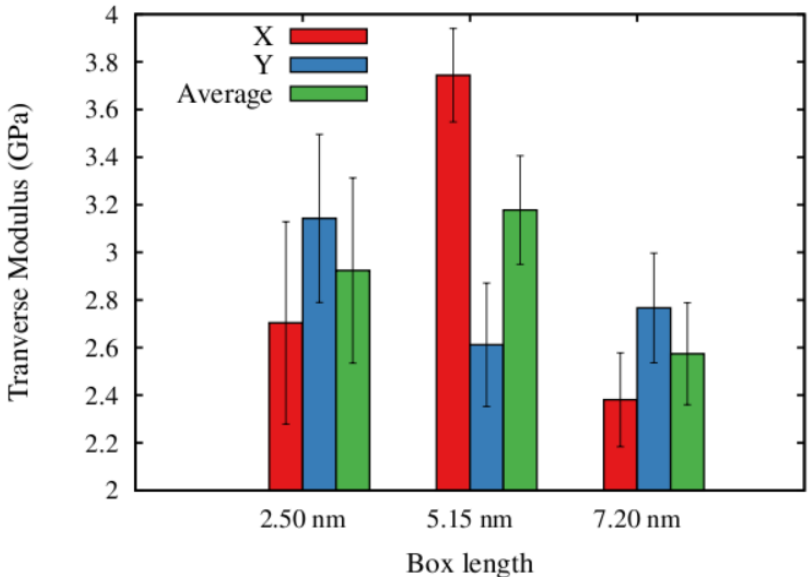

Fig. 4.12. Transverse moduli in the X and Y directions (and the average of the two) evaluated for different box lengths (in the Z direction)

by a factor of two. This is indicative of the large fluctuations expected in relatively small systems with complex microstructures. Further analysis should be performed to identify particular arrangements and lengths of the chains that result in easier sliding and thus, lower modulus.

4.6 Conclusions

We introduced MD-CF, a molecular-level model to describe a key processing step of carbon fibers and predict its molecular structure. The model considers the carbonization and graphitization of coarse-grain ladder chain structures and results in predicted microstructures containing key structural features observed in experiments. The main inputs to our model are as follows: (i) the initial molecular structure (determined by the arrangement of the ladder chains that represent the molecular structure of the stabilized carbon fiber) and (ii) the parameters used to determine the rates of the bond formation processes that convert ladder chains to graphitic sheets. Regard-

ing the first item, in this initial effort, we assume the chains to be perfectly aligned and infinitely periodic along the chain axis. This results in smaller simulation cells and enablesus to explore several aspects of the model effectively. The limitation is that we only predict the transverse microstructure and properties. Ongoing work is exploring larger simulation cells where long, finite chains are packed and cross-linked. In order to determine rates for chemical reactions, we use a simple but physically based approach based on the separation distance between carbon atoms and the relative angle between the ladder chains.

MD-CF predicts key microstructural features known to exist in carbon fibers, and the predicted diffraction patters are in good agreement with experiments. The predicted densities are approximately 10% higher than experimental values, again indicating good agreement. We attribute the overestimation to the high degree of order in the fiber axis we impose in this first effort. Future work will focus on an initial structure with long ladder chains and explore how the initial molecular structure affects the final microstructure and properties of the fibers. The predicted transverse Young's modulus is in the range of high modulus pitch- and PAN-based fibers but is slightly lower than the experimental values of high-strength PAN-based fibers. This is also explained by the perfect nature of our models; as sp^3 bonds, misaligned crystallites and amorphous regions are expected to significantly increase stiffness. The model presented here is a key first step towards predictive computational tools for carbon fibers. Accurate atomic models for microstructures are key for property predictions, not just elastic constants but also properties such as ultimate strength. Such predictive tools have the potential to contribute to the design of new carbon fibers with tailored properties.

4.7 Attempts at a three dimensional model and future work directions

Following the success of the MD-CF approach, we attempted to extend the model to generate three dimensional, realistic, carbon fiber microstructures and while we did

see some promising results, out predicted Young's moduli and strengths were higher than experimentally observed, indicating that we had not fully captured the defects present in the fibers, see Figure 4.13. However, our efforts began to bridge the gap between the current state of the art and the ideal limits of graphite. Other work following up on ours has had greater success, with some molecular dynamics studies able to capture a microstructure similar to observed in experiments, and with similar mechanical properties [139, 140].

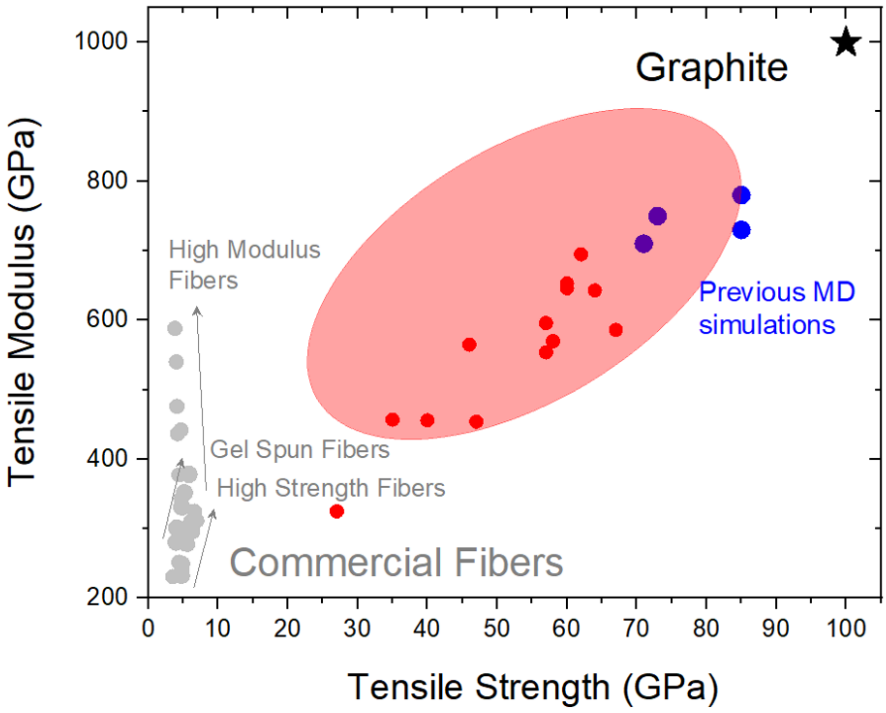

Fig. 4.13. A summary of the carbon fiber design space. Red points indicate the simulations using the MD-CF algorithms

Future work in this area could use the MD-CF approach to model fiber-matrix interfaces in composites, simulating the complex microstructures that govern the strength of the composites. Additional investigation of the chemical pathways governing the carbonization could lead to more informed versions of MD-CF that can simulate complex microstructures.

In this chapter, we explicitly used kinetic Monte Carlo approaches to access timescales beyond conventional MD simulations. However, when such techniques are not immediately applicable or are difficult to implement, a simpler way to access greater timescales (and length scales) is to leverage increasing computational resources. We will now see one approach to achieve cross-platform, portable performance in molecular dynamics simulations with state-of-the-art interatomic models by using novel libraries, algorithms and data structures.

5. GREATER TIMESCALES AND LENGTHSCALES VIA PERFORMANCE PORTABLE IMPLEMENTATIONS

In chapter 4 we saw an example of extending molecular dynamics (MD) simulations to access greater timescales by explicitly incorporating kinetic Monte Carlo methods. We now turn to another avenue to increase the lengthscales and timescales accessible via MD. This method relies on re-implementing MD algorithms to leverage increasingly common hybrid CPU-GPU architectures.

This chapter is organized as follows: We first describe the need for performance portable implementations for MD algorithms in Section 5.1. We then describe neural network and related, relatively new interatomic models in Section 5.2. In Section 5.3 we examine the neural network based interatomic kernels in detail, briefly describing the CabanaMD proxy application and its use of the Kokkos programming model and the Cabana particle toolkit to enable co-design for MD codes, as well as discussing the main ways in which the code was modified from a current LAMMPS implementation. On-node strong and weak scaling with the CabanaMD proxy application is demonstrated on both multi-core CPU and GPU hardware for various materials, comparing our code with the previous CPU-only implementation in Section 5.4. In addition, we explore parallelism and data layout improvements using unique Kokkos and Cabana library features and run simulations of up to 20 million (M) atoms on a single CPU node and up to 4M atoms on a single GPU on state of the art pre-exascale architectures in Section 5.5. Finally, we conclude and discuss impact and continuing work in Section 5.6. Our work highlights a path forward for data-driven and machine-learned interatomic models in classical MD at the exascale.

The work in this chapter has been published in the arXiv and can be found as: Saaketh Desai, Sam Reeve, Jim Belak, “Implementing a neural network in-

teratomic model with performance portability for emerging exascale architectures", arXiv preprint arXiv:2002.00054 (2020).

5.1 Introduction

Molecular dynamics (MD) simulations have been used for many years [11, 12] to study metals and alloys, polymers, 2D materials, proteins, and more. Advances in computing power have allowed MD simulations of billions of atoms in the last decade [36], elucidating complex processes such as solidification [141, 142] and plasticity [16], while also realizing microsecond timescales [37]. Such advances have been possible largely due to the advent of GPUs, necessitating the re-implementation of MD algorithms to extract maximal performance on both GPUs and increasingly hierarchical multi-core CPUs. These re-implementations have resulted in an increasing variety of parallel MD codes [143], particularly GPU-accelerated ones [37, 144–149], with calls for performance portability measures [150]. Despite this progress, many material science and physics problems remain unresolved to this date or are only beginning to be understood at the atomic scale. Important advances are being made by leveraging the improved accuracy of machine learned and data driven potentials, notably including neural network potentials (NNP) [50] to discover atomistic mechanisms of recrystallization in phase-change materials such as GeTe [151] and studying the diffusion of Li-ions in novel solid-state amorphous battery materials [152]; however, accessible system sizes are still relatively small compared to traditional models. With exascale computing on the horizon [153], modeling and simulations tools can be increasingly used to solve 'grand challenges' in material science, such as the problems stated above, or other high-impact problems such as understanding high-temperature superconductivity using quantum mechanical (QM) simulations, simulating energetic materials under extreme conditions at an atomistic level, and enabling additive manufacturing (AM) with multi-scale modeling of the processes involved.

Crucially, solving these 'grand challenges' requires efficient leverage of computing resources via exascale-ready simulation tools. The Exascale Computing Project (ECP) is focused on achieving this by delivering an exascale computing ecosystem via co-design between hardware architectures, software stack development, and application development [154, 155]. This ensures first that application requirements are met by hardware and software innovations and also that applications can effectively use cutting-edge hardware and software stacks. An important arm of the ECP initiative is within co-design centers across all three areas based on computational "motifs": operations on particles, linear algebra, structured/unstructured grids, graph operations, etc. These co-design centers have two main outputs to facilitate co-design and interoperability: libraries and proxy applications. Software libraries can be used directly within production applications, while proxy apps are instead designed as a testbed for the main features of a complex physics application, aiming to understand and optimize computation, memory access, network usage, etc., separate from the full application code.

The Co-design center for Particle Applications (CoPA) (`https://github.com/ECP-CoPA`) addresses motifs within all particle-based simulations, across many application areas, ranging from atoms to galaxies, via the development of software libraries and packages targeted towards accelerating the key motifs for each application. CoPA is developing the PROGRESS and BML libraries for linear algebra in quantum MD for chemistry and materials science [156], as well as the Cabana library for particle operations [157] within N-body gravity for cosmology, particle-in-cell methods for plasma physics, and classical MD for materials science, the focus of this work. Cabana was designed to leverage on-node parallelism from the Kokkos library [158] for cross-platform performance, adding particle specific algorithms and MPI communication for particles.

CabanaMD is a classical MD proxy app that uses the Cabana and Kokkos libraries and has shown performance portability across various hardware for a Lennard-Jones (LJ) interatomic model. At nearly 100 years old, the LJ interatomic model is still

widely used in qualitative studies and for van der Waals interactions in quantitative predictions, the simplicity of the kernel also proving valuable for assessing one extreme of MD performance. For most materials however, more complex interatomic models are required. Machine learned, data driven, and neural network models aim to offer near-quantum level accuracy in predicting energies and forces for a wide range of systems, while retaining the linear scaling offered by classical MD. Neural network interatomic models [50, 159] in particular offer unique and complex computational kernels, enabling many avenues for improvement of the computational cost, as well as expansion of features within both CabanaMD and Cabana. These neural network potentials (NNP) have notably recently gained traction by accurately simulating phase transitions in disparate materials [151, 160, 161]. Neural network models rely on the flexible functional form of a neural network, with thousands of tunable parameters, to approximate complex potential energy surfaces. In this work, we re-implement an NNP in CabanaMD, demonstrating use of the Kokkos and Cabana libraries to achieve performance portability, including a GPU implementation of a neural network interatomic model.

5.2 Neural network interatomic models

Machine learned and data-driven interatomic models offer the opportunity to perform nearly-quantum accurate MD simulations while retaining the scaling of a standard interatomic model. These next-generation models are a departure from traditional, empirically motivated, parametric models, e.g. embedded atom method [25], where the number of parameters and functions are fixed. Non-parametric interatomic models instead use descriptors, which characterize the atomic environment of each atom, with a separate regression model. These non-parametric descriptors are more mathematically-motivated, although still physically derived and hand-tuned in implementation, and include symmetry functions [50], bispectrum coefficients [51], and others [162–164]. Notably, all versions of non-parametric interatomic models are sys-

tematically improvable by adding additional basis functions or more nodes in the neural network [163], unlike standard, fixed-parameter models. The physics of the system, including any translational, rotational, and permutational invariances are included in the descriptors. These descriptors can also be learned with machine learning techniques such as a separate neural network, even a complex convolutional/residual networks to learn the environment of each atom, as opposed to hand-tuning a set of descriptors, thus requiring even less human intervention [53, 165, 166]. Assessing and improving the performance of learned-descriptor-based models is fundamentally different from descriptor-based models and will not be a part of this study.

Across non-parametric models, the descriptors and mapping techniques that relates the structural descriptors (learned or prescribed) to the observed quantities (e.g. energies, forces, and stresses from a QM calculation) varies widely. For instance, SNAP uses a linear regression mapping of bispectrum coefficient descriptors [51], while symmetry function descriptor based (Behler-style) NNP uses a simple feedforward neural network as the mapping [50]. For the learned-descriptor based NNPs, additional neural network(s) are used as mappings [53, 165, 166]. We note that these models are still under active development, with many iterations for each descriptor, mapping, and combination. Having chosen a descriptor and a mapping, the interatomic model parameters are typically obtained through a least-squares fitting procedure, converging parameters to a local optimum. Improvements on this strategy include combining local optimization techniques with global optimization for hyperparameters and using techniques such as genetic algorithms [167].

While not the focus of this work, it is important to note that the accuracy and range of environments represented by non-parametric models depends strongly on the QM data used to train the model, as with their parametric counterparts. See Ref. [55, 168, 169] for recent work on parametrizing machine learned interatomic models.

In this work, we focus on the performance of an NNP that uses symmetry functions as descriptors, which are Gaussian functions of neighbor distances with radial or angular character for each of the elements involved. The flexibility and accuracy

of neural network models is achieved at a high computational cost relative to empirical interatomic models, with comparative performance for many non-parametric models recently benchmarked [57]. Recent work has suggested improvements to NNP descriptors, aiming to reduce the number of symmetry functions required to describe multi-element neighborhoods [170], and thus improve performance. We aim to instead improve performance by implementing a cross-platform, thread-scalable version of a neural network model in the CabanaMD proxy app.

5.3 Model implementation on emerging exascale architectures

5.3.1 Neural network interatomic kernels

To understand the scope for improvements in the Behler-style NNP, we look at the computational kernel in detail. Figure 5.1(a) shows a traditional Lennard-Jones (LJ) kernel that shows the loop over each atom i and each of its neighbors j to compute the force on each atom. Note that the energy compute step is shown only for comparison and is not strictly required to compute the trajectory for an LJ kernel (and most often computed only occasionally). Figure 5.1(b) contrasts the simple LJ kernel with the computationally complex NNP kernel, broken down into 3 steps. Step 1 consists of a loop over each atom i and each of its neighbors j, but also has an added loop over each descriptor k to compute all radial atomic environment descriptors. Not shown is a similar angular computation with one extra loop over neighbors of neighbors. Step 2 is unique to the NNP kernel and involves matrix multiplications that forward-propagate the computed symmetry functions (descriptors) through the neural network to calculate the per-atom energy. Thus, the energy calculation is necessary for the dynamics of the system, unlike a standard model. Step 3 is again similar to the LJ kernel in that it consists of a loop over atom i and each of its neighbors j to compute forces, but again involves an extra loop over descriptor k, and contains both the gradient of the neural network (w.r.t descriptors) and the gradient of symmetry functions (w.r.t coordinate). Note that while the neural network is not particularly

large or deep (~30 x 2), each atom has its own neural network, resulting in many small sets of matrix multiplications (see Figure 5.1(c)). The use of machine learning libraries such as Tensorflow/Pytorch could be explored to gain additional speedup, but is left as future work, both because it is a relatively minor portion of the overall computation, as seen later, and because these libraries are focused primarily on large, deep networks.

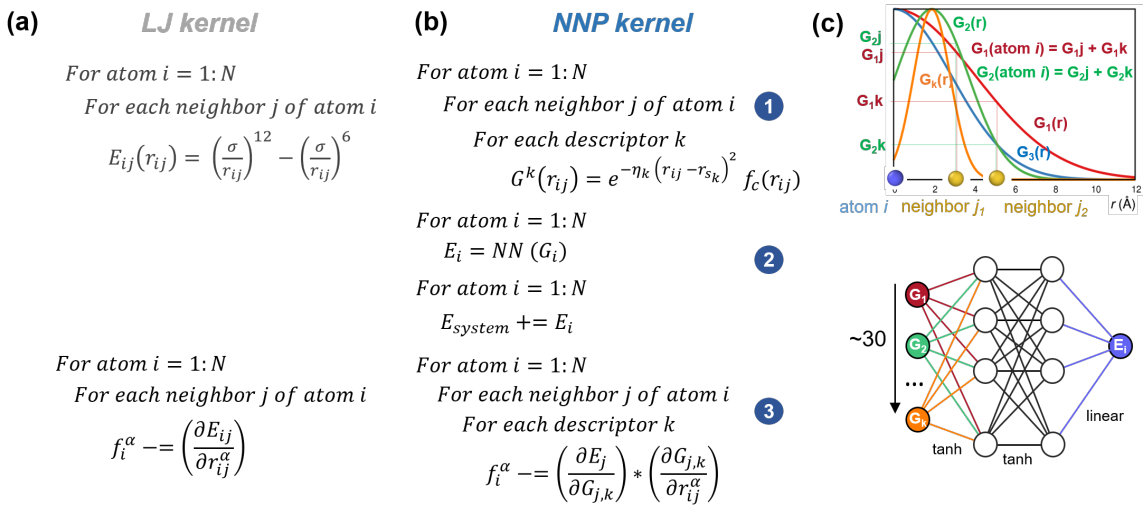

Fig. 5.1. (a) Traditional Lennard-Jones model showing the main loops over atoms and neighbors to compute energy and forces, (b) neural network interatomic model broken down into 3 steps: descriptors, neural network, and force calculations, and (c) representative NNP descriptors and schematic atomic neural network used to predict energy and forces.

5.3.2 CabanaMD proxy application

CabanaMD (`https://github.com/ECP-CoPA/CabanaMD`), our testbed for reimplementing the neural network interatomic model, contains representative units from production MD codes, with flexibility for parallelism and data layout improvements. CabanaMD inherits the modular design and Kokkos implementations of the ExaMiniMD proxy app (`https://github.com/ECP-CoPA/ExaMiniMD`) [158]. CabanaMD

primarily uses the Cabana library for flexibility of particle algorithms and data layout, with underlying on-node performance portability from Kokkos. The complete software stack is shown in Figure 5.2(a), highlighting that Cabana is a domain specific, direct extension of Kokkos. For parallelism across hardware in CabanaMD, the **Kokkos::parallel_for** is used for threading across atoms and **Cabana::neighbor_parallel_for** for threading over both atoms and neighbors (where either level can be serial or threaded), built on the hierarchical parallelism of Kokkos. Portable memory for particle data uses array-of-structs-of-arrays (**Cabana::AoSoA**), built directly on **Kokkos::Views**, with an additional compile-time array dimension that can map to a SIMD/SIMT instruction length which we refer to as the vector length, see Figure 5.2(b)). The AoSoA is intermediate between array-of-structs (AoS) and structs-of-arrays (SoA), more common data layouts which require a complete tradeoff between data locality and sequential data access. The AoSoA provides more complete layout flexibility and benefits of both AoS and SoA for intermediate vector lengths. **Cabana::AoSoAs** are used for particle positions, velocities, forces, and types, additionally storing symmetry functions and gradients of atomic energy with respect to symmetry functions for the NNP kernel. **Kokkos::Views** are also used for type-based storage, for potential parameters, masses, symmetry function parameters, and neural network parameters (weights and biases). This functionality has enabled CabanaMD to demonstrate portable performance for an LJ model across multi-core CPU compute nodes, as well as GPUs. As shown in Section 5.5, the use of appropriate levels of parallelism and data layouts for given hardware increases performance by up to 50% for the NNP kernel.

5.3.3 Modifications from previous implementation

In this work, we re-implement the NNP from the n2p2 package (`https://github.com/CompPhysVienna/n2p2`), a library-based LAMMPS [21] implementation with a model for water provided [159, 171]. Our implementation has been included in the

n2p2 package directly, available in the n2p2 Github repository [171], with an interface to the n2p2-CabanaMD functionality in CabanaMD. We test our implementation using NNP models for H2O, as well as Ni, Cu, Si, Ge, Li, and Mo from a recent comparative benchmarking study [57, 172]. Several key changes were necessary in translating the original n2p2 implementation to n2p2-Cabana, particularly for GPU architectures: (i) move from large storage classes and structs to minimal **Kokkos::Views** and **Cabana::AoSoAs**, (ii) replace explicit OpenMP pragmas with **Kokkos::parallel_for** and **Cabana::neighbor_parallel_for**, (iii) rewrite some classes of object-oriented code to use less memory (iv) replace code that is not GPU-compatible, such as replacing vector iterators with **Kokkos::parallel_for**, and (v) split computationally expensive kernels and use Kokkos and Cabana constructs to consider various levels of parallelism (parallelism over only atoms vs atoms and neighbors). Wherever possible we use the existing n2p2 library capabilities and inherit from existing n2p2 classes, extending and modifying to add the Kokkos and Cabana features discussed above.

The n2p2 LAMMPS implementation was created and optimized extremely well for standard CPU computing, avoiding floating point operations by precomputing and storing variables wherever possible. In addition, a focus on MPI-parallelization, with OpenMP included mainly for MPI+X, indicates that the code was not intended for significant threading performance, meaning that a direct re-implementation of n2p2 in CabanaMD would result in extremely low utilization of GPU resources. To avoid this, an almost complete reversal was necessary: store and move as little data as possible, recomputing where necessary. The n2p2 implementation neatly exposed potential points of parallelization and we took advantage of this with Kokkos and Cabana parallel constructs, using thread-safe atomics as needed for symmetry function and force updates. However, the data layout needed to be fully inverted, from structs of all atom and neighbor properties to minimal AoS or AoSoA in Cabana. We had to further replace some of the object-oriented code structure with kernel-based parallel code.

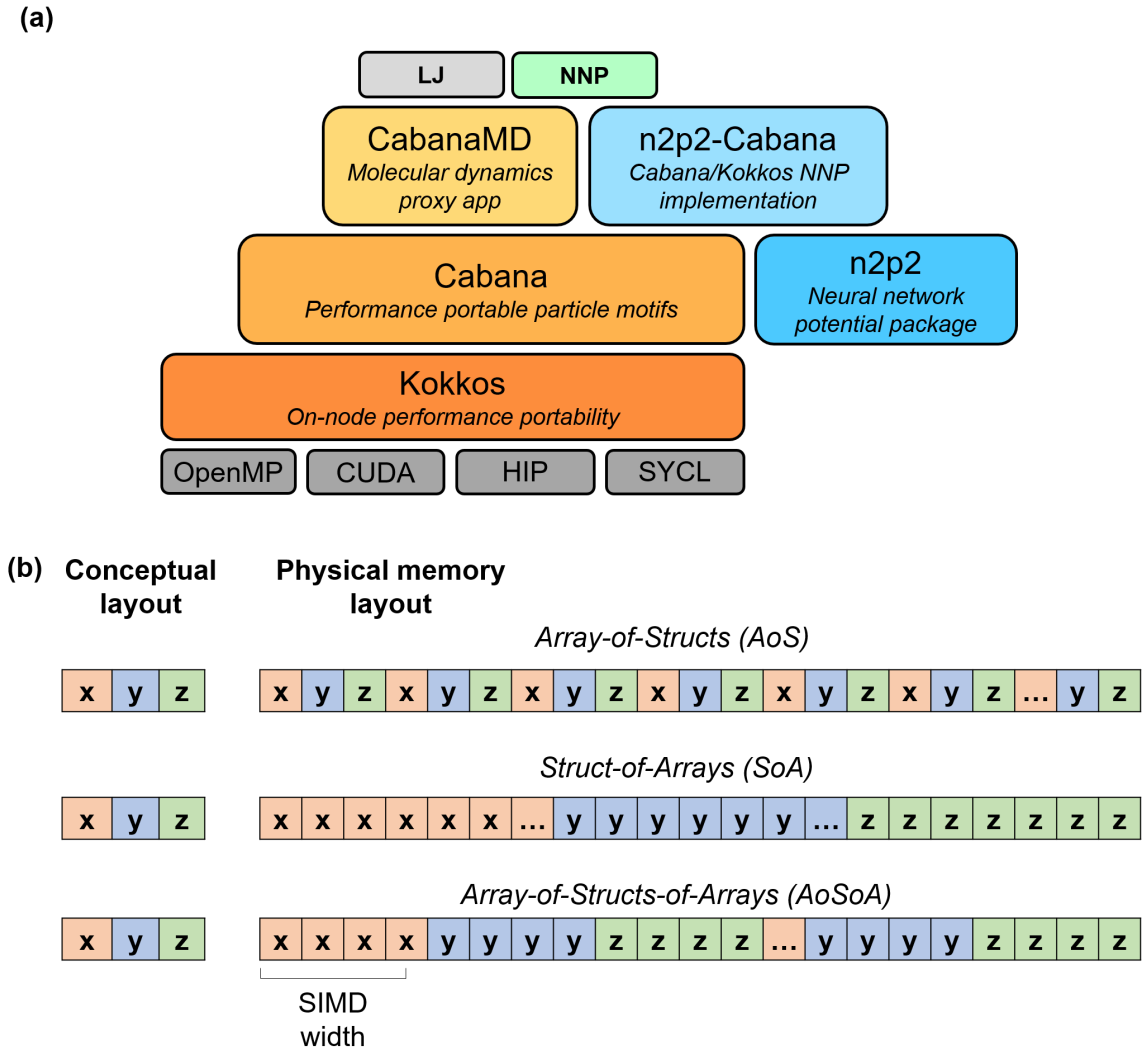

Fig. 5.2. (a) Software stack representing CabanaMD-NNP (b) Cabana AoSoA data structure allowing flexibility in data layout, compared to standard AoS and SoA.

5.4 Performance of our implementation

We show on-node performance with many-core CPU and GPU architectures on a current leadership-class supercomputer, the pre-exascale Lassen machine at Lawrence Livermore National Laboratory (LLNL), see Figure 5.3. CPU scaling behavior is documented for an IBM POWER9 node (with 40 cores per node and 4 threads per core),

while GPU results are shown for a single NVIDIA V100 GPU. This architecture matches the Summit and Sierra supercomputers at Oak Ridge National Laboratory and LLNL, respectively. Each test system was simulated in the ground state structure at room temperature with NVE dynamics, we used [173] to generate the water structures. Throughout, we used Kokkos v3.2, Cabana (commit a7ace07), n2p2 v2.0.1, gcc 7.3, and CUDA 10.1. The CabanaMD-NNP functionality will updated through both Github repositories: `https://github.com/ECP-CoPA/CabanaMD` and `https://github.com/CompPhysVienna/n2p2`.

5.4.1 CPU OpenMP Performance

Figure 5.3(a) shows strong scaling performance with increasing number of OpenMP threads on a single POWER9 node, comparing the n2p2 LAMMPS implementation with CabanaMD for a nickel test system [172]. The n2p2 implementation shows saturating performance due to a memory-bound implementation with large structs to store atom and neighbor properties, including symmetry functions and derivatives. In contrast, our thread-bound implementation shows linear thread scaling (with slight reduction from linear for hyperthreading), which results in a greater parallel efficiency for our implementation. The n2p2 algorithms show better performance for the smallest numbers of threads, where the heavy compute and low resource nature of the task result in performance gains from a storage-based approach, surpassing exposed parallelism. However, even in using hybrid MPI+OpenMP parallelization (a large focus of their effort), the n2p2 implementation is more limited in system size by memory than CabanaMD. Ultimately, CabanaMD shows a speedup of ~4x for a moderate system size (256K atoms) using the full node (all threads), while n2p2 is faster by ~10x when restricted to a few threads, see Fig. 5.3(a). Weak scaling from Figure 5.3(b) reinforces these points, where n2p2's superior performance for smaller systems with a single thread is overtaken by the parallelism from Cabana and Kokkos in CabanaMD. Figure 5.3(c) and 5.3(d) similarly compare n2p2 and CabanaMD strong

and weak scaling for H_2O [171], and while we see that n2p2 is faster than our implementation, CabanaMD shows better thread scalability. We attribute this to the larger cutoff for the H_2O potential (6.4Å vs 3.9Å for Ni), which significantly increases the number of neighbors for which computations are required. Thus, the H_2O model represents a cross-over point, where the memory and thread-based approaches are similar, for the POWER9 hardware. The tradeoff between recomputing values and storing them for more expensive models like H_2O will be a focus of future work in CabanaMD. Overall, CabanaMD is faster for many (but not all) systems with maximal resources, but n2p2 will always be faster in serial.

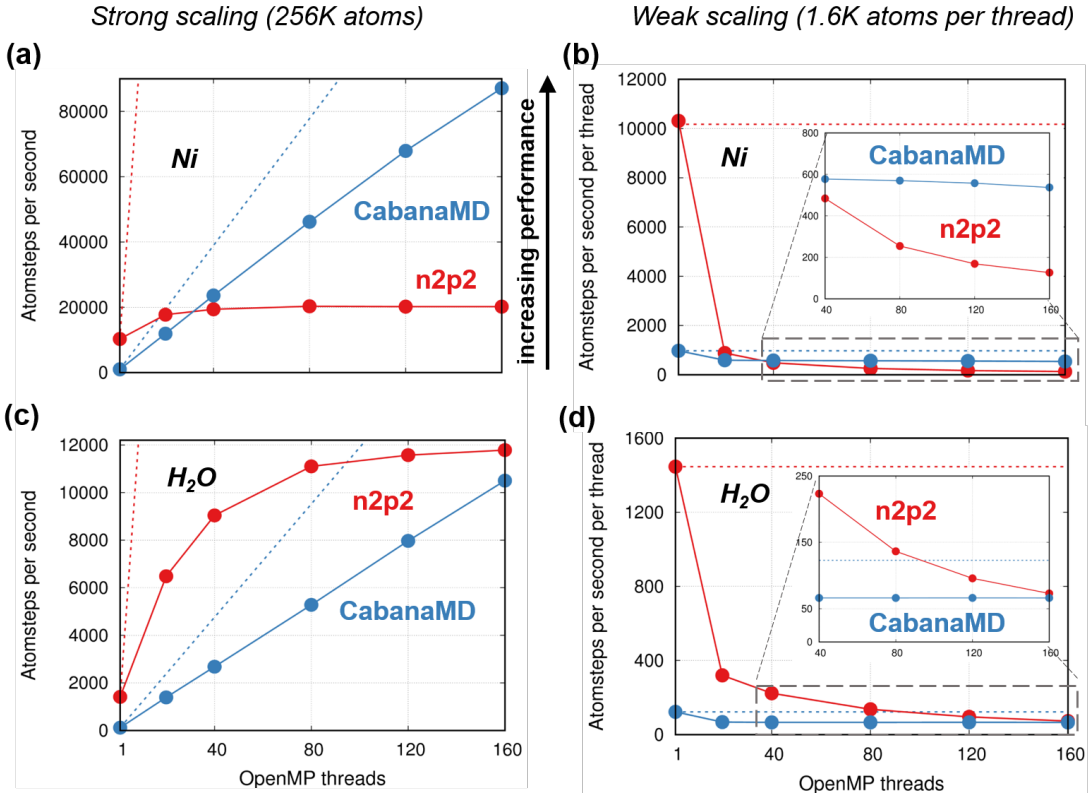

Fig. 5.3. (a) Strong and (b) weak scaling for Ni contrasting the n2p2 and CabanaMD implementations, evaluated on a single IBM POWER9 node with a maximum of 256K atoms. (c) Strong and (d) weak scaling for H_2O further comparing codes. Perfect scaling is shown in the dashed lines.

5.4.2 GPU Performance

The use of the Kokkos and Cabana libraries provides us a single-source solution for performance portability. Figure 5.4 highlights the performance of our implementation on the GPU. For H_2O, the GPU version shows a speedup of ~14x compared to the CPU compared to the CabanaMD CPU for Ni and ~3x for H_2O (and ~12x, ~11x and ~9x for other single element systems: Cu, Mo and Ge). Energy conservation for all systems is documented in Figure 5.5. While CabanaMD is ~4x faster than n2p2 for threaded-only CPU performance, Figure 5.6 shows that n2p2 is ultimately faster when using MPI across one node (40 ranks, 4 threads per rank), also by ~4x. The CabanaMD CPU performance does not significantly change when using MPI+OpenMP or only OpenMP; the GPU version is ~3x faster than the MPI n2p2 on 1 rank, and ~10x on the full node (4 GPU). Energy conservation for all systems is documented in Figure 5.5. Total energy is conserved for the neural network interatomic models of Ni, Cu, and Si and while Li, Mo, and Ge show small energy drifts, when comparing to the fluctuation of the potential energy. A further trained interatomic model could likely more accurately conserve energy for these systems. Crucially, the implementation decisions detailed in Section 5.3 resulted in faster GPU simulations, by first achieving good CPU thread-scaling. Scaling studies across nodes will be a focus of future efforts.

Our implementation also enables systems of up to 20M atoms to be run on a single CPU node and 4M atoms on a single GPU (Ni), where performance in Figure 5.4 is shown up to the memory limits of the hardware for each case. We achieve this by avoiding a memory-based approach, which would particularly limit the system sizes and speeds achievable on the GPU. For Ni, CabanaMD is faster across the range of system sizes, except for those below a few thousand atoms. For H_2O, n2p2 is faster except for the largest systems achievable. Of course, these comparisons change with different hardware; for example, Figure 5.7 shows results for Intel Xeon (Broadwell) CPUs, with fewer cores and less memory per node. The Xeon node has significantly

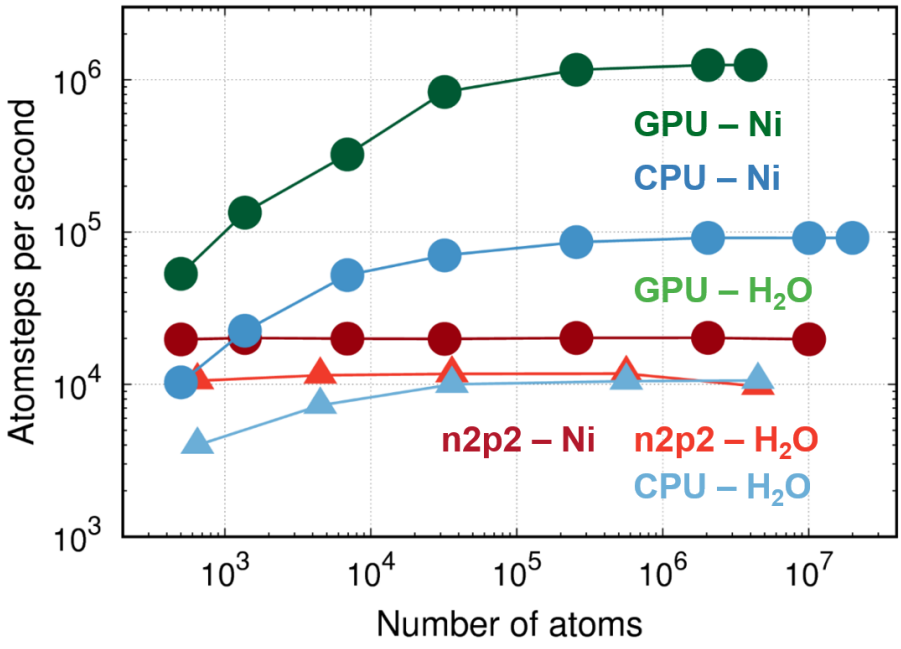

Fig. 5.4. Performance comparisons for Ni and H_2O, with CabanaMD on a single NVIDIA V100 GPU and CabanaMD with 176 threads on an IBM POWER9 CPU node, and the n2p2 implementation with the same CPU

less total memory and fewer cores (36 total threads) as compared to POWER9. On both CPUs, the memory-based n2p2 strategy is more limited in system size compared to CabanaMD.

5.5 Parallelism and Data Layout Improvements

The CabanaMD implementation shown above largely retains the algorithms used by n2p2 in the current LAMMPS implementation. However, the power of the Cabana implementation is in enabling easy exploration of new algorithms and parallelization strategies via separation of concerns, as well as exploration of data layouts via flexible AoSoAs. To better understand the performance of our implementation, we first breakdown the kernel timings into the three steps from Figure 5.1, shown in Figure 5.8. We observe that the radial calculations are a minor part of the computation

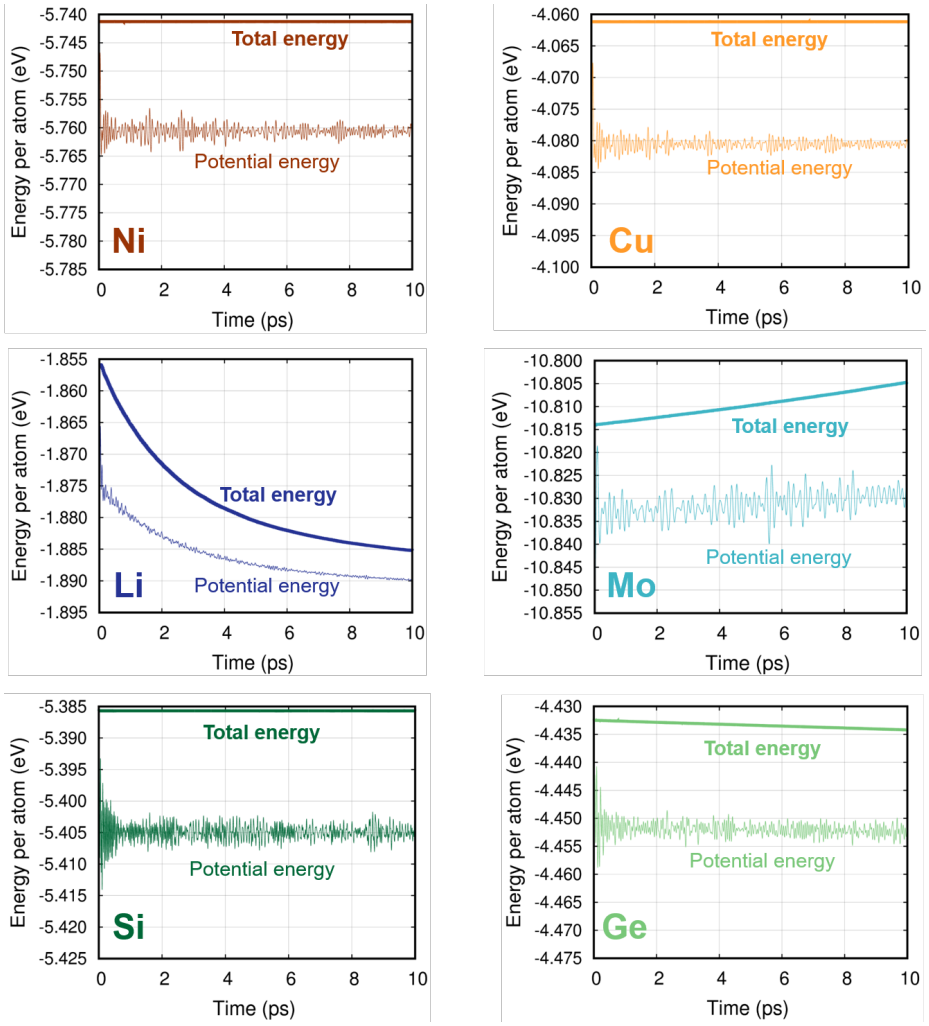

Fig. 5.5. Energy conservation with an NNP for various single element systems

across hardware, with the calculation of angular forces taking up the most time per MD step. In addition, the neural network time is negligible on the CPU and approximately one third of time for the GPU. This highlights that while the NNP kernel is novel and includes a new compute step (propagation through the neural network), optimizations targeted at other models, up to and including the LJ kernel, are still most useful for improvements.

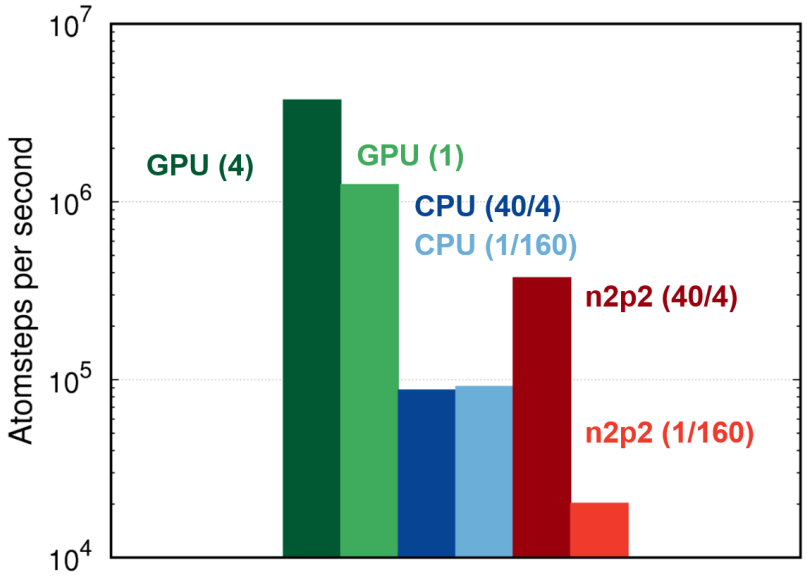

Fig. 5.6. MPI performance comparisons for Ni (2M atoms) with CabanaMD on 4 and 1 NVIDIA V100 GPU(s), CabanaMD with 40 MPI ranks (4 threads/core) and 1 MPI rank (160 threads) on an IBM POWER9 CPU node, and n2p2 with the same CPU

The main algorithmic changes involved recomputing rather than reading stored values throughout the code. In addition, we split the computation of symmetry functions into separate radial and angular symmetry function kernels and exposed greater parallelism, using **Cabana::neighbor_parallel_for** to parallelize computations over both atoms and neighbors. This functionality directly uses hierarchical parallelism, mapping loops over atoms and neighbors to the multiple levels of compute and memory hardware, which we compare and contrast with flat parallelism (threaded parallelism over atoms only). Use of the hierarchical parallelism require an additional atomic update of the symmetry functions, in addition to atomic update of the final forces that is always necessary (unless serial). Figure 5.9 shows that the use of hierarchical parallelism results in ~25% performance improvement on the CPU for large systems, while significantly reducing performance for a small system with ~1000 atoms. At the smallest sizes there is not enough total work to make the additional

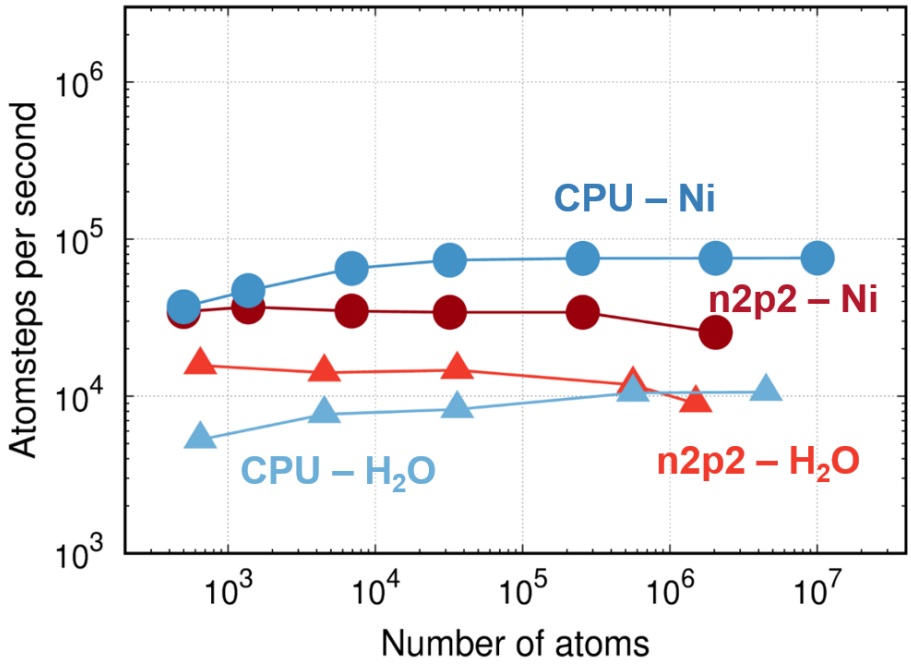

Fig. 5.7. Performance comparisons for Ni and H_2O on a single Intel Xeon (Broadwell) node with 36 threads between CabanaMD and n2p2

overheads of multiple levels of parallelism worthwhile; however, for most systems on the CPU the higher exposed parallelism outweighs the necessary additional atomic operations, primarily because the total number of threads is small. On the GPU, the use of hierarchical parallelism degrades performance by over a factor of 2, where in this case the much larger number of threads results in significant contention between them, making it difficult to leverage the greater exposed parallelism. There is again a cross-over for the smallest systems on the GPU where, without enough work to fill the GPU exposing neighbor parallelism improves performance (albeit with much lower performance than larger systems overall). Throughout sections 5.4.1 and 5.4.2, we thus use hierarchical parallelism on the CPU and flat parallelism on the GPU, except for the smaller atom counts in Figure 5.4 (where each system has a different cross-over point). This was done with a simple command line flag, a capability possible due to use of flexible Kokkos and Cabana constructs.

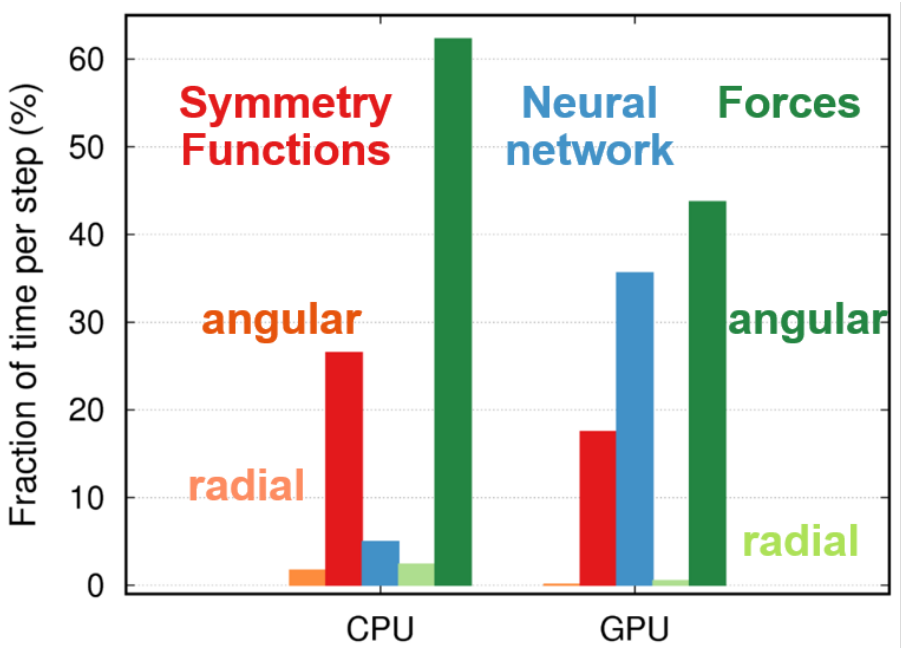

Fig. 5.8. Fraction of time taken by each portion of the CabanaMD-NNP compute kernel for a ~1.5M atom Ni system in computing symmetry functions, propagating those symmetry functions through the atomic neural networks, and computing forces. Note that the force contribution from the neural network is computed and counted within the force bar

A third level of parallelism, again using **Cabana::neighbor_parallel_for**, parallelizing computations over atoms, neighbor, and angular neighbors is also available, but did not show any performance improvement as compared to parallelizing over neighbors for this system. This third level of parallelism is mapped to vector units directly, not amenable to random access neighbor operations which dominate these simulations. However, for other materials, system sizes, interatomic models, or algorithms this feature could be relevant.

We can also improve performance by choosing the appropriate data layout for the architecture. Figure 5.10 shows performance gains obtained (on CPU and GPU) by varying the vector length for the **Cabana::AoSoA**, in addition to demonstrating the performance tradeoffs in using a single combined AoSoA vs using multiple separate

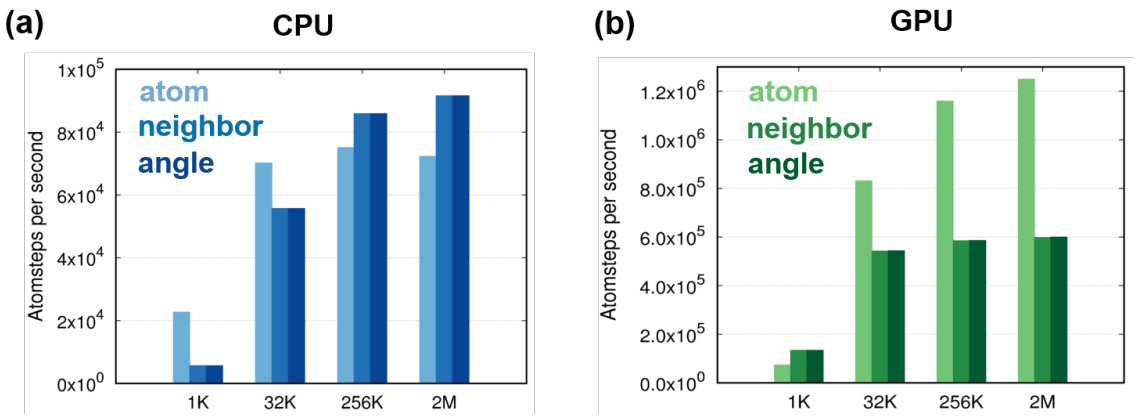

Fig. 5.9. Performance for Ni compared across various levels of parallelism and Ni system sizes on (a) a single IBM POWER9 node and (b) a single NVIDIA V100 GPU

AoSoAs (for each particle property, including NNP-specific arrays). Each feature contributes to tradeoffs between data locality and consecutive access. It should be noted that the vector length of the NNP specific AoSoAs were separately optimized to 1 for CPU and 32 for GPU.

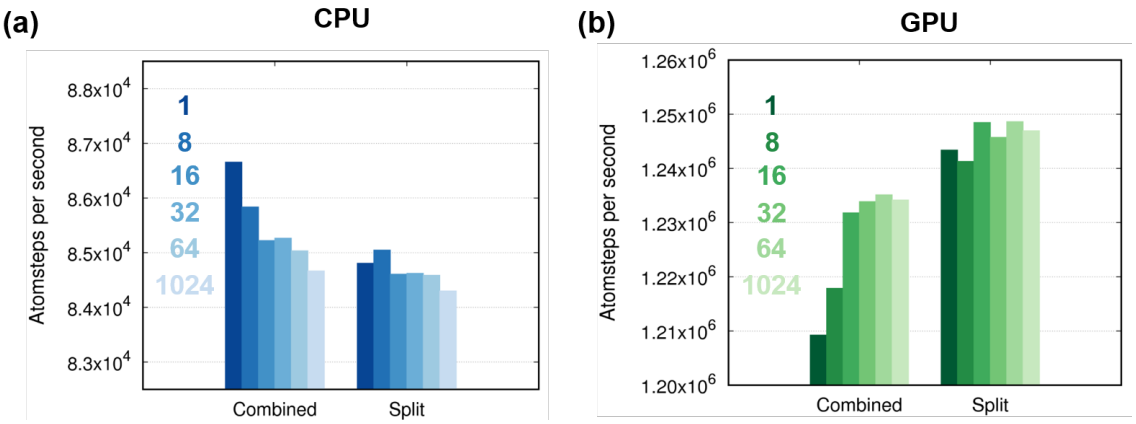

Fig. 5.10. Performance for Ni compared across various vector length and AoSoA sizes on (a) a single IBM POWER9 node and (b) a single NVIDIA V100 GPU for 256K (CPU) and 2M (GPU) atom Ni systems

Overall, the trends are reversed between CPU and GPU. For the CPU, a vector length of 1 (AoS) gives best performance. As the vector length increases, data locality is reduced and, without significant vectorization, the performance degrades, Figure 5.10(a). Because the majority of the simulation time is from random access (neighbor list) force computations, the vectorizable benefits from the intermediate vector length AoSoA layout are not currently taken advantage of; however, it is possible to rewrite MD kernels to be partially vectorizable [174, 175]. Breaking the AoSoA into individual particle properties does not significantly change performance, but is consistently slightly slower, indicating the CPU benefits from the higher data locality of the combined layout. The combined AoSoA with vector length of 1 results in ~3% better performance over the slowest data layout on the CPU and we accordingly use this layout throughout Section 5.4. On the GPU, performance slightly increases with increasing vector length, Figure 5.10(b). The loops over symmetry functions and neural networks can to some degree take advantage of the vector lengths near the warp size and above, but without significant speed up. We again see a 3% improvement between the fastest and slowest GPU data layout and used the split AoSoA with a vector length of 32 throughout.

5.6 Conclusions

We have demonstrated a performance portable neural network interatomic model for large-scale MD within the CabanaMD proxy app, comparing performance with the existing LAMMPS-based n2p2 library. CabanaMD-NNP extends n2p2 to interface with the Cabana library for particle methods, itself built on the Kokkos library for on-node performance portability, enabling continuing exploration of parallelization and data layout/access on emerging computing architectures. In addition to a thread-scalable CPU implementations, we demonstrated GPU performance for an NNP. These improvements are also relevant to the myriad of Behler-style NNP implementations and extensions [176–182] and other new models using NN in different

ways [183,184]. Further, learned-descriptor-based non-parametric models, containing completely unique kernels, will pose challenges to improving performance not seen here and will be subject to future work [53,165,166].

The main ideas of this work apply to any current CPU-only code looking to utilize GPU resources: with a performance portable library or programming model, data storage and movement should generally be avoided in favor of re-computing. In our case, this manifests itself in our choice to re-compute symmetry functions, derivatives, and distances as opposed to storing large data structures for each atom and neighbor. In addition, exposing as much parallelism as possible and determining which levels to thread over is integral. Overall, improving CPU multi-threading was a direct path to achieving significant GPU speedups. CabanaMD-NNP is available at `https://github.com/ECP-CoPA/CabanaMD`, with our modifications to the n2p2 code available at `https://github.com/CompPhysVienna/n2p2`.

The ability to simulate millions or billions of atoms with nearly-quantum accurate models, using performant MD implementations such as ours, will significantly expand the set of scientific problems within the reach of classical MD. We intend to use the CabanaMD-NNP implementation to investigate complex materials without currently available models: solid-solid phase transitions, AM-rate solidification processes, physical and chemical phenomena in energetic materials, and microstructural evolution in metallic alloys. In addition, for NNP models that have already been developed, CabanaMD-NNP enables unprecedented large-scale MD simulations of complicated processes such as recrystallization in phase change memory applications [151] and diffusion in amorphous solid-state battery electrolytes [152].

In Chapters 3, 4, and 5, we have seen how molecular dynamics simulations can be used, and improved, to understand material behavior and aid the design of novel materials. We now turn to another arm of the Materials Genome Initiative, the increasingly popular avenue of incorporating machine learning methods in material science. Specifically, we will focus on developing novel machine learning methods to extract interpretable laws directly from data.

6. PARSIMONIOUS NEURAL NETWORKS AND THEIR APPLICATION TO MOLECULAR DYNAMICS

Machine learning is playing an increasing role in the physical sciences and significant progress has been made towards embedding physics into domain-agnostic models. Less explored is the potential of machine learning to discover interpretable physical laws from observational data. In this chapter we will combine neural networks with evolutionary optimization to design Parsimonious Neural Networks (PNNs) that can extract interpretable laws from data. We will demonstrate the applicablility of PNNs by finding laws in two cases: (i) the time evolution of a point particle under a highly nonlinear potential, and (ii) the melting temperature of materials from fundamental properties.

The work in this chapter has been published in the arXiv and can be found as: Saaketh Desai, Alejandro Strachan, "Parsimonious neural networks learn classical mechanics, its underlying symmetries, and an accurate time integrator" arXiv preprint arXiv:2005.11144 (2020).

6.1 Introduction

Machine learning (ML) can provide predictive models in applications where data is plentiful and the underlying governing laws are unknown [185–187]. These approaches are playing an increasing role in the physical sciences where data is generally limited but underlying laws (sometimes approximate) exist [188–193]. For example, ML surrogate models are being used in electronic structure calculations [194] and molecular dynamics (MD) simulations [50, 195, 196]. One of the major drawbacks of the use of ML in the physical sciences is that models often do not learn the underlying physics of the system at hand, such as constraints or symmetries, limiting their ability to gen-

eralize. In addition, most ML models lack interpretability. That is, ML approaches generally neither learn physics nor can they explain their predictions. In many fields, these limitations are compensated by copious amounts of data, but this is often not possible in areas such as materials science where acquiring data is expensive and time consuming. To tackle this challenge, progress has been made towards using knowledge (even partial) of underlying physics to improve the accuracy of models and/or reduce the amount of data required during training [48, 197]. Less explored is the use of ML for scientific discovery, i.e., extracting physical laws from observational data, see Refs. [46, 47, 198] for some notable exceptions. In this work we combine neural networks (NNs) with stochastic optimization to find the simplest model capable of (i) describing the dynamics of a particle under a highly non-linear potential, and (ii) generating expressions to predict the melting temperature of materials, solely from observational data. Our hypothesis is that the requirement of parsimony will result in the discovery of the physical laws underlying the problem. We find that the resulting descriptions are not only interpretable but also satisfy non-trivial underlying symmetries of the physical system. This second feature makes the parsimonious neural networks (PNNs) significantly more accurate than generic NN models. Stochastic optimization has been previously used in conjunction with backpropagation to improve robustness or minimize overfitting in models [199–205], this work extends these ideas to learn physics from data.

The power of physics-based ML is well documented and remains an active area of research. Neural networks have been used to both parametrize and solve differential equations such as Navier Stokes [48] and Hamilton's equations of motion [49]. Recurrent architectures have also shown promise in predicting the time evolution of systems [58, 59]. These examples focus on using prior knowledge of the underlying physics to guide the model, often as a numerical constraint added to the models, or by using the underlying physics to numerically solve equations with variables predicted by the ML algorithms. In contrast, we are interested in learning the physics, including complete numerical solutions, directly from data, without prior knowledge.

Pioneering work along these lines used sparse regression to identify invariants in dynamical systems by constructing a library of candidate functions and matching partial derivatives of invariants to the numerical derivatives of the training data [46]. A similar approach was used to identify partial differential equations from data [47]. More recently, neural networks have been used to discover underlying equations directly from data [206–209], using conventional neural networks as encoders and connecting the encoders to custom networks aimed at equation discovery. Yet another approach has been to design custom deep neural networks whose layers can be evaluated via standard differential equation solvers [210]. We build on and extend these ideas to propose PNNs designed to find the simplest possible model consistent with the underlying data in a context where the discovered model must be highly accurate and satisfy conservation laws to merit use. We first apply PNNs to learn the equations of motion that govern the Hamiltonian dynamics of a particle under a highly non-linear external potential with and without friction. Our hypothesis is that by requiring parsimony (e.g. minimizing adjustable parameters and favoring linear relationships between variables) the resulting model will not only be easily interpretable but also will be forced to tease out the symmetries of the problem. We find that the resulting PNN not only lends itself to interpretation (as Newton's laws) but also provides a significantly more accurate description of the dynamics of the particle when applied iteratively as compared to a flexible feed forward neural network. The resulting PNNs conserve energy and are time reversible, i.e. they learn non-trivial symmetries hidden in the data but not explicitly provided. By virtue of being based on neural networks, PNNs explore a large function space and obviate the need for estimating numerical derivatives or matching a library of candidate functions, as was done in prior efforts [46,47,198]. The PNN approach also allows complex composition of functions by virtue of using neural networks, as opposed to prior sparse regression efforts, which combine functions linearly. The generalizability of PNNs is demonstrated with a second example: discovering models to predict the melting temperature of materials from atomic properties. By varying the relative importance of parsimony and

accuracy in the genetic optimization, we discover a family of pareto optimal models, including the celebrated Lindemann law [211].

6.2 Discovering molecular dynamics integration schemes from data

As a first example, we consider the dynamics of a particle under an external Lennard-Jones (LJ) potential with and without friction. In both cases the training data consists of accurate numerical trajectories with various totals energies. The choice of numerical input data was made for convenience, it could have been obtained experimentally.

6.2.1 Description of training data

The training trajectories are generated using the LAMMPS software package [21], a popular choice to perform molecular dynamics simulations. We generate trajectories of a two-particle system where interactions between the atoms are described by a Lennard-Jones potential with parameters $\epsilon = 0.5$ eV and $\sigma = 2.5$ Å. The system consists of one particle at the origin and the other particle fixed at x = 1.12σ (the y and z coordinates are fixed to be 0). The simulation cell is arbitrarily chosen to be cubic with a side length of 80σ to avoid interactions between periodic images of particles. Four different trajectories were generated under the microcanonical (NVE) ensemble, with different initial velocities for the atom at origin, corresponding to total energies of -0.999ϵ, -0.895ϵ, -0.797ϵ and -0.733ϵ, using a timestep of $0.011962\sqrt{\frac{\epsilon}{\sigma^2}}$. The equations of motion are integrated using the velocity Verlet algorithm and each trajectory contains 100,000 snapshots of atomic coordinates, velocities and forces at each step, for a total of 400,000 snapshots. We consider this highly accurate trajectory to be our baseline, from which we sample frames at different frequencies to obtain trajectories at different timesteps. So, to sample a trajectory with timestep $0.011962\sqrt{\frac{\epsilon}{\sigma^2}}$, we sample every 10^{th} snapshot from each trajectory. This amounts to 40,000 data points consisting of input and output arrays, with the input array

consisting of position, velocity and force at time t, while the output array consisted of position, velocity and force at time t+Δt (i.e., one step later). The data is split into training and validation sets in an 80:20 ratio, with an additional test set of 10,000 points was generated independently at an energy of -0.884 ϵ. For the damped dynamics cases, a frictional force proportional to negative the velocity is added, with frictional coefficient $\gamma = 0.004$ eVps/Å^2. Note that the PNN is not informed about the additional frictional force for the trajectory with friction. The only way for the PNN to learn about the damping is through the training data trajectories. While all quantities are mentioned in Lennard-Jones units, in practice we use a different unit system, with positions in pm, time in fs and energy in eV. This ensures that the inputs to the network are of the same scale, where the standard practice of normalizing the data could lead to a loss of interpretability in a general case. The training data is available on nanoHUB [212].

6.2.2 Performance of a feed forward neural network

Before describing the PNN model, we establish a baseline by training a standard feed forward neural network (FFNN) on our data for the case without friction and evaluating its performance as an integrator. This network consists of three hidden layers consisting of 20, 100 and 20 neurons respectively, with the rectified linear ('relu') activations applied to the hidden layers and linear activations applied to the output layer. We use mean squared error (MSE) as the loss function and the Adam optimizer [213] with a learning rate of 10^{-3}. The network is trained using the Keras package [214] and the training data and models are available on nanoHUB [212].

We find the FFNN to be capable of matching the training/validation/test data reasonably well. The root mean squared errors (RMSE) on the position and velocity, across the training, validation and testing sets are ($1.48\text{x}10^{-5}$, $9.73\text{x}10^{-5}$), ($1.54\text{x}10^{-5}$, $9.98\text{x}10^{-5}$) and ($1.59\text{x}10^{-5}$, $9.94\text{x}10^{-5}$) respectively. However, the network has poor predictive power. Using it iteratively to find the temporal evolution of the particle

results in significant drifts in total energy over time, and a lack of time reversibility, see Figure 6.1. Reversibility is judged by applying the NN sequentially 1,000 times, followed by time reversal (changing the sign of the particle's velocity) and applying the NN for a second set of 1,000 steps. We find that deeper architectures do not improve the RMSE, reversibility or energy conservation.

Needless to say, these FFNNs are not interpretable. These results highlight the difficulty of the problem at hand. Hamilton's equations for classical mechanics represent a chaotic set of differential equations and small errors in each timestep accumulate rapidly resulting in diverging trajectories. While some previous work attempts to address such issues by explicitly training for multiple steps using a recurrent architecture [208], we are interested in solutions stable over timescales far greater than those typically accessed by current recurrent architectures. Finding such models is non-trivial and the development of algorithms to integrate equations of motion with good energy conservation and time reversibility has a rich history [33, 35, 215, 216]. An example of such algorithms is the popular Verlet family of integrators [33,35] that are both reversible and symplectic [217]; their theoretical justification lies in Trotter's theorem [218].

6.2.3 Feed forward network with a force sub-network

Having established that a standard feed forward neural network performs poorly as integrator, we seek to establish whether a standard feed forward neural network with a force sub-net built in can perform well as an integrator. The purpose of this baseline is to establish if providing the network with the non-linear force model is sufficient to discover integration schemes. The force sub-net is a two-layer network (with 10 neurons in each layer) that attempts to predict the force given a position as input, as shown in Figure 1 of the manuscript. The hidden layers of this network have 'tanh' activations and the network is trained using the same protocol as the FFNN. We thus build a network as shown in Figure 6.2(a) and train the network, again

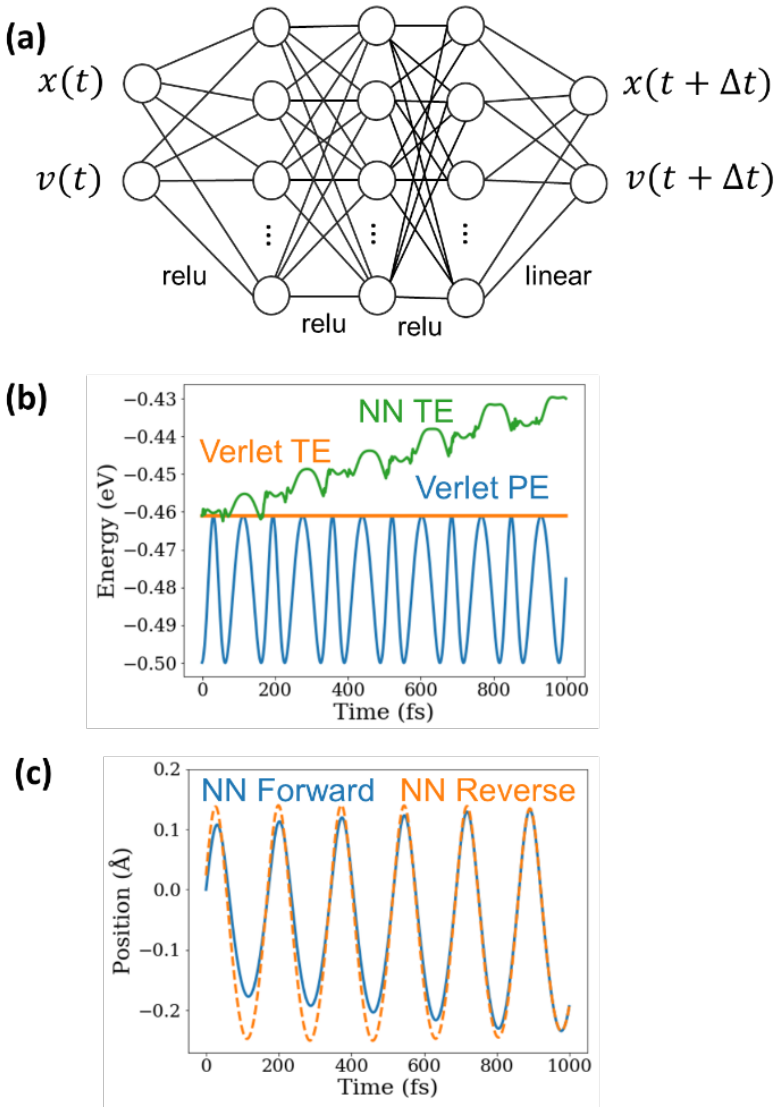

Fig. 6.1. (a) Standard feed forward neural network, attempting to predict positions and velocities one step ahead (b) Energy drift for the feed forward NN compared to the Verlet algorithm (c) Forward and reverse trajectories generated by the feed forward NN, showing the lack of reversibility

using the same protocols used for the FFNN. We find that the network still performs poorly as an integrator with RMSEs on the train, validation and test sets for the position and velocity of ($1.60\text{x}10^{-5}$, $3.84\text{x}10^{-4}$), ($1.76\text{x}10^{-5}$, $4.95\text{x}10^{-4}$) and ($1.56\text{x}10^{-5}$,

4.38×10^{-4}). Figure 6.2(b) and 6.2(c) document the poor energy conservation and reversibility. This shows that adding the information about the force is not the key to the development of accurate models for classical mechanics.

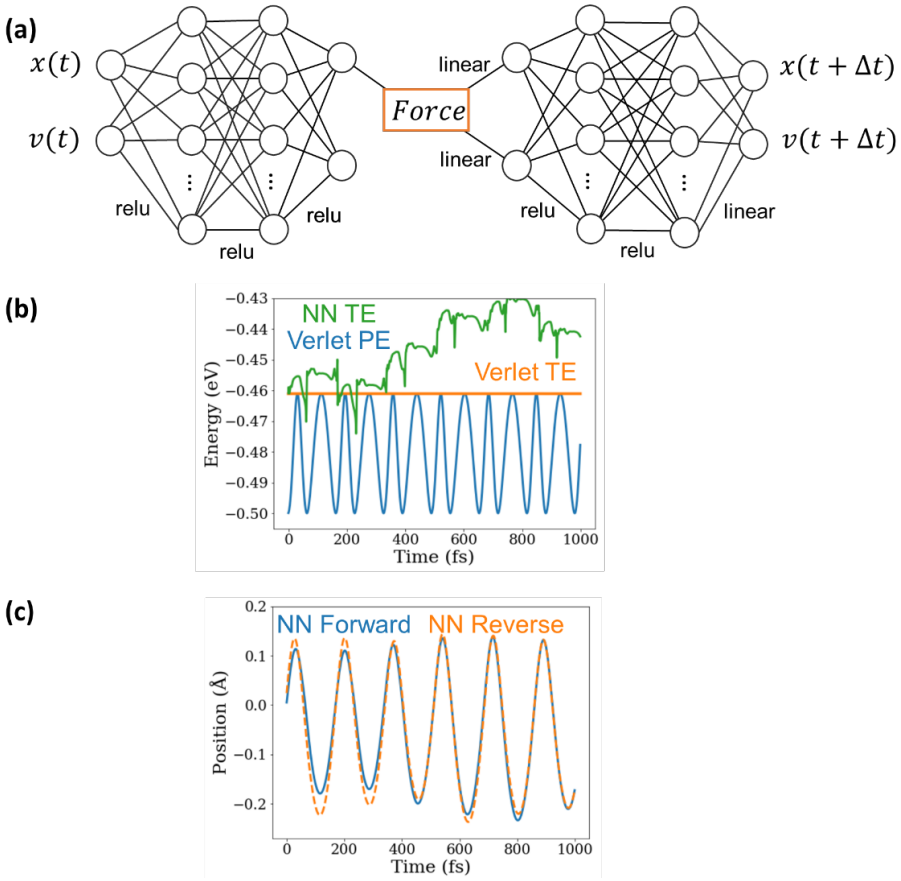

Fig. 6.2. (a) Feed forward neural network with a fixed force sub-net built in (b) Energy conservation (TE: total energy) and (c) reversibility are poor even for this network

6.2.4 Parsimonious neural networks

Having established the shortcomings of the feed-forward neural network, we now switch to parsimonious neural networks (PNN). We begin with a generic neural network shown in Figure 6.3 and use genetic algorithms to find the corresponding PNN.

The neural network consists of three hidden layers and an output layer with two outputs, the position and velocity of the particle one timestep ahead of the inputs. Each hidden layer has two neurons, with the central hidden layer including an additional force sub-net, a network pre-trained to predict the force on the atom given its position. Our use of a pre-trained force sub-net is motivated by the prior success of neural networks in predicting interatomic forces in a wide variety of materials significantly more complex than our example [160, 219, 220]. Our focus is on learning classical dynamics and the use of a force sub-net only incorporates the physical insight that the force is an important quantity. In the context of scientific discovery via neural networks, the force sub-net is analogous to the latent variable learnt by an encoding network [206].

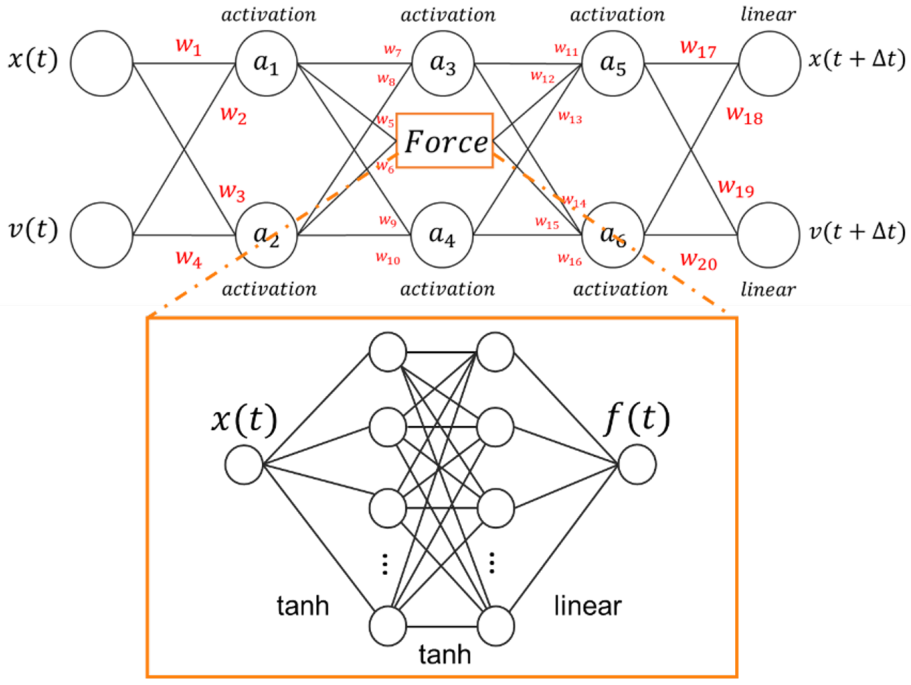

Fig. 6.3. Neural network used as the starting point to find the parsimonious neural network as the network that explains the data in the simplest manner possible. The force sub-network is highlighted in orange and is fed into the neural network as a pre-trained model, whose weights are subsequently kept fixed throughout

The starting neural network is a highly flexible mapping from input positions and velocities to output positions and velocities and finding a PNN model reduces to identifying the simplest function capable of describing this relationship. This is an optimization problem in the space of functions spanned by the possible activations and weights of the network. We consider four possible activation functions: linear, rectified linear unit (relu), hyperbolic tangent (tanh), and exponential linear unit (elu). The weights connecting the artificial neurons can be either fixed or trainable, with the fixed category allowing the following values: 0, $\frac{1}{2}$, 1, 2, $\frac{\Delta t}{2}$, Δt, and Δt, with Δt the timestep separating the inputs and outputs. This is motivated by the fact that physical laws often involve integer or simple fractional coefficients and that the timestep represents important information. Future work will consider other common constants found in physical laws, including additional fractional numbers and irrational numbers such as π and e as well as additional activation functions. Our network has twenty weights (each with eight possible settings) and six activation functions to optimize, see Figure 6.3 (top panel). A brute force approach to finding a PNN model would require training $\sim 10^{21}$ neural networks, an impossible computational task even for the relatively small networks here. We thus resort to evolutionary optimization, using a genetic algorithm to find the most parsimonious network consistent with our training data.

The genetic algorithm favors: (i) linear activation functions over non-linear ones, and (ii) non-trainable weights with simple values over optimizable weights. The objective function is defined to capture both the accuracy of a network in reproducing the testing data and its parsimony:

$$F = f_1(E_{test}) + p\left(\sum_{i=1}^{N} w_i^2 + \sum_{j=1}^{N} f_2(w_j)\right) \tag{6.1}$$

where E_{test} represents the mean squared error of the trained PNN on the testing set and f_1 is a logarithmic function that converts the wide range errors into a scale comparable to the other terms. p is a parsimony coefficient that decides the weight on the parsimony terms of the objective function. Increasing p will favor models that are

simple, while decreasing p will favor models that are complex but are more accurate. f_1 is defined as follows:

$$f_1(E_{test}) = 10log_{10}E_{test} \tag{6.2}$$

The second term runs over the six neurons of the network and is designed to favor simple activation functions. The linear, relu, tanh and elu activation functions are assigned scores of $w_i = 0$, 1, 2 and 3, respectively. The third term runs over the network weights and favors fixed, simple weights over trainable ones. A fixed weight value of 0 is assigned a score of 0, while other fixed weights are assigned the score 1, and a trainable weight is assigned a score of 2.

$$f_2(w_j) = \begin{cases} 0 & w_j = 0 \\ 1 & w_j = 1, 1/2, \Delta t/2, \Delta t, 2\Delta t \\ 2 & w_j = trainable \end{cases} \tag{6.3}$$

Given this fitness function, we can encode our network with 6 activations and 20 weights as an individual of length 26. The first 6 genes of this individual can take values from 0 to 3, and the next 20 genes can take values from 0 to 7. We use the DEAP package for the evolutionary optimization [221]. We start with populations of 200 and 500 individuals, evolving them for 50 generations each to discover PNNs. We use a two-point crossover with a crossover probability of 0.5, and a custom mutation operation, where the first 6 genes are randomly mutated between 0 and 3 while the next 20 genes are randomly mutated between 0 and 7, with a mutation probability of 0.3. Individuals are selected for crossover using a tournament selection scheme with a tournament size of 10. For each individual, the fitness described above is evaluated, training the network using the functional API in the Keras package [214] if the network has trainable weights. We use the simple evolutionary algorithm described in Chapter 7 of [222], as implemented in the DEAP package. For each population size, we perform five independent runs (using different seeds to initialize the initial population) to characterize the probability of successfully discovering an accurate integration scheme.

The evolution of the fitness for the best individual in each generation is shown in Figure 6.4. The complete code for the genetic algorithm optimization can be found in a Jupyter notebook on nanoHUB [212].

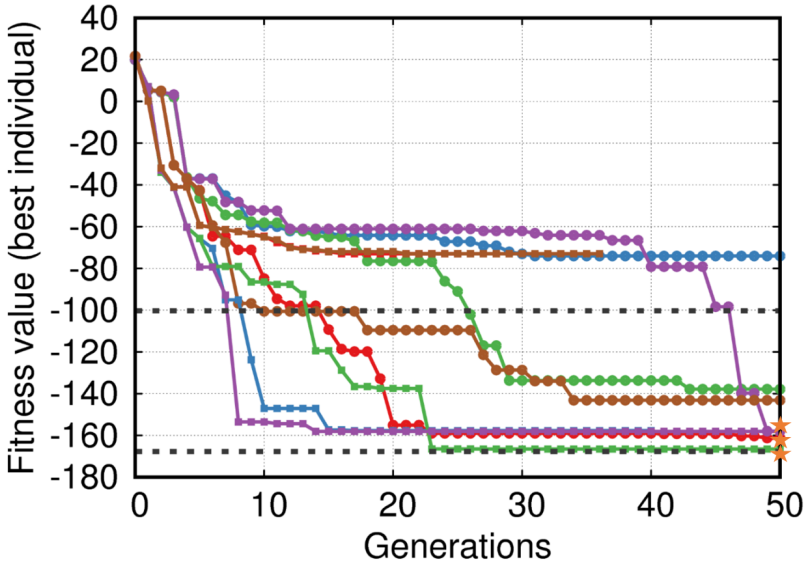

Fig. 6.4. Evolution of the fitness of the best individual in each generation. The colors represent different runs, while the black dashed lines represent the fitness value for the position Verlet algorithm and the Euler algorithm. Runs with a population size of 500 are marked with squares and runs with a population size of 200 are marked with circles. The orange stars represent the top 3 PNNs.

The resulting PNNs reproduce the training, validation and testing data more accurately than the architecturally complex FFNNs. Figure 6.5(a) compares the RMSE for positions and velocities from the optimal PNN (denoted PNN1) to the feedforward NN. Remarkably, the PNNs also result in excellent long-term energy conservation and time reversibility, evaluated using the same procedure as before. Figures 6.5(b) and 6.5(c) compare the total energy and trajectories generated by PNN1, the FFNN and the velocity Verlet integrator. We see that PNN1 learns both time-reversibility and that total energy is a constant of motion. This is in stark contrast to the physics-agnostic FFNN and even naive physics-based models like a

first order Euler integration. A few of the top-ranked PNNs perform similarly to PNN1 and they will be discussed below.

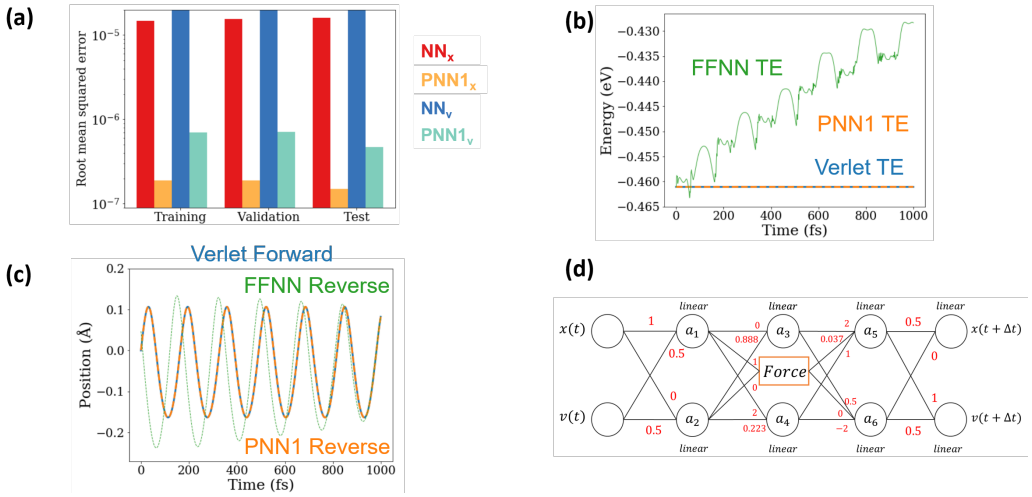

Fig. 6.5. (a) PNN model 1 RMSEs on the training/validation/test sets compared to the feed forward network (b) We see that energy conservation between PNN1 and the verlet integrator is comparable (TE: total energy) (c) Forward and reverse trajectories generated by PNN1 show good reversibility (d) A visualization of PNN model 1 found by the genetic algorithm, attempting to predict positions and velocities one step ahead

Having established that the PNNs learn the physics of the system and result in stable and accurate integrators, we now explore their interpretability in the hope of finding out how time-reversibility and energy conservation are achieved. In short: can the PNNs teach us what they learned? We find that the PNNs discover simple models, with many weights taking fixed values (including zero) and all activations functions taking the simplest possible alternative (linear functions). As an example, the parameters corresponding to PNN1 are shown in Figure 6.5(d). This simplicity allows us to trivially obtain position and velocity update equations. Equations (6.4-6.7) represent the top three PNNs discovered from the data, rewritten in terms of relevant quantities such as timestep and mass.

$$x(t+\Delta t) = x(t) + v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.4}$$

$$x(t+\Delta t) = x(t) + 1.00005v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.5}$$

$$x(t+\Delta t) = x(t) + 1.00035v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + 1.0007v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.6}$$

Figures 6.6 to 6.7 show other PNNs with comparable (but higher) objective functions.

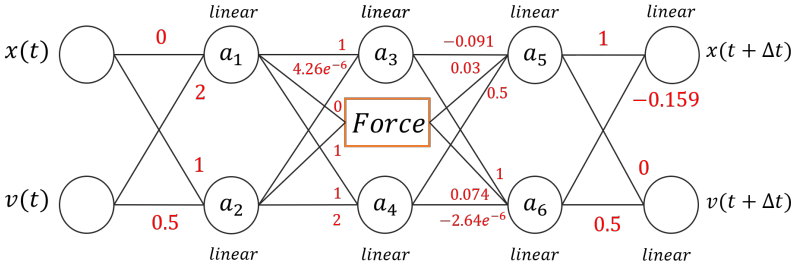

Fig. 6.6. Activation functions and weights for PNN Model 2

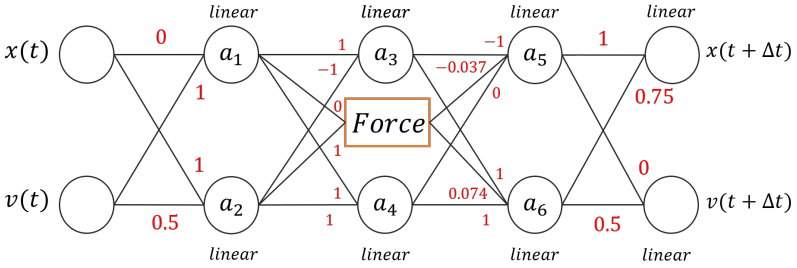

Fig. 6.7. Activation functions and weights for PNN Model 3

The RMSE values for PNN Model 1 on taking one step for the position and velocity are[-7] (1.88x10^{-7}, 7.05x10^{-7}), (1.89x10^{-7}, 7.12x10^{-7}), and (1.50x10^{-7}, 4.73x10^{-7}) respectively for the training, validation and testing set. The RMSE values for PNN Model

2 are ($1.87\text{x}10^{-7}$, $6.92\text{x}10^{-7}$), ($1.88\text{x}10^{-7}$, $7.07\text{x}10^{-7}$), and ($1.51\text{x}10^{-7}$, $5.44\text{x}10^{-7}$), while the RMSE values for PNN Model 3 are ($1.88\text{x}10^{-7}$, $8.25\text{x}10^{-7}$), ($1.88\text{x}10^{-7}$, $8.34\text{x}10^{-7}$), and ($1.51\text{x}10^{-7}$, $6.81\text{x}10^{-7}$).

Inspecting Figure 6.5(d) and Eq. (6.4) we find that PNN1 achieves time-reversibility by evaluating the force at the midpoint between inputs and outputs, this central force evaluation is key to many advanced numerical methods. In fact, PNN1 represents the position Verlet algorithm [35] except that the NN training makes an error in the mass of approximately 3 in 10,000. This algorithm is both reversible and symplectic, i.e. it conserves volume in phase space. The small error in mass actually seems to originate from the small inaccuracies of the force sub-net to describe the Lennard-Jones potential. The PNN models attempt to correct for the errors in the force sub-net, which could be the reason for the PNN models to learn the Verlet integration scheme with marginally different coefficients. Figure 6.8 shows the energy conservation for the Verlet integration scheme (using the force sub-net) and PNN 1. The Verlet network consists of weights corresponding to the position Verlet integration scheme, and we find that the network predicts a total energy slightly different from the true total energy, the discrepancy originating from an imperfect force sub-net. PNN 1 attempts to correct this discrepancy, resulting in better predictions of the total energy.

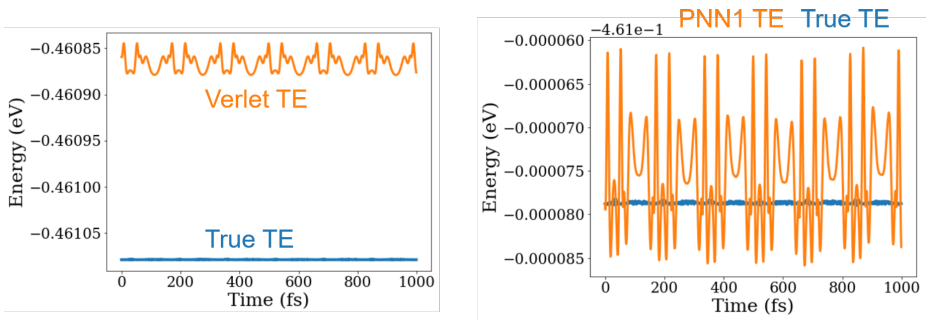

Fig. 6.8. Energy conservation for the Verlet model compared to PNN 1. While both networks conserve energy, PNN 1 attempts to correct for the discrepancy in the force sub-net, predicting the absolute value of the total energy more accurately

PNN2 and 3 are similarly interpretable and, quite remarkably, they also learn to evaluate the force at the half-step. The main difference between these networks and PNN 1 is that a few additional parameters remain adjustable and thus decrease the fitness value, learning a slightly inaccurate version of the position Verlet algorithm with minor energy drifts due to the slight asymmetry in effective mass in the position and velocity update equations.

6.2.5 Importance of parsimony in discovering equations from data

Having demonstrated that adding a force model to a standard feed forward network is insufficient to discover an accurate integration scheme, we will now demonstrate the importance of parsimony to discover an accurate scheme. We use the generic neural network from Figure 6.3 and attempt to learn the force sub-net, keeping some weights fixed and some trainable. Specifically, knowing the weights corresponding to a Verlet integrator, any weight that would be designated as 'fixed' by the genetic algorithm for this network (weights that have values 0, $\frac{1}{2}$, 1, 2, $\frac{\Delta t}{2}$, Δt, and 2Δt) are kept fixed at the values corresponding to the Verlet network, while any weight that is deemed trainable is trained. Our attempt here is to simulate a situation where the genetic algorithm has determined some weights to be fixed, and we wish to understand if the network can learn the force sub-net, and the remaining integrator weights directly from the data. We find that the discovered PNN learns a force sub-net that is a scaling factor times the true force, compensating for this scaling factor by learning the appropriate trainable weight in the integration. This is expected since a raw particle trajectory only provides information about acceleration (force/mass) and not the force and mass individually. We make an error of ~3% in the acceleration learnt by the network, recovering an accurate integration scheme. This shows that by enforcing parsimony, the network can learn the force sub-net to match the training data.

6.2.6 Discovering integration schemes for damped systems

Along similar lines, we tested the ability of the PNNs to discover the physics governing a damped dynamical system. The equations for the top two PNNs, with γ the damping constant, are:

$$x(t+\Delta t) = x(t) + v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.7}$$

$$x(t+\Delta t) = x(t) + 1.00005v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.8}$$

$$x(t+\Delta t) = x(t) + 1.00035v(t)\frac{\Delta t}{2} + \frac{1}{2}f(x(t) + 1.0007v(t)\frac{\Delta t}{2})\frac{\Delta t^2}{1.0005m} \tag{6.9}$$

The PNNs learn classical mechanics, the idea that the frictional force is proportional to negative the velocity, and discover the same stable integrators based on the position Verlet method, all from the observational data.

We consider the emergence of Verlet style integrators from data remarkable. This family of integrators is the preferred choice for molecular dynamics simulations due to their stability. Unlike other algorithms such as the Runge-Kutta family or the first order Euler method, Verlet integrators are symplectic and time reversible [34]. This class of integrators has been long known and proposed independently by several researchers over decades (see Ref. [34] for a review), but a detailed understanding of their properties and their justification from Trotter's theorem are relatively modern [35]. We stress that the equations of motion and an advanced integrator were obtained with only observational data of the motion of a particle and the force-displacement relationship. We believe that, at the expense of computational cost, the force sub-net could be learned together with the integrators (effectively learning the acceleration) from dynamical data. Finally, we note that evolutionary optimization is not the only way to achieve parsimony. For example, one could include hidden layers containing a library of possible activation functions and use sparsity to prune unnecessary activations. This has recently been used to discover simple kinematics equations [206].

An advantage of this approach over ours is simplicity and computational expedience since such networks can be trained using backpropagation alone. However, unlike the evolutionary approach used here, the solution can be sensitive to initialization and the weights of the network may not automatically evolve to interpretable values, finding sub-optimal adjustable values.

6.3 Discovering melting temperature laws from data

To demonstrate the generalizability of PNNs, we now explore the ability of PNNs to discover melting laws from experimental data. Our goal is to predict the melting temperature of materials from fundamental atomic and crystal properties.

6.3.1 Description of training data to discover melting laws

We collected a dataset of experimental melting temperatures for 218 materials (including single elements and perovskites) from the Pymatgen [223] and the Wolfram Alpha [224] repositories respectively. For each of these materials, we collected fundamental material properties such as the bulk modulus K, shear modulus G, volume per atom V_{at}, and density ρ by querying the Materials Project repository [7]. We then compute the average sound speed v_m in the material by averaging the S-wave v_s and P-wave v_p velocities for the material:

$$\begin{aligned} v_s &= \sqrt{\frac{G}{\rho}} \\ v_p &= \sqrt{\frac{K + \frac{4G}{3}}{\rho}} \end{aligned} \tag{6.10}$$

$$v_m = \left[\frac{3}{(\frac{1}{v_p})^3 + (2\frac{1}{v_s})^3} \right]^{1/3} \tag{6.11}$$

We also compute $a = V_{at}^{1/3}$ as an effective interatomic distance, and m to be the mean mass of an atom in the material. Using these quantities, we define four quantities with dimensions of temperature, as follows:

$$\begin{aligned} \theta_0 &= \frac{\hbar v_m}{k_b a} \\ \theta_1 &= \frac{\hbar^2}{m a^2 k_b} \\ \theta_2 &= \frac{a^3 G}{k_b} \\ \theta_3 &= \frac{a^3 K}{k_b} \end{aligned} \tag{6.12}$$

where $\hbar$ is the Planck's constant h divided by 2π, and k_b is the Boltzmann's constant. We then normalize all inputs and the output (experimental melting temperature) by θ_0, giving us three dimensionless inputs:

$$\begin{aligned} \theta_1' &= \frac{\hbar}{m a v_m} \\ \theta_2' &= \frac{a^4 G}{\hbar v_m} \\ \theta_3' &= \frac{a^4 K}{\hbar v_m} \end{aligned} \tag{6.13}$$

6.3.2 Parsimonious neural network architecture to discover melting laws

We now design a parsimonious neural network with three inputs θ_1', θ_2', θ_3', one hidden layer with three neurons, and one output, the melting temperature of the material. Each neuron can now take the linear, squared, multiply, inverse, and hyperbolic tangent (tanh) activations. The multiply and inverse activation functions allow for additional complexity in the discovered laws via product and inverse terms. The weights can again be fixed or trainable, with the fixed category allowing weight values of 0 and 1. Unlike discovering integration schemes, the timestep Δt is not a relevant quantity anymore, so we do not consider this in our list of fixed weights. This flexibility highlights the ability of PNNs to favor quantities relevant to the un-

derlying physics in comparison to other fixed quantities, as well as arbitrary weights obtained in standard neural network training. Our network thus has twelve weights (each with three possible settings) and four activation functions to optimize. The objective function is the same as Eq (6.1), with Eqs. (6.2) and (6.3) now modified to

$$f_1(E_{test}) = log_{10}(E_{test} + 1) \tag{6.14}$$

$$f_2(w_j) = \begin{cases} 0 & w_j = 0 \\ 1 & w_j = 1 \\ 2 & w_j = trainable \end{cases} \tag{6.15}$$

The scores assigned to the linear, squared, multiply, inverse and tanh activations are 0, 1, 2, 3, and 4 respectively.

Varying the parsimony of the objective function, Eq. (6.1), we obtain a family of melting laws, see Figure 6.9, that represent a tradeoff between complexity (defined as the sum of the second and third terms of Eq. (6.1), i.e., the sum of the activation function term and weight term) and accuracy.

We find extremely simple laws such as:

$$T_m^{PNNA} = 21.8671\theta_0 \tag{6.16}$$

$$T_m^{PNNB} = 13.9654\theta_0 + 0.003015\theta_2 \tag{6.17}$$

Most interestingly, we also find the celebrated Lindemann melting law, which describes the melting temperature of a material as:

$$T_m^{lind} = \frac{k_b}{9\hbar^2} f^2 a^2 m T_D^2 = C\frac{\theta_0^2}{\theta_1} \tag{6.18}$$

Where T_D is the debye temperature of the material and f (equivalently C) is an empirical constant.

We also find laws more complex than Lindemann that are more accurate:

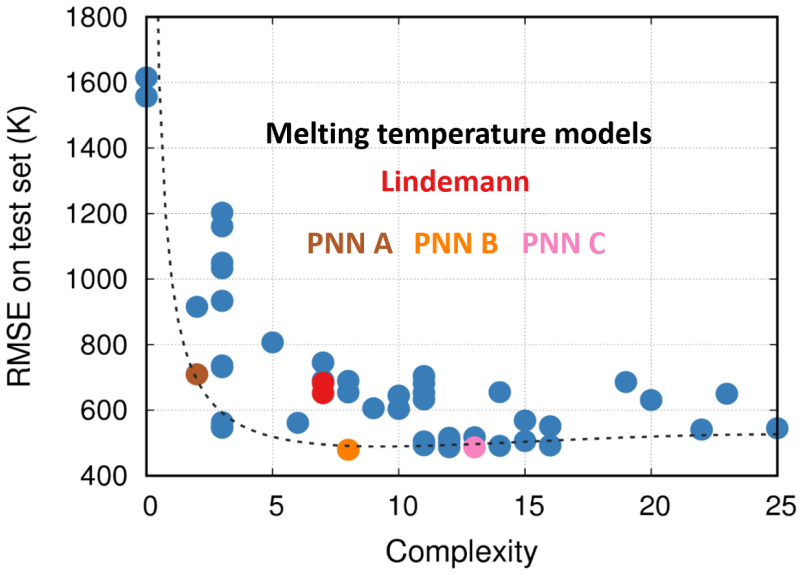

Fig. 6.9. Melting laws discovered by PNNs. The red points show the celebrated Lindemann law, while the blue points show other models discovered. The black dotted line denotes the pareto front of models, with some of the models performing better than the Lindemann law while also being simpler. Three models are highlighted and labeled

$$T_m^{PNNC} = 14.4216\theta_0 + 0.003286\theta_2 + \frac{1.24119\theta_0^2}{\theta_1 - 0.00326\theta_3} \tag{6.19}$$

The equations discovered above can be obtained by interpreting the PNNs, as shown in the figures below:

PNN A attempts to describe the melting temperature as a linear function of the debye temperature, and we find this to be the simplest expression apart from the expression $T_m = 0$. PNNs B and C attempt to improve on this model by adding a correction term that is linear with the shear modulus. PNN C additionally also uses a third correction term that is more complex and warrants further investigation. Figure 6.9 shows that the PNN methodology can not only find existing laws directly from data, but also find new, simple laws that better describe the data.

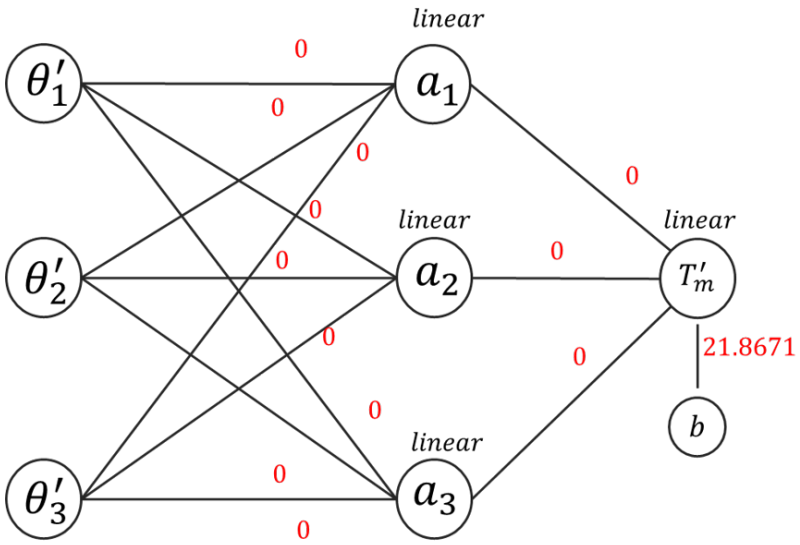

Fig. 6.10. Activation functions and weights for PNN Model A

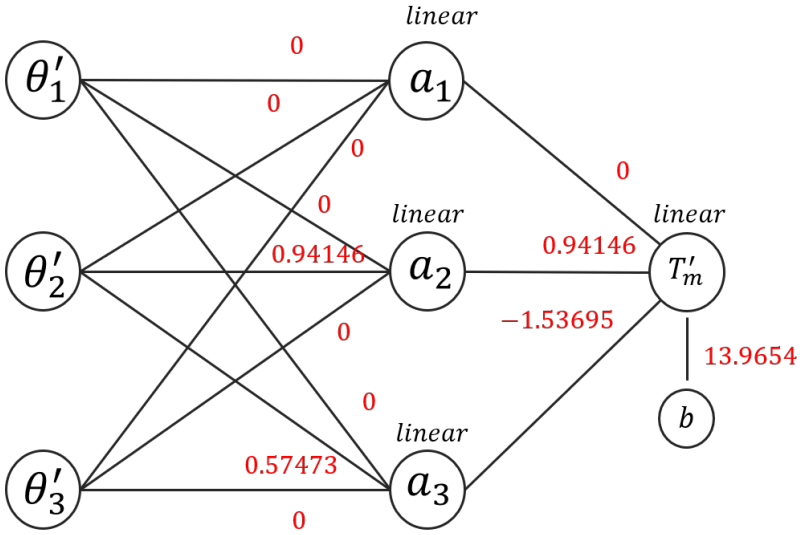

Fig. 6.11. Activation functions and weights for PNN Model B

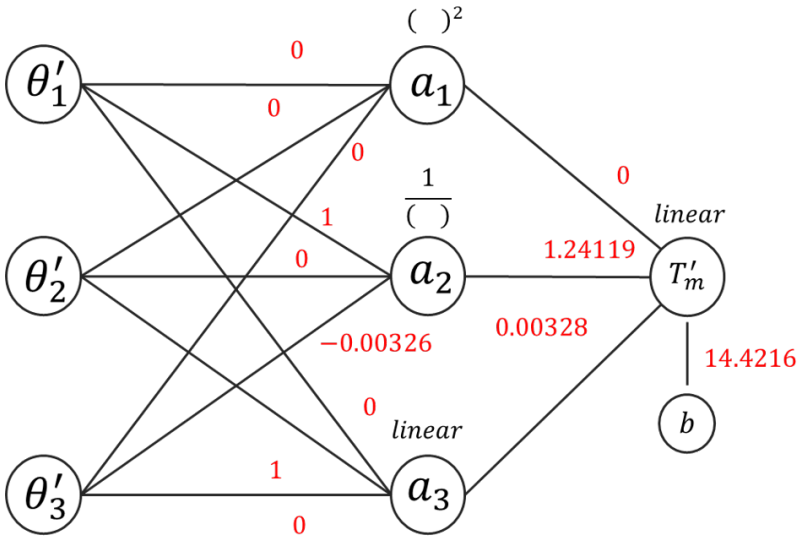

Fig. 6.12. Activation functions and weights for PNN Model C

6.4 Conclusions

In summary, we have shown that parsimonious neural networks are capable of learning interpretable physics models from data and extract underlying symmetries in the problem at hand. From data describing the classical evolution of a particle in an external potential, the PNN produces integration schemes that are accurate, conserve energy and satisfy time reversibility . Furthermore, they can be easily interpretable as discrete versions of Newton's equations of motion. Quite interestingly, the PNNs learn the non-trivial need to evaluate the force at the half step for time reversibility. The optimization could have learned the first order Runge-Kutta algorithm, which is not reversible, but it favored central-difference based integrators. We note that other high-order integrators are not compatible with our initial network, but these can easily be incorporated by starting with a more complex network. As discussed above, the resulting algorithms would not come as a surprise to experts in molecular dynamics simulations as this community has developed, over decades, accurate algorithms to integrate Newton's equations of motion. The fact that such knowledge

and algorithms can be extracted automatically from observational data has, however, deep implications if the approach presented is successful in other problems and fields. This is confirmed with a second example that shows the ability of PNNs to extract melting laws from experimental data. We discover a family of expressions that are pareto optimal in terms of accuracy vs. parsimony, our results show that the widely used Lindemann law, proposed in 1910, is remarkably close to the front but we find PNNs that outperform it.

7. CONCLUSIONS

7.1 Summary of current work

This thesis describes methods of using and improving molecular dynamics simulations to understand materials phenomena and aid the design of next generation materials. Chapter 3 discusses how molecular dynamics simulations can be used to understand how shape memory alloy properties such as thermal hysteresis and transformation temperatures are modified when a coherent 2^{nd} phase is added. We find that we can increase martensite stability and transformation temperature by incorporating a soft second phase with near zero lattice mismatch to the martensite phase. This finding, along with other trends we find, will aid the design of lightweight room-temperature shape memory alloys, whose transformation temperature is currently too low for practical use. Chapter 4 discusses how kinetic Monte Carlo methods can be incorporated in MD simulations to extend timescales and predict microstructure evolution in carbon fibers. We developed the MD-CF model which accurately predicted transverse microstructures, as well as predicted the transverse modulus of carbon fibers, a quantity difficult to measure in experiments. We briefly explored the role of processing parameters could affect the transverse microstructure and properties. Chapter 5 explores novel implementations of MD algorithms to achieve greater length and time scales. This resulted in CabanaMD-NNP, an implementation that relies on an existing LAMMPS implementation but leverages novel libraries to achieve cross-platform performance, resulting in massive speedups by the efficient utilization of the GPUs. Chapter 6 describes Parsimonious Neural Networks (PNNs), an ML technique that combines genetic algorithms and neural network training to discover interpretable laws directly from data. We demonstrate this method by discovering integration schemes for molecular dynamics and discovering novel melting laws directly

from data. We find the PNNs to discover the Verlet integration scheme as a simple model that also accurately describes the data. Similarly, the PNNs find the celebrated Lindemann law to describe melting, but also find other simpler descriptions that also describe the data better, effectively discovering new laws directly from data.

7.2 Potential future work

Future work following on Chapter 3 could generalize the trends we see in a metallurgically relevant precipitate geometries, providing specific guidelines for future experiments. Precipitate geometries could be generated using Atomsk [20] and the study could be repeated for various precipitate sizes. This could also be extended to polycrystalline systems, where other relevant variables such as grain size, and the effect of precipitate location with respect to the grain, could affect the transformation characteristics. Ongoing work is currently exploring this for a Ni-Al system.

Another avenue for future work could be the development of an interatomic model for the Mg-Sc alloy, parametrized to low Sc and high Sc regions of the Mg-Sc alloys. Recent work using Density Functional Theory calculations has shown that the addition of pure Mg as a coherent 2^{nd} phase can stabilize the martensite phase by decreasing its energy, thus increasing its stability and increasing the transformation temperature of the alloy. The availability of an Mg-Sc interatomic potential will allow us to predict the transformation temperature accurately, and predict the concentration of Mg required to attain room-temperature martensite phase stability. Ongoing work from collaborators is also exploring an automated search to match base martensitic phases with coherent 2^{nd} phases that can modify transformation characteristics in a desired manner.

Future work following on Chapter 4 would mainly focus on extending MD-CF to generate a three-dimensional atomisitc model of carbon fibers, a task that has proven to be challenging thus far. Other groups following up on our work have recently reported microstructures close to those experimentally observed [139, 140]. While

these works show remarkable success, the microstructures are still generated by short timescale MD evolutions of a basic unit such as a graphitic sheet or a structural unit generated by reactive MD simulations. Increasingly realistic carbon fiber microstructures could be obtained combining the initial structures used in the studies mentioned before, with the MD-CF approach to simulate long timescale microstructure evolution. This improvement is essentially focusing on an improved description of the initial ladder structures which represent the early reactions that occur during carbonization.

Specifically, we could build on work from Saha et al. [113] and Kowalik et al. [225], who have characterized the chemical reactions that occur during the initial stages of carbonization. The work from Saha et al. in particular suggests that ladder structures consisting of six-membered rings with heteroatoms first form five-membered rings and polyyne chains by the elimination of N_2. These structures then fuse to form six-membered rings over time, releasing H_2, with the occasional release of HCN and other gases throughout this process. This seems to suggest that ladder structures with one six-membered ring are unlikely to form. To incorporate this insight, the initial structure for a three dimensional MD-CF model could be a long set of ladder chains that have a width of two or three six-membered rings, similar to the initial structures considered in [139]. The subsequent MD-CF approach could then simulate long-term microstructure evolution.

Another approach that could provide realistic microstructures is to add bond-breaking capabilities in MD-CF. One limitation of the current MD-CF approach is that long ladder structures can be forced into configurations that are potentially unfavorable because of the formation of a few initial bonds between two ladders. In reality, many chemical reactions could allow the ladders to break away from this configuration and bond to another structural element that would lower the energy of the configuration. The addition of a bond-breaking reaction into the MD-CF scheme would allow partially bonded ladder structures to occasionally break the bonds they formed and explore other low-energy possibilities to create bonds. From a kMC standpoint, if we

consider the bond creation reaction as a forward process, the improvement suggested here is the same as specifying a non-zero backward reaction rate.

A full extension of the ideas presented above is indeed considering a wide set of chemical reactions that occur during carbonization, including bond creation and bond breaking, and explicitly including all reactions into the kMC piece of MD-CF. The initial ladder structure could then have heteroatoms in the aromatic rings, and allow for the complex reactions suggested by Saha et al.

Yet another avenue for improved microstructures would be starting from a three-dimensional stabilized structure that is in agreement with experiments. This agreement would not only have to consider the atomic structure of the ladders, as mentioned above, but also the three dimensional arrangement of these ladder structures, including defects, voids, and amorphous regions. These initial structures could be obtained using schemes similar to the ones used in recent work that simulates carbonization [139]. A full extension of this approach would be to consider the generation of ladder structures from precusor fibers, essentially applying the MD-CF idea to the stabilization process. This, in conjunction with MD-CF for carbonization, could be the first end-to-end atomistic model of carbon fiber processing, and provide atomistic insight into the effect of processing variables such as carbonization temperature, heating rate, extent of stretching during graphitization etc, on the mechanical properties of the created fibers. This insight can then be used as a guide to manufacture next generation, high-strength and high-stiffness fibers. It must be noted that an experimental database of carbon fiber microstructures, their associated characteristics such as crystallite length, and the observed mechanical properties, would be crucial for further modeling efforts to conslidate their approaches.

Apart from modeling the fibers, the MD-CF approach, or improved versions of it, could also be used to predict microstructures at the surface of carbon fibers. This would be relevant to understanding the behavior carbon fiber - epoxy matrices in composites and determine the mechanical behavior of these interfaces and the effect of fiber processing and microstructure on the properties of the composite.

Future work following on Chapter 5 will focus on scaling across multiple nodes and GPUs. Demonstrating scaling across the leading supercomputers in the U.S, such as the Summit machine at Oak Ridge National Laboratory, and Sierra at Lawrence Livermore National Laboratory, will be key to understanding the extent of simulations capable with these new algorithms. The strong and weak scaling results could point us to the possibility of near-quantum accurate, state-of-the-art neural network potential MD simulations of a billion atoms or more. The CabanaMD implementation is also limited to the NVE ensemble, and an integration into the LAMMPS software package to provide other thermostat algorithms is critical to expand the range of problems that could be solved with these algorithms. The scalability of these algorithms could be used to investigate scientific problems of interest, such as Li-ion diffusion in electrode materials, as well as recrystallization in phase change materials, problems which would require the availability of appropriate neural network potentials.

Another front of development could be investigating potential speedups in kernels where a bigger fraction of the compute time is spent on evaluating ML models, one example of which is [54]. These kernels are more tightly coupled to scalable libraries such as Tensorflow, and exploring methods to combine Cabana and Kokkos constructs with Tensorflow is an exciting avenue to access even greater length and time scales with neural network potentials.

Future following on Chapter 6 could benchmark the PNN approach against a standard dataset such as the recently introduced Feynman Symbolic Regression Database in Ref. [226]. An ongoing work is the use of PNNs to discover chemical kinetics for high energy materials directly from atomistic trajectory data. In this setup, an encoder-decoder network is trained learn effective low dimensional encodings of the high dimensional, raw atomic environment of each trajectory snapshot. The encoded representations are then passed through a PNN to attempt to detect kinetics laws that can be written in terms of the encoded variables. This approach is similar to other recent work [206] that uses encoder networks to generate input variables for equation learning networks. An interesting future direction here could be to enforce

constraints on the learnt encoded variables. This could be done by using a customized objective function for the encoder network, or by more advanced methods that automatically convert the encoded variables to dimensionless inputs [226]. Similarly, the encoded variables could be simplified based on translational or other symmetries. Future work could also re-implement the custom activation functions in the PNN scheme as Keras custom layers, which would allow easy scalability to include many more activation functions and physical constants as fixed weights.

Ongoing work is also attempting to use PNNs to learn equations that predict the detonation velocity of high-energy materials, where current predictive models could benefit from PNN's automated approach to combining and composing functions beyond human intuition. Each application of PNN will focus on automatically detect equations from data, significantly increasing confidence in the learnt models due to their interpretability.

Another application of machine learning methods to molecular dynamics is the use of recurrent networks to accelerate molecular dynamics simulations. As discussed in this work, MD simulations are limited by the small time steps required to resolve high frequency atomic vibrations accurately. Accelerated integration schemes such as RESPA [35] attempt to separate the force evaluation into a 'fast force' and a 'slow force', where the fast force is computationally cheap and the slow force is relatively expensive. The RESPA scheme then defines an integration scheme where the slow force is evaluated once every 'n' steps, where n is often 4 or 5. Instead of using conventional force models to compute the fast and slow forces, we could use recurrent neural networks to predict a time series of the forces on each atom. The recurrent network could then be used as a fast force, with the conventional force model used as a slow force. This integration can provide speedups of ~4-5x or even greater.

Finally, a continuing push should be made to make both computational and data-science workflows accessible to a wider audience. This could be via creating curated databases of simulation results and by creating new ways of interacting with the databases. An example of this would be to explore a database of Density Functional

Theory calculations to understand convergence criteria for predicting band gaps in semiconductors. Such studies would save valuable time for new users and provide an initial point for further studies.

REFERENCES

REFERENCES

[1] "Mgi white paper, 2011," https://www.mgi.gov/sites/default/files/documents/materials_genome_initiative-final.pdf, accessed: 2020-10-22.

[2] J. Gong, D. Snyder, T. Kozmel, C. Kern, J. E. Saal, I. Berglund, J. Sebastian, and G. Olson, "Icme design of a castable, creep-resistant, single-crystal turbine alloy," *JOM*, vol. 69, no. 5, pp. 880–885, 2017.

[3] J. Allison, M. Li, C. Wolverton, and X. Su, "Virtual aluminum castings: an industrial application of icme," *Jom*, vol. 58, no. 11, pp. 28–35, 2006.

[4] P. V. Balachandran, A. A. Emery, J. E. Gubernatis, T. Lookman, C. Wolverton, and A. Zunger, "Predictions of new ab o 3 perovskite compounds by combining machine learning and density functional theory," *Physical Review Materials*, vol. 2, no. 4, p. 043802, 2018.

[5] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, "Accelerated search for materials with targeted properties by adaptive design," *Nature communications*, vol. 7, no. 1, pp. 1–9, 2016.

[6] A. D. Sendek, E. D. Cubuk, E. R. Antoniuk, G. Cheon, Y. Cui, and E. J. Reed, "Machine learning-assisted discovery of solid li-ion conducting materials," *Chemistry of Materials*, vol. 31, no. 2, pp. 342–352, 2018.

[7] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder *et al.*, "Commentary: The materials project: A materials genome approach to accelerating materials innovation," *Apl Materials*, vol. 1, no. 1, p. 011002, 2013.

[8] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, "Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd)," *Jom*, vol. 65, no. 11, pp. 1501–1509, 2013.

[9] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy *et al.*, "Aflow: an automatic framework for high-throughput materials discovery," *Computational Materials Science*, vol. 58, pp. 218–226, 2012.

[10] The Minerals Metals & Materials Society (TMS), *Building a Materials Data Infrastructure: Opening New Pathways to Discovery and Innovation in Science and Engineering*. Pittsburgh, PA: TMS, 2017. [Online]. Available: dx.doi.org/10.7449/mdistudy_1

[11] B. J. Alder and T. E. Wainwright, "Studies in molecular dynamics. i. general method," *The Journal of Chemical Physics*, vol. 31, no. 2, pp. 459–466, 1959.

[12] ——, "Studies in molecular dynamics. ii. behavior of a small number of elastic spheres," *The Journal of Chemical Physics*, vol. 33, no. 5, pp. 1439–1451, 1960.

[13] A. Rahman, "Correlations in the motion of atoms in liquid argon," *Physical review*, vol. 136, no. 2A, p. A405, 1964.

[14] A. Rahman and F. H. Stillinger, "Molecular dynamics study of liquid water," *The Journal of Chemical Physics*, vol. 55, no. 7, pp. 3336–3359, 1971.

[15] M. Koslowski and A. Strachan, "Uncertainty propagation in a multiscale model of nanocrystalline plasticity," *Reliability Engineering & System Safety*, vol. 96, no. 9, pp. 1161–1170, 2011.

[16] L. A. Zepeda-Ruiz, A. Stukowski, T. Oppelstrup, and V. V. Bulatov, "Probing the limits of metal plasticity with molecular dynamics simulations," *Nature*, vol. 550, no. 7677, pp. 492–495, 2017.

[17] S. Qu, V. Shastry, W. Curtin, and R. E. Miller, "A finite-temperature dynamic coupled atomistic/discrete dislocation method," *Modelling and simulation in materials science and engineering*, vol. 13, no. 7, p. 1101, 2005.

[18] T. D. de la Rubia, H. M. Zbib, T. A. Khraishi, B. D. Wirth, M. Victoria, and M. J. Caturla, "Multiscale modelling of plastic flow localization in irradiated materials," *Nature*, vol. 406, no. 6798, pp. 871–874, 2000.

[19] F. F. Abraham, J. Q. Broughton, N. Bernstein, and E. Kaxiras, "Spanning the continuum to quantum length scales in a dynamic simulation of brittle fracture," *EPL (Europhysics Letters)*, vol. 44, no. 6, p. 783, 1998.

[20] P. Hirel, "Atomsk: a tool for manipulating and converting atomic data files," *Computer Physics Communications*, vol. 197, pp. 212–219, 2015.

[21] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of computational physics*, vol. 117, no. 1, pp. 1–19, 1995.

[22] L. Martínez, R. Andrade, E. G. Birgin, and J. M. Martínez, "Packmol: a package for building initial configurations for molecular dynamics simulations," *Journal of computational chemistry*, vol. 30, no. 13, pp. 2157–2164, 2009.

[23] B. P. Haley, N. Wilson, C. Li, A. Arguelles, E. Jaramillo, and A. Strachan, "Polymer modeler," Aug 2010. [Online]. Available: https://nanohub.org/resources/polymod

[24] D. Stradi, L. Jelver, S. Smidstrup, and K. Stokbro, "Method for determining optimal supercell representation of interfaces," *Journal of Physics: Condensed Matter*, vol. 29, no. 18, p. 185901, 2017.

[25] M. S. Daw and M. I. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Physical Review B*, vol. 29, no. 12, p. 6443, 1984.

[26] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon," *Physical review B*, vol. 31, no. 8, p. 5262, 1985.

[27] S. L. Mayo, B. D. Olafson, and W. A. Goddard, "Dreiding: a generic force field for molecular simulations," *Journal of Physical chemistry*, vol. 94, no. 26, pp. 8897–8909, 1990.

[28] A. C. Van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, "Reaxff: a reactive force field for hydrocarbons," *The Journal of Physical Chemistry A*, vol. 105, no. 41, pp. 9396–9409, 2001.

[29] J. Lutsko, D. Wolf, S. Phillpot, and S. Yip, "Molecular-dynamics study of lattice-defect-nucleated melting in metals using an embedded-atom-method potential," *Physical Review B*, vol. 40, no. 5, p. 2841, 1989.

[30] S. G. Volz and G. Chen, "Molecular-dynamics simulation of thermal conductivity of silicon crystals," *Physical Review B*, vol. 61, no. 4, p. 2651, 2000.

[31] L. Alzate-Vargas, M. E. Fortunato, B. Haley, C. Li, C. M. Colina, and A. Strachan, "Uncertainties in the predictions of thermo-physical properties of thermoplastic polymers via molecular dynamics," *Modelling and Simulation in Materials Science and Engineering*, vol. 26, no. 6, p. 065007, 2018.

[32] A. Strachan, A. C. van Duin, D. Chakraborty, S. Dasgupta, and W. A. Goddard III, "Shock waves in high-energy materials: The initial chemical events in nitramine rdx," *Physical Review Letters*, vol. 91, no. 9, p. 098301, 2003.

[33] L. Verlet, "Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules," *Physical review*, vol. 159, no. 1, p. 98, 1967.

[34] E. Hairer, C. Lubich, G. Wanner *et al.*, "Geometric numerical integration illustrated by the stormer-verlet method," *Acta numerica*, vol. 12, no. 12, pp. 399–450, 2003.

[35] M. Tuckerman, B. J. Berne, and G. J. Martyna, "Reversible multiple time scale molecular dynamics," *The Journal of chemical physics*, vol. 97, no. 3, pp. 1990–2001, 1992.

[36] K. Kadau, T. C. Germann, and P. S. Lomdahl, "Molecular dynamics comes of age: 320 billion atom simulation on bluegene/l," *International Journal of Modern Physics C*, vol. 17, no. 12, pp. 1755–1761, 2006.

[37] R. Salomon-Ferrer, A. W. Gotz, D. Poole, S. Le Grand, and R. C. Walker, "Routine microsecond molecular dynamics simulations with amber on gpus. 2. explicit solvent particle mesh ewald," *Journal of chemical theory and computation*, vol. 9, no. 9, pp. 3878–3888, 2013.

[38] A. F. Voter, "Hyperdynamics: Accelerated molecular dynamics of infrequent events," *Physical Review Letters*, vol. 78, no. 20, p. 3908, 1997.

[39] ——, "Parallel replica method for dynamics of infrequent events," *Physical Review B*, vol. 57, no. 22, p. R13985, 1998.

[40] L. Sandoval, D. Perez, B. P. Uberuaga, and A. F. Voter, "Competing kinetics and he bubble morphology in w," *Physical review letters*, vol. 114, no. 10, p. 105502, 2015.

[41] B. Uberuaga, R. Smith, A. Cleave, F. Montalenti, G. Henkelman, R. Grimes, A. Voter, and K. Sickafus, "Structure and mobility of defects formed from collision cascades in mgo," *Physical review letters*, vol. 92, no. 11, p. 115505, 2004.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[43] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[44] D. Jha, L. Ward, A. Paul, W.-k. Liao, A. Choudhary, C. Wolverton, and A. Agrawal, "Elemnet: Deep learning the chemistry of materials from only elemental composition," *Scientific reports*, vol. 8, no. 1, pp. 1–13, 2018.

[45] E. D. Cubuk, A. D. Sendek, and E. J. Reed, "Screening billions of candidates for solid lithium-ion conductors: A transfer learning approach for small data," *The Journal of chemical physics*, vol. 150, no. 21, p. 214701, 2019.

[46] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *science*, vol. 324, no. 5923, pp. 81–85, 2009.

[47] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," *Science Advances*, vol. 3, no. 4, p. e1602614, 2017.

[48] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," *arXiv preprint arXiv:1711.10561*, 2017.

[49] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 15 379–15 389.

[50] J. Behler and M. Parrinello, "Generalized neural-network representation of high-dimensional potential-energy surfaces," *Physical review letters*, vol. 98, no. 14, p. 146401, 2007.

[51] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, "Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials," *Journal of Computational Physics*, vol. 285, pp. 316–330, 2015.

[52] M. A. Wood and A. P. Thompson, "Extending the accuracy of the snap interatomic potential form," *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241721, 2018.

[53] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, "Schnet–a deep learning architecture for molecules and materials," *The Journal of Chemical Physics*, vol. 148, no. 24, p. 241722, 2018.

[54] H. Wang, L. Zhang, J. Han, and E. Weinan, "Deepmd-kit: A deep learning package for many-body potential energy representation and molecular dynamics," *Computer Physics Communications*, vol. 228, pp. 178–184, 2018.

[55] J. S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A. E. Roitberg, "Less is more: Sampling chemical space with active learning," *The Journal of chemical physics*, vol. 148, no. 24, p. 241733, 2018.

[56] M. A. Wood and A. P. Thompson, "Quantum-accurate molecular dynamics potential for tungsten," *arXiv preprint arXiv:1702.07042*, 2017.

[57] Y. Zuo, C. Chen, X. Li, Z. Deng, Y. Chen, J. Behler, G. Csányi, A. V. Shapeev, A. P. Thompson, M. A. Wood *et al.*, "Performance and cost assessment of machine learning interatomic potentials," *The Journal of Physical Chemistry A*, vol. 124, no. 4, pp. 731–745, 2020.

[58] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic recurrent neural networks," *arXiv preprint arXiv:1909.13334*, 2019.

[59] M. J. Eslamibidgoli, M. Mokhtari, and M. H. Eikerling, "Recurrent neural network-based model for accelerated trajectory analysis in aimd simulations," *arXiv preprint arXiv:1909.10124*, 2019.

[60] J. C. V. Gastelum, A. Strachan, and S. Desai, "Machine learning for materials science: Part 1," Feb 2019. [Online]. Available: https://nanohub.org/resources/mseml

[61] S. Desai, E. Kim, and V. Hegde, "Hands-on deep learning for materials," Jun 2020. [Online]. Available: https://nanohub.org/resources/citrinednn

[62] A. K. M. Kanakkithodi, "Machine learning defect behavior in semiconductors," Nov 2020. [Online]. Available: https://nanohub.org/resources/mldefect

[63] J. C. V. Gastelum and A. Strachan, "Citrine tools for materials informatics," Dec 2019. [Online]. Available: https://nanohub.org/resources/citrinetools

[64] S. Desai and A. Strachan, "Calibration using dakota," Mar 2019. [Online]. Available: https://nanohub.org/resources/dakotacalibrate

[65] B. M. Adams, W. J. Bohnhoff, K. Dalbey, J. Eddy, M. Eldred, D. Gay, K. Haskell, P. D. Hough, and L. P. Swiler, "Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 user's manual," *Sandia National Laboratories, Tech. Rep. SAND2010-2183*, 2009.

[66] A. Sutton and J. Chen, "Long-range finnis–sinclair potentials," *Philosophical Magazine Letters*, vol. 61, no. 3, pp. 139–146, 1990.

[67] S. Desai, M. Hunt, and A. Strachan, "Online tools for uncertainty quantification in nanohub," *JOM*, vol. 71, no. 8, pp. 2635–2645, 2019.

[68] K. Otsuka and X. Ren, "Physical metallurgy of Ti–Ni-based shape memory alloys," *Progress in materials science*, vol. 50, no. 5, pp. 511–678, 2005, publisher: Elsevier.

[69] NASA, "The little tires that could... go to mars," https://www.nasa.gov/feature/glenn/2020/the-little-tires-that-could-go-to-mars, 2020, accessed: 2020-11-02.

[70] J. M. Jani, M. Leary, A. Subic, and M. A. Gibson, "A review of shape memory alloy research, applications and opportunities," *Materials & Design (1980-2015)*, vol. 56, pp. 1078–1113, 2014, publisher: Elsevier.

[71] J. San Juan, M. L. Nó, and C. A. Schuh, "Nanoscale shape-memory alloys for ultrahigh mechanical damping," *Nature nanotechnology*, vol. 4, no. 7, p. 415, 2009, publisher: Nature Publishing Group.

[72] K. Bhattacharya, S. Conti, G. Zanzotto, and J. Zimmer, "Crystal symmetry and the reversibility of martensitic transformations," *Nature*, vol. 428, no. 6978, pp. 55–59, 2004, publisher: Nature Publishing Group.

[73] K. Bhattacharya, *Microstructure of martensite: why it forms and how it gives rise to the shape-memory effect*. Oxford University Press, 2003, vol. 2.

[74] J. Cui, Y. S. Chu, O. O. Famodu, Y. Furuya, J. Hattrick-Simpers, R. D. James, A. Ludwig, S. Thienhaus, M. Wuttig, and Z. Zhang, "Combinatorial search of thermoelastic shape-memory alloys with extremely small hysteresis width," *Nature materials*, vol. 5, no. 4, pp. 286–290, 2006, publisher: Nature Publishing Group.

[75] R. Zarnetta, R. Takahashi, M. L. Young, A. Savan, Y. Furuya, S. Thienhaus, B. Maaß, M. Rahim, J. Frenzel, and H. Brunken, "Identification of quaternary shape memory alloys with near-zero thermal hysteresis and unprecedented functional stability," *Advanced Functional Materials*, vol. 20, no. 12, pp. 1917–1923, 2010, publisher: Wiley Online Library.

[76] D. Xue, P. V. Balachandran, J. Hogden, J. Theiler, D. Xue, and T. Lookman, "Accelerated search for materials with targeted properties by adaptive design," *Nature communications*, vol. 7, no. 1, pp. 1–9, 2016, publisher: Nature Publishing Group.

[77] Y. Ogawa, D. Ando, Y. Sutou, and J. Koike, "A lightweight shape-memory magnesium alloy," *Science*, vol. 353, no. 6297, pp. 368–370, 2016, publisher: American Association for the Advancement of Science.

[78] Y. Ogawa, D. Ando, Y. Sutou, H. Somekawa, and J. Koike, "Martensitic Transformation in a β-Type Mg–Sc Alloy," *Shape Memory and Superelasticity*, vol. 4, no. 1, pp. 167–173, 2018, publisher: Springer.

[79] K. Guda Vishnu and A. Strachan, "Shape memory metamaterials with tunable thermo-mechanical response via hetero-epitaxial integration: A molecular dynamics study," *Journal of Applied Physics*, vol. 113, no. 10, p. 103503, 2013, publisher: American Institute of Physics.

[80] C. Chluba, W. Ge, R. L. de Miranda, J. Strobel, L. Kienle, E. Quandt, and M. Wuttig, "Ultralow-fatigue shape memory alloy films," *Science*, vol. 348, no. 6238, pp. 1004–1007, 2015, publisher: American Association for the Advancement of Science.

[81] D. R. Coughlin, P. J. Phillips, G. S. Bigelow, A. Garg, R. D. Noebe, and M. J. Mills, "Characterization of the microstructure and mechanical properties of a 50.3 Ni–29.7 Ti–20Hf shape memory alloy," *Scripta Materialia*, vol. 67, no. 1, pp. 112–115, 2012, publisher: Elsevier.

[82] Y. Gao, N. Zhou, F. Yang, Y. Cui, L. Kovarik, N. Hatcher, R. Noebe, M. J. Mills, and Y. Wang, "P-phase precipitation and its effect on martensitic transformation in (Ni, Pt) Ti shape memory alloys," *Acta Materialia*, vol. 60, no. 4, pp. 1514–1527, 2012, publisher: Elsevier.

[83] H. L. Wang, Y. L. Hao, S. Y. He, T. Li, J. M. Cairney, Y. D. Wang, Y. Wang, E. G. Obbard, F. Prima, and K. Du, "Elastically confined martensitic transformation at the nano-scale in a multifunctional titanium alloy," *Acta Materialia*, vol. 135, pp. 330–339, 2017, publisher: Elsevier.

[84] S. Pathak, N. Velisavljevic, J. K. Baldwin, M. Jain, S. Zheng, N. A. Mara, and I. J. Beyerlein, "Strong, ductile, and thermally stable bcc-mg nanolaminates," *Scientific reports*, vol. 7, no. 1, pp. 1–9, 2017.

[85] B. Derby, Y. Cui, J. Baldwin, R. Arróyave, M. J. Demkowicz, and A. Misra, "Processing of novel pseudomorphic Cu–Mo hierarchies in thin films," *Materials Research Letters*, vol. 7, no. 1, pp. 1–11, 2019, publisher: Taylor & Francis.

[86] S. T. Reeve, A. Belessiotis-Richards, and A. Strachan, "Harnessing mechanical instabilities at the nanoscale to achieve ultra-low stiffness metals," *Nature communications*, vol. 8, no. 1, pp. 1–7, 2017, publisher: Nature Publishing Group.

[87] S. T. Reeve, K. G. Vishnu, A. Belessiotis-Richards, and A. Strachan, "Tunability of martensitic behavior through coherent nanoprecipitates and other nanostructures," *Acta Materialia*, vol. 154, pp. 295–302, 2018, publisher: Elsevier.

[88] S. T. Reeve, K. G. Vishnu, and A. Strachan, "Uncharacteristic second order martensitic transformation in metals via epitaxial stress fields," *Journal of Applied Physics*, vol. 127, no. 4, p. 045107, 2020, publisher: AIP Publishing LLC.

[89] S. Tripathi, K. G. Vishnu, M. S. Titus, and A. Strachan, "Tunability of martensitic transformation in Mg-Sc shape memory alloys: A DFT study," *Acta Materialia*, vol. 189, pp. 1–9, 2020, publisher: Elsevier.

[90] J. Khalil-Allafi, A. Dlouhy, and G. Eggeler, "Ni4Ti3-precipitation during aging of NiTi shape memory alloys and its influence on martensitic phase transformations," *Acta Materialia*, vol. 50, no. 17, pp. 4255–4274, 2002, publisher: Elsevier.

[91] M. Ieong, B. Doris, J. Kedzierski, K. Rim, and M. Yang, "Silicon device scaling to the sub-10-nm regime," *Science*, vol. 306, no. 5704, pp. 2057–2060, 2004, publisher: American Association for the Advancement of Science.

[92] V. S. Guthikonda and R. S. Elliott, "An effective interaction potential model for the shape memory alloy AuCd," *Continuum Mechanics and Thermodynamics*, vol. 21, no. 4, pp. 269–295, 2009, publisher: Springer.

[93] E. B. Tadmor, R. S. Elliott, J. P. Sethna, R. E. Miller, and C. A. Becker, "The potential of atomistic simulations and the knowledgebase of interatomic models," *Jom*, vol. 63, no. 7, p. 17, 2011, publisher: Springer Science & Business Media.

[94] K. R. Morrison, M. J. Cherukara, K. G. Vishnu, and A. Strachan, "Role of atomic variability and mechanical constraints on the martensitic phase transformation of a model disordered shape memory alloy via molecular dynamics," *Acta materialia*, vol. 69, pp. 30–36, 2014, publisher: Elsevier.

[95] F. H. Streitz, J. N. Glosli, and M. V. Patel, "Beyond finite-size scaling in solidification simulations," *Physical review letters*, vol. 96, no. 22, p. 225701, 2006, publisher: APS.

[96] E. M. Lifshitz and L. P. Pitaevskii, *Statistical physics: theory of the condensed state*. Elsevier, 2013, vol. 9.

[97] A. P. Thompson, S. J. Plimpton, and W. Mattson, "General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions," *The Journal of chemical physics*, vol. 131, no. 15, p. 154107, 2009, publisher: American Institute of Physics.

[98] C. Jarzynski, "Nonequilibrium equality for free energy differences," *Physical Review Letters*, vol. 78, no. 14, p. 2690, 1997, publisher: APS.

[99] S. L. Mayo, B. D. Olafson, and W. A. Goddard, "DREIDING: a generic force field for molecular simulations," *Journal of Physical chemistry*, vol. 94, no. 26, pp. 8897–8909, 1990, publisher: ACS Publications.

[100] J. Buschbeck, I. Opahle, M. Richter, U. Rößler, P. Klaer, M. Kallmayer, H. Elmers, G. Jakob, L. Schultz, and S. Fähler, "Full tunability of strain along the fcc-bcc bain path in epitaxial films and consequences for magnetic properties," *Physical review letters*, vol. 103, no. 21, p. 216101, 2009.

[101] A. Arumbakkam, E. Davidson, and A. Strachan, "Heteroepitaxial integration of metallic nanowires: transition from coherent to defective interfaces via molecular dynamics," *Nanotechnology*, vol. 18, no. 34, p. 345705, 2007, publisher: IOP Publishing.

[102] P. Morgan, *Carbon fibers and their composites*. CRC press, 2005.

[103] X. Huang, "Fabrication and properties of carbon fibers," *Materials*, vol. 2, no. 4, pp. 2369–2403, 2009, publisher: Molecular Diversity Preservation International.

[104] H. G. Chae and S. Kumar, "Making strong fibers," *Science*, vol. 319, no. 5865, pp. 908–909, 2008, publisher: American Association for the Advancement of Science.

[105] M. MInus and S. Kumar, "The processing, properties, and structure of carbon fibers," *Jom*, vol. 57, no. 2, pp. 52–58, 2005, publisher: Springer.

[106] H. G. Chae, B. A. Newcomb, P. V. Gulgunje, Y. Liu, K. K. Gupta, M. G. Kamath, K. M. Lyons, S. Ghoshal, C. Pramanik, and L. Giannuzzi, "High strength and high modulus carbon fibers," *Carbon*, vol. 93, pp. 81–87, 2015, publisher: Elsevier.

[107] S. C. Bennett, D. J. Johnson, and W. Johnson, "Strength-structure relationships in PAN-based carbon fibres," *Journal of materials science*, vol. 18, no. 11, pp. 3337–3347, 1983, publisher: Springer.

[108] J. A. Hugo, V. A. Phillips, and B. W. Roberts, "Intimate structure of high modulus carbon fibres," *Nature*, vol. 226, no. 5241, pp. 144–144, 1970, publisher: Springer.

[109] M. Guigon, A. Oberlin, and G. Desarmot, "Microtexture and structure of some high-modulus, PAN-base carbon fibres," *Fibre Science and Technology*, vol. 20, no. 3, pp. 177–198, 1984, publisher: Elsevier.

[110] S. C. Bennett, "Strength-structure relationships in carbon fibers," PhD Thesis, University of Leeds, 1976.

[111] S. Kumar, D. P. Anderson, and A. S. Crasto, "Carbon fibre compressive strength and its dependence on structure and morphology," *Journal of materials science*, vol. 28, no. 2, pp. 423–439, 1993, publisher: Springer.

[112] D. J. Johnson and C. N. Tyson, "The fine structure of graphitized fibres," *Journal of Physics D: Applied Physics*, vol. 2, no. 6, p. 787, 1969, publisher: IOP Publishing.

[113] B. Saha and G. C. Schatz, "Carbonization in polyacrylonitrile (PAN) based carbon fibers studied by ReaxFF molecular dynamics simulations," *The Journal of Physical Chemistry B*, vol. 116, no. 15, pp. 4684–4692, 2012, publisher: ACS Publications.

[114] N. Gupta, V. I. Artyukhov, E. S. Penev, and B. I. Yakobson, "Carbonization with Misfusion: Fundamental Limits of Carbon-Fiber Strength Revisited," *Advanced Materials*, vol. 28, no. 46, pp. 10 317–10 322, 2016, publisher: Wiley Online Library.

[115] E. S. Penev, V. I. Artyukhov, and B. I. Yakobson, "Basic structural units in carbon fibers: Atomistic models and tensile behavior," *Carbon*, vol. 85, pp. 72–78, 2015, publisher: Elsevier.

[116] A. Ito and S. Okamoto, "Using molecular dynamics to assess mechanical properties of PAN-based carbon fibers comprising imperfect crystals with amorphous structures," *Int J Mech Aero Ind Mechatron Eng*, vol. 7, pp. 771–6, 2013.

[117] C. Zhu, X. Liu, X. Yu, N. Zhao, J. Liu, and J. Xu, "A small-angle X-ray scattering study and molecular dynamics simulation of microvoid evolution during the tensile deformation of carbon fibers," *Carbon*, vol. 50, no. 1, pp. 235–243, 2012, publisher: Elsevier.

[118] M. S. A. Rahaman, A. F. Ismail, and A. Mustafa, "A review of heat treatment on polyacrylonitrile fiber," *Polymer degradation and Stability*, vol. 92, no. 8, pp. 1421–1432, 2007, publisher: Elsevier.

[119] K. Morita, Y. Murata, A. Ishitani, K. Murayama, T. Ono, and A. Nakajima, "Characterization of commercially available PAN (polyacrylonitri1e)-based carbon fibers," *Pure and Applied Chemistry*, vol. 58, no. 3, pp. 455–468, 1986, publisher: De Gruyter.

[120] C. Li and A. Strachan, "Molecular simulations of crosslinking process of thermosetting polymers," *Polymer*, vol. 51, no. 25, pp. 6058–6070, 2010, publisher: Elsevier.

[121] V. Varshney, S. S. Patnaik, A. K. Roy, and B. L. Farmer, "A molecular dynamics study of epoxy-based networks: cross-linking procedure and prediction of molecular and material properties," *Macromolecules*, vol. 41, no. 18, pp. 6837–6842, 2008, publisher: ACS Publications.

[122] C. Li and A. Strachan, "Molecular scale simulations on thermoset polymers: A review," *Journal of Polymer Science Part B: Polymer Physics*, vol. 53, no. 2, pp. 103–122, 2015, publisher: Wiley Online Library.

[123] G. H. Vineyard, "Frequency factors and isotope effects in solid state rate processes," *Journal of Physics and Chemistry of Solids*, vol. 3, no. 1-2, pp. 121–127, 1957, publisher: Elsevier.

[124] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," *Journal of computational physics*, vol. 22, no. 4, pp. 403–434, 1976, publisher: Elsevier Science.

[125] S. Nosé, "A unified formulation of the constant temperature molecular dynamics methods," *The Journal of chemical physics*, vol. 81, no. 1, pp. 511–519, 1984, publisher: American Institute of Physics.

[126] W. G. Hoover, "Canonical dynamics: Equilibrium phase-space distributions," *Physical review A*, vol. 31, no. 3, p. 1695, 1985.

[127] M. Parrinello and A. Rahman, "Polymorphic transitions in single crystals: A new molecular dynamics method," *Journal of Applied physics*, vol. 52, no. 12, pp. 7182–7190, 1981, publisher: American Institute of Physics.

[128] G. J. Martyna, D. J. Tobias, and M. L. Klein, "Constant pressure molecular dynamics algorithms," *The Journal of chemical physics*, vol. 101, no. 5, pp. 4177–4189, 1994, publisher: American Institute of Physics.

[129] C. Li and A. Strachan, "Molecular dynamics predictions of thermal and mechanical properties of thermoset polymer EPON862/DETDA," *Polymer*, vol. 52, no. 13, pp. 2920–2928, 2011, publisher: Elsevier.

[130] S. P. Coleman, D. E. Spearot, and L. Capolungo, "Virtual diffraction analysis of Ni [0 1 0] symmetric tilt grain boundaries," *Modelling and Simulation in Materials Science and Engineering*, vol. 21, no. 5, p. 055020, 2013, publisher: IOP Publishing.

[131] O. L. Blakslee, D. G. Proctor, E. J. Seldin, G. B. Spence, and T. Weng, "Elastic constants of compression-annealed pyrolytic graphite," *Journal of applied physics*, vol. 41, no. 8, pp. 3373–3382, 1970, publisher: American Institute of Physics.

[132] E. J. Seldin and C. W. Nezbeda, "Elastic Constants and Electron-Microscope Observations of Neutron-Irradiated Compression-Annealed Pyrolytic and Single-Crystal Graphite," *Journal of applied physics*, vol. 41, no. 8, pp. 3389–3400, 1970, publisher: American Institute of Physics.

[133] S. Kawabata, "Measurement of the transverse mechanical properties of high-performance fibres," *Journal of the Textile Institute*, vol. 81, no. 4, pp. 432–447, 1990, publisher: Taylor & Francis.

[134] K. Naito, Y. Tanaka, and J.-M. Yang, "Transverse compressive properties of polyacrylonitrile (PAN)-based and pitch-based single carbon fibers," *Carbon*, vol. 118, pp. 168–183, 2017, publisher: Elsevier.

[135] D. W. Hadley, I. M. Ward, and J. Ward, "The transverse compression of anisotropic fibre monofilaments," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 285, no. 1401, pp. 275–286, 1965, publisher: The Royal Society London.

[136] R. Maurin, P. Davies, N. Baral, and C. Baley, "Transverse properties of carbon fibres by nano-indentation and micro-mechanics," *Applied Composite Materials*, vol. 15, no. 2, p. 61, 2008, publisher: Springer.

[137] W. C. Oliver and G. M. Pharr, "An improved technique for determining hardness and elastic modulus using load and displacement sensing indentation experiments," *Journal of materials research*, vol. 7, no. 6, pp. 1564–1583, 1992, publisher: Cambridge University Press.

[138] J. J. Vlassak and W. D. Nix, "Measuring the elastic properties of anisotropic materials by means of indentation experiments," *Journal of the Mechanics and Physics of Solids*, vol. 42, no. 8, pp. 1223–1245, 1994, publisher: Pergamon.

[139] K. Joshi, M. I. Arefev, and L. V. Zhigilei, "Generation and characterization of carbon fiber microstructures by atomistic simulations," *Carbon*, vol. 152, pp. 396–408, 2019.

[140] J. Zhu, Z. Gao, M. Kowalik, K. Joshi, C. M. Ashraf, M. I. Arefev, Y. Schwab, C. Bumgardner, K. Brown, D. E. Burden *et al.*, "Unveiling carbon ring structure formation mechanisms in polyacrylonitrile-derived carbon fibers," *ACS applied materials & interfaces*, vol. 11, no. 45, pp. 42 288–42 297, 2019.

[141] Y. Shibuta, S. Sakane, E. Miyoshi, S. Okita, T. Takaki, and M. Ohno, "Heterogeneity in homogeneous nucleation from billion-atom molecular dynamics simulation of solidification of pure metal," *Nature communications*, vol. 8, no. 1, pp. 1–9, 2017.

[142] Y. Shibuta, S. Sakane, T. Takaki, and M. Ohno, "Submicrometer-scale molecular dynamics simulation of nucleation and solidification from undercooled melt: linkage between empirical interpretation and atomistic nature," *Acta Materialia*, vol. 105, pp. 328–337, 2016.

[143] K.-i. Nomura, R. K. Kalia, A. Nakano, and P. Vashishta, "A scalable parallel algorithm for large-scale reactive force-field molecular dynamics simulations," *Computer Physics Communications*, vol. 178, no. 2, pp. 73–87, 2008.

[144] G. Xiao, M. Ren, and H. Hong, "50 million atoms scale molecular dynamics modelling on a single consumer graphics card," *Advances in Engineering Software*, vol. 124, pp. 66–72, 2018.

[145] M. J. Harvey, G. Giupponi, and G. D. Fabritiis, "Acemd: accelerating biomolecular dynamics in the microsecond time scale," *Journal of chemical theory and computation*, vol. 5, no. 6, pp. 1632–1639, 2009.

[146] J. A. Anderson, C. D. Lorenz, and A. Travesset, "General purpose molecular dynamics simulations fully implemented on graphics processing units," *Journal of computational physics*, vol. 227, no. 10, pp. 5342–5359, 2008.

[147] J. Glaser, T. D. Nguyen, J. A. Anderson, P. Lui, F. Spiga, J. A. Millan, D. C. Morse, and S. C. Glotzer, "Strong scaling of general-purpose molecular dynamics simulations on gpus," *Computer Physics Communications*, vol. 192, pp. 97–107, 2015.

[148] Q. Hou, M. Li, Y. Zhou, J. Cui, Z. Cui, and J. Wang, "Molecular dynamics simulations with many-body potentials on multiple gpus—the implementation, package and performance," *Computer Physics Communications*, vol. 184, no. 9, pp. 2091–2101, 2013.

[149] S. Le Grand, A. W. Götz, and R. C. Walker, "Spfp: Speed without compromise—a mixed precision model for gpu accelerated molecular dynamics simulations," *Computer Physics Communications*, vol. 184, no. 2, pp. 374–380, 2013.

[150] A. Sedova, J. D. Eblen, R. Budiardja, A. Tharrington, and J. C. Smith, "High-performance molecular dynamics simulation for biological and materials sciences: challenges of performance portability," in *2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*. IEEE, 2018, pp. 1–13.

[151] G. C. Sosso, G. Miceli, S. Caravati, J. Behler, and M. Bernasconi, "Neural network interatomic potential for the phase change material gete," *Physical Review B*, vol. 85, no. 17, p. 174103, 2012.

[152] W. Li, Y. Ando, E. Minamitani, and S. Watanabe, "Study of li atom diffusion in amorphous li3po4 with neural network potential," *The Journal of chemical physics*, vol. 147, no. 21, p. 214106, 2017.

[153] R. F. Service, "Design for us exascale computer takes shape," 2018.

[154] F. Alexander, A. Almgren, J. Bell, A. Bhattacharjee, J. Chen, P. Colella, D. Daniel, J. DeSlippe, L. Diachin, E. Draeger *et al.*, "Exascale applications: skin in the game," *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2166, p. 20190056, 2020.

[155] P. Messina, "The exascale computing project," *Computing in Science & Engineering*, vol. 19, no. 3, pp. 63–67, 2017.

[156] N. Bock, C. F. Negre, S. M. Mniszewski, J. Mohd-Yusof, B. Aradi, J.-L. Fattebert, D. Osei-Kuffuor, T. C. Germann, and A. M. Niklasson, "The basic matrix library (bml) for quantum chemistry," *The Journal of Supercomputing*, vol. 74, no. 11, pp. 6201–6219, 2018.

[157] S. Slattery, C. Junghans, D. Lebrun-Grandie, R. Halver, G. Chen, S. Reeve, A. Scheinberg, C. Smith, and R. Bird, "Ecp-copa/cabana: Cabana version 0.2.0," 2019.

[158] H. C. Edwards, C. R. Trott, and D. Sunderland, "Kokkos: Enabling many-core performance portability through polymorphic memory access patterns," *Journal of Parallel and Distributed Computing*, vol. 74, no. 12, pp. 3202–3216, 2014.

[159] A. Singraber, J. Behler, and C. Dellago, "Library-based lammps implementation of high-dimensional neural network potentials," *Journal of chemical theory and computation*, vol. 15, no. 3, pp. 1827–1840, 2019.

[160] H. Eshet, R. Z. Khaliullin, T. D. Kühne, J. Behler, and M. Parrinello, "Ab initio quality neural-network potential for sodium," *Physical Review B*, vol. 81, no. 18, p. 184107, 2010.

[161] R. Z. Khaliullin, H. Eshet, T. D. Kühne, J. Behler, and M. Parrinello, "Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface," *Physical Review B*, vol. 81, no. 10, p. 100103, 2010.

[162] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, "Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons," *Physical review letters*, vol. 104, no. 13, p. 136403, 2010.

[163] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," *Multiscale Modeling & Simulation*, vol. 14, no. 3, pp. 1153–1173, 2016.

[164] R. K. Lindsey, L. E. Fried, and N. Goldman, "Chimes: a force matched potential with explicit three-body interactions for molten carbon," *Journal of chemical theory and computation*, vol. 13, no. 12, pp. 6222–6229, 2017.

[165] L. Zhang, J. Han, H. Wang, R. Car, and E. Weinan, "Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics," *Physical review letters*, vol. 120, no. 14, p. 143001, 2018.

[166] R. Zubatyuk, J. S. Smith, J. Leszczynski, and O. Isayev, "Accurate and transferable multitask prediction of chemical properties with an atoms-in-molecules neural network," *Science advances*, vol. 5, no. 8, p. eaav6490, 2019.

[167] H. Chan, B. Narayanan, M. J. Cherukara, F. G. Sen, K. Sasikumar, S. K. Gray, M. K. Chan, and S. K. Sankaranarayanan, "Machine learning classical interatomic potentials for molecular dynamics from first-principles training data," *The Journal of Physical Chemistry C*, vol. 123, no. 12, pp. 6941–6957, 2019.

[168] C. Chen, Z. Deng, R. Tran, H. Tang, I.-H. Chu, and S. P. Ong, "Accurate force field for molybdenum by machine learning large materials data," *Physical Review Materials*, vol. 1, no. 4, p. 043603, 2017.

[169] J. S. Smith, B. T. Nebgen, R. Zubatyuk, N. Lubbers, C. Devereux, K. Barros, S. Tretiak, O. Isayev, and A. E. Roitberg, "Approaching coupled cluster accuracy with a general-purpose neural network potential through transfer learning," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[170] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, "wacsf—weighted atom-centered symmetry functions as descriptors in machine learning potentials," *The Journal of chemical physics*, vol. 148, no. 24, p. 241709, 2018.

[171] A. Singraber, "Compphysvienna/n2p2: Neural network potential package 1.0.0," 2018.

[172] C. C. Y. Zuo, S.P. Ong, "Benchmark suite for machine learning interatomic potentials for materials: Materials virtual lab," 2019.

[173] M. Orsi, "Comparative assessment of the elba coarse-grained model for water," *Molecular Physics*, vol. 112, no. 11, pp. 1566–1576, 2014.

[174] S. Páll and B. Hess, "A flexible algorithm for calculating pair interactions on simd architectures," *Computer Physics Communications*, vol. 184, no. 12, pp. 2641–2650, 2013.

[175] C. M. Mangiardi and R. Meyer, "A hybrid algorithm for parallel molecular dynamics simulations," *Computer Physics Communications*, vol. 219, pp. 196–208, 2017.

[176] N. Artrith and A. Urban, "An implementation of artificial neural-network potentials for atomistic materials simulations: Performance for tio2," *Computational Materials Science*, vol. 114, pp. 135–150, 2016.

[177] J. S. Smith, O. Isayev, and A. E. Roitberg, "Ani-1: an extensible neural network potential with dft accuracy at force field computational cost," *Chemical science*, vol. 8, no. 4, pp. 3192–3203, 2017.

[178] K. Yao, J. E. Herr, D. W. Toth, R. Mckintyre, and J. Parkhill, "The tensormol-0.1 model chemistry: a neural network augmented with long-range physics," *Chemical science*, vol. 9, no. 8, pp. 2261–2269, 2018.

[179] K. Lee, D. Yoo, W. Jeong, and S. Han, "Simple-nn: An efficient package for training and executing neural-network interatomic potentials," *Computer Physics Communications*, vol. 242, pp. 95–103, 2019.

[180] Y. Huang, J. Kang, W. A. Goddard III, and L.-W. Wang, "Density functional theory based neural network force fields from energy decompositions," *Physical Review B*, vol. 99, no. 6, p. 064103, 2019.

[181] R. Lot, F. Pellegrini, Y. Shaidu, and E. Küçükbenli, "Panna: Properties from artificial neural network architectures," *Computer Physics Communications*, p. 107402, 2020.

[182] M. Wen and E. B. Tadmor, "Hybrid neural network potential for multilayer graphene," *Physical Review B*, vol. 100, no. 19, p. 195419, 2019.

[183] G. P. Pun, R. Batra, R. Ramprasad, and Y. Mishin, "Physically informed artificial neural networks for atomistic modeling of materials," *Nature communications*, vol. 10, no. 1, pp. 1–10, 2019.

[184] Y. Zhang, C. Hu, and B. Jiang, "Embedded atom neural network potentials: Efficient and accurate machine learning with a physically inspired representation," *The Journal of Physical Chemistry Letters*, vol. 10, no. 17, pp. 4962–4967, 2019.

[185] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[186] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[187] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[188] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, no. 5, pp. 431–434, 2017.

[189] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020.

[190] B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. Doak, A. Thompson, K. Zhang, A. Choudhary, and C. Wolverton, "Combinatorial screening for new materials in unconstrained composition space with machine learning," *Physical Review B*, vol. 89, no. 9, p. 094104, 2014.

[191] J. Carrete, W. Li, N. Mingo, S. Wang, and S. Curtarolo, "Finding unprecedentedly low-thermal-conductivity half-heusler semiconductors via high-throughput materials modeling," *Physical Review X*, vol. 4, no. 1, p. 011019, 2014.

[192] L. Bassman, P. Rajak, R. K. Kalia, A. Nakano, F. Sha, J. Sun, D. J. Singh, M. Aykol, P. Huck, K. Persson *et al.*, "Active learning for accelerated design of layered materials," *npj Computational Materials*, vol. 4, no. 1, pp. 1–9, 2018.

[193] K. Kaufmann, D. Maryanovsky, W. M. Mellor, C. Zhu, A. S. Rosengarten, T. J. Harrington, C. Oses, C. Toher, S. Curtarolo, and K. S. Vecchio, "Discovery of high-entropy ceramics via machine learning," *Npj Computational Materials*, vol. 6, no. 1, pp. 1–9, 2020.

[194] J. C. Snyder, M. Rupp, K. Hansen, K.-R. Müller, and K. Burke, "Finding density functionals with machine learning," *Physical review letters*, vol. 108, no. 25, p. 253002, 2012.

[195] Z. Li, J. R. Kermode, and A. De Vita, "Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces," *Physical review letters*, vol. 114, no. 9, p. 096405, 2015.

[196] T. Jacobsen, M. Jørgensen, and B. Hammer, "On-the-fly machine learning of atomic potential in density functional theory structure optimization," *Physical review letters*, vol. 120, no. 2, p. 026102, 2018.

[197] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *Journal of Fluid Mechanics*, vol. 807, pp. 155–166, 2016.

[198] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 445–22 451, 2019.

[199] S. A. Harp, T. Samad, and A. Guha, "Designing application-specific neural networks using the genetic algorithm," in *Advances in neural information processing systems*, 1990, pp. 447–454.

[200] G. F. Miller, P. M. Todd, and S. U. Hegde, "Designing neural networks using genetic algorithms." in *ICGA*, vol. 89, 1989, pp. 379–384.

[201] S. W. Stepniewski and A. J. Keane, "Pruning backpropagation neural networks using modern stochastic optimisation techniques," *Neural Computing & Applications*, vol. 5, no. 2, pp. 76–98, 1997.

[202] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE transactions on neural networks*, vol. 8, no. 3, pp. 694–713, 1997.

[203] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms." in *IJCAI*, vol. 89, 1989, pp. 762–767.

[204] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," *arXiv preprint arXiv:1712.06567*, 2017.

[205] A. Costa, R. Dangovski, S. Kim, P. Goyal, M. Soljačić, and J. Jacobson, "Interpretable neuroevolutionary models for learning non-differentiable functions and programs," *arXiv preprint arXiv:2007.10784*, 2020.

[206] S. Kim, P. Y. Lu, S. Mukherjee, M. Gilbert, L. Jing, V. Čeperić, and M. Soljačić, "Integration of neural network-based symbolic regression in deep learning for scientific discovery," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[207] R. Iten, T. Metger, H. Wilming, L. Del Rio, and R. Renner, "Discovering physical concepts with neural networks," *Physical Review Letters*, vol. 124, no. 1, p. 010508, 2020.

[208] Z. Long, Y. Lu, and B. Dong, "Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network," *Journal of Computational Physics*, vol. 399, p. 108925, 2019.

[209] G. Martius and C. H. Lampert, "Extrapolation and learning equations," *arXiv preprint arXiv:1610.02995*, 2016.

[210] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, 2018, pp. 6571–6583.

[211] F. A. Lindemann, "Uber die berechnung molekulaner eigenfrequenzen," *Phys. Z*, pp. 609–612, 1910.

[212] S. Desai and A. Strachan, "Discovering discretized classical equations of motion using parsimonious neural networks," Jul 2020. [Online]. Available: https://nanohub.org/resources/pnndemo

[213] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[214] F. Chollet, "Keras," 2015.

[215] H. Yoshida, "Construction of higher order symplectic integrators," *Physics letters A*, vol. 150, no. 5-7, pp. 262–268, 1990.

[216] G. Rowlands, "A numerical algorithm for hamiltonian systems," *Journal of Computational Physics*, vol. 97, no. 1, pp. 235–239, 1991.

[217] R. De Vogelaere, "Methods of integration which preserve the contact transformation property of the hamilton equations," *Technical report (University of Notre Dame. Dept. of Mathematics)*, 1956.

[218] H. F. Trotter, "On the product of semi-groups of operators," *Proceedings of the American Mathematical Society*, vol. 10, no. 4, pp. 545–551, 1959.

[219] J. Behler, R. Martoňák, D. Donadio, and M. Parrinello, "Metadynamics simulations of the high-pressure phases of silicon employing a high-dimensional neural network potential," *Physical review letters*, vol. 100, no. 18, p. 185501, 2008.

[220] S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, "Towards exact molecular dynamics simulations with machine-learned force fields," *Nature communications*, vol. 9, no. 1, pp. 1–10, 2018.

[221] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "Deap: Evolutionary algorithms made easy," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2171–2175, 2012.

[222] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary computation 1: Basic algorithms and operators*. CRC press, 2018.

[223] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python materials genomics (pymatgen): A robust, open-source python library for materials analysis," *Computational Materials Science*, vol. 68, pp. 314–319, 2013.

[224] "Wolfram research inc. mathematica, version 12.0." 2019. [Online]. Available: https://www.wolfram.com/wolfram-alpha-notebook-edition

[225] M. Kowalik, C. Ashraf, B. Damirchi, D. Akbarian, S. Rajabpour, and A. C. Van Duin, "Atomistic scale analysis of the carbonization process for c/h/o/n-based polymers with the reaxff reactive force field," *The Journal of Physical Chemistry B*, vol. 123, no. 25, pp. 5357–5367, 2019.

[226] S.-M. Udrescu and M. Tegmark, "Ai feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, p. eaay2631, 2020.

APPENDIX

A. PUBLISHED WORK

Journal articles:

- Saaketh Desai, Chunyu Li, Tongtong Shen, Alejandro Strachan, "Molecular modeling of the microstructure evolution during carbon fiber processing", Journal of Chemical Physics 147, 224705 (2017)
- Saaketh Desai, Martin Hunt, Alejandro Strachan, "Online tools for uncertainty quantification in nanoHUB", JOM 71 (8) 2635-2645 (2019)
- Saaketh Desai, Sam Reeve, K.G Vishnu, Alejandro Strachan, "Tuning martensitic transformations via coherent second phases in nanolaminates using free energy landscape engineering", Journal of Applied Physics 127, 125112 (2020)
- Saaketh Desai, Sam Reeve, Jim Belak, "Implementing a neural network interatomic model with performance portability for emerging exascale architectures", arXiv preprint arXiv:2002.00054 (2020)
- Saaketh Desai, Alejandro Strachan, "Parsimonious neural networks learn classical mechanics, its underlying symmetries, and an accurate time integrator" arXiv preprint arXiv:2005.11144 (2020)

Software tools:

- Machine Learning for Material Science: Part 1, Juan Carlos Verduzco Gastelum, Alejandro Strachan, Saaketh Desai, `https://nanohub.org/tools/mseml`
- Calibration using DAKOTA: Saaketh Desai, Alejandro Strachan, `https://nanohub.org/resources/dakotacalibrate`