Michigan Technological University

Michigan Technological University
Digital Commons @ Michigan Tech

Dissertations, Master's Theses and Master's Reports

2024

MACHINE LEARNING FOR ELECTRONIC STRUCTURE PREDICTION

Shashank Pathrudkar
Michigan Technological University, sspathru@mtu.edu

Copyright 2024 Shashank Pathrudkar

Recommended Citation

Pathrudkar, Shashank, "MACHINE LEARNING FOR ELECTRONIC STRUCTURE PREDICTION", Open Access Dissertation, Michigan Technological University, 2024.
https://doi.org/10.37099/mtu.dc.etdr/1746

Follow this and additional works at: https://digitalcommons.mtu.edu/etdr

 Part of the Mechanics of Materials Commons, Other Computer Engineering Commons, and the Semiconductor and Optical Materials Commons

MACHINE LEARNING FOR ELECTRONIC STRUCTURE PREDICTION

By

Shashank Pathrudkar

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mechanical Engineering-Engineering Mechanics

MICHIGAN TECHNOLOGICAL UNIVERSITY

2024

© 2024 Shashank Pathrudkar

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Mechanical Engineering-Engineering Mechanics.

Department of Mechanical Engineering-Engineering Mechanics

Dissertation Advisor:	*Dr. Susanta Ghosh*
Committee Member:	*Dr. Amartya Banerjee*
Committee Member:	*Dr. Ranjit Pati*
Committee Member:	*Dr. Soumik Sarkar*
Department Chair:	*Dr. Jason R. Blough*

Dedication

To my parents and grandparents

without whom I would neither be who I am nor I would have been able to contribute to this work.

Contents

List of Figures

List of Tables

Preface

I would like to mention that some portions of this dissertation have been published and some are submitted to peer-reviewed journals.

The contents of Chapter 2 are published in the following journal article: "Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain", Shashank Pathrudkar, Hsuan Ming Yu, Susanta Ghosh, and Amartya S. Banerjee, Phys. Rev. B 105, 195141. DOI:10.1103/PhysRevB.105.195141. APS allows the authors to use the published article or portion of the article in their thesis or dissertation with the following copyright : ©2022 APS. Reprinted with permission from coauthors of the publication. I am grateful to them for allowing me to use the text and figures from the main manuscript and the supplementary material of the article. In this work I worked on the follwoing machine learning aspects: dimensionality reduction using Principal Component Analysis, Neural network mapping, and density based clustering. I would like to thank Dr. Amartya Banerjee's group at UCLA to work on the conceptualization, data generation, postprocessing of the machine learning obtained electron densities and other Density Functional Theory aspects of the work.

The contents of Chapter 3 are uploaded to arxiv and are submitted to npj Computational Materials as: "Electronic structure prediction of multi-million atom systems through uncertainty quantification enabled transfer learning." Shashank Pathrudkar, Ponkrshnan Thiagrajan, Shivang Agarwal, Amartya S. Banerjee, and Susanta Ghosh (2023). arXiv preprint arXiv:2308.13096. Reprinted with permission from coauthors of the publication. I am grateful to them for allowing me to use the text and figures from the main manuscript and the supplementary material of the article. In this work I worked on the following machine learning aspects: developing the overall machine learning framework, developing the descriptors, neural network mapping and generating inference for various material systems as well as million atom systems. I would like to thank Dr. Amartya Banerjee's group at UCLA to work on the conceptualization, data generation, postprocessing of the machine learning obtained electron densities and other Density Functional Theory aspects of the work.

Acknowledgments

I would like to express my sincere gratitude towards my advisor Dr. Susanta Ghosh for letting me pursue my PhD research under his able guidance and inspiring me throughout my journey. His support and constructive criticism has helped me paved my way and overcome the obstacles in this journey. Without his direction it would have been impossible for me to attempt this challenging work.

I am grateful to Dr. Amartya Banerjee, University of California, Los Angeles, Dr. Ranjit Pati, Michigan Technological University, and Dr. Soumik Sarkar, Iowa State University for taking out their time and being a part of my committee. I thank them for their feedback on my work.

I would like to thank the chair of Mechanical Engineering -Engineering Mechanics department at Michigan Technological University, Dr. William Predebon and Dr. Jason Blough for providing all the necessary facilities. I would also like to thank graduate school at Michigan Technological University for the Doctoral Finishing Fellowship to support me during my final semester of Ph.D.

I would like to thank Dr. Amartya S. Banerjee and his research group members at the University of California, Los Angeles for their unwavering dedication, expertise, and the fruitful collaboration that has greatly contributed to the success of my Ph.D.

My Ph.D. journey was fueled by excellent teachers who took the effort to introduce me to fundamental topics essential for my PhD research. I would thank all of them, especially Dr. Susanta Ghosh, Dr. Ibrahim Miskioglu, Dr. Ranjit Pati, Dr. Trisha Sain, Dr. Kui Zhang, Dr. Timothy Havens, and Dr. Anthony Pinar.

I would like to thank my parents, my brother Shailesh and my sister-in-law Revati without whom I would not have been able to reach this point in my life. I am grateful for their support, love, prayers, and sacrifices.

I am blessed with friends and lab mates who have always encouraged me and stood with me in all of my ups and downs in this journey. I would like to thank members of Dr. Susanta Ghosh's lab, Dr. Upendra Yadav, Dr. Ponkrshnan Thiagrajan, and Revanth Mattey for numerous discussions on the topic, which helped me construct this work. I would thank my friends Chinmay Sathe, Rushikesh Kulkarni, Rajat Onkar, Dr. Swapnil Bamane, Dr. Prashik Gaikwad, Vaidehi Karajgikar, Utkarsh Chowdhary, Dr. Priyanka Kadav, Rithvika Iyer, Dr. Nabhjit Goswami, Dr. Chaitanya Bhat and Dr. Sagar Patil to make this ride smooth for me. I would like to express my gratitude towards Shreyas Kulkarni, Aishwarya Unta and Revanth Mattey for their long-time friendship and empathy. I would never be able to thank them enough.

I humbly extend my thanks to all the people who are a part of this journey and pray to receive their continued support.

Abstract

Kohn-Sham density functional theory is the work horse of computational material science research. The core of Kohn-Sham density functional theory, the Kohn-Sham equations, output charge density, energy levels and wavefunctions. In principle, the electron density can be used to obtain several other properties of interest including total potential energy of the system, atomic forces, binding energies and electric constants. In this work we present machine learning models designed to bypass the Kohn-Sham equations by directly predicting electron density. Two distinct models were developed: one tailored to predict electron density for quasi one-dimensional materials under strain, while the other is applicable across a wide array of material systems, with a specific emphasis on metallic and alloy compositions.

The first model applies to important classes of material systems such as nanotubes, for which, tuning the interplay of mechanical deformations and electronic fields — i.e., strain engineering — is an active area of investigation. Using armchair single wall carbon nanotubes as a example, we demonstrate the use of the model to predict ground state electron density and the nuclear pseudocharges, when three parameters — namely, the radius of the nanotube, its axial stretch, and the twist per unit length — are specified as inputs. Other electronic properties of interest, including the ground

state electronic free energy, can be evaluated from these predicted fields with low-overhead post-processing, typically to chemical accuracy. We anticipate that this framework will find utility in the automated discovery of low–dimensional materials, as well as the multi-scale modeling of such systems.

The second model has an emphasis on metallic and alloy systems. One of the fundamental challenge for this model is generation of training data. The computational expense of KS-DFT scales cubically with system size which tends to stymie training data generation, making it difficult to develop quantifiably accurate ML models that are applicable across many scales and system configurations. Here, we address this fundamental challenge by employing transfer learning to leverage the multi-scale nature of the training data, while comprehensively sampling system configurations using thermalization. Our ML models are less reliant on heuristics, and being based on Bayesian neural networks, enable uncertainty quantification. We show that our models incur significantly lower data generation costs while allowing confident — and when verifiable, accurate — predictions for a wide variety of bulk systems well beyond training, including systems with defects, different alloy compositions, and at unprecedented, multi-million-atom scales. Moreover, such predictions can be carried out using only modest computational resources.

Chapter 1

Introduction

Density functional theory (DFT) provides a powerful framework for calculating electronic structure of any material and predicting a wide range of material properties, from electronic and magnetic properties to mechanical and thermodynamic behaviors. Its ability to accurately model complex systems has made DFT indispensable in materials design and optimization. However, DFT is computationally expensive and thus material research through DFT alone is laborious and prolonged. Hence, in this work, our objective was to develop a Machine Learning models that alleviates the computational burden of Density Functional Theory.

At its core, DFT solves Kohn-Sham Equations that output charge density, energy levels and wavefunctions. Solving these equations is the majority of computational

burden experienced in DFT. These outputs of Kohn-Sham equations can be postprocessed to obtain any material property. Thus, bypassing the Kohn Sham equations through Machine Learning will allow us to accelerate the material reserach by providing a computationally cheaper alternative to DFT.

Towards this, we have developed two distinct machine learning models. One model is tailored to predict electron density for low-dimensional materials like nanotubes, while the other is applicable across a wide array of material systems, with a specific emphasis on metallic and alloy compositions. Rest of the dissertation is arranged as follows. Chapter 2 explains the first machine learning model. Chapter 3 explains the second machine learning model. Chapter 4 highlights conclusions, discussions and future directions for both of these models. Chapter 5 lists the publications related to these two works. Chapter 2 and 3 include detailed introduction to the topic and related works in this direction.

Chapter 2

Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain

We present a machine learning based model that can predict the electronic structure of quasi–one–dimensional materials while they are subjected to deformation modes

©2022 APS. Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain, Shashank Pathrudkar, Hsuan Ming Yu, Susanta Ghosh, and Amartya S. Banerjee, Phys. Rev. B 105, 195141

such as torsion and extension/compression. The technique described here applies to important classes of materials systems such as nanotubes, nanoribbons, nanowires, miscellaneous chiral structures and nano–assemblies, for all of which, tuning the interplay of mechanical deformations and electronic fields — i.e., strain engineering — is an active area of investigation in the literature. Our model incorporates global structural symmetries and atomic relaxation effects, benefits from the use of *helical coordinates* to specify the electronic fields, and makes use of a specialized data generation process that solves the symmetry-adapted equations of Kohn-Sham Density Functional Theory in these coordinates. Using armchair single wall carbon nanotubes as a prototypical example, we demonstrate the use of the model to predict the fields associated with the ground state electron density and the nuclear pseudocharges, when three parameters — namely, the radius of the nanotube, its axial stretch, and the twist per unit length — are specified as inputs. Other electronic properties of interest, including the ground state electronic free energy, can be evaluated from these predicted fields with low-overhead post-processing, typically to chemical accuracy. Additionally, we show how the nuclear coordinates can be reliably determined from the predicted pseudocharge field using a clustering based technique. Remarkably, only about 120 data points are found to be enough to predict the three dimensional electronic fields accurately, which we ascribe to the constraints imposed by symmetry in the problem setup, the use of low-discrepancy sequences for sampling, and efficient representation of the intrinsic low-dimensional features of the electronic fields.

We comment on the interpretability of our machine learning model and anticipate that our framework will find utility in the automated discovery of low–dimensional materials, as well as the multi-scale modeling of such systems.

2.1 Introduction

Over the last decade, machine learning (ML) models have percolated into all areas of science and engineering. Indeed, data-driven research is already an important part of the medical sciences [3, 4, 5], chemistry [6, 7], and engineering fields like manufacturing [8, 9], applied thermodynamics [10, 11], and miscellaneous others (see e.g. [12, 13, 14, 15]). The recent interest in these techniques has been driven by the improvement in the machine learning algorithms themselves, as well as an exponential growth in computation power, and the abundance of data. Additionally, data analysis tasks such as regression, classification and dimensionality reduction, which are commonly used across all areas of science, are easily handled by machine learning algorithms by their innate design [16, 17], and this has contributed to the wide applicability of machine learning techniques.

Machine learning methods have also shown great promise for various materials physics problems [18, 19, 20, 21, 22, 23, 24, 25]. In particular, the use of high-throughput Density Functional Theory (DFT) [26, 27] calculations in conjunction with machine

learning techniques, has attracted much attention as a powerful tool for materials discovery [28, 29, 30, 31, 32]. A large section of the research in this direction so far, has been aimed at predicting specific material properties and screening novel materials for targeted applications such as energy storage. This includes electronic properties like the bandgap, chemical properties like adsorption and formation energies, and mechanical properties like Young's modulus and fracture toughness [13, 33, 34, 35, 36, 37, 38, 39, 40]. A common feature of most of these predicted material properties is that they are *low–dimensional* — usually, simple scalars. An alternative to these approaches is to use machine learning to directly predict electronic fields such as the ground state electron density for atomic configurations of interest. This is appealing since such fields contain all the information for predicting various material properties — at least in principle, and the machine learning model provides a way to bypass expensive DFT calculations which can compute these fields. Recent work in this direction includes [41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53]. The large majority of these contributions have focused on molecular systems (e.g. hydrocarbon chains and clusters), while a few have considered bulk materials. The current contribution can be viewed as an extension of the aforementioned efforts of machine learning based prediction of electronic fields to broader classes of nanomaterials — specifically, quasi-1D nanostructures. Notably, a separate strand of work has also explored improving Density Functional Theory predictions themselves, by trying to learn the elusive Hohenberg-Kohn functional [27] or by improving exchange correlation functionals

used in Kohn-Sham theory [42, 54, 55, 56, 57]. This latter class of developments will not have much bearing on the discussion that follows below.

Although machine learning based prediction of the electronic fields appears to be an attractive option for the aforementioned reasons, the high–dimensional nature of the fields usually makes it necessary to generate large amounts of data for model training and validation purposes. Additionally, since the models require a description of the atomic environments as input, it becomes necessary to choose a cutoff radius for limiting the size of the environments, or to focus on small sized systems, in order to make the models tractable. Furthermore, a careful choice of the atomic environment descriptors needs to be made to enforce symmetry and locality properties [48]. From this perspective, the current contribution is quite distinct in that global structural symmetries in lieu of environmental descriptors are utilized here, and strains are employed as model inputs. Our approach is related in spirit to [52] where the authors investigated machine learning models for the electronic fields in a hexagonal close packed crystalline material.

We present here a machine learning model that can predict the electronic structure of quasi-one-dimensional materials as they are subjected to strains commensurate with their geometries. One of the key motivations of our work is that the complex interplay of electronic fields and mechanical deformations in low–dimensional materials is an active area of investigation in the literature [58, 59, 60, 61, 62, 63, 63, 64], and

therefore, it is desirable to have machine learning models where strain parameters can be mapped to electronic fields for such systems. Additionally, the techniques described here are likely to find use in the discovery of novel phases of low–dimensional chiral matter [65] and multiscale modeling [66]. The data generation process for the ML model here is based on a recently formulated electronic structure calculation technique, that exploits the global symmetries of quasi-one-dimensional structures, and enables Kohn-Sham DFT calculations for such systems using a few representative atoms in a symmetry adapted unit cell [67, 68, 69, 70, 71, 72, 73]. This computational method, called Helical Density Functional Theory (Helical DFT), solves the symmetry adapted Kohn-Sham equations in so-called helical coordinates to yield electronic fields of interest, and is able to accommodate deformation modes such as extension, compression and torsion, commonly associated with tubular or wire-like nanostructures. Atomic relaxation effects as a response to the applied strains are automatically included, by driving the Hellman-Feynman forces [74] to zero. In order to map strain parameters to resultant electronic fields, we utilize a two-step machine learning model, motivated by recently developed techniques used to predict the high–dimensional deformation fields of multi walled carbon nanotubes [75]. Specifically, we use Principal Component Analysis (PCA) to perform dimensionality reduction of the electronic fields, and a neural network to learn in the reduced space. Using armchair single-wall carbon nanotubes as an example, we demonstrate that the ML model accurately predicts the ground-state electron density and the nuclear pseudocharge fields when the

radius of the nanotube, its axial stretch, and the twist per unit length are provided as inputs. We have also developed a novel technique based on clustering that allows us to determine the nuclear coordinates from the ML model predicted nuclear pseudocharge field, and we demonstrate the superior performance of this method when compared to alternatives. Other quantities of interest, including ground state energies and symmetry-adapted band diagrams can be readily computed from the ML model predicted fields through low-overhead postprocessing steps. The strategy of predicting smoothly varying ground state fields such as the electron density, and obtaining energies from this field, instead of predicting the latter directly, appears to work better in practice [42, 52]. In a similar manner, computation of the electronic bands using a non-self-consistent calculation involving the machine-learning based Hamiltonian (i.e., diagonalization of a symmetry adapted Kohn-Sham Hamiltonian, with the effective potential arising from machine learning predicted fields) is more straightforward when compared to prediction of the band diagram directly, as a function of the inputs. This is due to the complexities in the structure of the latter [76, 77], including e.g., the appearance of band crossings associated with insulator-metal transitions.

In our example, only about 160 simulations were performed, out of which around 120 are used for training purposes. Yet, ground state energies could be typically predicted to chemical accuracy (i.e., to better than 1.6 milli-Hartree per atom, or 1-kcal/mol), band-gap predictions were generally accurate to 0.05 eV, while the bandgap location

was predicted accurately every time. This suggests that the predictions of three-dimensional electronic fields themselves are rather accurate even with this limited training data, a fact also directly borne out by the low normalized root mean square errors in these quantities. The high accuracy of the present ML model is likely related to (i) the constraints imposed by symmetry in the problem setup, (ii) efficient exploration of the input space through quasi-random low-discrepancy sequences, and (iii) significant reduction in the dimensionality of the electronic fields. Indeed, only 7 and 15 principal component modes were found to be sufficient to capture most of the variations in the ground state electron density and the nuclear pseudocharge fields, respectively, which reinforces points (i) and (iii) above. We also observed that the electronic fields and post-processed quantities are accurately predicted for inputs whose values were not used during training, thus suggesting that our model can predict anywhere in the input space, even beyond the training data. Notably, the machine learning surrogate model is much cheaper computationally — while the DFT calculation can take up to hundreds of CPU hours (in order to include atomic relaxation effects through ab initio geometry optimization), the machine learning model prediction can be done in a fraction of a second, and the subsequent post-processing steps (including prediction of band diagrams) can be typically performed in about 30 to 40 minutes of wall time.

The rest of the paper is organized as follows. We first explain the scheme of the ab initio simulations, which are used to obtain the training data for our machine

learning model. Details regarding the system under consideration and the governing equations are presented in Section 2.2. This is followed by an overview of our machine learning model. Specifically, details of the dimension reduction of the electronic fields, neural network based regression, and a new approach to predict atomic coordinates are explained in Section 2.3. Post-processing of machine learning predicted electronic fields to evaluate various energy components, band structures and atomic coordinates is explained in Section 2.4. Next, we validate the machine learning model and quantify its accuracy in Section 2.5. We also comment on the model interpretability. We end with our conclusions and a discussion of future research directions.

2.2 Methodology: First principles calculations

In this section, we describe the system setup, key aspects of the first principles simulation method (Helical DFT). The atomic unit system with $m_e = 1, \hbar = 1, \frac{1}{4\pi\epsilon_0} = 1$ will be used throughout, unless otherwise mentioned.

For the rest of the paper, $\mathbf{e_X}$, $\mathbf{e_Y}$, $\mathbf{e_Z}$ will denote the standard orthonormal basis of $\mathbb{R}^3$. Vectors in three dimensions will be denoted using lowercase boldface letters, while 3×3 matrices will be denoted using uppercase boldface. Cartesian, cylindrical and helical coordinates will be denoted as (x, y, z), (r, ϑ, z), and (r, θ_1, θ_2), respectively,

and the relation between these is:

$$r = \sqrt{x^2 + y^2}\,,\ \theta_1 = \frac{z}{\tau}\,,$$
$$\theta_2 = \frac{1}{2\pi}\,\mathrm{arctan2}\,(y, x) - \alpha\frac{z}{\tau} = \frac{\vartheta}{2\pi} - \alpha\frac{z}{\tau}\,. \tag{2.1}$$

Here, α is related to the twist in the system as explained below.

2.2.1 System specification and global symmetries

We begin by providing a description of the geometry of the quasi-one-dimensional systems under study, and the associated computational domains. As a prototypical system, we consider a nanostructure aligned and infinite in extent along $\mathbf{e_Z}$. Since the system of interest is quasi-one-dimensional, it is of limited extent in the $\mathbf{e_X}$-$\mathbf{e_Y}$ plane. These conditions imply that the system can be embedded in a cylinder with axis $\mathbf{e_Z}$ (or annular cylinder, if the system is tubular — as considered here), of infinite height and finite radius, and this region of space will be referred to as the *global simulation domain*. The structures considered in this work may be undeformed, or more generally, they may include axial deformation (i.e., stretch or compression) along $\mathbf{e_Z}$, and/or torsional deformation about the same axis. As pointed out in the literature, helical and cyclic symmetries can be used to describe such systems

conveniently [68, 71, 78, 79, 80, 81, 82]. Thus if the atoms of the system have positions:

$$\mathcal{S} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \cdots : \mathbf{p}_i \in \mathbb{R}^3\}\,, \tag{2.2}$$

then we may identify a discrete group of isometries:

$$\mathcal{G} = \left\{\Upsilon_{m,n} = \left(\mathbf{R}_{(2\pi m\alpha + n\Theta)} | m\tau \mathbf{e}_{\mathbf{Z}}\right) : m \in \mathbb{Z}, n = 0, 1, \ldots, \mathfrak{N} - 1\right\}, \tag{2.3}$$

and a finite collection of atoms (called *simulated atoms* or *representative atoms*) with coordinates:

$$\mathcal{P} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_M : \mathbf{r}_i \in \mathbb{R}^3\}\,, \tag{2.4}$$

such that the structure can be described as the orbit of the group $\mathcal{G}$ on the set $\mathcal{P}$, i.e.,

$$\mathcal{S} = \bigcup_{\substack{m \in \mathbb{Z} \\ n=0,1,\ldots,\mathfrak{N}-1}} \bigcup_{i=1}^{M} \mathbf{R}_{(2\pi m\alpha + n\Theta)} \mathbf{r}_i + m\tau\, \mathbf{e}_{\mathbf{Z}}\,. \tag{2.5}$$

Here, $\Upsilon_{m,n}$ is an isometry operation (i.e., rigid body motion) consisting of the rotation matrix $\mathbf{R}_{(2\pi m\alpha + n\Theta)}$ with axis $\mathbf{e}_{\mathbf{Z}}$ and translation vector $m\tau\mathbf{e}_{\mathbf{Z}}$. It acts on an arbitrary point $\mathbf{x} \in \mathbb{R}^3$ by rotating it through the angle $2\pi m\alpha + n\Theta$ about the axis of the nanostructure, while simultaneously translating it by $m\tau$ about the same axis. The quantity $\mathfrak{N}$ is a natural number that captures any ($\mathfrak{N}$-fold) cyclic symmetries in the nanostructure, and the angle $\Theta = 2\pi/\mathfrak{N}$. The parameter τ is related to the axial

pitch of the structure and it can capture extensions and compressions about the axis, while the scalar α is related to the rate of applied or intrinsic twist in the structure, measured as $\beta = 2\pi\alpha/\tau$. For the structures considered here, we have $0 \leq \alpha < 1$, with $\alpha = 0$ representing untwisted structures. For undeformed armchair carbon nanotubes, the value of τ as suggested by the "roll-up construction" [79, 83] is $\sqrt{3}a$, where a is the interatomic bond-length of graphene (Figure 2.1). Note that the numbers $m \in \mathbb{Z}$ and $n \in \{0, 1, \ldots, \mathfrak{N} - 1\}$ introduced above serve to label the group elements of $\mathcal{G}$ (i.e., the isometries $\Upsilon_{m,n}$).

As pointed out in [68, 71], a key advantage of the above formulation is that with the knowledge of the relevant symmetry group, any quasi-one-dimensional material can be represented efficiently by means of the representative atoms alone — usually, just a few are adequate, and the behavior of the system under deformations (small or large) can be obtained by minimizing the system's free energy with respect to the coordinates of the representative atoms. The Helical Density Functional Theory (Helical DFT) technique described below, provides a computational framework for carrying out this procedure.

2.2.2 Helical Density Functional Theory (Helical DFT)

We use Helical Density Functional Theory (Helical DFT) [68, 71] to compute the electronic fields associated with the (possibly deformed) quasi-one-dimensional nanostructures of interest in this work. To accommodate the global symmetries of the system under study, Helical DFT solves the symmetry adapted equations of Kohn-Sham DFT within a *fundamental domain* (or *symmetry adapted unit cell*) that encapsulates the representative atoms. In the context of this work, provided that the simulated atoms have radial coordinates between R_{in} and R_{out}, a suitable fundamental domain is the following region (expressed in cylindrical coordinates):

$$\mathcal{D} = \left\{ (r, \vartheta, z) \in \mathbb{R}^3 : R_{in} \leq r \leq R_{out}, \frac{2\pi\alpha z}{\tau} \leq \vartheta \leq \frac{2\pi\alpha z}{\tau} + \Theta, 0 \leq z \leq \tau \right\}. \quad (2.6)$$

Due to the global symmetries of the system described above, the eigenstates of the Kohn-Sham Hamiltonian, and other quantities related to its spectrum can be labeled using the characters of the group (i.e., its complex one dimensional irreducible representations). For $m \in \mathbb{Z}$ and $n \in \{0, 1, 2, ..., \mathfrak{N} - 1\}$, these are [67, 68, 84, 85]:

$$\widehat{\mathcal{G}} = \left\{ e^{2\pi i \left(m\eta + \frac{n\nu}{\mathfrak{N}} \right)} : \eta \in \left[-\frac{1}{2}, \frac{1}{2} \right) ; \nu = \{0, 1, \ldots, \mathfrak{N} - 1\} \right\}. \quad (2.7)$$

Accordingly, Helical DFT uses (η, ν) to label the eigenvalues, the eigenvectors, and

the electronic occupations. For $j \in \mathbb{N}$, the symmetry adapted Kohn-Sham equations over the fundamental domain (i.e. $\mathbf{x} \in \mathcal{D}$) are:

$$\mathfrak{H}^{\mathrm{KS}}\,\psi_j(\mathbf{x};\eta,\nu) = \lambda_j(\eta,\nu)\,\psi_j(\mathbf{x};\eta,\nu)\,, \mathfrak{H}^{\mathrm{KS}} = -\frac{1}{2}\Delta + V_{\mathrm{xc}} + \Phi + \mathcal{V}_{\mathrm{nl}}\,, \tag{2.8}$$

with the eigenstates $\psi_j(\mathbf{x};\eta,\nu)$ satisfying the Helical Bloch conditions:

$$\psi_j(\Upsilon_{m,n} \circ \mathbf{x};\eta,\nu) = e^{-2\pi i\left(m\eta + \frac{n\nu}{\mathfrak{N}}\right)}\psi_j(\mathbf{x};\eta,\nu)\,. \tag{2.9}$$

In the above, $\mathfrak{H}^{\mathrm{KS}}$ denotes the Kohn-Sham operator, V_{xc} denotes the exchange correlation potential, Φ denotes the net electrostatic potential arising from the electrons and the nuclear pseudocharges (i.e., a combination of the Hartree and electron-nucleus interaction terms), and $\mathcal{V}_{nl}$ denotes the non-local pseudopotential operator. The field Φ obeys the following Poisson problem in terms of the electron density ρ and the nuclear pseudocharge field b:

$$-\Delta\Phi = 4\pi\big(\rho + b\big)\,. \tag{2.10}$$

The non-local pseudopotential operator can be expressed in Kleinman-Bylander form [86] as:

$$\mathcal{V}_{\mathrm{nl}} = \sum_{i=1}^{M}\sum_{p\in\Gamma_i}\gamma_{i,p}\hat{\chi}_{i,p}(\cdot;\eta,\nu;\mathbf{r}_i)\;\overline{\hat{\chi}_{i,p}(\cdot;\eta,\nu;\mathbf{r}_i)}\,, \tag{2.11}$$

with $\chi_{i,p}$, $\gamma_{i,p}$ and Γ_i denoting the atom-centered projection functions (associated with the i^{th} atom), the corresponding normalization constants, and the total set of projectors for the atom, respectively. Within Helical DFT, for a given set of atoms in the fundamental domain, the nuclear pseudocharge field b and the non-local pseudopotential operator $\mathcal{V}_{\text{nl}}$ are computed explicitly, along with a suitable starting guess for the electron density ρ. Following these computations, the symmetry adapted Kohn-Sham equations (eq. 2.8) are solved self-consistently [87]. At self-consistency, the free energy per unit fundamental domain and the Hellman-Feynman forces on the atoms may be computed following the expressions presented in [68, 71].

It is important to point out at this stage that construction of the symmetry adapted Kohn-Sham Hamiltonian requires knowledge of the atomic coordinates within the fundamental domain, due to the explicit dependence of the operator $\mathcal{V}_{\text{nl}}$ on the latter. Thus, unless all-electron calculations are being performed, it is not possible to compute the Kohn-Sham eigenstates via a simple diagonalization step, even if the electron density and the nuclear pseudocharge fields are known. As discussed later (Section 2.3.3), we address this issue in this work by means of an unsupervised learning technique that can pick out the atomic coordinates from the nuclear pseudocharge field, which in turn can be used to set up the operator $\mathcal{V}_{\text{nl}}$.

2.2.3 Use of helical coordinates

The fundamental domain $\mathcal{D}$ assumes the form of a cuboid $[R_{in}, R_{out}] \times [0, 1] \times [0, 1/\mathfrak{N}]$ in helical coordinates. Helical DFT uses a higher order finite difference scheme in these coordinates to discretize and solve the governing equations [68, 71]. Thus, the electronic fields computed by the method are available over a set of grid points (corresponding to the finite difference mesh) in the fundamental domain.

In addition to converting the complicated geometry of the fundamental domain to a simple cuboidal geometry for simulations, helical coordinates allow for additional simplifications in the data generation process. First, irrespective of the nanotube radius and the level of torsional and axial deformation imposed, the helical coordinates of an atom within the fundamental domain are such that θ_1 and $\mathfrak{N}\,\theta_2$ remain constant, as long as relaxation effects are negligible. Thus, even when relaxation effects are not small, this property can be used to provide good starting guesses to the structural relaxation procedure. Second, for nanotubes of any radii undergoing relatively small torsional or extensional distortions, the total number of grid points (and hence the size of the vector used for describing the electronic fields) can be kept constant, with relatively small changes to the overall accuracy of the calculations. To see this, we denote $\mathsf{N}_r, \mathsf{N}_{\theta_1}, \mathsf{N}_{\theta_2}$ as the number of grid points along the r, θ_1 and θ_2 directions, respectively. The electronic fields are then represented as vectors in dimension $\mathsf{N}_r \times$

$\mathsf{N}_{\theta_1} \times \mathsf{N}_{\theta_2}$, and the mesh spacings corresponding to these discretization choices are:

$$h_r = \frac{R_{out} - R_{in}}{\mathsf{N}_r}, h_{\theta_1} = \frac{1}{\mathsf{N}_{\theta_1}}, h_{\theta_2} = \frac{1/\mathfrak{N}}{\mathsf{N}_{\theta_2}} \tag{2.12}$$

The overall mesh spacing $h = \max\Big(h_r, \tau h_{\theta_1}, 2\pi(\frac{R_{\text{in}}+R_{\text{out}}}{2})h_{\theta_2}\Big)$ dictates the accuracy of the calculation. In the radial direction, by enforcing a constant amount of vacuum padding around the tubes, the mesh spacing h_r (and hence N_r) can be kept constant with respect to the tube diameter. In the axial direction, small changes to τ with respect to its equilibrium value (due to imposed strains) do not affect the overall calculation accuracy appreciably, as long as N_{θ_1} is large enough to accommodate the largest value of τ considered. Finally, in the θ_2 direction, assuming the nanotube is placed halfway between R_{in} and R_{out}, the effect of change in $\frac{R_{\text{in}}+R_{\text{out}}}{2}$ is offset by the corresponding change in cyclic group order $\mathfrak{N}$, thus helping keep the product $(\frac{R_{\text{in}}+R_{\text{out}}}{2})h_{\theta_2}$ constant. Thus the same value of N_{θ_2} can be chosen irrespective of the tube diameter.

2.2.4 Other details of first principles calculations

All Helical DFT calculations described in this work use a 4-atom fundamental domain as shown in Figure 2.1. To enable expeditious generation of data, calculations are done in two steps. First, ab initio geometry optimization calculations are done for a given

level of axial and torsional strains by using $h = 0.3$ Bohr, and by sampling 15 k-points in the η direction. These discretization choices are sufficient to produce chemically accurate forces and ground state energies for the Troullier-Martins norm conserving pseudopotential [88] used to model the carbon atoms in this work [71]. Atomic relaxation is carried out using the Fast Intertial Relaxation Engine [89], and the structures are relaxed till each atomic force component drops below 0.001 Ha/Bohr. Next, for each relaxed structure, we redo a self-consistent calculation to generate the electronic fields data for the machine learning model, using the finest discretization parameters that could be reliably afforded within computational resource constraints. This corresponds to a mesh spacing of $h = 0.25$ Bohr (resulting in $\mathsf{N}_r \times \mathsf{N}_{\theta_1} \times \mathsf{N}_{\theta_2} \approx 60{,}000$) and 21 k-points in the η-direction. Due to the use of the above two-step procedure to generate the data, the machine learning model automatically incorporates atomic relaxation effects in response to applied strains.

For all ab initio calculations, we used the Perdew-Wang parametrization [90] of the Local Density Approximation [26], a 12^{th} order finite difference discretization scheme [68, 91, 92, 93, 94, 95, 96, 97], vacuum padding of 11 Bohrs in the radial direction and 1 milli-Hartree of smearing using the Fermi-Dirac distribution.

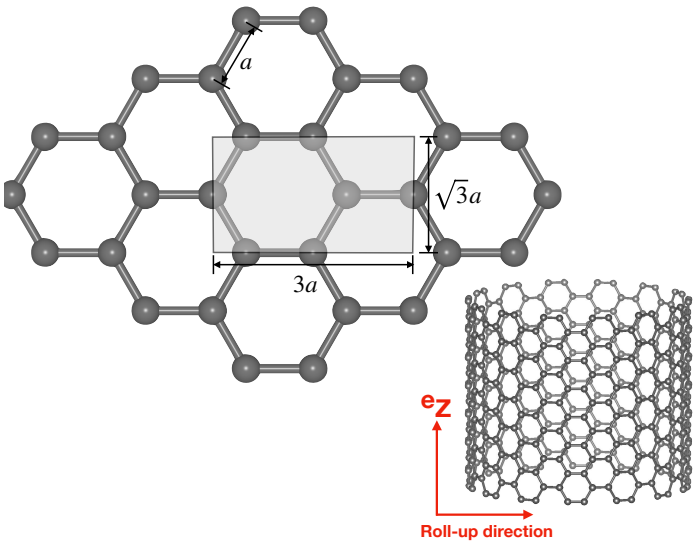


Figure 2.1: Roll-up construction of an undeformed armchair carbon nanotube, starting from a graphene sheet. The 4 atoms shown in the shaded region are used for the data generation process using Helical DFT. The parameter a represents the planar interatomic distance of 1.407 Angstrom.

2.3 Methodology: Machine Learning Model for Prediction of the Electronic Fields

This section describes the proposed Machine Learning (ML) model that aims to predict the electronic structure (high–dimensional) of quasi–one–dimensional materials under torsional and axial loads. The tubular structures considered in this work can be characterized by their radius — which is related to the degree of cyclic symmetry present in the structure, and the position of the atoms within the fundamental domain. Given strain parameters related to axial and torsional loads that the structure might be subject to, these atomic positions can be determined by minimizing

the system's energy with respect to them. Thus, the trio of parameters R_{avg} — the nanotube radius (or equivalently, the average radial coordinate of the atoms in the fundamental domain), α — the twist parameter, and τ — the axial pitch parameter, serve to specify a particular nanotube, along with the imposed torsional and axial strains. Accordingly, we let $\mathcal{H}$ denote the map from the space consisting of system and loading parameters $(R_{\text{avg}}, \alpha, \tau)$ to the electronic fields ρ, b of the deformed nanotubes:

$$\mathcal{H} : \{R_{\text{avg}}, \alpha, \tau\} \to \{\rho, b\} \tag{2.13}$$

The objective of this work is to approximate this map $\mathcal{H}$ using a machine learning model. Inputs of this map R_{avg}, α and τ are scalars, while the outputs $\rho\,(r,\, \theta_1,\, \theta_2)$ and $b\,(r,\, \theta_1,\, \theta_2)$ are high–dimensional discretized scalar fields (expressed in helical coordinates).

Approximating the map $\mathcal{H}$ directly through a supervised machine learning algorithm (such as a Neural Network (NN)) is infeasible since the output quantities $\rho\,(r,\, \theta_1,\, \theta_2)$ and $b\,(r,\, \theta_1,\, \theta_2)$ are very high–dimensional. For instance, with the discretization choices adopted in this work, the field $\rho\,(r,\, \theta_1,\, \theta_2)$ is represented by a vector of dimension close to $60,000$ (see Section 2.2). The difficulty in predicting such high–dimensional outputs using machine learning models is referred to as *curse of dimensionality*. Specifically, the number of discrete cells required to discretize the output space grows exponentially with its dimensionality, and an exponentially large

quantity of training data is then needed to ensure that the cells in the output space are accurately mapped from the input space [16].

In the present work, we circumvent this problem by using Principal Component Analysis (PCA) to reduce the dimensions of the electronic fields. Subsequently, the low–dimensional representation of the electronic fields is learned via neural networks in a supervised manner. This two-step approach, i.e., dimensionality reduction followed by learning in the reduced space, allows the prediction of the high–dimensional quantities such as electronic fields while reducing the data required for training. Schematic of the two-step ML model introduced above is given in Fig. 2.2. Recently, a similar approach has been found to have excellent accuracy in high–dimensional predictions related to purely mechanical problems [75, 98].

In the following sections, we detail various important aspects of the above ML model and also describe an auxiliary clustering based technique that allows us to determine the nuclear coordinates from the ML model predicted nuclear pseudocharge field.

2.3.1 Design of Experiments to Explore the Input Space

We now describe the use of Design of Experiments (DoE) [99, 100] techniques for efficient sampling in the input space. As described above, the triplet of input parameters $\{R_{\text{avg}}, \alpha, \tau\}$ specify a particular nanotube and the applied strains. The number

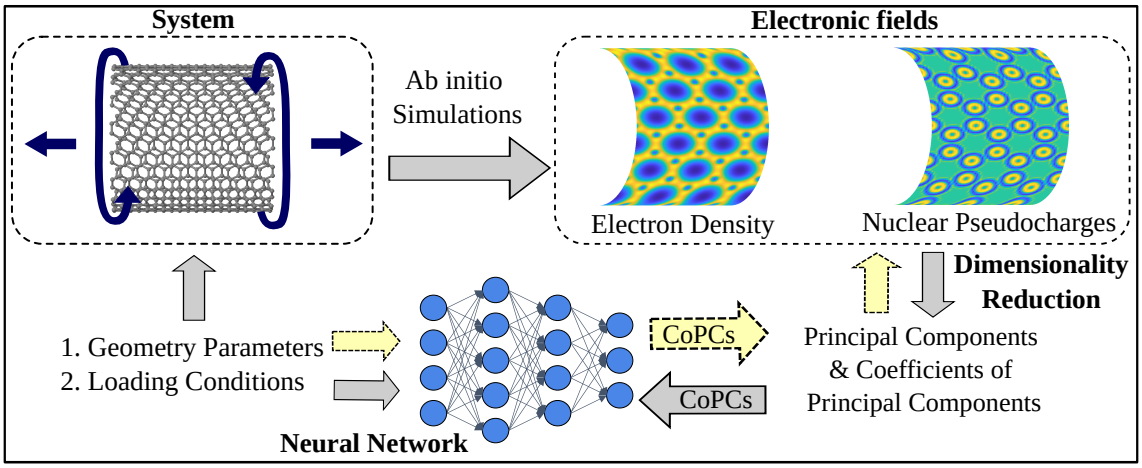


Figure 2.2: Schematic of the present Machine Learning (ML) model and the data generation process via DFT simulations. The firm arrows show the steps for data generation and training, and the dashed arrows show the steps for prediction via the ML model.

of possible combinations with these three input variables can be quite large even if finite bounds are emplaced for these variables. Given the relatively high cost of DFT simulations for the deformed nanotubes, it is infeasible to simulate nearly all possible combinations in the input space. Purely random sampling of the input space is not desirable either, since it may require a large number of sampling points to learn the pattern in the data accurately [101, 102, 103]. To address this challenge, we generate sequences of quasi-random sampling points in the input space to reduce the number of simulations required for training an accurate ML model.

Quasi-random sampling: Space-filling designs can be used to explore the input domain effectively since they sample the space uniformly without assuming any prior knowledge of the problem [101, 104]. Commonly used space-filling designs include low discrepancy sequences [105, 106], good lattice points [107], Latin Hypercube Sampling

[108] and Orthogonal Latin Hypercube sampling [109]. These methods are often evaluated based on their measure of uniformity [102, 110, 111], and such criteria suggest that Optimal Latin hypercube sampling [112] and Sobol sequences [106, 113] offer a great balance between uniform and random sampling. In this work, we have chosen Sobol sequences (low discrepancy quasirandom sequences), to sample the input space. The main advantage of this technique is that the samples generated via this procedure are spread out over the input variables space non-uniformly, but cover the space evenly [114], thus allowing efficient exploration of the input space. An additional benefit is that as the Sobol sequence progresses, the input variables space is refined successively. This latter feature allows us to add simulations to the training in a systematic manner, till the desired accuracy is achieved in the ML model. Further details of the sampling procedure used are provided in Appendix A.1.

2.3.2 Dimensionality Reduction of the Electronic Fields and Regression in the Reduced Dimension

Dimensionality Reduction of the Electronic Fields: We reduce the high dimensionality of the electronic fields using Principal Component Analysis (PCA) [115, 116, 117, 118]. PCA reduces the dimensionality of the data by projecting it onto a lower–dimensional space such that the maximum statistical information within the data is retained. The basis vectors for this low–dimensional space are uncorrelated with each

other and are called the principal components. Thus, PCA enables dimensionality reduction while minimizing the information loss.

To elaborate further, given the data points $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, \ldots, n)$, PCA allows one to obtain a lower dimensional approximation $\tilde{\mathbf{x}}_i \in \mathbb{R}^K$, such that, $K < d$, and:

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^{K} c_{ij}\mathbf{v}_j + \boldsymbol{\mu}\,. \tag{2.14}$$

Here, $\boldsymbol{\mu} = \dfrac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$ is the sample mean, the orthonormal vectors $\mathbf{v}_j$ are the principal components (PCs) and the scalars c_{ij} are the coefficients of the principal components (CoPCs). Importantly, the PCs ($\mathbf{v}_j$) depend on the entire dataset rather than being associated with a particular data point; therefore, all the points in the original dataset can be defined in terms of distinct c_{ij} values, but the same $\mathbf{v}_j$. The value of K depends on the degree of variance of the data that needs to be captured. We perform PCA on the electronic field (ρ and b) – data generated by the DFT simulations.

Regression for the Electronic Fields in the Reduced Dimension: We employ Neural Networks (NN) [16, 119] to perform regression for the electronic fields in the reduced dimension. The choice of NN is motivated by our previous work on the ML based modeling of complex rippling deformation fields of low-dimensional nanostructures [75]. The NN architecture consists of the input layer, multiple hidden layers, and the output layer. The neurons of the hidden layers contain a weighted linear transform of

neurons in the previous layer acted upon by a nonlinear activation function. During the training phase of the model, the neural network learns the map between input and output spaces by finding the weights of these linear transforms such that it can accurately predict output for a given input. We use NNs to predict the coefficients of the principal components (CoPCs) of the electronic fields for a given system and loading parameters. We deploy two different neural networks $\mathcal{N}_1$ and $\mathcal{N}_2$ to predict CoPCs for ρ and b respectively, which use the same input parameters. In the input layer, we have three neurons, for the input parameters (i) R_{avg} , (ii) α and (iii) τ. The neurons of the output layer correspond to the CoPCs (c_{ij}). Note that the number of CoPCs depends on the desired variance to be captured in the data.

Inference via the trained ML model involves the following two steps. First, the CoPCs are predicted for a given input using the neural network. Second, the predicted CoPCs are used to obtain the higher dimensional electronic fields using the principal components following Eq. 2.14. These two steps inference procedure via the ML model are shown in Fig. 2.2.

2.3.3 Prediction of Nuclear Coordinates from Pseudocharge Fields

As mentioned earlier, calculation of the Kohn-Sham Hamiltonian arising from ML predicted fields requires knowledge of the nuclear coordinates so that the non-local part of the pseudopotential operator may be constructed (see Section 2.2.2). In this section we deal with the problem of obtaining these coordinates as a function of the tube geometry and loading parameters, i.e., $\{R_{\text{avg}}, \alpha, \tau\}$.

One possible approach [52] is to directly train a neural network with these parameters as inputs and the desired nuclear coordinates as outputs. In our experience, however, this approach does not appear to work particularly well (see Appendix A.3), and the amount of training data that was found to be adequate for predicting the electronic fields ρ and b accurately, was found to result in unacceptable levels of error while predicting the nuclear coordinates. This led us to devise a new strategy for determining the nuclear coordinates from the ML predicted nuclear pseudocharge field $b(\mathbf{x})$, since this field is readily predicted with relatively high accuracy (Section 2.5.2), and it already contains the nuclear coordinate information in principle.

We make the observation that the nuclear pseudocharge field over the fundamental domain is a superposition of the individual atomic pseudocharges i.e., $b(\mathbf{x}) = \sum_{i=1}^{M} b_i(\mathbf{x})$.

Furthermore, each atomic pseudocharge field is spherically symmetric and atom centered (i.e. $b_i(\mathbf{x}) \equiv b_i(|\mathbf{x} - \mathbf{r}_i|)$), and under usual circumstances, also non-overlapping. This suggests that a clustering based approach that can identify agglomerations of positive charges arising from individual atoms might be fruitful, and the desired nuclear coordinates can then be determined as cluster centers. Clustering algorithms are widely employed to divide datasets into smaller subgroups in an unsupervised manner, such that the data points in each subgroup share some common attributes [16]. One of the most widely used and successful clustering algorithms is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [120, 121]. This technique creates clusters for volumes with a high density of points, and treats the points which lie in very low-density volumes as outliers. DBSCAN offers advantages over other clustering algorithms like $k-$nearest neighbors, since it does not require prior knowledge of the number of clusters present in the data, it can find out any arbitrarily shaped clusters and it is robust against errors induced by the outliers. In the present case, this means that when applied to the nuclear pseudocharge data, DBSCAN should be able to form clusters around every nucleus in the fundamental domain, without the total number of nuclei being specified apriori. However, we found that a direct application of DBSCAN to the pseudocharge field fails to determine the nuclear coordinates accurately. In the following, we identify two reasons for this failure and develop procedures to overcome them.

First, the fundamental domain is effectively periodic in the θ_1 and θ_2 directions. However, clustering algorithms are not typically aware of domain boundary conditions, as a result of which, pseudocharges associated with atoms close to the domain edges may result in the identification of clusters for which the cluster centers are not at the nuclear coordinates. This issue is readily addressed by expanding the fundamental domain into a supercell, applying the clustering procedure to the periodically replicated pseudocharge field in the supercell, and finally, retaining the cluster centers found to lie within the fundamental domain. Second, some atomic pseudocharges (such as the one associated with the Troullier Martins pseudopotential for Carbon used in this work), while being radially symmetric, may exhibit multiple peaks, when plotted as a function of atom center distance (see Figure 2.3). This can cause the clustering algorithm to identify multiple clusters near a single nucleus and the centers of these clusters will not coincide with the nuclei. To overcome this challenge we propose a map ($\mathcal{T}$) that truncates the pseudocharge field b to retain only the data around the first peak (see Figure 2.3):

$$\mathcal{T} : b(r,\theta_1,\theta_2) \rightarrow \bar{b}(r,\theta_1,\theta_2), \;\; \bar{b}(r,\theta_1,\theta_2) = \begin{cases} b(r,\theta_1,\theta_2), & \text{if } b(r,\theta_1,\theta_2) > c_t \\ \\ 0, & \text{if } b(r,\theta_1,\theta_2) \leq c_t \end{cases} \tag{2.15}$$

The only quantitative information needed for implementing this map is the height of the second peak c_t, which is readily available for the pseudopotentials used to produce the training data. The DBSCAN procedure, when applied on the truncated field $\bar{b}$

can readily identify the nuclear pseudocharge density cluster around each nucleus. Nuclear coordinates are subsequently computed as centers of these clusters.

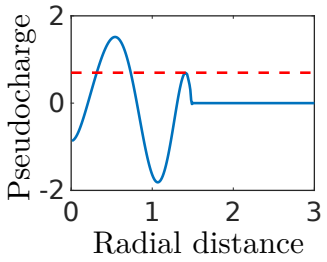


Figure 2.3: Atomic pseudocharge as a function of distance (in Bohr) from the atom for the Troullier-Martins pseudopotential for Carbon used in this work. The dashed red line indicates the truncation level employed before the DBSCAN procedure is used.

Together, the above set of strategies leads to a robust and efficient method for obtaining the nuclear coordinates as a function of the ML model inputs. The entire procedure outlined above executes within a few seconds of wall time on a desktop and is able to determine the nuclear coordinates to acceptable levels of accuracy in every case (see Table 2.2). Comparison of the accuracy of our clustering based approach, with that of nuclear coordinate predictions using a standard neural network are presented in Appendix A.3.

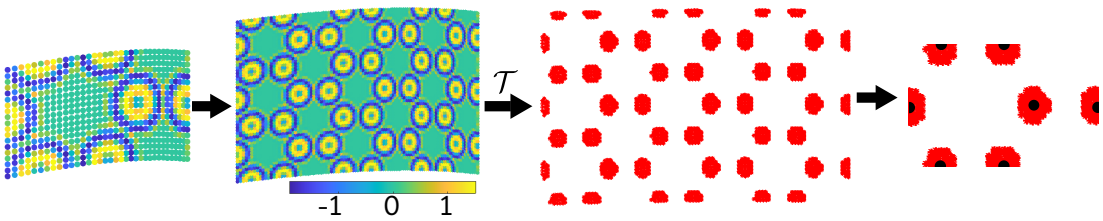


Figure 2.4: Cluster formation from nuclear pseudocharge field to determine nuclei position. A slice of the pseudocharge field at the average radial coordinate of the atoms in the fundamental domain is shown. Red clusters show the positive charge around the nucleus and the black dots are nuclei. The pseudocharge field on the fundamental domain is expanded to a supercell to avoid domain edge effects, a truncation is implemented to discard secondary peaks in the atomic pseudocharges, the DBSCAN procedure is then applied on the supercell and finally, the nuclear coordinates within the fundamental domain are identified.

2.4 Post-processing of ML Predicted Electronic Fields

In this section, we describe the postprocessing steps used for computing quantities of interest from the machine learning model predicted fields and atomic coordinates. The machine learning model produces electronic fields $\rho^{\text{ML}}(\mathbf{x})$ and $b^{\text{ML}}(\mathbf{x})$ that includes self-consistency and atomic relaxation effects. Within the ML model, however, we do not explicitly enforce any constraints regarding the net charges associated with these fields. Although in practice these constraints seem to be automatically obeyed by the model — at least approximately (see Table 2.1 within the section on Results), we find it useful to scale the ML predicted fields for postprocessing purposes [122], as shown

below:

$$\rho^{\text{Scaled}}(\mathbf{x}) = \rho^{\text{ML}}(\mathbf{x}) \times \frac{N_{\text{e}}}{\int_{\mathcal{D}} \rho^{\text{ML}}(\mathbf{x})\, d\mathbf{x}}\,,$$
$$b^{\text{Scaled}}(\mathbf{x}) = b^{\text{ML}}(\mathbf{x}) \times \frac{-N_{\text{e}}}{\int_{\mathcal{D}} b^{\text{ML}}(\mathbf{x})\, d\mathbf{x}}\,. \tag{2.16}$$

Using these scaled fields, we compute the net electrostatic potential Φ via iterative solution of eq. 2.10 using preconditioned GMRES [123] iterations. The exchange correlation potential V_{xc} is directly computed from the electron density. Next, we use a clustering based unsupervised learning technique (see Section 2.3.3) to pick out the nuclear coordinates from the nuclear pseudocharge field and use it to set up the non-local pseudopotential operator $\mathcal{V}_{\text{nl}}$. Thereafter, we diagonalize the Kohn-Sham Hamiltonian (eq. 2.8) resulting from these machine learning predicted quantities, to obtain the Kohn-Sham eigenstates. We use a combination of Generalized Preconditioned Locally Harmonic Residual (GPLHR) [124] and Arnoldi Iterations [125] to carry out the diagonalization, and initialize the calculations using random wavefunction vectors. The Fermi level of the system is subsequently determined from the Kohn-Sham eigenvalues by enforcing the constraint of having a fixed number of electrons within the fundamental domain.

Using the aforementioned post-processed quantities, the ground state free energy per

unit fundamental domain may be calculated as [71]:

$$\mathcal{F} = E_{\rm kin} + E_{\rm xc} + E_{\rm nl} + E_{\rm el} - T_{\rm e} S \,, \tag{2.17}$$

with the terms on the right hand side denoting the electronic kinetic energy, the exchange correlation energy, the non-local pseudopotential energy, the electrostatic energy and the electronic entropy contribution at temperature $T_{\rm e}$, respectively. Alternately, a more accurate estimate for the ground state free energy per unit fundamental domain may be obtained using the Harris-Foulkes functional [126, 127]:

$$\mathcal{F}^{\rm HF} = E_{\rm band} + E_{\rm xc} - \widetilde{E}_{\rm xc} + \widetilde{E}_{\rm el} + E_{\rm sc} - T_{\rm e}\, S \,. \tag{2.18}$$

In the above, the first term on the right hand side is the electronic band energy:

$$E_{\rm band} = 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\mathfrak{N}} \sum_{\nu=0}^{\mathfrak{N}-1} \sum_{j=1}^{\infty} \lambda_j(\eta,\nu)\, g_j(\eta,\nu)\, d\eta \,, \tag{2.19}$$

in which $g_j(\eta,\nu)$ denotes the electronic occupations. The term $E_{\rm xc}$ denotes the exchange correlation energy, while:

$$\widetilde{E}_{\rm xc} = \int_{\mathcal{D}} V_{\rm xc}(\rho(\mathbf{x}))\rho(\mathbf{x})\, d\mathbf{x} \,. \tag{2.20}$$

The term $\widetilde{E}_{\text{el}}$ is related to electrostatic interactions and has the form:

$$\widetilde{E}_{\text{el}} = \frac{1}{2}\int_{\mathcal{D}} \big(b(\mathbf{x}) - \rho(\mathbf{x})\big)\Phi(\mathbf{x})\, d\mathbf{x}\,. \tag{2.21}$$

Finally, E_{sc} accounts for nuclear pseudocharge self-interactions and overlap corrections [69], while the last term is related to the electronic entropy contribution. Notably, in the above breakdown for the Harris-Foulkes energy, E_{xc} and $\widetilde{E}_{\text{xc}}$ depend solely on the electron density field, E_{sc} depends on the nuclear coordinates and the nuclear pseudocharge field, the electrostatic term $\widetilde{E}_{\text{el}}$ depends on both the electron density and nuclear pseudocharge fields, while E_{band} depends on the Kohn-Sham operator eigenvalues (i.e., its dependence on ρ and b is implicit). Therefore, monitoring these terms in addition to $\mathcal{F}^{\text{HF}}$ allows us to estimate the accuracy of the machine learning based predictions of ρ, b and other post-processed quantities (such as the eigenstates), in the energetic sense (see Section 2.5 for more details).

2.5 Results

We now present the predictions of the machine learning (ML) model for armchair carbon nanotubes under torsional and axial loading. These are compared against Helical DFT simulations to quantify the ML model's accuracy and efficacy. Notably,

the inference process from the trained ML model is orders of magnitude faster compared to the cost of the ab initio simulations using Helical DFT. While the ML model requires 0.003 seconds and 0.009 seconds to predict the ρ and b fields respectively (average times on a desktop with a 2.2 GHz Intel Xeon Gold processor), a typical ab initio structural relaxation calculation using Helical DFT can stretch into hundreds of CPU hours. Post-processing of the ML predicted electronic fields (to calculate band structures, energies, etc.) can be typically performed in about 30 to 40 minutes of wall time. Training of the neural networks for ρ and b requires about 12 and 15 minutes, respectively, measured using the same hardware setup.

2.5.1 Principal Component Analysis and Neural Networks

Principal Component Analysis Results: As the first step in our two-step ML model, we utilize PCA to obtain reduced dimensional representations for the outputs of the map $\mathcal{H}$. To reconstruct the original electronic fields with minimum reconstruction error, we capture 99.99% variance of the data. As shown later (Section 2.5.3), this is generally sufficient for obtaining electronic ground state energies to chemical accuracy and also adequate for reproducing band structures correctly. For capturing this level of variance in the data, we required only 7 PCs in case of ρ and 15 PCs in case of b (Figure 2.5).

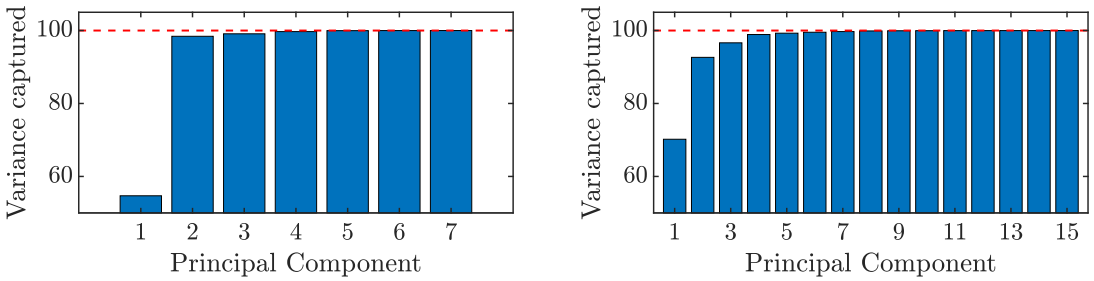


Figure 2.5: Cumulative percentage of variance vs Principal components for ρ (*left*) and b (*right*). The red dashed line shows 99.99% variance.

Neural Network: As the second step in our two-step ML model, two Neural Networks $\mathcal{N}_1$ and $\mathcal{N}_2$ are trained to predict CoPCs corresponding to ρ and b, respectively. Since 7 PCs in case of ρ and 15 PCs in case of b are required to capture 99.99% variance of the data, the number of neurons in output layers is 7 for $\mathcal{N}_1$ and 15 for $\mathcal{N}_2$. Following our architecture optimization strategy (elaborated in Appendix A.2) we choose 6 hidden layers of 150 neurons each for $\mathcal{N}_1$ and 2 hidden layers of 150 neurons each for $\mathcal{N}_2$. We use Rectified Linear Unit (ReLU) as an activation function for both networks. Mean Squared Error(MSE) is utilized as a loss function along with the elastic net regularization [128], and the Adam optimizer [129] with a learning rate of 0.001 was employed. Before the training phase, each input parameter column was scaled to zero mean and unit variance, thus standardizing the input features. 75% of the total data points were utilized for training (123 data points), 10% were utilized for validation (16 data points), and the remaining 15% were utilized for testing (25 data points). Further details of the neural network, including a discussion of the hyperparameters, and learning curves are provided in Appendix A.2.

2.5.2 Prediction of electronic fields by the ML model

We now discuss the overall performance of the machine learning model for the prediction of the electronic fields. The Pearson correlation coefficient (R) between the predicted and actual electronic fields at each point of the discretized domain for the test data was found to be 0.9949 and 0.9983 for ρ and b, respectively. In addition to the test

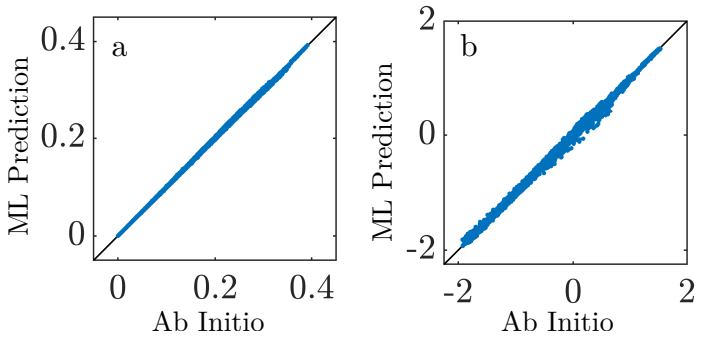


Figure 2.6: Parity plots for (a) test data of ρ ($R = 0.9949$), (b) test data of b ($R = 0.9983$).

data points described above, we have chosen three additional test data points where the input parameters were partially unseen during the training (i. $R_{\text{avg}} = 49.51\,\text{Bohr}$, $\alpha = 0.002$, $\tau = 4.5052\,\text{Bohr}$; ii. $R_{\text{avg}} = 35.55\,\text{Bohr}$, $\alpha = 0.00125$, $\tau = 4.6052\,\text{Bohr}$; iii. $R_{\text{avg}} = 53.32\,\text{Bohr}$, $\alpha = 0.0015$, $\tau = 4.5552\,\text{Bohr}$). For each of these three test cases, there is one input variable whose value was not used in the training data (e.g. data point with τ and α values present in the training data but the value of R_{avg} not present in the training data). Finally, we have randomly selected two additional test data points where none of the three input variables were seen by the ML model during

training (i. $R_{\text{avg}} = 49.51\,\text{Bohr}$, $\alpha = 0.00125$, $\tau = 4.5552\,\text{Bohr}$; ii. $R_{\text{avg}} = 30.46\,\text{Bohr}$, $\alpha = 0.00075$, $\tau = 4.6552\,\text{Bohr}$). These additional test data points with partial or wholly unseen input parameters help assess the ML model's capability to generalize beyond training data. Machine Learning predicted and actual (DFT) electronic fields for one of the test data points with all unknown input parameters are compared in Fig. 2.7.

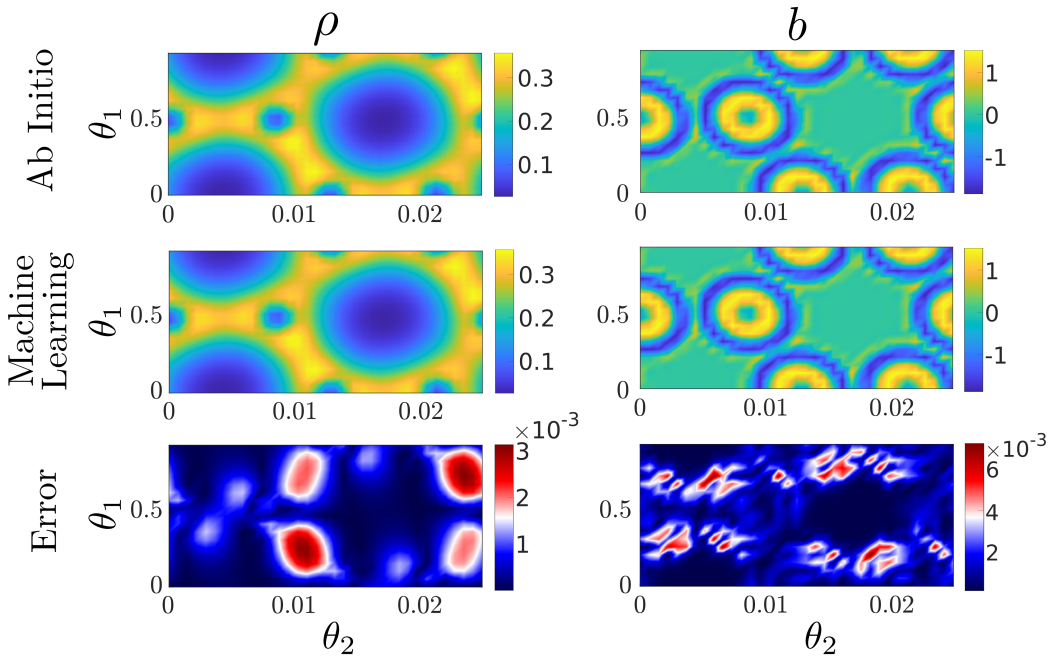


Figure 2.7: Comparison between ML predicted and DFT simulation obtained electronic fields for a test data point with all unknown input parameters ($R_{\text{avg}} = 49.51\,\text{Bohr}$, $\alpha = 0.00125$, $\tau = 4.5552\,\text{Bohr}$). A slice of the electronic fields at the average radial coordinate of the atoms in the fundamental domain is shown. The error is computed as $\frac{|\rho^{\text{DFT}} - \rho^{\text{ML}}|}{|\max(\rho^{\text{DFT}}) - \min(\rho^{\text{DFT}})|}$, similarly for b. Here, $\max(\cdot)$ and $\min(\cdot)$ denote maximum and minimum over the fundamental domain.

We quantify the error in the predicted electronic fields through the normalized root

mean square error (NRMSE) [52]:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{d}\sum_{i=1}^{d}\left(\rho_i^{\text{DFT}} - \rho_i^{\text{ML}}\right)^2}}{|\max(\rho^{\text{DFT}}) - \min(\rho^{\text{DFT}})|} \tag{2.22}$$

Here $\max(\cdot)$ and $\min(\cdot)$ denote maximum and minimum over the fundamental domain and d is the dimension of the data ($\sim 60,000$). NRMSE for b is calculated similarly. The NRMSE for various categories of test data points, including cases with partial or wholly unseen inputs, are presented in Table 2.1. The low NRMSE values on the test data are indicative of the general accuracy of the ML model. In particular, low NRMSE values for the input conditions beyond the training data establish the generalization capacity of the model.

In addition to evaluating the NRMSE values, we monitored the integrals of ρ^{ML} and b^{ML} over the fundamental domain. For a neutral system with N_{e} electrons within the computational unit cell, the electron density and the nuclear pseduocharge fields obey the normalization conditions $\int_{\mathcal{D}} \rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = N_{\text{e}}$ and $\int_{\mathcal{D}} b(\mathbf{x})\,d\mathbf{x} = -N_{\text{e}}$ respectively. Since these constraints were not built into the ML model, they allow additional quality checks on the ML predicted fields to be performed. As shown in Table 2.1, the errors associated with deviations from these constraints are quite low (0.000625 particles or lower, per electron), indicating high quality predictions of the electronic fields by the ML model.

Case	NRMSE (ρ)	$\left\lvert N_e - \int_{\mathcal{D}} \rho^{ML}(\mathbf{x})d\mathbf{x} \right\rvert$	NRMSE (b)	$\left\lvert (-N_e) - \int_{\mathcal{D}} b^{ML}(\mathbf{x})d\mathbf{x} \right\rvert$
Average for test data set	2.8×10^{-4}	3.2×10^{-3}	5.6×10^{-4}	5.4×10^{-3}
Random test data point $R_{avg} = 40.64\,, \alpha = 0.0015, \tau = 4.7052$	2.8×10^{-4}	9.2×10^{-3}	2.8×10^{-4}	8.6×10^{-3}
Test data point with unknown R_{avg} $R_{avg} = 49.51\,, \alpha = 0.002, \tau = 4.5052$	1.3×10^{-4}	5.2×10^{-4}	2.1×10^{-4}	2.5×10^{-3}
Test data point with unknown α $R_{avg} = 35.55\,, \alpha = 0.00125, \tau = 4.6052$	2.5×10^{-4}	1.2×10^{-3}	1.9×10^{-4}	5.6×10^{-3}
Test data point with unknown τ $R_{avg} = 53.32\,, \alpha = 0.0015, \tau = 4.5552$	1.6×10^{-4}	4.2×10^{-3}	2.5×10^{-4}	8.6×10^{-3}
Test data point with unknown R_{avg}, α, τ $R_{avg} = 49.51\,, \alpha = 0.00125, \tau = 4.5552$	2.1×10^{-4}	4.3×10^{-3}	4.8×10^{-4}	3.1×10^{-3}
Test data point with unknown R_{avg}, α, τ $R_{avg} = 30.46\,, \alpha = 0.00075, \tau = 4.6552$	2.3×10^{-4}	3.0×10^{-3}	2.8×10^{-4}	2.8×10^{-3}

Table 2.1
Table showing NRMSE for ML predicted ρ and b for various test cases. Also shown are errors in the integrals of electronic fields over the fundamental domain. R_{avg} and τ values are in Bohr.

2.5.3 Prediction of nuclear coordinates, energies and band structure

Finally, we post-process the ML predicted electronic fields for various test data points to obtain nuclear coordinates, electronic properties and energy components of interest. We compute the errors in these quantities for a random test data point, as well as the aforementioned five test cases for which the inputs were partially or wholly unseen by the ML model during training (Table 2.2). In general, the errors in the total ground state energy, as computed through the Harris-Foulkes functional (eq. 2.18) are found to be appreciably smaller than the chemical accuracy threshold (1.6×10^{-3} Ha/atom), except for one of the cases which had an unseen value of α. Considering the various components of the Harris-Foulkes energy, we see that the highest accuracies in the ML predictions are associated with the exchange correlation term E_{xc}, possibly due to the

sole dependence of this quantity on the electron density, which itself is predicted rather accurately. The energy component $\widetilde{E}_{\text{xc}}$ (eq. 2.18) also has a very similar behavior and is not shown in Table 2.2. The nuclear self-energy and correction terms which depend only on the nuclear pseudocharge field are also predicted with high accuracy. The electrostatic term which depends on both the nuclear pseudocharge field and the electron density, and the electronic band energy, which depends on the Kohn-Sham eigenvalues are seen to be associated with somewhat lower accuracy predictions, particularly for the test data points which had values of α and/or τ unseen by the ML model. However, even in these cases, the errors are less than 3.0×10^{-3} Ha/atom, and error cancellation leads to overall accurate ground state energy predictions. The ability to predict ground-state energies of deformed quasi-one-dimensional structures (while having atomic relaxation effects already included) with first principles accuracy, at a small computational cost is one of the great advantages of the proposed ML model, thus leading to its potential use in the multiscale modeling of low-dimensional systems [66].

The unsupervised learning procedure used for picking out nuclear coordinates is also found to be quite accurate, with typical errors (measured as the maximum error in the Cartesian coordinate components of all atoms in the fundamental domain) of the order of 0.02 to 0.03 Bohrs. The accuracy in the prediction of these coordinates is also reflected in the overall accuracy of the ML predicted Kohn-Sham Hamiltonian,

which in turn, affects the quality of electronic band diagrams and other eigenstate-dependent quantities computed from the Hamiltonian. We found strikingly good agreement between ML predicted and Helical DFT band diagrams for the test data points considered here, with a typical case (associated with wholly unseen inputs) demonstrated in Fig 2.10. Undeformed armchair carbon nanotubes are metallic [97, 130] but develop an oscillatory band gap as a function of imposed twist [71, 130]. The band gap (computed here as the difference between the smallest eigenvalue above the Fermi level and the largest eigenvalue below the Fermi level as the symmetry indices (η, ν) are varied) is particularly error prone since it is the difference of two quantities. However, the ML predicted location of the band-gap was correct for every test case and its value was correct to about 0.05 eV or better, every time. The ability of the ML model to predict the electronic structure of low-dimensional materials as a function of imposed deformation opens up the use of such techniques for strain-engineering applications [58, 59, 63].

Case	Ground state energy $\mathcal{F}^{\text{HF}}$ (Ha/atom)	Exch. Corr. energy E_{xc} (Ha/atom)	Electrostatic term $\widetilde{E}_{\text{el}}$ (Ha/atom)	Nuclear self energy & correction term E_{sc} (Ha/atom)	Band Energy E_{band} (Ha/atom)	Band gap (eV)	Atomic coordinates $\mathbf{r}_i$ (Bohr)
Random test data point $R_{\text{avg}} = 40.64\,, \alpha = 0.0015, \tau = 4.7052$	9.0×10^{-4}	4.1×10^{-5}	7.1×10^{-5}	1.5×10^{-4}	9.9×10^{-4}	0.017	0.026
Test data point with unknown R_{avg}: $R_{\text{avg}} = 49.51\,, \alpha = 0.0020, \tau = 4.5052$	6.7×10^{-4}	1.9×10^{-5}	4.2×10^{-4}	4.3×10^{-4}	6.7×10^{-4}	0.018	0.028
Test data point with unknown α $R_{\text{avg}} = 35.55\,, \alpha = 0.00125, \tau = 4.6052$	3.6×10^{-3}	8.9×10^{-5}	2.1×10^{-3}	3.2×10^{-4}	1.2×10^{-3}	0.042	0.019
Test data point with unknown τ $R_{\text{avg}} = 53.32\,, \alpha = 0.0015, \tau = 4.5552$	2.2×10^{-4}	5.4×10^{-5}	2.7×10^{-3}	4.7×10^{-5}	2.9×10^{-3}	0.008	0.023
Test data point with unknown R_{avg}, α, τ $R_{\text{avg}} = 49.51\,, \alpha = 0.00125, \tau = 4.5552$	6.5×10^{-4}	7.4×10^{-5}	1.9×10^{-3}	1.8×10^{-4}	1.4×10^{-3}	0.008	0.022
Test data point with unknown R_{avg}, α, τ $R_{\text{avg}} = 30.46\,, \alpha = 0.00075, \tau = 4.6552$	1.35×10^{-4}	8.0×10^{-5}	1.8×10^{-3}	4.1×10^{-4}	1.3×10^{-3}	0.042	0.034

Table 2.2
Errors in various post-processed quantities. Refer to eq. 2.18 and related discussion for interpretation of the various energetic terms. R_{avg} and τ values are in Bohr.

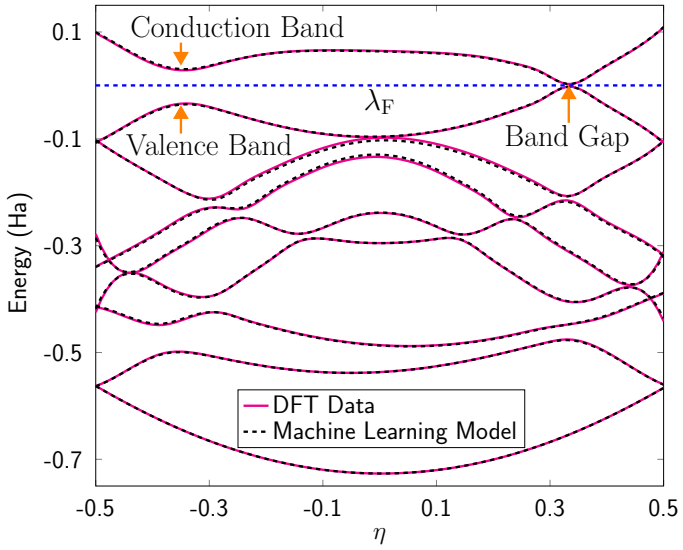


Figure 2.8: Symmetry adapted band diagram in η, at $\nu = 2$.

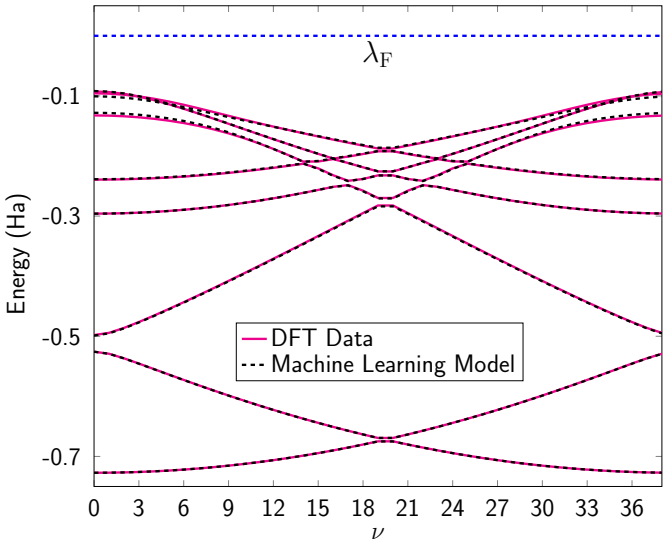


Figure 2.9: Symmetry adapted band diagram in ν, at $\eta = 0$.

Figure 2.10: Comparison of symmetry adapted band diagrams produced by the original DFT method and the machine learning model (with post processing) for the unknown test data point with $R_{\text{avg}} = 49.51$ Bohr, $\alpha = 0.00125$ and $\tau = 4.5552$ Bohr. The agreement appears excellent and the post–processed ML model is also able to precisely predict the location of the band–gap (at $\eta = \frac{1}{3}, \nu = 2$) as well as its value (0.128 eV from Helical DFT) to about 6% accuracy in this case. Note that λ_F denotes the system's Fermi level.

We attribute the high accuracy of the model here to the accuracy in both dimensionality reduction and learning through NNs. The fact that during training, the model uses only about 120 data points and that chemical accuracy requirements are met during prediction even for unseen input test cases, are particularly noteworthy and prove the effectiveness and generalizability of our model. Also, as pointed out earlier, in addition to being accurate, the proposed ML model is significantly more computationally efficient than DFT simulations.

2.5.4 Interpretation of PCA modes

The number of PCs required in this problem is significantly less than the original dimensions of the electronic fields data ($\sim 60,000$), thus indicating that these quantities are mostly confined to subspaces of much lower–dimension. The presence of these hidden lower–dimensional features, and the significant reduction of dimensionality of the data through PCA, in turn, implies that just a few CoPCs have to be predicted as a function of the input parameters by the second step of the ML model. This helps account for the fact that such predictions can be made with relatively little training data, as discussed earlier. Remarkably, just the first couple of PCs appear sufficient to capture well over 90% of the variance in both ρ and b. Figure 2.11 shows these two PCA modes for each quantity visualized using helical coordinates, specifically in a $\theta_1 - \theta_2$ plane located at the center of the simulation domain. As expected, the PCs of

ρ and b capture the most significant aspects of the variations in these quantities, with the modes of ρ reflecting changes in charge density along the carbon-carbon bonds, and those of b capturing shifts in the nuclear positions.

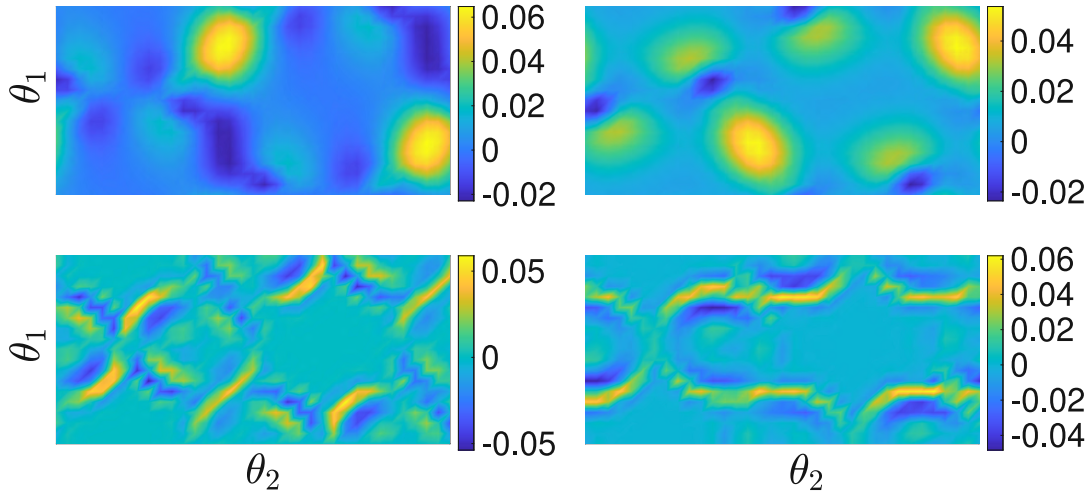


Figure 2.11: First two principal components for ρ (*top*) and b (*bottom*). A slice of the PCA modes at the average radial coordinate of the atoms in the fundamental domain is shown.

To elaborate on the above interpretations, we first recall that (see Section 2.2.3) in the absence of relaxation effects, an atom within the fundamental domain has the same values of θ_1 and $\mathfrak{N}\,\theta_2$, regardless of the tube radius or the level of axial/torsional strain imposed (as before, $\mathfrak{N}$ denotes the cyclic symmetry group order). Consequently, for all values of the input parameters, the nuclei are expected to be located in the same relative positions in a $\theta_1 - \theta_2$ planar plot, if atomic relaxation effects can be ignored. In the practice, upon imposition of strain, the nuclei re-adjust their positions to minimize the system energy during the structural relaxation procedure, leading to somewhat

different values of the helical coordinates associated with their pseudocharge centers, than would be suggested by purely geometrical considerations. The PCA modes for b illustrated in Figure 2.11 appear to be capturing this "motion" of the pseudocharge centers (i.e., nuclear coordinates) associated with the relaxation procedure. Furthermore, due to the changes in the nuclear positions, the carbon-carbon bond lengths change, and the PCA modes for ρ appear to be capturing changes in the electron density along these bonds while they are stretched or compressed due to the imposed strains. Notably, these bonds are at angles with respect to the $\theta_1 - \theta_2$ axes (see Figure 2.1 and top row of Figure 2.7), leading to the tilted appearance of the electron density lobes observable in Figure 2.11. Finally, the presence of more wiggles in the plots for the PCA modes for b, as compared to those for ρ can be explained by observing that at a discrete level, the latter is a smoother quantity. Specifically, the discretized b field can have sharper local jumps since it is the sum of individual atom centered pseudocharges, while ρ is more smeared out (also see top row of Figure 2.7). Indeed, this difference in relative degrees of smoothness at the discrete level probably contributes to the different number of PCA modes for these quantities needed to capture the same level of variance in the data (Figure 2.5).

Chapter 3

Electronic Structure Prediction of Multi-million Atom Systems Through Uncertainty Quantification Enabled Transfer Learning

The ground state electron density — obtainable using Kohn-Sham Density Functional Theory (KS-DFT) simulations — contains a wealth of material information,

[0]Uploaded to arxiv, arXiv:2308.13096. Submitted to npj Computational Materials [131]

making its prediction via machine learning (ML) models attractive. However, the computational expense of KS-DFT scales cubically with system size which tends to stymie training data generation, making it difficult to develop quantifiably accurate ML models that are applicable across many scales and system configurations. Here, we address this fundamental challenge by employing transfer learning to leverage the multi-scale nature of the training data, while comprehensively sampling system configurations using thermalization. Our ML models are less reliant on heuristics, and being based on Bayesian neural networks, enable uncertainty quantification. We show that our models incur significantly lower data generation costs while allowing confident — and when verifiable, accurate — predictions for a wide variety of bulk systems well beyond training, including systems with defects, different alloy compositions, and at unprecedented, multi-million-atom scales. Moreover, such predictions can be carried out using only modest computational resources.

3.1 Introduction

Over the past several decades, Density Functional Theory (DFT) calculations based on the Kohn-Sham formulation [132, 133] have emerged as a fundamental tool in the prediction of electronic structure. Today, they stand as the de facto workhorse of computational materials simulations [134, 135, 136, 137], offering broad applicability and versatility. Although formulated in terms of orbitals, the fundamental unknown in

Kohn Sham Density Functional Theory (KS-DFT) is the *electron density*, from which many ground state material properties — including structural parameters, elastic constants, magnetic properties, phonons/vibrational spectra, etc., may be inferred. The ground state electron density is also the starting point for calculations of excited state phenomena, including those related to optical and transport properties [138, 139].

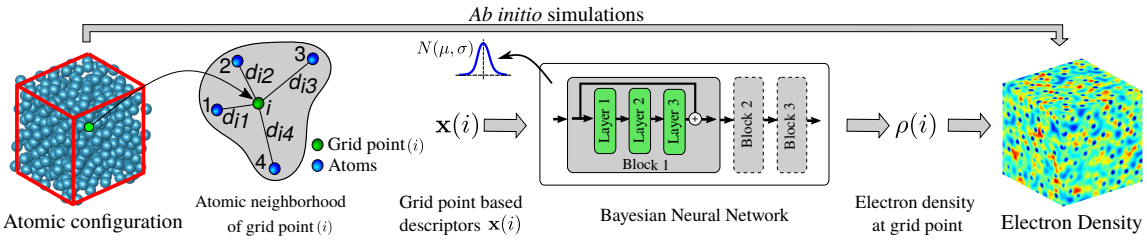


Figure 3.1: Overview of the present Machine Learning (ML) model. The first step is the training data generation via *ab initio* simulations shown by the arrow at the top. The second step is to generate atomic neighborhood descriptors $\mathbf{x}(i)$ for each grid point, i, in the training configurations. The third step is to create a probabilistic map (Bayesian Neural Network with DenseNet like blocks consisting of skip connections) from atomic neighborhood descriptors $\mathbf{x}(i)$ to the charge density at the corresponding grid point $\rho(i)$. The trained model is then used for inference which includes (i) descriptor generation for all grid points in the query configuration, (ii) forward propagation through the Bayesian Neural Network, and (iii) aggregation of the point-wise charge density $\rho(i)$ to obtain the charge density field ρ.

In spite of their popularity, conventional KS-DFT calculations scale in a cubic manner with respect to the number of atoms within the simulation cell, making calculations of large and complex systems computationally burdensome. To address this challenge, a number of different approaches, which vary in their computational expense and their range of applicability, have been proposed over the years. Such techniques generally avoid explicit diagonalization of the Kohn-Sham Hamiltonian in

favor of computing the single particle density matrix [140]. Many of these methods are able to scale linearly with respect to the system size when bulk insulators or metals at high temperatures are considered [140, 141, 142, 143, 144], while others exhibit sub-quadratic scaling when used for calculations of low-dimensional materials (i.e., nanostructures)[145, 146]. Contrary to these specialized approaches, there are only a handful of first-principles electronic structure calculation techniques that operate universally across bulk metallic, insulating, and semiconducting systems, while performing more favorably than traditional cubic scaling methods (especially, close to room temperature). However, existing techniques in this category, e.g. [147, 148], tend to face convergence issues due to aggressive use of density matrix truncation, and in any case, have only been demonstrated for systems containing at most a few thousand atoms, due to their overall computational cost. Keeping these developments in mind, a separate thread of research has also explored reducing computational wall times by lowering the prefactor associated with the cubic cost of Hamiltonian diagonalization, while ensuring good parallel scalability of the methods on large scale high-performance computing platforms [149, 150, 151, 152]. In spite of demonstrations of these and related methods to study a few large example problems (e.g. [153, 154, 155]), their routine application to complex condensed matter systems, using modest, everyday computing resources appears infeasible.

The importance of being able to routinely predict the electronic structure of generic bulk materials, especially, metallic and semiconducting systems with a large number of

representative atoms within the simulation cell, cannot be overemphasized. Computational techniques that can perform such calculations accurately and efficiently have the potential to unlock insights into a variety of material phenomena and can lead to the guided design of new materials with optimized properties. Examples of materials problems where such computational techniques can push the state-of-the-art include elucidating the core structure of defects at realistic concentrations, the electronic and magnetic properties of disordered alloys and quasicrystals [156, 157, 158, 159], and the mechanical strength and failure characteristics of modern, compositionally complex refractory materials [160, 161]. Moreover, such techniques are also likely to carry over to the study of low dimensional matter and help unlock the complex electronic features of emergent materials such as van-der-Waals heterostructures [162] and moiré superlattices [163]. Notably, a separate direction of work has also explored improving Density Functional Theory predictions themselves, by trying to learn the Hohenberg-Kohn functional or exchange correlation potentials[42, 55, 164]. This direction of work will not have much bearing on the discussion that follows below.

An attractive alternative path to overcoming the cubic scaling bottleneck of KS-DFT — one that has found much attention in recent years — is the use of Machine Learning (ML) models as surrogates [25, 165]. Indeed, a significant amount of research has already been devoted to the development of ML models that predict the energies and forces of atomic configurations matching with KS-DFT calculations, thus spawning ML-based *interatomic potentials* that can be used for molecular dynamics calculations

with *ab initio* accuracy [166, 167, 168, 169, 170, 171]. Parallelly, researchers have also explored direct prediction of the ground state electron density via ML models trained on the self-consistent electron density obtained from KS-DFT simulations [42, 44, 48, 172, 173, 174]. This latter approach is particularly appealing, since, in principle, the ground state density is rich in information that goes well beyond energies and atomic forces, and such details can often be extracted through simple post-processing steps. Development of ML models of the electron density can also lead to electronic-structure-aware potentials, which are likely to overcome limitations of existing Machine Learning Interatomic Potentials, particularly in the context of reactive systems [175, 176]. Having access to the electron density as an intermediate verifiable quantity is generally found to also increase the quality of ML predictions of various material properties [172, 177], and can allow training of additional ML models. Such models can use the density as a descriptor to predict specific quantities, such as defect properties of complex alloys [178, 179] and bonding information [50]. Two distinct approaches have been explored in prior studies to predict electron density via Machine Learning, differing in how they represent the density – the output of the machine learning model. One strategy involves representing the density by expanding it as a sum of atom-centered basis functions [43, 47]. The other involves predicting the electron density at each grid point in a simulation cell. Both strategies aim to predict the electron density using only the atomic coordinates as inputs. While the former strategy allows for a compact representation of the electron density, it requires

the determination of an optimized basis set that is tuned to specific chemical species. It has been shown in [43] that the error in the density decomposition through this strategy can be reduced to as low as 1%. In contrast, the latter strategy does not require such optimization but poses a challenge in terms of inference - where the prediction for a single simulation cell requires inference on thousands of grid points (even at the grid points in a vacuum region). The former strategy has shown good results for molecules [43] while the latter has shown great promise in density models for bulk materials especially metals [48, 53, 174]. In this work, we use the latter approach.

A key challenge in building surrogate models of the ground state electron density from KS-DFT calculations is the process of data generation itself, which can incur significant offline cost [180]. In recent work [177], we have demonstrated how this issue can be addressed for chiral nanomaterials [181]. For such forms of matter, the presence of underlying structural symmetries allows for significant dimensionality reduction of the predicted fields, and the use of specialized algorithms for ground state KS-DFT calculations [68, 182, 183]. However, such strategies cannot be adopted for bulk materials with complex unit cells, as considered here. For generic bulk systems, due to the confining effects of periodic boundary conditions, small unit-cell simulations alone cannot represent a wide variety of configurations. To obtain ML models that can work equally well across scales and for a variety of configurations (e.g. defects [184, 185]), data from large systems is also essential. However, due to

the aforementioned cubic scaling of KS-DFT calculations, it is relatively inexpensive to generate a lot of training data using small sized systems (say, a few tens of atoms), while larger systems (a few hundred atoms) are far more burdensome, stymieing the data generation process. Previous work on electron density prediction [44, 174] has been made possible by using data from large systems exclusively. However, this strategy is likely to fail when complex systems such as multi-principal element alloys are dealt with, due to the large computational cells required for such systems. This is especially true while studying compositional variations in such systems since such calculations are expected to increase the overall computational expense of the process significantly.

In this work, we propose a machine-learning model that accurately predicts the ground state electron density of bulk materials at any scale, while quantifying the associated uncertainties. Once trained, our model significantly outperforms conventional KS-DFT-based computations in terms of speed. To address the high cost of training data generation associated with KS-DFT simulations of larger systems — a key challenge in developing effective ML surrogates of KS-DFT — we adopt a transfer learning (TL) approach [186]. Thus, our model is first trained using a large quantity of cheaply generated data from simulations of small systems, following which, a part of the model is retrained using a small amount of data from simulations of a few large systems. This strategy significantly lowers the training cost of the ML model, without compromising its accuracy. Along with the predicted electron density fields, our model also produces

a detailed spatial map of the uncertainty, that enables us to assess the confidence in our predictions for very large scale systems (thousands of atoms and beyond), for which direct validation via comparison against KS-DFT simulations data is not possible. The uncertainty quantification (UQ) properties of our models are achieved through the use of Bayesian Neural Networks (BNNs), which systematically obtain the variance in prediction through their stochastic parameters, and tend to regularize better than alternative approaches [5, 187, 188]. They allow us to systematically judge the generalizability of our ML model, and open the door to Active Learning approaches [189] that can be used to further reduce the work of data generation in the future.

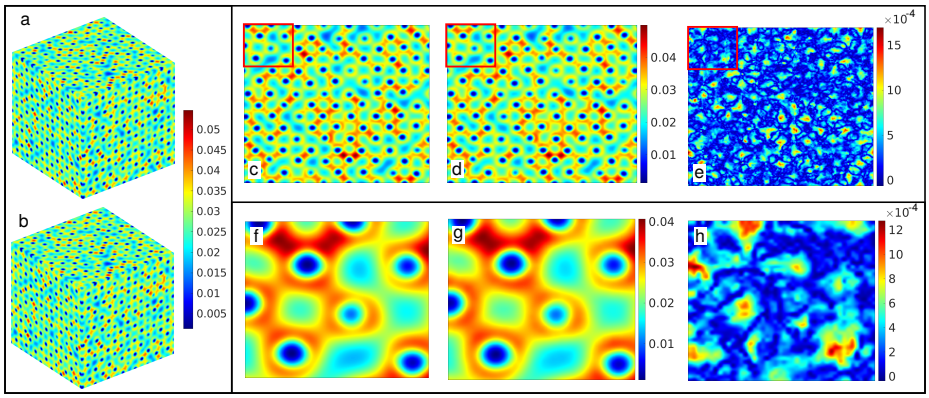
To predict the electron density at a given point, the ML model encodes the local atomic neighborhood information in the form of descriptors, that are then fed as inputs to the BNN. Our neighborhood descriptors are rather simple: they include distance and angle information from nearby atoms in the form of scalar products and avoid choosing the basis set and "handcrafted" descriptors adopted by other workers [167, 190, 191, 192, 193]. Additionally, we have carried out a systematic algorithmic procedure to select the optimal set of descriptors, thus effectively addressing the challenge associated with the high dimensionality of the descriptor-space . We explain this feature selection process in section 3.3.4. To sample this descriptor space effectively, we have employed thermalization, i.e., ab initio molecular dynamics (AIMD) simulations at various temperatures, which has allowed us to carry out accurate predictions

for systems far from training. Overall, our ML model reduces the use of heuristics adopted by previous workers in notable ways, making the process of ML based prediction of electronic structure much more systematic. Notably, the point-wise prediction of the electronic fields via the trained ML model, make this calculation scale linearly with respect to the system size, enabling a wide variety of calculations across scales.

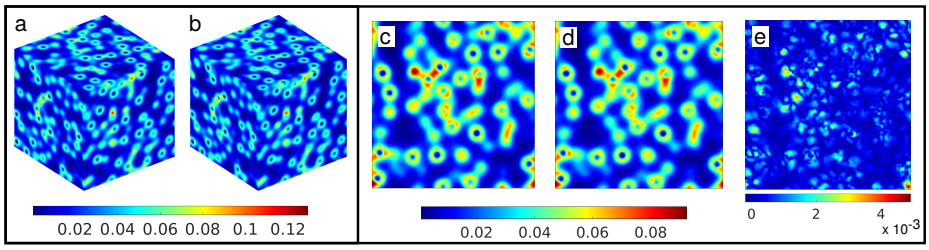
In the following sections, we demonstrate the effectiveness of our model by predicting the ground state electron density for bulk metallic and semiconducting alloy systems. In particular, we present: (i) Predictions and error estimates for systems well beyond the training data, including systems with defects and varying alloy compositions; (ii) Demonstration of the effectiveness of the transfer learning approach; (iii) Uncertainty quantification capabilities of the model, and the decomposition of the uncertainty into epistemic and aleatoric parts; and (iv) Computational advantage of the model over conventional KS-DFT calculations, and the use of the model to predict the electron density of systems containing millions of atoms.

3.2 Results

In this section, we present electron density predictions by the proposed machine learning (ML) model for two types of bulk materials — pure aluminum and alloys of silicon-germanium. These serve as prototypical examples of metallic and covalently



(a) 1372 atom aluminum simulation cell at 631 K

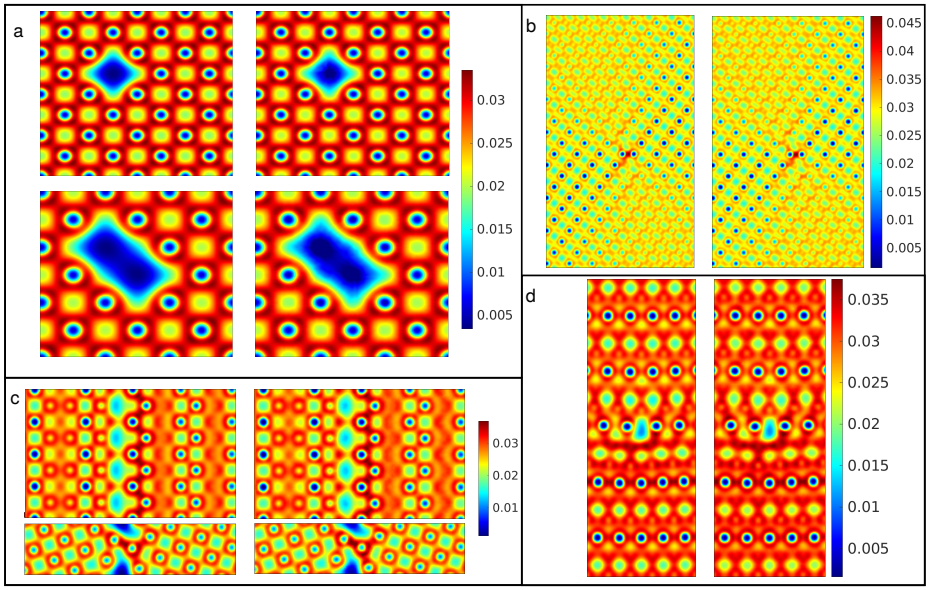


(b) 512 atoms $Si_{0.5}Ge_{0.5}$ simulation cell at 2300 K

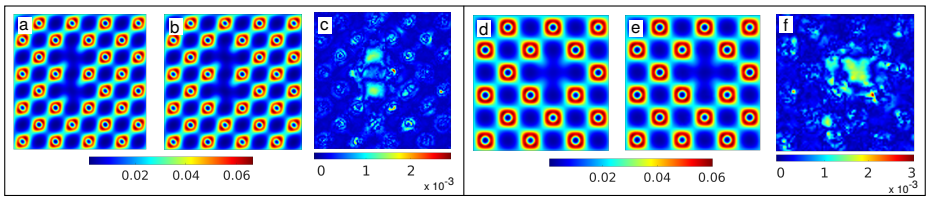
Figure 3.2: Electron densities (a) calculated by DFT and (b) predicted by ML. The two-dimensional slice of (b) that has the highest mean squared error, as calculated by (c) DFT and predicted by (d) ML. (e) Corresponding absolute error in ML with respect to DFT. (i(f)-i(h)) Magnified view of the rectangular areas in (i(c)-i(e)) respectively. The unit for electron density is e/Bohr^3, where e denotes the electronic charge.

bonded semiconducting systems, respectively. These materials were chosen for their technological importance and because the nature of their electronic fields is quite distinct (see Fig. B.4 in the supplemental material), thus presenting distinct challenges to the ML model. Additionally, being metallic, the aluminum systems do not show simple localized electronic features often observed in insulators [194, 195], further complicating electron density prediction.

The overview of the present ML model is given in Fig. 3.1. The models are trained



(a) Aluminum system with defects



(b) SiGe system with defects

Figure 3.3: (i) Electron density contours for aluminum systems with localized and extended defects — Left: calculated by DFT, Right: predicted by ML. (i.a) (Top) Mono-vacancy in 256 atom aluminum system, (Bottom) Di-Vacancy in 108 atom aluminum system, (i.b) (1 1 0) plane of a perfect screw dislocation in aluminum with Burgers vector $\frac{a_0}{2}[110]$, and line direction along [110]. The coordinate system was aligned along $[1\bar{1}2]$–$[\bar{1}11]$–[110], (i.c) (Top) (0 1 0) plane, (bottom) (0 0 1) plane of a [001] symmetric tilt grain boundary (0 inclination angle) in aluminum, (i.d) Edge dislocation in aluminum with Burgers vector $\frac{a_0}{2}[110]$. The coordinate system was aligned along [110]–$[\bar{1}11]$–$[1\bar{1}2]$ and the dislocation was created by removing a half-plane of atoms below the glide plane. (ii) Electron density contours and absolute error in ML for SiGe systems with ii(a-c) Si double vacancy defect in 512 atom system ii(d-f) Ge single vacancy defect in 216 atom system. Densities ii(a,d) calculated by DFT, ii(b,e) predicted by ML, and ii(c,f) error in ML predictions. Note that the training data for the above systems did not include any defects. The unit for electron density is e/Bohr^3, where e denotes the electronic charge.

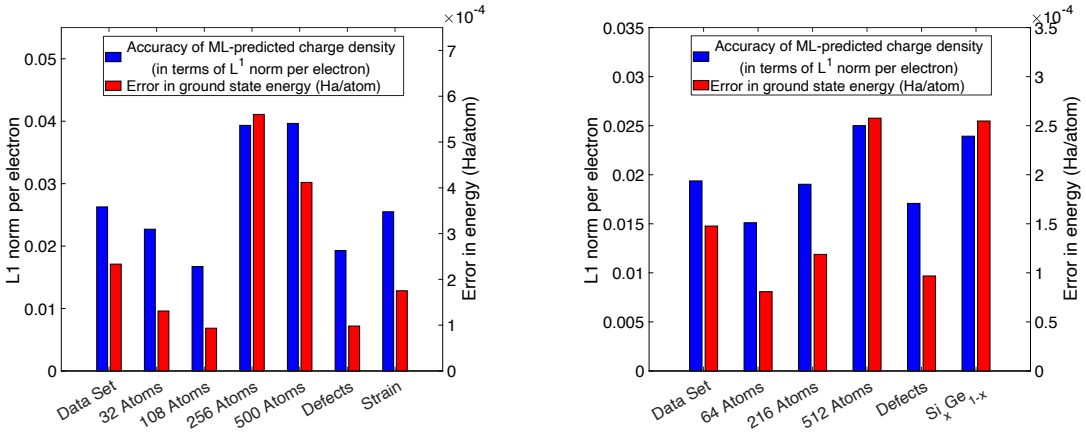


Figure 3.4: A comparison of the accuracy in the prediction of the charge density (in terms of the L^1 norm per electron between ρ^{DFT} and ρ^{scaled}), and the error (in Ha/atom) in the ground state total energy computed using ρ^{DFT} and ρ^{scaled}, for Al (left), and SiGe (right) systems. ρ^{scaled} is the scaled ML predicted electron density as given in Eq. 3.6. We observe that the errors are far better than chemical accuracy, i.e., errors below 1 kcal/mol or 1.6 milli-Hartree/atom, for both systems, even while considering various types of defects and compositional variations. Note that for Si_xGe_{1-x}, we chose $x = 0.4, 0.45, 0.55, 0.6$.

using a transfer learning approach, with thermalization used to sample a variety of system configurations. In the case of aluminum (Al), the model is trained initially on a 32-atom and subsequently on a 108-atom system. Corresponding system sizes for silicon germanium (SiGe) are 64 and 216 atoms respectively. Details of the ML model are provided in section 3.3.

We evaluate the performance of the ML models for a wide variety of test systems, which are by choice, well beyond the training data. This is ensured by choosing system sizes far beyond training, strained systems, systems containing defects, or alloy compositions not included in the training. We assess the accuracy of the ML models by comparing predicted electron density fields and ground state energies against DFT

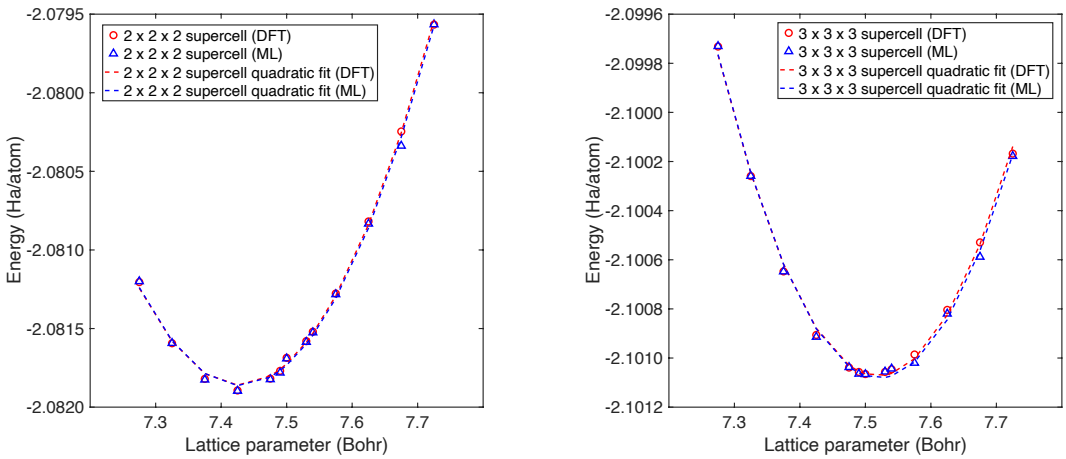
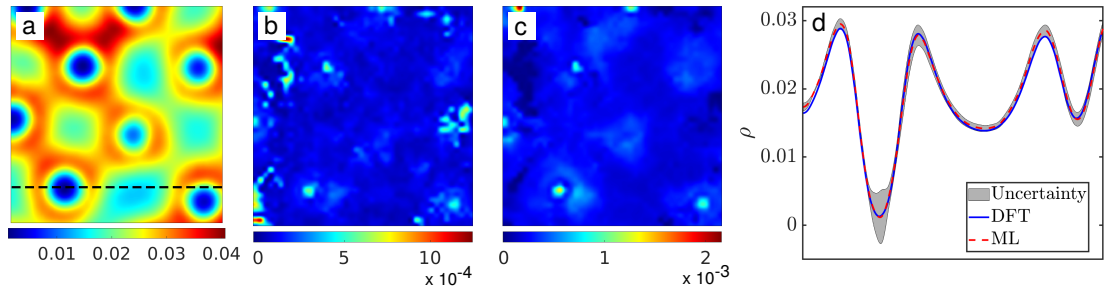
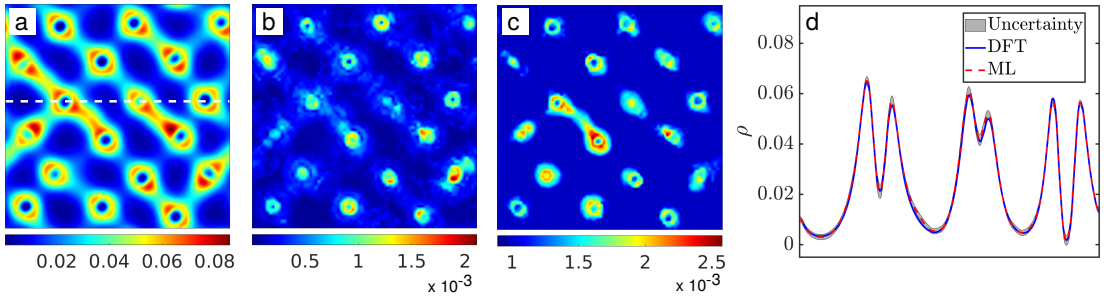


Figure 3.5: The energy curve with respect to different lattice parameters for a $2\times2\times2$ (left) and $3\times3\times3$ (right) supercell of aluminum atoms. Overall, we see excellent agreement in the energies (well within chemical accuracy). The lattice parameter (related to the first derivative of the energy plot) calculated in each case agrees with the DFT-calculated lattice parameter to $\mathcal{O}(10^{-2})$ Bohr or better (i.e., it is accurate to a fraction of a percent). The bulk modulus calculated (related to the second derivative of the energy plot) from DFT data and ML predictions agree to within 1%. For the $3 \times 3 \times 3$ supercell, the bulk modulus calculated via DFT calculations is 76.39 GPa, close to the experimental value of about 76 GPa [1]. The value calculated from ML predictions is 75.80 GPa.

simulations. In addition, we quantify the uncertainty in the model's predictions. We decompose the total uncertainty into two parts: "aleatoric" and "epistemic". The first is a result of inherent variability in the data, while the second is a result of insufficient knowledge about the model parameters due to limited training data. The inherent variability in the data might arise due to approximations and round-off errors incurred in the DFT simulations and calculation of the ML model descriptors. On the other hand, the modeling uncertainty arises due to the lack of or incompleteness in the data. This lack of data is inevitable since it is impossible to exhaustively sample all possible atomic configurations during the data generation process. Decomposing the



(a) 1372 atom aluminum system



(b) $Si_{0.4}Ge_{0.6}$ system

Figure 3.6: Uncertainty quantification for aluminum and SiGe systems. (a) ML prediction of the electron density, (b) Epistemic Uncertainty (c) Aleatoric Uncertainty (d) Total Uncertainty shown along the dotted line from the ML prediction slice. The uncertainty represents the bound $\pm 3\sigma_{total}$, where, σ_{total} is the total uncertainty. The unit for electron density is e/Bohr^3, where e denotes the electronic charge.

total uncertainty into these two parts helps distinguish the contributions of inherent randomness and incompleteness in the data to the total uncertainty. In the present work, a "heteroscedastic" noise model is used to compute the aleatoric uncertainty, which captures the spatial variation of the noise/variance in the data.

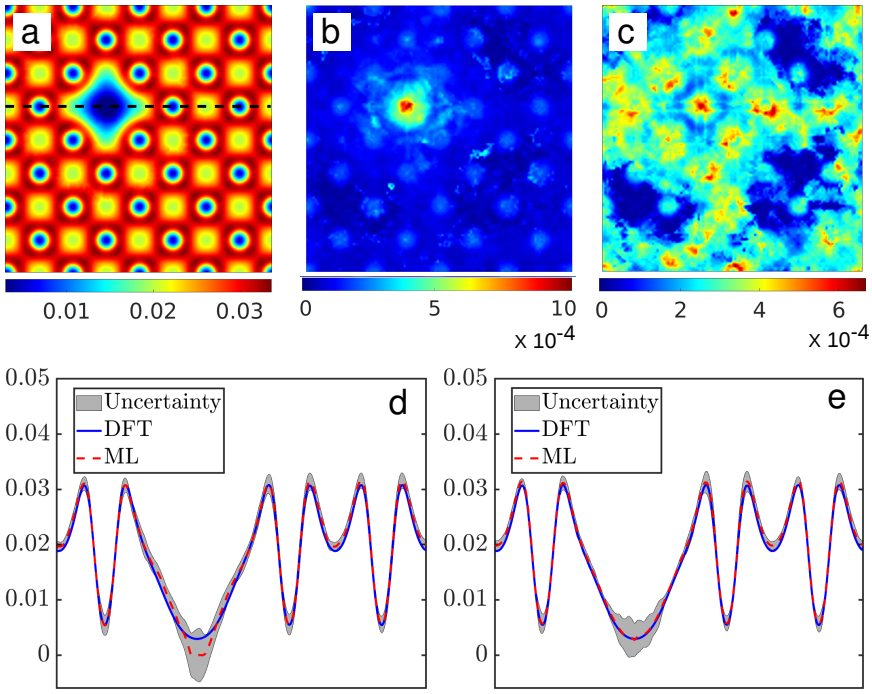


Figure 3.7: Uncertainty quantification for a 256 atom aluminum system with a mono vacancy defect. (a) ML prediction of the electron density shown on the defect plane, (b)) Epistemic uncertainty (c) Aleatoric uncertainty d) Uncertainty shown along the black dotted line from the ML prediction slice. The uncertainty represents the bound $\pm 3\sigma_{total}$, where, σ_{total} is the total uncertainty. Note that the model used to make the predictions in (a-d) is not trained on the defect data, as opposed to the model used for (e), where defect data from the 108 atom aluminum system was used to train the model. The uncertainty and error at the location of the defect reduce with the addition of defect data in the training, as evident from (d) and (e). The unit for electron density is e/Bohr^3, where e denotes the electronic charge.

3.2.1 Error Estimation

To evaluate the accuracy of the model, we calculated the Root Mean Squared Error (RMSE) for the entire test dataset, including systems of the same size as the training

data as well as sizes bigger than training data. For aluminum, the RMSE was determined to be 4.1×10^{-4}, while for SiGe, it was 7.1×10^{-4}, which shows an improvement over RMSE values for Al available in [44]. The L^1 norm per electron for Aluminuum is 2.63×10^{-2} and for SiGe it is 1.94×10^{-2} for the test dataset. Additionally, the normalized RMSE is obtained by dividing the RMSE value by the range of respective ρ values for aluminum and SiGe. The normalized RMSE for aluminum and SiGe test dataset was found to be 7.9×10^{-3} for both materials. Details of training and test dataset are presented in SM section B.6. To assess the generalizability of the model, we evaluate the accuracy of the ML model using systems much larger than those used in training, but accessible to DFT. We consider two prototypical systems, an Aluminium system having 1372 atoms (Fig. 3.2(a)) and a Silicon Germanium ($Si_{0.5}\,Ge_{0.5}$) system having 512 atoms (Fig. 3.2(b)). The model shows remarkable accuracy for both of these large systems. The RMSE is 3.8×10^{-4} and 7.1×10^{-4} for aluminum and SiGe respectively, which confirms the high accuracy of the model for system sizes beyond those used in training.

We now evaluate the performance of the ML model for systems containing extended and localized defects, although such systems were not used in training. We consider the following defects: mono-vacancies, di-vacancies, grain boundaries, edge, and screw dislocations for Al, and mono-vacancies and di-vacancies for SiGe. The electron density fields predicted by the ML models match with the DFT calculations extremely well, as shown in Figs. 3.3(a) and 3.3(b). The error magnitudes (measured as the

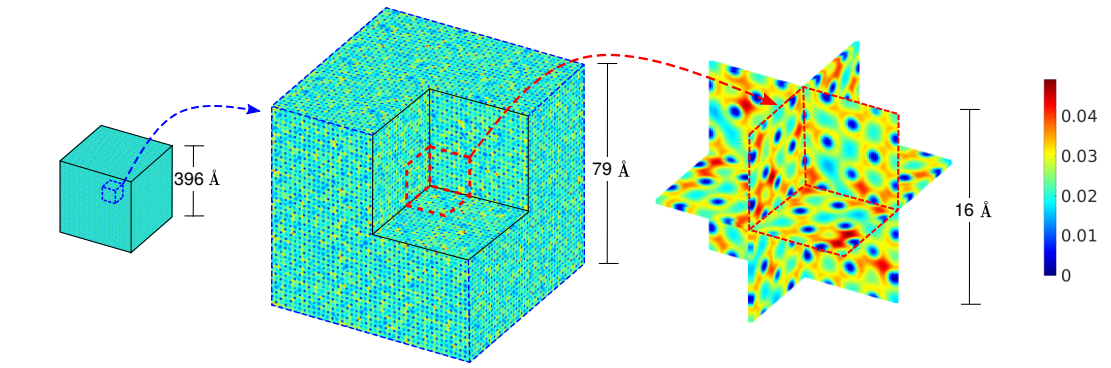
L^1 norm of the difference in electron density fields, per electron) are about 2×10^{-2} (see Fig. 3.4). The corresponding NRMSE is 7.14×10^{-3} . We show in Section 3.2.2, that the model errors and uncertainty can be both brought down significantly, by including a single snapshot with defects, during training.

Another stringent test of the generalizability of the ML models is performed by investigating $Si_x\, Ge_{1-x}$ alloys, for $x \neq 0.5$. Although only equi-atomic alloy compositions (i.e., $x = 0.5$) were used for training, the error in prediction (measured as the L^1 norm of the difference in electron density fields, per electron) is lower than 3×10^{-2} (see Fig. 3.4). The corresponding RMSE is 8.04×10^{-4} and NRMSE is 7.32×10^{-3} . We would like to make a note that we observed good accuracy in the immediate neighborhood ($x = 0.4$ to 0.6) of the training data ($x = 0.5$). Prediction for $x = 0.4$ is shown in Fig. 6(ii). The prediction accuracy however decreases as we move far away from the training data composition. This generalization performance far away from the training data is expected. We have also carried out tests with aluminum systems subjected to volumetric strains, for which the results were similarly good.

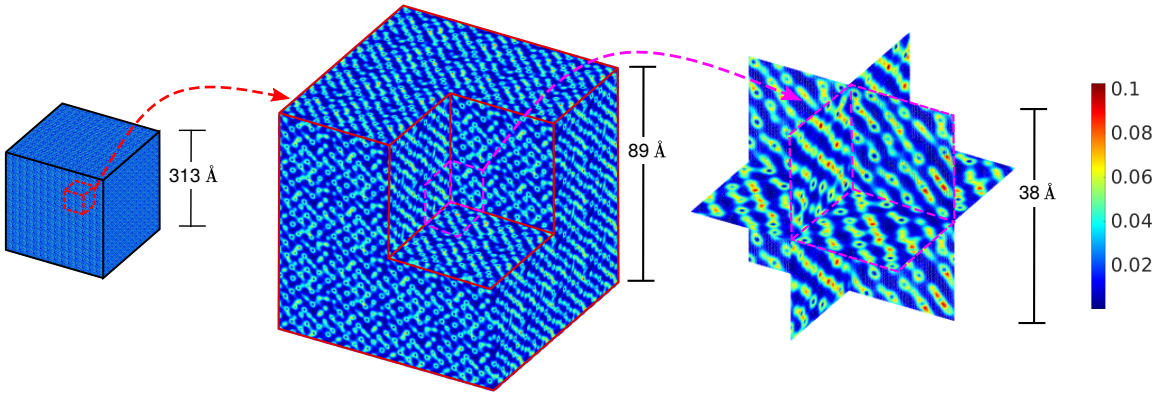
Our electron density errors are somewhat lower than compared to the earlier works [44, 48], At the same time, thanks to the sampling and transfer learning techniques adopted by us, the amount of time spent on DFT calculations used for producing the training data is also smaller. To further put into context the errors in the electron density, we evaluate the ground state energies from the charge densities predicted by

the ML model through a postprocessing step and compare these with the true ground state energies computed via DFT. Details on the methodology for postprocessing can be found in the 'Methods' section, and a summary of our postprocessing results can be seen in Fig. 3.4, and in Tables B.4 and B.5, in the supplemental material. On average, the errors are well within chemical accuracy for all test systems considered and are generally $\mathcal{O}(10^{-4})$ Ha/atom, as seen in Fig. 3.4. Furthermore, not only are the energies accurate, but the derivatives of the energies, e.g., with respect to the supercell lattice parameter, are found to be quite accurate as well (see Fig. 3.5). This enables us to utilize the ML model to predict the optimum lattice parameter — which is related to the first derivative of the energy curve, and the bulk modulus — which is related to the second derivative of the energy curve, accurately. We observe that the lattice parameter is predicted accurately to a fraction of a percent, and the bulk modulus is predicted to within 1% of the DFT value (which itself is close to experimental values [1]). Further details can be found in the supplemental material. This demonstrates the utility of the ML models to predict not only the electron density but also other relevant physical properties.

Overall, the generalizability of our models is strongly suggestive that our use of thermalization to sample the space of atomic configurations, and the use of transfer-learning to limit training data generation of large systems are both very effective. We discuss uncertainties arising from the use of these strategies and due to the neural network model, in addition to the noise in the data, in the following sections.



(a) Aluminum



(b) SiGe

Figure 3.8: Prediction of electronic structure for aluminum system containing ≈ 4.1 million and $Si_{0.5}Ge_{0.5}$ system containing ≈ 1.4 million atoms. The unit for electron density is e/Bohr^3, where e denotes the electronic charge.

3.2.2 Uncertainty quantification

The present work uses a Bayesian Neural Network (BNN) which provides a systematic route to uncertainty quantification (UQ) through its stochastic parameters as opposed to other methods for UQ, for instance ensemble averaging [196]. Estimates of epistemic and aleatoric uncertainties for the following systems are shown: a defect-free Al system with 1372 atoms (Fig. 3.6(a)), a 256-atom Al system with a mono-vacancy

(Fig. 3.7(a-d)), and a $Si_{0.4}Ge_{0.6}$ alloy (Fig. 3.6(b)). Note, for the results in Fig. 3.7(a-d) the training data does not contain any systems having defects, and for the results in Fig. 3.6(b) the training data contains only $50 - 50$ composition.

In these systems, the aleatoric uncertainty has the same order of magnitude as the epistemic uncertainty. This implies that the uncertainty due to the inherent randomness in the data is of a similar order as the modeling uncertainty. The aleatoric uncertainty is significantly higher near the nuclei (Fig. 3.6(a) and Fig. 3.6(b)) and also higher near the vacancy (Fig 3.7). This indicates that the training data has high variability at those locations. The epistemic uncertainty is high near the nucleus (Fig. 3.6(a) and Fig. 3.6(b)) since only a small fraction of grid points are adjacent to nuclei, resulting in the scarcity of training data for such points. The paucity of data near a nucleus is shown through the distribution of electron density in Fig. B.4 of the supplemental material. For the system with vacancy, the aleatoric uncertainty is higher in most regions, as shown in Fig. 3.7(c). However, the epistemic uncertainty is significantly higher only at the vacancy (Fig. 3.7(b)), which might be attributed to the complete absence of data from systems with defects in the training.

To investigate the effect of adding data from systems with defects in the training, we added a single snapshot of 108 atom aluminum simulation with mono vacancy defect to the training data. This reduces the error at the defect site significantly and also reduces the uncertainty (Fig. 3.7(e)). However, the uncertainty is still quite higher at

the defect site because the data is biased against the defect site. That is, the amount of training data available at the defect site is much less than the data away from it. Thus, this analysis distinguishes uncertainty from inaccuracy.

To investigate the effect of adding data from larger systems in training, we compare two models. The first model is trained with data from the 32-atom system. The second model uses a transfer learning approach where it is initially trained using the data from the 32-atom system and then a part of the model is retrained using data from the 108-atom system. We observe a significant reduction in the error and in the epistemic uncertainty for the transfer learned model as compared to the one without transfer learning. The RMSE on the test system (256 atom) decreases by 50% when the model is transfer learned using 108 atom data. The addition of the 108-atom system's data to the training data decreases epistemic uncertainty as well since the 108-atom system is less restricted by periodic boundary conditions than the 32-atom system. Further, it is also statistically more similar to the larger systems used for testing as shown in Fig. B.5 of the supplemental material. These findings demonstrate the effectiveness of the Bayesian Neural Network in pinpointing atomic arrangements or physical sites where more data is essential for enhancing the ML model's performance. Additionally, they highlight its ability to measure biases in the training dataset. The total uncertainty in the predictions provides a confidence interval for the ML prediction. This analysis provides an upper bound of uncertainty arising out of two key heuristic strategies adopted in our ML model: data generation

through thermalization of the systems and transfer learning.

3.2.3 Computational efficiency gains and confident prediction for unprecedented system sizes

Conventional KS-DFT calculations scale as $\mathcal{O}({N_a}^3)$ with respect to the number of atoms N_a, whereas, our ML model scales linearly (i.e., $\mathcal{O}(N_a)$), as shown in Fig. 3.9. This provides computational advantage for ML model over KS-DFT with increasing number of atoms. For example, even with 500 atoms, the calculation wall times for ML model is 2 orders of magnitude lower than KS-DFT. The linear scaling behavior of the ML model with respect to the number of atoms can be understood as follows. As the number of atoms within the simulation domain increases, so does the total simulation domain size, leading to a linear increase in the total number of grid points (keeping the mesh size constant, to maintain calculation accuracy). Since the machine learning inference is performed for each grid point, while using information from a fixed number of atoms in the local neighborhood of the grid point, the inference time is constant for each grid point. Thus the total ML prediction time scales linearly with the total number of grid points, and hence the number of atoms in the system.

Taking advantage of this trend, the ML model can be used to predict the electronic structure for system sizes far beyond the reach of conventional calculation techniques,

including systems containing millions of atoms, as demonstrated next. We anticipate that with suitable parallel programming strategies (the ML prediction process is embarrassingly parallel) and computational infrastructure, the present strategy can be used to predict the electronic structure of systems with hundreds of millions or even billions of atoms. Very recently, there have been attempts in the direction of making predictions at an unprecedented scale. In [197], a machine learning based potential is developed for germanium–antimony–tellurium alloys, effectively working for device scale systems containing over half a million atoms. Another contribution comes from Fiedler et al. [174], where they present a model predicting electronic structure for systems containing over 100,000 atoms.

We show the electron densities, as calculated by our ML model, for a four million atom system of Al and a one million atom system of SiGe, in Figs. 3.8(a) and 3.8(b) respectively. In addition to predicting electron densities, we also quantify uncertainties for these systems. We found that the ML model predicts larger systems with equally high certainty as smaller systems (see Fig. B.3 of supplemental material). The confidence interval obtained by the total uncertainty provides a route to assessing the reliability of predictions for these million atom systems for which KS-DFT calculations are simply not feasible. A direct comparison of ML obtained electron density with DFT for large systems is not done till date, mainly because simulating such systems with DFT is impractical. However, recent advancements in DFT techniques hold promise for simulating large-scale systems [152, 198, 199]. In future, it

will be worthwhile to compare ML predicted electron density for large systems and the electron density obtained through DFT, utilizing these recently introduced DFT techniques.

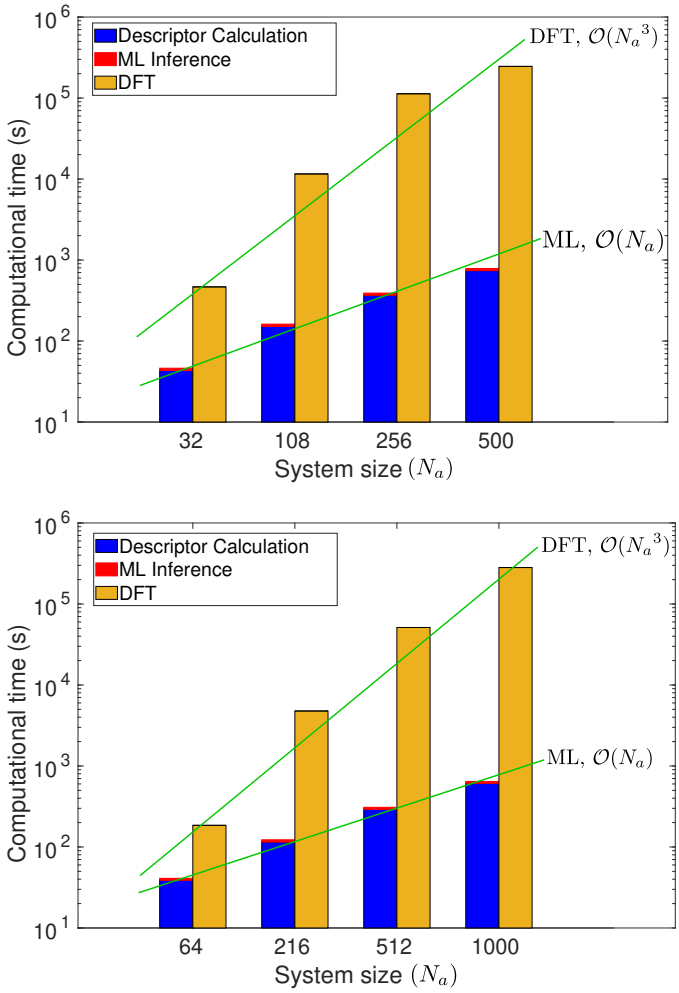


Figure 3.9: Computational time comparison between DFT calculations and prediction via trained ML model. (Top) Aluminum, (Bottom) SiGe. The DFT calculations scale $\mathcal{O}(N_a{}^3)$ with respect to the system size (number of atoms N_a), whereas, the present ML model scales linearly (i.e., $\mathcal{O}(N_a)$). The time calculations were performed using the same number of CPU cores and on the same system (Perlmutter CPU).

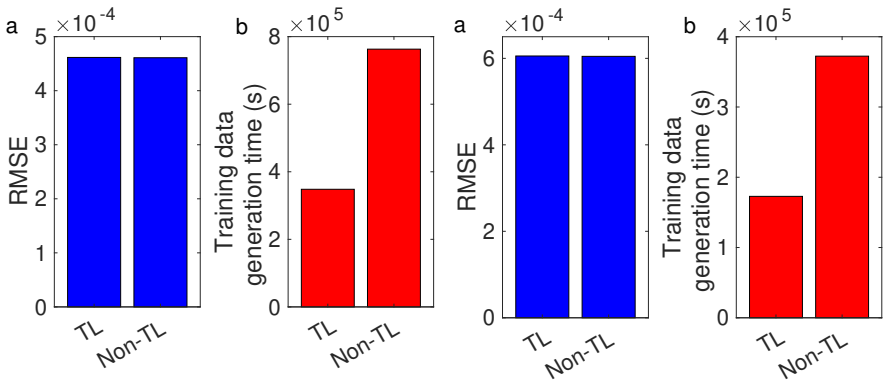
3.2.4 Reduction of training data generation cost via transfer learning

One of the key challenges in developing an accurate ML model for electronic structure prediction is the high computational cost associated with the generation of the training data through KS-DFT, especially for predicting the electron density for systems across length-scales. A straightforward approach would involve data generation using sufficiently large systems wherein the electron density obtained from DFT is unaffected by the boundary constraints. However, simulations of larger bulk systems are significantly more expensive than smaller systems. To address the computational burden of simulating large systems, strategies such as "fragmentation" have been used in electronic structure calculations [200, 201]. Further, certain recent studies on Machine Learning Interatomic Potentials suggest utilizing portions of a larger system for training the models [202, 203]. To the best of our knowledge, there is no corresponding work that utilizes fragmentation in ML modeling of the electron density. In this work, to address the issue, we employed a transfer learning (TL) approach. We first trained the ML model on smaller systems and subsequently trained a part of the neural network using data from larger systems. This strategy allows us to obtain an efficient ML model that requires fewer simulations of expensive large-scale systems compared

to what would have been otherwise required without the TL approach. The effectiveness of the TL approach stems from its ability to retain information from a large quantity of cheaper, smaller scale simulation data. We would like to note however, that the transfer learning approach is inherently bound by the practical constraints associated with simulating the largest feasible system size.

As an illustration of the above principles, we show in Fig. 3.10, the RMSE obtained on 256 atom data (system larger than what was used in the training data) using the TL model and the non-TL model. We also show the time required to generate the training data for both models. For the Al systems, we trained the TL model with 32-atom data first and then 108-atom data. In contrast, the non-TL model was trained only on the 108-atom data.

The non-TL model requires significantly more 108-atom data than the TL model to achieve a comparable RMSE on the 256-atom dataset. Moreover, the TL model's training data generation time is approximately 55% less than that of the non-TL model. This represents a substantial computational saving in developing the ML model for electronic structure prediction, making the transfer learning approach a valuable tool to expedite such model development. Similar savings in training data generation time were observed for SiGe as shown in Fig. 3.10. In the case of SiGe, the TL model was first trained using 64 atom data and then transfer learned using 216 atom data.



(a) Aluminum (b) SiGe

Figure 3.10: Models with Transfer Learning (TL) and without Transfer Learning (Non-TL): (a) Root mean square error (RMSE) on the test dataset and (b) Computational time to generate the training data. In the case of aluminum, the TL model is trained using 32 and 108 atom data. For SiGe, the TL model was trained using 64 and 216 atom data. In the case of aluminum, the non-TL model is trained using 108 atom data. Whereas, in the case of SiGe, the non-TL model is trained using 216 atom data.

3.3 Methods

3.3.1 *Ab Initio* Molecular Dynamics

To generate training data for the model, *Ab Initio* Molecular Dynamics (AIMD) simulations were performed using the finite-difference based SPARC code [204, 205, 206]. We used the GGA PBE exchange-correlation functional [207] and ONCV pseudopotentials [208]. For aluminum, a mesh spacing of 0.25 Bohrs was used while for SiGe, a mesh spacing of 0.4 Bohrs was used. These parameters are more than sufficient to

produce accurate energies and forces for the pseudopotentials chosen, as was determined through convergence tests. A tolerance of 10^{-6} was used for self-consistent field (SCF) convergence and the Periodic-Pulay [87] scheme was deployed for convergence acceleration. These parameters and pseudopotential choices were seen to produce the correct lattice parameters and bulk modulus values for the systems considered here, giving us confidence that the DFT data being produced is well rooted in the materials physics.

For AIMD runs, a standard NVT-Nosé Hoover thermostat [209] was used, and Fermi-Dirac smearing at an electronic temperature of 631.554 K was applied. The time step between successive AIMD steps was 1 femtosecond. The atomic configuration and the electron density of the system were captured at regular intervals, with sufficient temporal spacing between snapshots to avoid the collection of data from correlated atomic arrangements. To sample a larger subspace of realistic atomic configurations, we performed AIMD simulations at temperatures ranging from 315 K to about twice the melting point of the system, i.e. 1866 K for Al and 2600 K for SiGe. Bulk disordered SiGe alloy systems were generated by assigning atoms randomly to each species, consistent with the composition.

We also generate DFT data for systems with defects and systems under strain, in order to demonstrate the ability of our ML model to predict unseen configurations. To this end, we tested the ML model on monovacancies and divacancies, edge and

screw dislocations, and grain boundaries. For vacancy defects, we generated monovacancies by removing an atom from a random location, and divacancies by removing two random neighboring atoms before running AIMD simulations. Edge and screw dislocations for aluminum systems were generated using Atomsk [210]. Further details can be found in Fig. 3.3(a). Grain boundary configurations were obtained based on geometric considerations of the tilt angle — so that an overall periodic supercell could be obtained, and by removing extra atoms at the interface. For aluminum, we also tested an isotropic lattice compression and expansion of up to 5%; these systems were generated by scaling the lattice vectors accordingly (while holding the fractional atomic coordinates fixed).

3.3.2 Machine learning map for charge density prediction

Our ML model maps the coordinates $\{\mathbf{R}_I\}_{I=1}^{N_a}$ and species (with atomic numbers $\{Z_I\}_{I=1}^{N_a}$) of the atoms, and a set of grid points $\{\mathbf{r}_i\}_{i=1}^{N_{\text{grid}}}$ in a computational domain, to the electron density values at those grid points. Here, N_a and N_{grid} refer to the number of atoms and the number of grid points, within the computational domain, respectively. We compute the aforementioned map in two steps. *First*, given the atomic coordinates and species information, we calculate atomic neighborhood descriptors for each grid point. *Second*, a neural network is used to map the descriptors to the electron density at each grid point. These two steps are discussed in more

detail subsequently.

3.3.3 Atomic neighborhood descriptors

In this work, we use a set of scalar product-based descriptors to encode the local atomic environment. The scalar product-based descriptors for the grid point at $\mathbf{r}_i$ consist of distance between the grid point and the atoms at $\mathbf{R}_I$; and the cosine of angle at the grid point $\mathbf{r}_i$ made by the pair of atoms at $\mathbf{R}_I$ and $\mathbf{R}_J$. Here $i = 1, \ldots, N_{\text{grid}}$ and $I, J = 1, \ldots, N_a$. We refer to the collections of distances i.e., $||\mathbf{r}_i - \mathbf{R}_I||$ as set I descriptors, and the collections of the cosines of the angles i.e., $\frac{(\mathbf{r}_i - \mathbf{R}_I)\cdot(\mathbf{r}i - \mathbf{R}_J)}{||\mathbf{r}_i - \mathbf{R}_I||\,||\mathbf{r}i - \mathbf{R}_J||}$ are referred to as set II descriptors.

Higher order scalar products such as the scalar triple product, and the scalar quadruple product which involve more than two atoms at a time can also be considered. However, these additional scalar products are not included in the descriptor set in this work since they do not appear to increase the accuracy of predictions as elaborated in the supplemental material.

3.3.4 Selection of optimal set of descriptors

As has been pointed out by previous work on ML prediction of electronic structure [44, 48], the nearsightedness principle [195, 211] and screening effects [212] indicate that the electron density at a grid point has little influence from atoms sufficiently far away. This suggests that only descriptors arising from atoms close enough to a grid point need to be considered in the ML model, a fact which is commensurate with our findings in Fig. 3.11.

Using an excessive number of descriptors can increase the time required for descriptor-calculation, training, and inference, is susceptible to curse of dimensionality, and affect prediction performance [75, 213, 214, 215]. On the other hand, utilizing an insufficient number of descriptors can result in an inadequate representation of the atomic environments and lead to an inaccurate ML model.

Based on this rationale, we propose a procedure to select an optimal set of descriptors for a given atomic system. We select a set of M ($M \leq N_a$) nearest atoms from the grid point to compute the descriptors and perform a convergence analysis to strike a balance between the aforementioned conditions to determine the optimal value of M. It is noteworthy that the selection of optimal descriptors has been explored in previous works, in connection with Behler-Parinello symmetry functions such as [216]

and [217]. These systematic procedures for descriptor selection eliminate trial-and-error operations typically involved in finalizing a descriptor set. In [217], the authors have demonstrated for Behler-Parinello symmetry functions that using an optimal set of descriptors enhances the efficiency of machine learning models.

For M nearest atoms, we will have $N_{\text{set I}}$ distance descriptors, and $N_{\text{set II}}$ angle descriptors, with $N_{\text{set I}} = M$ and $N_{\text{set II}} \leq {}^{M}C_2$.

The total number of descriptors is $N_{\text{desc}} = N_{\text{set I}} + N_{\text{set II}}$. To optimize N_{desc}, we first optimize $N_{\text{set I}}$, till the error converges as shown in Fig. 3.11. Subsequently, we optimize $N_{\text{set II}}$. To do this, we consider a nearer subset of atoms of size $M_a \leq M$, and for each of these M_a atoms, we consider the angle subtended at the grid point, by the atoms and their k nearest neighbors. This results in $N_{\text{set II}} = M_a \times k$, angle based descriptors, with M_a and k varied to yield the best results, as shown in Fig. 3.11. The pseudo-code for this process can be found in Algorithms 2 and 3 in the supplemental material. Further details on feature convergence analysis are provided in the supplemental material.

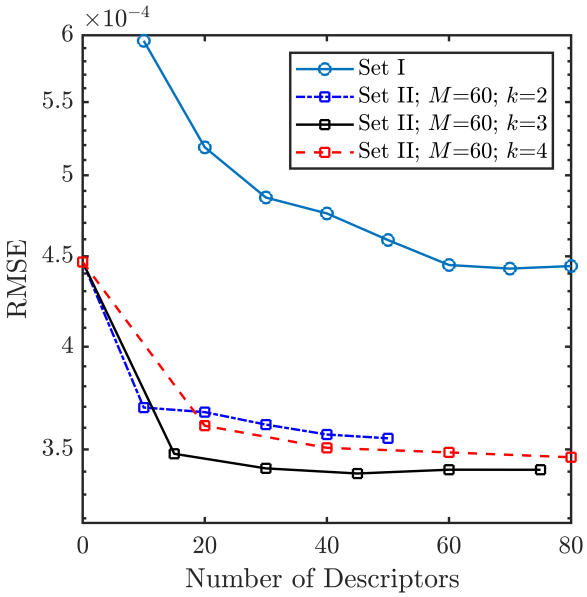


Figure 3.11: Convergence of error with respect to the number of descriptors, shown for aluminum. The blue line shows the convergence with respect to $N_{\text{set I}}$, while the other three lines show convergence with respect to $N_{\text{set II}}$. The optimal $N_{\text{set I}}$ and $N_{\text{set II}}$ are obtained where their test RMSE values converge.

3.3.5 Bayesian Neural Network

Bayesian Neural Networks (BNNs) have stochastic parameters in contrast to deterministic parameters used in conventional neural networks. BNNs provide a mathematically rigorous and efficient way to quantify uncertainties in their prediction.

We use a Bayesian neural network to estimate the probability $P(\rho|\mathbf{x}, \mathcal{D})$ of the output electron density ρ for a given input descriptor $\mathbf{x} \in \mathbb{R}^{N_{\text{desc}}}$ and training data set

$\mathcal{D} = \{\mathbf{x}_i, \rho_i\}_{i=1}^{N_d}$. The probability is evaluated as:

$$P(\rho|\mathbf{x}, \mathcal{D}) = \int_{\Omega_w} P(\rho|\mathbf{x}, \mathbf{w}) P(\mathbf{w}|\mathcal{D}) d\mathbf{w} \,. \tag{3.1}$$

Here $\mathbf{w} \in \Omega_w$ is the set of parameters of the network and N_d is the size of the training data set. Through this marginalization over parameters, a BNN provides a route to overcome modeling biases via averaging over an ensemble of networks. Given a prior distribution $P(\mathbf{w})$ on the parameters, the posterior distribution of the parameters $P(\mathbf{w}|\mathcal{D})$ are learned via the Bayes' rule as $P(\mathbf{w}|\mathcal{D}) = P(\mathcal{D}|\mathbf{w})P(\mathbf{w})/P(\mathcal{D})$, where $P(\mathcal{D}|\mathbf{w})$ is the likelihood of the data.

This posterior distribution of parameters $P(\mathbf{w}|\mathcal{D})$ is intractable since it involves the normalizing factor $P(\mathcal{D})$, which in turn is obtained via marginalization of the likelihood through a high dimensional integral. Therefore, it is approximated through techniques such as variational inference [187, 218, 219] or Markov Chain Monte Carlo methods [220]. In variational inference, as adopted here, a tractable distribution $q(\mathbf{w}|\boldsymbol{\theta})$ called the "variational posterior" is considered, which has parameters $\boldsymbol{\theta}$. For instance, if the variational posterior is a Gaussian distribution the corresponding parameters are its mean and standard deviation, $\boldsymbol{\theta} = (\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta)$. The optimal value of parameters $\boldsymbol{\theta}$ is obtained by minimizing the statistical dissimilarity between the true and variational posterior distributions. The dissimilarity is measured through the KL divergence $\mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \,||\, P(\mathbf{w}|\mathcal{D})\right]$. This yields the following optimization problem:

$$\begin{aligned}\boldsymbol{\theta}^* &= \arg\min_{\boldsymbol{\theta}} \mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \,||\, P(\mathbf{w}|\mathcal{D})\right] \\ &= \arg\min_{\boldsymbol{\theta}} \int q(\mathbf{w}|\boldsymbol{\theta}) \log\left[\frac{q(\mathbf{w}|\boldsymbol{\theta})}{P(\mathbf{w})P(\mathcal{D}|\mathbf{w})} P(\mathcal{D})\right] d\mathbf{w}\,.\end{aligned} \tag{3.2}$$

This leads to the following loss function for BNN that has to be minimized:

$$\mathcal{F}_{KL}(\mathcal{D}, \boldsymbol{\theta}) = \mathrm{KL}\left[q(\mathbf{w}|\boldsymbol{\theta}) \,||\, P(\mathbf{w})\right] - \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\theta})}[\log P(\mathcal{D}|\mathbf{w})]\,. \tag{3.3}$$

This loss function balances the simplicity of the prior and the complexity of the data through its first and second terms respectively, yielding regularization [5, 187].

Once the parameters $\boldsymbol{\theta}$ are learned, the BNNs can predict the charge density at any new input descriptor $\mathbf{x}$. In this work, the mean of the parameters ($\boldsymbol{\mu}_\theta$) are used to make point estimate predictions of the BNN.

3.3.6 Uncertainty quantification

The variance in the output distribution $P(\rho|\mathbf{x}, \mathcal{D})$ in Eq. (3.1) is the measure of uncertainty in the BNN's prediction. Samples from this output distribution can be drawn in three steps: In the first step, a j^{th} sample of the set of parameters, $\widehat{\mathbf{w}}_{j=1,\ldots,N_s}$, is drawn from the variational posterior $q(\mathbf{w}|\boldsymbol{\theta})$ which approximates the

posterior distribution of parameters $P(\mathbf{w}|\mathcal{D})$. Here, N_s is the number of samples drawn from the variational posterior of parameters. In the second step, the sampled parameters are used to perform inference of the BNN (f_N) to obtain the j^{th} prediction $\widehat{\rho}_j = f_N^{\widehat{\mathbf{w}}_\mathbf{j}}(\mathbf{x})$. In the third step, the likelihood is assumed to be a Gaussian distribution: $P(\rho|\mathbf{x}, \widehat{\mathbf{w}}_\mathbf{j}) = \mathcal{N}(\widehat{\rho}_j, \sigma(\mathbf{x}))$, whose mean is given by the BNN's prediction, $\widehat{\rho}_j$, and standard deviation by a heterogenous observation noise, $\sigma(\mathbf{x})$. A sample is drawn from this Gaussian distribution $\mathcal{N}(\widehat{\rho}_j, \sigma(\mathbf{x}))$ that approximates a sample from the distribution $P(\rho|\mathbf{x}, \mathcal{D})$. The total variance of such samples can be expressed as:

$$\text{var}(\rho) = \sigma^2(\mathbf{x}) + \left[\frac{1}{N_s}\sum_{j=1}^{N_s}(\widehat{\rho}_j)^2 - (\mathbb{E}(\widehat{\rho}_j))^2\right]. \tag{3.4}$$

Here, $\mathbb{E}(\widehat{\rho}_j) = \frac{1}{N_s}\sum_{j=1}^{N_s} f_N^{\widehat{\mathbf{w}}_\mathbf{j}}(\mathbf{x})$. The first term, $\sigma^2(\mathbf{x})$, in Eq. (3.4) is the aleatoric uncertainty that represents the inherent noise in the data and is considered irreducible. The second term (in the square brackets) in Eq. (3.4) is the epistemic uncertainty, that quantifies the modeling uncertainty.

In this work, the aleatoric uncertainty is learned via the BNN model along with the charge densities ρ. Therefore, for each input $\mathbf{x}$, the BNN learns two outputs: $f_N^{\mathbf{w}}(\mathbf{x})$ and $\sigma(\mathbf{x})$. For a Gaussian likelihood, the noise σ is learned through the likelihood

term of the loss function Eq. (3.3) following [221] as:

$$\log P(\mathcal{D}|\mathbf{w}) = \sum_{i=1}^{N_d} -\frac{1}{2}\log\sigma_i^2 - \frac{1}{2\sigma_i^2}(f_N^{\mathbf{w}}(\mathbf{x}_i) - \rho_i)^2 \,. \tag{3.5}$$

Here, N_d is the size of the training data set. The aleatoric uncertainty, σ, enables the loss to adapt to the data. The network learns to reduce the effect of erroneous labels by learning a higher value for σ^2, which makes the network more robust or less susceptible to noise. On the other hand, the model is penalized for predicting high uncertainties for all points through the $\log\sigma^2$ term.

The epistemic uncertainty is computed by evaluating the second term of Eq.(3.4), via sampling $\widehat{\mathbf{w}}_\mathbf{j}$ from the variational posterior.

3.3.7 Transfer Learning using multi-scale data

Conventional DFT simulations for smaller systems are considerably cheaper than those for larger systems, as the computational cost scales cubically with the number of atoms present in the simulation cell. However, the ML models cannot be trained using simulation data from small systems alone. This is because, smaller systems are far more constrained in the number of atomic configurations they can adopt, thus limiting their utility in simulating a wide variety of materials phenomena. Additionally, the electron density from simulations of smaller systems differs from that of larger

systems, due to the effects of periodic boundary conditions.

To predict accurately across all length scales while reducing the cost of training data generation via DFT simulations, we use a transfer learning approach here. Transfer learning is a machine learning technique where a network, initially trained on a substantial amount of data, is later fine-tuned on a smaller dataset for a different task, with only the last few layers being updated while the earlier layers remain unaltered [15, 186]. The initial layers (called "frozen layers") capture salient features of the inputs from the large dataset, while the re-trained layers act as decision-makers and adapt to the new problem.

Transfer learning has been used in training neural network potentials, first on Density Functional Theory (DFT) data, and subsequently using datasets generated using more accurate, but expensive quantum chemistry models [222]. However, its use in predicting electronic structure, particularly, by leveraging the multi-scale aspects of the problem — as done here — is novel. Furthermore, the present transfer learning approach leverages the statistical dissimilarity in data distributions between various systems and the largest system. This process is employed to systematically select the training data, ultimately reducing reliance on heuristics, as detailed in the supplemental material (see Fig. B.5). This approach allows us to make electron density predictions across scales and system configurations, while significantly reducing the cost of training data generation.

In the case of aluminum, at first, we train the model using a large amount of data from DFT simulations of (smaller) 32-atom systems. Subsequently, we freeze the initial one-third layers of the model and re-train the remaining layers of the model using a smaller amount (40%) of data from simulations of (larger) 108-atom systems. Further training using data from larger bulk systems was not performed, since the procedure described above already provides good accuracy (Figs. 3.4,3.10), which we attribute to the statistical similarity of the electron density of 108 atom systems and those with more atoms (Fig.B.5 of the supplemental material). A similar transfer learning procedure is used for the SiGe model, where we initially train with data from 64-atom systems and subsequently retrain using data from 216-atom systems. Overall, due to the non-linear data generation cost using DFT simulations, the transfer learning approach reduces training data generation time by over 50%.

3.3.8 Postprocessing of ML predicted electron density

One way to test the accuracy of the ML models is to compute quantities of interest (such as the total ground state energy, exchange-correlation energy, and Fermi level) using the predicted electron density, ρ^{ML}. Although information about the total charge in the system is included in the prediction, it is generally good practice to first re-scale the electron density before postprocessing [50, 177], as follows:

$$\rho^{\text{scaled}}(\mathbf{r}) = \rho^{\text{ML}}(\mathbf{r}) \frac{N_{\text{e}}}{\int_{\Omega} \rho^{\text{ML}}(\mathbf{r}) d\mathbf{r}} \,. \tag{3.6}$$

Here, Ω is the periodic supercell used in the calculations, and N_{e} is the number of electrons in the system. Using this scaled density, the Kohn-Sham Hamiltonian is set up within the SPARC code framework, which was also used for data generation via AIMD simulations [204, 205, 206]. A single step of diagonalization is then performed, and the energy of the system is computed using the Harris-Foulkes formula [126, 127]. The errors in predicting $\rho^{\text{ML}}(\mathbf{r})$, and the ground state energy thus calculated, can be seen in Fig. 3.4. More detailed error values can be found in Table B.4 and Table B.5 in the supplemental material.

Chapter 4

Conclusions, Discussions and Future directions

Electronic structure prediction of quasi one dimensional materials: This work proposes a machine learning model, which predicts electronic fields of quasi-one-dimensional materials under torsional and axial loads. We have demonstrated the utility of the technique by predicting the electron density and the nuclear pseudocharges for armchair carbon nanotubes as a function of the tube geometry and applied strains. The data generation process of the ML model uses a specialized symmetry adapted version of Kohn-Sham Density Functional Theory that is particularly well suited for the problem geometries and loading conditions considered here. The machine learning model has several salient features as we now summarize. *First*, to

populate the input space, quasi–random low–discrepancy sequences (Sobol sequence) are employed and DFT simulations are performed at these inputs to generate the data for training of the machine learning model. This strategy allows for obtaining an accurate machine learning model with a minimal number of data points (123 training simulations in this case). *Second*, a two–step approach is taken to predict electronic fields. This involves dimensionality reduction of electronic fields followed by supervised learning in the reduced space. This two–step approach enables accurate prediction of high–dimensional electronic fields. The proposed ML model is remarkably accurate even for test cases with geometry and loading conditions that were not seen by the model during training. Moreover, the ML model is several orders of magnitude faster than the specialized, efficient Helical DFT technique used here for data generation. *Third*, a new technique based on a density–based clustering approach is developed to determine the atomic coordinates from the nuclear pseudocharges field predicted by the machine learning model. The atomic coordinates so obtained, are used to compute the non-local part of pseudopotentials that appear in the Kohn-Sham Hamiltonian, which would not have been possible otherwise. The electronic fields predicted by the machine learning model are postprocessed to obtain band structures, bandgaps, total energies, and various energy components.

We anticipate that machine learning models of the type developed here, will find use in computational investigations of strain engineering in low-dimensional systems and the multiscale modeling of the electromechanical response of such systems. One of the

key advantages of the current ML model is its incorporation of symmetries commonly associated with quasi-one-dimensional systems, which makes it easier to explore the composition-structure space of such materials [78, 79]. Therefore, we anticipate that in conjunction with techniques for DFT calculations of large scale systems [150, 223], the current ML model and its extensions are likely to help in the exploration of novel phases of chiral matter [224] and compositionally complex nanotubes [225]. Development of machine learning models that can capture atomic species specific features, as well as ones that can perform the post-processing steps associated with calculations of quantities such as energy components and band diagrams, serve as worthy directions for future research.

Electronic structure model for multi-million atom systems: We have developed an uncertainty quantification (UQ) enabled machine learning (ML) model that creates a map from the descriptors of atomic configurations to the electron densities. We use simple scalar product-based descriptors to represent the atomic neighborhood of a point in space. These descriptors, while being easy to compute, satisfy translational, rotational, and permutational invariances. In addition, they avoid any handcrafting. We systematically identify the optimal set of descriptors for a given dataset. Once trained, our model enables predictions across multiple length scales and supports embarrassingly parallel implementation. As far as we can tell, our work is the first attempt to systematically quantify uncertainties in ML predicted electron densities across different scales relevant to materials physics. To alleviate the high cost of

training data generation via KS-DFT, we propose a two-pronged strategy: i) we use thermalization to comprehensively sample system configurations, leading to a highly transferable ML model; and ii) we employ transfer learning to train the model using a large amount of inexpensively generated data from small systems while retraining a part of the model using a small amount of data from more expensive calculations of larger systems. The transfer learning procedure is systematically guided by the probability distributions of the data. This approach enables us to determine the maximum size of the training system, reducing dependence on heuristic selection. As a result of these strategies, the cost of training data generation is reduced by more than 50%, while the models continue to be highly transferable across a large variety of material configurations. Our use of Bayesian Neural Networks (BNNs) allows the uncertainty associated with these aforementioned strategies to be accurately assessed, thus enabling confident predictions in scenarios involving millions of atoms, for which ground-truth data from conventional KS-DFT calculations is infeasible to obtain. Overall, our ML model significantly decreases the reliance on heuristics used by prior researchers, streamlining the process of ML-based electronic structure prediction and making it more systematic.

We demonstrate the versatility of the proposed machine learning models by accurately predicting electron densities for multiple materials and configurations. We focus on bulk aluminum and Silicon-Germanium alloy systems. The ML model shows remarkable accuracy when compared with DFT calculations, even for systems containing

thousands of atoms. In the future, a similar model can be developed to test the applicability of the present descriptors and ML framework for molecules across structural and chemical space [226, 227, 228, 229]. As mentioned above, the ML model also has excellent generalization capabilities, as it can predict electron densities for systems with localized and extended defects, and varying alloy compositions, even when the data from such systems were not included in the training. It is likely that the ensemble averaging over model parameters in the BNNs, along with comprehensive sampling of the descriptor space via system thermalization together contribute to the model generalization capabilities. Our findings also show a strong agreement between physical parameters calculated from the DFT and ML electron densities (e.g. lattice constants and bulk moduli).

To rigorously quantify uncertainties in the predicted electron density, we adopt a Bayesian approach. Uncertainty quantification by a Bayesian neural network (BNN) is mathematically well-founded and offers a more reliable measure of uncertainty in comparison to non-Bayesian approaches such as the method of ensemble averaging. Further, we can decompose the total uncertainty into aleatoric and epistemic parts. This decomposition allows us to distinguish and analyze the contributions to the uncertainty arising from (i) inherent noise in the training data (i.e. aleatoric uncertainty) and (ii) insufficient knowledge about the model parameters due to the lack of information in the training data (i.e. epistemic uncertainty). The aleatoric uncertainty or the noise in the data is considered irreducible, whereas the epistemic uncertainty can

be reduced by collecting more training data. As mentioned earlier, the UQ capability of the model allows us to establish an upper bound on the uncertainty caused by two key heuristic strategies present in our ML model, namely, data generation via the thermalization of systems and transfer learning.

The reliability of the ML models is apparent from the low uncertainty of its prediction for systems across various length-scales and configurations. Furthermore, the magnitude of uncertainty for the million-atom systems is similar to that of smaller systems for which the accuracy of the ML model has been established. This allows us to have confidence in the ML predictions of systems involving multi-million atoms, which are far beyond the reach of conventional DFT calculations.

The ML model can achieve a remarkable speed-up of more than two orders of magnitude over DFT calculations, even for systems involving a few hundred atoms. As shown here, these computational efficiency gains by the ML model can be further pushed to regimes involving multi-million atoms, not accessible via conventional KS-DFT calculations.

In the future, we intend to leverage the uncertainty quantification aspects of this model to implement an active learning framework. This framework will enable us to selectively generate training data, reducing the necessity of extensive datasets and significantly lowering the computational cost associated with data generation. Moreover, we anticipate that the computational efficiencies offered via the transfer learning

approach, are likely to be even more dramatic while considering more complex materials systems, e.g. compositionally complex alloys [230, 231].

Chapter 5

Publications and Presentations

Published articles

† **Pathrudkar, S.**, Yu, H. M., Ghosh, S., & Banerjee, A. S. (2022). Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain. Physical Review B, 105(19), 195141.

† Yadav, U., **Pathrudkar, S.**, & Ghosh, S. (2021). Interpretable machine learning model for the deformation of multiwalled carbon nanotubes. Physical Review B, 103(3), 035407.

† Yadav, U., **Pathrudkar, S.**, & Ghosh, S. (2021, November). Deformation Manifold Learning Model for Deformation of Multi-Walled Carbon Nano-Tubes: Exploring the Latent Space. In ASME International Mechanical Engineering

Congress and Exposition (Vol. 85680, p. V012T12A010). American Society of Mechanical Engineers.

Articles in review

† S. Pathrudkar, P. Thiagarajan, S. Agarwal, A. S. Banerjee, and S. Ghosh, "Electronic structure prediction of multi-million atom systems through uncertainty quantification enabled transfer learning", Under second round of review in npj Computational Materials. arXiv preprint arXiv:2308.13096 (2023).

Conference Presentations

† Pathrudkar, S., Thiagarajan, P., Agarwal, S., Banerjee, A. S., & Ghosh, S., Title: Predicting the electronic structure of Large-Scale Bulk Metallic Systems using Machine Learning Conference: 17th U. S. National Congress on Computational Mechanics, Albuquerque, New Mexico, July, 2023

† Pathrudkar, S., Thiagarajan, P., Agarwal, S., Banerjee, A. S., & Ghosh, S., Title: Predicting the electron density of bulk metals and alloys at large scales using machine learning Workshop: USACM, Data-Driven and Computational Modeling of Materials Across Scales, Los Angeles, California, May, 2023

† Pathrudkar, S., Yadav, U., & Ghosh, S., Title: A Manifold Learning Model for the Deformation of Multiwalled Carbon Nanotubes under Torsion and Bending.

Conference: 10th International Conference on Multiscale Materials Modeling (MMM10), Baltimore, Maryland, October, 2022

† Pathrudkar, S., Yu, H. M., Ghosh, S., & Banerjee, A. S., Title: Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain Conference: 19th U.S. National Conference for Theoretical and Applied Mechanics (USNC/TAM), Austin, Texas, June, 2022

References

[1] S. Raju, K. Sivasubramanian, and E. Mohandas, "The high temperature bulk modulus of aluminium: an assessment using experimental enthalpy and thermal expansion data," *Solid state communications*, vol. 122, no. 12, pp. 671–676, 2002.

[2] A. S. Cooper, "Precise lattice constants of germanium, aluminum, gallium arsenide, uranium, sulphur, quartz and sapphire," *Acta Crystallographica*, vol. 15, no. 6, pp. 578–582, 1962.

[3] T. Ching, D. S. Himmelstein, B. K. Beaulieu-Jones, A. A. Kalinin, B. T. Do, G. P. Way, E. Ferrero, P.-M. Agapow, M. Zietz, M. M. Hoffman, *et al.*, "Opportunities and obstacles for deep learning in biology and medicine," *Journal of The Royal Society Interface*, vol. 15, no. 141, p. 20170387, 2018.

[4] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep

learning in biomedicine," *Molecular pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[5] P. Thiagarajan, P. Khairnar, and S. Ghosh, "Explanation and use of uncertainty quantified by bayesian neural network classifiers for breast histopathology images," *IEEE Transactions on Medical Imaging*, vol. 41, no. 4, pp. 815–825, 2022.

[6] O. V. Prezhdo, "Advancing physical chemistry with machine learning," 2020.

[7] A. Richardson, B. M. Signor, B. A. Lidbury, and T. Badrick, "Clinical chemistry in higher dimensions: machine-learning and enhanced prediction from routine clinical chemistry data," *Clinical biochemistry*, vol. 49, no. 16-17, pp. 1213–1220, 2016.

[8] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, "Machine learning in manufacturing: advantages, challenges, and applications," *Production & Manufacturing Research*, vol. 4, no. 1, pp. 23–45, 2016.

[9] D. T. Pham and A. A. Afify, "Machine-learning techniques and their applications in manufacturing," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 219, no. 5, pp. 395–412, 2005.

[10] J. Chen, R. Randall, B. Peeters, W. Desmet, and H. Van der Auweraer, "Artificial neural network based fault diagnosis of ic engines," in *Key Engineering Materials*, vol. 518, pp. 47–56, Trans Tech Publ, 2012.

[11] F. Wang, S. Ma, H. Wang, Y. Li, and J. Zhang, "Prediction of nox emission for coal-fired boilers based on deep belief network," *Control Engineering Practice*, vol. 80, pp. 26–35, 2018.

[12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[13] X. Liu, C. E. Athanasiou, N. P. Padture, B. W. Sheldon, and H. Gao, "A machine learning approach to fracture mechanics problems," *Acta Materialia*, vol. 190, pp. 105–112, 2020.

[14] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, vol. 52, pp. 477–508, 2020.

[15] R. Mattey and S. Ghosh, "A novel sequential method to train physics informed neural networks for allen cahn and cahn hilliard equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 390, p. 114474, 2022.

[16] C. M. Bishop, "Pattern recognition," *Machine learning*, vol. 128, no. 9, 2006.

[17] B. Efron and T. Hastie, *Computer age statistical inference*, vol. 5. Cambridge University Press, 2016.

[18] K. Rajan, "Materials informatics: The materials "gene" and big data," *Annual Review of Materials Research*, vol. 45, pp. 153–169, 2015.

[19] J.-P. Correa-Baena, K. Hippalgaonkar, J. van Duren, S. Jaffer, V. R. Chandrasekhar, V. Stevanovic, C. Wadia, S. Guha, and T. Buonassisi, "Accelerating materials development via automation, machine learning, and high-performance computing," *Joule*, vol. 2, no. 8, pp. 1410–1420, 2018.

[20] Y. Liu, T. Zhao, W. Ju, and S. Shi, "Materials discovery and design using machine learning," *Journal of Materiomics*, vol. 3, no. 3, pp. 159–177, 2017.

[21] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, "Recent advances and applications of machine learning in solid-state materials science," *npj Computational Materials*, vol. 5, no. 1, pp. 1–36, 2019.

[22] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, "Machine learning for molecular and materials science," *Nature*, vol. 559, no. 7715, pp. 547–555, 2018.

[23] G. S. Na, S. Jang, Y.-L. Lee, and H. Chang, "Tuplewise material representation based machine learning for accurate band gap prediction," *The Journal of Physical Chemistry A*, vol. 124, no. 50, pp. 10616–10623, 2020.

[24] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, "Machine learning in materials science," *InfoMat*, vol. 1, no. 3, pp. 338–358, 2019.

[25] G. R. Schleder, A. C. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, "From dft to machine learning: recent approaches to materials science–a review," *Journal of Physics: Materials*, vol. 2, no. 3, p. 032001, 2019.

[26] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Physical Review*, vol. 140, no. 4A, pp. 1133–1138, 1965.

[27] P. C. Hohenberg and W. Kohn, "Inhomogenous electron gas," *Physical Review*, vol. 136, no. 3B, pp. 864–871, 1964.

[28] K. Kim, L. Ward, J. He, A. Krishna, A. Agrawal, and C. Wolverton, "Machine-learning-accelerated high-throughput materials screening: Discovery of novel quaternary heusler compounds," *Physical Review Materials*, vol. 2, no. 12, p. 123801, 2018.

[29] S. P. Ong, "Accelerating materials science with high-throughput computations and machine learning," *Computational Materials Science*, vol. 161, pp. 143–150, 2019.

[30] C. Kim, G. Pilania, and R. Ramprasad, "From organized high-throughput data to phenomenological theory using machine learning: the example of dielectric breakdown," *Chemistry of Materials*, vol. 28, no. 5, pp. 1304–1311, 2016.

[31] Z. Li, Q. Xu, Q. Sun, Z. Hou, and W.-J. Yin, "Thermodynamic stability landscape of halide double perovskites via high-throughput computing and machine learning," *Advanced Functional Materials*, vol. 29, no. 9, p. 1807280, 2019.

[32] P. De Luna, J. Wei, Y. Bengio, A. Aspuru-Guzik, and E. Sargent, "Use machine learning to find energy materials," 2017.

[33] G. Pilania, A. Mannodi-Kanakkithodi, B. Uberuaga, R. Ramprasad, J. Gubernatis, and T. Lookman, "Machine learning bandgaps of double perovskites," *Scientific reports*, vol. 6, no. 1, pp. 1–10, 2016.

[34] A. C. Rajan, A. Mishra, S. Satsangi, R. Vaish, H. Mizuseki, K.-R. Lee, and A. K. Singh, "Machine-learning-assisted accurate band gap predictions of functionalized mxene," *Chemistry of Materials*, vol. 30, no. 12, pp. 4031–4038, 2018.

[35] S. Chibani and F.-X. Coudert, "Machine learning approaches for the prediction of materials properties," *APL Materials*, vol. 8, no. 8, p. 080701, 2020.

[36] T. Toyao, K. Suzuki, S. Kikuchi, S. Takakusagi, K.-i. Shimizu, and I. Takigawa, "Toward effective utilization of methane: machine learning prediction of adsorption energies on metal alloys," *The Journal of Physical Chemistry C*, vol. 122, no. 15, pp. 8315–8326, 2018.

[37] K. Takahashi and I. Miyazato, "Rapid estimation of activation energy in heterogeneous catalytic reactions via machine learning," *Journal of computational chemistry*, vol. 39, no. 28, pp. 2405–2408, 2018.

[38] M. De Jong, W. Chen, R. Notestine, K. Persson, G. Ceder, A. Jain, M. Asta,

and A. Gamst, "A statistical learning framework for materials science: application to elastic moduli of k-nary inorganic polycrystalline compounds," *Scientific reports*, vol. 6, no. 1, pp. 1–11, 2016.

[39] J. D. Evans and F.-X. Coudert, "Predicting the mechanical properties of zeolite frameworks by machine learning," *Chemistry of Materials*, vol. 29, no. 18, pp. 7833–7839, 2017.

[40] B. A. Calfa and J. R. Kitchin, "Property prediction of crystalline solids from composition and crystal structure," *AIChE Journal*, vol. 62, no. 8, pp. 2605–2613, 2016.

[41] J. R. Moreno, G. Carleo, and A. Georges, "Deep learning the hohenberg-kohn maps of density functional theory," *Physical Review Letters*, vol. 125, no. 7, p. 076402, 2020.

[42] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Bypassing the kohn-sham equations with machine learning," *Nature communications*, vol. 8, no. 1, pp. 1–10, 2017.

[43] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, "Transferable machine-learning model of the electron density," *ACS central science*, vol. 5, no. 1, pp. 57–64, 2018.

[44] A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad,

"Solving the electronic structure problem with machine learning," *npj Computational Materials*, vol. 5, no. 1, pp. 1–7, 2019.

[45] D. Kamal, A. Chandrasekaran, R. Batra, and R. Ramprasad, "A charge density prediction model for hydrocarbons using deep neural networks," *Machine Learning: Science and Technology*, vol. 1, no. 2, p. 025003, 2020.

[46] M. Bogojeski, F. Brockherde, L. Vogt-Maranto, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, "Efficient prediction of 3d electron densities using machine learning," *arXiv preprint arXiv:1811.06255*, 2018.

[47] A. Fabrizio, A. Grisafi, B. Meyer, M. Ceriotti, and C. Corminboeuf, "Electron density learning of non-covalent systems," *Chemical science*, vol. 10, no. 41, pp. 9424–9432, 2019.

[48] L. Zepeda-Núñez, Y. Chen, J. Zhang, W. Jia, L. Zhang, and L. Lin, "Deep density: circumventing the kohn-sham equations via symmetry preserving neural networks," *Journal of Computational Physics*, vol. 443, p. 110523, 2021.

[49] M. Tsubaki and T. Mizoguchi, "Quantum deep field: Data-driven wave function, electron density generation, and atomization energy prediction and extrapolation with machine learning," *Physical Review Letters*, vol. 125, no. 20, p. 206401, 2020.

[50] J. M. Alred, K. V. Bets, Y. Xie, and B. I. Yakobson, "Machine learning electron density in sulfur crosslinked carbon nanotubes," *Composites Science and Technology*, vol. 166, pp. 3–9, 2018.

[51] S. Gong, T. Xie, T. Zhu, S. Wang, E. R. Fadel, Y. Li, and J. C. Grossman, "Predicting charge density distribution of materials using a local-environment-based graph convolutional network," *Physical Review B*, vol. 100, no. 18, p. 184103, 2019.

[52] Y. Shi Teh, S. Ghosh, and K. Bhattacharya, "Machine-learned prediction of the electronic fields in a crystal," *arXiv e-prints*, pp. arXiv–2104, 2021.

[53] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, "Accelerating finite-temperature kohn-sham density functional theory with deep neural networks," *Physical Review B*, vol. 104, no. 3, p. 035120, 2021.

[54] R. Nagai, R. Akashi, S. Sasaki, and S. Tsuneyuki, "Neural-network kohn-sham exchange-correlation potential and its out-of-training transferability," *The Journal of chemical physics*, vol. 148, no. 24, p. 241737, 2018.

[55] B. Kanungo, P. M. Zimmerman, and V. Gavini, "Exact exchange-correlation potentials from ground-state electron densities," *Nature communications*, vol. 10, no. 1, pp. 1–9, 2019.

[56] J. Schmidt, C. L. Benavides-Riveros, and M. A. Marques, "Machine learning the physical nonlocal exchange–correlation functional of density-functional theory," *The journal of physical chemistry letters*, vol. 10, no. 20, pp. 6425–6431, 2019.

[57] B. Kanungo, P. M. Zimmerman, and V. Gavini, "A comparison of exact and model exchange–correlation potentials for molecules," *The Journal of Physical Chemistry Letters*, vol. 12, pp. 12012–12019, 2021.

[58] Z. Dai, L. Liu, and Z. Zhang, "Strain engineering of 2d materials: issues and opportunities at the interface," *Advanced Materials*, vol. 31, no. 45, p. 1805417, 2019.

[59] C. Si, Z. Sun, and F. Liu, "Strain engineering of graphene: a review," *Nanoscale*, vol. 8, no. 6, pp. 3207–3217, 2016.

[60] D. G. Schlom, L.-Q. Chen, C. J. Fennie, V. Gopalan, D. A. Muller, X. Pan, R. Ramesh, and R. Uecker, "Elastic strain engineering of ferroic oxides," *Mrs Bulletin*, vol. 39, no. 2, pp. 118–130, 2014.

[61] V. M. Pereira and A. C. Neto, "Strain engineering of graphene's electronic structure," *Physical Review Letters*, vol. 103, no. 4, p. 046801, 2009.

[62] J. Li, Z. Shan, and E. Ma, "Elastic strain engineering for unprecedented materials properties," *MRS Bulletin*, vol. 39, no. 2, pp. 108–114, 2014.

[63] J.-W. Jiang, "Strain engineering for thermal conductivity of single-walled carbon nanotube forests," *Carbon*, vol. 81, pp. 688–693, 2015.

[64] H. M. Ghassemi, C. H. Lee, Y. K. Yap, and R. S. Yassar, "Field emission and strain engineering of electronic properties in boron nitride nanotubes," *Nanotechnology*, vol. 23, no. 10, p. 105702, 2012.

[65] C. D. Aiello, M. Abbas, J. Abendroth, A. S. Banerjee, D. Beratan, J. Belling, B. Berche, A. Botana, J. R. Caram, L. Celardo, *et al.*, "A chirality-based quantum leap: A forward-looking review," *arXiv preprint arXiv:2009.00136*, 2020.

[66] Y. Hakobyan, E. Tadmor, and R. James, "Objective quasicontinuum approach for rod problems," *Physical Review B*, vol. 86, no. 24, p. 245435, 2012.

[67] A. S. Banerjee, *Density Functional Methods for Objective Structures: Theory and Simulation Schemes*. PhD thesis, University of Minnesota, Minneapolis, 2013.

[68] A. S. Banerjee, "Ab initio framework for systems with helical symmetry: Theory, numerical implementation and applications to torsional deformations in nanostructures," *Journal of the Mechanics and Physics of Solids*, vol. 154, p. 104515, 2021.

[69] A. S. Banerjee and P. Suryanarayana, "Cyclic density functional theory: A route to the first principles simulation of bending in nanostructures," *Journal of the Mechanics and Physics of Solids*, vol. 96, pp. 605–631, 2016.

[70] S. Ghosh, A. S. Banerjee, and P. Suryanarayana, "Symmetry-adapted real-space density functional theory for cylindrical geometries: Application to large group-iv nanotubes," *Phys. Rev. B*, vol. 100, p. 125143, Sep 2019.

[71] H. M. Yu and A. S. Banerjee, "Density functional theory method for twisted geometries with application to torsional deformations in group-iv nanotubes," 2021.

[72] S. Agarwal and A. Banerjee, "A spectral scheme for kohn-sham density functional theory of helical structures," *Bulletin of the American Physical Society*, 2021.

[73] S. Agarwal and A. Banerjee, "Solution of the schrödinger equation for quasi-one-dimensional materials using helical waves." (In preparation), 2022.

[74] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press, first ed., 2004.

[75] U. Yadav, S. Pathrudkar, and S. Ghosh, "Interpretable machine learning model for the deformation of multiwalled carbon nanotubes," *Physical Review B*, vol. 103, no. 3, p. 035407, 2021.

[76] A. Damle, A. Levitt, and L. Lin, "Variational formulation for wannier functions with entangled band structure," *Multiscale Modeling & Simulation*, vol. 17, no. 1, pp. 167–191, 2019.

[77] N. Marzari and D. Vanderbilt, "Maximally localized generalized wannier functions for composite energy bands," *Physical review B*, vol. 56, no. 20, p. 12847, 1997.

[78] R. D. James, "Objective structures," *Journal of the Mechanics and Physics of Solids*, vol. 54, no. 11, pp. 2354–2390, 2006.

[79] T. Dumitrica and R. D. James, "Objective molecular dynamics," *Journal of the Mechanics and Physics of Solids*, vol. 55, no. 10, pp. 2206 – 2236, 2007.

[80] P. Koskinen, "Electromechanics of twisted graphene nanoribbons," *Applied Physics Letters*, vol. 99, no. 1, p. 013105, 2011.

[81] P. Koskinen, "Electronic and optical properties of carbon nanotubes under pure bending," *Phys. Rev. B*, vol. 82, p. 193409, Nov 2010.

[82] T. Dumitrica, "Computational nanomechanics of quasi-one-dimensional structures in a symmetry-adapted tight binding framework," in *Trends in Nanophysics* (V. Barsan and A. Aldea, eds.), vol. 0 of *Engineering Materials*, pp. 29–55, Springer Berlin Heidelberg, 2010.

[83] E. B. Barros, A. Jorio, G. G. Samsonidze, R. B. Capaz, A. G. S. Filho, J. M. Filho, G. Dresselhaus, and M. S. Dresselhaus, "Review on the symmetry-related properties of carbon nanotubes," *Physics Reports*, vol. 431, no. 6, pp. 261 – 302, 2006.

[84] R. McWeeny, *Symmetry:An Introduction to Group Theory and Its Applications*. Dover, first ed., 2002.

[85] M. Hammermesh, *Group Theory and Its Application to Physical Problems*. Dover, first ed., 1989.

[86] L. Kleinman and D. Bylander, "Efficacious form for model pseudopotentials," *Physical Review Letters*, vol. 48, no. 20, p. 1425, 1982.

[87] A. S. Banerjee, P. Suryanarayana, and J. E. Pask, "Periodic Pulay method for robust and efficient convergence acceleration of self-consistent field iterations," *Chemical Physics Letters*, vol. 647, pp. 31–35, 2016.

[88] N. Troullier and J. L. Martins, "Efficient pseudopotentials for plane-wave calculations," *Physical review B*, vol. 43, no. 3, p. 1993, 1991.

[89] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, "Structural relaxation made simple," *Physical Review Letters*, vol. 97, no. 17, p. 170201, 2006.

[90] J. P. Perdew and Y. Wang, "Accurate and simple analytic representation of the electron-gas correlation energy," *Physical Review B*, vol. 45, no. 23, p. 13244, 1992.

[91] J. R. Chelikowsky, N. Troullier, and Y. Saad, "Finite-difference-pseudopotential

method: Electronic structure calculations without a basis," *Physical review letters*, vol. 72, no. 8, p. 1240, 1994.

[92] J. R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad, "Higher order finite difference pseudopotential method: An application to diatomic molecules," *Phys. Rev. B*, vol. 50, pp. 11355–11364, 1994.

[93] X. Jing, N. Troullier, D. Dean, N. Binggeli, J. R. Chelikowsky, K. Wu, and Y. Saad, "Ab initio molecular-dynamics simulations of si clusters using the higher-order finite-difference-pseudopotential method," *Physical Review B*, vol. 50, no. 16, pp. 12234–12237, 1994.

[94] H. Kikuji, O. Tomoya, F. Yoshitaka, and T. Shigeru, *First-principles calculations in real-space formalism: electronic configurations and transport properties of nanostructures*. World Scientific, 2005.

[95] S. Ghosh and P. Suryanarayana, "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Isolated clusters," *Computer Physics Communications*, vol. 212, pp. 189–204, 2017.

[96] S. Ghosh and P. Suryanarayana, "SPARC: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Extended systems," *Computer Physics Communications*, vol. 216, pp. 109–125, 2017.

[97] S. Ghosh, A. S. Banerjee, and P. Suryanarayana, "Symmetry-adapted real-space density functional theory for cylindrical geometries: Application to large group-IV nanotubes," *Physical Review B*, vol. 100, no. 12, p. 125143, 2019.

[98] S. Pathrudkar, "Deformation manifold learning model for multi walled carbon nanotubes," Master's thesis, Michigan Technological University, 2021.

[99] H. Robbins, "Some aspects of the sequential design of experiments," *Bulletin of the American Mathematical Society*, vol. 58, no. 5, pp. 527–535, 1952.

[100] S. L. Lohr, *Sampling: design and analysis*. Chapman and Hall/CRC, 2019.

[101] K.-T. Fang, R. Li, and A. Sudjianto, *Design and modeling for computer experiments*. Chapman and Hall/CRC, 2005.

[102] J. Santiago, M. Claeys-Bruno, and M. Sergent, "Construction of space-filling designs using wsp algorithm for high dimensional spaces," *Chemometrics and Intelligent Laboratory Systems*, vol. 113, pp. 26–31, 2012.

[103] T. W. Simpson, J. Poplinski, P. N. Koch, and J. K. Allen, "Metamodels for computer-based engineering design: survey and recommendations," *Engineering with computers*, vol. 17, no. 2, pp. 129–150, 2001.

[104] D. K. Lin, T. W. Simpson, and W. Chen, "Sampling strategies for computer experiments: design and analysis," *International Journal of Reliability and applications*, vol. 2, no. 3, pp. 209–240, 2001.

[105] J. M. Hammersley, "Monte carlo methods for solving multivariable problems," *Annals of the New York Academy of Sciences*, vol. 86, no. 3, pp. 844–874, 1960.

[106] I. M. Sobol', "On the distribution of points in a cube and the approximate evaluation of integrals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 7, no. 4, pp. 784–802, 1967.

[107] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[108] R. Jin, W. Chen, and A. Sudjianto, "An efficient algorithm for constructing optimal design of computer experiments," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 37009, pp. 545–554, 2003.

[109] A. B. Owen, "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica*, pp. 439–452, 1992.

[110] M. E. Johnson, L. M. Moore, and D. Ylvisaker, "Minimax and maximin distance designs," *Journal of statistical planning and inference*, vol. 26, no. 2, pp. 131–148, 1990.

[111] M. Gunzburger and J. Burkardt, "Uniformity measures for point sample in hypercubes," *Rapp. tech. Florida State University (cf. p. 73)*, 2004.

[112] J.-S. Park, "Optimal latin-hypercube designs for computer experiments," *Journal of statistical planning and inference*, vol. 39, no. 1, pp. 95–111, 1994.

[113] H. Niederreiter, "Low-discrepancy and low-dispersion sequences," *Journal of number theory*, vol. 30, no. 1, pp. 51–70, 1988.

[114] M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu, "A framework for data-driven analysis of materials under uncertainty: Countering the curse of dimensionality," *Computer Methods in Applied Mechanics and Engineering*, vol. 320, pp. 633–667, 2017.

[115] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[116] I. T. Jolliffe, "Springer series in statistics," *Principal component analysis*, vol. 29, 2002.

[117] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[118] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[119] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*, pp. 81–100, Springer, 2003.

[120] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *kdd*, vol. 96,34, pp. 226–231, 1996.

[121] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[122] J. M. Alred, K. V. Bets, Y. Xie, and B. I. Yakobson, "Machine learning electron density in sulfur crosslinked carbon nanotubes," *Composites Science and Technology*, vol. 166, pp. 3–9, 2018. Carbon nanotube composites for structural applications.

[123] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[124] E. Vecharynski, C. Yang, and F. Xue, "Generalized preconditioned locally harmonic residual method for non-hermitian eigenproblems," *SIAM Journal on Scientific Computing*, vol. 38, pp. A500—A527, 2015.

[125] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*. SIAM, Revised ed., 2011.

[126] J. Harris, "Simplified method for calculating the energy of weakly interacting fragments," *Physical Review B*, vol. 31, no. 4, p. 1770, 1985.

[127] W. M. C. Foulkes and R. Haydock, "Tight-binding models and density-functional theory," *Physical review B*, vol. 39, no. 17, p. 12520, 1989.

[128] H. Zou and T. Hastie *Journal of the Royal Statistical Society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[129] D. P. Kingma and J. Ba *arXiv preprint arXiv:1412.6980*, 2014.

[130] J. Ding, X. Yan, and J. Cao, "Analytical relation of band gaps to both chirality and diameter of single-wall carbon nanotubes," *Physical Review B*, vol. 66, no. 7, p. 073401, 2002.

[131] S. Pathrudkar, P. Thiagarajan, S. Agarwal, A. S. Banerjee, and S. Ghosh, "Electronic structure prediction of multi-million atom systems through uncertainty quantification enabled transfer learning," *arXiv preprint arXiv:2308.13096*, 2023.

[132] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Physical review*, vol. 140, no. 4A, p. A1133, 1965.

[133] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Physical review*, vol. 136, no. 3B, p. B864, 1964.

[134] T. Van Mourik, M. Bühl, and M.-P. Gaigeot, "Density functional theory across chemistry, physics and biology," 2014.

[135] P. Makkar and N. N. Ghosh, "A review on the use of dft for the prediction of the properties of nanomaterials," *RSC advances*, vol. 11, no. 45, pp. 27897–27924, 2021.

[136] J. Hafner, C. Wolverton, and G. Ceder, "Toward computational materials design: the impact of density functional theory on materials research," *MRS bulletin*, vol. 31, no. 9, pp. 659–668, 2006.

[137] A. E. Mattsson, P. A. Schultz, M. P. Desjarlais, T. R. Mattsson, and K. Leung, "Designing meaningful density functional theory calculations in materials science—a primer," *Modelling and Simulation in Materials Science and Engineering*, vol. 13, no. 1, p. R1, 2004.

[138] R. M. Martin, L. Reining, and D. M. Ceperley, *Interacting electrons*. Cambridge University Press, 2016.

[139] S. Datta, *Quantum transport: atom to transistor*. Cambridge university press, 2005.

[140] S. Goedecker, "Linear scaling electronic structure methods," *Reviews of Modern Physics*, vol. 71, no. 4, p. 1085, 1999.

[141] D. Bowler, T. Miyazaki, and M. Gillan, "Recent progress in linear scaling ab initio electronic structure techniques," *Journal of Physics: Condensed Matter*, vol. 14, no. 11, p. 2781, 2002.

[142] E. Artacho, D. Sánchez-Portal, P. Ordejón, A. Garcia, and J. M. Soler, "Linear-scaling ab-initio calculations for large and complex systems," *physica status solidi (b)*, vol. 215, no. 1, pp. 809–817, 1999.

[143] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, "Introducing onetep: Linear-scaling density functional simulations on parallel computers," *The Journal of chemical physics*, vol. 122, no. 8, 2005.

[144] P. P. Pratapa, P. Suryanarayana, and J. E. Pask, "Spectral quadrature method for accurate o (n) electronic structure calculations of metals and insulators," *Computer Physics Communications*, vol. 200, pp. 96–107, 2016.

[145] L. Lin, M. Chen, C. Yang, and L. He, "Accelerating atomic orbital-based electronic structure calculation via pole expansion and selected inversion," *J. Phys.: Condens. Matter*, vol. 25, p. 295501, 2013.

[146] L. Lin, C. Yang, J. C. Meza, J. Lu, L. Ying, and W. E, "Selinv—an algorithm for selected inversion of a sparse symmetric matrix," *ACM Transactions on Mathematical Software (TOMS)*, vol. 37, no. 4, pp. 1–19, 2011.

[147] P. Motamarri and V. Gavini, "Subquadratic-scaling subspace projection method for large-scale kohn-sham density functional theory calculations using spectral finite-element discretization," *Physical Review B*, vol. 90, no. 11, p. 115127, 2014.

[148] L. Lin, J. Lu, L. Ying, R. Car, and W. E, "Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems," *Communications In Mathematical Sciences*, vol. 7, p. 755, 2009.

[149] A. S. Banerjee, L. Lin, W. Hu, C. Yang, and J. E. Pask, "Chebyshev polynomial filtered subspace iteration in the discontinuous galerkin method for large-scale electronic structure calculations," vol. 145, no. 15, p. 154101, 2016.

[150] A. S. Banerjee, L. Lin, P. Suryanarayana, C. Yang, and J. E. Pask, "Two-level chebyshev filter based complementary subspace method: pushing the envelope of large-scale electronic structure calculations," *Journal of chemical theory and computation*, vol. 14, no. 6, pp. 2930–2946, 2018.

[151] A. Marek, V. Blum, R. Johanni, V. Havu, B. Lang, T. Auckenthaler, A. Heinecke, H.-J. Bungartz, and H. Lederer, "The elpa library: scalable parallel eigenvalue solutions for electronic structure theory and computational science," *Journal of Physics: Condensed Matter*, vol. 26, no. 21, p. 213201, 2014.

[152] V. Gavini, S. Baroni, V. Blum, D. R. Bowler, A. Buccheri, J. R. Chelikowsky, S. Das, W. Dawson, P. Delugas, M. Dogan, *et al.*, "Roadmap on electronic structure codes in the exascale era," *Modelling and Simulation in Materials Science and Engineering*, vol. 31, no. 6, p. 063301, 2023.

[153] W. Hu, H. An, Z. Guo, Q. Jiang, X. Qin, J. Chen, W. Jia, C. Yang, Z. Luo, J. Li, *et al.*, "2.5 million-atom ab initio electronic-structure simulation of complex metallic heterostructures with dgdft," in *SC22: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–13, IEEE, 2022.

[154] W. Hu, X. Qin, Q. Jiang, J. Chen, H. An, W. Jia, F. Li, X. Liu, D. Chen, F. Liu, *et al.*, "High performance computing of DGDFT for tens of thousands of atoms using millions of cores on sunway taihulight," *Science Bulletin*, vol. 66, no. 2, pp. 111–119, 2021.

[155] M. Dogan, K.-H. Liou, and J. R. Chelikowsky, "Real-space solution to the electronic structure problem for nearly a million electrons," *The Journal of Chemical Physics*, vol. 158, no. 24, 2023.

[156] S.-H. Wei, L. Ferreira, J. E. Bernard, and A. Zunger, "Electronic properties of random alloys: Special quasirandom structures," *Physical Review B*, vol. 42, no. 15, p. 9622, 1990.

[157] M. Jaros, "Electronic properties of semiconductor alloy systems," *Reports on Progress in Physics*, vol. 48, no. 8, p. 1091, 1985.

[158] S. Fischer, S. Kaul, and H. Kronmüller, "Critical magnetic properties of disordered polycrystalline cr 75 fe 25 and cr 70 fe 30 alloys," *Physical Review B*, vol. 65, no. 6, p. 064443, 2002.

[159] G. T. de Laissardière, D. Nguyen-Manh, and D. Mayou, "Electronic structure of complex hume-rothery phases and quasicrystals in transition metal aluminides," *Progress in Materials Science*, vol. 50, no. 6, pp. 679–788, 2005.

[160] O. N. Senkov, G. Wilks, J. Scott, and D. B. Miracle, "Mechanical properties of nb25mo25ta25w25 and v20nb20mo20ta20w20 refractory high entropy alloys," *Intermetallics*, vol. 19, no. 5, pp. 698–706, 2011.

[161] O. Senkov, G. Wilks, D. Miracle, C. Chuang, and P. Liaw, "Refractory high-entropy alloys," *Intermetallics*, vol. 18, no. 9, pp. 1758–1765, 2010.

[162] A. K. Geim and I. V. Grigorieva, "Van der waals heterostructures," *Nature*, vol. 499, no. 7459, pp. 419–425, 2013.

[163] S. Carr, S. Fang, and E. Kaxiras, "Electronic-structure methods for twisted moiré layers," *Nature Reviews Materials*, vol. 5, no. 10, pp. 748–763, 2020.

[164] 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.

[165] H. J. Kulik, T. Hammerschmidt, J. Schmidt, S. Botti, M. A. Marques, M. Boley, M. Scheffler, M. Todorović, P. Rinke, C. Oses, *et al.*, "Roadmap on machine learning in electronic structure," *Electronic Structure*, vol. 4, no. 2, p. 023004, 2022.

[166] G. Csányi, T. Albaret, M. Payne, and A. De Vita, ""learn on the fly": A hybrid classical and quantum-mechanical molecular dynamics simulation," *Physical review letters*, vol. 93, no. 17, p. 175503, 2004.

[167] 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.

[168] A. Seko, A. Takahashi, and I. Tanaka, "Sparse representation for a potential energy surface," *Physical Review B*, vol. 90, no. 2, p. 024101, 2014.

[169] 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.

[170] C. Chen and S. P. Ong, "A universal graph deep learning interatomic potential for the periodic table," *Nature Computational Science*, vol. 2, no. 11, pp. 718–728, 2022.

[171] R. Freitas and Y. Cao, "Machine-learning potentials for crystal defects," *MRS Communications*, vol. 12, no. 5, pp. 510–520, 2022.

[172] A. M. Lewis, A. Grisafi, M. Ceriotti, and M. Rossi, "Learning electron densities in the condensed phase," *Journal of Chemical Theory and Computation*, vol. 17, no. 11, pp. 7203–7214, 2021.

[173] P. B. Jørgensen and A. Bhowmik, "Equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids," *npj Computational Materials*, vol. 8, no. 1, p. 183, 2022.

[174] L. Fiedler, N. A. Modine, S. Schmerler, D. J. Vogel, G. A. Popoola, A. P. Thompson, S. Rajamanickam, and A. Cangi, "Predicting electronic structures at any length scale with machine learning," *npj Computational Materials*, vol. 9, no. 1, p. 115, 2023.

[175] B. Deng, P. Zhong, K. Jun, J. Riebesell, K. Han, C. J. Bartel, and G. Ceder, "Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling," *Nature Machine Intelligence*, vol. 5, no. 9, pp. 1031–1041, 2023.

[176] T. W. Ko and S. P. Ong, "Recent advances and outstanding challenges for machine learning interatomic potentials," *Nature Computational Science*, pp. 1–3, 2023.

[177] S. Pathrudkar, H. M. Yu, S. Ghosh, and A. S. Banerjee, "Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain," *Physical Review B*, vol. 105, no. 19, p. 195141, 2022.

[178] G. Arora, A. Manzoor, and D. S. Aidhy, "Charge-density based evaluation and prediction of stacking fault energies in ni alloys from dft and machine learning," *Journal of Applied Physics*, vol. 132, no. 22, 2022.

[179] B. Medasani, A. Gamst, H. Ding, W. Chen, K. A. Persson, M. Asta, A. Canning, and M. Haranczyk, "Predicting defect behavior in b2 intermetallics by merging ab initio modeling and machine learning," *npj Computational Materials*, vol. 2, no. 1, p. 1, 2016.

[180] Y. S. Teh, S. Ghosh, and K. Bhattacharya, "Machine-learned prediction of the electronic fields in a crystal," *Mechanics of Materials*, vol. 163, p. 104070, 2021.

[181] C. D. Aiello, J. M. Abendroth, M. Abbas, A. Afanasev, S. Agarwal, A. S. Banerjee, D. N. Beratan, J. N. Belling, B. Berche, A. Botana, *et al.*, "A chirality-based quantum leap," *ACS nano*, vol. 16, no. 4, pp. 4989–5035, 2022.

[182] H. M. Yu and A. S. Banerjee, "Density functional theory method for twisted geometries with application to torsional deformations in group-iv nanotubes," *Journal of Computational Physics*, vol. 456, p. 111023, 2022.

[183] S. Agarwal and A. S. Banerjee, "Solution of the schrodinger equation for quasi-one-dimensional materials using helical waves," *arXiv preprint arXiv:2210.12252*, 2022.

[184] C. Woodward and S. Rao, "Flexible ab initio boundary conditions: Simulating isolated dislocations in bcc mo and ta," *Physical review letters*, vol. 88, no. 21, p. 216402, 2002.

[185] V. Gavini, K. Bhattacharya, and M. Ortiz, "Vacancy clustering and prismatic

dislocation loop formation in aluminum," *Physical Review B*, vol. 76, no. 18, p. 180101, 2007.

[186] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[187] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*, pp. 1613–1622, Proceedings of Machine Learning Research, 2015.

[188] P. Thiagarajan and S. Ghosh, "Jensen-shannon divergence based novel loss functions for bayesian neural networks," *arXiv preprint arXiv:2209.11366*, 2022.

[189] B. Settles, "Active learning literature survey," 2009.

[190] H. Huo and M. Rupp, "Unified representation of molecules and crystals for machine learning," *Machine Learning: Science and Technology*, vol. 3, no. 4, p. 045017, 2022.

[191] A. P. Bartók, R. Kondor, and G. Csányi, "On representing chemical environments," *Physical Review B*, vol. 87, no. 18, p. 184115, 2013.

[192] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, "Fast and accurate modeling of molecular atomization energies with machine learning," *Physical review letters*, vol. 108, no. 5, p. 058301, 2012.

[193] F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi, and M. Ceriotti, "Physics-inspired structural representations for molecules and materials," *Chemical Reviews*, vol. 121, no. 16, pp. 9759–9815, 2021.

[194] R. Resta and S. Sorella, "Electron localization in the insulating state," *Physical Review Letters*, vol. 82, no. 2, p. 370, 1999.

[195] E. Prodan and W. Kohn, "Nearsightedness of electronic matter," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 33, pp. 11635–11638, 2005.

[196] A. T. Fowler, C. J. Pickard, and J. A. Elliott, "Managing uncertainty in data-derived densities to accelerate density functional theory," *Journal of Physics: Materials*, vol. 2, no. 3, p. 034001, 2019.

[197] Y. Zhou, W. Zhang, E. Ma, and V. L. Deringer, "Device-scale atomistic modelling of phase-change memory materials," *Nature Electronics*, vol. 6, no. 10, pp. 746–754, 2023.

[198] P. Suryanarayana, P. P. Pratapa, A. Sharma, and J. E. Pask, "Sqdft: Spectral quadrature method for large-scale parallel o(n) kohn–sham calculations at high temperature," *Computer Physics Communications*, vol. 224, pp. 288–298, 2018.

[199] S. Das, P. Motamarri, V. Subramanian, D. M. Rogers, and V. Gavini, "Dft-fe

1.0: A massively parallel hybrid cpu-gpu density functional theory code using finite-element discretization," *Computer Physics Communications*, vol. 280, p. 108473, 2022.

[200] L.-W. Wang, B. Lee, H. Shan, Z. Zhao, J. Meza, E. Strohmaier, and D. H. Bailey, "Linearly scaling 3d fragment method for large-scale electronic structure calculations," in *SC'08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, pp. 1–10, IEEE, 2008.

[201] W. Yang and T.-S. Lee, "A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules," *The Journal of chemical physics*, vol. 103, no. 13, pp. 5674–5678, 1995.

[202] M. Herbold and J. Behler, "Machine learning transferable atomic forces for large systems from underconverged molecular fragments," *Physical Chemistry Chemical Physics*, vol. 25, no. 18, pp. 12979–12989, 2023.

[203] M. Herbold and J. Behler, "A hessian-based assessment of atomic forces for training machine learning interatomic potentials," *The Journal of Chemical Physics*, vol. 156, no. 11, 2022.

[204] Q. Xu, A. Sharma, B. Comer, H. Huang, E. Chow, A. J. Medford, J. E. Pask, and P. Suryanarayana, "Sparc: Simulation package for ab-initio real-space calculations," *SoftwareX*, vol. 15, p. 100709, 2021.

[205] Q. Xu, A. Sharma, and P. Suryanarayana, "M-sparc: Matlab-simulation package for ab-initio real-space calculations," *SoftwareX*, vol. 11, p. 100423, 2020.

[206] S. Ghosh and P. Suryanarayana, "Sparc: Accurate and efficient finite-difference formulation and parallel implementation of density functional theory: Isolated clusters," *Computer Physics Communications*, vol. 212, pp. 189–204, 2017.

[207] J. P. Perdew, K. Burke, and M. Ernzerhof, "Generalized gradient approximation made simple," *Physical review letters*, vol. 77, no. 18, p. 3865, 1996.

[208] D. Hamann, "Optimized norm-conserving vanderbilt pseudopotentials," *Physical Review B*, vol. 88, no. 8, p. 085117, 2013.

[209] D. J. Evans and B. L. Holian, "The nose–hoover thermostat," *The Journal of chemical physics*, vol. 83, no. 8, pp. 4069–4074, 1985.

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

[211] W. Kohn, "Density functional and density matrix method scaling linearly with the number of atoms," *Physical Review Letters*, vol. 76, no. 17, p. 3168, 1996.

[212] N. W. Ashcroft and N. D. Mermin, *Solid state physics*. Cengage Learning, 2022.

[213] V. Hamer and P. Dupont, "An importance weighted feature selection stability measure," *J. Mach. Learn. Res.*, vol. 22, jan 2021.

[214] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, p. 1157–1182, mar 2003.

[215] U. Yadav, S. Pathrudkar, and S. Ghosh, "Deformation manifold learning model for deformation of multi-walled carbon nano-tubes: Exploring the latent space," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 85680, p. V012T12A010, American Society of Mechanical Engineers, 2021.

[216] 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, 2018.

[217] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, "Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials," *The Journal of chemical physics*, vol. 148, no. 24, 2018.

[218] G. E. Hinton and D. Van Camp, "Keeping the neural networks simple by minimizing the description length of the weights," in *Proceedings of the sixth annual conference on Computational learning theory*, pp. 5–13, 1993.

[219] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[220] R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson, "Cyclical stochastic gradient mcmc for bayesian deep learning," in *International Conference on Learning Representations*, 2020.

[221] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *Advances in neural information processing systems*, vol. 30, 2017.

[222] 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, p. 2903, 2019.

[223] W. Hu, L. Lin, and C. Yang, "Dgdft: A massively parallel method for large scale density functional theory calculations," *The Journal of chemical physics*, vol. 143, no. 12, p. 124110, 2015.

[224] F. Qin, W. Shi, T. Ideue, M. Yoshida, A. Zak, R. Tenne, T. Kikitsu, D. Inoue, D. Hashizume, and Y. Iwasa, "Superconductivity in a chiral nanotube," *Nature communications*, vol. 8, no. 1, pp. 1–6, 2017.

[225] A. Korde, B. Min, E. Kapaca, O. Knio, I. Nezam, Z. Wang, J. Leisen, X. Yin, X. Zhang, D. S. Sholl, *et al.*, "Single-walled zeolitic nanotubes," *Science*, vol. 375, no. 6576, pp. 62–66, 2022.

[226] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, "Comparing molecules and solids across structural and alchemical space," *Physical Chemistry Chemical Physics*, vol. 18, no. 20, pp. 13754–13769, 2016.

[227] A. P. Bartók, S. De, C. Poelking, N. Bernstein, J. R. Kermode, G. Csányi, and M. Ceriotti, "Machine learning unifies the modeling of materials and molecules," *Science advances*, vol. 3, no. 12, p. e1701816, 2017.

[228] V. L. Deringer, A. P. Bartók, N. Bernstein, D. M. Wilkins, M. Ceriotti, and G. Csányi, "Gaussian process regression for materials and molecules," *Chemical Reviews*, vol. 121, no. 16, pp. 10073–10141, 2021.

[229] A. Grisafi, D. M. Wilkins, G. Csányi, and M. Ceriotti, "Symmetry-adapted machine learning for tensorial properties of atomistic systems," *Physical review letters*, vol. 120, no. 3, p. 036002, 2018.

[230] Y. Ikeda, B. Grabowski, and F. Körmann, "Ab initio phase stabilities and mechanical properties of multicomponent alloys: A comprehensive review for high entropy alloys and compositionally complex alloys," *Materials Characterization*, vol. 147, pp. 464–511, 2019.

[231] E. P. George, D. Raabe, and R. O. Ritchie, "High-entropy alloys," *Nature reviews materials*, vol. 4, no. 8, pp. 515–534, 2019.

[232] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: a big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[233] X. Glorot, A. Bordes, and Y. Bengio in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.

[234] X. Glorot and Y. Bengio in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

[235] L. Prechelt, *Early Stopping — But When?*, pp. 53–67. Springer, 2012.

[236] W. S. Sarle, "Stopped training and other remedies for overfitting," *Computing science and statistics*, pp. 352–360, 1996.

[237] L. Himanen, M. O. Jäger, E. V. Morooka, F. F. Canova, Y. S. Ranawat, D. Z. Gao, P. Rinke, and A. S. Foster, "Dscribe: Library of descriptors for machine learning in materials science," *Computer Physics Communications*, vol. 247, p. 106949, 2020.

[238] S. Das, B. Kanungo, V. Subramanian, G. Panigrahi, P. Motamarri, D. Rogers, P. Zimmerman, and V. Gavini, "Large-scale materials modeling at quantum accuracy: Ab initio simulations of quasicrystals and interacting extended defects in metallic alloys," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12, 2023.

[239] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[240] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[241] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[242] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[243] 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.

[244] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, "mala-project." `https://github.com/mala-project/mala`, 2021.

Appendix A

Chapter 2 - Supplementary material

A.1 Data Generation

Data generation for Machine Learning: We use the following bounds for the input space in order to generate the data: $R_{\text{avg}} \in [20.32\,, 101.60\,]$ Bohr, $\alpha \in [0, 0.0025]$ and $\tau \in [4.4052,\ 4.8052]$ Bohr. This corresponds to choosing armchair CNTs with cyclic symmetry group orders between 16 and 80, i.e., with radii in the experimentally relevant 1 to 5 nanometer range. The increments in α and τ are 0.0005 and 0.1 Bohr, respectively. The maximum applied torsional strain of about 3.86 degrees/nm —

close to the regime in which torsional instabilities may start to appear [79], and the maximum axial strain considered here is about 4.3%. Helical DFT simulations were performed in the input space following Sobol sequencing. Note that, the Sobol sequence would not always generate a sample point that is feasible for simulations, given the discrete nature of the nanotube radius. For such cases, we have carried out simulations at the nearest feasible value of R_{avg} and strain parameters.

To achieve the desired accuracy in the prediction of the electronic fields with the minimum number of DFT simulations we start with a set of points guided by the Sobol sequence. Subsequently, we add simulations in smaller sets (referred to as Sobol sets here) to the training data, till we attain the desired accuracy in the prediction of electronic fields. Our first set consists of 85 simulations followed by 48 and 31 simulations in the second and third sets, respectively. As mentioned in Section 2.3.1 these three sets of simulations successively refine the input space. Fig. A.1 shows NRMSE for test data obtained when the ML model was trained using these three Sobol sets cumulatively. The first bar denotes test set NRMSE when only set I (85 data points) was used; the second bar denotes test set NRMSE when sets 1 and 2 (85+48 data points) were used; the third bar denotes test set NRMSE when sets 1,2 and 3 (85+48+31 data points) were used. Note that for each of these cases, 15% of the data available to the ML model was used for testing.

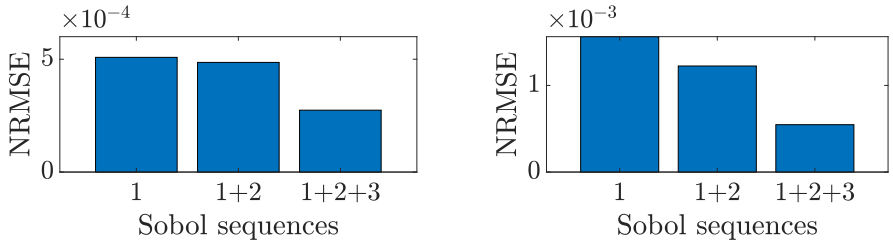


Figure A.1: Mean of NRMSE for test data when the machine learning model is trained using Sobol sets. (*Left*) Error bars for ρ, (*Right*) Error bars for b.

A.2 Training of Neural Networks

The Learning curves for the neural networks $\mathcal{N}_1$ (for ρ) and $\mathcal{N}_2$ (for b) are presented in Fig. A.2. The loss function used to train the neural networks is computed on CoPCs and is given in Eq. A.1 below.

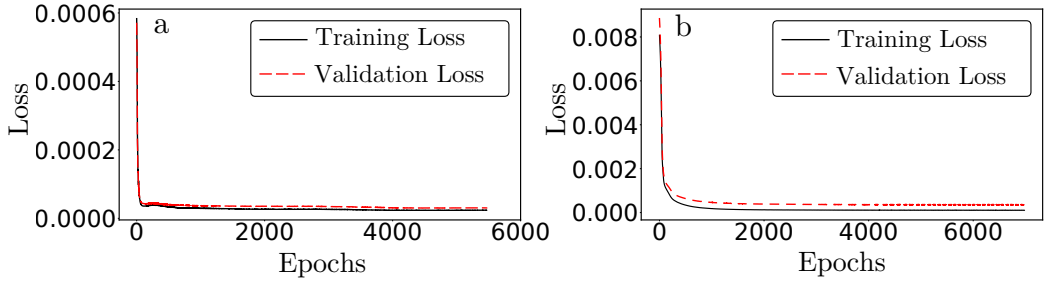


Figure A.2: (a) Learning curve for $\mathcal{N}_1$, (b) Learning curve for $\mathcal{N}_2$.

Hyperparameter Optimization and Regularization

The proposed machine learning model contains various parameters associated with the NNs that can affect the model's overall accuracy and performance. In particular, so-called hyperparameters associated with controlling the learning process need to be tuned. Important hyperparameters for our model include the architectures for NNs, the activation function, the learning rate, and the number of iterations. We discuss each of these below.

Architecture: The predictive capability of a NN and the accuracy obtained in the prediction depends on the number of hidden layers and the number of neurons in the hidden layers. We optimized the number of hidden layers and number of neurons per layer using the grid search method [232]. Fig. A.3 shows the test error for $\mathcal{N}_1$ and $\mathcal{N}_2$ trained for varying number of layers and varying number of nodes per layer. For $\mathcal{N}_1$, six layers of 150 neurons each yielded the least test error, and for $\mathcal{N}_2$, two layers of 150 neurons each yielded the least test error.

Activation Function: We used Rectified Linear Unit (ReLU) as an activation function for both neural networks $\mathcal{N}_1$ and $\mathcal{N}_2$. This choice avoids problems of vanishing or exploding gradients encountered by other common activation functions like Sigmoid, Tanh [233, 234].

Learning Rate: The learning rate was set to 0.001 as suggested in [129]. Other parameters pertinent to the Adam optimizer were set at suggested values [129] based

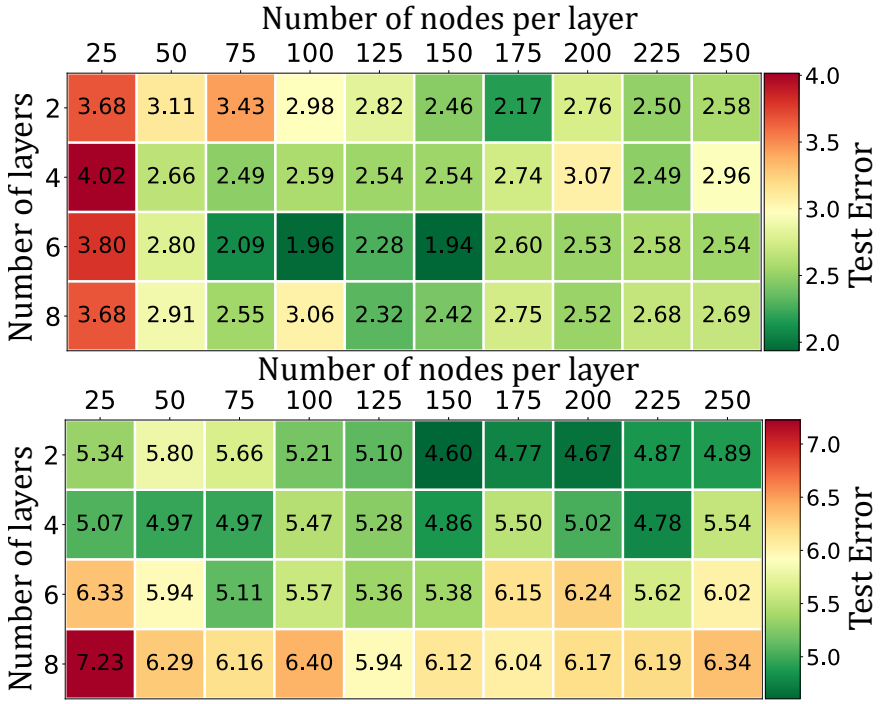


Figure A.3: (*Top*) Test error ($\times 10^{-5}$) for various architectures of $\mathcal{N}_1$ (trained for ρ), (*Bottom*) Test error ($\times 10^{-4}$) for various architectures of $\mathcal{N}_2$ (trained for b.)

on good results for other machine learning problems ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$).

Number of Iterations: In order to avoid overfitting and help ensure good generalization performance of the ML model, we used early stopping [235, 236]. We stopped the training when the validation loss does not improve over a specific number of *patience epochs*. We employed *patience* epochs of 1000, and the maximum number of epochs was set to 200000.

Elastic net regularization: Along with early stopping, we used elastic net regularization to avoid overfitting. This technique is a combination of $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization methods [128], and overcomes the individual drawbacks of each. The loss

function including $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization can be written as:

$$\tilde{\mathcal{J}}(y, \mathcal{N}(x, \bar{\mathbf{w}})) \quad = \mathcal{J}(y, \mathcal{N}(x, \bar{\mathbf{w}})) + \lambda_1 ||\bar{\mathbf{w}}||_1 + \lambda_2 ||\bar{\mathbf{w}}||_2 \,, \qquad \lambda_1, \lambda_2 \in \mathbb{R} \,. \quad \text{(A.1)}$$

Here, $\mathcal{J}(y, \mathcal{N}(x, \bar{\mathbf{w}}))$ is the mean squared error over the true outputs y and the neural network ($\mathcal{N}$) predicted outputs $y' = \mathcal{N}(x, \bar{\mathbf{w}})$. Furthermore, $\bar{\mathbf{w}}$ are the weights and biases of $\mathcal{N}$, x is the input, and $||\bar{\mathbf{w}}||_1$ and $||\bar{\mathbf{w}}||_2$ are the $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization terms, respectively. We have used $\lambda_1 = \lambda_2 = 10^{-5}$ for both $\mathcal{N}_1$ and $\mathcal{N}_2$.

A.3 Comparison of the Clustering and Neural Network Approaches to Obtaining the Nuclear Coordinates

We compare our clustering based approach to determine atomic coordinates with a neural network that was trained to predict the atomic coordinates directly from the inputs : R_{avg}, α and τ. We found that the error (distance between actual atomic coordinates and predicted atomic coordinates) was significantly higher in the case of the neural network model than the DBSCAN based approach proposed here. The errors

in atomic coordinates using the neural network approach and our clustering approach are shown in Fig A.4. The superior performance of the clustering based approach is likely related to the ability of the method to make use of the specific structure of the b field (i.e., it is the superposition of a set of non-overlapping, atom-centered, spherically symmetric charge distributions), as opposed to the neural network model which does not incorporate such information.

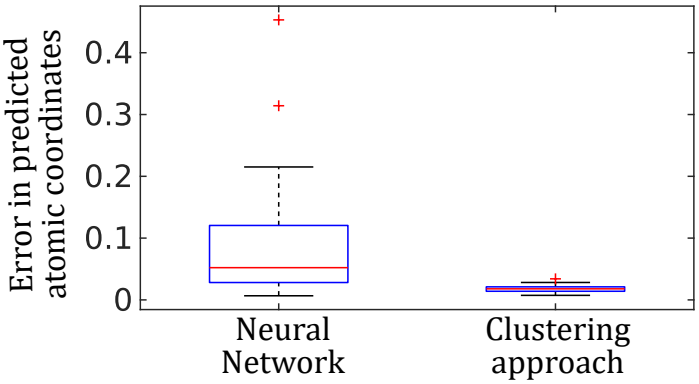


Figure A.4: Error in predicted atomic coordinates (in Bohr), i.e., the distance between true and predicted nucleus positions, using a neural network and the DBSCAN based clustering approach.

Appendix B

Chapter 3 - Supplementary material

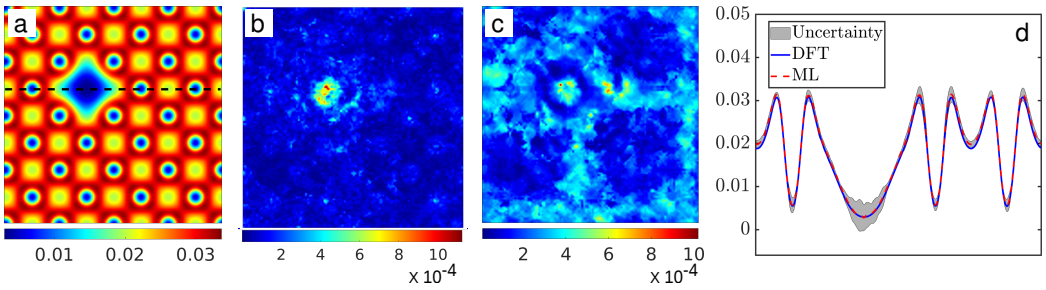


Figure B.1: Uncertainty quantification for a 256 atom aluminum system with mono vacancy defect. From left: i) ML prediction of the electron density shown on the defect plane, ii) Epistemic uncertainty iii) Aleatoric uncertainty iv) Uncertainty shown on the black dotted line from the ML prediction slice. The uncertainty represents the bound $\pm 3\sigma$, where, σ is the total uncertainty.

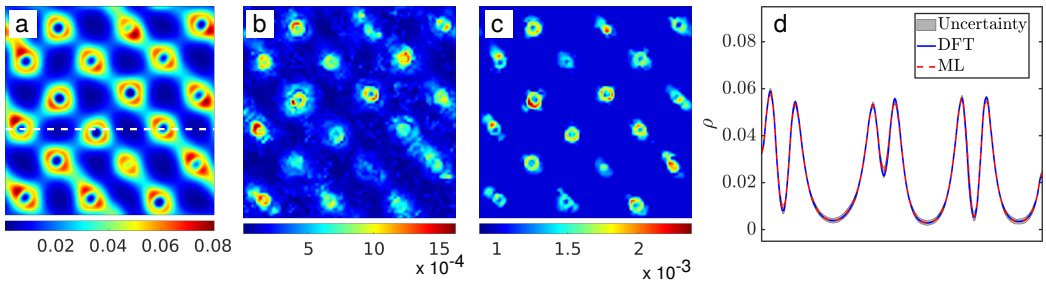


Figure B.2: Uncertainty quantification $Si_{0.5}Ge_{0.5}$ system containing 216 atoms. (a) ML prediction of the electron density, (b) Epistemic Uncertainty (c) Aleatoric Uncertainty (d) Total Uncertainty shown along the dotted line from the ML prediction slice. The uncertainty represents the bound $\pm 3\sigma_{total}$, where, σ_{total} is the total uncertainty.

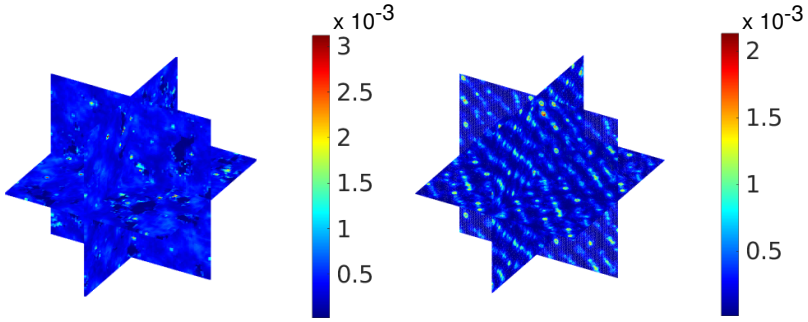


Figure B.3: (Left) Total uncertainty for the Al system ($\sim$ 4.1 million atoms) shown in Fig. 3.8(a) of the main text. (Right) Total uncertainty for the SiGe system ($\sim$ 1.4 million atoms) shown in Fig. 3.8(b) of the main text (right).

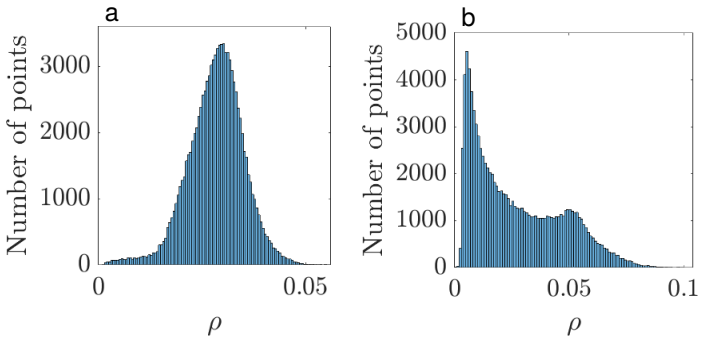


Figure B.4: Histogram showing the distribution of charge density (ρ) for (a) aluminum and (b) SiGe.

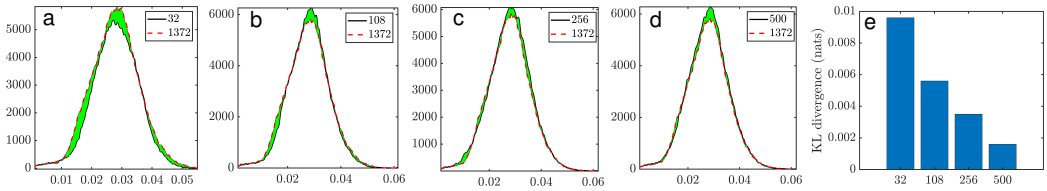


Figure B.5: (a-d) Comparison of the histograms of electron density of aluminum for the largest system with that of smaller systems. The shaded green areas show the difference between the histograms. The largest aluminum system has 1372 atoms, whereas the smaller systems have 32, 108, 256, and 500 atoms. e) Kullback–Leibler (KL) divergence between the probability distributions corresponding to the histograms in a-d and that of the largest system. The values of the KL divergence decreases with the increase in system size.

B.1 Efficient generation of atomic neighborhood descriptors

The atomic neighborhood descriptors to encode the atomic neighborhood of the grid point are $||\mathbf{r}_i - \mathbf{R}_J||$ and $\frac{(\mathbf{r}_i - \mathbf{R}_K)\cdot(\mathbf{r}i - \mathbf{R}_S)}{||\mathbf{r}_i - \mathbf{R}_K||\,||\mathbf{r}i - \mathbf{R}_S||}$, as described in the section 3.3 of the main text. Our implementation of descriptor generation employs a tree data structure to reduce computational complexity and is outlined as a pseudocode in Algorithm 1.

The descriptors described above satisfy the following conditions outlined in [190] and [237]: (i) invariance with respect to rotations and translations of the system (ii) invariance with respect to the permutation of atomic indices, i.e., the descriptors are independent of the enumeration of the atoms. (iii) for a given atomic neighborhood, the descriptors are unique. (iv) the descriptors encode the atomic neighborhood

effectively while keeping the overall count low. (v) the descriptors generation process is computationally inexpensive and uses standard linear algebra operations.

Algorithm 1 Generation of Descriptors

M = Number of nearest neighbor atoms to compute distances
M_a = Number of nearest neighbor atoms to compute angles
k = Number of angles obtained for each M_a atoms
Build supercell by extending unit cell in all directions
KDTree = K-D tree for atoms in supercell
for g do ▷ **g**: grid point
  $D \leftarrow$ distances to M nearest atoms from **g** using K-D tree
  for $j = 1$ to M_a **do**
    $\mathbf{a}_i$ ▷ coordinates of i^{th} nearest atom from g using K-D tree
    $\mathbf{v}_1 \leftarrow \mathbf{a}_i - \mathbf{g}$
    for $j = 1$ to k **do**
      $\mathbf{A}_j$ ▷ coordinates of j^{th} nearest atom from $\mathbf{a}_i$
      $\mathbf{v}_2 \leftarrow \mathbf{A}_j - \mathbf{g}$
      $\mathcal{A}_{ij} \leftarrow \frac{\mathbf{v}_1 . \mathbf{v}_2}{||\mathbf{v}_1||\,||\mathbf{v}_2||}$
    end for
  end for
  $\mathcal{A} \leftarrow$ flatten($\mathcal{A}$)
  descriptors $\leftarrow [D, \mathcal{A}]$
end for
Note: Inner two **for** loops are vectorized and Outermost **for** is parallelized in the implementation

Descriptors are obtained by implementing a parallelized version of Algorithm 1. In the case of SiGe systems, instead of explicitly encoding the species information, we follow [44] and concatenate the descriptors obtained for Si and Ge, to form inputs to the neural network. To encode the relative placement of Si and Ge atoms with respect to each other, we also consider the cosine of angles between Si and Ge atoms formed at the grid point for the SiGe case.

B.2 Computational Efficiency

Computational time comparison between DFT calculation and ML prediction is given in tables B.1 and B.2 for aluminum and SiGe, respectively. DFT calculations were performed using CPUs, whereas the ML predictions used a combination of GPU (inference step) and CPU (descriptor generation) resources.

The primary contributor to ML prediction time is descriptor generation, constituting the majority of the computational effort and the remaining time is neural network inference (See Tables B.1 and B.2). Given that neural network inference is well-suited for GPU execution and is commonly performed on GPUs, our assessment of parallelization performance focuses on descriptor generation time. In Figure B.6, we present the parallelization performance of descriptor generation for the Aluminum system with 500 atoms. This parallelization was executed using the MATLAB's 'parfor' function on NERSC Perlmutter CPUs and we observe 66.6% strong scaling for 64 processors.

The DFT and ML calculations presented in this work were performed through a combination of resources, namely, desktop workstations, the Hoffman2 cluster at UCLA's Institute of Digital Research and Education (IDRE), the Applied Computing GPU cluster at MTU, and NERSC's supercomputer, Perlmutter. Every compute

node of the Hoffman2 cluster has two 18-core Intel Xeon Gold 6140 processors (24.75 MB L3 cache, clock speed of 2.3 GHz), 192 GB of RAM and local SSD storage. Every compute node on Perlmutter has a 64-core AMD EPYC 7763 processor (256 MB L3 cache, clock speed of 2.45 GHz), 512 GB of RAM and local SSD storage. The GPU resources on Perlmutter consist of NVIDIA A100 Tensor Core GPUs. The GPU nodes used at UCLA and MTU consist of Tesla V100 GPUs.

Large system generation: The million atom systems presented were generated by repeating one of the available test systems in all three directions and adding random perturbations in the atomic coordinates for each atom in the resulting system. This process ensures that the million-atom system is distinct from the smaller system employed in its generation and that the atomic neighborhoods generated within the million-atom system are not identical to those in the smaller system. Additionally, it is noteworthy that the systems replicated to achieve the million-atom configurations are entirely excluded from the training dataset (e.g. in the case of Aluminum, 1372 atom system was employed to generate the 4.1 million-atom system, while the training process utilized 32 and 108 atom systems. In the case of SiGe, a 512 atom system was used to generate the 1.4 million atom system). The perturbations used were sampled from a normal distribution with a zero mean and a 0.1 Bohr standard deviation. The choice of standard deviation was deliberate, aiming to prevent impractical distances between atoms and ensure realistic configurations.

Large system calculations: We present electron density calculation for Al and SiGe systems, each with an excess of a million atoms, in Fig. 3.8(a) and Fig. 3.8(b) of the main text, respectively. To predict the charge density while avoiding memory overload issues, we partition these multi-million atom systems into smaller systems, while retaining the atomic neighborhood information consistent with the larger original systems. In the case of aluminum, we break down the 4.1M atom system into smaller units comprising 1372 atoms and a grid consisting of 175^3 points. Computation of descriptors for this 1372-atom chunk takes approximately 34.72 seconds on a desktop workstation system equipped with a 36-core Intel(R) Xeon(R) Gold 5220 CPU @ 2.20GHz. Subsequently, the charge density prediction requires approximately 1.6 seconds on an Nvidia V100 GPU. Overall, the charge density prediction for the 4.1M Al system takes around 30.72 hours of wall time on combined CPU and GPU resources.

Analogously, for SiGe, we partition the 1.4M atom system into smaller systems composed of 1000 atoms and a grid with dimensions of 132^3 points. The computation of descriptors for this 1000-atom SiGe chunk requires 22.17 seconds on the aforementioned desktop system. The subsequent charge density prediction takes approximately 1.1 seconds. Overall, it takes around 6.8 hours of wall time on combined CPU and GPU resources, to predict the electron density of the SiGe system with 1.4M atoms.

Thus, the techniques described here make it possible to routinely predict the electronic

structure of systems at unprecedented scales, while using only modest resources on standard desktop systems.

Number of Atoms		32	108	256	500
DFT Time (CPU)		466	11560	112894	245798
ML Time	Descriptor Generation	43.25	151.52	367.54	739.58
	ρ Prediction (CPU)	2.76	9.67	23.46	47.20
	ρ Prediction (GPU)	0.60	0.64	0.75	0.99
	Total (With GPU)	43.85	152.16	368.29	740.57
DFT time / Total ML time		10.63	75.97	306.53	331.90

Table B.1

Comparison of DFT and ML wall times for prediction of electron density for an aluminum system. All times are in seconds. The DFT calculations were performed on Hoffman CPUs, ML descriptor generation was done on Hoffman CPUs, and the ML inference was performed on Tesla V100 GPUs.

Number of Atoms		64	216	512	1000
DFT Time		185	4774	51247	281766
ML Time	Descriptor Generation	38.82	115.23	291.45	611.2
	ρ Prediction (CPU)	2.22	7.37	17.37	33.05
	ρ Prediction (GPU)	0.50	0.62	0.75	0.89
	Total (With GPU)	39.32	115.85	292.20	612.09
DFT time / Total ML time		4.70	41.21	175.38	460.33

Table B.2

Comparison of DFT and ML wall times for prediction of electron density for a SiGe system. All times are in seconds. The DFT calculations were performed on Perlmutter CPUs, ML descriptor generation was done on Perlmutter CPUs and the ML inference was performed on Tesla V100 GPUs.

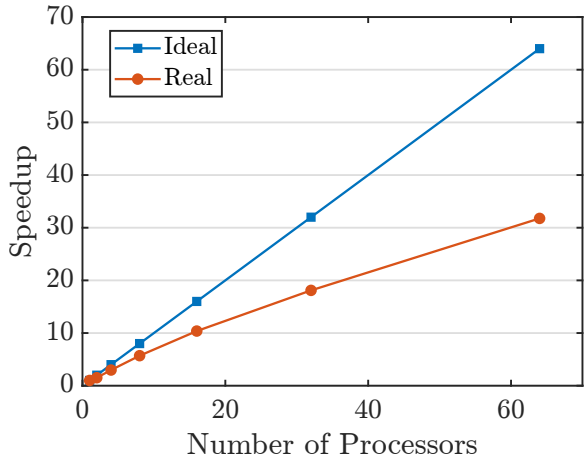


Figure B.6: Speedup of ML prediction time with respect to number of processors (strong parallel scaling). The plot is shown for a 500 atom Aluminum system. Speedup is obtained with reference to 1 processor. The computation was performed on NERSC Perlmutter CPUs.

B.3 Feature Convergence Analysis

Algorithm 2 and Algorithm 3 describe the process used to obtain the optimal number of descriptors. In algorithm 2 only distances (set I) are considered as descriptors. The size of the set I (i.e. M) is selected for which the RMSE for the test dataset converges.

As an illustration, for the aluminum systems, following algorithm 2 we use an increment of $m = 10$. The algorithm converges to $M = 60$ as seen in Fig. 3.11 of the main text. Therefore, the set I consists of 60 descriptors. Next, Set II descriptors consist of angles subtended at the grid point by a pair of atoms taken from the set of M neighboring atoms in the set I determined by algorithm 2. Each pair of the neighboring atoms forms an angle at the grid point, yielding a total of $M(M-1)/2$

Algorithm 2 Optimal nearest neighbors

```
M = 0                                              ▷ Initialization
ϵ_0 = ϵ_{-m} = δ_1 = δ_2 = A large number          ▷ Initialization
η = tolerance in RMSE
while δ_1 ≥ η & δ_2 ≥ η do
    M = M + m                                      ▷ Increase M by m ∈ ℤ^+
    N_set I ← M                                    ▷ M nearest atoms
    N ← N_set I                                    ▷ Only set I descriptors
    Compute N descriptors
    Train f_N                                      ▷ Train the BNN
    ϵ_M ← RMSE                                     ▷ Compute RMSE
    δ_1 ← |ϵ_M − ϵ_{M−m}|
    δ_2 ← |ϵ_M − ϵ_{M−2m}|
end while
M = M − 2m
```

angles, which quickly becomes computationally intractable with increasing M. To alleviate this issue, we reduce the number of Set II descriptors by eliminating large angles, which are not expected to play a significant role. This amounts to choosing angles originating from $M_a < M$ atoms closest to the grid point, and the $k-$ nearest neighbors of each of these atoms. This yields a total of $M_a \times k$ angle descriptors. For various fixed values of k, we iteratively choose M_a till the RMSE over the test dataset converges (Fig. 3.11 of the main text).

Following algorithm 3 we use an increment of $m = 5$. Fig. 3.11 of the main text shows the convergence plot for angles for $k = 2, 3$, and 4. For $M = 60$, the RMSE value is the minimum for $k = 3$. The RMSE value for $k = 3$ converges at $M_a = 15$, which results in a total of $M_a \times k = 45$ angles. Therefore, set II consists of 45 descriptors. To summarize, following the present feature selection strategy, the total number of descriptors used for the aluminum model is $N = N_{\text{set I}} + N_{\text{set II}} = 105$.

Algorithm 3 Optimal number of angles

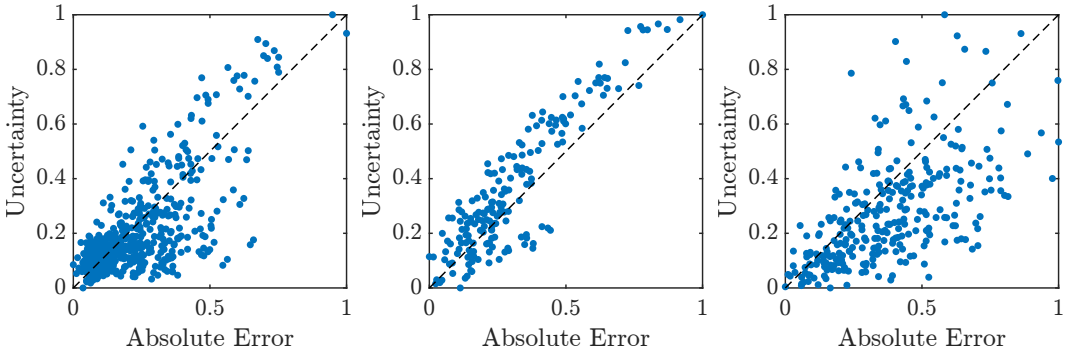
$k = 0$ ▷ Initialization
$\epsilon_0 = \epsilon_{-m} = \delta_1 = \delta_2 = \delta_3 =$ A large number ▷ Initialization
$\eta =$ tolerance in RMSE
while $\delta_3 \geq \eta$ **do**
  $k = k + 1$
  $M_a = 0$
  while $\delta_1 \geq \eta$ & $\delta_2 \geq \eta$ **do**
    $M_a = M_a + m_a$ ▷ Increase M_a by $m_a \in \mathbb{Z}^+$
    $N_{\text{set II}} \leftarrow M_a \times k$ ▷ k neighbors of each of M_a nearest atoms
    $N \leftarrow N_{\text{set I}} + N_{\text{set II}}$ ▷ Number of total descriptors
    Compute N descriptors
    Train f_N ▷ Train the BNN
    $\epsilon_{M_a} \leftarrow$ RMSE ▷ Compute RMSE
    $\delta_1 \leftarrow |\epsilon_{M_a} - \epsilon_{M_a - m_a}|$
    $\delta_2 \leftarrow |\epsilon_{M_a} - \epsilon_{M_a - 2m_a}|$
  end while
  $M_a = M_a - 2m_a$
  $\epsilon'_k \leftarrow \epsilon_{M_a}$
  $\delta_3 \leftarrow |\epsilon'_k - \epsilon'_{k-1}|$
end while
$k = k - 1$

We found that including scalar triple products and scalar quadruple products in the descriptor, in addition to the dot products, did not improve the accuracy of the ML model. To interpret why this is the case, we observe that the (normalized) scalar triple product can be interpreted in terms of the corner solid angle (polar sine function) of the parallelepiped generated by three vectors starting at the given grid point and ending at three atoms chosen in the neighborhood of the grid point. However, this quantity can also be calculated through the dot products between these vectors and is, therefore, already incorporated in the second set of descriptors. Therefore, the scalar triple product does not furnish any additional information. Similar arguments can be made for quadruple and higher products.

B.4 Details on Uncertainty Quantification

We provide additional results on uncertainty quantification (UQ) in this section. One of the key advantages of the inbuilt UQ capabilities of the present ML model is that it allows us to assess the model's generalizability. To illustrate this, we consider systems with defects and varying alloy compositions. The uncertainty estimates of a model trained without any defect data in training are shown in Fig. 3.7 of the main text. The model is more confident in its prediction of defects even if a small amount (single snapshot) of defect data is added in training. This is evident by comparing Fig. B.1 and main text Fig. 3.7. This result is in agreement with the fact that unavailability or insufficient training data could yield high epistemic uncertainties at locations where such incompleteness of data exists. In addition to high uncertainty, the error at the defect location increases when data from systems with defects are not used in training. This implies a positive correspondence between error and uncertainty in the Bayesian neural network model. A similar effect of higher uncertainty for unknown compositions is observed for the SiGe systems. Since the model is trained only with data from SiGe systems with 50-50 composition, the uncertainties quantified for this composition shown in Fig. B.2 is less in comparison to the prediction for 60-40 composition (Fig. 3.6(b) of the main text). However, the uncertainty for the 60-40 composition is not significantly higher than the 50-50 composition, demonstrating the generalization capability of the ML model.

In the following, we investigate the correlation between error and epistemic uncertainty. The epistemic uncertainty is chosen since it captures the uncertainty due to modeling error. We found positive correlations between the uncertainty and the error for configurations that were not present in the training and therefore exhibit higher errors. Examples include vacancies in Aluminum and alloy compositions away from the training data, as shown in Fig. B.7. We have also observed that for systems similar to training data, the errors as well as uncertainties are quite low, and do not exhibit strong correlations. This indicates that for systems predicted with high uncertainties, uncertainty values may be used to identify regions with high error.



(a) 256 atom Aluminum with single vacancy defect

(b) 108 atom Aluminum with double vacancy defect

(c) 216 atom $Si_{0.4}Ge_{0.6}$

Figure B.7: Correlation between epistemic uncertainty and error. All three cases show a positive correlation with $R = 0.75, 0.90, 0.59$, respectively. The uncertainty values and absolute error values are normalized using the min-max method. Each data point in the plots corresponds to uncertainty and error values are averaged over the neighborhood that is used to compute descriptors for the data point.

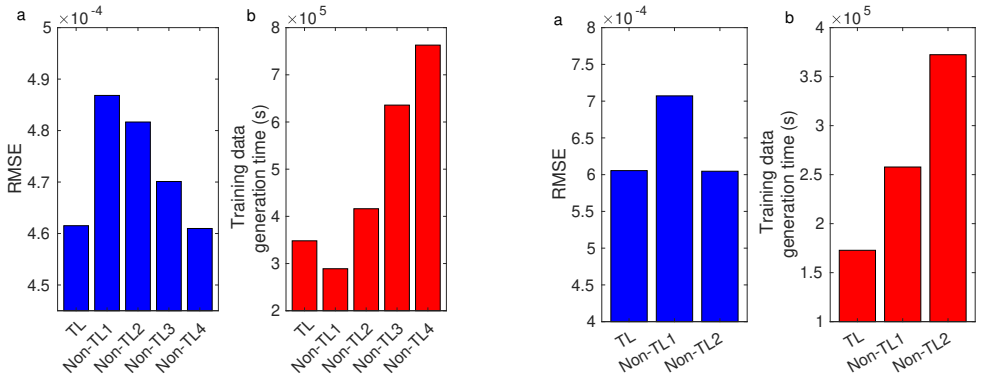
Results of uncertainty quantification $\approx$ 4.1 million atom aluminum system and $\approx$ 1.4 million atom SiGe system are shown in Fig. B.3. With an increase in system size, we extrapolate farther away from the system size included in the training data. Despite

this, the total uncertainty of millions of atom systems is similar to that of smaller systems. This implies that the model can predict systems with millions of atoms with the same level of confidence as smaller systems, which in turn assures the accuracy of the predictions. Looking ahead, we plan to further enhance the credibility of million-atom predictions by validating against results obtained from upcoming and state-of-the-art techniques involving Density Functional Theory (DFT) computations at a large scale [152, 198, 199, 238].

We found that the ML model is less confident in predicting charge densities near the nucleus in comparison to the away from the nucleus for various systems, which is reflected in the high values of uncertainties at those locations. We attribute this to fewer grid points close to the nuclei, and the availability of more data away from them. This imbalance in the data is evident from the histograms for the distribution of charge densities shown in Fig. B.4, where grid points with low values of the electron density — as is the case with points very close to the nuclei — are seen to be very few.

B.5 Details on the advantages of transfer learning

As demonstrated in prior research [48] and in this work, employing data from larger systems for training enhances the accuracy of machine learning models. However,

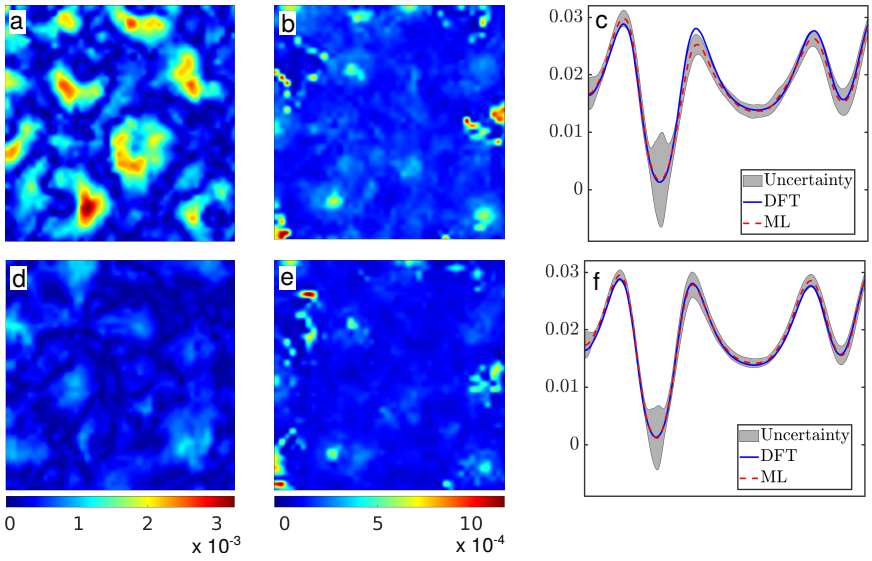


(a) Al system (b) SiGe System

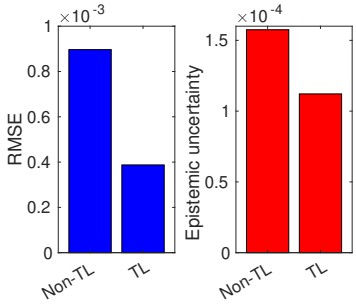
Figure B.8: Comparison of (a) error and (b) training data generation time between models with and without transfer learning.

the following question persists: what is the appropriate largest sizes of the training system to achieve a sufficiently accurate machine learning model that works across scales? To answer this question, we propose the following approach.

To ensure accurate predictions for bulk systems (comprising thousands or more atoms), it is imperative that our model be trained on data that statistically resembles such systems. Small-scale systems with only a few tens of atoms may not adequately represent the bulk limit, primarily due to the periodicity constraints inherent in simulations. This calls for training the model using larger systems. To determine appropriate training system sizes that adequately represents bulk systems, we employ the Kullback-Leibler (KL) divergence [239]. We consider the largest available system as the most faithful representation of bulk systems and use it to determine the largest size of the training systems. For the case of Aluminum, a system consisting of 1372 atoms can be reliably calculated using KS-DFT and is chosen as the reference.



(a)



(b)

Figure B.9: (i) Decrease in error and uncertainty for a larger system (1372 atom) with transfer learning. Comparison is shown between predictions by a non-TL model trained using data only from the 32-atom system i(a-c) and a TL model trained by transfer learning using additional data from the 108-atom system i(d-f). The slice considered is shown in Fig. 3.6(a)(a) of the main text. i(a and d) Error in ML prediction, i(b and e) Epistemic uncertainty, i(c and f) Total uncertainty along a line, as shown in Fig. 3.6(a)(a) of the main text. Color bars are the same for i(a) and (c), and i(b) and (d). (ii) Bar plot showing a decrease in RMSE error and epistemic uncertainty. ii(a) The decrease in RMSE error is 56% and ii(b) the decrease in the mean epistemic uncertainty is 29%.

We compare the electron density distributions from various available systems against this reference system. The KL divergence values then guide us in selecting the largest training system needed to train a model that can accurately predict even at large

scales (relevant to the reference system). Specifically, the largest training systems chosen by us contain 108 atoms, as these systems are found to be sufficiently statistically similar to the 1372-atom reference system (as illustrated in Fig. B.5). This meticulous selection process guarantees that our machine learning model is accurate at large scales while providing a judicious stopping point to our transfer learning scheme by determining the largest system needed for training. Thus, we present an approach that answers the question of selecting training system size and reduces the reliance on ad hoc heuristics for doing so.

The transfer learning approach [240] significantly reduces the root-mean-square error of a test dataset while costing much less computation for the training data generation. To depict this, a comparison of the transfer learned model with various non-transfer learned models is shown in Fig. B.8.

We found that transfer learning helps to reduce the error and uncertainty in prediction for larger systems. By adding data from the 108-atom aluminum systems in training, during the transfer learning approach, we significantly reduce the error (by 56%) and uncertainty (by 29%) of the predictions for a 1372-atom test system in comparison to a non-TL model trained using data only from the 32-atom systems, as shown in Fig. B.9.

B.6 Details on Bayesian Neural Network

Architecture: We use a Densenet [241] type architecture with three Dense blocks for the Bayesian Neural networks in this work. Each Dense block is composed of three hidden layers with 250 nodes per layer and a GELU activation function [242]. The skip connections in the Densenet-type architecture are weighted by a trainable coefficient. These skip connections have multiple advantages. Firstly, they prevent gradients from diminishing significantly during backpropagation. Further, they facilitate improved feature propagation by allowing each layer to directly access the feature generated by previous layers. Finally, these skip connections promote feature reuse, thereby substantially reducing the number of parameters. Such skip connections have been used for electron density predictions in the literature [48].

Due to the stochastic weights of Bayesian neural networks, each weight is represented by its mean and standard deviation. Thus, the number of parameters in a Bayesian neural network is twice as compared to a deterministic network with the same architecture. In addition, the output of the Bayesian Neural networks used in this work has two neurons, one for predicting the charge density (ρ) and the other for predicting the aleatoric uncertainty (σ).

Training Details: The parameters of the BNNs for the 32-atom Al system and 64-atom SiGe systems were initialized randomly with values drawn from the Gaussian distribution. The mean of the parameters were initialized with values drawn from $\mathcal{N}(0, 0.1)$. The standard deviations were parameterized as $\sigma = \log(1 + \exp(\tau))$ so that σ is always non-negative. The parameter τ was initialized with values drawn from $\mathcal{N}(-3, 0.1)$. The priors for all the network parameters were assumed to be Gaussians: $\mathcal{N}(0, 0.1)$. With these initializations and prior assumptions the initial models (i.e. model for 32-atom Al system and 64-atom SiGe system) were trained using standard back-propagation for BNNs. The Adam optimizer [129] was used for training and the learning rate was set to 10^{-3} for all the networks used in this work. In the case of transfer learning, we freeze both the mean and standard deviation of the initial one-third layers of the model and re-train the mean and standard deviations of the remaining layers of the model. The prior assumptions, initialization of the learnable parameters, and their learning procedures remained the same as described above for the 32-atom Al and 64-atom SiGe systems. The training time for the Al and SiGe systems are presented in Table B.3. All the Bayesian Neural networks are trained on NVIDIA A100 Tensor Core GPUs

The amount of data used in training for the two systems is as follows:

† Al: 127 snaps from 32 atom data and in addition 25 snaps from 108 atom data. The 108 atom data has $90 \times 90 \times 90$ grid points, while the 32 atom system has

$60 \times 60 \times 60$ grid points.

† SiGe: 160 snaps of 64 atom data and in addition 30 snaps of 216 atom data. The 64 atom system has $53 \times 53 \times 53$ grid points, while the 216 atom system has $79 \times 79 \times 79$ grid points.

System	Size	Epochs	Training wall time (s)	
			Per epoch	Total
Al	32	20	906	31060
	108	20	647	
SiGe	64	20	651	18030
	216	10	501	

Table B.3
GPU Training times for the BNNs. The training was performed on the NVIDIA Tesla A100 GPU.

Validation and Testing Details: 20% of the data from the systems used for training is used as validation data. Testing is performed on snapshots not used for training and validation, and systems that are larger than those used for generating the training data in order to determine the accuracy in electron density prediction.

B.7 Postprocessing results

In tables B.4 and B.5 we compare the errors in the electron densities and the ground state energies for various Al and SiGe systems. We see errors well below the millihartree per atom range for total energies, even in the presence of defects and some degree of compositional variations — these systems being quite far from the ones used

to generate the training data. The average L^1 norm per electron between ML and DFT electron densities for the largest available aluminum system (containing 1372 atoms — this is the largest aluminum system for which the DFT calculations could be carried out reliably within computational resource constraints), is 1.14×10^{-2}. In the case of SiGe, where the largest available system consists of 1728 atoms, the average L^1 norm per electron is 8.25×10^{-3}. We observe that the errors for these largest systems are somewhat smaller than the typical errors associated with the systems listed in Tables B.4 and B.5, contradictory to what is anticipated. This can be attributed to the fact that the available AIMD trajectories for larger systems are typically not long enough (due to computational constraints) to induce significant variations in atomic configurations with respect to the equilibrium configuration, unlike the longer AIMD trajectories available for smaller systems. Consequently, the largest systems tested here are more amenable to accurate prediction, resulting in lower errors.

The time for the calculation of the total energy and forces from ML-predicted densities via postprocessing involves computation of the electrostatic, exchange correlation and band-energy terms, and uses a single diagonalization step to compute wave-function dependent quantities. Therefore, its computational time is similar to that of a single self-consistent field (SCF) step in a regular DFT calculation, provided the same eigensolver is used. For reference, using the MATLAB version of the SPARC code [205] on a single CPU core, the postprocessing time is about 174 seconds for a 32 atom aluminum system while it is about 1600 seconds for 108 atoms. This also includes the

time for computation of the Hellmann-Feynman forces. We would also like to mention here that this postprocessing step can be significantly sped up by the ML prediction of other relevant quantities, such as the band energy and electrostatic fields [52]. As for the atomic forces, i.e., energy derivatives with respect to atomic coordinates, automatic differentiation of the underlying neural networks can be employed to speed up calculations. All of these constitute ongoing and future work.

Case	Accuracy of electron density (L^1 norm per electron)	Ground-state energy (Ha/atom)	Exch. Corr. energy (Ha/atom)	Fermi level (Ha)	Max error in eigenvalue (Ha)
Entire test data set	2.62×10^{-2}	2.33×10^{-4}	4.36×10^{-4}	4.61×10^{-4}	4.58×10^{-3}
Al (32 atoms)	2.27×10^{-2}	1.30×10^{-4}	1.07×10^{-3}	9.80×10^{-4}	4.10×10^{-3}
Al (108 atoms)	1.67×10^{-2}	9.33×10^{-5}	9.82×10^{-5}	1.13×10^{-4}	1.87×10^{-3}
Al (256 atoms)	3.93×10^{-2}	5.60×10^{-4}	4.18×10^{-4}	2.03×10^{-4}	6.67×10^{-3}
Al (500 atoms)	3.96×10^{-2}	4.11×10^{-4}	2.41×10^{-4}	5.04×10^{-4}	8.52×10^{-3}
Al vacancy defects	1.92×10^{-2}	9.80×10^{-5}	1.42×10^{-4}	2.98×10^{-4}	3.85×10^{-3}
Strain imposed Al	2.54×10^{-2}	1.75×10^{-4}	8.91×10^{-4}	6.64×10^{-4}	3.11×10^{-3}

Table B.4

Accuracy of the ML predicted electron density in terms of the L^1 norm per electron, calculated as $\frac{1}{N_e} \times \int_\Omega \left| \rho^{\text{scaled}}(\mathbf{r}) - \rho^{\text{DFT}}(\mathbf{r}) \right| d\mathbf{r}$, for various test cases for an FCC aluminum bulk system (N_e is the number of electrons in the system). Also shown in the table are errors in the different energies as computed from ρ^{scaled}. The test data set for post-processing was chosen such that it covered examples from all system sizes, configurations, and temperatures. For calculating the relevant energies, ρ^{scaled} was used as the initial guess for the electron density, and a single Hamiltonian diagonalization step was performed. Energies were then computed.

B.8 Calculation of the bulk modulus for aluminum

We show a comparison between some material properties calculated using the electron density predicted by the ML model, and as obtained through DFT calculations.

Case	Accuracy of electron density (L^1 norm per electron)	Ground-state energy (Ha/atom)	Exch. Corr. energy (Ha/atom)	Fermi level (Ha)	Max error in eigenvalue (Ha)
Entire test data set	1.93×10^{-2}	1.47×10^{-4}	9.34×10^{-4}	1.43×10^{-3}	7.29×10^{-3}
$Si_{0.5}Ge_{0.5}$ (64 atoms)	1.51×10^{-2}	8.08×10^{-5}	1.40×10^{-3}	8.71×10^{-4}	5.07×10^{-3}
$Si_{0.5}Ge_{0.5}$ (216 atoms)	1.90×10^{-2}	1.18×10^{-4}	2.50×10^{-4}	3.08×10^{-4}	4.99×10^{-3}
$Si_{0.5}Ge_{0.5}$ (512 atoms)	2.50×10^{-2}	2.57×10^{-4}	3.70×10^{-4}	1.32×10^{-3}	1.27×10^{-2}
$Si_{0.5}Ge_{0.5}$ vacancy defects	1.70×10^{-2}	9.68×10^{-5}	2.36×10^{-4}	2.82×10^{-3}	6.85×10^{-3}
Si_xGe_{1-x} ($x \neq 0.5$)	2.39×10^{-2}	2.54×10^{-4}	2.41×10^{-3}	1.25×10^{-3}	9.36×10^{-3}

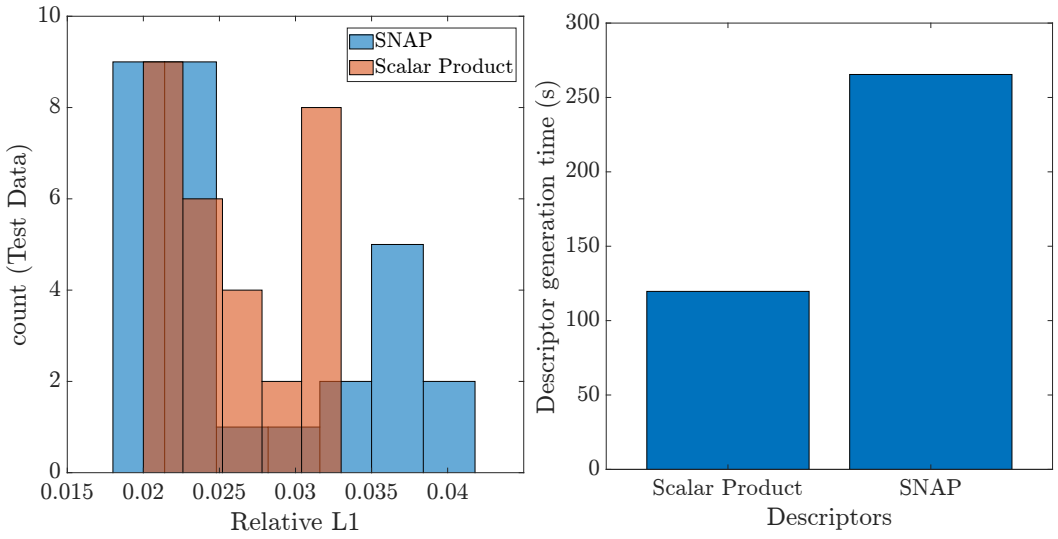
Table B.5

Accuracy of the ML predicted electron density in terms of L^1 norm per electron, calculated as $\frac{1}{N_e} \times \int_\Omega \left| \rho^{\text{scaled}}(\mathbf{r}) - \rho^{\text{DFT}}(\mathbf{r}) \right| d\mathbf{r}$, for various test cases for $Si_{0.5}Ge_{0.5}$ (N_e is the number of electrons in the system). Also shown in the table are errors in the different energies as computed from ρ^{scaled}. The test data set for post-processing was chosen such that it covered examples from all system sizes and temperatures. For calculating the relevant energies, ρ^{scaled} was used as the initial guess for the electron density, and a single Hamiltonian diagonalization step was performed. Energies were then computed. For Si_xGe_{1-x}, we used $x = 0.40, 0.45, 0.55, 0.60$.

Specifically, we compute the optimum lattice parameter and the bulk modulus for aluminum — these corresponding to the first and second derivatives of the post-processed energy curves (Fig. 3.5 of the main text), respectively. A summary of our results can be found in Table B.6. It can be seen that bulk modulus differs by only about 1%, while the lattice parameters are predicted with even higher accuracy. Notably, the predicted lattice parameter and the bulk modulus are very close to experimental values [1], and the deviation from experiments is expected to decrease upon using larger supercells to simulate the bulk, a trend also seen in Table B.6. This is consistent with the overall results shown in the main manuscript and further reinforces the predictive power of our model for non-ideal systems.

B.9 Comparison with models based on other descriptors

In the main text, we have presented errors achieved in electron density prediction by our model. The results indicate that our approach is generally as accurate as (and in some cases outperforms) previous work [44, 48]. To further compare it with existing similar approaches, we compare it with electron density predictions made via the well known SNAP descriptors [53, 243]. Specifically, we have compared the relative L1 error (as defined in [48]) on 29 test snapshots using the dataset of an Aluminum system with 32 atoms. We used the same training dataset and employed a neural network for both the descriptors. Both the descriptors yield nearly identical L1 errors (although the distribution of errors is different as shown in Fig. B.10). At the same time, the calculation of the scalar product descriptors employed here exhibits computational efficiency, requiring about 50% less time than generation of the SNAP descriptors. To ensure a fair and accurate comparison of descriptor computation time, the computations for both descriptors were performed on a single-core CPU. We utilized the data of Be 128 atoms provided by [174] and the SNAP code provided by [53, 244], for comparing descriptor calculation time.



(a) Error Histogram (b) Time comparison

Figure B.10: Comparison with SNAP descriptors

Material property	$2 \times 2 \times 2$ supercell	$3 \times 3 \times 3$ supercell
Lattice parameter (Bohr)	7.4294 (7.4281)	7.5208 (7.5188)
Bulk modulus (GPa)	92.2774 (92.7708)	75.7977 (76.3893)

Table B.6

A comparison between the calculated lattice parameter and the bulk modulus for aluminum using ρ^{ML} and ρ^{DFT} (DFT values in parentheses). We observe that the predicted lattice parameter closely matches the value given by DFT calculations. The "true" optimized lattice parameter for Al, using a fine k-space mesh, is found to be 7.5098 Bohr while experimental values are about 7.6 Bohr [2]). The ML predicted value of the bulk modulus matches the DFT value very closely, which itself is very close to the experimental value of approximately 76 GPa [1], at room temperature.

Appendix C

Letters of Permission

The contents of Chapter 2 are published in the following journal article:Machine learning based prediction of the electronic structure of quasi-one-dimensional materials under strain, Shashank Pathrudkar, Hsuan Ming Yu, Susanta Ghosh, and Amartya S. Banerjee, Phys. Rev. B 105, 195141.

As per the APS website,

"As the author of an APS-published article, may I include my article or a portion of my article in my thesis or dissertation?

Yes, the author has the right to use the article or a portion of the article in a thesis or dissertation without requesting permission from APS, provided the bibliographic citation and the APS copyright credit line are given on the appropriate pages."