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

Shashank Pathrudkar [1], Hsuan Ming Yu [2], Susanta Ghosh [1,3] and Amartya S. Banerjee [2,*]

[1]*Department of Mechanical Engineering–Engineering Mechanics, Michigan Technological University, Houghton, Michigan 49931, USA*
[2]*Department of Materials Science and Engineering, University of California, Los Angeles, California 90095, USA*
[3]*Center for Data Sciences, Michigan Technological University, Houghton, Michigan 49931, USA*

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 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 nanoassemblies, 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 postprocessing, 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 multiscale modeling of such systems.

## I. 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 [1–3], chemistry [4,5], and engineering fields like manufacturing [6,7], applied thermodynamics [8,9], and miscellaneous others (see, e.g., [10–13]). 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 [14,15], 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 [16–23]. In particular, the use of high-throughput density functional theory

(DFT) [24,25] calculations in conjunction with machine learning techniques has attracted much attention as a powerful tool for materials discovery [26–30]. 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 band gap, chemical properties like adsorption and formation energies, and mechanical properties like Young's modulus and fracture toughness [11,31–38]. 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 [39–51]. 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

*asbanerjee@ucla.edu

efforts of machine learning based prediction of electronic fields to broader classes of nanomaterials, specifically, quasi-one-dimensional (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 [25] or by improving exchange-correlation functionals used in Kohn-Sham theory [40,52–55]. 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 [46]. 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 [50] 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 [56–62], 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 [63] and multiscale modeling [64]. 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 [65–71]. 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 wirelike nanostructures. Atomic relaxation effects as a response to the applied strains are automatically included, by driving the Hellman-Feynman forces [72] 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 multiwalled carbon nanotubes [73]. 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 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 [40,50]. 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 [74,75], 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 0.02 eV, while the band-gap 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 quasirandom 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 postprocessed 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 postprocessing 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 Sec. II. 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 an approach to predict atomic coordinates are explained in Sec. III. Postprocessing of machine learning predicted electronic fields to evaluate various energy components, band structures, and atomic coordinates is explained in Sec. IV. Next, we validate the machine learning model and quantify its accuracy in Sec. V. We also comment on the model interpretability. We end with our conclusions and a discussion of future research directions.

## II. 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 $3 \times 3$ matrices will be denoted using uppercase boldface. Cartesian, cylindrical, and helical coordinates will be denoted as $(x, y, z)$, $(r, \vartheta, z)$, and $(r, \theta_1, \theta_2)$, respectively, and the relation between these is

$$r = \sqrt{x^2 + y^2}, \quad \theta_1 = \frac{z}{\tau},$$

$$\theta_2 = \frac{1}{2\pi}\text{arctan2}\,(y, x) - \alpha\frac{z}{\tau} = \frac{\vartheta}{2\pi} - \alpha\frac{z}{\tau}. \quad (1)$$

Here, $\alpha$ is related to the twist in the system as explained below.

### A. 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 [66,69,76–80]. Thus if the atoms of the system have positions

$$\mathcal{S} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \cdots : \mathbf{p}_i \in \mathbb{R}^3\}, \quad (2)$$

then we may identify a discrete group of isometries:

$$\mathcal{G} = \{\Upsilon_{m,n} = (\mathbf{R}_{(2\pi m\alpha + n\Theta)}|m\tau\mathbf{e_Z}) : m \in \mathbb{Z}, n = 0, 1,$$
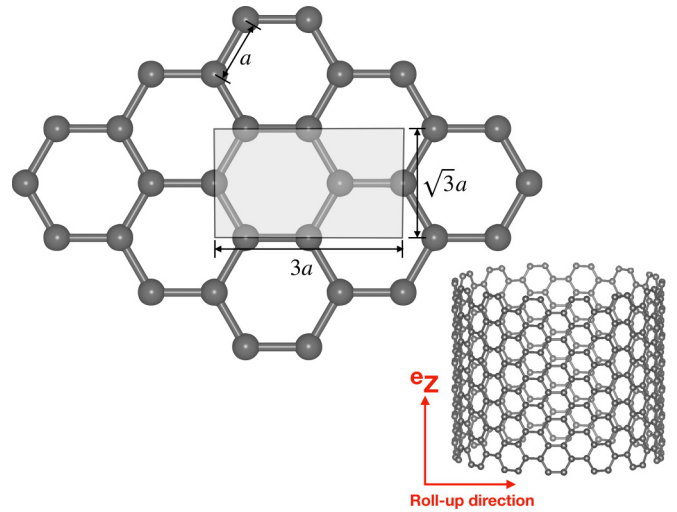$$\times \ldots, \mathfrak{N} - 1\}, \quad (3)$$



FIG. 1. Rollup construction of an undeformed armchair carbon nanotube, starting from a graphene sheet. The four 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 Å .

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

$$\mathcal{P} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \ldots, \mathbf{r}_M : \mathbf{r}_i \in \mathbb{R}^3\}, \quad (4)$$

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

$$\mathcal{S} = \bigcup_{\substack{m\in\mathbb{Z} \\ n=0,1,\ldots,\mathfrak{N}-1}} \bigcup_{i=1}^{M} \mathbf{R}_{(2\pi m\alpha + n\Theta)}\mathbf{r}_i + m\tau\,\mathbf{e_Z}. \quad (5)$$

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 $\mathbf{e_Z}$ and translation vector $m\tau\mathbf{e_Z}$. 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 $\tau$ is related to the axial pitch of the structure and it can capture extensions and compressions about the axis, while the scalar $\alpha$ 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 $0 \leqslant \alpha < 1$, with $\alpha = 0$ representing untwisted structures. For undeformed armchair carbon nanotubes, the value of $\tau$ as suggested by the "rollup construction" [77,81] is $\sqrt{3}a$, where $a$ is the interatomic bond length of graphene (Fig. 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 [66,69], 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.

### B. Helical Density Functional Theory (Helical DFT)

We use Helical Density Functional Theory (Helical DFT) [66,69] 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 $R_{\text{in}}$ and $R_{\text{out}}$, a suitable fundamental domain is the following region (expressed in cylindrical coordinates):

$$\mathcal{D} = \left\{ (r, \vartheta, z) \in \mathbb{R}^3 : R_{\text{in}} \right.$$
$$\left. \leqslant r \leqslant R_{\text{out}}, \frac{2\pi\alpha z}{\tau} \leqslant \vartheta \leqslant \frac{2\pi\alpha z}{\tau} + \Theta, 0 \leqslant z \leqslant \tau \right\}. \quad (6)$$

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 [65,66,82,83]

$$\widehat{\mathcal{G}} = \left\{ e^{2\pi i \left( m\eta + \frac{n\nu}{\mathfrak{N}} \right)} : \eta \in \left[ -\frac{1}{2}, \frac{1}{2} \right); \nu = \{0, 1, \dots, \mathfrak{N} - 1\} \right\}. \quad (7)$$

Accordingly, Helical DFT uses $(\eta, \nu)$ 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

$$\mathfrak{H}^{\text{KS}} \psi_j(\mathbf{x}; \eta, \nu) = \lambda_j(\eta, \nu) \psi_j(\mathbf{x}; \eta, \nu),$$
$$\mathfrak{H}^{\text{KS}} = -\frac{1}{2}\Delta + V_{\text{xc}} + \Phi + \mathcal{V}_{\text{nl}}, \quad (8)$$

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

$$\psi_j(\Upsilon_{m,n} \circ \mathbf{x}; \eta, \nu) = e^{-2\pi i \left( m\eta + \frac{n\nu}{\mathfrak{N}} \right)} \psi_j(\mathbf{x}; \eta, \nu). \quad (9)$$

In the above, $\mathfrak{H}^{\text{KS}}$ denotes the Kohn-Sham operator, $V_{\text{xc}}$ denotes the exchange-correlation potential, $\Phi$ 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 $\mathcal{V}_{\text{nl}}$ denotes the nonlocal pseudopotential operator. The field $\Phi$ obeys the following Poisson problem in terms of the electron density $\rho$ and the nuclear pseudocharge field $b$:

$$-\Delta\Phi = 4\pi(\rho + b). \quad (10)$$

The nonlocal pseudopotential operator can be expressed in Kleinman-Bylander form [84] as

$$\mathcal{V}_{\text{nl}} = \sum_{i=1}^{M} \sum_{p \in \Gamma_i} \gamma_{i,p} \hat{\chi}_{i,p}(\cdot; \eta, \nu; \mathbf{r}_i) \, \overline{\hat{\chi}_{i,p}(\cdot; \eta, \nu; \mathbf{r}_i)}, \quad (11)$$

with $\chi_{i,p}$, $\gamma_{i,p}$, and $\Gamma_i$ 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 nonlocal pseudopotential operator $\mathcal{V}_{\text{nl}}$ are computed explicitly, along with a suitable starting guess for the electron density $\rho$. Following these computations, the symmetry-adapted Kohn-Sham equations [Eq. (8)] are solved self-consistently [85]. 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 [66,69].

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 (Sec. III C), 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 $\mathcal{V}_{\text{nl}}$.

### C. Use of helical coordinates

The fundamental domain $\mathcal{D}$ assumes the form of a cuboid $[R_{\text{in}}, R_{\text{out}}] \times [0, 1] \times [0, 1/\mathfrak{N}]$ in helical coordinates. Helical DFT uses a higher-order finite-difference scheme in these coordinates to discretize and solve the governing equations [66,69]. 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 $\theta_1$ 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 $N_r$, $N_{\theta_1}$, $N_{\theta_2}$ as the number of grid points along the $r$, $\theta_1$, and $\theta_2$ directions, respectively. The electronic fields are then represented as vectors in

dimension $N_r \times N_{\theta_1} \times N_{\theta_2}$, and the mesh spacings corresponding to these discretization choices are

$$h_r = \frac{R_{out} - R_{in}}{N_r}, \quad h_{\theta_1} = \frac{1}{N_{\theta_1}}, \quad h_{\theta_2} = \frac{1/\mathfrak{N}}{N_{\theta_2}}. \quad (12)$$

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

### D. Other details of first-principles calculations

All Helical DFT calculations described in this work use a four-atom fundamental domain as shown in Fig. 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 $h = 0.3$ bohrs, and by sampling 15 $k$ points in the $\eta$ direction. These discretization choices are sufficient to produce chemically accurate forces and ground-state energies for the Troullier-Martins norm-conserving pseudopotential [86] used to model the carbon atoms in this work [69]. Atomic relaxation is carried out using the fast inertial relaxation engine [87], and the structures are relaxed until each atomic force component drops below 0.001 Ha/bohrs. 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 $h = 0.25$ bohrs (resulting in $N_r \times N_{\theta_1} \times N_{\theta_2} \approx 60\,000$) and 21 $k$ points in the $\eta$ 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 [88] of the local density approximation [24], a 12th-order finite-difference discretization scheme [66,68,89–94], vacuum padding of 11 bohrs in the radial direction, and 1 milli-Hartree of smearing using the Fermi-Dirac distribution.

## III. 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), $\alpha$, the twist parameter, and $\tau$, 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 $(R_{avg}, \alpha, \tau)$ to the electronic fields $\rho$, $b$ of the deformed nanotubes:

$$\mathcal{H}: \{R_{avg}, \alpha, \tau\} \rightarrow \{\rho, b\}. \quad (13)$$

The objective of this work is to approximate this map $\mathcal{H}$ using a machine learning model. Inputs of this map $R_{avg}$, $\alpha$, and $\tau$ 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 Sec. II). 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 [14].

In this 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. Recently, a similar approach has been found to have excellent accuracy in high-dimensional predictions related to purely mechanical problems [73,95].

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.

### A. Design of experiments to explore the input space

We now describe the use of design of experiments (DoE) [96,97] techniques for efficient sampling in the input space. As described above, the triplet of input parameters $\{R_{avg}, \alpha, \tau\}$ specify a particular nanotube and the applied strains. The number 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
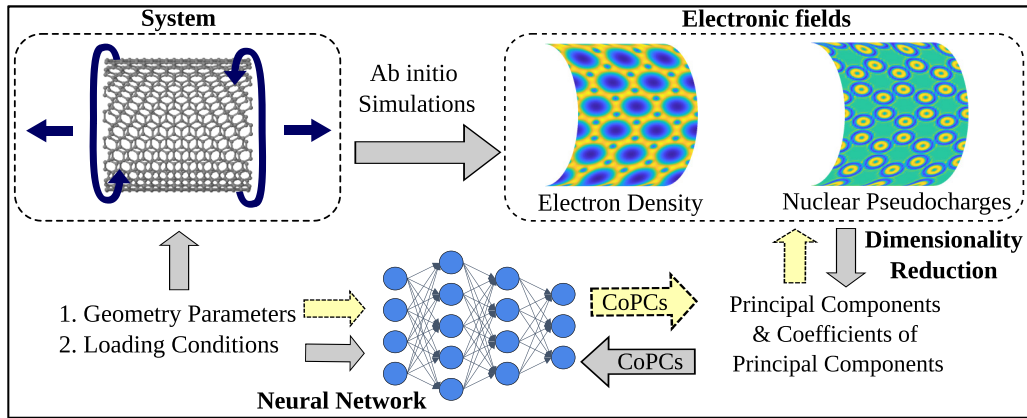
FIG. 2. Schematic of the present machine learning (ML) model and the data generation process via DFT simulations. The firm arrows show the steps for data generation and training, and the dashed arrows show the steps for prediction via the ML model.

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 [98–100]. To address this challenge, we generate sequences of quasirandom sampling points in the input space to reduce the number of simulations required for training an accurate ML model.

*Quasirandom 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 [98,101]. Commonly used space-filling designs include low-discrepancy sequences [102,103], good lattice points [104], latin hypercube sampling [105], and orthogonal latin hypercube sampling [106]. These methods are often evaluated based on their measure of uniformity [99,107,108], and such criteria suggest that optimal latin hypercube sampling [109] and Sobol sequences [103,110] 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 nonuniformly, but cover the space evenly [111], 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, until the desired accuracy is achieved in the ML model. Further details of the sampling procedure used are provided in Appendix A.

### B. 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) [112–115]. PCA reduces the dimensionality of the data in an unsupervised manner. 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, PCA allows one to obtain a lower-dimensional approximation $\tilde{\mathbf{x}}_i \in \mathbb{R}^K$, such that $K < d$, and

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^{K} c_{ij} \mathbf{v}_j + \boldsymbol{\mu}. \tag{14}$$

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 data set rather than being associated with a particular data point; therefore, all the points in the original data set 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 ($\rho$ and $b$), data generated by the DFT simulations. Specifically, in Helical DFT, the discretized grid of the fundamental domain of electronic fields has 89 points along the radial direction, 19 points along the $\theta_1$ direction, and 35 points along the $\theta_2$ direction. Given this discretization, the total number of grid points in the fundamental domain is 59 185. The electronic fields are therefore represented as a nearly 60 000-dimensional vector by the DFT simulations. PCA enables us to represent these high-dimensional electronic fields in terms of a few CoPCS only (7 and 15 for $\rho$ and $b$, respectively, as described in Sec. V A).

*Regression for the electronic fields in the reduced dimension:* We employ neural networks (NN) [14,116] 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 [73]. 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 (in a supervised manner) 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 $\rho$ and $b$, respectively, which use the same input parameters. In the input layer, we have three neurons, for the *input* parameters (i) $R_{\text{avg}}$, (ii) $\alpha$, and (iii) $\tau$. 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. (14). These two steps for inference via the ML model are shown in Fig. 2.

### C. 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 nonlocal part of the pseudopotential operator may be constructed (see Sec. II B). 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 [50] 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 C), and the amount of training data that was found to be adequate for predicting the electronic fields $\rho$ and $b$ accurately was found to result in unacceptable levels of error while predicting the nuclear coordinates. This led us to devise a 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 (Sec. V B), and it already contains the nuclear coordinate information in principle. Our proposed strategy to find nuclear coordinates is based on a clustering approach and identifies the nuclear coordinates from the nuclear pseudocharge field in an unsupervised manner.

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 nonoverlapping. 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 data sets into smaller subgroups in an unsupervised manner, such that the data points in each subgroup share some common attributes [14]. One of the most widely used and successful clustering algorithms is DBSCAN (density-based spatial clustering of
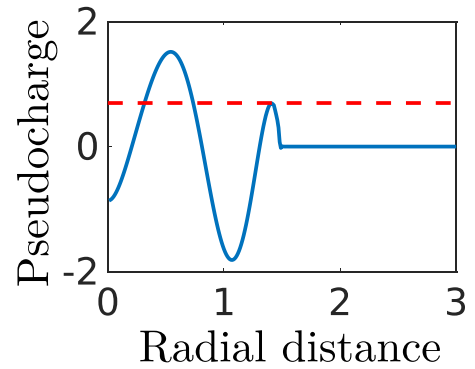


FIG. 3. Atomic pseudocharge as a function of distance (in bohrs) from the atom for the Troullier-Martins pseudopotential for carbon used in this work. The dashed red line indicates the truncation level employed before the DBSCAN procedure is used.

applications with noise) [117,118]. 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 *a priori*. 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 $\theta_1$ and $\theta_2$ 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 Fig. 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 Fig. 3):

$$\mathcal{T}: b(r, \theta_1, \theta_2) \rightarrow \bar{b}(r, \theta_1, \theta_2),$$

$$\bar{b}(r, \theta_1, \theta_2) = \begin{cases} b(r, \theta_1, \theta_2), & \text{if } b(r, \theta_1, \theta_2) > c_t, \\ 0, & \text{if } b(r, \theta_1, \theta_2) \leqslant c_t. \end{cases} \quad (15)$$
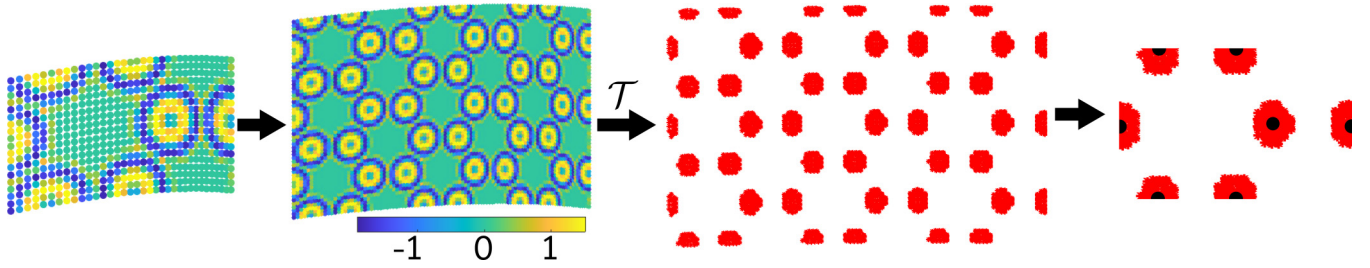
FIG. 4. Cluster formation from nuclear pseudocharge field to determine nuclei position. A slice of the pseudocharge field at the average radial coordinate of the atoms in the fundamental domain is shown. Red clusters show the positive charge around the nucleus and the black dots are nuclei. The pseudocharge 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.

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 as shown in Fig. 4.

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 II). Comparison of the accuracy of our clustering-based approach with that of nuclear coordinate predictions using a standard neural network is presented in Appendix C.

## IV. POSTPROCESSING 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 $\rho^{\mathrm{ML}}(\mathbf{x})$ and $b^{\mathrm{ML}}(\mathbf{x})$ 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 I within the section on Results), we find it useful to scale the ML predicted fields for postprocessing purposes [48], as shown below:

$$\rho^{\mathrm{scaled}}(\mathbf{x}) = \rho^{\mathrm{ML}}(\mathbf{x}) \frac{N_{\mathrm{e}}}{\int_{\mathcal{D}} \rho^{\mathrm{ML}}(\mathbf{x})\, d\mathbf{x}},$$

$$b^{\mathrm{scaled}}(\mathbf{x}) = b^{\mathrm{ML}}(\mathbf{x}) \frac{-N_{\mathrm{e}}}{\int_{\mathcal{D}} b^{\mathrm{ML}}(\mathbf{x})\, d\mathbf{x}}. \tag{16}$$

Using these scaled fields, we compute the net electrostatic potential $\Phi$ via iterative solution of Eq. (10) using preconditioned GMRES [119] iterations. The exchange-correlation potential $V_{\mathrm{xc}}$ is directly computed from the electron density. Next, we use a clustering-based unsupervised learning technique (see Sec. III C) to pick out the nuclear coordinates from the nuclear pseudocharge field and use it to set up the nonlocal pseudopotential operator $\mathcal{V}_{\mathrm{nl}}$. Thereafter, we diagonalize the Kohn-Sham Hamiltonian [Eq. (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) [120] and Arnoldi iterations [121] to carry out the diagonalization, and

TABLE I. Table showing NRMSE for ML predicted $\rho$ and $b$ for various test cases. Also shown are errors in the integrals of electronic fields over the fundamental domain. $R_{\mathrm{avg}}$ and $\tau$ values are in bohrs.

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

TABLE II. Errors in various postprocessed quantities. Refer to Eq. (18) and related discussion for interpretation of the various energetic terms. $R_{\text{avg}}$ and $\tau$ values are in bohrs.

| Case | Ground-state energy $\mathcal{F}^{\text{HF}}$ (Ha/atom) | Exch. Corr. energy $E_{\text{xc}}$ (Ha/atom) | Electrostatic term $\widetilde{E}_{\text{el}}$ (Ha/atom) | Nuclear self-energy and correction term $E_{\text{sc}}$ (Ha/atom) | Band energy $E_{\text{band}}$ (Ha/atom) | Band gap (eV) | Atomic coordinates $\mathbf{r}_i$ (bohrs) |
|---|---|---|---|---|---|---|---|
| Random test data point | $9.0 \times 10^{-4}$ | $4.1 \times 10^{-5}$ | $7.1 \times 10^{-5}$ | $1.5 \times 10^{-4}$ | $9.9 \times 10^{-4}$ | 0.017 | 0.026 |
| $R_{\text{avg}} = 40.64$ , $\alpha = 0.0015$, $\tau = 4.7052$ | | | | | | | |
| Test data point with unknown $R_{\text{avg}}$ | $6.7 \times 10^{-4}$ | $1.9 \times 10^{-5}$ | $4.2 \times 10^{-4}$ | $4.3 \times 10^{-4}$ | $6.7 \times 10^{-4}$ | 0.018 | 0.028 |
| $R_{\text{avg}} = 49.51$ , $\alpha = 0.0020$, $\tau = 4.5052$ | | | | | | | |
| Test data point with unknown $\alpha$ | $3.6 \times 10^{-3}$ | $8.9 \times 10^{-5}$ | $2.1 \times 10^{-3}$ | $3.2 \times 10^{-4}$ | $1.2 \times 10^{-3}$ | 0.042 | 0.019 |
| $R_{\text{avg}} = 35.55$ , $\alpha = 0.00125$, $\tau = 4.6052$ | | | | | | | |
| Test data point with unknown $\tau$ | $2.2 \times 10^{-4}$ | $5.4 \times 10^{-5}$ | $2.7 \times 10^{-3}$ | $4.7 \times 10^{-5}$ | $2.9 \times 10^{-3}$ | 0.008 | 0.023 |
| $R_{\text{avg}} = 53.32$ , $\alpha = 0.0015$, $\tau = 4.5552$ | | | | | | | |
| Test data point with unknown $R_{\text{avg}}, \alpha, \tau$ | $6.5 \times 10^{-4}$ | $7.4 \times 10^{-5}$ | $1.9 \times 10^{-3}$ | $1.8 \times 10^{-4}$ | $1.4 \times 10^{-3}$ | 0.008 | 0.022 |
| $R_{\text{avg}} = 49.51$ , $\alpha = 0.00125$, $\tau = 4.5552$ | | | | | | | |
| Test data point with unknown $R_{\text{avg}}, \alpha, \tau$ | $1.35 \times 10^{-4}$ | $8.0 \times 10^{-5}$ | $1.8 \times 10^{-3}$ | $4.1 \times 10^{-4}$ | $1.3 \times 10^{-3}$ | 0.002 | 0.034 |
| $R_{\text{avg}} = 30.46$ , $\alpha = 0.00075$, $\tau = 4.6552$ | | | | | | | |

initialize the calculations using random wave-function 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. From all this information, band-structure-dependent quantities, such as the value of the band gap and its location in $(\eta, \nu)$ space, can be calculated.

Using the aforementioned postprocessed quantities, the ground-state free energy per unit fundamental domain may be calculated as [69]

$$\mathcal{F} = E_{\text{kin}} + E_{\text{xc}} + E_{\text{nl}} + E_{\text{el}} - T_e S, \qquad (17)$$

with the terms on the right-hand side denoting the electronic kinetic energy, the exchange correlation energy, the nonlocal pseudopotential energy, the electrostatic energy, and the electronic entropy contribution at temperature $T_e$, respectively. Alternately, a more accurate estimate for the ground-state free energy per unit fundamental domain may be obtained using the Harris-Foulkes functional [122,123]

$$\mathcal{F}^{\text{HF}} = E_{\text{band}} + E_{\text{xc}} - \widetilde{E}_{\text{xc}} + \widetilde{E}_{\text{el}} + E_{\text{sc}} - T_e S. \qquad (18)$$

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

$$E_{\text{band}} = 2 \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\mathfrak{N}} \sum_{\nu=0}^{\mathfrak{N}-1} \sum_{j=1}^{\infty} \lambda_j(\eta, \nu)\, g_j(\eta, \nu)\, d\eta, \qquad (19)$$

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

$$\widetilde{E}_{\text{xc}} = \int_{\mathcal{D}} V_{\text{xc}}(\rho(\mathbf{x}))\rho(\mathbf{x})\, d\mathbf{x}. \qquad (20)$$

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

$$\widetilde{E}_{\text{el}} = \frac{1}{2} \int_{\mathcal{D}} [b(\mathbf{x}) - \rho(\mathbf{x})]\Phi(\mathbf{x})\, d\mathbf{x}. \qquad (21)$$

Finally, $E_{\text{sc}}$ accounts for nuclear pseudocharge self-interactions and overlap corrections [67], while the last

term is related to the electronic entropy contribution. Notably, in the above breakdown for the Harris-Foulkes energy, $E_{\text{xc}}$ and $\widetilde{E}_{\text{xc}}$ depend solely on the electron density field, $E_{\text{sc}}$ 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 $E_{\text{band}}$ depends on the Kohn-Sham operator eigenvalues (i.e., its dependence on $\rho$ 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 $\rho$, $b$, and other postprocessed quantities (such as the eigenstates) in the energetic sense (see Sec. V for more details).

## V. 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 and 0.009 s to predict the $\rho$ 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. Postprocessing 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 $\rho$ and $b$ requires about 12 and 15 minutes, respectively, measured using the same hardware setup.

### A. 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
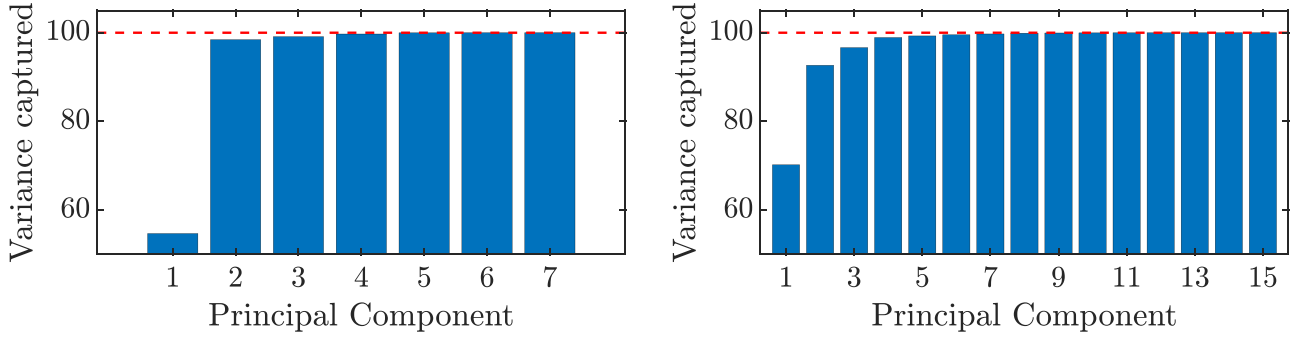
FIG. 5. Cumulative percentage of variance vs principal components for $\rho$ (left) and $b$ (right). The red dashed line shows 99.99% variance.

reconstruction error, we capture 99.99% variance of the data. As shown later (Sec. V C), 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 $\rho$ and 15 PCs in case of $b$ (Fig. 5).

*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 $\rho$ and $b$, respectively. Since 7 PCs in case of $\rho$ 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 B) 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 [124], and the Adam optimizer [125] 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 B.

### B. 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 $\rho$ and $b$, respectively (Fig. 6). In addition to the test data points described above, we have chosen three additional test data points where the input parameters were partially unseen during the training [(i) $R_{\text{avg}} = 49.51$ bohrs, $\alpha = 0.002$, $\tau = 4.5052$ bohrs; (ii) $R_{\text{avg}} = 35.55$ bohrs, $\alpha = 0.00125$, $\tau = 4.6052$ bohrs; (iii) $R_{\text{avg}} = 53.32$ bohrs, $\alpha = 0.0015$, $\tau = 4.5552$ bohrs]. 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 $\tau$ and $\alpha$ values present in the training data but the value of $R_{\text{avg}}$ 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) $R_{\text{avg}} = 49.51$ bohrs, $\alpha = 0.00125$, $\tau = 4.5552$ bohrs; (ii) $R_{\text{avg}} = 30.46$ bohrs, $\alpha = 0.00075$, $\tau = 4.6552$ bohrs]. 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. 7.

We quantify the error in the predicted electronic fields through the normalized root-mean-square error (NRMSE) [50]:

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{d} \sum_{i=1}^{d} \left( \rho_i^{\text{DFT}} - \rho_i^{\text{ML}} \right)^2}}{|\max(\rho^{\text{DFT}}) - \min(\rho^{\text{DFT}})|}. \quad (22)$$

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 I. 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 $\rho^{\text{ML}}$ and $b^{\text{ML}}$ over the fundamental domain. For a neutral system with $N_e$ electrons within the computational unit cell, the electron density and the
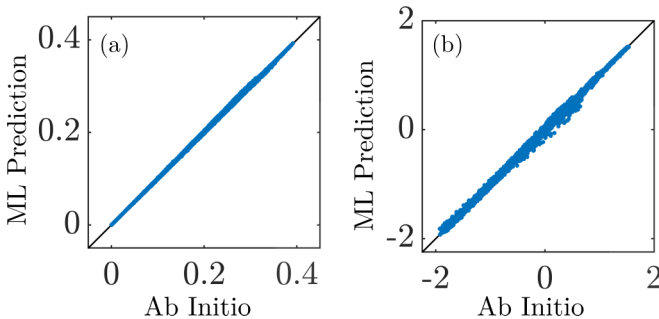


FIG. 6. Parity plots for (a) test data of $\rho$ ($R = 0.9949$), (b) test data of $b$ ($R = 0.9983$).

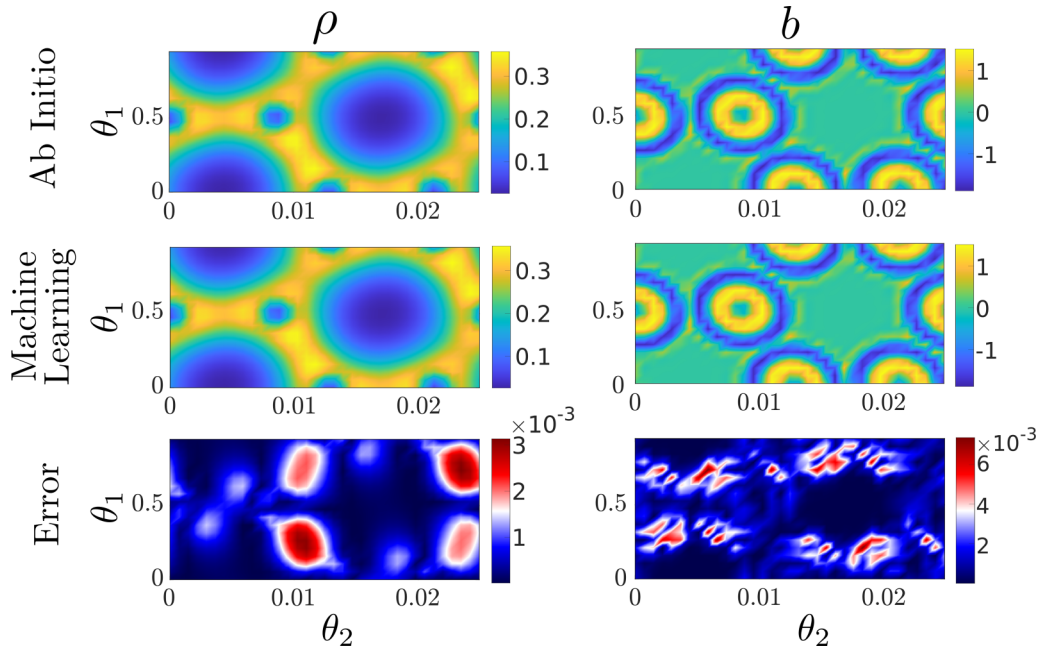FIG. 7. Comparison between ML predicted and DFT simulation obtained electronic fields for a test data point with all unknown input parameters ($R_{avg} = 49.51$ bohrs, $\alpha = 0.001\,25$, $\tau = 4.5552$ bohrs). A slice of the electronic fields at the average radial coordinate of the atoms in the fundamental domain is shown. The error is computed as $\frac{|\rho^{DFT} - \rho^{ML}|}{|\max(\rho^{DFT}) - \min(\rho^{DFT})|}$, similarly for $b$. Here, $\max(\cdot)$ and $\min(\cdot)$ denote maximum and minimum over the fundamental domain.

nuclear pseudocharge fields obey the normalization conditions $\int_{\mathcal{D}} \rho(\mathbf{x})\,d\mathbf{x} = N_e$ and $\int_{\mathcal{D}} b(\mathbf{x})\,d\mathbf{x} = -N_e$, 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 I, the errors associated with deviations from these constraints are quite low (0.000 625 particles or lower, per electron), indicating high-quality predictions of the electronic fields by the ML model.

## C. Prediction of nuclear coordinates, energies, and band structure

Finally, we postprocess 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 II). In general, the errors in the total ground-state energy, as computed through the Harris-Foulkes functional [Eq. (18)] are found to be appreciably smaller than the chemical accuracy threshold ($1.6 \times 10^{-3}$ Ha/atom), except for one of the cases which had an unseen value of $\alpha$. 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 $\tilde{E}_{xc}$ [Eq. (18)] also has a very similar behavior and is not shown in Table II. 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 $\alpha$ and/or $\tau$ unseen by the ML model. However, even in these cases, the errors are less than $3.0 \times 10^{-3}$ 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 [64].

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. 8. Undeformed armchair carbon nanotubes are metallic [68,126] but develop an oscillatory band gap as a function of imposed twist [69,126]. 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 $(\eta, \nu)$ are varied] is particularly error prone since it is the difference of two quantities. Additionally, as
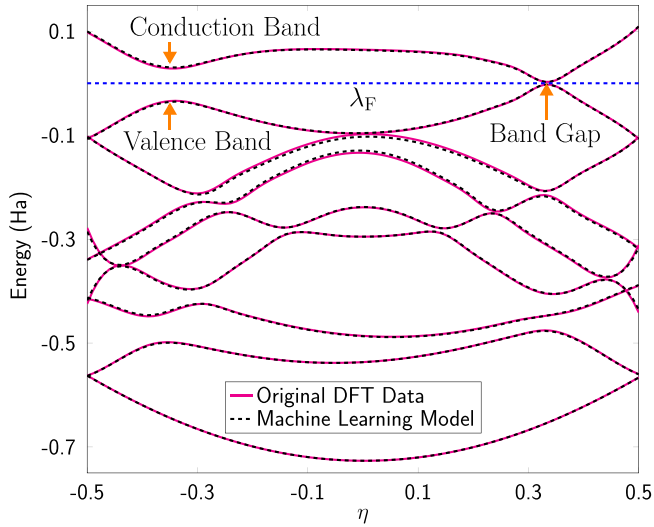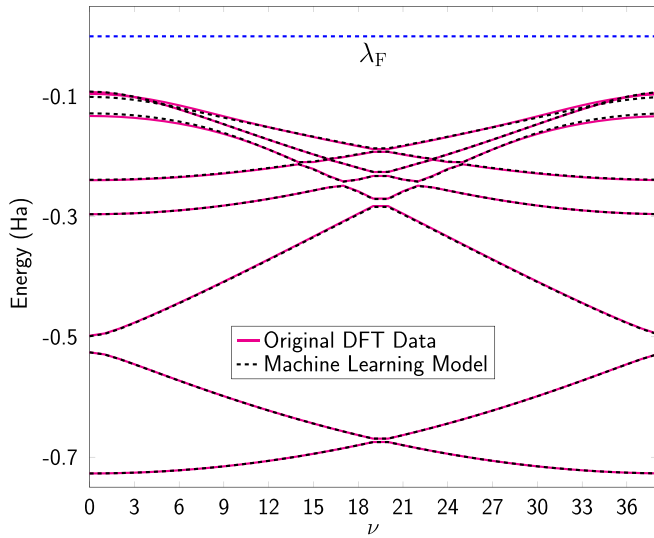
(a) Symmetry adapted band diagram in $\eta$, at $\nu = 2$.



(b) Symmetry adapted band diagram in $\nu$, at $\eta = 0$.

FIG. 8. Comparison of symmetry-adapted band diagrams produced by the Helical DFT method and the machine learning model (with postprocessing) for the unknown test data point with $R_{\mathrm{avg}} = 49.51$ bohrs, $\alpha = 0.00125$, and $\tau = 4.5552$ bohrs. The agreement appears excellent and the postprocessed 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 $\lambda_{\mathrm{F}}$ denotes the system's Fermi level.

the tube is deformed, the location and nature (i.e., direct vs indirect) of the band gap is expected to change [69]. However, the ML predicted location of the band gap was correct for every test case and its value was correct to about 0.02 eV or better, in almost every test case. In this regard, our approach to predicting band-structure-dependent quantities by means of a postprocessing step applied to the ML predicted electronic fields appears to be especially effective. In contrast, as shown in Appendix D, direct prediction of such quantities using a neural network which takes in the input parameters

$(R_{\mathrm{avg}}, \alpha, \tau)$, can be error prone, particularly for the relatively small amount of data that was required for accurate prediction of the electronic fields themselves. Overall, the ability of our approach to accurately and efficiently 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 [56,57,61].

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. We also anticipate that the approach described here, i.e., obtaining complex electronic-structure-dependent quantities through postprocessing of ML predicted electronic fields, instead of predicting such quantities through ML directly, is going to find utility in computational studies related to the polarization and transport properties of low-dimensional systems.

### D. 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 $\rho$ and $b$. Figure 9 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 $\rho$ and $b$ capture the most significant aspects of the variations in these quantities, with the modes of $\rho$ reflecting changes in charge density along the carbon-carbon bonds, and those of $b$ capturing shifts in the nuclear positions.

To elaborate on the above interpretations, we first recall that (see Sec. II C) in the absence of relaxation effects, an atom within the fundamental domain has the same values of $\theta_1$ and $\mathfrak{N}\theta_2$, regardless of the tube radius or the level of axial or 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 readjust 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 Fig. 9 appear to be capturing this "motion" of the pseudocharge centers (i.e., nuclear
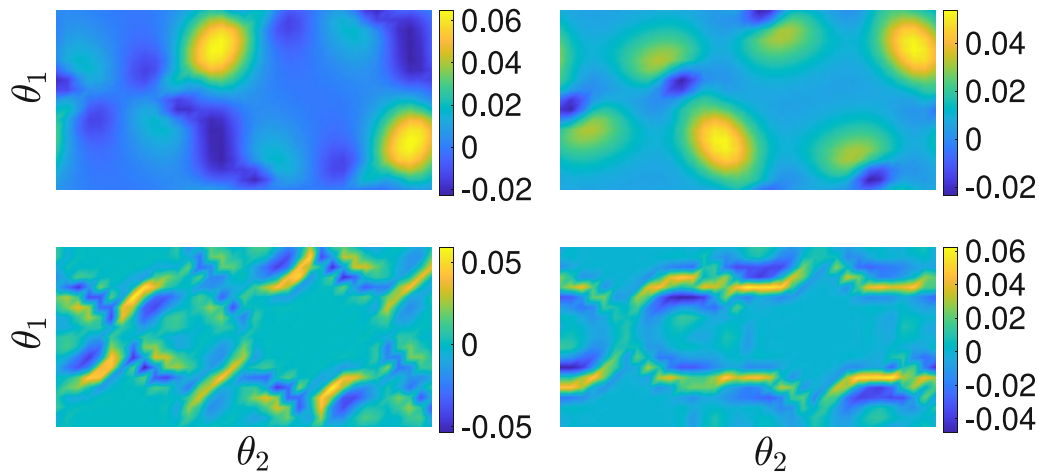
FIG. 9. First two principal components for $\rho$ (top) and $b$ (bottom). A slice of the PCA modes at the average radial coordinate of the atoms in the fundamental domain is shown.

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 $\rho$ 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 Fig. 1 and top row of Fig. 7), leading to the tilted appearance of the electron density lobes observable in Fig. 9. Finally, the presence of more wiggles in the plots for the PCA modes for $b$, as compared to those for $\rho$ 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 $\rho$ is more smeared out (also see top row of Fig. 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 (Fig. 5).

## VI. CONCLUSIONS AND FUTURE DIRECTIONS

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, quasirandom 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 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 nonlocal 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, band gaps, 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 [76,77]. Therefore, we anticipate that in conjunction with techniques for DFT calculations of large-scale systems [127,128], the current ML model and its extensions are likely to help in the exploration of novel phases of chiral matter [63,129] and compositionally complex nanotubes [130]. Development of machine learning models that can capture atomic species specific features, as well as ones that can perform the postprocessing steps associated with calculations of quantities such as energy components and band diagrams, serve as worthy directions for future research.

## APPENDIX A: DATA GENERATION

We use the following bounds for the input space in order to generate the data: $R_{\text{avg}} \in [20.32, 101.60]$ bohrs, $\alpha \in [0, 0.0025]$, and $\tau \in [4.4052, 4.8052]$ bohrs. 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 $\alpha$ and $\tau$ are 0.0005 and 0.1 bohrs, respectively. The maximum applied torsional strain of about 3.86 degrees/nm, close to the regime in which torsional instabilities may start to appear [77], 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 $R_{\text{avg}}$ 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, until 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 Sec. III A these three sets of simulations successively refine the input space. Figure 10 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.
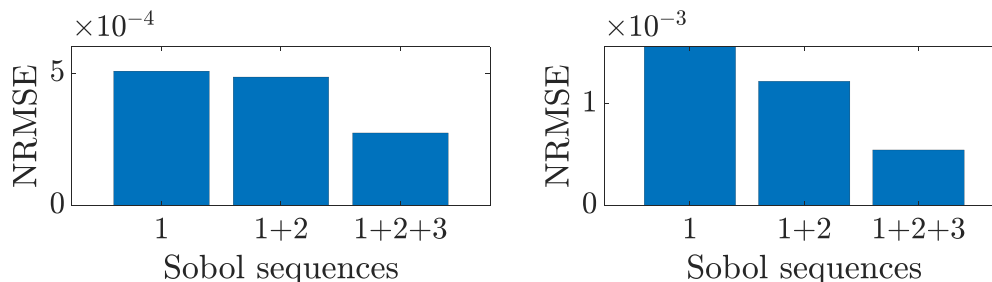
## APPENDIX B: TRAINING OF NEURAL NETWORKS

The learning curves for the neural networks $\mathcal{N}_1$ (for $\rho$) and $\mathcal{N}_2$ (for $b$) are presented in Fig. 11. The loss function used to train the neural networks is computed on CoPCs and is given in Eq. (B1) below.

### 1. 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.

### 2. 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 [131]. Figure 12 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.

### 3. 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 [132,133].

### 4. Learning rate

The learning rate was set to 0.001 as suggested in [125]. Other parameters pertinent to the Adam optimizer were set at suggested values [125] based on good results for other machine learning problems ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$).
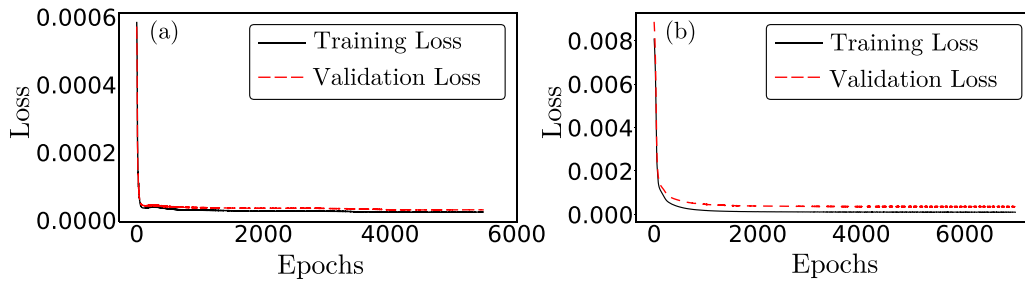


FIG. 10. Mean of NRMSE for test data when the machine learning model is trained using Sobol sets. (Left) Error bars for $\rho$, (right) error bars for $b$.

FIG. 11.  (a) Learning curve for $\mathcal{N}_1$, (b) learning curve for $\mathcal{N}_2$.

#### 5. Number of iterations

In order to avoid overfitting and help ensure good generalization performance of the ML model, we used early stopping [134,135]. 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 200 000.

#### 6. 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 [124], and overcomes the individual drawbacks of each. The loss function including $\mathcal{L}_1$ and $\mathcal{L}_2$ regularization can be written as

$$\tilde{\mathcal{J}}(y, \mathcal{N}(x, \bar{\mathbf{w}})) = \mathcal{J}(y, \mathcal{N}(x, \bar{\mathbf{w}})) + \lambda_1 ||\bar{\mathbf{w}}||_1 + \lambda_2 ||\bar{\mathbf{w}}||_2,$$
$$\lambda_1, \lambda_2 \in \mathbb{R}. \tag{B1}$$

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 $\mathcal{N}_2$.
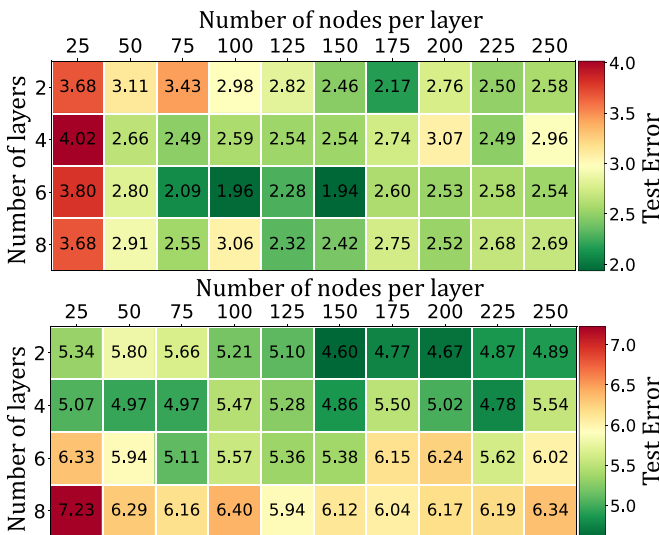


FIG. 12.  (Top) Test error ($\times 10^{-5}$) for various architectures of $\mathcal{N}_1$ (trained for $\rho$). (Bottom) Test error ($\times 10^{-4}$) for various architectures of $\mathcal{N}_2$ (trained for $b$.)

### APPENDIX C: 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: $R_{\mathrm{avg}}$, $\alpha$, and $\tau$. 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. 13. 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 nonoverlapping, atom-centered, spherically symmetric charge distributions), as opposed to the neural-network model which does not incorporate such information.

### APPENDIX D: COMPARISON OF OUR APPROACH WITH A DIRECT NEURAL-NETWORK-BASED PREDICTION OF THE BAND-GAP LOCATION

It is well known [69,126] that the location of the band gap in an armchair carbon nanotube shifts in the $(\eta, \nu)$ space, as the tube is distorted. For small twists in particular, the band gap continues to be a direct one and remains at $\eta = \frac{1}{3}$, while transitioning to different values of the cyclic symmetry index $\nu$. We show in Table III below that for the same amount of training data, this behavior is correctly captured by
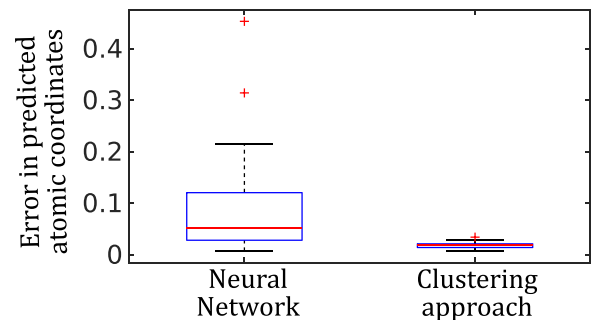


FIG. 13.  Error in predicted atomic coordinates (in bohrs), i.e., the distance between true and predicted nucleus positions, using a neural network and the DBSCAN-based clustering approach.

TABLE III. Comparison of our approach (ML model followed by postprocessing) with direct neural-network- (NN-) based prediction of cyclic symmetry index ($\nu$) associated with band-gap location. VBM denotes the valence band maximum and CBM denotes the conduction band minimum. DFT denotes reference first-principles results calculated using Helical DFT. $R_{avg}$ and $\tau$ values are in bohrs.

| Case | | $\nu$ for VBM | | | $\nu$ for CBM | |
|---|---|---|---|---|---|---|
| | DFT | Our Approach | NN | DFT | Our Approach | NN |
| Random test data point | | | | | | |
| $R_{avg} = 40.64$ , $\alpha = 0.0015$, $\tau = 4.7052$ | 2 | 2 | 1.76 | 2 | 2 | 1.82 |
| Test data point with unknown $R_{avg}$ | | | | | | |
| $R_{avg} = 49.51$ , $\alpha = 0.0020$, $\tau = 4.5052$ | 3 | 3 | 3.35 | 3 | 3 | 3.15 |
| Test data point with unknown $\alpha$ | | | | | | |
| $R_{avg} = 35.55$ , $\alpha = 0.00125$, $\tau = 4.6052$ | 1 | 1 | 1.06 | 1 | 1 | 1.12 |
| Test data point with unknown $\tau$ | | | | | | |
| $R_{avg} = 53.32$ , $\alpha = 0.0015$, $\tau = 4.5552$ | 3 | 3 | 2.91 | 3 | 3 | 2.86 |
| Test data point with unknown $R_{avg}, \alpha, \tau$ | | | | | | |
| $R_{avg} = 49.51$ , $\alpha = 0.00125$, $\tau = 4.5552$ | 2 | 2 | 2.06 | 2 | 2 | 2.06 |
| Test data point with unknown $R_{avg}, \alpha, \tau$ | | | | | | |
| $R_{avg} = 30.46$ , $\alpha = 0.00075$, $\tau = 4.6552$ | 0 | 0 | 0.4 | 0 | 0 | 0.36 |

our method of predicting band-structure-dependent quantities (i.e., carrying out a low-overhead postprocessing step on the ML predicted smooth electronic fields), in contrast to a direct neural-network-based prediction.

[1] 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.*, J. R. Soc., Interface **15**, 20170387 (2018).

[2] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, Mol. Pharmaceutics **13**, 1445 (2016).

[3] P. Thiagarajan, P. Khairnar, and S. Ghosh, IEEE Trans. Med. Imaging **41**, 815 (2022).

[4] O. V. Prezhdo, J. Phys. Chem. Lett. **11**, 9656 (2020).

[5] A. Richardson, B. M. Signor, B. A. Lidbury, and T. Badrick, Clin. Biochem. **49**, 1213 (2016).

[6] T. Wuest, D. Weimer, C. Irgens, and K.-D. Thoben, Prod. Manuf. Res **4**, 23 (2016).

[7] D. T. Pham and A. A. Afify, Proc. Inst. Mech. Eng., Part B **219**, 395 (2005).

[8] J. Chen, R. Randall, B. Peeters, W. Desmet, and H. Van der Auweraer, *Key Engineering Materials*, Vol. 518 (Trans Tech Publications, Freienbach, Switzerland, 2012), pp. 47–56.

[9] F. Wang, S. Ma, H. Wang, Y. Li, and J. Zhang, Control Eng. Practice **80**, 26 (2018).

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, J. Comput. Phys. **378**, 686 (2019).

[11] X. Liu, C. E. Athanasiou, N. P. Padture, B. W. Sheldon, and H. Gao, Acta Mater. **190**, 105 (2020).

[12] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Annu. Rev. Fluid Mech. **52**, 477 (2020).

[13] R. Mattey and S. Ghosh, Comput. Methods Appl. Mech. Eng. **390**, 114474 (2022).

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*, Vol. 128 (Springer-Verlag, New York, 2006).

[15] B. Efron and T. Hastie, *Computer Age Statistical Inference*, Vol. 5 (Cambridge University Press, Cambridge, 2016).

[16] K. Rajan, Annu. Rev. Mater. Res. **45**, 153 (2015).

[17] J.-P. Correa-Baena, K. Hippalgaonkar, J. van Duren, S. Jaffer, V. R. Chandrasekhar, V. Stevanovic, C. Wadia, S. Guha, and T. Buonassisi, Joule **2**, 1410 (2018).

[18] Y. Liu, T. Zhao, W. Ju, and S. Shi, J. Materiomics **3**, 159 (2017).

[19] J. Schmidt, M. R. Marques, S. Botti, and M. A. Marques, npj Comput. Mater. **5**, 83 (2019).

[20] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nature (London) **559**, 547 (2018).

[21] G. S. Na, S. Jang, Y.-L. Lee, and H. Chang, J. Phys. Chem. A **124**, 10616 (2020).

[22] J. Wei, X. Chu, X.-Y. Sun, K. Xu, H.-X. Deng, J. Chen, Z. Wei, and M. Lei, InfoMat **1**, 338 (2019).

[23] G. R. Schleder, A. C. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, J. Phys.: Mater. **2**, 032001 (2019).

[24] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).

[25] P. C. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964).

[26] K. Kim, L. Ward, J. He, A. Krishna, A. Agrawal, and C. Wolverton, Phys. Rev. Materials **2**, 123801 (2018).

[27] S. P. Ong, Comput. Mater. Sci. **161**, 143 (2019).

[28] C. Kim, G. Pilania, and R. Ramprasad, Chem. Mater. **28**, 1304 (2016).

[29] Z. Li, Q. Xu, Q. Sun, Z. Hou, and W.-J. Yin, Adv. Funct. Mater. **29**, 1807280 (2019).

[30] P. De Luna, J. Wei, Y. Bengio, A. Aspuru-Guzik, and E. Sargent, Nature **552**, 23 (2017).

[31] G. Pilania, A. Mannodi-Kanakkithodi, B. Uberuaga, R. Ramprasad, J. Gubernatis, and T. Lookman, Sci. Rep. **6**, 19375 (2016).

[32] A. C. Rajan, A. Mishra, S. Satsangi, R. Vaish, H. Mizuseki, K.-R. Lee, and A. K. Singh, Chem. Mater. **30**, 4031 (2018).

[33] S. Chibani and F.-X. Coudert, APL Mater. **8**, 080701 (2020).

[34] T. Toyao, K. Suzuki, S. Kikuchi, S. Takakusagi, K.-i. Shimizu, and I. Takigawa, J. Phys. Chem. C **122**, 8315 (2018).

[35] K. Takahashi and I. Miyazato, J. Comput. Chem. **39**, 2405 (2018).

[36] M. De Jong, W. Chen, R. Notestine, K. Persson, G. Ceder, A. Jain, M. Asta, and A. Gamst, Sci. Rep. **6**, 34256 (2016).

[37] J. D. Evans and F.-X. Coudert, Chem. Mater. **29**, 7833 (2017).

[38] B. A. Calfa and J. R. Kitchin, AIChE J. **62**, 2605 (2016).

[39] J. R. Moreno, G. Carleo, and A. Georges, Phys. Rev. Lett. **125**, 076402 (2020).

[40] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, Nat. Commun. **8**, 872 (2017).

[41] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, ACS Central Science **5**, 57 (2019).

[42] A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad, npj Comput. Mater. **5**, 22 (2019).

[43] D. Kamal, A. Chandrasekaran, R. Batra, and R. Ramprasad, Mach. Learn.: Sci. Technol. **1**, 025003 (2020).

[44] M. Bogojeski, F. Brockherde, L. Vogt-Maranto, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, arXiv:1811.06255.

[45] A. Fabrizio, A. Grisafi, B. Meyer, M. Ceriotti, and C. Corminboeuf, Chem. Sci. **10**, 9424 (2019).

[46] L. Zepeda-Núñez, Y. Chen, J. Zhang, W. Jia, L. Zhang, and L. Lin, J. Comput. Phys. **443**, 110523 (2021).

[47] M. Tsubaki and T. Mizoguchi, Phys. Rev. Lett. **125**, 206401 (2020).

[48] J. M. Alred, K. V. Bets, Y. Xie, and B. I. Yakobson, Compos. Sci. Technol. **166**, 3 (2018).

[49] S. Gong, T. Xie, T. Zhu, S. Wang, E. R. Fadel, Y. Li, and J. C. Grossman, Phys. Rev. B **100**, 184103 (2019).

[50] Y. S. Teh, S. Ghosh, and K. Bhattacharya, Mech. Mater. **163**, 104070 (2021).

[51] J. A. Ellis, L. Fiedler, G. A. Popoola, N. A. Modine, J. A. Stephens, A. P. Thompson, A. Cangi, and S. Rajamanickam, Phys. Rev. B **104**, 035120 (2021).

[52] R. Nagai, R. Akashi, S. Sasaki, and S. Tsuneyuki, J. Chem. Phys. **148**, 241737 (2018).

[53] B. Kanungo, P. M. Zimmerman, and V. Gavini, Nat. Commun. **10**, 4497 (2019).

[54] J. Schmidt, C. L. Benavides-Riveros, and M. A. Marques, J. Phys. Chem. Lett. **10**, 6425 (2019).

[55] B. Kanungo, P. M. Zimmerman, and V. Gavini, J. Phys. Chem. Lett. **12**, 12012 (2021).

[56] Z. Dai, L. Liu, and Z. Zhang, Adv. Mater. **31**, 1805417 (2019).

[57] C. Si, Z. Sun, and F. Liu, Nanoscale **8**, 3207 (2016).

[58] D. G. Schlom, L.-Q. Chen, C. J. Fennie, V. Gopalan, D. A. Muller, X. Pan, R. Ramesh, and R. Uecker, MRS Bull. **39**, 118 (2014).

[59] V. M. Pereira and A. H. Castro Neto, Phys. Rev. Lett. **103**, 046801 (2009).

[60] J. Li, Z. Shan, and E. Ma, MRS Bull. **39**, 108 (2014).

[61] J.-W. Jiang, Carbon **81**, 688 (2015).

[62] H. M. Ghassemi, C. H. Lee, Y. K. Yap, and R. S. Yassar, Nanotechnology **23**, 105702 (2012).

[63] 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, J. R. Caram, G. L. Celardo, G. Cuniberti, A. Garcia-Etxarri, A. Dianat, I. Diez-Perez, Y. Guo, R. Gutierrez, C. Herrmann, J. Hihath *et al.*, ACS Nano **16**, 4989 (2022).

[64] Y. Hakobyan, E. B. Tadmor, and R. D. James, Phys. Rev. B **86**, 245435 (2012).

[65] A. S. Banerjee, Density functional methods for objective structures: Theory and simulation schemes, Ph.D. thesis, University of Minnesota, Minneapolis, 2013.

[66] A. S. Banerjee, J. Mech. Phys. Solids **154**, 104515 (2021).

[67] A. S. Banerjee and P. Suryanarayana, J. Mech. Phys. Solids **96**, 605 (2016).

[68] S. Ghosh, A. S. Banerjee, and P. Suryanarayana, Phys. Rev. B **100**, 125143 (2019).

[69] H. M. Yu and A. S. Banerjee, J. Comput. Phys. **456**, 111023 (2022).

[70] S. Agarwal and A. Banerjee, Bull. Am. Phys. Soc. **66** (2021).

[71] S. Agarwal and A. Banerjee (unpublished).

[72] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, 1st ed. (Cambridge University Press, Cambridge, 2004).

[73] U. Yadav, S. Pathrudkar, and S. Ghosh, Phys. Rev. B **103**, 035407 (2021).

[74] A. Damle, A. Levitt, and L. Lin, Multiscale Model. Simul. **17**, 167 (2019).

[75] N. Marzari and D. Vanderbilt, Phys. Rev. B **56**, 12847 (1997).

[76] R. D. James, J. Mech. Phys. Solids **54**, 2354 (2006).

[77] T. Dumitrica and R. D. James, J. Mech. Phys. Solids **55**, 2206 (2007).

[78] P. Koskinen, Appl. Phys. Lett. **99**, 013105 (2011).

[79] P. Koskinen, Phys. Rev. B **82**, 193409 (2010).

[80] T. Dumitrica, in *Trends in Nanophysics*, Engineering Materials, edited by V. Barsan and A. Aldea (Springer, Berlin, 2010), pp. 29–55.

[81] E. B. Barros, A. Jorio, G. G. Samsonidze, R. B. Capaz, A. G. S. Filho, J. M. Filho, G. Dresselhaus, and M. S. Dresselhaus, Phys. Rep. **431**, 261 (2006).

[82] R. McWeeny, *Symmetry: An Introduction to Group Theory and Its Applications*, 1st ed. (Dover, New York, 2002).

[83] M. Hammermesh, *Group Theory and Its Application to Physical Problems*, 1st ed. (Dover, New York, 1989).

[84] L. Kleinman and D. M. Bylander, Phys. Rev. Lett. **48**, 1425 (1982).

[85] A. S. Banerjee, P. Suryanarayana, and J. E. Pask, Chem. Phys. Lett. **647**, 31 (2016).

[86] N. Troullier and J. L. Martins, Phys. Rev. B **43**, 1993 (1991).

[87] E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, and P. Gumbsch, Phys. Rev. Lett. **97**, 170201 (2006).

[88] J. P. Perdew and Y. Wang, Phys. Rev. B **45**, 13244 (1992).

[89] J. R. Chelikowsky, N. Troullier, and Y. Saad, Phys. Rev. Lett. **72**, 1240 (1994).

[90] J. R. Chelikowsky, N. Troullier, K. Wu, and Y. Saad, Phys. Rev. B **50**, 11355 (1994).

[91] X. Jing, N. Troullier, D. Dean, N. Binggeli, J. R. Chelikowsky, K. Wu, and Y. Saad, Phys. Rev. B **50**, 12234 (1994).

[92] 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, Singapore, 2005).

[93] S. Ghosh and P. Suryanarayana, Comput. Phys. Commun. **212**, 189 (2017).

[94] S. Ghosh and P. Suryanarayana, Comput. Phys. Commun. **216**, 109 (2017).

[95] S. Pathrudkar, Deformation manifold learning model for multiwalled carbon nanotubes, Master's thesis, Michigan Technological University, 2021.

[96] H. Robbins, Bull. Am. Math. Soc. **58**, 527 (1952).

[97] S. L. Lohr, *Sampling: Design and Analysis* (CRC Press, Boca Raton, FL, 2019).

[98] K.-T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments* (CRC Press, Boca Raton, FL, 2005).

[99] J. Santiago, M. Claeys-Bruno, and M. Sergent, Chemom. Intell. Lab. Syst. **113**, 26 (2012).

[100] T. W. Simpson, J. Poplinski, P. N. Koch, and J. K. Allen, Eng. Comput. **17**, 129 (2001).

[101] D. K. J. Lin, T. W. Simpson, and W. Chen, Int. J. Reliab. Appl. **2**, 209 (2001).

[102] J. M. Hammersley, Ann. N.Y. Acad. Sci. **86**, 844 (1960).

[103] I. M. Sobol', Zh. Vychisl. Mat. Mat. Fiz. **7**, 784 (1967) [USSR Computational Mathematics and Mathematical Physics **7**, 86 (1967)].

[104] M. D. McKay, R. J. Beckman, and W. J. Conover, Technometrics **42**, 55 (2000).

[105] R. Jin, W. Chen, and A. Sudjianto, in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 37009 (American Society of Mechanical Engineers, New York, 2003), pp. 545–554.

[106] A. B. Owen, Statistica Sinica **2**, 439 (1992).

[107] M. E. Johnson, L. M. Moore, and D. Ylvisaker, J. Stat. Plan. Inference **26**, 131 (1990).

[108] M. Gunzburger and J. Burkardt, Rapp. Tech. Florida State University (cf. p. 73) (2004).

[109] J.-S. Park, J. Stat. Plan. Inference **39**, 95 (1994).

[110] H. Niederreiter, J. Number Theory **30**, 51 (1988).

[111] M. A. Bessa, R. Bostanabad, Z. Liu, A. Hu, D. W. Apley, C. Brinson, W. Chen, and W. K. Liu, Comput. Methods Appl. Mech. Eng. **320**, 633 (2017).

[112] S. Wold, K. Esbensen, and P. Geladi, Chemom. Intell. Lab. Syst. **2**, 37 (1987).

[113] I. T. Jolliffe, Principal Component Anal. **29**, 1 (2002).

[114] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction* (Springer, Berlin, 2007).

[115] I. T. Jolliffe and J. Cadima, Philos. Trans. R. Soc. A **374**, 20150202 (2016).

[116] S.-C. Wang, *Interdisciplinary Computing in Java Programming* (Springer, Berlin, 2003), pp. 81–100.

[117] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (AAAI Press, Portland, Oregon, 1996), pp. 226–231.

[118] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, ACM Trans. Database Syst. **42**, 1 (2017).

[119] Y. Saad and M. H. Schultz, SIAM J. Sci. Stat. Comput. **7**, 856 (1986).

[120] E. Vecharynski, C. Yang, and F. Xue, SIAM J. Sci. Comput. **38**, A500 (2016).

[121] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, revised ed. (SIAM, Philadelphia, 2011).

[122] J. Harris, Phys. Rev. B **31**, 1770 (1985).

[123] W. M. C. Foulkes and R. Haydock, Phys. Rev. B **39**, 12520 (1989).

[124] H. Zou and T. Hastie, J. R. Stat. Soc.: Ser. B **67**, 301 (2005).

[125] D. P. Kingma and J. Ba, arXiv:1412.6980.

[126] J. W. Ding, X. H. Yan, and J. X. Cao, Phys. Rev. B **66**, 073401 (2002).

[127] A. S. Banerjee, L. Lin, P. Suryanarayana, C. Yang, and J. E. Pask, J. Chem. Theory Comput. **14**, 2930 (2018).

[128] W. Hu, L. Lin, and C. Yang, J. Chem. Phys. **143**, 124110 (2015).

[129] F. Qin, W. Shi, T. Ideue, M. Yoshida, A. Zak, R. Tenne, T. Kikitsu, D. Inoue, D. Hashizume, and Y. Iwasa, Nat. Commun. **8**, 14465 (2017).

[130] A. Korde, B. Min, E. Kapaca, O. Knio, I. Nezam, Z. Wang, J. Leisen, X. Yin, X. Zhang, D. S. Sholl *et al.*, Science **375**, 62 (2022).

[131] P. Liashchynskyi and P. Liashchynskyi, arXiv:1912.06059.

[132] X. Glorot, A. Bordes, and Y. Bengio, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, Vol. 15 (PMLR, Fort Lauderdale, FL, 2011), pp. 315–323.

[133] X. Glorot and Y. Bengio, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, edited by Y. W. Teh and M. Titterington, Proceedings of Machine Learning Research (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010), pp. 249–256.

[134] L. Prechelt, Early stopping–but when?, in *Neural Networks: Tricks of the Trade: Second Edition* (Springer Berlin, 2012), pp. 53–67.

[135] W. S. Sarle, in *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics* (1995), pp. 352–360.