

Lean CNNs for mapping electron charge density fields to material properties

*Pranoy Ray^{1,2}, Kamal Choudhury^{3,4}, Surya R. Kalidindi^{*1,2}*

¹ George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, USA

² School of Computational Science and Engineering, Georgia Institute of Technology, USA

³ Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, USA

⁴ Theiss Research, La Jolla, USA

* Correspondence: surya.kalidindi@me.gatech.edu

ABSTRACT

This work introduces a lean CNN (convolutional neural network) framework, with a drastically reduced number of fittable parameters (<81K) compared to the benchmarks in current literature, to capture the underlying low-computational cost (i.e., surrogate) relationships between the electron charge density (ECD) fields and their associated effective properties. These lean CNNs are made possible by adding a pre-processing step (i.e., a feature engineering step) that involves the computation of the ECD fields' spatial correlations (specifically, 2-point spatial correlations). The viability and benefits of the proposed lean CNN framework are demonstrated by establishing robust structure-property relationships involving the prediction of effective material properties using the feature-engineered ECD fields as the only input. The framework is evaluated on a dataset of crystalline cubic systems consisting of 1410 molecular structures spanning 62 different elemental species and 3 space groups.

Keywords: Density Functional Theory, Electron Charge Density, Materials Informatics, Convolutional Neural Networks

1. INTRODUCTION

Density Functional Theory (DFT) [1, 2] is often used to estimate molecular systems' physical and chemical properties and facilitates rapid screening of potential chemical compositions for advanced materials design [3, 4, 5, 6, 7]. A DFT simulation utilizes the raw molecular structure of a material as input and self-consistently computes the electron charge density (ECD) field, which is then used to estimate a multitude of the material's effective properties. However, the high cost of the DFT computations often precludes the direct estimation of the material properties over the large spaces of chemical compositions explored in material design efforts.

Emergent AI/ML (artificial intelligence/machine learning) toolsets [8, 9, 10, 11, 12] make it possible to learn the underlying molecular structure-property relationships [13, 14, 15, 16, 17, 18, 19, 20, 21] on a relatively small set of DFT computations and subsequently employ the extracted surrogate models to identify molecular structures [13, 22] of high potential interest (i.e., expected to exhibit a desired combination of material properties) for further analyses by DFT simulations. Indeed, AI/ML tools are already being used extensively in the establishment of Process-Structure-Property (P-S-P) relationships at multiple material length scales [23, 24, 25]. Specifically, for

mesoscale considerations, it has been shown that the framework of 2-point spatial correlations [26, 27, 28, 29, 30, 31, 32, 33, 34, 35] combined with convolutional neural networks (CNNs [36]) or Gaussian process regression (GPRs [37]) has proven to be extremely valuable in establishing the desired S-P surrogates. Recent work has begun to apply these same strategies to molecular structures and has demonstrated promise [15, 22, 36]. One of the main advantages of utilizing spatial correlations is that it provides a systematic quantification framework for the rich defect populations encountered in the multiscale material internal structure. Specifically, the framework allows for the extraction of a hierarchy of feature vectors quantifying the higher-order spatial patterns in the material structure. Furthermore, it can be shown that the complete set of all theoretically possible material structures occupies a convex region in the feature vector space (i.e., the space of n-point spatial correlations) for a specific length scale. It is noted that many of the other feature engineering approaches [18, 36, 37, 38, 39] explored in current literature for the material structure, such as graph neural networks [14, 16, 40] or expert-selected feature lists [8, 20, 21, 41, 42], do not guarantee that the physically allowed material structures occupy a contiguous region in the latent vector space. Considering convex or contiguous design spaces becomes important when pursuing materials design efforts, where the primary goal is to evaluate a large number of novel theoretically feasible candidate structures that have not yet been confirmed in practice.

Prior work ([46], [47], [48], [49], [50], [51], [52], [53], [54]) has identified the ECD field (denoted as ρ) as the central physics-capturing molecular structure field. One of the main advantages of considering the ECD field as the molecular structure field is that it implicitly embeds elemental species information and exhibits an inherent capability to extrapolate to elemental species that are not present in the training dataset [22, 38]. Past literature has explored the effectiveness of the ECD field as the direct input vector to CNN-based surrogates ([46], [50], [51], [54]). In general, these models needed to employ a large number (up to 10 million) of training parameters, suggesting that these models might represent overfits to relatively small datasets of ~ 2500 samples used in these studies. However, it is expected that effective properties of the molecular structures can be directly related to the spatial correlations extracted from the ECD field. Specifically, it is expected that the formalism of n-point spatial correlations [26] can produce a systematic approach for feature engineering the ECD field. The spatial correlations of a material structure have proven their efficacy for establishing accurate and robust surrogate modeling across multiple material length scales [15, 22, 26, 35, 52, 53, 54], including molecular length scales [22, 38]. Our central hypothesis for this work is that the use of autocorrelations of the ECD field* as the input will result in a significantly smaller CNN (a lean CNN without the need for fully connected layers) model compared to those in current literature that utilize the ECD field directly as the input. Similar lean CNNs using autocorrelations of material structures as input features have already demonstrated robust structure-property linkages at the mesoscale [57].

The primary goal of this paper is to develop and demonstrate the benefits of building lean CNNs to serve as surrogate models connecting the autocorrelations of the ECD field to the effective properties of interest. We demonstrate these benefits on a small dataset of high structural complexity and chemical diversity. Specifically, our dataset includes 1410 compositions covering

* Note that the computational cost of getting the ECD field from DFT computations can be significantly lower than the cost of estimating the effective properties from the ECD field using the DFT computations. Furthermore, a single ECD field is capable of informing us about most of the effective properties of interest.

3 space groups, across 62 unique chemical species. The properties targeted in this work include (a) Bulk Modulus and (b) Total Energy of the material system.

2. DATA COLLECTION AND PRE-PROCESSING

2.1. DATASET

A dataset consisting of ECD fields, bulk moduli, and total energies of approximately 1410 cubic crystalline compounds is compiled from the JARVIS-DFT [58] repository. These properties were calculated using density functional theory (DFT) with the Vienna Ab initio Simulation Package (VASP) ([59], [60], [61]), employing a plane-wave basis set and the projector augmented wave (PAW) method with the OptB88vdW exchange-correlation functional. VASP-recommended pseudopotentials were utilized for each element. The compounds included in this dataset comprised 15 unary, 494 binary, 947 ternary, 93 quaternary, and 1 quinary compound spanning 62 unique chemical species. Figure 1(a) shows the distribution of the compounds in this dataset across three space groups of which $Fm\bar{3}m$ and $F43m$ are FCC (face-centered cubic), and $Pm\bar{3}m$ is SC (simple cubic).

In this study, we selected two very different types of target properties to evaluate critically the versatility of our feature engineering approach (i.e., using autocorrelations of the ECD field as input to our proposed lean CNNs). One of the targets was selected as the bulk modulus (represents an example of a structural property) and the other is total energy (represents a functional property). These properties are physically related to each other by the Birch-Murnaghan equation of state ([62], [63]). Figure 1(b) and (c) represent the distribution of DFT-computed bulk moduli and total energies of the compounds present in the dataset. Note that the samples exhibit an uneven distribution in both the property targets.

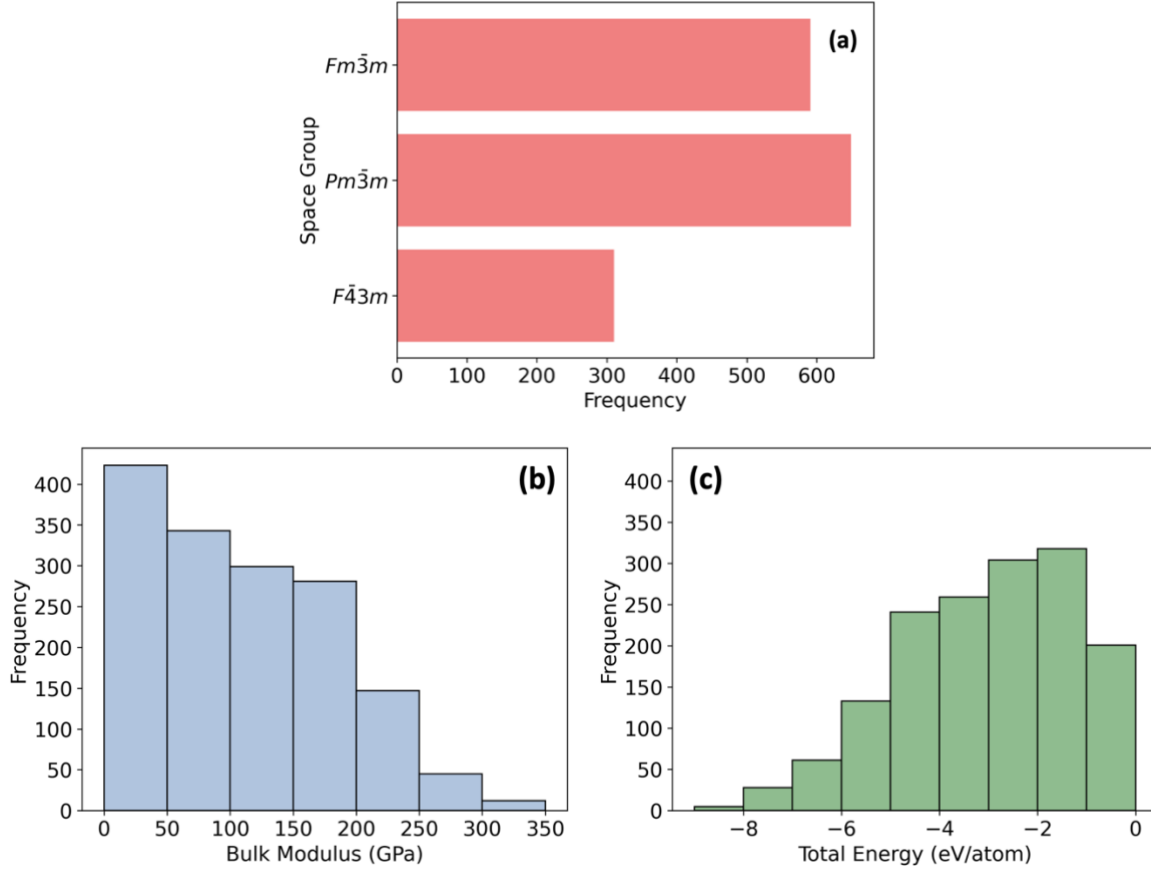


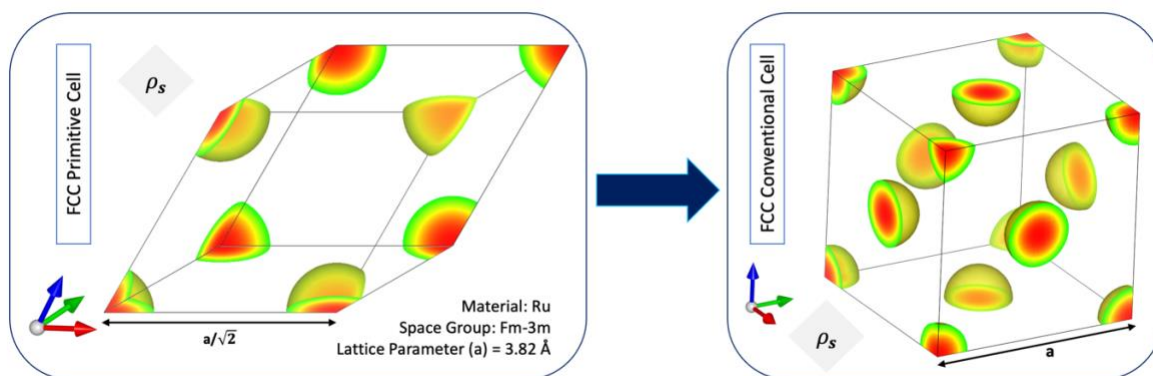
Figure 1: (a) Distribution of the three cubic space groups for the different compounds present in the dataset. (b) Histogram of the bulk moduli of the compounds in the dataset. (c) Histogram of the total energies of the compounds in the dataset.

2.2. DATA PRE-PROCESSING

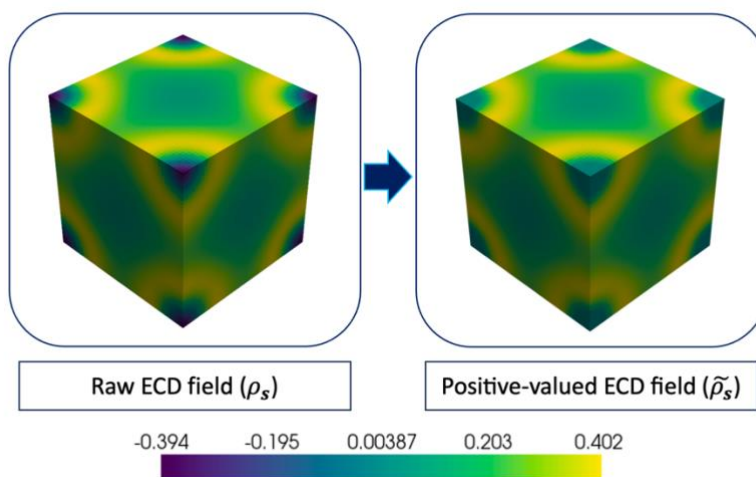
2.2.1. PRIMITIVE TO CONVENTIONAL CELL TRANSFORMATION

For reducing the computational cost of DFT simulations, unit cells are frequently depicted using their primitive basis to reduce the number of atoms present while still taking advantage of periodicity. Specifically, the FCC primitive cells in our dataset were primarily represented using a periodic non-orthogonal basis. Consequently, it was necessary to convert the material structure representation from the primitive basis into the conventional (orthogonal & periodic) basis. This conversion was performed using trilinear interpolation, employing the SimpleITK package [64] in Python. The accuracy of this conversion was evaluated by ensuring negligible changes in the peak and minimum values of the ECD and the total number of valence electrons before and after transformation for all compounds included in our study. The SC material systems did not need this conversion as their SC primitive cell was already expressed on an orthogonal basis. The resampled conventional unit cell representation of the ECD (i.e., ECD values sampled on a uniform orthogonal grid) is used as the default ECD from hereon and is denoted by the array $\rho_{\mathbf{s}}$, where the subscript \mathbf{s} indexes the location for every voxel in the ECD. Figure 2(a) provides an example of

the primitive to conventional ECD field transformation performed on all the primitive FCC ECD fields utilized in this work.



(a) Primitive to Conventional Cell transformation for an example FCC compound.



(b) Re-normalization of the ECD Field for an example SC compound.

Figure 2: Data pre-processing steps performed on all ECD fields in the dataset.

2.2.2. HANDLING NEGATIVE VALUES

In certain ECD fields, we observed regions of negative electronic charge density, especially when using projected augmented wave (PAW) pseudopotentials [65]. This phenomenon occurs because these pseudopotentials take the "soft" charge density—derived from Kohn-Sham wavefunctions—and apply an additional "correction" known as "augmentation" to the charge density. Although this "augmentation" is often described in the literature ([66],[67]) as an addition of charge, it can also result in a decrease in charge density in specific regions. If the soft charge density in a given area is lower than the amount of charge removed by the augmentation, negative charge density values

arise. Causes of this effect can be attributed to various factors, but one common cause is Fourier aliasing, which occurs when the augmentation grid is insufficiently refined to accurately represent the augmentation charge density. The ECD fields (ρ_s) utilized in this study were sourced from a curated repository where each DFT simulation was run with a different augmentation charge grid. Since negative values for ECD are non-physical, we removed them and re-normalized the ECD values using the protocols described next.

The DFT-computed charge density array, denoted by ρ_s introduced earlier, exhibits the following property:

$$\text{Total Number of valence electrons (in the simulation cell } S) = \sum_{s \in S} \rho_s V_s \quad (1)$$

where V_s denotes the volume of the voxel indexed by s (constant in our case for all voxels in a single DFT computation result). Our goal is to re-normalize the ECD field, i.e., transform ρ_s into positive-valued $\tilde{\rho}_s$, while keeping the same total number of valence electrons in the simulation cell S (i.e., $\sum_{s \in S} \rho_s V_s = \sum_{s \in S} \tilde{\rho}_s V_s$).

We achieve this by first identifying connected regions of negative ECD values in the simulation cell S , referred henceforth as clusters (C). These clusters typically correspond to individual atoms in the molecule. For each cluster C_i , we calculate the sum of the negative ECD values as $\rho_{C_i}^-$. Next, we identify a boundary layer of ECD voxels (B_j) immediately outside C_i (the voxels exhibiting positive ECD values). The sum of the positive ECD values in B_j is denoted as $\rho_{B_j}^+$. Where $\rho_{B_j}^+ \leq |\rho_{C_i}^-|$, we zero-out the ECD values in B_j , and update $\rho_{C_i}^-$ by adding $\rho_{B_j}^+$ (i.e. $\rho_{C_i}^- = \rho_{C_i}^- + \rho_{B_j}^+$). We repeat this process until we encounter the situation $\rho_{B_j}^+ > |\rho_{C_i}^-|$. For this final layer of correction, we redistribute the remnant positive ECD values (i.e., $\rho_{B_j}^+ + \rho_{C_i}^-$) evenly across the voxels in B_j and set the negative ECD values in C_i to 0. This process is repeated for all clusters (C) until there are no negative ECD values remaining in the simulation cell (see Figure 2(b)).

3. FEATURE ENGINEERING & MODEL-BUILDING

3.1. COMPUTATION OF THE SPATIAL CORRELATIONS

The general mathematical framework for statistical quantification of a stochastic material structure $m(\mathbf{x}, h)$ has already been established in prior work [29, 30, 33, 34, 53, 55, 56]. More specifically, the recently developed VAST framework [22, 38, 59] represents the molecular structure by its ECD field (i.e., $m(\mathbf{x}, h) = \tilde{\rho}(\mathbf{x})$). Next, we featurize the pre-processed ECD field ($\tilde{\rho}_s$) to capture the underlying translationally-invariant spatial patterns using the framework of 2-point spatial correlations, which will then serve as an input to our lean CNN model. Towards this goal, we

compute the discrete autocorrelations of the discrete ECD field (specifically, its square root¹) with Fast Fourier Transform (FFT) algorithms as

$$f_{\mathbf{r}} = \frac{1}{|\mathcal{S}|} \mathcal{F}^{-1} \left[\mathcal{F} \left(\sqrt{\tilde{\rho}_{\mathcal{S}}} \right) * \mathcal{F} \left(\sqrt{\tilde{\rho}_{\mathcal{S}}} \right) \right] \quad (2)$$

where $|\mathcal{S}|$ is the total number of voxels in the simulation cell, \mathcal{F} represents the Discrete Fourier Transform operation, \mathcal{F}^{-1} represents the inverse Discrete Fourier Transform, and $*$ denotes the complex conjugate [26, 30]. The components of the autocorrelations vector defined by Eq. (2) form a high-dimensional feature vector, which can be used as an input to our proposed lean CNN models.

The discretized ECD fields utilized in this study were sourced from a curated repository of DFT simulation results, which employed different voxel lengths in the different simulations to accommodate the differences in the lattice parameters of the compounds studied. Our goal in the present study is to establish a CNN model that generalizes across diverse material classes (including different symmetry groups). As a result, we needed to standardize the input to the CNN model using a fixed voxel length for all compounds in our study. Note also that the voxel length can be standardized in the representation of the raw ECD field itself, i.e., before the computation of the 2-point spatial correlations as explored in prior work ([15], [22], [38]). In the present study, we opted to compute the 2-point correlations using the voxel lengths naturally implied in the ECD fields downloaded from the curated repositories, and subsequently express the required inputs to the CNN model (i.e., autocorrelations of the ECD field) using a standardized voxel length (i.e., kept the same for all compounds included in this study). This approach allows us to take full advantage of the periodicity of the ECD fields in the computation of the autocorrelations (using FFT algorithms). The interpolation of the computed periodic autocorrelations to the standardized uniform grid was accomplished using established trigonometric interpolation schemes [55, 56]. The standardized autocorrelations are denoted as $F_{\mathbf{r}}$. Since the autocorrelations are typically much smoother compared to the original fields, our strategy to interpolate the autocorrelations as opposed to the raw ECD fields is expected to reduce significantly the interpolation errors involved.

In the present study, we adopted a standardized voxel width (λ) of 0.2 Å per voxel, and a uniform spatial grid of size $51 \times 51 \times 51$ for expressing the autocorrelations as the input to our lean CNN models. As a result, the spatial correlations included in this study covered vectors with components in the range of -5 Å to +5 Å. Although the compounds included in our study exhibited lattice parameters in the range from 2 Å – 12 Å, a large fraction of them had lattice parameters smaller than 5 Å. The decision to truncate the autocorrelations at ± 5 Å was needed to limit the computational cost. The mathematical details of the trigonometric interpolation used in this work have been already presented in our prior work ([22]), Figure 3 provides an example of the data transformations performed on all the ECD fields downloaded and utilized in this work. This specific example corresponds to a SC unit cell of Fe₃SnC. For this compound, the ECD field from the public repository was available as an $80 \times 80 \times 80$ array covering a cubic volume with a side length of 3.83 Å. In the first step, the autocorrelations of the ECD field were computed using FFT algorithms ([27], [30], [34]), and expressed as an $80 \times 80 \times 80$ array covering vectors \mathbf{r} whose

¹Equation (1) has been defined with the square root of the ECD field $\tilde{\rho}_{\mathcal{S}}$ so that the average charge density of the system is essentially signified by f_0

components lie within $\pm 1.915 \text{ \AA}$. In the second step, taking advantage of the periodicity of the autocorrelations and the trigonometric interpolation schemes, the autocorrelations are re-sampled as an array of size $51 \times 51 \times 51$ on a uniform grid covering vector components within $\pm 5 \text{ \AA}$. The value at the center of this array (corresponds to $\mathbf{r} = 0$) reflects the average charge density for Fe_3SnC [26].

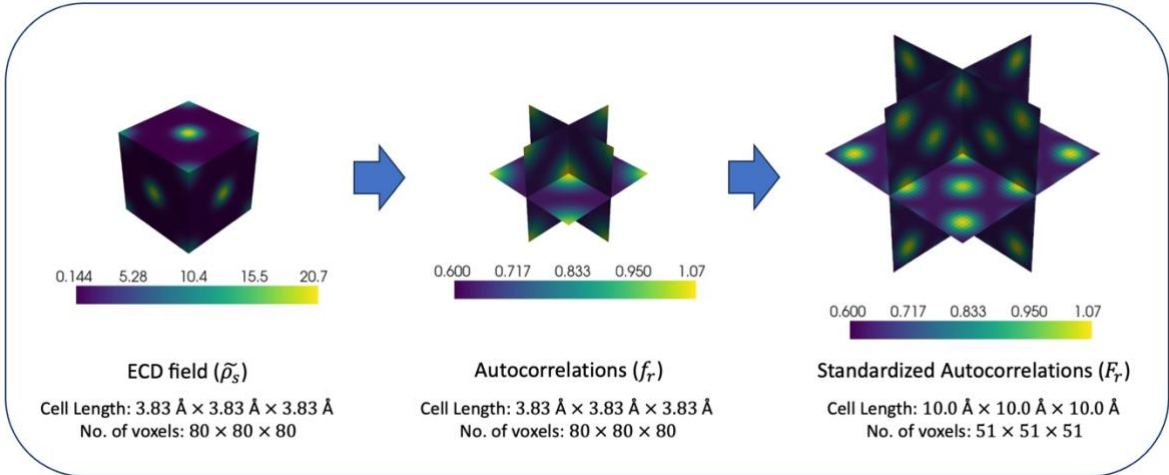


Figure 3: Illustration of the feature engineering steps performed on the ECD field of an example SC compound (Fe_3SnC). All the color bars represent $\text{eV}/\text{\AA}^3$ values.

3.2. CONVOLUTIONAL NEURAL NETWORKS

Neural Networks (NNs) ([73], [74]) are powerful tools for learning complex nonlinear mappings between any selected set of inputs (i.e., features) and outputs (i.e., targets) encountered in solving engineering problems. A specific type, Convolutional Neural Networks (CNNs) ([36], [75], [76]), is ideally suited for problems where the inputs are images (i.e., spatial data). CNNs are typically designed by sequencing convolutional, pooling, and fully connected layers, which perform a variety of data transformations connecting the inputs to the outputs. The convolutional and fully connected layers introduce fittable parameters, which are calibrated to the training dataset. Some CNN architectures also employ strides with convolutions to achieve computationally efficient mappings between high-dimensional inputs and low-dimensional outputs (as is the case for the present application). Strides help coarsen inputs by reducing feature map dimensions during a convolution. The many options available in building CNNs result in several hyperparameters that affect model performance (usually quantified by higher accuracy or higher generalizability across the input domain). Examples of such hyperparameters include the number of layers, filter/kernels sizes, number of filters/kernels in each layer, choice of non-linear activation functions, and choice of loss functions. Tuning these hyperparameters is essential for producing a reliable CNN model. For example, while increasing a CNN's number of layers and the number of filters in each layer can improve learning, it also raises the risk of overfitting and increases computational costs. Therefore, it is important to attain the right balance in designing the CNN architecture for a given problem. CNNs have already been successfully applied to various materials science problems at the mesoscale ([35], [55], [57], [68], [77], [78], [79], [80], [81]) and the molecular scale ([32], [51], [70]). However, typical CNN applications often include fully connected (FC) layers, which dramatically increase the number of fittable parameters in the learned model. In this study, we

address a homogenization problem that deals with 3D representations of the autocorrelations of the ECD field as the input. Since spatial variations in the input hold the keys to connecting to the target properties, it would make most physical sense to rely heavily on convolution layers and a final global average pooling (GAP) to predict the target properties of interest. Indeed, this simple strategy was successfully employed at the mesoscale in prior work [57]. Such simple CNN architectures comprising exclusively of only convolution layers and a final GAP layer produce very lean machine learning models (often less than 100K fittable parameters); these are referred as lean CNNs in this work. This reduction in model complexity is mainly attributed to using standardized autocorrelations (F_r) of the ECD field as input to the CNN models.

Because the mapping involved goes from a high-dimensional input (i.e., autocorrelations of ECD) to scalar outputs (i.e., bulk properties of interest), we have mainly focused on inverted architectures, where we gradually reduce the number of kernels in each subsequent layer. In study, we have explored a large number of possible architectures before settling on the best architecture for the present application. Models A, B & C (see Table 1) exemplify three of the architectures explored in this work. Models A & B represents single output CNNs that predict either the bulk modulus, or the total energy, as the individual scalar target. Model C represents a multi-output CNN that predicts both the bulk modulus and total energy simultaneously. One of our goals was to examine the relative performances of two independent single output CNNs versus a single multi-output CNN for our selected problem. This is interesting because the salient feature sets influencing the different outputs (i.e., bulk modulus and total energy) are expected to be dramatically different from each other. The comparison of the different CNN models produced in this work allowed us to examine critically the inherent feature-engineering capabilities of CNNs for diverse outputs in a single architecture. Furthermore, the CNN architectures examined in this work are deliberately simple, with a maximum of seven layers, and the best-performing model having six. Hence, we performed a manual search of architectures, given the limited complexity and the reduced computational cost compared to automated architecture searches. This approach allowed for effective intuition-driven optimization without the need for exhaustive automation, which we deemed unnecessary for a six-layer CNN with relatively fewer fittable parameters (see Table 1).

Table 1 describes the composition of the CNN architectures. Each layer is described using the notation $p@b/c$, which indicates a $p \times p \times p$ (3-D) kernel, applied with a stride of c , and b channels. The default value of stride, when not mentioned, is one. For example, the notation $64@5/2$ refers to 64 filters of size of $5 \times 5 \times 5$, and a stride of 2. It should be noted that all the models we produced in this work utilize a significantly lesser number of fittable parameters (see the comparison with current benchmarks from literature in Table 1). Most interestingly, it will be seen that our lean Model A (see Figure 4), with the smallest number of fittable parameters, outperforms significantly some of the heavier models explored in this study.

For all the CNN models produced in this work, the target properties were normalized prior to model training and subsequently returned to their original scale after prediction. This was found to improve the convergence rate and ensured that both outputs received equal consideration in the loss function despite the large differences in their numerical values. The CNN models were produced using PyTorch [82]. PReLU (Parametric Rectified Linear Unit) [83] was adopted as the non-linear activation in this work, and is defined as

$$PReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha x, & \text{otherwise} \end{cases} \quad (3)$$

where α is a fittable parameter associated with each filter (in addition to the weights and biases).

Model	Zhao et al. [46]	Mirzaee et al. [54]	A	B	C
Input	ρ_s	ρ_s	F_r	F_r	F_r
L1	5@32	3@16	5@64/2	5@128/2	5@64/2
L2	MaxPool	MaxPool	3@32	3@64	3@32
L3	2@64	3@32	3@16	3@32	3@16
L4	MaxPool	MaxPool	3@8	3@16	3@8
L5	3@64	3@64	1@1	3@8	1@2
L6	MaxPool	MaxPool	AvgPool	1@1	AvgPool
L7	FC-1000	FC-11200	-	AvgPool	-
L8	FC-128	FC-2048	-	-	-
L9	FC-64	FC-1024	-	-	-
L10	FC-32	FC-8	-	-	-
Outputs	1	8	1	1	2
Fittable Parameters	2,360,736	2,129,672	80,825	310,201	80,834

Table 1: Examples of different CNN architectures explored in this work along with selected benchmarks from current literature.

Optimizing CNN performance required systematic hyperparameter tuning through multiple iterations, evaluating each configuration's impact on model fidelity for the specific task at hand. Beyond the hyper-parameters described earlier for the CNN architecture, we also tuned the initial learning rate and the number of epochs. The model's weights were optimized using stochastic gradient descent with the Adam Optimizer [84], and the learning rate was optimized using the Cosine Annealing Learning Rate Scheduler (with Warm Restarts) [85]. The initial learning rate was set to be 5×10^{-3} , and the training batch size was set to be 50. The Mean Squared Error (MSE) Loss function was used to train the models implemented in this work. Further, all CNN models were trained for several epochs until the respective loss functions converged to their minimum values. Through multiple trials, we established the number of training epochs at 650. By fixing the number of epochs, we enable a meaningful comparison of the performances of the various CNN architectures examined in this study. Dropout layers were included as a regularization technique,

along with cross-validation measures ([86], [87]) to prevent overfitting and ensure model robustness.

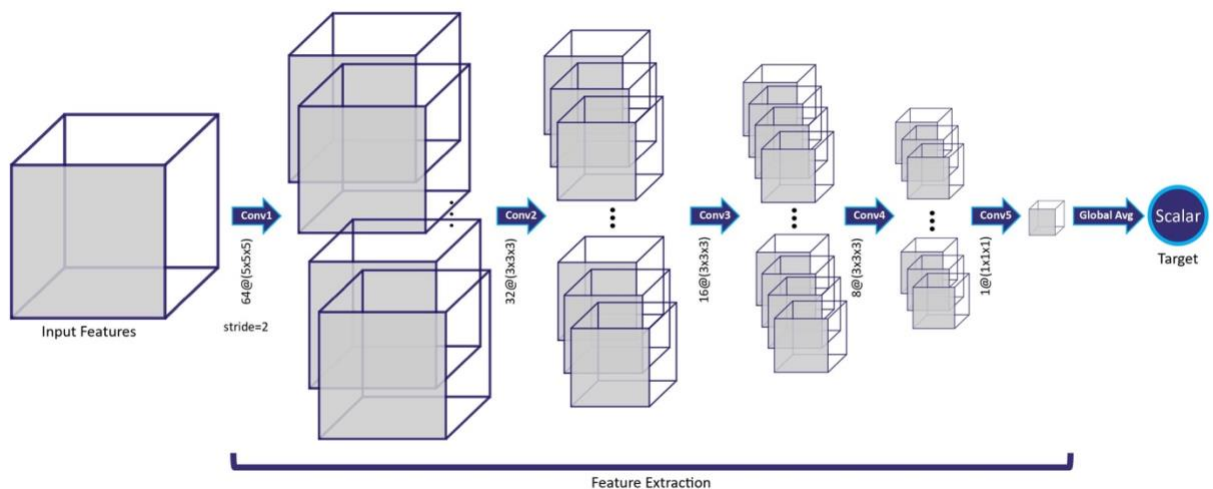


Figure 4: The CNN architecture (for Model A) implemented in this work.

4. RESULTS & DISCUSSION

The available limited dataset of 1410 datapoints exhibited high structural and chemical diversity, which necessitated careful consideration of data allocation into test, train, and cross-validation groups. To ensure the model captures this diversity and develops robust predictive capabilities, a large portion of the data was employed in training the models. In this work, we have utilized 90% of the data in training and cross-validation, and used only 10% for testing. This approach maximizes the use of available data for model development and ensures reliable performance assessment (through cross-validation).

The model building effort started with tuning of the CNN architecture using a 9-fold cross-validation (CV), where the 10% of the dataset was used in each fold for cross-validation. The CV errors were used as the main metric for the robustness of the trained model. In other words, model architectures producing lower CV errors were selected for further evaluation, and those exhibiting high CV errors were abandoned as over-fit models. Once the architecture was finalized, we used the full 90% training dataset for establishing the final version of the model. Model training was performed on a single NVIDIA A100 GPU with 32 GB of memory, and the training times averaged around 6-7 minutes for models A & C, and 11-12 minutes for Model B.

Table 2 summarizes the Mean Absolute Error (MAE) of the train and test sets for the lean CNN models produced in this work. For both bulk modulus and total energy predictions, Model A emerged as the best-performing model amongst all the CNN architectures investigated in this study (this includes the numerous other CNNs we produced beyond those reported in Table 2). The accuracy and robustness of model A for bulk modulus and total energy predictions can also be visualized in the parity plots shown in Figure 5. In this work, model performance was evaluated both in terms of accuracy (i.e., lowest MAE on the test set) and robustness (i.e., lowest difference between MAE values on the test and train sets). It can also be seen that Model B exhibits higher CV and test errors compared to Model A, even though it had almost four times as many fittable parameters. Clearly, Model B represents an overfit model.

Model	A		B		C	
Properties	Bulk Modulus	Total Energy	Bulk Modulus	Total Energy	Bulk Modulus	Total Energy
Units	GPa	eV/atom	GPa	eV/atom	GPa	eV/atom
CV-MAE	9.94	0.82	14.42	0.87	12.44	0.86
Train-MAE	7.49	0.73	14.57	0.81	10.74	0.74
Test-MAE	7.73	0.78	15.33	0.84	12.95	0.95

Table 2: Summary of the performance metrics for the lean CNN models trained in this work.

In order that we can compare the relative performance of the models for the two different targets selected for this study, we employed a Normalized MAE (NMAE) percentage defined as

$$NMAE = \frac{1}{N} \sum_{i=1}^N \left| \frac{S_i - \hat{S}_i}{S_{average}} \right| \times 100$$

where S_i represents the ground-truth value for the target, \hat{S}_i represents the corresponding model prediction, and $S_{average}$ denotes the ensemble average of the ground-truth values from a set of N observations. Model A achieved a test NMAE of $\sim 7\%$ for bulk modulus predictions and a test NMAE of $\sim 28\%$ for total energy predictions. We observed that all the models produced in this work struggled to make accurate predictions for total energy. This is most likely because the total energy needs higher-order spatial correlations as inputs. Recall that we only utilized the 2-point spatial correlations (i.e., auto-correlations) as input to our CNN model. It is generally known that higher-order spatial correlations contain the critical predictors needed for the more complex mappings to the effective properties. In principle, CNNs can incorporate these higher-order correlations as additional channels in the input layer. It is clear from our investigation that CNNs are unable to automatically feature-engineer these higher-order spatial correlations from the auto-correlations we utilized as inputs to our models. This is particularly clear from the fact that a deeper CNN model, such as Model B, actually performs worse than the simpler Model A (see Table 2).

A comparison of the performance of Model C against the performance of Model A revealed that the multi-output strategy was ineffective for the targets selected in this study. We also observed that the performance of Model C did not improve even when we made the model bigger by adding more layers and channels. This implies that the multi-output CNNs were ineffective in learning the superset of features needed to predict both targets accurately. Indeed, the single output CNNs were much more effective in learning the features separately for each target. The results of this study indicate clearly that a multi-output strategy is not likely to be advantageous in situations where the selected targets are expected to exhibit very different types of mappings with the inputs (i.e., S-P mappings). On the other hand, the multi-output strategy might still be beneficial for situations where the S-P mappings of interest have significant overlap. Furthermore, to elucidate on the expressivity of spatial features, we trained a couple of random forest models on the same

dataset with the “matminer” features (based on chemical-formula-based descriptors [20]), for single-property predictions. These models obtained test-NMAEs of 22.3% for bulk modulus predictions, and 31.9% for total energy predictions, which are significantly lower in accuracy than the results yielded by this work.

The computational savings achieved by the single-output lean CNN models (i.e., Model A) presented in this study is significant. Most impressively, Model A used <81K parameters to train, which represents a model with ~25-30 times lower number of trainable parameters used in similar models reported in recent literature ([46], [54]). Table 4 summarizes the main characteristics of Model A against these recent benchmarks. Despite being a leaner model, Model A significantly outperforms the benchmarks. This is impressive, keeping in mind that the benchmarks utilized larger datasets and covered only one crystal structure. Figure 6 provides histograms for the NMAE across the test set, for both the targets predicted by Model A. Most importantly, we note that ~55% of the test NMAE lie below 10% for total energy predictions, while for bulk modulus predictions ~70% the test NMAE lies under 10%. It is observed that ~5% of the bulk modulus predictions in the test set exhibits NMAE above 30%, while for total energy predictions ~12% of the testing set has NMAE above 30%. These high errors were caused by the fact that some of the chemical species encountered in the test set did not have adequate representation in the train set. Our analyses indicates that if a larger training dataset was available, these high test errors can be reduced significantly. We also observed no correlation between the high test errors and compounds with larger lattice parameters. This justifies our decision to limit the auto-correlations in a vector space with components ranging from -5 Å to +5 Å. This truncated set of auto-correlations appears to capture adequately the salient features needed for building the S-P models of interest in this work.

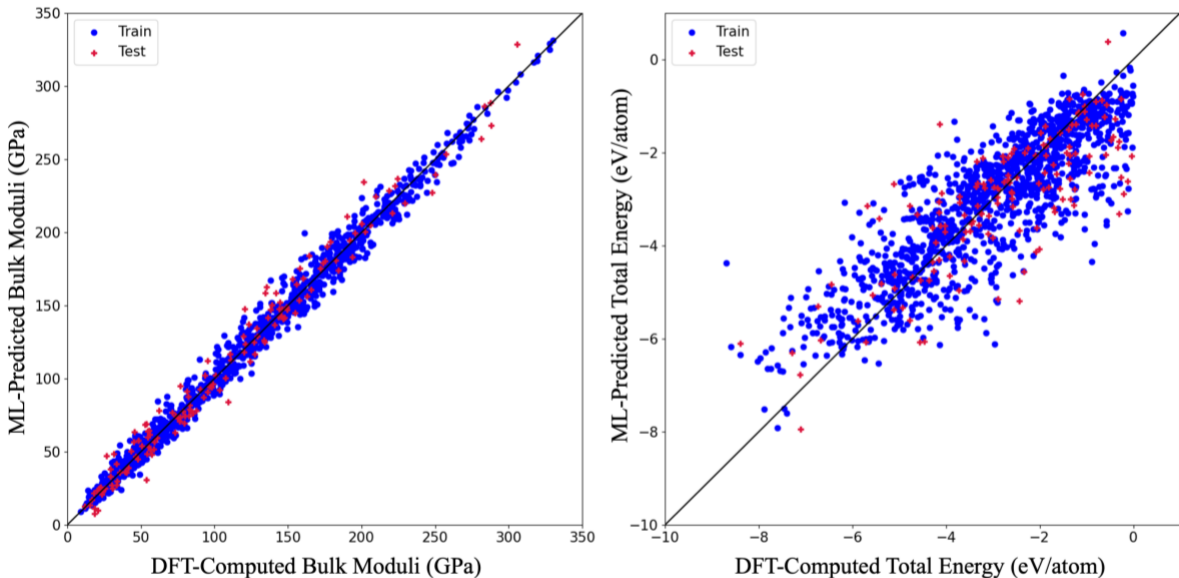


Figure 5: Parity plot of the ML-predicted (model A) v/s DFT-computed (a) Bulk Modulus and (b) Total Energy. The test points (red) are superimposed on the train points (blue).

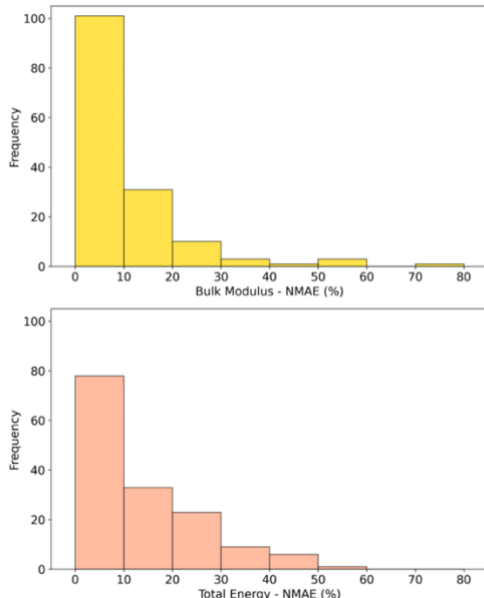


Figure 6: Normalized Mean Absolute Errors (NMAE) of both the target properties predicted by the best lean CNN model in this work (Model A).

Model	Lean CNN (this work)	3D-CNN (Zhao et al. [46])	3D-CNN (Mirzaee et al. [54])
Input	F_r	ρ_s	ρ_s
Total Dataset size	1410	2170	2720
Structural Complexities	FCC & SC	FCC	FCC
Trainable Parameters	81K	2.36M	2.13M
RMSE (GPa)	10.11	26.82	6.158

Table 3: Comparison of the lean CNN model trained in this work v/s benchmark model from literature ([46], [54]), for bulk modulus prediction.

5. CONCLUSIONS

This paper introduces a lean CNN architecture that significantly enhances the accurate and robust prediction of material properties (bulk modulus and total energy) from ECD fields. Our findings challenge the conventional wisdom that CNNs are primarily model-building techniques to circumvent explicit feature engineering. Instead, we demonstrate that the predictive capabilities of these networks can be substantially improved through physics-informed featurization, particularly with the use of spatial correlations. The lean CNNs described in this work are particularly useful in situations where there is limited data. The lean CNN's exceptional performance, achieving an NMAE of $\sim 7\%$ for bulk modulus predictions across 62 different elemental species, positions it as

a new benchmark for future model-building efforts. This unprecedented accuracy, achieved using ECD fields as the sole structural descriptors of atomic space, opens new avenues for materials discovery and optimization. While the featurization framework introduced here has been demonstrated on cubic systems, it generalizes to arbitrary symmetries as it imposes no symmetry constraints, provided sufficient training data for diverse structures is available. Furthermore, the distribution of errors in predictions provides objective guidance on where additional training data should be included, as we observe that the target ranges are significant even with the data imbalance.

6. DECLARATION OF COMPETING INTEREST

The authors declare that no known competing financial interests have influenced the work reported in this paper.

7. ACKNOWLEDGEMENTS

PR and SK acknowledge support from NSF DMREF Award 2119640. The authors acknowledge Prathik Kaundinya and Conlain Kelly for their support and encouragement. This work used the Hive cluster, which is supported by the National Science Foundation under grant number 1828187. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA. K.C. thanks computational resources from the National Institute of Standards and Technology (NIST). This work was performed with funding from the CHIPS Metrology Program, part of CHIPS for America, National Institute of Standards and Technology, U.S. Department of Commerce. Certain commercial equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identifications are not intended to imply recommendation or endorsement by NIST, nor it is intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

8. DATA AVAILABILITY

The dataset used for this study will be made available upon reasonable request, after internal review.

9. REFERENCES

- [1] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas," *Phys. Rev.*, vol. 136, no. 3B, pp. B864–B871, Nov. 1964, doi: 10.1103/PhysRev.136.B864.
- [2] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects," *Phys. Rev.*, vol. 140, no. 4A, pp. A1133–A1138, Nov. 1965, doi: 10.1103/PhysRev.140.A1133.
- [3] J. Deb, A. Kundu, N. Garg, U. Sarkar, and B. Chakraborty, "Copper decorated graphyne as a promising nanocarrier for cisplatin anti-cancer drug: A DFT study," *Appl. Surf. Sci.*, vol. 622, p. 156885, 2023.

- [4] B. Chakraborty, P. Ray, N. Garg, and S. Banerjee, “High capacity reversible hydrogen storage in titanium doped 2D carbon allotrope Ψ -graphene: Density Functional Theory investigations,” *Int. J. Hydrog. Energy*, vol. 46, no. 5, pp. 4154–4167, Jan. 2021, doi: 10.1016/j.ijhydene.2020.10.161.
- [5] H. T. Nair, A. Kundu, P. Ray, P. K. Jha, and B. Chakraborty, “Ti-decorated C₃₀ as a High-capacity Hydrogen Storage Material: Insights from Density Functional Theory,” *Sustain. Energy Fuels*, Sep. 2023, doi: 10.1039/D3SE00845B.
- [6] A. Kundu, A. Jaiswal, P. Ray, S. Sahu, and B. Chakraborty, “Zr doped C₂₄ fullerene as efficient hydrogen storage material: insights from DFT simulations,” *J. Phys. Appl. Phys.*, vol. 57, no. 49, p. 495502, Dec. 2024, doi: 10.1088/1361-6463/ad75a1.
- [7] L. Chen, T. D. Huan, A. Huzayyin, Y. C. Quintero, and R. Ramprasad, “First-principles study of aluminum-polyethylene interfaces,” in *2014 IEEE Conference on Electrical Insulation and Dielectric Phenomena (CEIDP)*, Oct. 2014, pp. 887–890. doi: 10.1109/CEIDP.2014.6995833.
- [8] R. Pederson, J. Chen, S. R. White, and K. Burke, “Conditional probability density functional theory,” *Phys. Rev. B*, vol. 105, no. 24, p. 245138, Jun. 2022, doi: 10.1103/PhysRevB.105.245138.
- [9] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, “Machine learning in materials informatics: recent applications and prospects,” *Npj Comput. Mater.*, vol. 3, no. 1, Art. no. 1, Dec. 2017, doi: 10.1038/s41524-017-0056-5.
- [10] S. Kalidindi, S. Niezgodna, I. Giacomo L, and T. Fast, “A Novel Framework for Building Materials Knowledge Systems,” *Comput. Mater. Contin.*, vol. 17, no. 2, pp. 103–126, 1970, doi: 10.3970/cm.2010.017.103.
- [11] D. L. McDowell and S. R. Kalidindi, “The materials innovation ecosystem: A key enabler for the Materials Genome Initiative,” *MRS Bull.*, vol. 41, no. 4, pp. 326–337, Apr. 2016, doi: 10.1557/mrs.2016.61.
- [12] S. R. Kalidindi, A. J. Medford, and D. L. McDowell, “Vision for Data and Informatics in the Future Materials Innovation Ecosystem,” *JOM*, vol. 68, no. 8, pp. 2126–2137, Aug. 2016, doi: 10.1007/s11837-016-2036-5.
- [13] R. Gurnani *et al.*, “polyG2G: A Novel Machine Learning Algorithm Applied to the Generative Design of Polymer Dielectrics,” *Chem. Mater.*, vol. 33, no. 17, pp. 7008–7016, Sep. 2021, doi: 10.1021/acs.chemmater.1c02061.
- [14] K. Choudhary and B. DeCost, “Atomistic Line Graph Neural Network for improved materials property predictions,” *Npj Comput. Mater.*, vol. 7, no. 1, Art. no. 1, Nov. 2021, doi: 10.1038/s41524-021-00650-1.
- [15] P. R. Kaundinya, K. Choudhary, and S. R. Kalidindi, “Machine learning approaches for feature engineering of the crystal structure: Application to the prediction of the formation energy of cubic compounds,” *Phys. Rev. Mater.*, vol. 5, no. 6, p. 063802, Jun. 2021, doi: 10.1103/PhysRevMaterials.5.063802.
- [16] T. Xie and J. C. Grossman, “Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties,” *Phys. Rev. Lett.*, vol. 120, no. 14, p. 145301, Apr. 2018, doi: 10.1103/PhysRevLett.120.145301.
- [17] G. Montavon *et al.*, “Machine learning of molecular electronic properties in chemical compound space,” *New J. Phys.*, vol. 15, no. 9, p. 095003, 2013.
- [18] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet – A deep learning architecture for molecules and materials,” *J. Chem. Phys.*, vol. 148, no. 24, p. 241722, Mar. 2018, doi: 10.1063/1.5019779.

- [19] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K.-R. Müller, “Bypassing the Kohn-Sham equations with machine learning,” *Nat. Commun.*, vol. 8, no. 1, Art. no. 1, Oct. 2017, doi: 10.1038/s41467-017-00839-3.
- [20] L. Ward *et al.*, “Matminer: An open source toolkit for materials data mining,” *Comput. Mater. Sci.*, vol. 152, pp. 60–69, Sep. 2018, doi: 10.1016/j.commatsci.2018.05.018.
- [21] L. Ward, A. Agrawal, A. Choudhary, and C. Wolverton, “A general-purpose machine learning framework for predicting properties of inorganic materials,” *Npj Comput. Mater.*, vol. 2, no. 1, Art. no. 1, Aug. 2016, doi: 10.1038/npjcompumats.2016.28.
- [22] M. C. Barry, J. R. Gissinger, M. Chandross, K. E. Wise, S. R. Kalidindi, and S. Kumar, “Voxelized atomic structure framework for materials design and discovery,” *Comput. Mater. Sci.*, vol. 230, p. 112431, Oct. 2023, doi: 10.1016/j.commatsci.2023.112431.
- [23] H. Park, Z. Li, and A. Walsh, “Has generative artificial intelligence solved inverse materials design?,” *Matter*, vol. 7, no. 7, pp. 2355–2367, Jul. 2024, doi: 10.1016/j.matt.2024.05.017.
- [24] I. Papadimitriou, I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris, “AI methods in materials design, discovery and manufacturing: A review,” *Comput. Mater. Sci.*, vol. 235, p. 112793, Feb. 2024, doi: 10.1016/j.commatsci.2024.112793.
- [25] M. Manica *et al.*, “Accelerating material design with the generative toolkit for scientific discovery,” *Npj Comput. Mater.*, vol. 9, no. 1, p. 69, May 2023, doi: 10.1038/s41524-023-01028-1.
- [26] S. R. Kalidindi, *Hierarchical Materials Informatics*. Elsevier, 2015. doi: 10.1016/C2012-0-07337-1.
- [27] T. Fast and S. R. Kalidindi, “Formulation and calibration of higher-order elastic localization relationships using the MKS approach,” *Acta Mater.*, vol. 59, no. 11, pp. 4595–4605, Jun. 2011, doi: 10.1016/j.actamat.2011.04.005.
- [28] A. Iskakov, Y. C. Yabansu, S. Rajagopalan, A. Kapustina, and S. R. Kalidindi, “Application of spherical indentation and the materials knowledge system framework to establishing microstructure-yield strength linkages from carbon steel scoops excised from high-temperature exposed components,” *Acta Mater.*, vol. 144, pp. 758–767, Feb. 2018, doi: 10.1016/j.actamat.2017.11.024.
- [29] A. E. Robertson and S. R. Kalidindi, “Efficient generation of anisotropic N-field microstructures from 2-point statistics using multi-output Gaussian random fields,” *Acta Mater.*, vol. 232, p. 117927, Jun. 2022, doi: 10.1016/j.actamat.2022.117927.
- [30] A. Cecen, T. Fast, and S. R. Kalidindi, “Versatile algorithms for the computation of 2-point spatial correlations in quantifying material structure,” *Integrating Mater. Manuf. Innov.*, vol. 5, no. 1, pp. 1–15, Dec. 2016, doi: 10.1186/s40192-015-0044-x.
- [31] A. P. Bartók, M. J. Gillan, F. R. Manby, and G. Csányi, “Machine-learning approach for one- and two-body corrections to density functional theory: Applications to molecular and condensed water,” *Phys. Rev. B*, vol. 88, no. 5, p. 054104, Aug. 2013, doi: 10.1103/PhysRevB.88.054104.
- [32] S. Kajita, N. Ohba, R. Jinnouchi, and R. Asahi, “A Universal 3D Voxel Descriptor for Solid-State Material Informatics with Deep Convolutional Neural Networks,” *Sci. Rep.*, vol. 7, no. 1, Art. no. 1, Dec. 2017, doi: 10.1038/s41598-017-17299-w.
- [33] D. T. Fullwood, S. R. Niezgodá, B. L. Adams, and S. R. Kalidindi, “Microstructure sensitive design for performance optimization,” *Prog. Mater. Sci.*, vol. 55, no. 6, pp. 477–562, Aug. 2010, doi: 10.1016/j.pmatsci.2009.08.002.

- [34] D. T. Fullwood, S. R. Kalidindi, S. R. Niezgodna, A. Fast, and N. Hampson, “Gradient-based microstructure reconstructions from distributions using fast Fourier transforms,” *Mater. Sci. Eng. A*, vol. 494, no. 1, pp. 68–72, Oct. 2008, doi: 10.1016/j.msea.2007.10.087.
- [35] C. Kelly and S. R. Kalidindi, “Recurrent localization networks applied to the Lippmann-Schwinger equation,” *Comput. Mater. Sci.*, vol. 192, p. 110356, May 2021, doi: 10.1016/j.commatsci.2021.110356.
- [36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, 2016.
- [37] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005. doi: 10.7551/mitpress/3206.001.0001.
- [38] M. C. Barry, S. Kumar, and S. R. Kalidindi, “Voxelized Representations of Atomic Systems for Machine Learning Applications,” in *Machine Learning in Molecular Sciences*, vol. 36, C. Qu and H. Liu, Eds., in Challenges and Advances in Computational Chemistry and Physics, vol. 36. , Cham: Springer International Publishing, 2023, pp. 67–89. doi: 10.1007/978-3-031-37196-7_3.
- [39] F. Faber, A. Lindmaa, O. A. von Lilienfeld, and R. Armiento, “Crystal structure representations for machine learning models of formation energies,” *Int. J. Quantum Chem.*, vol. 115, no. 16, pp. 1094–1101, 2015, doi: 10.1002/qua.24917.
- [40] S. Honrao, B. E. Anthonio, R. Ramanathan, J. J. Gabriel, and R. G. Hennig, “Machine learning of ab-initio energy landscapes for crystal structure predictions,” *Comput. Mater. Sci.*, vol. 158, pp. 414–419, Feb. 2019, doi: 10.1016/j.commatsci.2018.08.041.
- [41] A. Davariashtiyani, Z. Kadkhodaie, and S. Kadkhodaie, “Predicting synthesizability of crystalline materials via deep learning,” *Commun. Mater.*, vol. 2, no. 1, Art. no. 1, Nov. 2021, doi: 10.1038/s43246-021-00219-x.
- [42] X. Lei and A. J. Medford, “A Universal Framework for Featurization of Atomistic Systems,” *J. Phys. Chem. Lett.*, vol. 13, no. 34, pp. 7911–7919, Sep. 2022, doi: 10.1021/acs.jpcclett.2c02100.
- [43] P. R. Kaundinya, K. Choudhary, and S. R. Kalidindi, “Prediction of the Electron Density of States for Crystalline Compounds with Atomistic Line Graph Neural Networks (ALIGNN),” *JOM*, vol. 74, no. 4, pp. 1395–1405, Apr. 2022, doi: 10.1007/s11837-022-05199-y.
- [44] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, “Comparing molecules and solids across structural and alchemical space,” *Phys. Chem. Chem. Phys.*, vol. 18, no. 20, pp. 13754–13769, May 2016, doi: 10.1039/C6CP00415F.
- [45] R. Pederson, B. Kalita, and K. Burke, “Machine learning and density functional theory,” *Nat. Rev. Phys.*, vol. 4, no. 6, Art. no. 6, Jun. 2022, doi: 10.1038/s42254-022-00470-2.
- [46] Y. Zhao, K. Yuan, Y. Liu, S.-Y. Louis, M. Hu, and J. Hu, “Predicting Elastic Properties of Materials from Electronic Charge Density Using 3D Deep Convolutional Neural Networks,” *J. Phys. Chem. C*, vol. 124, no. 31, pp. 17262–17273, Aug. 2020, doi: 10.1021/acs.jpcc.0c02348.
- [47] K. Ryczko, D. Strubbe, and I. Tamblyn, “Deep Learning and Density Functional Theory,” *Phys. Rev. A*, vol. 100, no. 2, p. 022512, Aug. 2019, doi: 10.1103/PhysRevA.100.022512.
- [48] K. Ryczko, J. T. Krogel, and I. Tamblyn, “Machine Learning Diffusion Monte Carlo Energies,” *J. Chem. Theory Comput.*, vol. 18, no. 12, pp. 7695–7701, Dec. 2022, doi: 10.1021/acs.jctc.2c00483.

- [49] K. Ryczko, S. J. Wetzel, R. G. Melko, and I. Tamblyn, “Toward Orbital-Free Density Functional Theory with Small Data Sets and Deep Learning,” *J. Chem. Theory Comput.*, vol. 18, no. 2, pp. 1122–1128, Feb. 2022, doi: 10.1021/acs.jctc.1c00812.
- [50] A. D. Casey, S. F. Son, I. Billionis, and B. C. Barnes, “Prediction of Energetic Material Properties from Electronic Structure Using 3D Convolutional Neural Networks,” *J. Chem. Inf. Model.*, vol. 60, no. 10, pp. 4457–4473, Oct. 2020, doi: 10.1021/acs.jcim.0c00259.
- [51] G. Arora, S. Kamrava, P. Tahmasebi, and D. S. Aidhy, “Charge-density based convolutional neural networks for stacking fault energy prediction in concentrated alloys,” *Materialia*, vol. 26, p. 101620, Dec. 2022, doi: 10.1016/j.mtla.2022.101620.
- [52] A. V. Sinitskiy and V. S. Pande, “Deep Neural Network Computes Electron Densities and Energies of a Large Set of Organic Molecules Faster than Density Functional Theory (DFT),” Sep. 07, 2018, *arXiv*: arXiv:1809.02723. doi: 10.48550/arXiv.1809.02723.
- [53] B. Kolb, L. C. Lentz, and A. M. Kolpak, “Discovering charge density functionals and structure-property relationships with PROPhet: A general framework for coupling machine learning and first-principles methods,” *Sci. Rep.*, vol. 7, no. 1, Art. no. 1, Apr. 2017, doi: 10.1038/s41598-017-01251-z.
- [54] H. Mirzaee *et al.*, “Elastic constants from charge density distribution in FCC high-entropy alloys using CNN and DFT,” *APL Mach. Learn.*, vol. 2, no. 4, p. 046104, Dec. 2024, doi: 10.1063/5.0229105.
- [55] D. Montes de Oca Zapiain, A. Shanker, and S. R. Kalidindi, “Convolutional Neural Networks for the Localization of Plastic Velocity Gradient Tensor in Polycrystalline Microstructures,” *J. Eng. Mater. Technol.*, vol. 144, no. 011004, May 2021, doi: 10.1115/1.4051085.
- [56] G. H. Harrington, C. Kelly, V. Attari, R. Arroyave, and S. R. Kalidindi, “Application of a Chained-ANN for Learning the Process–Structure Mapping in Mg₂SixSn_{1-x} Spinodal Decomposition,” *Integrating Mater. Manuf. Innov.*, vol. 11, no. 3, pp. 433–449, Sep. 2022, doi: 10.1007/s40192-022-00274-3.
- [57] A. Mann and S. R. Kalidindi, “Development of a Robust CNN Model for Capturing Microstructure-Property Linkages and Building Property Closures Supporting Material Design,” *Front. Mater.*, vol. 9, 2022, Accessed: May 30, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fmats.2022.851085>
- [58] K. Choudhary *et al.*, “The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design,” *Npj Comput. Mater.*, vol. 6, no. 1, Art. no. 1, Nov. 2020, doi: 10.1038/s41524-020-00440-1.
- [59] G. Kresse and J. Hafner, “*Ab initio* molecular dynamics for liquid metals,” *Phys. Rev. B*, vol. 47, no. 1, pp. 558–561, Jan. 1993, doi: 10.1103/PhysRevB.47.558.
- [60] G. Kresse and J. Furthmüller, “Efficient iterative schemes for *ab initio* total-energy calculations using a plane-wave basis set,” *Phys. Rev. B*, vol. 54, no. 16, pp. 11169–11186, Oct. 1996, doi: 10.1103/PhysRevB.54.11169.
- [61] G. Kresse and J. Furthmüller, “Efficiency of *ab-initio* total energy calculations for metals and semiconductors using a plane-wave basis set,” *Comput. Mater. Sci.*, vol. 6, no. 1, pp. 15–50, Jul. 1996, doi: 10.1016/0927-0256(96)00008-0.
- [62] F. Birch, “Finite Elastic Strain of Cubic Crystals,” *Phys. Rev.*, vol. 71, no. 11, pp. 809–824, Jun. 1947, doi: 10.1103/PhysRev.71.809.
- [63] F. D. Murnaghan, “Finite Deformations of an Elastic Solid,” *Am. J. Math.*, vol. 59, no. 2, p. 235, Apr. 1937, doi: 10.2307/2371405.

- [64] B. Lowekamp, D. Chen, L. Ibanez, and D. Blezek, “The Design of SimpleITK,” *Front. Neuroinformatics*, vol. 7, 2013, Accessed: Jun. 01, 2023. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fninf.2013.00045>
- [65] G. Kresse and D. Joubert, “From ultrasoft pseudopotentials to the projector augmented-wave method,” *Phys. Rev. B*, vol. 59, no. 3, pp. 1758–1775, Jan. 1999, doi: 10.1103/PhysRevB.59.1758.
- [66] N. A. W. Holzwarth, “Updated comments on projector augmented wave (PAW) implementations within various electronic structure code packages,” *Comput. Phys. Commun.*, vol. 243, pp. 25–29, Oct. 2019, doi: 10.1016/j.cpc.2019.05.009.
- [67] R. M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, 1st ed. Cambridge University Press, 2004. doi: 10.1017/CBO9780511805769.
- [68] A. Cecen, H. Dai, Y. C. Yabansu, S. R. Kalidindi, and L. Song, “Material structure-property linkages using three-dimensional convolutional neural networks,” *Acta Mater.*, vol. 146, pp. 76–84, Mar. 2018, doi: 10.1016/j.actamat.2017.11.053.
- [69] A. E. Robertson and S. R. Kalidindi, “Digital Representation and Quantification of Discrete Dislocation Structures,” *JOM*, vol. 73, no. 7, pp. 2143–2158, Jul. 2021, doi: 10.1007/s11837-021-04669-z.
- [70] M. C. Barry, K. E. Wise, S. R. Kalidindi, and S. Kumar, “Voxelized Atomic Structure Potentials: Predicting Atomic Forces with the Accuracy of Quantum Mechanics Using Convolutional Neural Networks,” *J. Phys. Chem. Lett.*, vol. 11, no. 21, pp. 9093–9099, Nov. 2020, doi: 10.1021/acs.jpcclett.0c02271.
- [71] A. Gupta and S. R. Kalidindi, “Addressing biases in spectral databases for increasing accuracy and computational efficiency of crystal plasticity computations,” *Int. J. Plast.*, vol. 138, p. 102945, Mar. 2021, doi: 10.1016/j.ijplas.2021.102945.
- [72] M. Knezevic, S. R. Kalidindi, and D. Fullwood, “Computationally efficient database and spectral interpolation for fully plastic Taylor-type crystal plasticity calculations of face-centered cubic polycrystals,” *Int. J. Plast.*, vol. 24, no. 7, pp. 1264–1276, Jul. 2008, doi: 10.1016/j.ijplas.2007.12.002.
- [73] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.
- [74] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “(1986) D. E. Rumelhart, G. E. Hinton, and R. J. Williams, ‘Learning internal representations by error propagation,’ *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. I, D. E. Rumelhart and J. L. McClelland (Eds.) Cambridge, MA: MIT Press, pp. 318-362,” in *Neurocomputing, Volume 1*, J. A. Anderson and E. Rosenfeld, Eds., The MIT Press, 1988, pp. 675–695. doi: 10.7551/mitpress/4943.003.0128.
- [75] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [76] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.
- [77] Z. Yang *et al.*, “Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches,” *Acta Mater.*, vol. 166, pp. 335–345, Mar. 2019, doi: 10.1016/j.actamat.2018.12.045.

- [78] W. A. Saidi, W. Shadid, and I. E. Castelli, "Machine-learning structural and electronic properties of metal halide perovskites using a hierarchical convolutional neural network," *Npj Comput. Mater.*, vol. 6, no. 1, p. 36, Apr. 2020, doi: 10.1038/s41524-020-0307-8.
- [79] I. Peivaste, S. Ramezani, G. Alahyarizadeh, R. Ghaderi, A. Makradi, and S. Belouettar, "Rapid and accurate predictions of perfect and defective material properties in atomistic simulation using the power of 3D CNN-based trained artificial neural networks," *Sci. Rep.*, vol. 14, no. 1, p. 36, Jan. 2024, doi: 10.1038/s41598-023-50893-9.
- [80] P. Napoletano, F. Piccoli, and R. Schettini, "Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity," *Sensors*, vol. 18, no. 1, p. 209, Jan. 2018, doi: 10.3390/s18010209.
- [81] L. Wu *et al.*, "A deep CNN-based constitutive model for describing of statics characteristics of rock materials," *Eng. Fract. Mech.*, vol. 279, p. 109054, Feb. 2023, doi: 10.1016/j.engfracmech.2023.109054.
- [82] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 03, 2019, *arXiv*: arXiv:1912.01703. doi: 10.48550/arXiv.1912.01703.
- [83] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," 2015, *arXiv*. doi: 10.48550/ARXIV.1502.01852.
- [84] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 29, 2017, *arXiv*: arXiv:1412.6980. doi: 10.48550/arXiv.1412.6980.
- [85] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," May 03, 2017, *arXiv*: arXiv:1608.03983. doi: 10.48550/arXiv.1608.03983.
- [86] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*. in Springer Series in Statistics. New York, NY: Springer New York, 2001. doi: 10.1007/978-0-387-21606-5.
- [87] C. M. Bishop, *Pattern recognition and machine learning*. in Information science and statistics. New York: Springer, 2006.