THE JOURNAL OF
PHYSICAL CHEMISTRY C

# Predicting Elastic Properties of Materials from Electronic Charge Density Using 3D Deep Convolutional Neural Networks

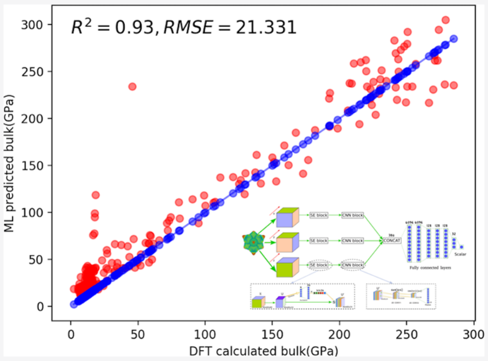
*Published as part of The Journal of Physical Chemistry virtual special issue "Machine Learning in Physical Chemistry".*

Yong Zhao, Kunpeng Yuan, Yinqiao Liu, Steph-Yves Louis, Ming Hu,* and Jianjun Hu*

**Cite This:** *J. Phys. Chem. C* 2020, 124, 17262−17273 Read Online

ACCESS | Metrics & More | Article Recommendations

**ABSTRACT:** Materials representation plays a key role in machine learning-based prediction of materials properties and new materials discovery. Currently both graph and three-dimensional (3D) voxel representation methods are based on the heterogeneous elements of the crystal structures. Here, we propose to use electronic charge density (ECD) as a generic unified 3D descriptor for materials property prediction with the advantage of possessing close relation with the physical and chemical properties of materials. We developed an ECD-based 3D convolutional neural networks (CNNs) for predicting the elastic properties of materials, in which CNNs can learn effective hierarchical features with multiple convolving and pooling operations. Extensive benchmark experiments over 2170 $Fm\bar{3}m$ face-centered-cubic materials show that our ECD-based CNNs can achieve good performance for elasticity prediction. Especially, our CNN models based on the fusion of elemental Materials-Agnostic Platform for Informatics and Exploration features and ECD descriptors achieved the best fivefold cross-validation performance. More importantly, we showed that our ECD-based CNN models can achieve significantly better extrapolation performance when evaluated over nonredundant data sets, where there are few neighbor-training samples around test samples. As an additional validation, we evaluated the predictive performance of our models on 329 materials of space group $Fm\bar{3}m$ by comparing to density functional theory calculated values, which shows a better prediction power of our model for bulk modulus than shear modulus. Because of the unified representation power of ECD, it is expected that our ECD-based CNN approach can also be applied to predict other physical and chemical properties of crystalline materials.

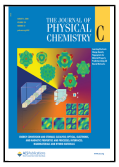


## ■ INTRODUCTION

Because of its time and cost efficiency, data-driven machine learning (ML) approaches have been increasingly used for material property prediction[1,2] and materials screening and discovery.[3,4] Although the great potential of machine learning in material discovery is widely acknowledged, it has yet to achieve high success as it has in other scientific fields. There are two major challenges to address to realize their potential. The first one is that, in materials science, there is usually only a small amount of characterization/property data (labeled samples) available, the so-called small data set problem.[5] For example, the number of materials with characterized thermal conductivity is less than 400,[6] while the number of materials with characterized ionic conductivity is even less than 50.[3] With limited data, a major challenge for building an accurate prediction model for a target material property is how to find suitable material descriptors, which is a key factor that determines the prediction performance of machine learning models. A descriptor encodes materials' elemental, structural, and other physical information into a representation that machine learning algorithms can map to materials properties.[7−10]

In the past decade, a large number of descriptors have been proposed to encode materials,[7−19] which is one of the most critical factors in machine learning applications in materials property prediction as shown in the review by Liu et al.[20] In general, those descriptors are based on materials composition, their electronic or geometric structures as shown by the integrated feature calculation routines as implemented in the Matminer package.[21] A widely used set of material composition features is the Materials-Agnostic Platform for Informatics and Exploration (Magpie) features, which are

**Received:** March 16, 2020
**Revised:** June 7, 2020
**Published:** June 12, 2020



https://dx.doi.org/10.1021/acs.jpcc.0c02348

based on the statistics of elemental properties in a material.[8] Mendeleev numbers (MN) has also been used by P. Villars et al.[12] to classify chemical systems by using the minimum and maximum MN versus the ratio between the minimum and the maximum MN. Ghiringhelli et al.[17] developed 23 primary features, based on atomic properties, to explore the energy difference of zinc blende, wurtzite, and rock salt semiconductors. Han et al.[22] leveraged three key factors as the descriptors for classic machine learning methods to predict thermal conductivity effectively. Ward et al.[8] presented a comprehensive set of features for a wide variety of material compositions. This set contains four unique categories: stoichiometric attributes, elemental property statistics, electronic structure attributes, and ionic compound attributes. Elemental descriptors have achieved great success in predicting band gaps,[23] formation energies,[24] crystal system, etc., but these descriptors have their severe limitations: elemental descriptors are merely based on material compositions, while most materials properties are strongly dependent on their atomic structures. There are also materials that share the same composition but exist in completely different structures. It is a common understanding that the most important information for analyzing a material's property is its structure. How atoms coordinate and interact with each other conveys rich information on the properties of the materials. Therefore, structural features play a key role in developing prediction models of materials. Currently, there are several successful applications that use structural features to predict materials properties.[7,9,10,13−16,19] Rupp and colleagues applied the Coulomb matrix (CM) features for predicting the atomization energies of small isolated organic molecules.[11,13,14] CM formulates the electrostatic interaction between nuclei into a matrix representation. Pham et al.[19] developed the orbital-field matrix (OFM) descriptor, based on the distribution of valence-shell electrons, to predict formation energies and atomization energies with high accuracy. Bartók et al.[15] proposed the smooth overlap of atomic positions (SOAP), which describes the similarity between two atomic environments to define a metric in the structural cell. The local similarity can be combined further to form a global measure of similarity for the evaluation of molecular properties.[10] More recently, voxel grid representation with atom features has been proposed to predict Hartree energies.[9] Atom density and related continuous representations have also been proposed for materials representation and are used for crystal structure generation.[25,26] On the one hand, graph neural networks have also been introduced to learn structural representation from material structures for predicting formation energy, band gaps, bulk modulus, etc. with great success.[27,28] On the other hand, deep learning has been utilized to extract three-dimensional (3D) spatial features for material property prediction. In refs 7 and 9, 3D CNNs have been applied to extract 3D geometric features from material microstructures represented as 3D matrices.[7] In this work, a data set with 5900 microstructures was created, where a microstructure is the quantification of the material structure. Each microstructure is represented by a feature matrix of dimensions 51 × 51 × 51, where each feature corresponds to a vector. Kajita et al.[9] developed a descriptor called Reciprocal 3D Voxel Space Descriptor (R3DVS) from the distributions of the valence electron density for 680 oxides. The authors enlarge the data set by rotating R3DVS for testifying the invariance of R3DVS to rotation and translation. R3DVS compacts the density of electrons in the bond generation.

In this paper, we propose to leverage convolutional neural networks (CNNs) to learn physically meaningful features from the three-dimensional electronic charge density (ECD) of materials for elastic property prediction. Since physical and chemical properties of materials are related to the transferability between atoms (nuclei) and the presence of electronic charges or electronic multipoles on atoms or molecules,[29−32] extraction of informative features from materials ECD can help predict materials properties. The ECD of a material can be calculated as a 3D matrix that describes the amount of electronic charge per volume. It represents the charge of electrons in the effective material space. By explicitly encoding the geometry of materials, ECD is supposed to have high transferability with respect to different compositions and structures.[33] As the ECD captures both geometrical and electronic structural features, the 3D distribution of electronic charge density would have the advantage over classical one-dimensional (1D) and two-dimensional (2D) descriptors as well as heterogeneous 3D structural descriptors in terms of the correlation with the electrochemical properties of materials. Indeed, ECD and its related electronic properties such as the electrostatic potential, electron localization function, and noncovalent interaction index have been used to analyze many materials characteristics, including bonding, defects, stability, and reactivity and electron, ion, and thermal transport.[33] For example, ECD was used to predict eight materials properties by using the Fourier coefficients of the planar averaged Kohn−Sham charge density fingerprint features.[34] Abraham et al.[35] calculated 2D charge density to predict the chemical bonding and charge transfer in magnetic compounds. However, both approaches failed to take advantage of the flexibility of the 3D representation.[36] Compared to conventional ML models, 3D CNNs can better link 3D descriptors to the properties efficiently as shown by ref 7 (linkage between microstructure and homogenized property) and ref 9 (linkage between R3DVS and Hartree energies, testify the invariance to rotation for R3DVS). We believe that the unified representation of ECD makes it easier to learn unified continuous representation for facilitating downstream prediction tasks by deep convolutional neural networks.[25] In ref 37, the authors used the particle packing and the quartet structure generation set (QSGS) methods to generate microstructures of 3D composites. Instead of directly applying 3D CNNs to the 3D matrix, 2D CNNs are used to predict thermal conductivity of composites in the work by obtaining a series of cross-section slices from the 3D structure, which are stacked in order as the channel direction. This approach, however, may lead to too much global information loss in our ECD-based elastic property prediction.

We explored two types of convolutional neural network models for ECD-based elastic property prediction. One is the standard 3D CNNs with two convolutional layers. The other one is a projected 2D CNN model, in which the ECD matrix is converted to three different imagelike representations that are then fed to three 2D CNN networks, whose outputs are then fused together. The difference of 2D CNNs used in this work and in ref 37 is that we compressed 3D structures from three directions, and this strategy can preserve the global structural information to a large extent, compared to selecting some slices from the 3D matrix. This allows us to exploit the powerful hierarchical representation learning capability of 2D CNNs as demonstrated by their success in computer vision.[38−42] We then conducted extensive benchmark experiments based on a data set consisting of 2170 face-centered cubic (Fcc)-structured materials and 11 nonredundant data sets generated by leaving one element out at a time.

Our contributions can be summarized as follows:

- We propose to exploit the ECD descriptor as unified 3D materials representation and combine it with two types of 3D CNNs for materials elastic property prediction. We also developed a fusion CNN model based on CNN +Magpie and CNN+ECD models.
- We developed a standard benchmark ECD data set, named "FCC2170", calculated from 2710 Fcc structured materials from Inorganic Crystal Structure Database (ICSD). This database is characterized by its highly redundant samples with very similar compositions. We also developed 11 nonredundant data sets for evaluating the extrapolation capability of ECD+CNN models.
- We performed extensive prediction experiments over the aforementioned data sets using fivefold cross validation. Our results show that our ECD+CNN can be complementary to elemental Magpie feature-based models, while it can significantly outperform them over nonredundant data sets, thus demonstrating superior performance on some extrapolation experiments.
- We analyzed the situations when our ECD+CNN models perform better by visually inspecting the distribution of test samples and training samples in the 2D space mapped from the learned features.
- We validated the prediction performance of our models by comparison with density functional theory (DFT)-calculated bulk and shear modulus for a set of 329 materials of the space group $Fm\overline{3}m$ collected from the Open Quantum Materials Database (OQMD) database.

## ■ METHODS

**Data Sets.** Here we discuss how we create the benchmark data sets for training and validating our proposed method. Because of the high computational cost to calculate electronic charge density for all the materials in the Materials Project (MP) database, we decide to focus only on the materials of one specific space group. First we retrieved 2170 material structures of the $Fm\overline{3}m$ space group (excluding lanthanides and actinides) from the Materials Project database (https://materialsproject.org). We chose the $Fm\overline{3}m$ structure, because its structure is simple, and it takes less time to calculate the related elastic properties using DFT. Most materials of the $Fm\overline{3}m$ space group do not have the charge density or elastic properties available in the MP database. Hence, we calculated both the electronic charge density[43] and the elastic property[44] using VASP[45−47] for the 2170 samples to form the "FCC2170" data set. Table 1 lists the top 11 elements that are contained in at least 200 materials of our FCC2070 data set. Among them, most are from group 1 (lithium, sodium, potassium, rubidium, and cesium), group 13 (indium and thallium), and group 17 (fluorine, chlorine, and bromine). The rest includes scandium from group 3. With this data set, we then use the commonly used cross-validation method to evaluate our model's interpolation performance as done in most machine learning-based property prediction studies.[28]

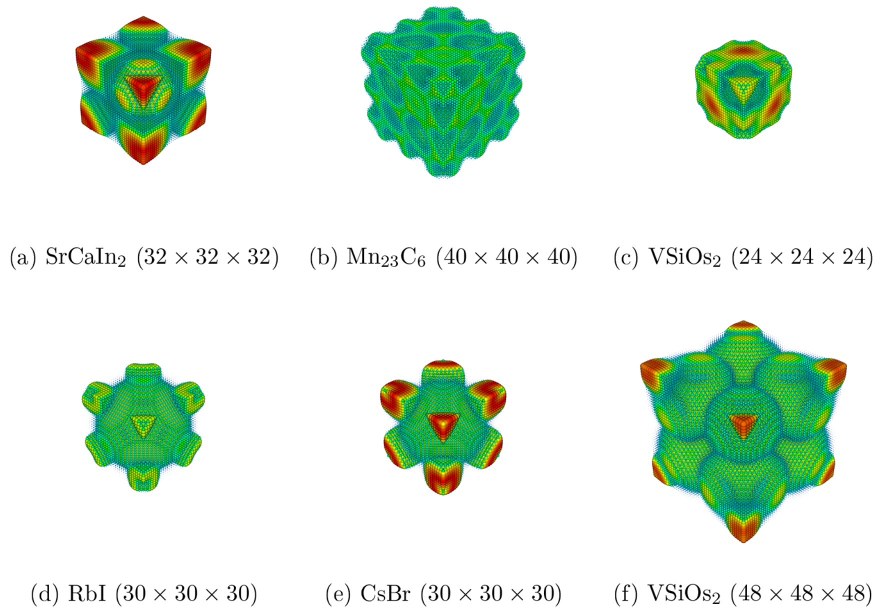
Table 1. Statistics of Nonredundant Datasets

| element | F | K | Rb | Cs |
|---|---|---|---|---|
| data set | FCC-F-1775 | FCC-K-1800 | FCC-Rb-1802 | FCC-Cs-1814 |
| train set size | 1755 | 1800 | 1802 | 1814 |
| test set size | 415 | 370 | 368 | 356 |
| element | Na | Cl | In | Br |
| data set | FCC-Na-1877 | FCC-Cl-1880 | FCC-In-1937 | FCC-Br-1938 |
| train set size | 1877 | 1880 | 1937 | 1938 |
| test set size | 293 | 290 | 233 | 232 |
| element | Li | Sc | Tl | |
| data set | FCC-Li-2148 | FCC-Sc-1952 | FCC-Tl-1966 | |
| train set size | 2148 | 1952 | 1966 | |
| test set size | 222 | 218 | 204 | |

To validate our model's extrapolation capability, we define a set of leave-one-element-out data sets, which are better for evaluating the extrapolation capability of ML models.[48−50] For all samples in FCC2170, we first select those samples containing one specific element E as the test set and then designate the remaining samples as the training set. These data sets are called FCC-E-N data sets, where E is the element of interest, and N is the number of training samples without element E. Statistics of all these nonredundant data sets generated from FCC2170 for elements contained in more than 200 materials are shown in Table 1.

**Representations of Materials.** We studied and compared two material representations for elastic property prediction including Magpie[8] and ECD.[51]

- *Magpie Features*. Materials-Agnostic Platform for Informatics and Exploration is an extensive set of features related to the constituent elements in materials. The set covers a broad range of physical and chemical properties that fall into four different categories: stoichiometric features, elemental property statistics, electronic structure features, and ionic compound features.[8] Stoichiometric features only contain the number of elements in the compound and their several $L^p$ norms.[8] Elemental property statistics are calculated by computing several statistics (e.g., average, minimum, maximum, range, and mode) of 22 different elemental properties.[8] These properties include row and column on the periodic table, average atomic number, the range of atomic radii between all elements presenting in compositions, Mendeleev number, atomic weight, covalent radius, and electronegativity. Electronic structure features are the average fraction of s, p, d, and f valence electrons.[18] Ionic compound features include the capability of forming an ionic compound (when we assume all elements present in a single oxidation state) and two adaptions for calculating the fractions of a compound based on electronegativity.[52]
- *Electronic Charge Density*. ECD in the form of 3D structural matrix represents the spatial distribution of electronic charge density in crystalline materials. It can be calculated by local quantum-mechanical functions related to the Pauli exclusion principle.[51] The ab initio method is used to calculate the Hartree−Fock wave function and the electron localization function (ELF).[43] A single determinant wave function is calculated on a grid in the 3D space by hartree Fock or Kohn−Sham orbitals $\varphi_i$ as follows



(a) $SrCaIn_2$ (32 × 32 × 32) (b) $Mn_{23}C_6$ (40 × 40 × 40) (c) $VSiOs_2$ (24 × 24 × 24)

(d) RbI (30 × 30 × 30) (e) CsBr (30 × 30 × 30) (f) $VSiOs_2$ (48 × 48 × 48)

**Figure 1.** Visualization of ECDs for six materials showing clearly contrasting structural features (top and bottom rows). The top row shows materials with high bulk modulus, and the bottom row shows materials with low bulk modulus. $l \times w \times h$ is the actual length, width, and height of each ECD matrix.

$$\mathrm{ELF} = \frac{1}{1 + \left(\frac{D}{D_{\mathrm{h}}}\right)^2} \tag{1}$$

where

$$D = \frac{1}{2}\sum_i |\Delta\varphi_i|^2 - \frac{1}{8}\frac{|\Delta\rho|^2}{\rho}$$

$$D_{\mathrm{h}} = \frac{3}{10}(3\pi^2)^{5/3}\rho^{5/3} \tag{2}$$

where ELF has values between 0 and 1, where 1 means the perfect localization. Figure 1 shows the visualizations for the ECDs of six representative materials, namely, $SrCaIn_2$, $Mn_{23}C_6$, $VSiOs_2$, RbI, CsBr, and $Rb_2TeBr_6$, where $SRCaIn_2$, $Mn_{23}C_6$, and $VSiOs_2$ possess high bulk modulus. These visualizations consist of points that correspond to the values in a material's ECD matrix. The color and area of each point represents the size of each value, and together they show the distribution of a material's electron clouds. When the values of these points were plotted, we found that points appear in both thick and thin clouds, within the cubes, as shown in Figure 1a–c. Figure 1d–f shows a clear difference from the top-row figures. In these figures, there are some empty spaces in the cubes and some dense clusters present in the remaining area. These observations correspond to the physical reality that materials with high bulk modulus usually have active electrons orbiting across the whole space strongly when compared to materials with lower bulk modulus. Among all six materials, we find that, although the ECD visualizations share many similar characteristics, there are a few distinct differences between them. These minor variations make it possible for us to employ 3D CNNs to learn the structural and physical patterns that may characterize the material's elastic properties.

**Machine Learning Methods.** In this work, we use Random Forest (RF) and Convolutional Neural Networks (CNNs) with Magpie features as the baseline methods. We propose that CNNs with ECD can capture certain characteristic relationships between material structures and their elastic properties.

Random Forest[53] is a widely used machine learning model in material informatics because of its high accuracy and robustness.[54−56] As an ensemble learning algorithm, an RF aggregates the results from different decision trees (50 in this work). The decision trees are randomly trained based on subsets of training samples and features. Within a decision tree, a set of decision rules (e.g., melting temperature > 200.0) is learned by minimizing the variance of the decision tree. For predicting elastic properties, RF calculates the final results by averaging outputs of all decision trees.

Convolutional Neural Networks are a type of feed-forward neural network interleaved with convolutional, pooling, and fully connected layers. It has achieved state-of-the-art (SOTA) performance when applied to computer vision and natural language processing.[39,42,57] The convolutional unit is the core building block of CNNs, which is inspired by the multilayered organization of the visual cortex. The unit consists of multiple learnable filters with a given receptive field and weight parameters. In our case, the filters are convolved across the full depth of the input volume of the ECD.[58] The filters are learned hierarchically, where low-level features generate more condensed representations. The computational unit can be constructed by a transformation $U = F_{\mathrm{tr}}(X)$, $X \in \mathbb{R}^{L' \times W' \times H' \times C'}$,$U \in \mathbb{R}^{L \times W \times H \times C}$. $F_{\mathrm{tr}}$ denotes the convolutional operation. Let $V = [v_1, v_2, ..., v_C]$ be the learnable convolutional filters. Then the outputs of $F_{\mathrm{tr}}$ can be written as $U = [u_1, u_2, ..., u_C]$, where

$$u_c = v_c * X = \sum_{i=1}^{C'} v_c^i * x^i \tag{3}$$

Here * denotes the dot product, $v_c = [v_c^1, v_c^2, ..., v_c^{C'}]$, $X = [x^1, x^2, ..., x^{C'}]$. We removed the bias terms for simplicity. $v_c^i$ is a 3D spatial filter convolving on a single channel of $X$. Stacked outputs of filters produce a four-dimensional (4D) tensor

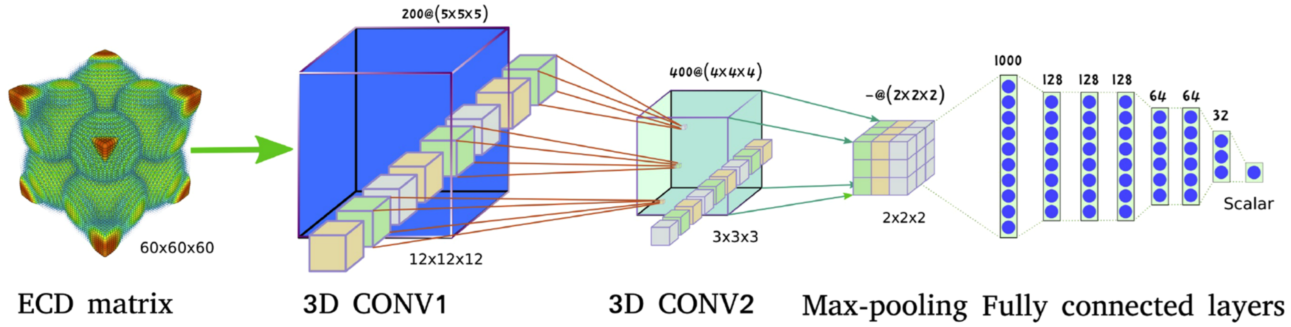


**Figure 2.** Architecture of 3D CNN with ECD representation. The "Scalar" stands for the bulk or shear modulus. The numbers above each convolutional layer are its parameter settings. For instance, 200@(5 × 5 × 5) means 200 filters with size of 5 × 5 × 5. Unless it is specified, the stride is always the same with the filter size. Two consecutive convolutional layers are followed by a max-pooling with pooling size and strides both of 2 × 2 × 2. The numbers below are outputs of each layer. For fully connected layers, the numbers above them are the number of neurons.

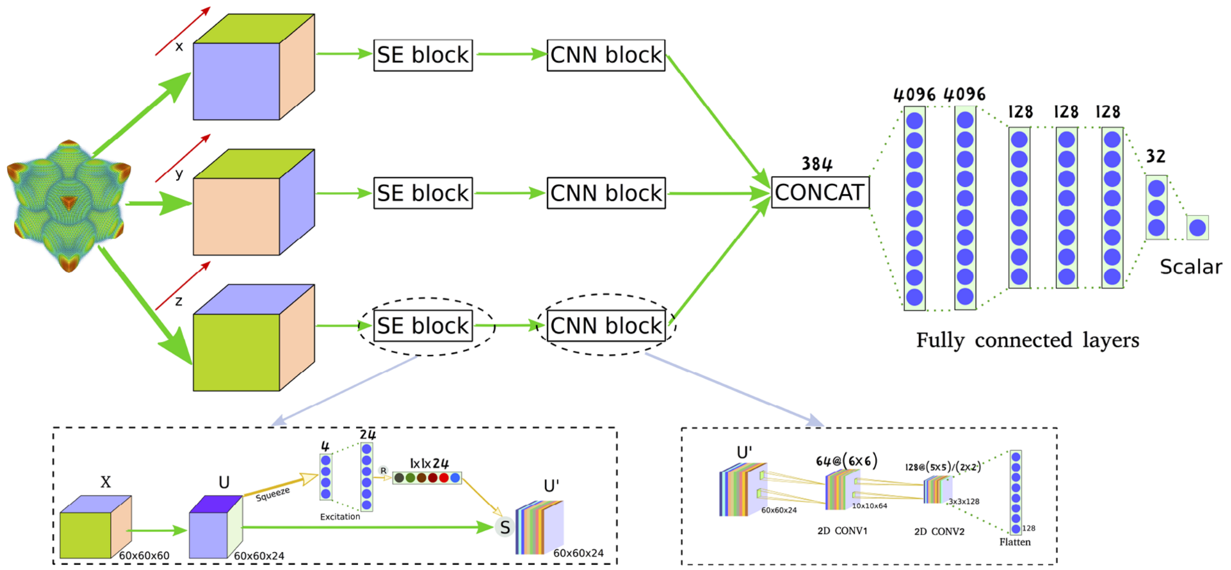


**Figure 3.** Architecture of the 2D CNN with ECD representation. The Scalar stands for the bulk or shear modulus. The model includes three parts: mainframe, SE block, and CNN block. In the mainframe, we have three branches whose outputs are concatenated and fed into six fully connected layers. The numbers above each component/layer are the number of neurons of that layer. In the SE block, the labels of **R** and **S** are reshape and channel-wise multiplication operations. For simplicity, we ignore the max-pooling layers following every convolutional layer in the CNN block. Numbers below each component are the output dimension of that layer.

activation map.[58] A pooling layer is used to do nonlinear downsampling. It partitions the 3D input into a set of rectangular boxes. In max-pooling, the pooling layer outputs the maximum value of each subregion. Then a 3D tensor is activated through a rectified linear unit (ReLu).[42] The ReLu operation can be denoted by max(0, $P$), where $P$ is the tensor generated by the max-pooling operation. The same procedure can be applied repeatedly to the whole activation map. Finally, the output of the convolutional layers is fed to one or more fully connected layers to accomplish the regression step. Similar procedures are applied in the CNN block in Figure 3.

We implemented two types of convolutional neural networks for learning ECD-based features for elastic property prediction. Figure 2 depicts the 3D CNN architecture in our work. This model has two consecutive convolutional layers followed by a max pooling layer and then seven fully connected layers. For simplicity, we did not show the ReLu[42] activation for all neural layers in Figure 2. The filter sizes of two convolution layers are 5 × 5 × 5 and 4 × 4 × 4, respectively, and the stride has the same size as that of the convolution filters. For all max-pooling layers, the sizes of the filters and strides are 2 × 2 × 2. The ECD matrices are fed to the 3D convolutional and pooling layers, and then the output matrix is flattened and passed to succeeding fully connected layers to calculate final predictions.

Figure 3 shows the architecture of our 2D CNNs for elastic property prediction. The ECD matrix does not have the concept of channel as images. Thus, we rotated the ECD matrix so that we can face the cube from $x$,$y$,$z$ axes as shown in different colors of cubes in Figure 3. Then the direction facing toward us is considered as the channel direction. To model the interdependencies between channels, we used the Squeeze-and-Excitation (SE) network,[59] which can exploit this interdependency by feature recalibration. This model selectively highlights the informative features and suppresses less useful ones. An SE block is shown in the left corner of Figure 3. In this module, 24 filters of size of 1 × 1 are used to downsample the ECD matrix, which was first proposed in ref 60. A nonlinearity operation is performed on each pixel across the channels. After the nonlinear projection, the ECD matrix $X$ of size 60 × 60 × 60 is reduced to the feature map $U$ of size 60 × 60 × 24. A global average pooling is then used to shrink the

Table 2. Performance Comparisons of Models with Magpie and ECD Descriptors using Fivefold Cross Validation

| type | RF+Magpie | | CNN+Magpie | | 3D-CNN+ECD | | 2D CNN+ECD | | fusion | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE |
| bulk | 0.943 | 18.721 | 0.944 | 18.423 | 0.884 | 26.819 | 0.912 | 23.401 | 0.955 | 16.530 |
| shear | 0.794 | 16.142 | 0.745 | 17.959 | 0.745 | 17.944 | 0.768 | 17.192 | 0.804 | 15.780 |

feature map into a vector of size 24 along with the dimensions of width and height. Then we use a small set of fully connected layers to transform this vector into higher-level features. The number of neurons on each layer is 4 and 24, respectively. The output $s$ of the last fully connected layer is reshaped into the size of 1 × 1 × 24. The last step is nonlinear excitation, and the final output $U'$ of a block is achieved by rescaling the $U$ with the activated $s$

$$U' = U \odot \sigma(s) \tag{4}$$

where $\sigma$ is the sigmoid activation function that implements the nonlinear transformation, and $\odot$ denotes the channel-wise multiplication between the scalar $s$ and the feature map $U'$.

The SE block in our 2D CNN architecture is followed by CNN blocks. A CNN block has two regular convolutional layers followed by a max-pooling layer. The first convolution neural has the same filter size and strides of 6 × 6, and there are 64 filters in this layer. The second CNN layer has a filter size of 5 × 5 and stride of 2 × 2, and there are 128 filters in total. All max-pooling layers have the same pooling size and strides of 2 × 2, respectively.

For each of the projection maps of $x$, $y$, and $z$, there is an SE and CNN block for feature extraction. The outputs of them are concatenated into a vector of size 384. Six fully connected layers are then used to map these learned features into elastic property values. The numbers of neurons on these fully connected layer are 4096, 4096, 128, 128, 128, and 32, respectively.

For the baseline algorithm, we also train a 2D CNN model with the Magpie features. To do that, we append 12 zeros to the Magpie features to get a vector of 1 × 144, which is then reshaped into a 2D matrix of size 12 × 12. The CNN model for Magpie features has two consecutive convolutional layers followed by an average pooling layer. Then an additional convolutional layer is added followed by two fully connected layers. The model parameters are set as follows: the kernel size and strides of the first convolutional layer are 3 × 3 and 1 × 1, and the number of filters is 32; the kernel size and strides of the second convolutional layer are 3 × 3 and 1 × 1, and the number of filters is 48; the pooling size and strides of the average pooling layer are both set as 2 × 2; the kernel size and strides of the third convolutional layer are 3 × 3 and 1 × 1, and the number of filters is 64; the numbers of neurons of the two fully connected layers are 48 and 32, respectively.

**Training and Implementation.** Figure 2 shows the detailed architecture of our 3D CNN and its parameters. The models are implemented using the open-source libraries of TensorFlow (https://www.tensorflow.org) and Keras (https://keras.io). The performance of the models is evaluated using fivefold cross validation. The input ECD has a shape of 60 × 60 × 60 by interpolation for smaller matrices. The CNN for Magpie is also trained using the Adam optimizer[61] with a batch size of 32 and learning rate of 0.001. The 3D CNN model parameters are learned using the Adam optimizer[61] with an initial learning rate of 0.0005. For the 2D CNNs with ECD, we use the stochastic gradient descent (SGD) optimizer to learn the model parameters. The initial learning rate is 0.001, and it drops by $0.5\left\lfloor \frac{\text{epoch}}{10} \right\rfloor$, where *epoch* is the current epoch. The mean absolute error (MAE) is used as the loss function for all three CNN models. The open source matminer (https://hackingmaterials.lbl.gov/matminer/) is utilized to calculate the Magpie features.

## RESULTS AND DISCUSSIONS

In this section, we discuss the experiments demonstrating the potential of ECD for material representation and elastic property prediction. The experiments are separated into two parts in terms of the evaluation approaches: experiments with fivefold cross validation and experiments focusing on extrapolation performance evaluation. All experiments of CNN models are repeated five times, and the result presented herein is the average of their outputs.

**Fivefold Cross Validation Experiments with Redundant Data Set.** Table 2 shows the results from fivefold cross validation on the whole data set with 2170 samples. We find that the baseline models using Magpie features are better than the CNNs with ECD across all evaluation metrics for predicting bulk and shear moduli. Overall, RF with Magpie performs slightly better than CNNs with Magpie. Although the $R^2$ of RF with Magpie is 0.001 lower than that of CNNs with Magpie in predicting bulk modulus, the RF with Magpie achieves much better results in predicting shear modulus ($R^2$ is 0.049 higher). Similar observations apply to performance evaluated in terms of root mean square error (RMSE). This better performance of Magpie-based RF models is not unexpected. First, all samples in this FCC2170 data set belong to the $Fm\bar{3}m$ space group. By sharing similar structures, the Magpie features are able to capture most of the elastic property variation due to composition difference. The high structural similarity of the data set helps the baseline methods based on composition Magpie features predict the elastic properties well. Another reason is that the FCC2170 contains many similar samples in terms of compositions. The high-redundant samples also make the baseline models with Magpie features to make precise predictions by exploiting redundant neighbor samples in the training set when evaluated on the test set during cross-validation. However, the machine learning models trained with a redundant training set can lead to low extrapolation performance as shown in our previous study.[48] In terms of dimension size of the Magpie and ECD descriptors, the ECD has a much larger dimension of 60 × 60 × 60 compared to 132 of the Magpie features. Since higher input data dimensions usually lead to machine learning models with more parameters, and more training samples are needed to achieve good prediction performance. From this perspective, the limited data set size in our problem actually favors the baseline models with Magpie features.

Here we show that the ECD can be used as a complementary materials descriptor for elastic property prediction together with the Magpie features. To verify this, we pretrained a CNN model with Magpie features and a 2D

Table 3. Extrapolation Prediction Performance Comparison on Nonredundant Leave-One-Element-Out Datasets

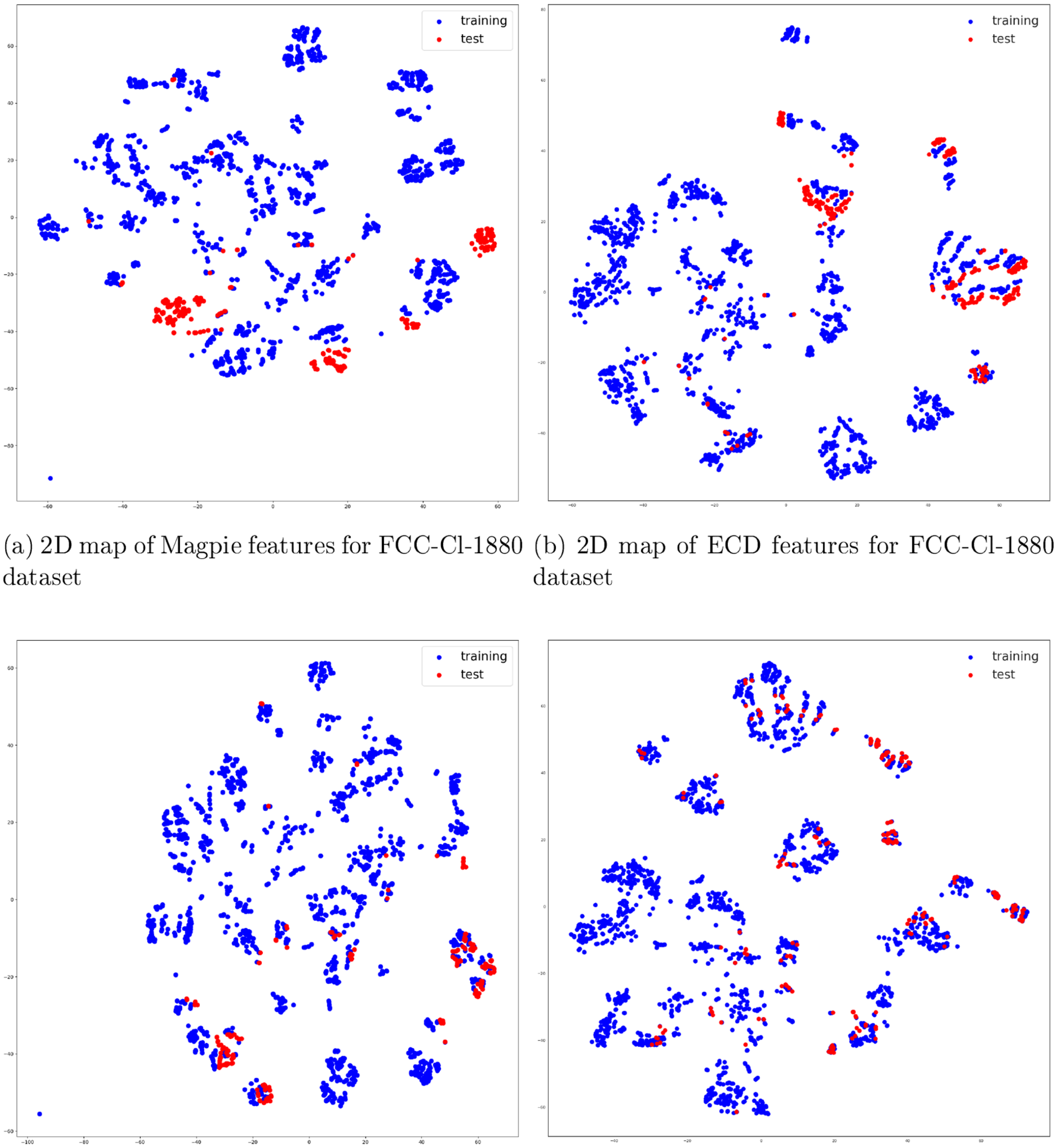
| | | RF+Magpie | | CNN+Magpie | | 3D-CNN+ECD | | 2D-CNN+ECD | | CGCNN | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| elem | type | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE | $R^2$ | RMSE |
| F | bulk | −0.529 | 26.797 | −0.809 | 29.102 | −0.051 | 22.212 | −0.448 | 26.080 | −2.217 | 35.554 |
| | shear | −3.350 | 18.117 | −6.912 | 24.315 | −1.202 | 12.878 | −1.293 | 13.151 | −0.548 | 10.657 |
| K | bulk | 0.776 | 6.067 | 0.646 | 7.573 | 0.510 | 8.969 | 0.570 | 8.397 | 0.474 | 9.055 |
| | shear | 0.810 | 2.641 | 0.548 | 4.014 | 0.389 | 4.733 | 0.367 | 4.817 | 0.146 | 5.523 |
| Rb | bulk | 0.867 | 4.603 | 0.869 | 4.579 | 0.753 | 6.287 | 0.777 | 5.966 | 0.275 | 10.290 |
| | shear | 0.778 | 2.767 | 0.727 | 3.064 | 0.608 | 3.657 | 0.719 | 3.111 | 0.268 | 4.944 |
| Cs | bulk | −0.128 | 11.232 | 0.760 | 5.166 | 0.448 | 7.818 | 0.067 | 10.158 | −0.144 | 10.934 |
| | shear | −4.327 | 11.199 | 0.492 | 3.446 | 0.014 | 4.743 | −1.137 | 7.083 | 0.344 | 3.881 |
| Na | bulk | 0.630 | 16.398 | 0.833 | 11.013 | 0.660 | 15.708 | 0.616 | 16.689 | 0.605 | 16.223 |
| | shear | 0.545 | 8.366 | 0.386 | 9.716 | 0.548 | 8.340 | 0.451 | 9.196 | 0.351 | 9.863 |
| Cl | bulk | 0.410 | 15.935 | 0.529 | 14.151 | 0.591 | 13.009 | 0.716 | 11.05 | 0.534 | 13.119 |
| | shear | −0.477 | 10.715 | 0.213 | 7.765 | 0.339 | 7.160 | 0.093 | 8.394 | −0.197 | 9.366 |
| In | bulk | 0.829 | 20.780 | 0.780 | 23.550 | 0.725 | 26.326 | 0.773 | 23.908 | 0.761 | 24.460 |
| | shear | 0.791 | 8.250 | 0.771 | 8.618 | 0.683 | 10.136 | 0.793 | 8.207 | 0.655 | 10.416 |
| Br | bulk | 0.921 | 4.464 | 0.923 | 4.585 | 0.912 | 4.700 | 0.923 | 4.411 | 0.631 | 9.245 |
| | shear | 0.630 | 2.290 | −0.078 | 3.857 | 0.755 | 1.861 | 0.824 | 1.579 | −2.661 | 6.975 |
| Li | bulk | 0.418 | 29.869 | 0.867 | 14.253 | 0.519 | 27.142 | 0.454 | 28.937 | 0.732 | 20.121 |
| | shear | −0.232 | 17.799 | 0.416 | 12.239 | 0.428 | 12.126 | 0.451 | 11.881 | 0.388 | 12.488 |
| Sc | bulk | 0.855 | 23.276 | 0.908 | 18.538 | 0.756 | 30.195 | 0.850 | 23.688 | 0.818 | 25.983 |
| | shear | 0.781 | 12.996 | 0.707 | 15.024 | 0.682 | 15.650 | 0.635 | 16.786 | 0.667 | 16.007 |
| Tl | bulk | −0.370 | 24.574 | 0.421 | 15.973 | 0.219 | 18.529 | 0.501 | 14.833 | 0.550 | 14.040 |
| | shear | 0.456 | 6.745 | 0.437 | 6.815 | 0.559 | 6.068 | 0.557 | 6.084 | 0.427 | 6.818 |
| No. of the best | | 5 | | 6 | | 4 | | 5 | | 2 | |

CNN model with ECD. Then we fused these two models by concatenating the outputs of the penultimate layers of these two models to generate an output latent feature vector of dimension 64, which is then fed to three fully connected layers with 128, 64, and 32 neurons, respectively. The Adam optimizer[61] is used for training with a learning rate of 0.001. This fusion neural network model with mixed Magpie and ECD descriptor yielded the best $R^2$ and RMSE of 0.955 (0.804) and 16.530 (15.780) in predicting bulk (shear) modulus, respectively, as shown in Table 2. This confirms that ECD and Magpie can work together to achieve better performance for elastic property prediction. In addition, our experiments also showed that the projected 2D CNN achieved significantly better performance than the basic 3D CNN models. The $R^2$ and RMSE of 2D CNN with ECD are 0.912 and 23.401 in predicting the bulk modulus compared to 0.884 and 26.819 of 3D CNNs with ECD. The $R^2$ and RMSE of 2D CNN with ECD are 0.768 and 17.192 in predicting shear modulus compared to 0.745 and 17.944 of 3D CNNs with ECD.

**Extrapolation Experiments with Nonredundant Data Sets.** ML models with elemental descriptors such as Magpie can achieve good cross-validation performance for data sets consisting of redundant (computationally very similar samples) such as FCC2170. However, the better performance of the fusion model with CNN with Magpie and 2D CNN with ECD implies that the ECD descriptor can help to make better predictions over a certain subset of test samples. In this section, we aim to construct a nonredundant data set and show that our CNN models with the ECD descriptor can achieve better performance on nonredundant data sets or for test samples with few highly similar neighbor samples.

For these extrapolation experiments, we trained and tested the prediction models over the FCC-E-N data sets as described in the Data Sets Section. The performance comparison results of the extrapolation experiments for bulk and shear modulus prediction are shown in Table 3. There are 22 sets of experiments with 11 of them for predicting bulk modulus and the other 11 for predicting shear modulus by five different algorithms including RF+Magpie, CNN+Magpie, 3D-CNN +ECD, 2D-CNN+ECD, and the latest crystal graph convolutional neural network (CGCNN),[27] which also uses structural information. We highlighted the best performance scores for each experiment and counted how many experiments each algorithm achieved the best scores. As shown in Table 3, the RF with Magpie and CNN with Magpie worked the best for 5 and 6 experiments, respectively. However, impressively, for these nonredundant training/testing experiments, our ECD descriptor-based 3D-CNN-ECD and 2D-CNN-ECD outperformed the others for 4 and 5 experiments, respectively, which reflects the importance of the structure-based ECD descriptor for elastic property prediction. In contrast, the popular CGCNN only achieved the best performance out of 2 experiments, which demonstrated the advantage of our ECD-based atomic structure representation.

**Visualization Study of when ECD Descriptor Works Better.** To understand why our ECD-based CNN models worked better than Magpie features on some data sets but not others, we conducted a visualization study for all the extrapolation experiments. For magpie features, we directly apply the t-distributed Stochastic Neighbor Embedding (t-SNE)[62] to the data set. For the ECD-based features, directly applying t-SNE is not feasible due to the memory limit. So we first applied max-pooling to the 3D ECD matrices with strides of (6, 6, 6) and pooling size of (6, 6, 6) before feeding them into t-SNE. Hence the final size of the ECD matrices is (10, 10, 10), which are then flattened to a 1D vector of 1000 elements. Then we applied the t-SNE to this 1D vector to reduce the dimension to 2.



(a) 2D map of Magpie features for FCC-Cl-1880 dataset

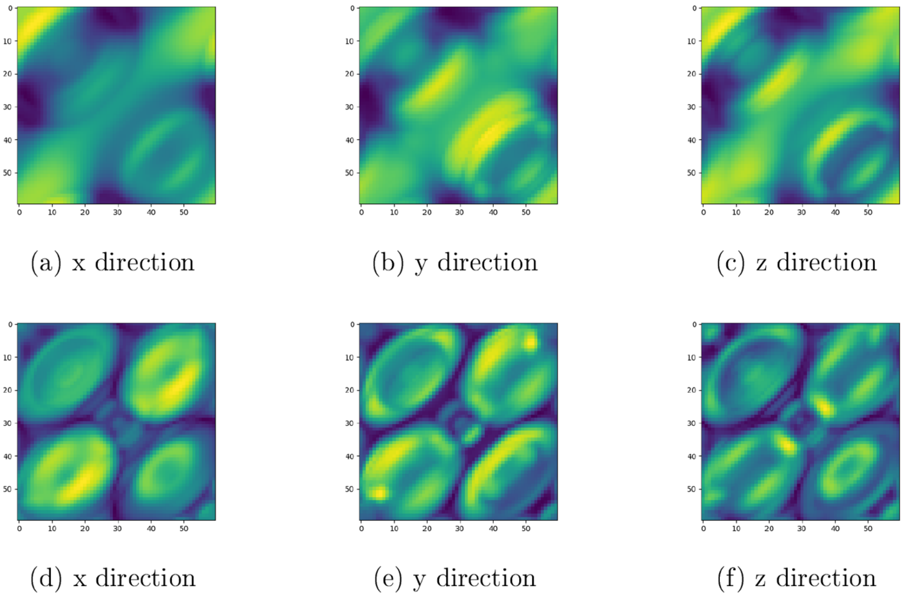
(b) 2D map of ECD features for FCC-Cl-1880 dataset

(c) 2D map of Magpie features for FCC-Tl-1966 dataset

(d) 2D map of ECD features for FCC-Tl-1966 dataset

**Figure 4.** Visualization of high-dimensional features for elements chlorine and thallium by t-SNE. Blue dots are training data, and red dots are test data. (a) In data set FCC-Cl-1880, materials of the training set (blue dots) are separate with materials in the test set (red dots), which makes it difficult for ML models with Mapgpie features to achieve good performance due to their low extrapolation capability. (b) The training samples and test samples are mixed together with the learned representation from ECD, which makes it easy for ML models to achieve good performance. (c) The training samples and test samples are mixed better than in (a), which leads to better ML prediction performance. (d) The training samples and test samples are mixed better than in (c), which allows the ML to get even better prediction performance.

Figure 4 shows 2D visualization of the high-dimension Magpie and ECD features for two data sets: FCC-Chlorine-1880 and FCC-Thallium-1966, over which the ECD-based models outperform Magpie feature-based models. The training samples are labeled as blue points, while the test samples are red points. First, Figure 4a,b shows the distribution of training and test samples with Magpie features and with ECD features, respectively, for the FCC-Chlorine-1880 data set. In Figure 4a, we found that there exist three large clusters of test samples (red points) that have few similar training samples around. This corresponds to the low prediction performance for Magpie-based models. The best performance for both bulk and shear modulus prediction is achieved by CNN+Magpie with $R^2$ values of 0.529 and 0.213, respectively. In contrast, Figure 4b



(a) x direction (b) y direction (c) z direction

(d) x direction (e) y direction (f) z direction

**Figure 5.** Visualization of average output of 24 channels of the SE block for three directions for $SrCaIn_2$ and $K_3YI_6$. From left to right are *x*, *y*, and *z* directions, respectively. $SrCaIn_2$ (top row) and $K_3YI_6$ (bottom row) are materials with high and low bulk modulus, respectively.

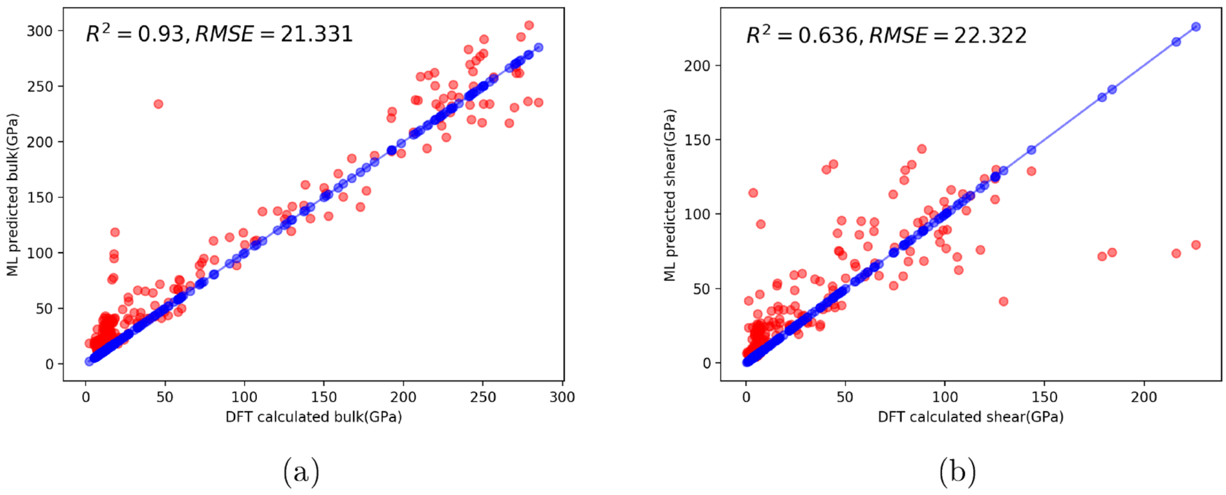


**Figure 6.** (a, b) ML-predicted vs DFT-calculated bulk and shear modulus, respectively.

shows the 2D distribution of the samples represented with ECD features. It can be found that the test samples are mostly mixed with training samples, leading to much better prediction performance: the best performance for bulk modulus prediction is achieved by 2D-CNN+ECD with $R^2$ of 0.716, which is significantly better (35%) than 0.529, the best prediction performance achieved by Magpie-based models. The best performance for shear prediction is achieved by 3D-CNN+ECD with $R^2$ of 0.339, which is also 59% better than 0.213, the best $R^2$ score of Magpie-based models.

Figure 4c,d shows the distribution of training and test samples with Magpie features and with ECD features, respectively, for the FCC-Thallium-1966 data set. In Figure 4c, we found that clusters of test samples (red points) are closer to training samples compared to Figure 4a. There is no large cluster of isolated test samples. The best performance for bulk modulus is achieved by CNN+Magpie with $R^2$ of 0.421. The best performance for shear modulus prediction is achieved by RF+Magpie with $R^2$ of 0.456. In contrast, Figure 4d shows the 2D distribution of the samples represented with ECD features. It can be found that the test samples are better mixed with training samples than those of Figure 4a, leading to better prediction performance. The best performance for both bulk modulus prediction is achieved by 2D-CNN+ECD with $R^2$ of 0.501, and the best shear modulus prediction performance is achieved by 3D-CNN+ECD with $R^2$ of 0.559. In this data set, the best ECD-based model is $(0.559 - 0.421)/0.421 = 19\%$ better than the best Magpie-based model for bulk modulus prediction. The performance gap is much smaller compared to that (35%) of the FCC-Chlorine-1880 data set. The best ECD-based model is also $(0.559 - 0.456)/0.456 = 24.9\%$ better than the best Magpie-based model for shear modulus prediction, which is however much smaller than the performance gap over the FCC-Chlorine-1880 data set, which is 59%. These findings can partially explain why ECD-based models are superior to Magpie-based models in predicting elastic properties for these two data sets. It shows the structure-based ECD descriptor can be a complementary descriptor to elemental Magpie features for elastic property prediction due to their better neighborhood structure of the samples. This analysis is consistent with the observation that neighbor sample distribution significantly affects the performance of neural network-based prediction models.[63]

**Visualization of Averaged SE Block Outputs.** Figure 5 shows the visualization of the average output of the 24 channels of the SE block as shown in Figure 3 from the *x*, *y*, and *z* directions. On the top row, bright and dark areas/patterns overlap together in the $SrCaIn_2$ with high bulk modulus with fuzzy boundaries. However, clear boundaries (four clear ovals in each direction) between dark and bright regions can be found on the bottom row, which are the patterns for the material with low bulk modulus. These findings are consistent with the patterns, as we discussed in the ECD representation subsection. This distinct pattern extracted by 2D CNNs helps to differentiate materials and effectively predict their elastic properties. Moreover, although visualizations for three directions share many similar patterns for the same materials, there are variations among them. For example, the darkness of Figure 5d−f is different. Among them, overall Figure 5e has the darkest area, and Figure 5d has the brightest ones. We believe that the slight variations detected by the 2D CNNs might be one of the reasons that 2D CNNs outperform 3D CNNs in predicting elastic properties.

**DFT Validation.** To further validate our neural network models, we predict the bulk and shear modulus of a set of external materials from the OQMD[64] database and compare them to DFT-calculated ones. We first collect all the materials of the space group $Fm\bar{3}m$ from OQMD and then remove the duplicates existing in the Material Project database that we used as the training set. We also filter out the materials having more than 40 atoms in the unit cell. We finally obtain 329 materials as our test set. Then we apply the trained fusion model (Magpie + ECD features) trained with Material Project samples to predict the bulk and shear modulus of the 329 samples in the test set and compare them with DFT-calculated ones as shown in Figure 6. We find that our fusion model successfully predicted the bulk modulus for the 329 materials with good alignment with DFT-calculated values. The $R^2$ and RMSE in predicting bulk are 0.93 and 21.331 as shown in Figure 6a. However, we also find that the ML-predicted shear modulus values deviate much more from the DFT-calculated ones compared to the bulk modulus, which reflects the fact that it is more difficult to predict shear modulus than bulk modulus. We also observe that most of the deviations of the predicted values compared with DFT-calculated ones are from the regions with low bulk or shear modulus and that the predicted values usually are larger than the DFT-calculated ones.

## CONCLUSIONS

We propose to combine deep convolutional neural networks and electronic charge density for materials elasticity prediction. We demonstrate that the ECD descriptor can be used to predict bulk and shear modulus with the CNNs model. We created a benchmark data set named FCC2170 with 2170 materials of the $Fm\bar{3}m$ space group from the Materials Project database and derived 11 nonredundant leave-one-element-out data sets for benchmarking the proposed ML models with ECD and elemental Magpie features. Our computational experiments showed that, due to the structural similarity among the samples of the FCC2170 data set, the elemental Magpie feature with CNN models achieved the best results, which however, can be enhanced by the fusion models with both Magpie and ECD features. In addition, our benchmark studies on the nonredundant data sets showed that the structure-based ECD feature with CNNs can achieve better extrapolation prediction performances over half prediction tasks out of the total 22 experiments for prediction bulk and shear modulus.

To further understand the power of the ECD descriptor, we visualized the distribution of training and test data sets of two descriptor types using t-SNE. It shows that, when the training set and testing set of the nonredundant data sets have a higher level of mixing, the Magpie-based CNN models work better. When they have a lower level of mixing, the ECD descriptor-based models significantly outperform the Magpie-based CNN models. The results demonstrate the importance of structure-based features for achieving higher extrapolation and generalization prediction capability. It is expected that our ECD descriptor with CNN models can also be applied to a variety of problems in material science, especially with the development of algorithms for predicting ECD.[33] Currently, we are generating more ECD data sets with more space groups to extend this method to more materials with diverse structures.

## AUTHOR INFORMATION

### Corresponding Authors

**Ming Hu** − *Department of Mechanical Engineering, University of South Carolina, Columbia 29208, South Carolina, United States;* orcid.org/0000-0002-8209-0139; Phone: +1 (803) 576-7206; +1 (803) 777-7304; Email: hu@sc.edu; Fax: +1 (803) 777-3767

**Jianjun Hu** − *Department of Computer Science and Engineering, University of South Carolina, Columbia 29208, South Carolina, United States; School of Mechanical Engineering, Guizhou University, Guiyang 550025, China;* orcid.org/0000-0002-8725-6660; Email: jianjunh@cse.sc.edu

### Authors

**Yong Zhao** − *Department of Computer Science and Engineering, University of South Carolina, Columbia 29208, South Carolina, United States;* orcid.org/0000-0002-6762-266X

**Kunpeng Yuan** − *Department of Mechanical Engineering, University of South Carolina, Columbia 29208, South Carolina, United States; Key Laboratory of Ocean Energy Utilization and Energy Conservation of Ministry of Education School of Energy and Power Engineering, Dalian University of Technology, Dalian 116024, China*

**Yinqiao Liu** − *Department of Mechanical Engineering, University of South Carolina, Columbia 29208, South Carolina, United States; Key Laboratory of Ocean Energy Utilization and Energy Conservation of Ministry of Education School of Energy and Power Engineering, Dalian University of Technology, Dalian 116024, China*

**Steph-Yves Louis** − *Department of Computer Science and Engineering, University of South Carolina, Columbia 29208, South Carolina, United States*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jpcc.0c02348

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

Research reported in this work was supported in part by the National Science Foundation (NSF) under Grant Nos.

1940099 and 1905775 and by the NSF SC EPSCoR Program via Grant No. OIA-1655740 (GEAR-CRP 19-GC02). The views, perspective, and content do not necessarily represent the official views of the SC EPSCoR Program nor those of the NSF. This work was also partially supported by the Department of Energy under Grant No. DE-SC0020272.

## ■ REFERENCES

(1) Cao, Z.; Dan, Y.; Xiong, Z.; Niu, C.; Li, X.; Qian, S.; Hu, J. *Crystals* **2019**, *9*, 191.

(2) Olsthoorn, B.; Geilhufe, R. M.; Borysov, S. S.; Balatsky, A. V. *Adv. Quantum Technol.* **2019**, *2*, 1900023.

(3) Sendek, A. D.; Cubuk, E. D.; Antoniuk, E. R.; Cheon, G.; Cui, Y.; Reed, E. J. *Chem. Mater.* **2019**, *31*, 342−352.

(4) Avery, P.; Wang, X.; Oses, C.; Gossett, E.; Proserpio, D. M.; Toher, C.; Curtarolo, S.; Zurek, E. *npj Comput. Mater.* **2019**, *5*, 1−11.

(5) Cubuk, E. D.; Sendek, A. D.; Reed, E. J. *J. Chem. Phys.* **2019**, *150*, 214701.

(6) Chen, L.; Tran, H.; Batra, R.; Kim, C.; Ramprasad, R. *Comput. Mater. Sci.* **2019**, *170*, 109155.

(7) Cecen, A.; Dai, H.; Yabansu, Y. C.; Kalidindi, S. R.; Song, L. *Acta Mater.* **2018**, *146*, 76−84.

(8) Ward, L.; Agrawal, A.; Choudhary, A.; Wolverton, C. *npj Comput. Mater.* **2016**, *2*, 16028.

(9) Kajita, S.; Ohba, N.; Jinnouchi, R.; Asahi, R. *Sci. Rep.* **2017**, *7*, 16991.

(10) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754−13769.

(11) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(12) Villars, P.; Cenzual, K.; Daams, J.; Chen, Y.; Iwata, S. J. *J. Alloys Compd.* **2004**, *367*, 167−175.

(13) Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. *Int. J. Quantum Chem.* **2015**, *115*, 1094−1101.

(14) Rupp, M. *Int. J. Quantum Chem.* **2015**, *115*, 1058−1073.

(15) Bartók, A. P.; Kondor, R.; Csányi, G. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2013**, *87*, 184115.

(16) Szlachta, W. J.; Bartók, A. P.; Csányi, G. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *90*, 104108.

(17) Ghiringhelli, L. M.; Vybiral, J.; Levchenko, S. V.; Draxl, C.; Scheffler, M. *Phys. Rev. Lett.* **2015**, *114*, 105503.

(18) Meredig, B.; Agrawal, A.; Kirklin, S.; Saal, J. E.; Doak, J.; Thompson, A.; Zhang, K.; Choudhary, A.; Wolverton, C. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2014**, *89*, 094104.

(19) Lam Pham, T.; Kino, H.; Terakura, K.; Miyake, T.; Tsuda, K.; Takigawa, I.; Chi Dam, H. *Sci. Technol. Adv. Mater.* **2017**, *18*, 756.

(20) Liu, Y.; Zhao, T.; Ju, W.; Shi, S. *Journal of Materiomics* **2017**, 3, 159−177.

(21) Ward, L.; Dunn, A.; Faghaninia, A.; Zimmermann, N. E.; Bajaj, S.; Wang, Q.; Montoya, J.; Chen, J.; Bystrom, K.; Dylla, M.; et al. *Comput. Mater. Sci.* **2018**, *152*, 60−69.

(22) Wei, H.; Zhao, S.; Rong, Q.; Bao, H. *Int. J. Heat Mass Transfer* **2018**, *127*, 908−916.

(23) Zhuo, Y.; Mansouri Tehrani, A.; Brgoch, J. *J. Phys. Chem. Lett.* **2018**, *9*, 1668−1673.

(24) Jha, D.; Choudhary, K.; Tavazza, F.; Liao, W.-k.; Choudhary, A.; Campbell, C.; Agrawal, A. *Nat. Commun.* **2019**, *10*, 1−12.

(25) Noh, J.; Kim, J.; Stein, H. S.; Sanchez-Lengeling, B.; Gregoire, J. M.; Aspuru-Guzik, A.; Jung, Y. *Matter* **2019**, *1*, 1370.

(26) Willatt, M. J.; Musil, F.; Ceriotti, M. *J. Chem. Phys.* **2019**, *150*, 154110.

(27) Xie, T.; Grossman, J. C. *Phys. Rev. Lett.* **2018**, *120*, 145301.

(28) Chen, C.; Ye, W.; Zuo, Y.; Zheng, C.; Ong, S. P. *Chem. Mater.* **2019**, *31*, 3564−3572.

(29) Henkelman, G.; Arnaldsson, A.; Jónsson, H. *Comput. Mater. Sci.* **2006**, *36*, 354−360.

(30) Ouyang, T.; Hu, M. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2015**, *92*, 235204.

(31) Qin, G.; Qin, Z.; Yue, S.-Y.; Yan, Q.-B.; Hu, M. *Nanoscale* **2017**, *9*, 7227−7234.

(32) Qin, G.; Qin, Z.; Wang, H.; Hu, M. *Nano Energy* **2018**, *50*, 425−430.

(33) Gong, S.; Xie, T.; Zhu, T.; Wang, S.; Fadel, E. R.; Li, Y.; Grossman, J. C. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2019**, *100*, 184103.

(34) Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. *Sci. Rep.* **2013**, *3*, 2810.

(35) Abraham, J. A.; Pagare, G.; Sanyal, S. P. *Indian J. Mater. Sci.* **2015**, *2015*, 1.

(36) Choi, H.; Sohn, K.-S.; Pyo, M.; Chung, K.-C.; Park, H. *J. Phys. Chem. C* **2019**, *123*, 4682−4690.

(37) Rong, Q.; Wei, H.; Huang, X.; Bao, H. *Compos. Sci. Technol.* **2019**, *184*, 107861.

(38) Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D Object Detection Network for Autonomous Driving. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Honolulu, Hawaii, July 21−26, 2016; IEEE Computer Society, 2017; pp 1907−1915 DOI: 10.1109/CVPR.2017.691.

(39) Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition*arXiv(Computer Vision and Pattern Recognition)*, 1409.1556, ver. 6. https://arxiv.org/abs/1409.1556

(40) Maturana, D.; Scherer, S. Voxnet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Hamburg, Germany, September 28−October 2, 2015; IEEE, 2015; pp 922−928 DOI: 10.1109/IROS.2015.7353481.

(41) Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. A. Inception-v4, inception-ResNet, and the Impact of Residual Connections on Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence,* AAAI, 2017.

(42) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NeurIPS* **2012**, 1097−1105.

(43) Artmann, K. *Ann. Phys.* **1948**, *437*, 87−102.

(44) Wu, Z.-j.; Zhao, E.-j.; Xiang, H.-p.; Hao, X.-f.; Liu, X.-j.; Meng, J. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2007**, *76*, 054115.

(45) Kresse, G.; Hafner, J. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1993**, *47*, 558.

(46) Kresse, G.; Furthmüller, J. *Comput. Mater. Sci.* **1996**, *6*, 15−50.

(47) Kresse, G.; Furthmüller, J. *Phys. Rev. B: Condens. Matter Mater. Phys.* **1996**, *54*, 11169.

(48) Xiong, Z.; Cui, Y.; Liu, Z.; Zhao, Y.; Hu, M.; Hu, J. *Comput. Mater. Sci.* **2020**, *171*, 109203.

(49) Kim, Y.; Kim, E.; Antono, E.; Meredig, B.; Ling, J. Machine-learned metrics for predicting the likelihood of success in materials discovery*arXiv(cond-mat.mtrl.sci)*, 1911.11201, ver. 2, 2019. https://arxiv.org/abs/1911.11201

(50) Meredig, B.; Antono, E.; Church, C.; Hutchinson, M.; Ling, J.; Paradiso, S.; Blaiszik, B.; Foster, I.; Gibbons, B.; Hattrick-Simpers, J.; et al. *Mol. Syst. Des. Eng.* **2018**, 3, 819−825.

(51) Silvi, B.; Savin, A. *Nature* **1994**, *371*, 683.

(52) Callister, W. D.; Rethwisch, D. G. *Materials Science and Engineering: An Introduction;* John Wiley & Sons: New York, 2007; Vol. 7.

(53) Breiman, L. *Machine learning* **2001**, *45*, 5−32.

(54) Takahashi, K.; Takahashi, L. J. *J. Phys. Chem. Lett.* **2019**, *10*, 283−288.

(55) Oviedo, F.; Ren, Z.; Sun, S.; Settens, C.; Liu, Z.; Hartono, N. T. P.; Savitha, R.; DeCost, B. L.; Tian, S. I.; Romano, G.; et al. Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks*arXiv(physics.data-an)*, 1811.08425, ver. 2. https://arxiv.org/abs/1811.08425

(56) Ward, L.; Liu, R.; Krishna, A.; Hegde, V. I.; Agrawal, A.; Choudhary, A.; Wolverton, C. *Phys. Rev. B: Condens. Matter Mater. Phys.* **2017**, *96*, 024104.

(57) Kim, Y. Convolutional Neural Networks for Sentence Classification*arXiv(cs.CL)*, 1408.5882, ver. 2. https://arxiv.org/abs/1408.5882

(58) LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. *Proc. IEEE* **1998**, *86*, 2278−2324.

(59) Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* Salt Lake City, UT, June 18−23, 2018; IEEE, 2018; pp 7132−7141 DOI: 10.1109/CVPR.2018.00745.

(60) Lin, M.; Chen, Q.; Yan, S. Network In Network*arXiv(cs.NE)*, 1312.4400, ver. 3. https://arxiv.org/abs/1312.4400

(61) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization*arXiv(cs.LG)*, 1412.6980, ver. 9. https://arxiv.org/abs/1412.6980

(62) Maaten, L. v. d.; Hinton, G. J. Visualizing Data using t-SNE. *Mach. Learn. Res.* **2008**, *9*, 2579−2605.

(63) Janet, J. P.; Duan, C.; Yang, T.; Nandy, A.; Kulik, H. J. *Chem. Sci.* **2019**, *10*, 7913.

(64) Saal, J. E.; Kirklin, S.; Aykol, M.; Meredig, B.; Wolverton, C. *JOM* **2013**, *65*, 1501−1509.