Predicting charge density distribution of materials using a local-environment-based graph convolutional network

Sheng Gong,[1] Tian Xie,[1] Taishan Zhu,[1] Shuo Wang,[2] Eric R. Fadel,[1] Yawei Li,[3] and Jeffrey C. Grossman[1,*]

[1]*Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*

[2]*Department of Materials Science and Engineering, University of Maryland, College Park, Maryland 20742, USA*

[3]*Department of Chemical Engineering, Pennsylvania State University, University Park, Pennsylvania 16802, USA*

(Received 24 July 2019; published 7 November 2019)

The electron charge density distribution of materials is one of the key quantities in computational materials science as theoretically it determines the ground state energy and practically it is used in many materials analyses. However, the scaling of density functional theory calculations with number of atoms limits the usage of charge-density-based calculations and analyses. Here we introduce a machine-learning scheme with local-environment-based graphs and graph convolutional neural networks to predict charge density on grid points from the crystal structure. We show the accuracy of this scheme through a comparison of predicted charge densities as well as properties derived from the charge density, and that the scaling is $O(N)$. More importantly, the transferability is shown to be high with respect to different compositions and structures, which results from the explicit encoding of geometry.

DOI: 10.1103/PhysRevB.100.184103

I. INTRODUCTION

The electron charge density distribution is of enormous importance to the computational understanding and design of materials, as many fundamental properties relevant to a wide range of applications are directly related to the magnitude, shape, and variation of the charge density. The charge density and its related properties, such as the electrostatic potential [1], electron localization function [2], and noncovalent interaction index [3], are directly used in analyses for many materials characteristics, including bonding [4], defects [5], stability [6], reactivity [7], and electron [8,9], ion [10,11], and thermal [12] transport, to name only a few. Recently, with the rapid development of machine-learning (ML) applications in physics [13,14], chemistry [15,16], and materials science [17–19], charge densities have been increasingly used as input features for predicting other materials properties in order to improve performance [20–22]. Currently the most common approach used to calculate charge density is density functional theory (DFT). However, the relatively high computational cost and high memory demands of DFT [23] limit its use for large systems with more than several hundred atoms. Therefore, it is important to develop methods capable of accurately predicting charge density with less computational demand, to "bypass the Kohn-Sham equations" [24], and ML is a promising tool for this goal due to the success of its application in predicting other DFT-computed properties [14,25–28].

In principle, an ideal ML algorithm should meet three requirements: high accuracy, high transferability, and low computational cost [29]. Very recently, there have been attempts [24,30] to employ ML to predict the charge density of molecules by expanding the density as a sum of atom basis functions. For the case of periodic systems, Schmidt *et al.* [31] employed basis functions, summing over the contributions from only neighboring atoms to achieve transferability between different cell sizes and lower memory demands, while Chandrasekaran *et al.* [23] encoded the position of each grid point to neighboring atoms by well-designed invariants to predict charge density. In both of these works the ML schemes were able to generate high-quality charge densities with $O(N)$ scaling, although compositional and structural transferability remains a challenge, as these methods account for variations in one structure at a time (i.e., strained lattices or different molecular dynamics snapshots).

Here, we develop a ML-based approach that can predict charge density for different structures with varying compositions, structural features, and defects for a given class of materials in a single training, which is necessary for application to systems such as amorphous hydrocarbons or glasses where local structures are highly complex. In previous works, a three-step process was followed: (1) record the distance between each grid point and all neighboring atoms, (2) add all distances together to form a feature vector, and (3) compute charge density by regression on the final feature vector. For multielemental systems, the first two steps are repeated for each element type and the feature vectors are concatenated together. In order to build upon this approach with increasing transferability between different structures, in addition to recording the distance between grid points and atoms, we propose to both explicitly encode the geometry of the cluster formed by neighboring atoms and account for all elements simultaneously, as opposed to separately. Encoding the geometry, on the one hand, avoids the problem of different local environments leading to a similar sum of atom contributions; on the other hand, it enables the model to learn from the geometry of existing structural features and speculate on new ones. Greater structural transferability should also

*Corresponding author: jcg@mit.edu

lead to improved accuracy in the prediction of charge density for defect structures, as new structural features can form during the formation of defects. To accommodate different elements, the dimension of the final feature vector should be independent of composition; otherwise the regression process (matrix-vector multiplication) cannot be done.

A graph representation, encoding both nodes and bonds, has a number of advantages that meet the requirements listed above. Graph representations have been used recently to encode information on both the level of atom and geometry with high accuracy and transferability across composition, structure, and property space [14,17], and the feature vectors can be of the same dimension for different compositions if properly designed. In this work, we encode environments of grid points as graphs and employ the crystal graph convolution neural network [14] (CGCNN) to find a relationship between local environment and charge density with $O(N)$ scaling. We train and test our scheme on two classes of crystalline materials, polymers and zeolites. For each case training data is from some structures and the model is applied to others to test transferability, and the accuracy of the predicted charge density is evaluated through statistics, visualization, and accuracy of its derivative and related properties (i.e., dipole moment). The most important contribution of the present work is to demonstrate that by using graphs to encode local environment, the model achieves high transferability to unseen structures.

II. METHOD

We encode three-dimensional space in the unit cell using CGCNN by placing an imaginary atom at each grid point in the unit cell (Fig. 1). The local environment is computed for a given grid point by identifying atoms within a cutoff radius (R_{cut}) from the imaginary atom, as shown in Fig. 1(b). Next as shown in Fig. 1(c), atoms outside R_{cut} are removed, and the remaining structure is placed in a larger cell to avoid interactions between periodic images. Here R_{cut} is 4 Å, larger than typical bond lengths for the materials considered in this work [32], and the lattice parameters of the larger cell are set to be no less than $3 \times R_{\text{cut}}$. Finally, the remaining structure together with the imaginary atom are converted into a graph representation as shown in Fig. 1(d) by connecting neighbors. The CGCNN is then trained on the local-environment-based graphs with the charge density on the grid points from DFT calculations as the target property (with units of $e/\text{Å}^3$). Details of the DFT calculations and representations of the imaginary atom are given in the Supplemental Material [33] (see also Refs. [34–37] therein).

The neural network structure is summarized in Fig. 1(d). Once given a graph, the convolutional layers iteratively update the atom feature vector v_i based on surrounding atoms and bonds with a convolution function:

$$v_i^{(t+1)} = \text{Conv}\left(v_i^{(t)}, v_j^{(t)}, \mu_{ij}\right), \tag{1}$$

where $v_{i(j)}^{(t)}$ is the atom feature vector of the $i(j)$th atom after t convolutions, μ_{ij} represents the bond vector between the ith and jth atoms, and "Conv" stands for the convolution function. Here the convolution function designed in Ref. [14] is used, which was shown to be accurate for encoding interaction strengths and produces feature vectors with constant dimension for different compositions. A pooling function is then used to create an overall feature vector to satisfy permutational and size invariance:

$$v = \text{Pool}\left(v_0^{(0)}, \ldots, v_0^{(T)}, \ldots, v_N^{(T)}\right). \tag{2}$$

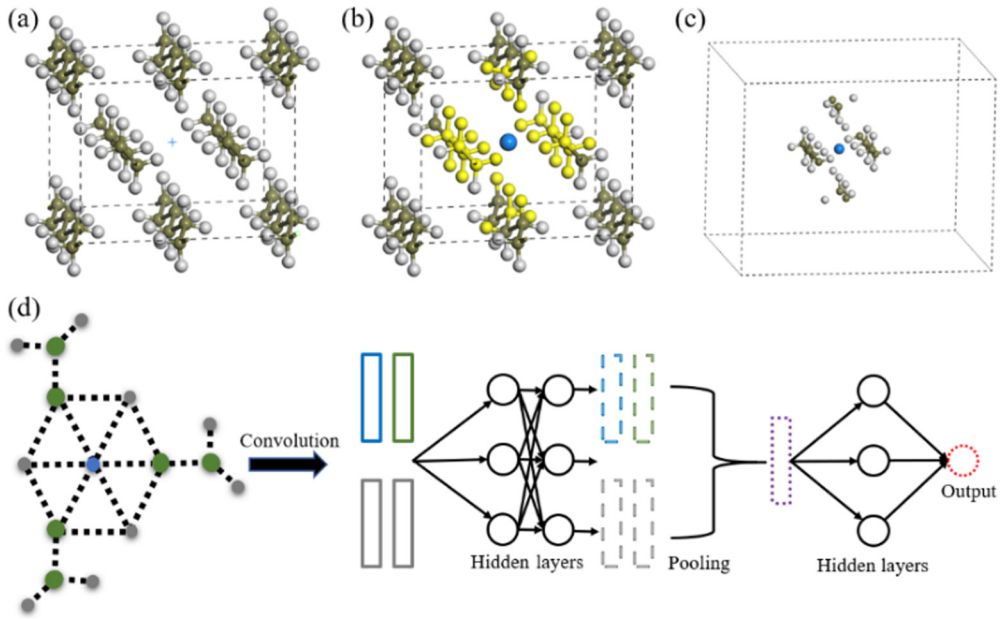

FIG. 1. (a) Crystal structure of crystalline ethylene. The blue plus symbol in the center denotes a grid point we are interested in. (b) Crystalline ethylene with the imaginary atom. Highlighted atoms are those within the cutoff radius. (c) Local environment around the imaginary atom. (d) Sketch of local-environment-based graph and CGCNN architecture. Color coding: Green, carbon; gray, hydrogen; blue, imaginary atom; yellow, highlighted atoms within the cutoff radius.

Here, the mean of atom vectors is taken as the feature after pooling for simplicity, while other pooling functions can also be used.

In addition, two hidden layers are used to capture the complex relationship between structure and property, and finally an output layer is used to give the target property. This process meets both of the requirements as mentioned above, since after convolution the atom feature vector for the imaginary atom encodes the distances between one grid point and neighboring lattice atoms, while that for lattice atoms encodes their position with respect to not only other lattice atoms but also the imaginary atom. The pooling process incorporates all the information together and makes the final feature vector of the same dimension for materials with different compositions.

III. RESULTS

In the case of crystalline polymers, we extract 30 000 graphs (grid points) from 37 different structures as training data, while in the case of zeolites 8000 graphs are generated from 5 different structures for training. The list of structures from which training data are obtained is provided in Table S2 of the Supplemental Material [33]. Further details related to data set construction and grid spacing are also provided in the Supplemental Material [33].

In Fig. 2(a), we show how the prediction performance changes as a function of the training size. The straight-line-like trends in Fig. 2(a) indicate that better performance is possible with larger training sets. In addition, the steeper slope for the case of zeolites indicates their reduced chemical complexity compared to the polymers as discussed below. As for the computational cost, although direct comparison between computation time of DFT and ML is difficult as they are based on different computing architectures, in Fig. 2(b) the relation between computational time and number of atoms in the system is plotted for prediction of the charge density of crystalline p-xylylene using our ML model and DFT calculations, from which one can see the linear scaling of the ML approach.

In order to test the degree of transferability toward different structures, we apply our model to predict the charge density of 17 crystalline polymers and 9 zeolites not included in the training sets, as shown in Table I. In both cases, the nomex polymer and NPO zeolite also have versions with explicitly created defect structures (denoted as nomex_defect and NPO_defect) in order to represent additional chemical complexity. These materials are not subsets of the training sets in terms of structure or size. Structural features are represented by coordinations of skeleton atoms (C/O in the case of polymer/zeolite). For example, C2H2 means there are 2 C atoms and 2 H atoms coordinated with the central atom. For polymers, in Fig. 3(a) the frequency of different coordinations for carbon atoms is shown for both the training and test sets, from which one can see that nearly 20 different coordinations appear, showing considerable bonding complexity. More importantly, there are three coordinations in the test set that are not included in the training set (H4, C1H1, and C4). For zeolites, the training set is simpler than the polymer set in terms of structure, as only two coordinations exist, and in the test set only the structure with a defect, NPO_defect, has the coordination of Si1, while all other structures have coordination Si2. From the perspective of size, for polymers, structures in the training set span a range from 8 to 288 atoms in the unit cell, while the structures in the test set span a range from 24 to 504 atoms, and for zeolites the size ranges are 120 to 366 atoms and 18 to 576 atoms for the training and test set, respectively. Further details regarding the chemical complexity based on composition are provided in the Supplemental Material [33] (see also Refs. [38–40] therein).

Here, we choose two metrics, root-mean-square errors (RMSEs) and coefficients of determination (R^2), to quantify errors in the ML-predicted charge density. These metrics, also used in Schmidt *et al.* [31], provide insights on both the magnitude of absolute errors (by RMSEs) and relative performance of the predictions (by R^2). As shown in Table I, the RMSEs of the predicted charge densities are all less than 0.2 $e/\text{Å}^3$, which are comparable to the errors in Ref. [31], and

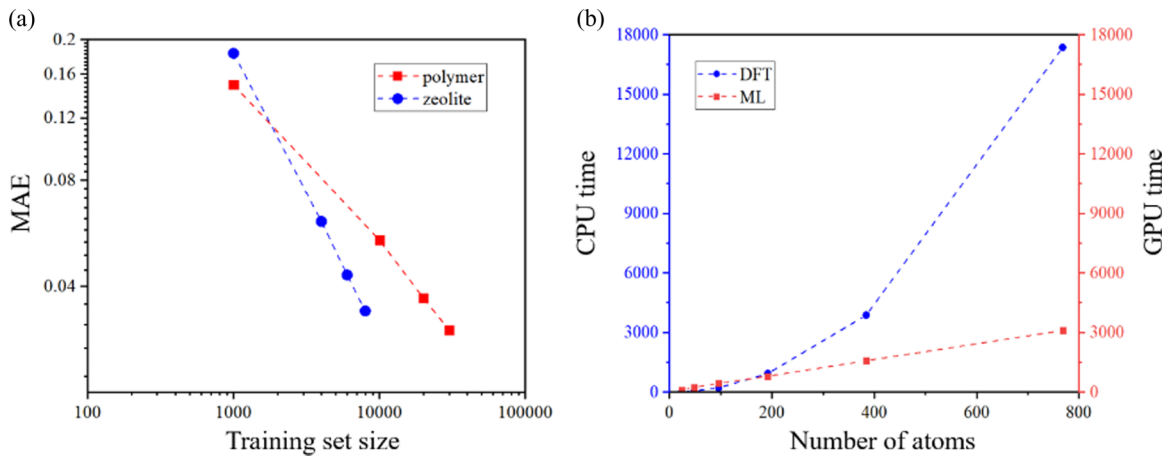

FIG. 2. (a) Mean average error (MAE, in $e/\text{Å}^3$) of the ML-predicted charge density of the test sets (grid points) from the training structures versus training set size for polymer and zeolite materials. (b) CPU time (in seconds) for DFT calculations and GPU time (in seconds) for ML prediction versus number of atoms in the cell for crystalline p-xylylene. DFT calculations are performed by 24 Intel Xeon CPUs with RAM of 128 GB, while ML calculations are carried out on a single NVIDIA GeForce GTX 1070 GPU with RAM of 2 GB.

TABLE I. Root-mean-square errors (RMSEs) and coefficients of determination (R^2) of the ML-predicted charge density (ρ, in $e/\text{Å}^3$) and Laplacian of charge density ($\nabla^2\rho$, in $e/\text{Å}^5$). For each structure, the error metrics are computed over all grid points in the unit cell. The last nine structures with 3-letter abbreviations are zeolites, and others are crystalline polymers.

Name	Formula (inside the cell)	RMSE (ρ)	R^2 (ρ)	RMSE ($\nabla^2\rho$)	R^2 ($\nabla^2\rho$)
1,3-dioxolane-II	$C_{24}H_{48}O_{16}$	0.0628	0.9933	0.4190	0.9934
acetaldehyde	$C_{32}H_{64}O_{16}$	0.0818	0.9848	0.5007	0.9850
cis-1,4-butadiene	$C_{16}H_8$	0.0902	0.9805	0.3502	0.9822
glycolide	$C_8H_8O_8$	0.0681	0.9943	0.4502	0.9941
gutta-percha-alpha	$C_{20}H_{32}$	0.0369	0.9953	0.1998	0.9939
i-4m1p	$C_{168}H_{336}$	0.0666	0.9729	0.4521	0.9656
i-alpha-vnaph	$C_{192}H_{160}$	0.0661	0.9816	0.4311	0.9778
i-ortho-mths	$C_{144}H_{160}$	0.0593	0.9831	0.3678	0.9798
i-propylene-alpha	$C_{36}H_{72}$	0.0491	0.9881	0.2992	0.9852
isobutylene	$C_{64}H_{128}$	0.0910	0.9569	0.6114	0.9541
nomex	$C_{14}H_{10}O_2N_2$	0.0626	0.9926	0.3333	0.9899
nomex_defect	$C_{13}H_9O_2N_2$	0.0665	0.9913	0.3590	0.9882
oxymethylene	$C_4H_8O_4$	0.0786	0.9926	0.4765	0.9906
p-xylylene	$C_{16}H_8$	0.0580	0.9890	0.2735	0.9916
s-propylene-1	$C_{24}H_{12}$	0.0523	0.9835	0.3359	0.9814
tetramtht	$C_{12}H_{12}O_4$	0.0502	0.9960	0.3538	0.9954
trans-decenamer	$C_{10}H_{18}$	0.0309	0.9970	0.3590	0.9882
NPO	Si_6O_{12}	0.0977	0.9893	0.5602	0.9885
NPO_defect	Si_5O_{12}	0.1798	0.9745	0.9289	0.9721
JBW	Si_6O_{12}	0.0847	0.9914	0.5702	0.9887
CAN	$Si_{12}O_{24}$	0.0831	0.9906	0.6014	0.9893
AFY	$Si_{16}O_{32}$	0.0778	0.9894	0.5418	0.9879
JSN	$Si_{16}O_{32}$	0.0785	0.9911	0.5221	0.9901
MTN	$Si_{136}O_{272}$	0.0821	0.9903	0.2809	0.9886
TUN	$Si_{192}O_{384}$	0.0754	0.9920	0.1986	0.9922
UOV	$Si_{176}O_{352}$	0.0912	0.9881	0.2039	0.9876

the level of accuracy was demonstrated to be sufficient for most applications relying on the accuracy of the density representation [41]. The RMSEs of test structures are also close to that of the training sets (0.067 $e/\text{Å}^3$ and 0.064 $e/\text{Å}^3$ for crystalline polymers and zeolites, respectively), indicating little overfitting. More importantly, the R^2 are larger than 0.95 for all test structures, suggesting a high prediction performance. The results for the case of zeolites show that for such a simple materials class, accurate prediction of the charge density can be achieved with a relatively small training set (less than 10 000 training data in this case). In addition to these general trends, we highlight the cases with different coordination environments (i-4m1p, isobutylene, and the nomex_defect). Although larger errors are observed in these cases, they are not far from other structures, suggesting good transferability to unseen structural features.

Next, the Laplacian of the charge density is computed in order to test the ML model's ability to capture variation in

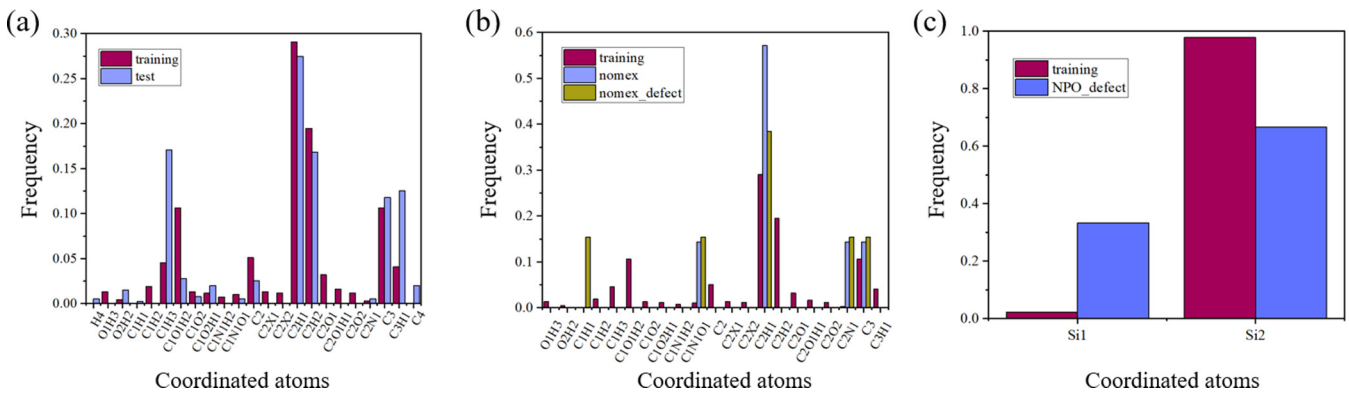

FIG. 3. (a) and (b) Appearance frequency of coordinated atoms of carbon atoms in the training set for the case of crystalline polymers versus the test set as a whole and nomex and nomex_defect, respectively. Here the "*X*"s in "C2X1" and "C2X2" denote rare elements in our case (Cl, F, S, Si, Hg). (c) Appearance frequency of oxygen coordinated atoms in the training set for the case of zeolites versus the structure of NPO_defect.

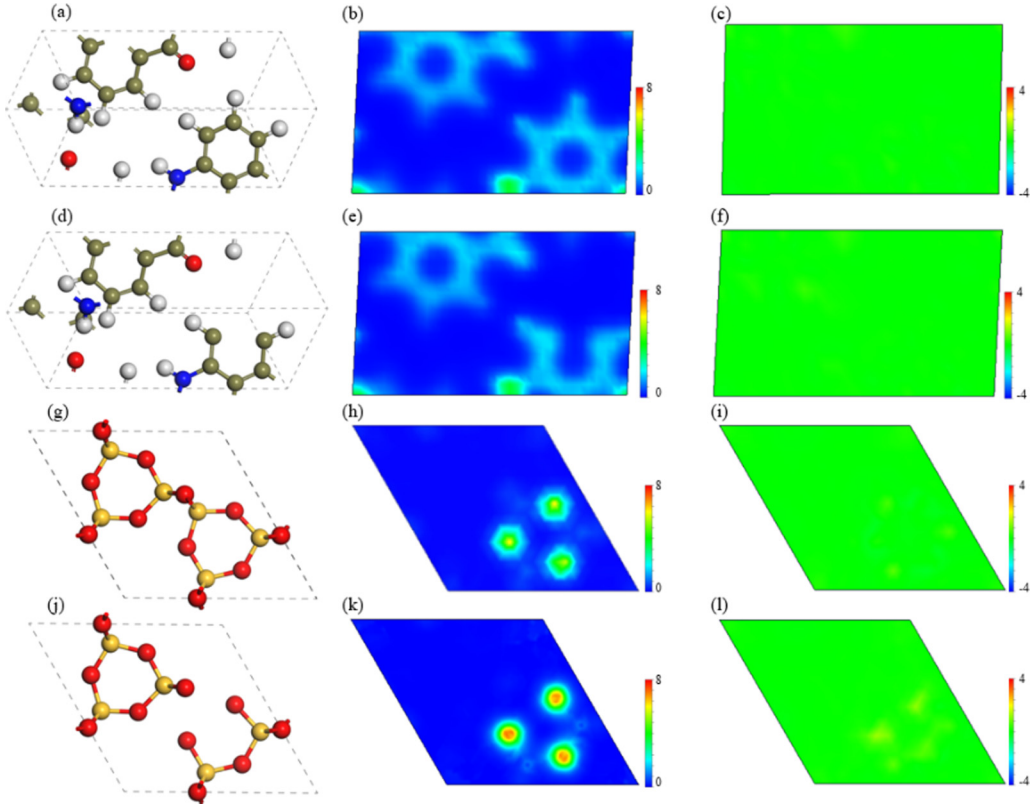

FIG. 4. Visualization of electron charge density (ρ, in $e/\text{Å}^3$). Panels (a), (b), (c) and (d), (e), (f) show crystal structure, ML-predicted ρ, and difference between ML-predicted ρ and DFT-calculated ρ on the C six-ring plane of pristine nomex and nomex with a carbon and a hydrogen vacancy, respectively. Panels (g), (h), (i) and (j), (k), (l) show crystal structure, ML-predicted ρ, and difference between ML-predicted ρ and DFT-calculated ρ on the Si-O six-ring plane of pristine NPO and NPO with a Si vacancy, respectively. Atom color coding: Green, carbon; gray, hydrogen; red, oxygen; blue, nitrogen; yellow, silicon.

charge. The Laplacian of the charge density is of great importance to functional construction [42] and materials analysis [43], and from Table I we can see that the Laplacian is also well predicted with $R^2 > 0.95$.

In order to visualize the performance and transferability of our model, we compare the ML-computed charge densities and difference between charge densities from ML and DFT of pristine nomex, nomex with a C-H vacancy, pristine NPO, and NPO with a Si vacancy in Fig. 4. In all the cases, the building blocks of structures (e.g., the C six-ring and Si-O six-ring) are well represented. For defect structures, although there are more significant differences between ML and DFT, the magnitude of the difference is still low compared with the charge density itself, suggesting high transferability toward defect structures.

We further compare the value of ML-predicted ρ versus DFT-calculated ρ as shown in Fig. 5. The ML model successfully captures the charge densities in most regions for the four structures with good alignment. As shown in Figs. 5(b) and 5(d), our ML model is able to accurately capture the charge density of a vacancy even though no defect structures were present in the training sets. Meanwhile, we can see that most of the deviation in the ML approach compared with DFT is from regions with ultrahigh charge density (near atom cores as shown in Fig. 4), providing insight into directions for further improvement as discussed below.

Last, we evaluate the accuracy of our model for predicting the dipole moment, a materials property that can be derived from the charge density and the crystal structure:

$$\nu_e = \int_{\text{cell}} \mathbf{r} \cdot \rho(\mathbf{r}) d\mathbf{r},$$
$$\nu_i = \int_{\text{cell}} \mathbf{r} \cdot Z(\mathbf{r}) d\mathbf{r}, \quad \mu = |\nu_e + \nu_i|/V_{\text{cell}}, \tag{3}$$

where $\mathbf{r}$ denotes position vector, V_{cell} is the volume of cell, $\rho(\mathbf{r})$ and $Z(\mathbf{r})$ are charges from electron and ion (opposite sign) on $\mathbf{r}$, ν_e and ν_i represent electron and ion dipole vectors, and μ is the dipole moment per volume. The results are shown in Table II, and we can see that the differences between the two electron dipole vectors (from ML and DFT) are very small in all the cases with a high R^2 of 0.99. As for the total dipole moment, although comparative deviations increase after the cancellation of contributions from positive and negative charge, our model can still achieve a R^2 of 0.89, close to that of machine-learning schemes designed specifically for dipole moments (0.93 in Pereira *et al.* [44] and 0.91 in Bereau *et al.* [45]).

IV. DISCUSSION

In order to probe the origin of the transferability of our model, we propose that the difficulty of transferability

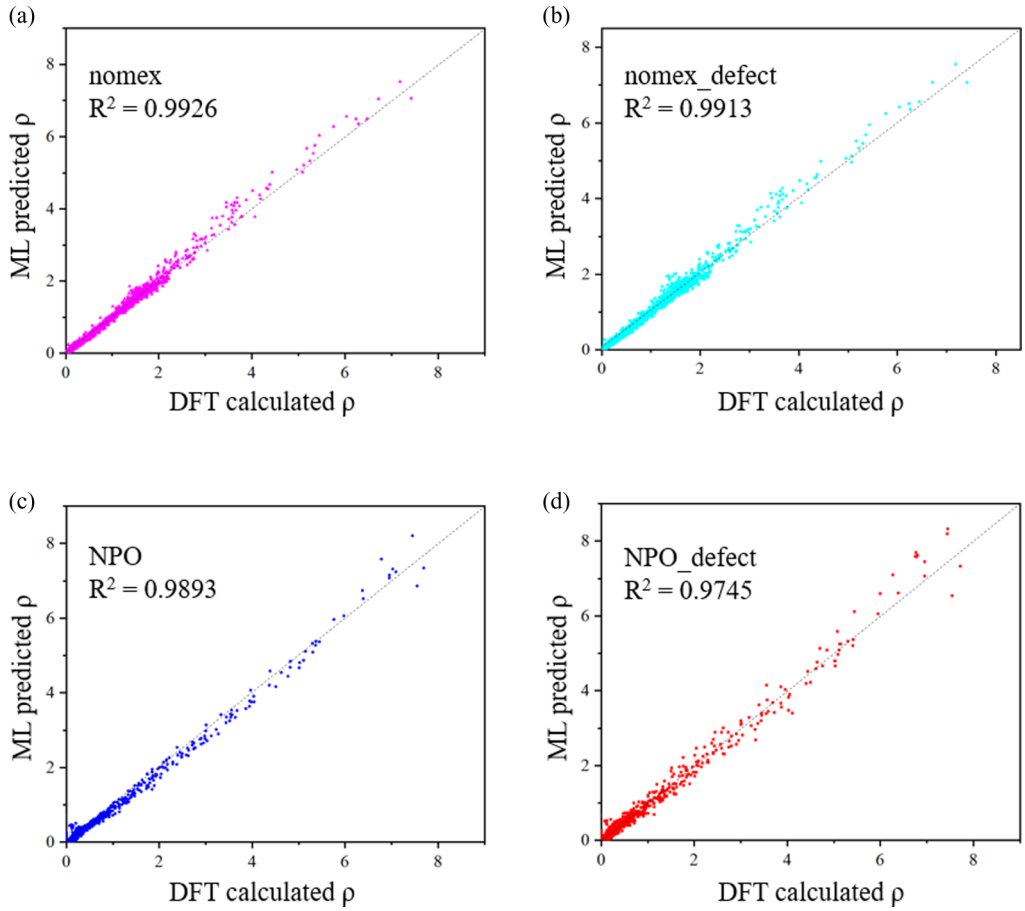

FIG. 5. Panels (a), (b), (c), and (d) show ML-predicted charge density (ρ, in $e/\text{Å}^3$) versus DFT calculated ρ for pristine nomex, nomex_defect, pristine NPO, and NPO_defect, respectively.

between different structures arises from both training and prediction: in training, the model has to distinguish between environments that seems to be "similar" but have very different values of charge, and in prediction, the model has to find similarities between new and existing features. Here, the geometry of neighboring atoms contained in our graph representation simultaneously provides the information for the two tasks, leading to the improved transferability of our model. On the one hand, encoding the geometry makes the local environments more distinguishable; on the other hand, learning the geometry enables the model to speculate on new structural features from existing ones, which also helps to predict the shape of charge density around the defects from the shape of structural features.

TABLE II. Electron dipole vectors (ν_e, in e Å) and total dipole moment (μ, in debyes/Å^3) from ML-predicted ρ and DFT-calculated ρ in the unit cells of half of the test structures, respectively.

Name	ν_e (ML)	ν_e (DFT)	μ (ML)	μ (DFT)
nomex	(20.9, 20.0, 431.4)	(19.1, 19.2, 422.3)	0.652	0.781
nomex_defect	(31.9, 27.5, 440.3)	(29.9, 26.9, 432.0)	0.647	0.782
s-propylene-1	(858.7, 323.8, 438.0)	(898.9, 339.8, 458.5)	0.141	0.260
glycolide	(219.0, 273.9, 295.4)	(216.1, 271.2, 291.8)	0.366	0.463
p-xylylene	(38.6, 363.9, 161.0)	(38.0, 362.6, 160.4)	1.095	1.076
tetramtht	(80.0, 78.8, 381.5)	(83.2, 81.8, 396.5)	0.572	0.383
trans-decenamer	(42.6, 234.5, 306.2)	(44.0, 243.2, 317.5)	0.649	0.409
NPO	(185.4, 321.2, 220.3)	(196.7, 340.6, 232.8)	1.025	1.248
NPO_defect	(202.4, 333.0, 229.0)	(184.4, 325.6, 226.7)	1.390	1.322
JBW	(240.6, 349.7, 362.6)	(232.7, 341.2, 350.4)	1.799	1.547
CAN	(572.9, 992.4, 466.0)	(573.9, 994.1, 465.2)	1.294	1.302
AFY	(787.4, 1363.8, 1094.4)	(777.0, 1345.8, 1081.6)	1.247	1.143
JSN	(857.1, 827.1, 1855.7)	(866.6, 831.6, 1869.2)	1.170	1.260

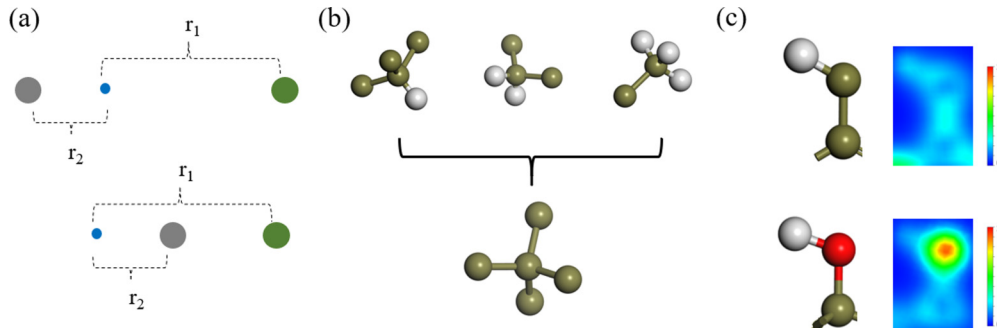

FIG. 6. (a) Sketch of two different local environments with similar sum of atom contributions. (b) Geometries of central carbon atoms with coordinated C1H3, C2H2, C3H1, and C4 atoms. (c) Shape of C-C-H and C-O-H and their charge density distributions (ρ, in $e/\text{Å}^3$). Atom color coding: Green, carbon; gray, hydrogen; red, oxygen.

In order to illustrate the impact of encoding the geometry of neighboring atoms for distinguishing local environments, we sketch two local environments in Fig. 6(a). If the environments of grid points are simply described by considering distances to each atom separately and then summing atom contributions as in the previous models, the two environments would appear to be very similar. However, they are actually quite different, and the difference can be explicitly encoded by the distance between the two atoms. For speculating on new structural features from existing ones, we plot the geometries of central carbon atoms with coordinated C1H3, C2H2, C3H1, and C4 atoms in Fig. 6(b). When predicting charge density around C4, our model can learn from the geometries of C1H3, C2H2, and C3H1 in the training set that the tetrahedral shape of C4 corresponds to an sp^3-hybridized central carbon atom, which gives key information for charge distribution around the central carbon atom. As for transferability to defect-induced structural features, although in the nomex_defect case there is a structural feature (C1H1) that does not exist in the training set with all pristine structures, as shown in Fig. 6(c), the shape of C1H1 (C-C-H, an obtuse angle) is very similar to that of C-O-H in the training set. Therefore, the charge distributions around the two structural features should both be in the shape of an obtuse angle. With the information of geometries, our model can capture such similarity and predict the obtuse-angle-like charge density around C1H1, and the ratio of charge density between C-C and C-H atoms can be learned from the 20+ structural features listed in Fig. 3.

We further conduct a toy-model experiment to verify the above statement regarding geometry-induced transferability, shown in Fig. 7(a). First, a CGCNN model is trained on 3000 grid points within 4 Å of a linear C-C-C molecule, and then used to predict the charge density of an orthogonal C-C-C molecule. To examine the effect of geometry toward predicting new structural features, we sample a new set of 3000 grid points equally from both the linear C-C-C molecule and an orthogonal C-O-C molecule and train another CGCNN model. We find that after incorporating the orthogonal geometry into the training set, the prediction error to the orthogonal C-C-C molecule decreases and is lower than that of the two single-training molecule cases, which shows that encoding geometry helps to predict new structural features. In order to further verify that it is the geometry, not the specific training molecule, that leads to the improvement, we repeat the experiments with the replacement of O by B, N, and F as shown in Table S3 of the Supplemental Material [33], in which one can see that in all cases the prediction error with two training molecules is lower than that with single training molecules. Another insight from this experiment is that currently transferability between elements is still limited in the sense that it is difficult to predict the ratio of charge density between C-C only from that of C-O, which can be attributed to the poor design of the element feature vector, a subject of further investigation in future work.

In the introduction, we also mentioned the effect of the same dimension of the features. We do believe that the same-dimensional features facilitate the training process, since if the dimension of features scales linearly with the number of elements, then the time of training will also largely depend on it, which is undesirable in multielemental systems. However, the same dimension of the features is not the fundamental origin of transferability, and it is less important than encoding geometry, the main origin of transferability as proposed above. In order to further verify the dominant role of geometry in transferability, we perform another toy-model experiment

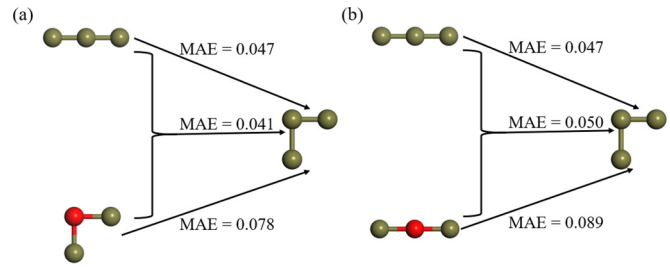

FIG. 7. (a) Illustration of the first toy-model experiment. The top and bottom MAEs (in $e/\text{Å}^3$) are from the predictions to the orthogonal C-C-C molecule by one of the two training molecules (linear C-C-C and orthogonal C-O-C), while the middle one is from the prediction trained on both of the training molecules. (b) Illustration of the second toy-model experiment. The top and bottom MAEs (in $e/\text{Å}^3$) are from the predictions to the orthogonal C-C-C molecule by one of the two training molecules (linear C-C-C and linear C-O-C), while the middle one is from the prediction trained on both of the training molecules (linear C-C-C and linear C-O-C). Atom color coding: Green, carbon; red, oxygen.

[Fig. 7(b)]. In this experiment, we sample a set of 3000 grid points from both the linear C-C-C molecule and a new linear C-O-C molecule and train a CGCNN model with the same settings as the first experiment. Therefore, the dramatic increase of the prediction error to the orthogonal C-O-C molecule and the fact that it is higher than that of the case with the linear C-C-C molecule as the single training molecule can only be attributed to the geometry of training molecules, which shows that the transferability achieved in the first experiment is the result of only geometry, not other conditions including the dimension of features.

We propose further improvements in the scheme presented in three aspects. First, as mentioned we will design architectures to efficiently generate more materials properties based on charge density, especially the total energy of the unit cell, for which both traditional methods (e.g., Kohn-Sham equations [46] or the embedded-atom method [47]) and machine-learning approaches [21,22] are options under consideration. Second, as discussed above regions near nuclei possess the highest deviations, and to improve the sensitivity of our model for small distances between imaginary and real atoms, transformations to weight small distances during the learning can be designed. Last, as mentioned we aim to develop new atom feature vectors that can achieve better transferability between different elements, with one possible approach to learn atomic features back from charge density distributions around each type of atom.

V. CONCLUSION

In summary, we have developed a machine-learning model to predict the electron charge density distribution of materials based on graph convolutional neural networks with $O(N)$ scaling. In the case studies of crystalline polymers and zeolites, local-environment-based graphs are extracted from some structures and features learned, and the learned models are applied to structures different from the training sets. The accuracy and usability of our model have been evaluated by statistical errors, visualization, and quality of charge density-based properties. The most important benefit of our model is high transferability between different structures, which can be attributed to the ability of the graph representation to explicitly encode the geometry of local environments.

ACKNOWLEDGMENTS

This work was supported by the Toyota Research Institute. Computational support was provided by the DOE Office of Science User Facility supported by the Office of Science of the US Department of Energy under Contract No. DE-AC02-05CH11231, and the Extreme Science and Engineering Discovery Environment, supported by National Science Foundation Grant No. ACI-1053575.

[1] J. S. Murray and P. Politzer, WIREs Comput. Mol. Sci. **1**, 153 (2011).
[2] A. Savin, R. Nesper, S. Wengert, and T. F. Fassler, Angew Chem. Int. Ed. **36**, 1809 (1997).
[3] E. R. Johnson, S. Keinan, P. Mori-Sanchez, J. Contreras-Garcia, A. J. Cohen, and W. T. Yang, J. Am. Chem. Soc. **132**, 6498 (2010).
[4] S. Gong, C. Zhang, S. Wang, and Q. Wang, J. Phys. Chem. C **121**, 10258 (2017).
[5] S. Gong and Q. Wang, J. Phys. Chem. C **121**, 24418 (2017).
[6] P. Keblinski, S. K. Nayak, P. Zapol, and P. M. Ajayan, Phys. Rev. Lett. **89**, 255503 (2002).
[7] R. F. W. Bader and P. J. Macdougall, J. Am. Chem. Soc. **107**, 6788 (1985).
[8] A. Kumar and S. R. Gadre, Phys. Chem. Chem. Phys. **17**, 15030 (2015).
[9] S. Zhao, E. Kan, and Z. Li, WIREs Comput. Mol. Sci. **6**, 430 (2016).
[10] Z. Rong, D. Kitchaev, P. Canepa, W. Huang, and G. Ceder, J. Chem. Phys. **145**, 074112 (2016).
[11] N. E. R. Zimmermann, D. C. Hannah, Z. Rong, M. Liu, G. Ceder, M. Haranczyk, and K. A. Persson, J. Phys. Chem. Lett. **9**, 628 (2018).
[12] D. Lee, J. W. Kim, and B. G. Kim, J. Phys. Chem. B **110**, 4323 (2006).
[13] S. Gong, W. Wu, F. Q. Wang, J. Liu, Y. Zhao, Y. Shen, S. Wang, Q. Sun, and Q. Wang, Phys. Rev. A **99**, 022110 (2019).
[14] T. Xie and J. C. Grossman, Phys. Rev. Lett. **120**, 145301 (2018).
[15] R. Gómez-Bombarelli, Chem **4**, 1189 (2018).
[16] R. Gomez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernandez-Lobato, B. Sanchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, ACS Cent. Sci. **4**, 268 (2018).
[17] T. Xie and J. C. Grossman, J. Chem. Phys. **149**, 174111 (2018).
[18] Y. Zhang and C. Ling, npj Comput. Mater. **4**, 25 (2018).
[19] Z. Ahmad, T. Xie, C. Maheshwari, J. C. Grossman, and V. Viswanathan, ACS Cent. Sci. **4**, 996 (2018).
[20] H. Choi, K. Sohn, M. Pyo, K. Chung, and H. Park, J. Phys. Chem. C **123**, 4682 (2019).
[21] B. Kolb, L. C. Lentz, and A. M. Kolpak, Sci. Rep. **7**, 1192 (2017).
[22] G. Pilania, C. Wang, X. Jiang, S. Rajasekaran, and R. Ramprasad, Sci. Rep. **3**, 2810 (2013).
[23] A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad, npj Comput. Mater. **5**, 22 (2019).
[24] F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, and K. R. Muller, Nat. Commun. **8**, 872 (2017).
[25] C. M. Handley, G. I. Hawe, D. B. Kell, and P. L. Popelier, Phys. Chem. Chem. Phys. **11**, 6365 (2009).
[26] G. Pilania, J. E. Gubernatis, and T. Lookman, Comput. Mater. Sci. **129**, 156 (2017).
[27] J. Lee, A. Seko, K. Shitara, K. Nakayama, and I. Tanaka, Phys. Rev. B **93**, 115104 (2016).
[28] K. T. Schutt, H. E. Sauceda, P. J. Kindermans, A. Tkatchenko, and K. R. Muller, J. Chem. Phys. **148**, 241722 (2018).
[29] M. Wessel, M. Brandmeier, and D. Tiede, Remote Sens. **10**, 1419 (2018).
[30] A. Grisafi, A. Fabrizio, B. Meyer, D. M. Wilkins, C. Corminboeuf, and M. Ceriotti, ACS Cent. Sci. **5**, 57 (2019).

[31] E. Schmidt, A. T. Fowler, J. A. Elliott, and P. D. Bristowe, Comput. Mater. Sci. **149**, 250 (2018).
[32] W. M. Haynes, *CRC Handbook of Chemistry and Physics* (CRC Press, Boca Raton, FL, 2014).
[33] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.100.184103 for further details of the calculations, data set constructions, and results.
[34] G. Kresse and J. Furthmuller, Phys. Rev. B **54**, 11169 (1996).
[35] P. E. Blochl, Phys. Rev. B **50**, 17953 (1994).
[36] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[37] S. Grimme, J. Comput. Chem. **27**, 1787 (2006).
[38] D. A. Purser, A. A. Stec, and T. R. Hull, in *Fire Toxicity*, edited by A. Stec and R. Hull (Woodhead, Cambridge, 2010), pp. 515–540.
[39] J. Y. Liu, G. Y. Zhang, K. Clark, and H. Q. Lin, ACS Appl. Mater. Inter. **11**, 10933 (2019).
[40] M. Kuwata, W. Shao, R. Lebouteiller, and S. T. Martin, Atmospheric Chem. Phys. **13**, 5309 (2013).
[41] V. Pichonpesme, C. Lecomte, and H. Lachekar, J. Phys. Chem. **99**, 6242 (1995).
[42] J. P. Perdew and L. A. Constantin, Phys. Rev. B **75**, 155109 (2007).
[43] Z. Shi and R. J. Boyd, J. Phys. Chem. **95**, 4698 (1991).
[44] F. Pereira and J. Aires-de-Sousa, J. Cheminf. **10**, 43 (2018).
[45] T. Bereau, D. Andrienko, and O. A. von Lilienfeld, J. Chem. Theory Comput. **11**, 3225 (2015).
[46] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965).
[47] M. S. Daw and M. I. Baskes, Phys. Rev. B **29**, 6443 (1984).