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

Shashank Pathrudkar [1], Ponkrshnan Thiagarajan [1], Shivang Agarwal [2], Amartya S. Banerjee [3,*] and Susanta Ghosh [1,4,†]

[1] *Department of Mechanical Engineering–Engineering Mechanics, Michigan Technological University*
[2] *Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, USA*
[3] *Department of Materials Science and Engineering, University of California, Los Angeles, CA 90095, USA*
[4] *Faculty member of the Center for Data Sciences, Michigan Technological University*

(Dated: May 13, 2024)

I. EFFICIENT GENERATION OF ATOMIC NEIGHBORHOOD DESCRIPTORS

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

The descriptors described above satisfy the following conditions outlined in [12] and [10]: (i) invariance with respect to rotations and translations of the system (ii) invariance with respect to the permutation of atomic indices, i.e., the descriptors are independent of the enumeration of the atoms. (iii) for a given atomic neighborhood, the descriptors are unique. (iv) the descriptors encode the atomic neighborhood effectively while keeping the overall count low. (v) the descriptors generation process is computationally inexpensive and uses standard linear algebra operations.

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

Algorithm1 Generation of Descriptors

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

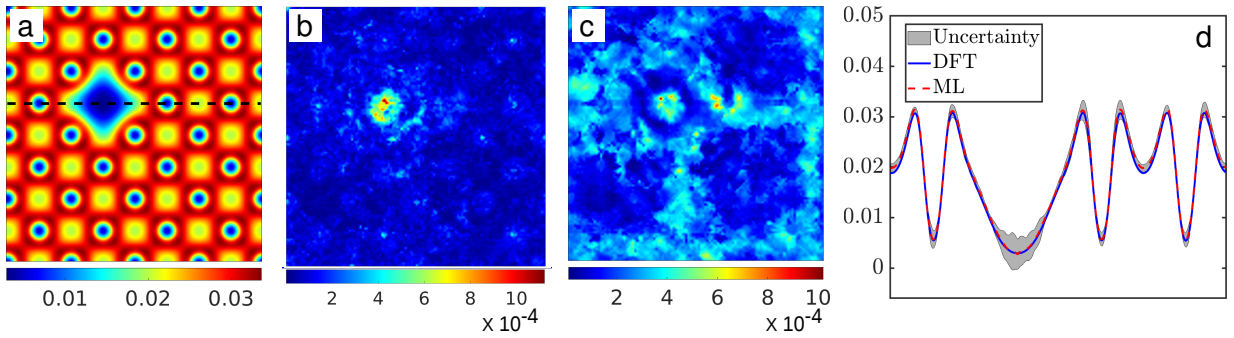
II. COMPUTATIONAL EFFICIENCY

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

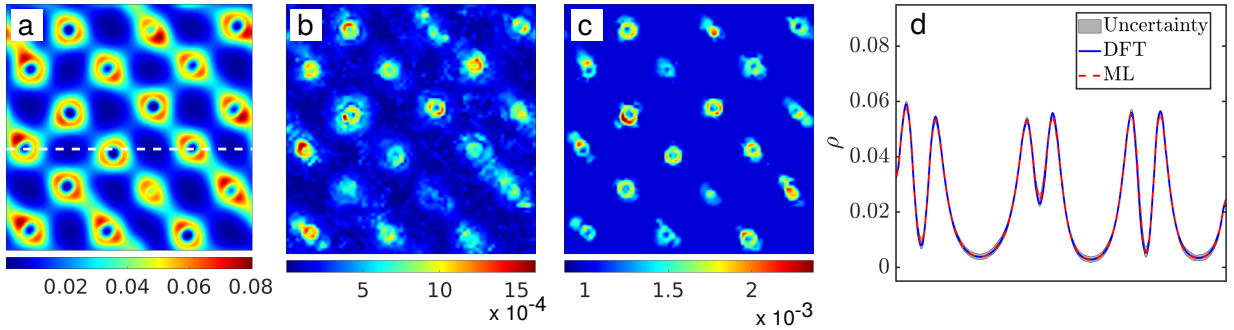
The primary contributor to ML prediction time is descriptor generation, constituting the majority of the computational effort and the remaining time is neural network inference (See Supplementary Tables 1 and 2). Given that neural network inference is well-suited for GPU execution and is commonly performed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

The DFT and ML calculations presented in this work were performed through a combination of resources, namely, desktop workstations, the Hoffman2 cluster at

* asbanerjee@ucla.edu
† susantag@mtu.edu



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

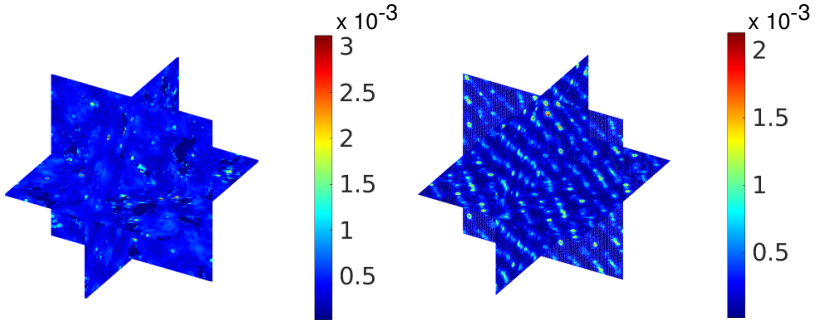


Supplementary Figure 2. Uncertainty quantification $Si_{0.5}Ge_{0.5}$ system containing 216 atoms. (a) ML prediction of the electron density, (b) Epistemic Uncertainty (c) Aleatoric Uncertainty (d) Total Uncertainty shown along the dotted line from the ML prediction slice. The uncertainty represents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

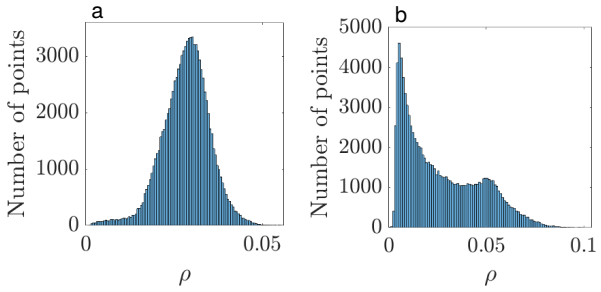
UCLA's Institute of Digital Research and Education (IDRE), the Applied Computing GPU cluster at MTU, and NERSC's supercomputer, Perlmutter. Every compute node of the Hoffman2 cluster has two 18-core Intel Xeon Gold 6140 processors (24.75 MB L3 cache, clock speed of 2.3 GHz), 192 GB of RAM and local SSD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

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

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



Supplementary Figure 3. (Left) Total uncertainty for the Al system ($\sim$ 4.1 million atoms) shown in Fig. II A of the main text. (Right) Total uncertainty for the SiGe system ($\sim$ 1.4 million atoms) shown in Fig. II A of the main text (right).



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

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

Supplementary Table 1. Comparison of DFT and ML wall times for prediction of electron density for an aluminum system. All times are in seconds. The DFT calculations were performed on Hoffman CPUs, ML descriptor generation was done on Hoffman CPUs, and the ML inference was performed on Tesla V100 GPUs.

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

Supplementary Table 2. Comparison of DFT and ML wall times for prediction of electron density for a SiGe system. All times are in seconds. The DFT calculations were performed on Perlmutter CPUs, ML descriptor generation was done on Perlmutter CPUs and the ML inference was performed on Tesla V100 GPUs.

proximately 1.6 seconds on an Nvidia V100 GPU. Overall, the charge density prediction for the 4.1M Al system takes around 30.72 hours of wall time on combined CPU and GPU resources.

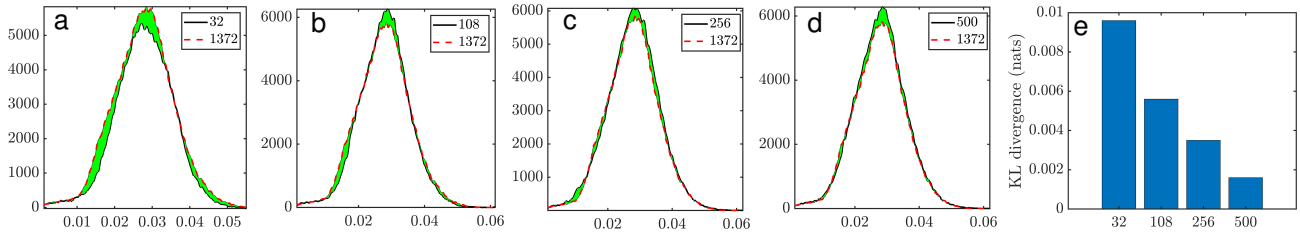
Analogously, for SiGe, we partition the 1.4M atom system into smaller systems composed of 1000 atoms and a grid with dimensions of 132^3 points. The computation of descriptors for this 1000-atom SiGe chunk requires 22.17 seconds on the aforementioned desktop system. The subsequent charge density prediction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

Thus, the techniques described here make it possible to routinely predict the electronic structure of systems at unprecedented scales, while using only modest resources on standard desktop systems.

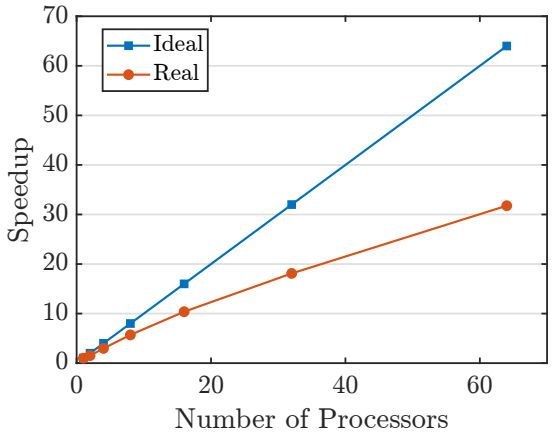
III. FEATURE CONVERGENCE ANALYSIS

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

As an illustration, for the aluminum systems, following algorithm 2 we use an increment of $m = 10$. The algorithm converges to $M = 60$ as seen in Fig. 15 of the main text. Therefore, the set I consists of 60 descriptors. Next, Set II descriptors consist of angles subtended at the grid point by a pair of atoms taken from the set of M neighboring atoms in the set I determined by algorithm 2. Each pair of the neighboring atoms forms an angle at the grid point, yielding a total of $M(M-1)/2$ angles, which quickly becomes computationally intractable with increasing M. To alleviate this issue, we reduce the number of Set II descriptors by eliminating large angles, which are not expected to play a significant role. This



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



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

Algorithm2 Optimal nearest neighbors

$M = 0$ ▷ Initialization
$\epsilon_0 = \epsilon_{-m} = \delta_1 = \delta_2 =$ A large number ▷ Initialization
$\eta =$ tolerance in RMSE
while $\delta_1 \geq \eta$ & $\delta_2 \geq \eta$ **do**
  $M = M + m$ ▷ Increase M by $m \in \mathbb{Z}^+$
  $N_{\text{set I}} \leftarrow M$ ▷ M nearest atoms
  $N \leftarrow N_{\text{set I}}$ ▷ Only set I descriptors
  Compute N descriptors
  Train f_N ▷ Train the BNN
  $\epsilon_M \leftarrow$ RMSE ▷ Compute RMSE
  $\delta_1 \leftarrow |\epsilon_M - \epsilon_{M-m}|$
  $\delta_2 \leftarrow |\epsilon_M - \epsilon_{M-2m}|$
end while
$M = M - 2m$

amounts to choosing angles originating from $M_a < M$ atoms closest to the grid point, and the $k-$ nearest neighbors of each of these atoms. This yields a total of $M_a \times k$ angle descriptors. For various fixed values of k, we iteratively choose M_a till the RMSE over the test dataset converges (Fig. 15 of the main text).

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

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

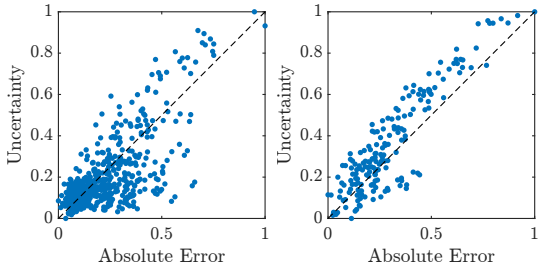
Algorithm3 Optimal number of angles

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

IV. DETAILS ON UNCERTAINTY QUANTIFICATION

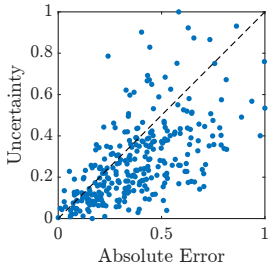
We provide additional results on uncertainty quantification (UQ) in this section. One of the key advantages of the inbuilt UQ capabilities of the present ML model is that it allows us to assess the model's generalizability. To illustrate this, we consider systems with defects and varying alloy compositions. The uncertainty estimates of a model trained without any defect data in training are shown in Fig. 10 of the main text. The model is more confident in its prediction of defects even if a small amount (single snapshot) of defect data is added in training. This is evident by comparing Fig. 1 and main text Fig. 10. This result is in agreement with the fact that unavailability or insufficient training data could yield high epistemic uncertainties at locations where such incompleteness of data exists. In addition to high uncertainty, the error at the defect location increases when data from systems with defects are not used in training. This implies a positive correspondence between error and uncertainty in the Bayesian neural network model. A similar effect of higher uncertainty for unknown compositions is observed for the SiGe systems. Since the model is trained only with data from SiGe systems with 50-50 composition, the uncertainties quantified for this composition shown in Fig. 2 is less in comparison to the prediction for 60-40 composition (Fig. II of the main text). However, the uncertainty for the 60-40 composition is not significantly higher than the 50-50 composition, demonstrating the generalization capability of the ML model.

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



(i) 256 atom Aluminum with single vacancy defect

(ii) 108 atom Aluminum with double vacancy defect



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

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

Results of uncertainty quantification $\approx$ 4.1 million atom aluminum system and $\approx$ 1.4 million atom SiGe system are shown in Fig. 3. With an increase in system size, we extrapolate farther away from the system size included in the training data. Despite this, the total uncertainty of millions of atom systems is similar to that of smaller systems. This implies that the model can predict systems with millions of atoms with the same level of confidence as smaller systems, which in turn assures the accuracy of the predictions. Looking ahead, we plan to further enhance the credibility of million-atom predictions by validating against results obtained from upcoming and state-of-the-art techniques involving Density Functional Theory (DFT) computations at a large scale [3, 4, 8, 17].

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

V. DETAILS ON THE ADVANTAGES OF TRANSFER LEARNING

As demonstrated in prior research [22] and in this work, employing data from larger systems for training enhances the accuracy of machine learning models. However, the following question persists: what is the appropriate largest sizes of the training system to achieve a sufficiently accurate machine learning model that works across scales? To answer this question, we propose the following approach.

To ensure accurate predictions for bulk systems (comprising thousands or more atoms), it is imperative that our model be trained on data that statistically resembles such systems. Small-scale systems with only a few tens of atoms may not adequately represent the bulk limit, primarily due to the periodicity constraints inherent in simulations. This calls for training the model using larger systems. To determine appropriate training system sizes that adequately represents bulk systems, we employ the Kullback-Leibler (KL) divergence [15]. We consider the largest available system as the most faithful representation of bulk systems and use it to determine the largest size of the training systems. For the case of Aluminum, a system consisting of 1372 atoms can be reliably calculated using KS-DFT and is chosen as the reference. We compare the electron density distributions from various available systems against this reference system. The KL divergence values then guide us in selecting the largest training system needed to train a model that can accurately predict even at large scales (relevant to the reference system). Specifically, the largest training systems chosen by us contain 108 atoms, as these systems are found to be sufficiently statistically similar to the 1372-atom reference system (as illustrated in Fig. 5). This meticulous selection process guarantees that our machine learning model is accurate at large scales while providing a judicious stopping point to our transfer learning scheme by determining the largest system needed for training. Thus, we present an approach that answers the question of selecting training system size and reduces the reliance on ad hoc heuristics for doing so.

The transfer learning approach [20] significantly reduces the root-mean-square error of a test dataset while costing much less computation for the training data generation. To depict this, a comparison of the transfer learned model with various non-transfer learned models is shown in Fig. 8.

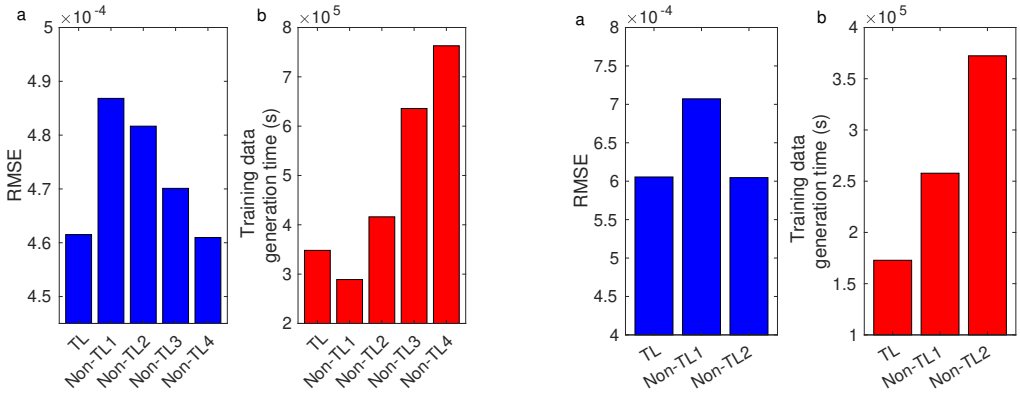
We found that transfer learning helps to reduce the error and uncertainty in prediction for larger systems. By adding data from the 108-atom aluminum systems in training, during the transfer learning approach, we significantly reduce the error (by 56%) and uncertainty (by 29%) of the predictions for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

VI. DETAILS ON BAYESIAN NEURAL NETWORK

Architecture: We use a Densenet [11] type architecture with three Dense blocks for the Bayesian Neural networks in this work. Each Dense block is composed of three hidden layers with 250 nodes per layer and a GELU activation function [9]. The skip connections in the Densenet-type architecture are weighted by a trainable coefficient. These skip connections have multiple advantages. Firstly, they prevent gradients from diminishing significantly during backpropagation. Further, they facilitate improved feature propagation by allowing each layer to directly access the feature generated by previous layers. Finally, these skip connections promote feature reuse, thereby substantially reducing the number of parameters. Such skip connections have been used for electron density predictions in the literature [22].

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

Training Details: The parameters of the BNNs for the 32-atom Al system and 64-atom SiGe systems were initialized randomly with values drawn from the Gaussian distribution. The mean of the parameters were initialized with values drawn from $\mathcal{N}(0, 0.1)$. The standard deviations were parameterized as $\sigma = \log(1 + \exp(\tau))$ so that σ is always non-negative. The parameter τ was initialized with values drawn from $\mathcal{N}(-3, 0.1)$. The priors for all the network parameters were assumed to be Gaussians: $\mathcal{N}(0, 0.1)$. With these initializations and prior assumptions the initial models (i.e. model for 32-atom Al system and 64-atom SiGe system) were trained using standard back-propagation for BNNs. The Adam optimizer [13] was used for training and the learning rate was set to 10^{-3} for all the networks used in this work. In the case of transfer learning, we freeze both the mean and standard deviation of the initial one-third layers of the model and re-train the mean and standard devia-



(i) Al system (ii) SiGe System

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

tions of the remaining layers of the model. The prior assumptions, initialization of the learnable parameters, and their learning procedures remained the same as described above for the 32-atom Al and 64-atom SiGe systems. The training time for the Al and SiGe systems are presented in Supplementary Table 3. All the Bayesian Neural networks are trained on NVIDIA A100 Tensor Core GPUs

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

- Al: 127 snaps from 32 atom data and in addition 25 snaps from 108 atom data. The 108 atom data has $90 \times 90 \times 90$ grid points, while the 32 atom system has $60 \times 60 \times 60$ grid points.

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

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

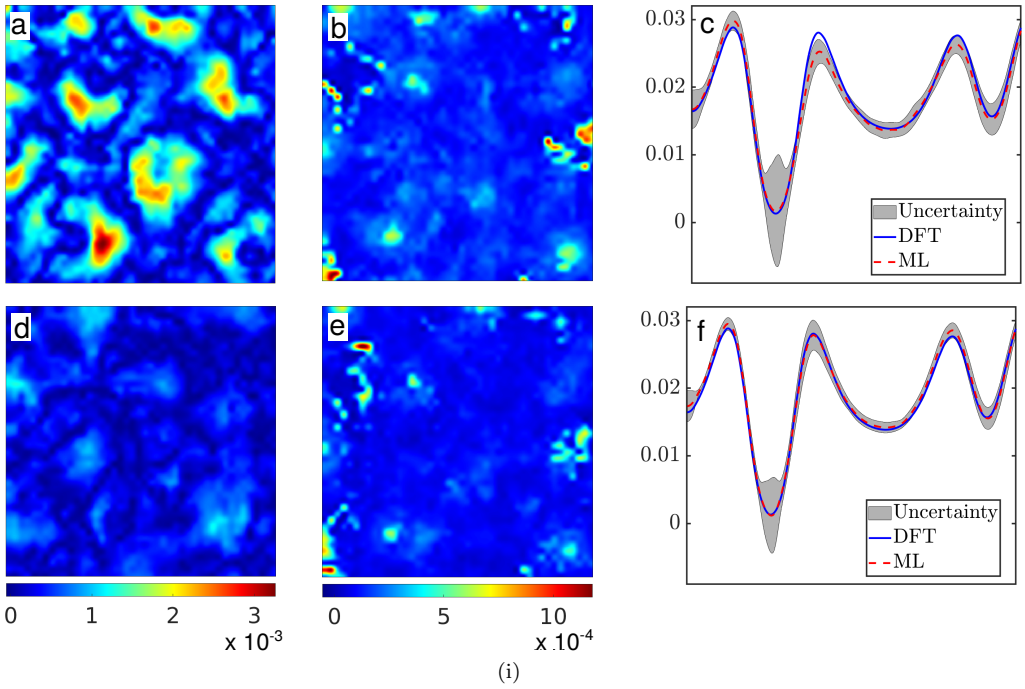
Supplementary Table 3. GPU Training times for the BNNs. The training was performed on the NVIDIA Tesla A100 GPU.

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

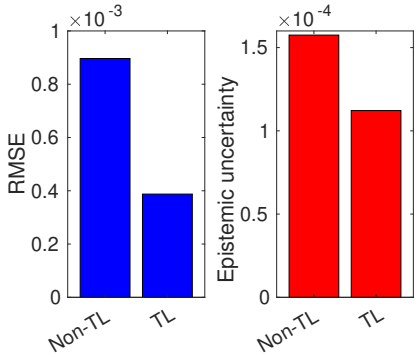
VII. POSTPROCESSING RESULTS

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

The time for the calculation of the total energy and forces from ML-predicted densities via postprocessing involves computation of the electrostatic, exchange correlation and band-energy terms, and uses a single diagonalization step to compute wave-function dependent quantities. Therefore, its computational time is similar to that of a single self-consistent field (SCF) step in a regular DFT calculation, provided the same eigensolver is used. For reference, using the MATLAB version of the SPARC code [21] on a single CPU core, the postprocess-



(i)



(ii)

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

ing time is about 174 seconds for a 32 atom aluminum system while it is about 1600 seconds for 108 atoms. This also includes the time for computation of the Hellmann-Feynman forces. We would also like to mention here that this postprocessing step can be significantly sped up by the ML prediction of other relevant quantities, such as the band energy and electrostatic fields [18]. As for the atomic forces, i.e., energy derivatives with respect to atomic coordinates, automatic differentiation of the underlying neural networks can be employed to speed up calculations. All of these constitute ongoing and future work.

VIII. CALCULATION OF THE BULK MODULUS FOR ALUMINUM

We show a comparison between some material properties calculated using the electron density predicted by the ML model, and as obtained through DFT calculations. Specifically, we compute the optimum lattice parameter and the bulk modulus for aluminum — these corresponding to the first and second derivatives of the post-processed energy curves (Fig. 7 of the main text), respectively. A summary of our results can be found in Supplementary Table 6. It can be seen that bulk modulus

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

Supplementary Table 4. Accuracy of the ML predicted electron density in terms of the L^1 norm per electron, calculated as $\frac{1}{N_e} \times \int_\Omega \left|\rho^{\text{scaled}}(\mathbf{r}) - \rho^{\text{DFT}}(\mathbf{r})\right| d\mathbf{r}$, for various test cases for an FCC aluminum bulk system (N_e is the number of electrons in the system). Also shown in the Supplementary Table are errors in the different energies as computed from ρ^{scaled}. The test data set for post-processing was chosen such that it covered examples from all system sizes, configurations, and temperatures. For calculating the relevant energies, ρ^{scaled} was used as the initial guess for the electron density, and a single Hamiltonian diagonalization step was performed. Energies were then computed.

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

Supplementary Table 5. Accuracy of the ML predicted electron density in terms of L^1 norm per electron, calculated as $\frac{1}{N_e} \times \int_\Omega \left|\rho^{\text{scaled}}(\mathbf{r}) - \rho^{\text{DFT}}(\mathbf{r})\right| d\mathbf{r}$, for various test cases for $Si_{0.5}Ge_{0.5}$ (N_e is the number of electrons in the system). Also shown in the Supplementary Table are errors in the different energies as computed from ρ^{scaled}. The test data set for post-processing was chosen such that it covered examples from all system sizes and temperatures. For calculating the relevant energies, ρ^{scaled} was used as the initial guess for the electron density, and a single Hamiltonian diagonalization step was performed. Energies were then computed. For Si_xGe_{1-x}, we used $x = 0.40, 0.45, 0.55, 0.60$.

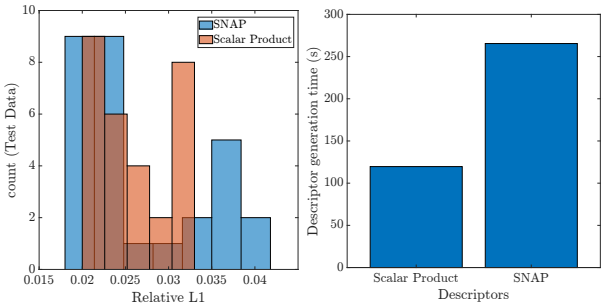
differs by only about 1%, while the lattice parameters are predicted with even higher accuracy. Notably, the predicted lattice parameter and the bulk modulus are very close to experimental values [16], and the deviation from experiments is expected to decrease upon using larger supercells to simulate the bulk, a trend also seen in Supplementary Table 6. This is consistent with the overall results shown in the main manuscript and further reinforces the predictive power of our model for non-ideal systems.

IX. COMPARISON WITH MODELS BASED ON OTHER DESCRIPTORS

In the main text, we have presented errors achieved in electron density prediction by our model. The results indicate that our approach is generally as accurate as (and in some cases outperforms) previous work [1, 22]. To further compare it with existing similar approaches, we compare it with electron density predictions made via the well known SNAP descriptors [5, 19]. Specifically, we have compared the relative L1 error (as defined in [22]) on 29 test snapshots using the dataset of an Aluminum system with 32 atoms. We used the same training dataset and employed a neural network for both the descriptors. Both the descriptors yield nearly identical L1 errors (although the distribution of errors is different as shown in Fig. 10). At the same time, the calculation of the scalar product descriptors employed here exhibits computational efficiency, requiring about 50% less time than generation of the SNAP descriptors. To ensure a fair and accurate comparison of descriptor computation time, the computations for both descriptors were performed on a single-core CPU. We utilized the data of Be 128 atoms provided by [7] and the SNAP code provided by [5, 6], for comparing descriptor calculation time.

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

Supplementary Table 6. A comparison between the calculated lattice parameter and the bulk modulus for aluminum using ρ^{ML} and ρ^{DFT} (DFT values in parentheses). We observe that the predicted lattice parameter closely matches the value given by DFT calculations. The "true" optimized lattice parameter for Al, using a fine k-space mesh, is found to be 7.5098 Bohr while experimental values are about 7.6 Bohr [2]). The ML predicted value of the bulk modulus matches the DFT value very closely, which itself is very close to the experimental value of approximately 76 GPa [16], at room temperature.

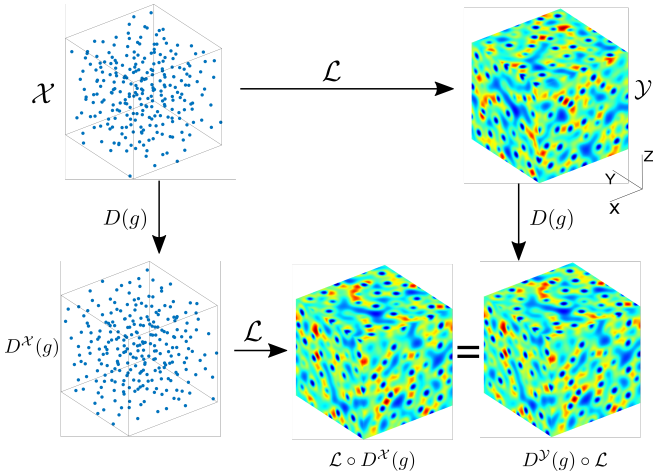


(i) Error Histogram (ii) Time comparison

Supplementary Figure 10. Comparison with SNAP descriptors

X. EQUIVARIANCE OF THE MODEL

In this section we show numerically that our model is equivariant, i.e., the predicted electron density is invariant with respect to overall rotation, translation, and permutation of atomic indices of the underlying material system. As mentioned in [14], equivariance can be achieved by designing invariant features and predicting the electron density as a scalar valued variable. Since our model is based on these strategies, our machine learning model is expected to be equivariant, theoretically. We substantiate this claim numerically in Supplementary Figure 11.



Supplementary Figure 11. Schematic showing preservation of equivariance in our model. $\mathcal{X}$ is 256 atom Aluminum system at high temperature (chosen such that there are no obvious intrinsic rotational symmetries of the system). $\mathcal{Y}$ is the corresponding electron density. $D(g)$ corresponds to the rotation of $\frac{\pi}{2}$ around the Y-axis. $\mathcal{L}$ is the composite map from the system to the electron density. We observe numerically that, $||\mathcal{L} \circ D^{\mathcal{X}}(g) - D^{\mathcal{Y}}(g) \circ \mathcal{L}||_{\infty} \approx 10^{-10}$. $||\mathcal{L} \circ D^{\mathcal{X}}(g) - D^{\mathcal{Y}}(g) \circ \mathcal{L}||_{\infty}$ is not exactly zero because of roundoff errors in billions of floating point operations involved in descriptor calculations and forward propagation through neural networks. Thus, $\mathcal{L} \circ D^{\mathcal{X}}(g) = D^{\mathcal{Y}}(g) \circ \mathcal{L}$, and hence equivariance is preserved.

[1] A. Chandrasekaran, D. Kamal, R. Batra, C. Kim, L. Chen, and R. Ramprasad. Solving the electronic structure problem with machine learning. *npj Computational Materials*, 5(1):22, 2019.

[2] A. S. Cooper. Precise lattice constants of germanium, aluminum, gallium arsenide, uranium, sulphur, quartz and sapphire. *Acta Crystallographica*, 15(6):578–582, 1962.

[3] S. Das, et al. Large-scale materials modeling at quantum accuracy: Ab initio simulations of quasicrystals and interacting extended defects in metallic alloys. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2023.

[4] S. Das, P. Motamarri, V. Subramanian, D. M. Rogers, and V. Gavini. Dft-fe 1.0: A massively parallel hybrid cpu-gpu density functional theory code using finite-element discretization. *Computer Physics Communications*, 280:108473, 2022.

[5] J. A. Ellis, et al. Accelerating finite-temperature kohn-sham density functional theory with deep neural networks. *Physical Review B*, 104(3):035120, 2021.

[6] J. A. Ellis, et al. mala-project. `https://github.com/mala-project/mala`, 2021.

[7] L. Fiedler, et al. Predicting electronic structures at any length scale with machine learning. *npj Computational Materials*, 9(1):115, 2023.

[8] V. Gavini, et al. Roadmap on electronic structure codes in the exascale era. *Modelling and Simulation in Materials Science and Engineering*, 31(6):063301, 2023.

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

[10] L. Himanen, et al. Dscribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 247:106949, 2020.

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

[12] H. Huo and M. Rupp. Unified representation of molecules and crystals for machine learning. *Machine Learning: Science and Technology*, 3(4):045017, 2022.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] T. Koker, K. Quigley, and L. Li. Higher order equivariant graph neural networks for charge density prediction. In *NeurIPS 2023 AI for Science Workshop*, 2023.

[15] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[16] S. Raju, K. Sivasubramanian, and E. Mohandas. The high temperature bulk modulus of aluminium: an assessment using experimental enthalpy and thermal expansion data. *Solid state communications*, 122(12):671–676, 2002.

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

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

[19] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics*, 285:316–330, 2015.

[20] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

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

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