Spatial density neural network force fields with first-principles level accuracy and application to thermal transport

Alejandro Rodriguez,[1] Yinqiao Liu,[1,2] and Ming Hu[1,*]

[1]*Department of Mechanical Engineering, University of South Carolina, Columbia, South Carolina 29208, USA*
[2]*Key Laboratory of Materials Modification by Laser, Ion and Electron Beams, Dalian University of Technology, Ministry of Education, Dalian 116024, China*

(Received 11 February 2020; revised 15 May 2020; accepted 24 June 2020; published 17 July 2020)

Constrained by the fixed mathematical form of most empirical potentials used in classical molecular dynamics (MD) simulations, many properties of materials cannot be captured within experimental accuracy. On the other hand, accurate electronic structure calculations based on quantum theory, most notably density functional theory (DFT), are limited to several hundred atoms within a picosecond, which makes the method inadequate for modeling systems beyond the nanoscale. A combination of speed from classical MD and fidelity from DFT can be achieved through machine learning methods. Herewith, we developed an approach named spatial density neural network force fields (SDNNFFs) by training neural networks to "learn" and predict DFT-level forces. Our model focuses on the usage of a three-dimensional mesh of density functions, which together act as a mapping of the atomic environment and provides a physical representation of the forces acting on the central atom. Several notable advantages arise from the SDNNFF, including (1) the avoidance of the chain rule on the total energy and other variables by direct calculation of the forces from the neural network, (2) the ever large $N \times t$ scaling of the training data, where N is the number of atoms in a supercell and t is the number of evaluated structures by first-principles, and (3) the significant reduction in parameters and human effort needed to successfully train a force- and/or property-converged neural network force field. Overall, we focus on modeling DFT-level forces with minimal computational cost and parametrization for rapid prediction of phonon-based properties and future molecular dynamics of large-scale systems. To demonstrate the SDNNFF, we trained several models on diamond structures, including bulk silicon (Si), diamond, silicon carbide (SiC), and boron arsenide (BAs), and predicted their phonon dispersions and lattice thermal conductivities using the direct solution to the phonon Boltzmann transport equation. For phonon properties, we utilized a fitting method for obtaining the second- and third-order force constants, which outperforms the highly force-sensitive finite displacement method when employing neural network force fields. In comparison to DFT lattice thermal conductivity, we obtained high precision results from our SDNNFF within 0.7% for Si, 6.2% for diamond, 2.76% for SiC, and 7.46% for BAs, with further agreement with experiments. The phonon dispersions from the SDNNFF also matched those from direct DFT and experiments. The developed approach for accurately predicting phonon transport properties of crystalline materials would largely benefit the design of advanced materials with improved performance, such as complex thermoelectric devices and low thermal resistance interfaces for nanoelectronics. Future applications of our SDNNFF model could be extended toward including atomic energy into the algorithm and simulating large-scale heterogeneous systems for quasielectronic representations for various properties.

DOI: 10.1103/PhysRevB.102.035203

I. INTRODUCTION

The representation of atomic interactions, or the potential energy surface (PES), is key for capturing realistic behavior of materials and their properties. Classical molecular dynamics (MD) and density functional theory (DFT) are two examples of conventional atomistic simulation methods. In MD, the movements of point particles representing atomic nuclei are dependent on the pairwise or many-bodied potential functions with fixed functional forms. Since interactions vary between atomic species, these potential functions typically consist of adjustable coefficients based on empirical data, scientific knowledge, force-matching with *ab initio* [1], or a combination of the three. An exemplary empirical potential accounts for imaginary electrons surrounding the nuclei and any realistic pair-wise or many-bodied interactions associated with them, e.g., bond stretching, van der Waals forces, torsions, etc. [2]. While the simple functional forms of these potentials allow quick evaluation of the atomic forces and resulting Newtonian dynamics, they are strongly limited in modeling chemical reactions and in predicting multiple properties comparable to experimental data [3]. In addition, any improvement to classical potentials is not straightforward because the dependencies between parameters are not known *a priori* [4]. On the other hand, DFT takes into explicit account the electron densities surrounding nuclei and computes

*Author to whom all correspondence should be addressed: hu@sc.edu

stable structures via iterative minimization of global energy according to the Schrödinger equation [5]. The PES in DFT is represented by the Born-Oppenheimer approximation, where the kinetic energies of nuclei are separated from the electronic Hamiltonian due to their massive sizes in comparison to electrons [6]. Since DFT is a "bottoms-up" or first-principles approach, an in-depth understanding of material behavior is possible, such as the phonon branch contributions to inverse temperature thermal conductivity for monolayer gallium nitride [7]. However, contrary to the almost linear scaling with the number of atoms in MD, the explicit consideration of electronic degrees of freedom limits DFT to several hundred atoms, which makes the modeling of behaviors beyond the nanoscale, such as the growth of a silicon anode in charging a lithium-ion battery [8], an arduous or otherwise impossible task.

Considering the drawbacks of classical and quantum-based methods, it becomes clear why researchers are pursuing methods that combine the best in both simulation methods by bridging the subatomic and atomic scales together. Recent popularity in machine learning (ML) has inspired researchers to merge this family of computational techniques with atomistic simulation [9]. Inspired by biological processes, ML is oftentimes correlated with the "trial-and-error" mindset in which the predictive power of a model is improved from reference data with little to no human intervention. Here, the reference data "teach" or fit the model iteratively for the most optimum function between provided inputs and outputs with a high degree of flexibility [10]. Recent computing advances has given strong leeway for ML into materials research, where the generation of data such as the $\sim 10^5$ compounds on the Materials Project Genome [11] has led to some major accomplishments [12]. These efforts include, but are not limited to, the discovery of eight new solid superionic lithium ion conductors [13] and the prediction of interfacial thermal resistance between two materials related to temperature, coupling strength, and in-plane tensile strains [14].

This paper focuses on the application of ML for the quantitative development of interatomic interactions from accurate quantum-based data. The machine learned potential (MLP) has been a topic of consistent research for more than a decade, with the interest of developing an accurate mapping of the PES from high fidelity data with minimal prior knowledge of the interatomic interactions. Out of the many ML algorithms, feed-forward neural networks were chosen in this study as the algorithm of choice due to their recent popularity in commercial and academic research, and are applied to the atomic simulation field as neural network potentials (NNPs) [15]. These NNPs have yielded agreeable DFT energies and forces for a wide variety of systems, including but certainly not limited to condensed-phase water [16], Li-Si alloys [17], $C_{10}H_{20}$ isomers [18], copper surfaces [19], and amorphous Li_3PO_4 [20]. Other than the composition, structure, and size of modeled systems, papers with NNPs are further broken down by the treatment methods on network inputs prior to training. Since ML algorithms are purely mathematical models with no physical basis, the ability to learn meaningful relationships in data is largely dependent on the quality and consistency of the dataset itself [21]. For NNPs, the representations of the local chemical environment in the training data are critical for accurate predictions of central-atom energies and forces. As an example, suppose the inputs to a NNP are all the Cartesian coordinates $\vec{r}_i = (x_i, y_i, z_i)$ of atoms i, with the total system energy as the output. A rotation of these atoms about an axis should realistically yield same energy prior to rotation, since the relative distances and angles of the atoms are consistent. Nonetheless, since the values of the atomic coordinates suffer dramatic changes from the rotation, an NNP not trained to support the rotated coordinates will most likely output an energy largely different than that prior to the spatial transformation.

Contrary to most referenced NNPs, including the high-dimensional neural network potential (HDNNP) proposed by Behler *et al.* [22], here we develop a neural network named the spatial density neural network force field (SDNNFF) that *directly computes and predicts force fields rather than the PES*. Originally as an inspiration to our work, here we discuss the HDNNP briefly. In the HDNNP, the Cartesian coordinates of local atoms within a cutoff are transformed into a series of radial and angular symmetry functions that maintain invariance between similar atomic environments, resulting in consistent total energies. The net force F_i acting on atom i is subsequently computed from total energy (E) with respect to its coordinates according to $F_i = -\nabla_i E$. Here, our model diverges from this approach as we pursue the per-atom forces without the energies, since the forces are of interest to compute the interatomic force constants (IFCs) and the dynamics of atomic systems. Direct evaluation of the forces could save computational resources in MD via removal of the total energy and its derivative. In the HDNNP, the per-atom forces are computed by the chain rule and require the evaluation of both the symmetry functions and their derivatives, requiring more CPU time than classical potentials [23]. However, to output the central-atom forces directly, radial and angular symmetry functions as inputs are not a feasible option since force vectors are indistinguishable in a polar mapping of the environment, e.g., a rotation of an atomic environment would yield an unchanging set of symmetry function values while the force vector components would experience dramatic changes. As such, we decided to *develop our own functional mapping of nearby atoms in three-dimensional (3D) space rather than polar space*. Briefly, this is done by generating a mesh of points about every central atom, where predefined density functions measure the local density of the atomic neighbors at each mesh point. Then, the collective group of atomic densities distributed in 3D space serve as representative atomic environments for the neural network to recognize DFT-level force components in response to relative neighbor positions. Overall, the goal of the SDNNFF is to output first-principles based atomic forces at network speeds by proper treatment of instantaneous atomic coordinates.

The structure of the paper is organized as follows: A technical background for the SDNNFF and other details for network training are given in Secs. II A–II D. Then, in Sec. II E, to show the reliability of our SDNNFF model, we compute the harmonic (second order) and anharmonic (third order) force constants and phonon properties (i.e., phonon dispersion and lattice thermal conductivity) of Si from the developed neural networks by the conventional finite displacement method (FDM) [24] and the force constant fitting package ALM [25].

Afterwards, in Sec. III, the ALM method is applied to neural networks trained for similar bulk crystalline materials, including diamond, SiC, and BAs. The resulting phonon properties for each material were evaluated by running convergence tests on various SDNNFF parameters and comparing the phonon properties to those evaluated from DFT-FDM. In Sec. IV, we analyze the results and comment on the SDNNFF in light of other recently developed methods. Finally, in Sec. V we provide a summary and some perspective for future study with the SDNNFF method.

II. MODEL CONSTRUCTION AND VALIDATION

A. Theoretical background of SDNNFF

As previously mentioned, a successful neural network force field must be capable of predicting DFT-level force components as a function of neighboring atomic positions. In a 3D orthogonal *xyz* coordinate system, because the force vector $\vec{F_i} = (F_i^x,\ F_i^y,\ F_i^z)$ is projected along the reference global axes, the input descriptors should also quantify the atomic positions in the same *xyz* space. A simple way to do this is to take the Cartesian coordinates of nearby atoms and directly insert them into a neural network. However, many issues arise from this approach. First, during neural network training, stability may be compromised from training data that varies greatly due to translated or rotated coordinates. Here, we intend to train networks on force components that differ between spatially transformed varieties of the same system, which is unlike the case when training for the energy. The question then becomes how should the atomic coordinates be fed into the neural network, such that the inputs are consistent to avert large changes in the weights during training? An example of instability from the inputs is the order in which the atomic coordinates are fed into the neural network. Since the weights of a given network are fixed numerical values, inserting two equivalent atomic environments with random ordering for the neighbor coordinates will certainly change the network output. This problem is trivial for ordered systems with no significant diffusion of atoms, since the network can be fed inputs according to positionally equivalent atoms in space (e.g., by ordering the *xyz* coordinates from greatest to least for all atomic environments as the input). On the other hand, the training of atomic coordinates in disordered systems is infeasible since there are rarely any equivalent atomic structures, which adds an element of randomness to the inputs and prevents the fine tuning of the weights during training. The second point against Cartesian coordinates as inputs arises from the entering and exiting of atoms with respect to a certain cutoff radius, which occurs frequently during MD simulations. Treating the Cartesian coordinates as inputs is impractical since neural networks are assigned a fixed number of inputs at the start of training, which is incompatible with the variable number of neighbors within the cutoff that may arise during simulations. Again, while this may be negligible for materials with a fixed number of neighbors (e.g., low-temperature solids), the method becomes unreliable for systems where atomic neighbors travel far from the central atom. For the compatibility of neural network force fields with any arbitrary system, the focus is to represent the space surrounding the central atom independent of any periodicity or possible atomic environments found in DFT or MD.

Inspired by other works on atomic representations, we decided to apply the concept of symmetry functions to represent 3D variations in the environment to avoid the issues associated with Cartesian coordinates. We follow the simple forms presented by Huang *et al.* [26], where equally spaced cosine functions serve as mathematical indicators for the atomic occupation surrounding a central atom. This form is especially desirable considering that only three adjustable hyperparameters are present: the cutoff radius (R_c), the grid resolution (k), and the density function width (D). The first two parameters are physically intuitive since R_c determines the distance truncation of the force contribution to the central atom and k provides the spatial resolution of the grid points surrounding the central atom. The third parameter D measures the range of each cosine function, and we show later in this section a universal value for given structures can be found using network convergence testing. Therefore, a good representation of the local environment and the resulting central-atom forces depends on the convergence of the force error and material properties by adjustment of R_c and k. Additionally, the grid-centered cosine functions were modified to transform *xyz* coordinates to a form independent of the quantity or periodicity of the neighboring atoms. In other words, our method was designed to represent any given atomic environment, provided that the inputs to the neural network are treated as fixed observers of the neighbors, and the quantity of atoms inside R_c does not change the number of inputs to the neural network. Therefore, a high flexibility is achieved in these SDNNFF functions, formally named density functions, which allow neural networks to capture the complex force fields involved in a wide variety of systems.

The theoretical background for the SDNNFF density functions is provided here. For a 3D grid with $k > 1$ grid divisions, a given spherical cutoff radius R_c, $\vec{R_n} = (x_n, y_n,\ z_n)$ as the coordinates of the nth atom relative to a given central atom, and $\vec{\alpha} = (\alpha^x, \alpha^y, \alpha^z)$ as the grid positions with each component ranging from 1 to k, the proposed spherical density function is shown in Eq. (4), with Eqs. (1)–(3) provided as definitions:

$$L = \frac{2R_c}{k-1}, \tag{1}$$

$$\vec{R_g} = -R_c + (\vec{\alpha} - 1) * L, \tag{2}$$

$$R_n^{\vec{\alpha}} = \sqrt{\left(x_n - R_g^x\right)^2 + \left(y_n - R_g^y\right)^2 + \left(z_n - R_g^z\right)^2}. \tag{3}$$

If $|\vec{R_g}| = \sqrt{R_g^{x2} + R_g^{y2} + R_g^{z2}} \leqslant R_c$,

$$\varphi_{\vec{\alpha}} = \begin{cases} \sum_{n=1}^{N} \frac{1}{2} * \left[\cos\left(\frac{\pi}{D*L} * R_n^{\vec{\alpha}}\right) + 1\right] & \text{if} \quad R_n^{\vec{\alpha}} < D * L \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where

L – is the distance between equally spaced grid points with side length R_c,

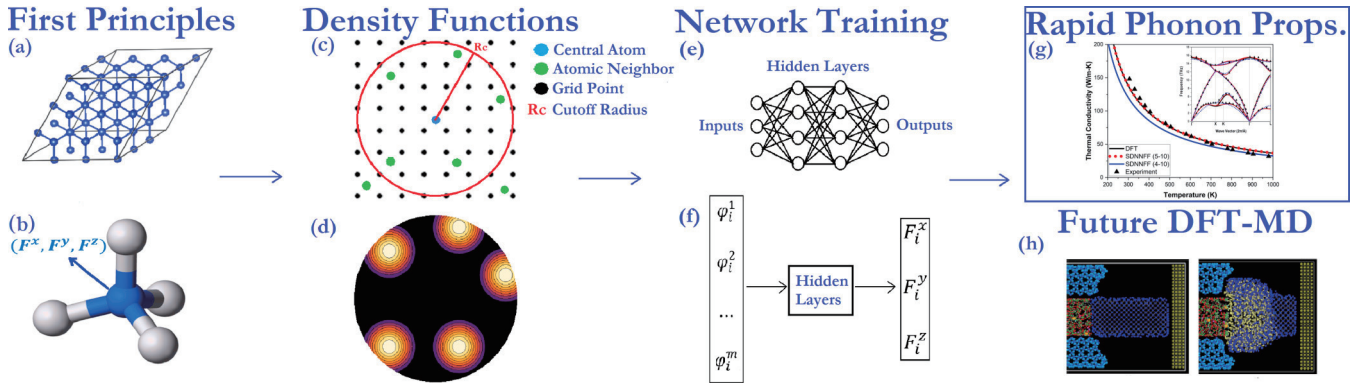

FIG. 1. A schematic containing the workflow for the SDNNFF. First, we gather (a) the crystal structure and its (b) per-atom forces from first principles, then for each atom we describe the local atomic environments by (c) density functions with a finite neighbor cutoff radius. The result is a (d) atomic density mapping of the environment which can be fed into a (e) neural network. Specifically, we train the network on (f) density values found at each grid point to obtain the force vector for various atomic configurations. Then, we can make DFT-level force predictions at significantly improved speeds for applications such as (g) rapid phonon property predictions done here (shown: predicted phonon dispersion and lattice thermal conductivity of Si) or (h) in the future simulate MD at DFT accuracy (shown: lithium ion diffusion through silicon anode [8]).

$\vec{R}_g$ – is the Cartesian coordinates of grid point $\vec{\alpha}$ with $\vec{R}_g = (R_g^x, R_g^y, R_g^z)$,

$R_n^{\vec{\alpha}}$ – is the distance between an atom n and a grid point $\vec{\alpha}$,

N – is number of atoms within R_c,

D – is the local density function cutoff factor,

$\varphi_{\vec{\alpha}}$ – is the local atomic density value at grid point $\vec{\alpha}$.

The basic requirement of the SDNNFF density functions is to provide a description of the 3D atomic occupation in space, as seen in Fig. 1, c-d. Hence, the grid space surrounding the central atom is defined by the diameter of the cutoff sphere, or $2R_c$. This range is then divided into k partitions, generating the intergrid distance L according to Eq. (1). In Eq. (2), the real-space coordinates of the grid points $\vec{R}_g$ are given from $-R_c$ at $\alpha^{x,y,z} = 1$ to R_c at $\alpha^{x,y,z} = k$ in each of the x-/y-/z directions. Equation (3) provides $R_n^{\vec{\alpha}}$ as the relative distance between a neighboring atom n and a given grid point $\vec{\alpha}$, which measures the spherical vicinity of atoms with respect to a grid point.

These definitions lead to the density function in Eq. (4) provided as a cosine function ranging from 0 to 1 varying with radial distance $R_n^{\vec{\alpha}}$, as seen in Fig. 2. The density function $\varphi_{\vec{\alpha}}$ considers the radial influence of up to N atoms inside R_c, and their contributions are summed into one value per grid point to provide a quasidensity of atoms in space. The summation also acts as a necessary conditioning step for usage in neural networks since the number of atoms within the spherical cutoff becomes independent of the number of inputs to the network. Here, the number of grid points $\vec{\alpha}$ translates to an equal number of fixed inputs to the SDNNFF and scales with the grid resolution k. Additionally, the "if" statement preceding the density function in Eq. (4) is a physical treatment, where only the grid points within the atomic cutoff are considered due to two primary reasons: (1) Like in MD, the forces are strictly dependent on the radial proximity between the central atom and the neighbors, so the consideration of atoms within a cubic grid is obviously intangible. Instead, we limit the grids to those within the cutoff sphere, making the description of the forces in the neural network more realistic. (2) The number of grids considered is significantly reduced when truncating to the grids within the sphere of radius R_c versus a cube with side length $2R_c$. This effectively reduces the number of necessary inputs to the neural network by a significant margin (at least by volume fraction $\frac{\frac{4}{3}*\pi\times R_c^3}{(2*R_c)^3} \times 100\% = 52.36\%$). In turn, we can increase the grid resolution k to describe the atomic environment more precisely within the cutoff sphere.

A discussion of the density function parameter D is necessary to understand its role in the SDNNFF. As per the definition in Eq. (4), the density function ranges from unity at $R_n^{\vec{\alpha}} = 0$ and zero at a distance $R_n^{\vec{\alpha}} = D * L$. Beyond $R_n^{\vec{\alpha}} = D * L$, the function is treated as zero throughout the rest of

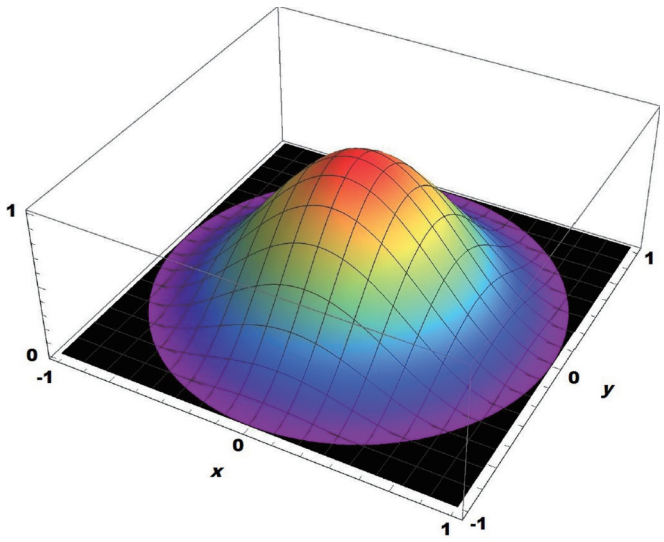

FIG. 2. The shape of the per-node density function in two dimensions, with radius $D \times L = 1$ centered at $\vec{R}_g = (0, 0)$. Atoms in the black region would result in a value of zero, increasing to unity as the atom approaches $\vec{R}_g$.

the domain, providing a smooth curve delineating the local concentration of atoms at each grid point. Here, the purpose of the cutoff factor D is to control the size of the local spherical domain that scales with a constant grid length L. The choice in D is important to guarantee no gaps in the domain between adjacent density functions, where a small value may cause the density functions to completely neglect the presence of certain neighbors critical for computing the forces. On the other hand, a value that is excessively large would both desensitize the $\varphi_{\vec{\alpha}}$ set to smaller atomic displacements due to the greater width of the local density functions and make each $\varphi_{\vec{\alpha}}$ less distinguishable to those of the surrounding grid points due to the inclusion of more atoms in the summation. Furthermore, because all the density functions here are uniform in size and shape, D controls the degree of overlap with nearby density functions and in turn serves as an implicit representation of the angular positions of atoms. While the density functions only depend on radial distance R and not on angular values, multiple sufficiently wide density functions that react simultaneously to atomic movement can provide a good interpolation of the atomic densities between the points in the local environment. The idea is analogous to GPS systems, where multiple orbiting satellites transmit long-range signals to triangulate a position on the surface of the Earth. Here, the value $\varphi_{\vec{\alpha}}$ from each grid point "satellite" is simultaneously fed into the neural network, and as a multidimensional nonlinear model, we expect the network to recognize the set of $\varphi_{\vec{\alpha}}$ values positioned at equally spaced 3D points as a unique description of the atomic environment. As such, in Sec. II D, we find the optimal value for D by comparing the performance of the resulting material-based neural networks.

B. Data generation

For training the SDNNFF, data acquisition is vital for the force prediction of all possible atomic configurations found during long-time-scale simulations. Realistically, we would require a nearly infinite number of data points to cover all possible combinations of atomic environments. However, since neural networks excel at interpolation, the training data are at a minimum required to cover the possible range of inputs [27]. This implies that neural networks must be trained to suit the application, and because the maximum displacement of atoms scales with the temperature of the system, the coverage of possible atomic configurations can be achieved by training neural networks on a variety of temperatures. Here, the goal was to predict the second- and third-order constants of crystalline materials to find phonon-based properties with *ab initio* accuracy. Since the force constants require small atomic displacements less than 0.03 Å, we decided it would be enough to train neural networks on data far from the melting temperatures (i.e., solid crystalline phase).

Six temperatures were arbitrarily chosen, namely 0, 50, 100, 300, 500, and 800 K, to train the material specific SDNNFFs on a sufficiently wide range of atomic configurations. Here, the 0-K configurations are perfect crystal structures. To model a variety of atomic environments, we use *ab initio* MD (AIMD) with DFT from the Vienna *ab initio* simulation package (VASP) with the projector augmented wave (PAW) method and a 10^{-6}-eV energy convergence criterion to generate the structures. The AIMD simulations were sped up with a low $1 \times 1 \times 1$ k-space resolution since at this stage there was no concern for the accuracy of the forces but for the rapid generation of the atomic configurations at various temperatures. For each crystal at nonzero temperature, 5000 low-resolution steps were run for $3 \times 3 \times 3$ primitive supercells (54 atoms in total) containing Si, diamond, SiC, and BAs. Also, a time step of 1 fs was chosen for Si, SiC, and BAs, while a time step of 0.5 fs was chosen for diamond due to its known high vibrational frequency. Then, 1250 structures were randomly chosen out of the 5000 AIMD configurations and pooled together to avoid biased training, where high-resolution $4 \times 4 \times 4$ k-space static DFT calculations with 10^{-6} eV energy and 10^{-3} eV/Å force convergence criterions were performed on each structure. A total of 4×10^5 nonzero temperature atomic configurations for each Si, diamond, SiC, and BAs were obtained. An arbitrary 1250 0-K (perfect crystal) structures were added to the data pool for the inclusion of equilibrium structures during training. Note that the 0-K configuration for each material was run in static DFT to check for equilibrium (i.e., 0 eV/Å forces) and was not run in AIMD.

C. Neural network details

After the static DFT calculations by VASP, postprocessing for the output data was performed in Python 3.6.8 with the NumPy library package [28]. Our in-house Python script converts the generated atomic configurations from the VASPoutput to the SDNNFF input. The neural network inputs were processed according to Eqs. (1)–(4) and saved in a separate file along with the corresponding DFT forces. For training the SDNNFF, the TensorFlow library for Python was utilized [29]. To compare with other papers containing NNPs, all neural networks here were similarly constructed with two hidden layers and 500 nodes each with a default minibatch size of 32 [26,30]. The SDNNFF data from our earlier Python script was read and split into 80% training data, 10% validation data, and 10% testing data. In addition, the input density function column vector and output force vector were normalized so that they ranged from -1 to 1 according to

$$I_{\text{network}} = 2 * \frac{I - I_{\min}}{I_{\max} - I_{\min}} - 1, \tag{5}$$

where I is a density function or force component from DFT and I_{network} is the converted input/output for the neural network [27]. The values for $I_{\max}$ and $I_{\min}$ were taken as the maximum/minimum values for each component out of the entire dataset. As such, these values were saved in a separate file since all future uses of the network require the reapplication of Eq. (5) to the inputs and outputs for consistency with the trained weights. The He uniform initializer was used to generate the initial weights and biases of the network [31], the Adam algorithm was used as the optimizer for updating the weights with a learning rate of 0.0001 [32], and the exponential linear unit (ELU) was implemented as the activation function of choice [33]. The custom loss function Γ was defined as an RMSE of the three force components predicted by the neural network F^k_{NN} against the DFT force

components F^k_{DFT}, or

$$\Gamma = \sqrt{\frac{1}{3} * \sum_{k=x,y,z} \left(F^k_{\mathrm{NN}} - F^k_{\mathrm{DFT}}\right)^2}. \quad (6)$$

After training, the testing loss was computed across the entire testing set with Eq. (6) for comparison with other networks trained on various cutoffs and grid resolutions.

D. Determining the local density cutoff factor

With the data generated and the network parameters defined, we proceeded to investigate the necessary value for the cutoff factor D in the density functions. At this stage, we decided to start with high-resolution networks $k = 12$ (672 inputs) and a large R_c containing the fifth neighbors for Si (6 Å), diamond (3.93 Å), SiC (4.82 Å), and BAs (5.3 Å). Note that these cutoff distances are approximately 0.05 Å greater than the fifth-neighbor distance to guarantee the capturing of displaced atoms when later computing the force constants in Sec. II E. In addition, the number of inputs is quite large in comparison to recent reports for HDNNP (i.e., 41 inputs) [30]. Later these parameters would be reduced to decrease the number of required density function calculations for the network input and to improve the efficiency of the force calculator, but for now these are simply test cases to minimize noise error in analyzing D. To observe if a universal value exists for D in the case of diamondlike materials, we trained multiple high-resolution networks for each material with different values of D and compared their testing RMSE values according to Eq. (6). Specifically, we sampled values $D = \frac{1}{2}$, $\frac{\sqrt{2}}{2}$, $\frac{\sqrt{3}}{2}$, 1, $\sqrt{2}$, $\sqrt{3}$, and 2, which correspond to a $D \times L$ equal to the grid length ($D = 1 \times L$), the diagonal of the grid length in 2D ($D \times L = \sqrt{2}L$), and the diagonal of the grid length in 3D ($D = \sqrt{3} \times L$) in multiples of $\frac{1}{2}$, unity, and 2.

From Fig. 3, it is clearly seen that D plays a significant role in the force accuracy of the network. Specifically, we can see that there is a converged RMSE value at $D = \sqrt{2}$ for all materials examined here. From an intuitive standpoint, the convergence is due to the overlapping of density functions with those at nearby grid points, which supports the earlier idea that the angular information is preserved in the set of input density functions and therefore the representation of the atomic environment is improved. The overlap begins at $D = 1$, where the edges of the density functions lie on nearby grid points. The results seem to converge at a slightly higher value of $D = \sqrt{2}$, where the grid points are well within the influence of multiple neighboring density functions. Recognizing the negligible differences in RMSE when $D \geqslant \sqrt{2}$, we decided to arbitrarily use $D = \sqrt{3}$ for all material neural networks investigated here. These RMSE differences could be due to statistical noise in training the network, and thus $D = \sqrt{3}$ was chosen purely to guarantee force convergence due to the overlap of density functions. However, we recommend in future studies that a convergence test for D should be performed for new structures other than diamond, considering how the packing of atomic neighbors may impose other requirements on the range of the density functions for converged forces. In addition, care should be taken in choosing a large cutoff factor, since as mentioned before the density function values could become "clouded" from the excessive number of atoms included in the summation, thus generating difficulty for the network to detect atomic configurations from the $\varphi_{\vec{\alpha}}$ set. Furthermore, the inclusion of more atoms in the summation for $\varphi_{\vec{\alpha}}$ can increase computational cost, since more cosines are required to be computed for every atom inside $D \times L$ at each grid point.

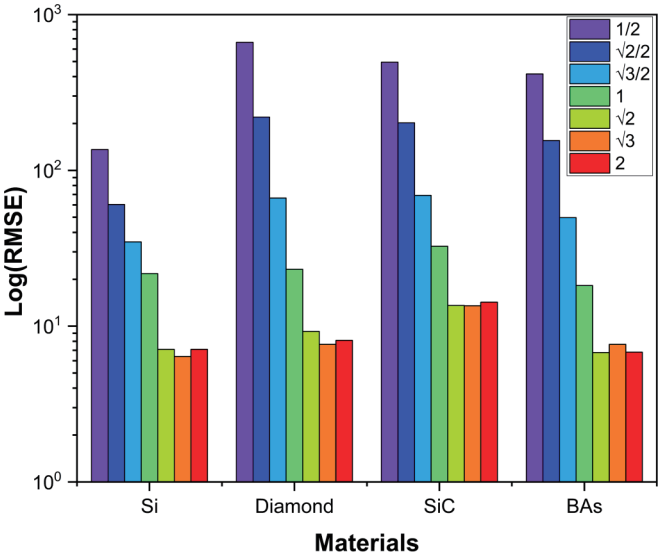

FIG. 3. The logarithm of the testing RMSE for various D values evaluated by the high-resolution SDNNFF networks for the case of bulk Si.

For the trained networks with $D = \sqrt{3}$, we also observed the learning curves to compare the training and validation losses. Normally we expect the validation loss to decay to a minimum then rise after a given number of epochs (training cycles), which is a symptom for overfitting. During training, the network with the lowest validation error is saved since this model can best generalize untrained configurations. However, we noticed that none of our networks experienced this phenomenon, rather the validation curves were seen to continue to slowly decay with no bound as seen in Fig. 4. While overfitting could have occurred at an epoch much larger than 500, at this point the two curves were visually close to an asymptote, so we stop at 500 for all networks to save computational resources.

E. Generation of the force constants and phonon properties

The performance of the neural networks was primarily determined by their ability to reproduce DFT-level phonon properties. Initially we investigated the properties by calculation of the force constants via conventional FDM. Here, we started with Si as a benchmark case since it is a well-known semiconductor. In this paper, we used the phonopy [34] and phono3py [35] packages for evaluating the second- and third-order force constants and for obtaining phonon dispersions as well as lattice thermal conductivity. We used the same high-resolution Si network as developed in Sec. II D to see if our method could yield agreeable phonon properties to those from direct DFT. With FDM, 181 3 × 3 × 3 primitive supercell

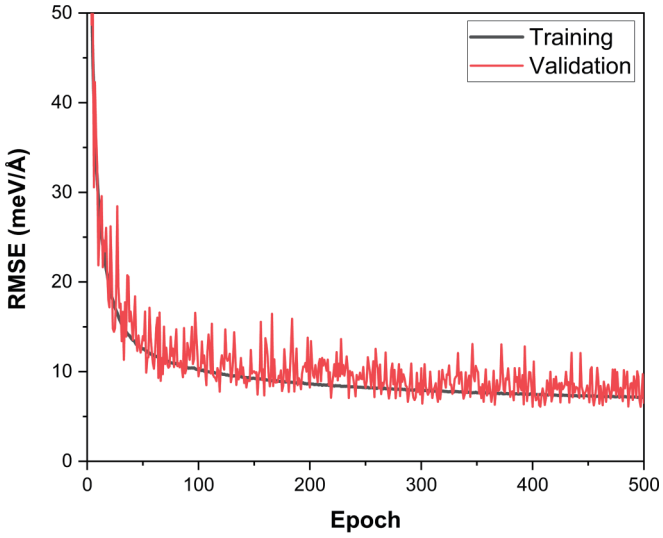

FIG. 4. The training and validation learning curves for the high-resolution Si network.

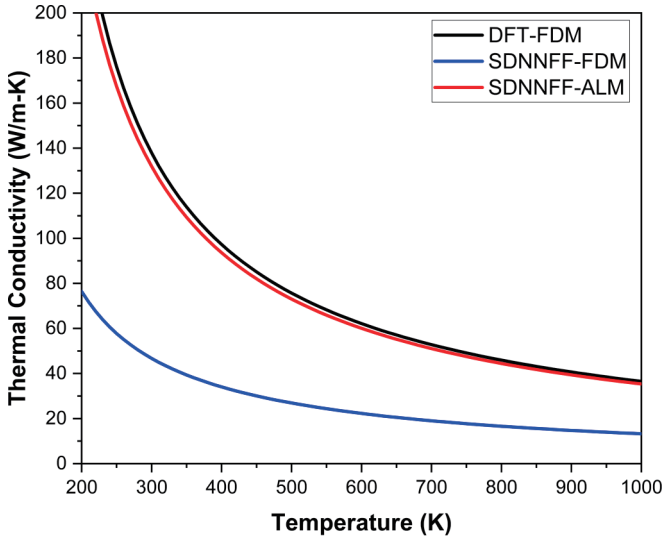

FIG. 6. The temperature-dependent lattice conductivity of bulk Si evaluated by DFT with FDM, SDNNFF with FDM, and SDNNFF with ALM.

structures of Si were generated with the default displacement of 0.03 Å. We computed the forces for these structures by static DFT with the parameters in Sec. II B and by the trained high-resolution neural network. Then, the second- and third-order force constants were computed followed by the phonon dispersion and lattice thermal conductivity. The lattice thermal conductivity was obtained with a 20 × 20 × 20 q-point mesh using the direct solution of linearized BTE.

As seen from Figs. 5 and 6, while the phonon dispersion from the SDNNFF-FDM agrees relatively well with that from DFT-FDM, the comparison of thermal conductivity between the two force calculators for FDM was significantly worse. We assumed that the phonon dispersions, which only require second-order IFCs, are much less sensitive to the force noise error from the neural network than the thermal conductivity, which require both the second- and third-order IFCs. To prove

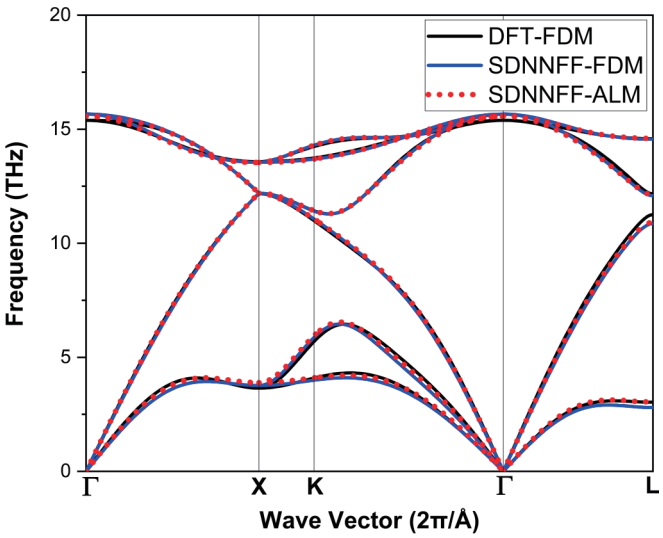

FIG. 5. The phonon dispersion of bulk Si evaluated by DFT with FDM, SDNNFF with FDM, and SDNNFF with ALM.

this, we performed computer experiments by artificially imposing random noise into the DFT forces to see if the phonon properties deteriorate in a similar manner. Noise values of up to $\pm 10^{-n}$ eV/Å, with $n = 2,\ 3,\ 4,\ 5$, were added to the DFT forces. The second- and third-order force constants, along with the phonon dispersions and conductivities, were obtained by the same procedure as before but with the noise-induced forces. We show the resulting plots in Figs. 1 and 2 of the Supplemental Material [36]. Briefly, we obtained a converged phonon dispersion with $n = 3$, while convergence for the thermal conductivity occurred later at $n = 5$. These figures agree well with our earlier speculation, where the thermal conductivity requires a demanding 10^{-5} eV/Å or 0.01 meV/Å force accuracy to match DFT results which is well below the capabilities of the neural network even at a high resolution. In Fig. 7 we summarize this point by plotting the thermal conductivity of Si at 300 K against the artificial noise in DFT. We observe again that the thermal conductivity from DFT converges at a noise level of $n = 5$ (up to $n = 8$ plotted to show convergence) while the thermal conductivity from the SDNNFF was consistent with DFT noise levels between $n = 3$ and $n = 4$, sharing values far below pure DFT results.

We quickly realized that FDM was not suitable for the evaluation of the force constants via SDNNFF due to the observable strong force sensitivity. While the possibility of reducing the RMSE exists by increasing the grid resolution, the number of inputs to the network would have quickly made the method excessively demanding for, say, large scale MD. Instead, we decided to use an alternative force-constant calculator, namely the ALM method. While the details of the method can be found from Tadano *et. al.* [25], a short discussion is made here. In FDM, the force constants are computed by systematic displacement patterns in crystalline structures, where a minimum number of unique displacement patterns or structures are generated. From a series expansion on the potential energy, the second-order force constants require one atomic displacement, the third-order force constants

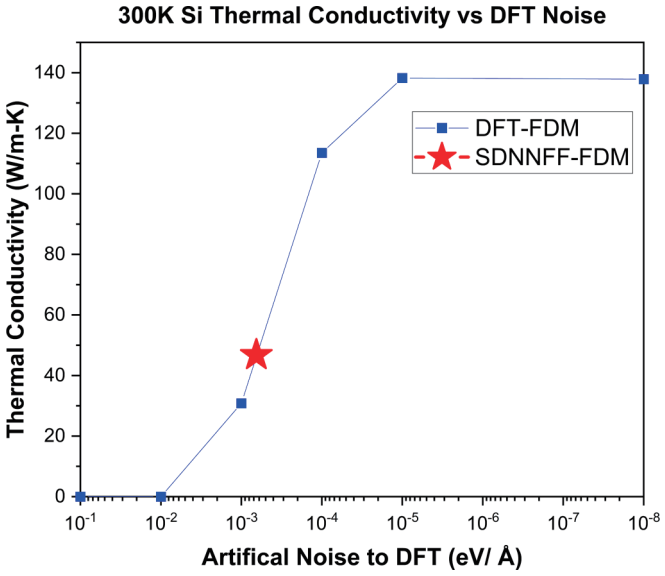

FIG. 7. 300-K thermal conductivity of bulk Si vs artificial noise to DFT. The result from SDNNFF-FDM is also shown for comparison.

require a pair of atomic displacements, and further order terms require more simultaneous atomic displacements [37]. Then, the electronic structure calculations are done on all resulting structures for the DFT forces and the second- and third-order force constants can be computed. However, contrary to FDM, ALM approaches the force-constants calculation stochastically by fitting the force constants rather than solving for them directly. This is done by applying random displacements to all atoms in the supercell up to a user-specified amplitude. The result is an arbitrary number of randomly displaced structures which, with statistical fitting methods, yields the force constants. The downside to ALM is that many data points are needed to obtain converged force constants, so a number of structures greater than that from FDM need to be evaluated and is therefore more time-consuming when using direct DFT.

With the ALM extension in phonopy/3py, we were able to evaluate the force constants from the same neural network for Si. We chose to randomly displace atoms by 0.03 Å for an arbitrary 1000 structures. Here, the neural network evaluated the forces for the Si $3 \times 3 \times 3$ primitive supercell $\sim 10^3$ times faster than DFT, so the force calculation for 1000 structures was a trivial concern. In addition, we expect the neural network cost to scale linearly with N atoms since the calculation only depends on the relatively constant number of neighbors in the local environment, similar to MD and contrary to the scaling with the cubic electron count ($\sim N_e^3$) in DFT [23]. Viewing the resulting phonon properties as seen in Figs. 5 and 6, we realized the exceptional performance in replicating the DFT-FDM results from the SDNNFF-ALM force constants. The most likely reason for the improvement is in the statistical cancellation of noise provided by ALM, where many atomic displacements (and therefore unique interactions) are necessary to eliminate small discrepancies in the predicted forces. Indeed, FDM is limited to a fixed number of displaced structures with only one to two atoms displaced at a time for the second- and third-order force constants. The small sample of displacements increases the demand for accurate forces, as we have seen previously with the artificial disturbance of the DFT forces in FDM structures. Additionally, the finite displacements in DFT generates a nonzero force for all the atoms in the unit cell, whereas a real-space force cutoff approach such as the SDNNFF generates equilibrium forces (i.e., 0± noise) for atoms outside the cutoff of the displaced atom(s). As such, the truncation of the forces outside the cutoff is possibly another source of the incompatibility between the SDNNFF and FDM. Note that we strongly wish to emphasize the difference between the real-space force cutoff (R_c) and the pairwise atomic displacement cutoff used often in FDM code, where the latter is used to reduce the cost of third-order IFC calculations by neglecting structures with distances between displaced atoms beyond the specified cutoff. In practice, the pairwise displacement cutoff in DFT-FDM does not define a force cutoff for atoms in the cell, since the force is dependent on the gradient of the electronic Hamiltonian [38] and is attributed by all atoms in the supercell regardless of the pairwise displacement cutoff. Although for SDNNFF-FDM we tried a pairwise displacement cutoff equal to the real-space force cutoff to capture as many atoms as possible for the force calculation, some structures still had undisplaced atoms outside the real-space force cutoff of the two displaced atoms, and force information loss was inevitable. Accounting for both information loss and inherent NNFF noise, the reduced sensitivity of the force constants led us to prioritize ALM for the evaluation of phonon properties using force calculators involving noise.

TABLE I. The Corresponding number of grids / network inputs to the grid resolution.

Grid resolution (k)	8	9	10	12
Number of grids/network inputs	160	257	360	672

III. APPLICATION TO THERMAL TRANSPORT

With success in reproducing Si DFT results, the next step is to evaluate the phonon properties of the suggested materials by adjustment of the SDNNFF parameters. Similar to bulk Si, we began by testing the high-resolution neural networks for bulk diamond, SiC, and BAs from Sec. II D using ALM. We also developed other networks with combinations of fourth- and fifth-neighbor cutoffs and $k = 8,\ 9,\ 10,$ and 12 for convergence to DFT-FDM results with a minimum number of grid points and atomic neighbors. Note that we again applied a cutoff that is 0.05 Å greater than that for the fourth- and fifth-neighbor distances to account for displacements in ALM. In addition, we noticed similar learning curve behaviors as seen in Fig. 4 in these networks and do not show them here for brevity. However, the final testing RMSE values for the various atomic cutoffs and resolutions are displayed in Fig. 8 for comparison. We also provide the number of input nodes (or the length of the $\varphi_{\vec{\alpha}}$ set) corresponding to each k in Table I, since as mentioned in Sec. II A only the nodes within the cutoff radius are considered as the network inputs.

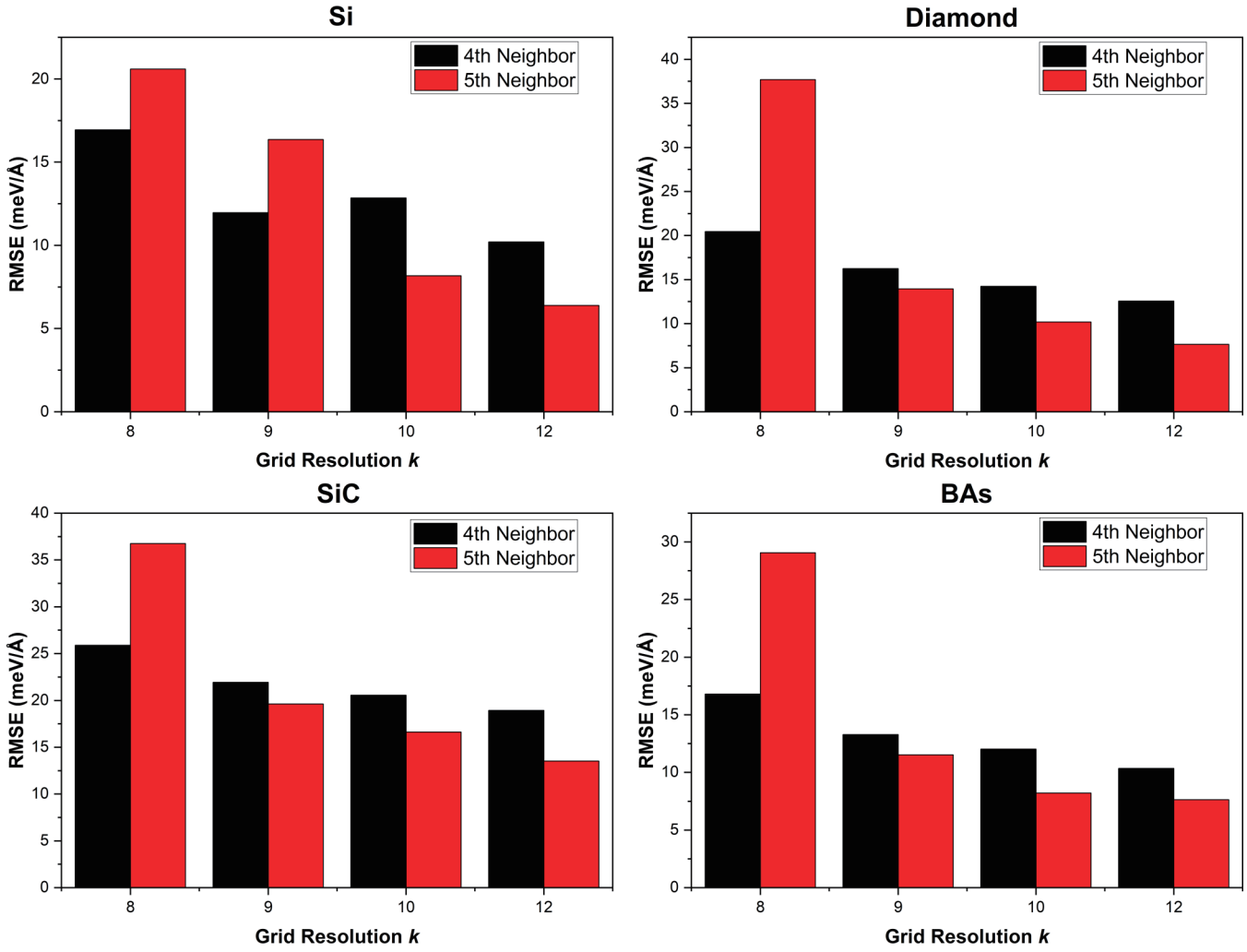

FIG. 8. The resulting testing RMSE values from the material-specific SDNNFFs at various atomic neighbor cutoffs and grid resolutions.

Because of the previous disagreement with the thermal conductivity via FDM, we prioritized our analysis on the conductivity and show this first in Fig. 9. Here, we truncated the converged solutions to DFT-FDM results for display purposes, and only show one curve for each neighbor cutoff. Plots that are not shown for a given combination of neighbor cutoff and k have either equivalent or worse agreement to DFT-FDM. Afterwards, the phonon dispersions corresponding to the shown thermal conductivity curves are plotted against DFT-FDM dispersions in Fig. 10. For the phonon dispersion of SiC, we applied a nonanalytical term correction due to its known LO-TO splitting near the gamma point [39]. We also provide experimental data points in Figures 9 and 10 for comparison except for the thermal conductivity of BAs, which requires four-phonon scattering theory to correct the overestimated results here [40]. We also provide the force line comparison plots for the same SDNNFF networks in Fig. 3 of the Supplemental Material [36].

IV. DISCUSSION

As seen from Fig. 9, we obtained excellent agreement in the lattice thermal conductivity from the SDNNFF forces with the ALM IFC calculator. A maximum percent difference of 0.7% for the 5–10 Si network, 6.2% for the 5–9 diamond network, 2.76% for the 4–10 SiC network, and 7.46% for the 5–10 BAs network in comparison to the DFT-FDM curves was achieved. Considering BAs as the highest percent error for the thermal conductivity, we decided to also plot DFT-ALM to see if the percent error originates from the IFC calculator (FDM vs ALM) rather than the atomic force accuracy (DFT vs SDNNFF). Here we ran 300 ALM structures for BAs instead of the previous 1000 due to the demanding number of DFT calculations and the expectation that fewer structures are required for fitting the force constants with accurate DFT forces. The resulting thermal conductivity from DFT-ALM was slightly lower than that of DFT-FDM and improved the agreement with SDNNFF-ALM to 4.81%. While we do not analyze the effects of the force-constant calculators on the thermal conductivity here, the comparison between FDM and ALM shows that the disagreement is not purely due to the SDNNFF noise. Other sources of error that may have contributed to the resulting percent difference in the SDNNFF thermal conductivity include the number of reference DFT structures in the training set, the temperature range of the generated AIMD structures, the neural network training parameters (e.g., number of layers, nodes, activation functions, etc.), the number of random ALM structures, and

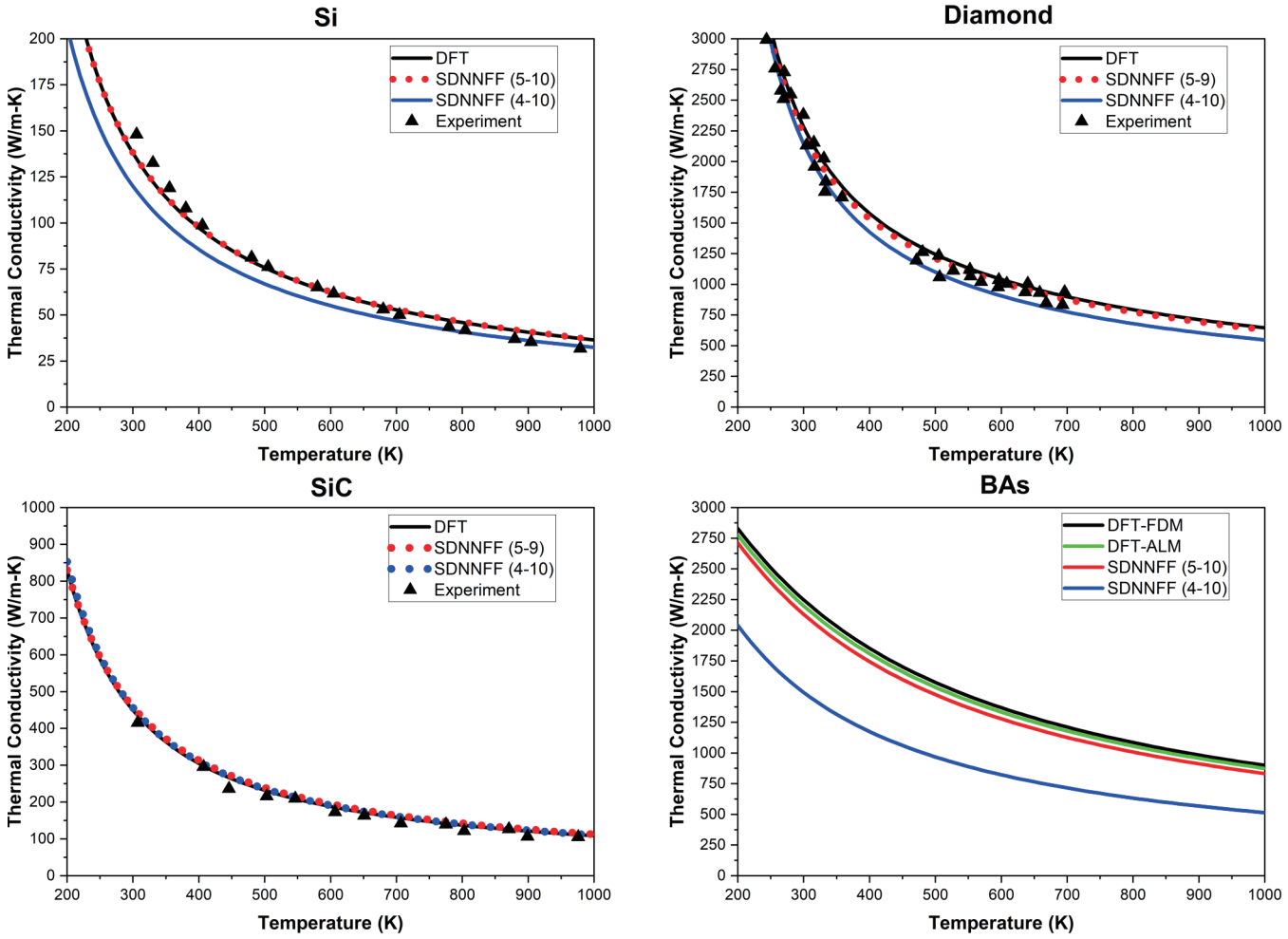

FIG. 9. The temperature-dependent lattice thermal conductivities of all four diamond-structure materials studied here. For curves computed from SDNNFF, the notation is the neighbor cutoff then the grid resolution k (e.g., 5–10 means a network with fifth-neighbor cutoff with $k = 10$). All DFT curves were computed from FDM and all SDNNFF curves were computed from ALM unless otherwise specified. Experimental results for Si [41], diamond [42], and SiC [43,44] are shown as symbols.

the amplitude of the random displacement. However, here we focused on providing a generalization of the SDNNFF performance, and as such did not share excessive concern for tuning these parameters for each material. We only focused on tuning the SDNNFF parameters (i.e., the neighbor cutoff, the grid resolution, and the local cutoff factor) to showcase its potential to successfully reproduce the conventional DFT-FDM and experimental results. Given the various known sources of errors, we were satisfied with the less than 10% error in thermal conductivity and the agreement with both DFT-FDM and experimental trends.

From the predicted phonon properties, we see that the combination of the number of neighbors and the grid resolution is not trivial and has serious impact when reproducing DFT results. Here, the resolution represents the number of inputs to the neural network, i.e., a higher number of grids equals more information for the network to identify the atomic environment and the corresponding central-atom forces. However, we can see that the number of inputs required for the network to sufficiently learn the forces reaches an upper limit, since k values greater than what were provided for each material in Fig. 9 do not improve the results. In other words, the additional information obtained from higher grid resolutions becomes redundant for predicting the thermal conductivities accurately, even though the network RMSE continues to drop as seen in Fig. 8. On the other hand, the number of neighbors provides physical meaning to the neural network, where atoms beyond the cutoff are neglected in the contributions to the central-atom forces. At a minimum, the cutoff should be large enough to where atomic contributions outside the cutoff are negligible or otherwise inconsequential for property prediction. We noticed for the thermal conductivity that the fifth-neighbor contribution is quite significant in Si, diamond, and BAs. However, the fourth neighbors seem to have nearly equal contributions to the thermal conductivity in SiC. For the phonon dispersions, the fifth-neighbor contributions for Si are necessary to match DFT as seen by limited agreement in the fourth neighbors with the low group velocity near the Brillouin-zone center. Otherwise, the phonon dispersions truncated at the fourth neighbors for the other materials are relatively close to the DFT-FDM dispersions. Overall, the SDNNFF parameters R_c and k seem nontransferrable between materials and require further convergence testing for optimal network performances.

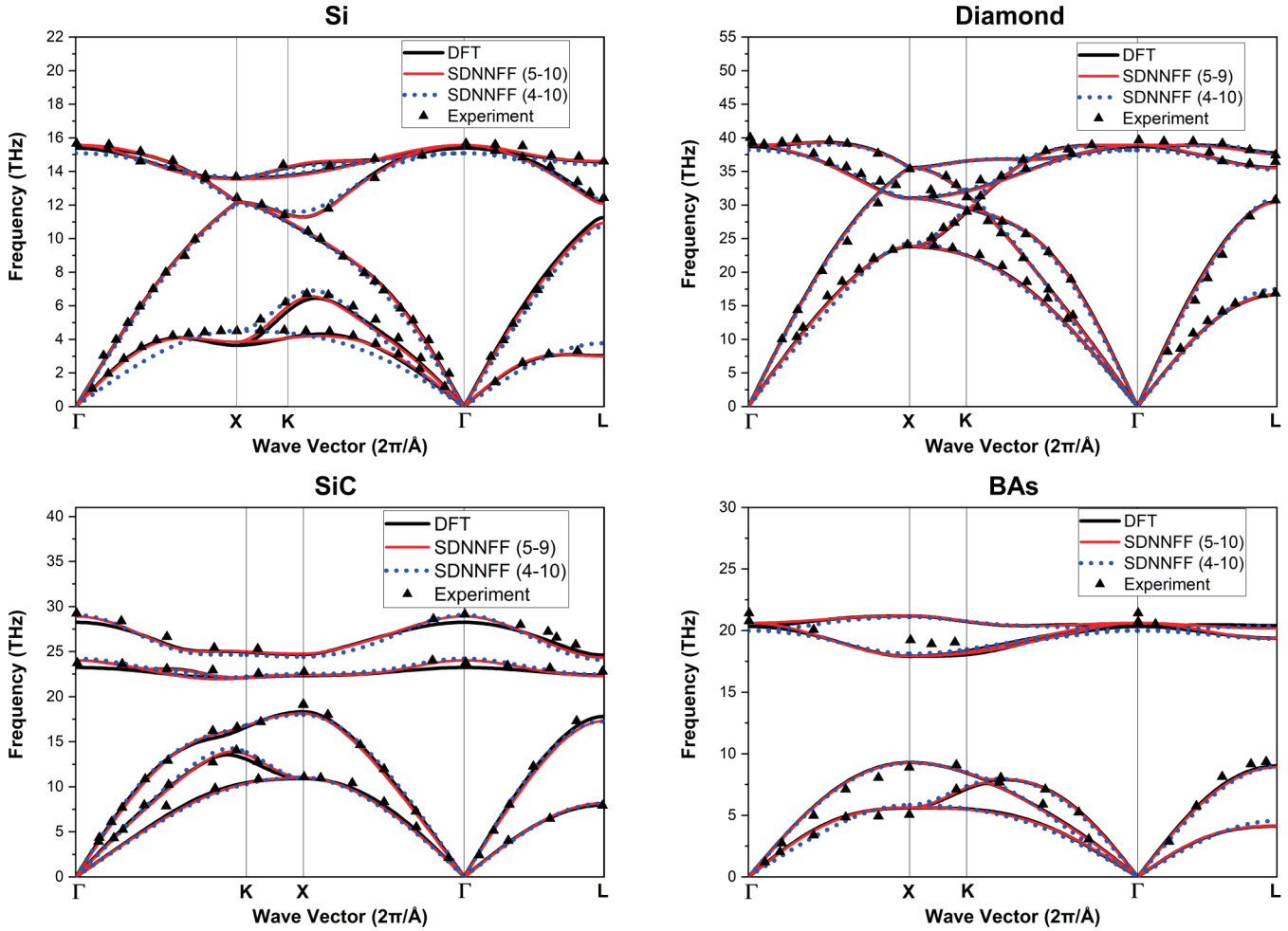

FIG. 10. The phonon dispersions of all four diamond-structure materials studied here. All DFT curves were computed from FDM and all SDNNFF curves were computed from ALM. Experimental results for Si [45], diamond [46–48], SiC [49,50], and BAs [51,52] are shown as symbols.

Summarizing the performance of our method, a trained SDNNFF is quite capable of reproducing DFT-level and experimental phonon properties with significantly lower cost compared to full *ab initio*. For comparison with other papers, the RMSE for the Si network containing the minimum neighbor cutoff and resolution in agreement with DFT-FDM phonon properties was 8.17 meV/Å, while others reported values as low as 25.5 meV/Å [30]. The well-tuned predictive capabilities of the SDNNFF is a result of the direct training on forces, allowing the use of 3D-distributed radial density functions in contrast to arguably more complex descriptors such as a simultaneous feed of radial and angular symmetry functions. However, the SDNNFF, like any other machine learned model for atomic interactions, requires both the generation of the *ab initio* data and the training of the neural network before evaluating the force constants, which is arguably slower than by conventional DFT-FDM for the simple periodic systems like the ones studied here. We recognized that the SDNNFF method would be more valuable in situations where FDM requires a higher number of displaced structures than that required for training and DFT becomes computationally expensive. ALM is one example that requires the evaluation of a potentially high number of randomly displaced structures for converged force constants. Here, we found the phonon properties of simple crystals to show the effectiveness of the SDNNFF in reproducing these properties from electronic structure calculations and experiment. Once the training data is generated and the network is trained, the speed for calculating the properties is significantly faster than that of DFT by the order of $\sim 10^3$.

Currently, the SDNNFF detects atomic environments purely based on spatial configurations, i.e., no chemical information is included in the inputs. Based on this, we were curious how the performance of the SDNNFF was acceptable for materials with more than one atom type, i.e., SiC and BAs, since each species may contribute to the central atom forces distinctly. We decided to analyze the SiC network in further detail by utilizing a recent tool in machine learning, *t*-distributed stochastic neighbor embedding (*t-sne*) [53]. While details about *t-sne* can be found in the referenced paper, in short, a *t-sne* plot reduces high-dimensional data into a 2D/3D set of points, where the proximity of the points defines their correlation (i.e., clustered points represents high similarity). Because we wanted to observe how the neural network was

"thinking" in response to the local atomic configurations to produce the forces, we applied *t-sne* to the final hidden layer (i.e., before the force output) containing 500 nodes given the SiC testing set. For brevity, we provide the resulting plot in Fig. 4 of the Supplemental Material [36], which contains two significant clusters of points. We found that each cluster corresponds to configurations with central atoms corresponding to an atom inside the primitive cell, one Si and one C atom, which is intuitive since the primitive cell is the smallest unit needed to reproduce lattice symmetry and its unique atomic environments. For comparison, we also generated a *t-sne* plot for monoatomic Si (not shown) and again found two clusters, confirming our observations about the unit-cell representation trained into the SDNNFF. From these results, we were led to believe that the SDNNFF was able to successfully reproduce the atomic forces for materials like SiC and BAs because the chemical environment is *consistent* with the structure, where each atomic configuration is always comprised of specific elements during training. For example, if we were to provide a set of inputs to the neural network corresponding to a central Si atom, then the network would recognize that the second neighbors correspond to C, the third neighbors with Si, etc., and correctly identify the forces associated with the values in one of the two *t-sne* clusters. We realize that each density function in space is consistently receiving the *same signals* from the *same atoms,* so the neural network learns indirectly what chemical species is contained in each density function. Based on this observation, we expect the performance of the SDNNFF will degrade when structure and species become independent (i.e., in alloy or multispecies amorphous material). In this case, each *t-sne* cluster would only preserve the local structure and contain a mixture of unknown elements, thereby further confusing the network. We are currently working on inserting chemical information into our density functions for future use in complex atomic environments.

The primary difference between the SDNNFF and other methods like HDNNP is the training on the atomic forces rather than the total energy. This brings about an advantage when training neural networks since the atomic forces are per-atom quantity, unlike the total energy which is a global value. Here, the DFT data for network training scale with $N \times t$, where N is the number of atoms in a supercell and t is the number of structures to be evaluated. In our case, we had 7500 structures from AIMD to be evaluated by static DFT for each material, and for a $3 \times 3 \times 3$ primitive diamond supercell with 64 atoms, the resulting dataset contained $7500 \times 64 = 4.8 \times 10^5$ atomic configurations per material. This is unlike training a network on the total energy, where the data set only scales with t (one total energy per structure) and therefore may require additional DFT calculations to generate a sufficiently large dataset for optimal neural network performance. We recognize other recent works that have provided schemes to take full advantage of the data from DFT, such as the extraction of and training on atomic energies from DFT by Huang *et al.* [26], the inclusion of both total energies and atomic forces in the loss function in training the HDNNP by Minimitani *et al.* [30], and the additional inclusion of stress tensor by Marques *et al.* [54]. However, in the latter two cases, atomic quantities such as the force are still treated as a single update in the HDNNP since the loss function involves an *averaging* across all atoms, which is contrary to our loss function in Eq. (6) where the network is updated based on each *individual* atomic force [27]. This becomes a drawback for network training in networks like the HDNNP since vital information for the force minimization is lost with an update through one single average and the new direction for the weights/biases during training becomes less precise. In contrast, the SDNNFF preserves the $N \times t$ nature by maintaining a loss function dependent solely on each atomic force vector, and the network learns from a more robust set of examples. In addition, our networks require no additional postprocessing of the DFT data since the forces are directly output from DFT, and no intermediate calculations for the loss function are necessary since Eq. (6) only has dependence on the output forces. In other words, the training on forces in the SDNNFF not only increases the dataset size obtained from DFT calculations by explicit and independent training on each individual case, but also serves as a potential computational cost savings for training data generation and MD force calculations.

In addition to the force accuracy, another advantage of the SDNNFF is that it is an "out-of-the-box" method, where the description of the atomic environment can be controlled by only three parameters: the grid resolution, the atomic cutoff, and the local cutoff factor. As mentioned in Sec. II D, while we recommend performing a convergence test for the cutoff factor D, it was observed that a converged D value in one material is highly transferable to other structurally equivalent materials as seen in Fig. 3. This makes the training of the forces very intuitive by reducing the network search to the other two parameters: the grid resolution and the atomic cutoff. Compared to the $2 \times i$ parameters for the type-2 radial symmetry functions plus $3 \times j$ parameters for the type-4 angular symmetry functions along with the atomic cutoff in the HDNNP method (i and j are the number of radial and angular symmetry functions respectively, also to be determined), the simplicity of the SDNNFF makes the generation of DFT force fields potentially less laborious than some existing methods [55].

Further comparing with the HDNNP, the disadvantage of the SDNNFF is that because of the equal spatial distribution of density functions in the atomic environment, a larger number of inputs may be required to cover all the important features of the atomic forces. For Si, we used at minimum 360 nodes ($k = 10$) to accurately predict its thermal conductivity, while Minamitani *et al.* obtained competitive results with only 41 inputs using the HDNNP [30]. Because the HDNNP owns more tunability, the method can obtain the feature space more efficiently than SDNNFF, requiring a smaller number of descriptors for the neural network to identify atomic configurations. Additionally, calculation of NNP/NNFF inputs requires $M \times N$ distances computed, where M is the number of radial (symmetry/density) functions and N is the number of atoms inside the cutoff, which is a disadvantage for the SDNNFF since M scales with the number of points in xyz space and not radial space (i.e., calculating distances between $(\pm x, \pm y, \pm z)$ points and all neighbors vs radial points with respect to the central atom and all neighbors). However, we should point out that the discussion about $M \times N$ points neglects the evaluation of additional angular symmetry functions, which is not needed

in the SDNNFF since the atomic information is already represented in Cartesian space. Furthermore, as we mentioned previously, the customizability of the feature space in the HDNNP requires an increased effort to optimize its handful of parameters and is slowed down by the required total energy to evaluate the atomic forces, but on the other hand the evaluation of the HDNNP itself is quicker due to the smaller number of inputs and required neighbor distances to calculate. Therefore, at this stage it is difficult to judge which method is faster for application in MD, and we plan on comparing these two methods in future work. For now, however, we can say that the SDNNFF is capable of directly predicting the atomic forces and the resulting phonon properties of various materials with DFT accuracy, providing an emphasis on force accuracy via inclusion of 3D descriptors and inheriting a more straightforward implementation strategy for representing the atomic environment.

In comparison to classical potentials, which provides both the per-atom energies and the forces, the SDNNFF presented herein can only provide the atomic forces so far. However, we decided to use the training, validation, and testing data generated and made available by Huang *et al.* to predict both forces and atomic energies of amorphous silicon using our SDNNFF method [26]. Because of the flexibility of neural networks, we were able to generate a SDNNFF+E, where we output both the force components and the atomic energies. We also generated a SDNNFF with only forces to compare with the SDNNFF+E. Briefly covering the details, we closely followed the parameters chosen by Huang *et al.*, including a cutoff of 6 Å, 360 input features (i.e., $k = 10$), which is on the same order of magnitude as their 550 features, $D = \sqrt{3}$, two hidden layers with 500 nodes, training up to 250 cycles, and a similar loss function defined as $\text{Loss} = \text{RMSE}(\{E_i\}) + \alpha \times \text{RMSE}(\{F_i\})$ (note that we arbitrarily used $\alpha = 1$, as well as RMSE instead of MSE). The resulting RMSE from the SDNNFF+E were 47.29 meV/Å and 15.31 meV for the forces and energies, respectively, while the RMSE from the SDNNFF was 42.94 meV/Å, which was slightly more accurate in exchange for no atomic energy information. In comparison to the SANNP, the SDNNFF+E obtained an unprecedented ∼50% and ∼70% improvement in the forces and atomic energies, respectively. We realized that the method presented here was flexible for predicting the atomic energies, although we should note that the energy is not completely invariant to spatial transformations due to the 3D representation of atomic environments. Future work should determine whether this significantly improved RMSE justifies the inclusion of rotationally variant energy into the SDNNFF for, e.g., bulk Green-Kubo thermal conductivity calculations. Nonetheless, the improvement in both force and energy RMSE is testament for the improved description of the atomic environment via 3D density functions in contrast to radial and angular symmetry functions. Moreover, we recognize that the evaluation of thermal conductivity through the nonequilibrium molecular dynamics (NEMD), such as the Müller-Plathe method, is possible with only the forces available, since the atomic kinetic-energy exchange is used for launching the heat flux in the Fourier's law and does not involve the atomic potential energy [56]. Overall, further study is required to bridge neural network force fields with thermal conductivity calculations in MD, with the intent to model the conductivity including all levels of phonon anharmonicity.

V. CONCLUSION

We have developed and trained a neural network force field, dubbed "SDNNFF," capable of reproducing DFT forces and phonon properties of materials by transformation of the local atomic environment through density functions. Here, each atom was surrounded by a grid containing equidistant points within a cutoff radius. We then measured the local density of atomic neighbors at each point via a summation of radial cosine functions across all neighbors. The resulting set of density function values across all grid points led to distinguishable atomic environments for neural networks to successfully predict DFT-level force components as well as phonon dispersions and lattice thermal conductivities.

In the realm of modeling interatomic interactions through machine learning models, three achievements are accomplished in this paper. First, our neural networks are directly trained on the DFT atomic forces without the need to compute derivatives from the total energy. The removal of intermediate calculations for the atomic forces simplifies the method, where a single neural network can directly predict atomic forces without the need to impose the chain rule involving the total energy and the atomic positions. Without the total energy, a direct 3D representation of the environment provides a more explicit description of force fields in comparison to that from radial and angular descriptors, as shown from our consistently improved force RMSE. If the atomic energy is needed, the SDNNFF can output this along with the forces, although the calculation for the atomic energy is relatively new and, according to our knowledge, not yet standardized in DFT packages such as VASP. Additionally, the treatment of the atomic neighbors in three dimensions prevents the SDNNFF from preserving the rotational invariance of the atomic energy, although the improved accuracy of the SDNNFF may or may not generate negligible differences in energy. Second, training on atomic quantities such as forces allows the dataset to scale largely with $N \times t$, the atomic population of the structure times the number of evaluated DFT steps. Training on such a tremendously large data set strengthens the prediction capabilities of neural networks for the detection of a wide range of atomic structures one may encounter during dynamics. Consequentially, maximizing the data collected from costly DFT calculations by training the forces on a case-by-case basis becomes advantageous for neural network training. Third, we found a significant reduction in human/computational effort in the search for converged neural network force fields by representation of the atomic environment with only three parameters: the atomic cutoff radius, the grid resolution, and the local density cutoff factor. The cutoff factor D may own transferability with similar structures and thus even reduce the network search down to two parameters, but transferability needs to be shown in the future for structures other than diamond structure (i.e., BCC, hexagonal, etc.).

We conclude this paper with a brief discussion on future applications of the SDNNFF. While the SDNNFF is capable of computing phonon properties with DFT-level accuracy,

the speed of the SDNNFF is not utilized to its fullest potential. As mentioned before, for the above simple systems the calculation of the phonon properties with DFT-FDM is much faster than SDNNFF-ALM since the latter requires one to generate DFT data, train the network, and finally fit the force constants with the evaluated random structures before computing the phonon properties. The SDNNFF is more beneficial in applications where large systems, in particular inhomogeneous systems like amorphous materials or interfaces, and/or long time steps are required for force evaluation, which cannot be handled by DFT due to unbearable computational cost. In other words, with a well-trained SDNNFF, large scale quasielectronic MD simulations become possible. The DFT-level force predictions from neural networks such as the SDNNFF are critical for the analysis of microscale systems where direct DFT becomes unreachable and classical MD falls short in realistically modeling these systems. In addition to thermal conductivity, applications of SDNNFF in MD simulations also include the evaluation of other dynamical properties such as diffusion coefficients and ionic conductivity, mechanical deformation, and other observables such as radial distribution function. As such, our next step for the SDNNFF is implementation into MD simulation package [e.g., the [large-scale atomic/molecular massively parallel simulator (LAMMPS)] and investigating large-scale dynamics of systems of interest, such as lithium ion diffusion in crystalline and amorphous silicon [57], or the bulk thermal conductivity of materials and interfacial heat transfer via Müller-Plathe method, at improved speeds by directly computing DFT-level forces, as well as finding other ways to improve the computational efficiency of the method (e.g., the reduction of network inputs or the explicit consideration of atomic species in the neural network as seen in recent studies [58,59]).

ACKNOWLEDGMENTS

A.R. was financially supported by the NASA South Carolina Space Grant Consortium for the Minorities in STEM fellowship and the DOE fellowship for the Integrated University Program (IUP). Research reported in this publication was supported in part by the National Science Foundation (NSF) (Award No. 1905775) and SC EPSCoR/IDeA Program under NSF OIA-1655740 via SC EPSCoR/IDeA 20-SA05. We deeply thank Dr. E. Minamitani for her insight on the application of neural networks in atomic simulations, and Dr. A. Togo for his explanations on ALM and help with compiling phonopy/3py containing the ALM package. Personal thanks from A.R. to Dr. G. Qin (Hunan University, China) for fruitful discussions on DFT calculations of the structures.

[1] M. Salanne, B. Rotenberg, S. Jahn, R. Vuilleumier, C. Simon, and P. A. Madden, Theor. Chem. Acc. **131**, 1143 (2012).
[2] C. J. Cramer, *Essentials of Computational Chemistry: Theories and Models* (John Wiley & Sons, Chichester, 2017).
[3] T. Xie, A. France-Lanord, Y. Wang, Y. Shao-Horn, and J. C. Grossman, Nat. Commun. **10**, 2667 (2019).
[4] L. M. Raff, M. Malshe, M. Hagan, D. I. Doughan, M. G. Rockley, and R. Komanduri, J. Chem. Phys. **122**, 84104 (2005).
[5] R. O. Jones, Rev. Mod. Phys. **87**, 897 (2015).
[6] J. H. Jensen, *Molecular Modeling Basics* (CRC, Boca Raton, FL, 2010).
[7] G. Qin, Z. Qin, H. Wang, and M. Hu, Phys. Rev. B **95**, 195416 (2017).
[8] D. E. Galvez-Aranda, V. Ponce, and J. M. Seminario, J. Mol. Model. **23**, 120 (2017).
[9] Y. Liu, T. Zhao, W. Ju, and S. Shi, J. Materiomics **3**, 159 (2017).
[10] J. Schmidhuber, Neural Netw. **61**, 85 (2015).
[11] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, APL Mater. **1**, 11002 (2013).
[12] I. Tanaka, K. Rajan, and C. Wolverton, MRS Bull. **43**, 659 (2018).
[13] A. D. Sendek, E. D. Cubuk, E. R. Antoniuk, G. Cheon, Y. Cui, and E. J. Reed, Chem. Mater. **31**, 342 (2018).
[14] H. Yang, Z. Zhang, J. Zhang, and X. C. Zeng, Nanoscale **10**, 19092 (2018).
[15] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, npj Comput. Mater. **5**, 83 (2019).
[16] L. Zhang, J. Han, H. Wang, R. Car, and W. E, Phys. Rev. Lett. **120**, 143001 (2018).
[17] B. Onat, E. D. Cubuk, B. D. Malone, and E. Kaxiras, Phys. Rev. B **97**, 094106 (2018).
[18] J. S. Smith, O. Isayev, and A. E. Roitberg, Chem. Sci. **8**, 3192 (2017).
[19] N. Artrith and J. Behler, Phys. Rev. B **85**, 045439 (2012).
[20] W. Li, Y. Ando, E. Minamitani, and S. Watanabe, J. Chem. Phys. **147**, 214106 (2017).
[21] B. Klein and D. Rossin, Omega **27**, 569 (1999).
[22] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[23] A. Singraber, J. Behler, and C. Dellago, J. Chem. Theory Comput. **15**, 1827 (2019).
[24] K. Parlinski, Z. Q. Li, and Y. Kawazoe, Phys. Rev. Lett. **78**, 4063 (1997).
[25] T. Tadano and S. Tsuneyuki, J. Phys. Soc. Jpn. **87**, 41015 (2018).
[26] Y. Huang, J. Kang, W. A. Goddard, and L.-W. Wang, Phys. Rev. B **99**, 064103 (2019).
[27] J. Behler, Int. J. Quantum Chem. **115**, 1032 (2015).
[28] S. van der Walt, S. C. Colbert, and G. Varoquaux, Comput. Sci. Eng. **13**, 22 (2011).
[29] M. Abadi, M. Isard, and D. G. Murray, in *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages - MAPL 2017* (ACM Press, New York, 2017), pp. 1–7.
[30] E. Minamitani, M. Ogura, and S. Watanabe, Appl. Phys. Express **12**, 95001 (2019).
[31] K. He, X. Zhang, S. Ren, and J. Sun, in *2015 IEEE International Conference on Computer Vision (ICCV)* (IEEE, Piscataway, NJ, United States, 2015), pp. 1026–1034.
[32] D. P. Kingma and J. Ba, arXiv:1412.6980.

[33] D. Clevert, T. Unterthiner, and S. Hochreiter, arXiv:1511.07289.
[34] A. Togo and I. Tanaka, Scr. Mater. **108**, 1 (2015).
[35] A. Togo, L. Chaput, and I. Tanaka, Phys. Rev. B **91**, 094306 (2015).
[36] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevB.102.035203 for more information on the artificial noise testing in the Si phonon dispersion and lattice thermal conductivity, the force comparison plots between DFT and the SDNNFF for the materials studied here, and the *t-sne* plot for SiC obtained from the SDNNFF network.
[37] A. J. H. McGaughey, A. Jain, and H.-Y. Kim, J. Appl. Phys. **125**, 11101 (2019).
[38] S. Cottenier, *Density Functional Theory and the Family of (L)APW Methods:A Step-by-step Introduction, 2002 –2013*, 2nd ed, http://susi.theochem.tuwien.ac.at/reg_user/textbooks/DFT_and_LAPW_2nd.pdf.
[39] J. Serrano, J. Strempfer, M. Cardona, M. Schwoerer-Böhning, H. Requardt, M. Lorenzen, B. Stojetz, P. Pavone, and W. J. Choyke, Mater. Sci. Forum **433–436**, 257 (2003).
[40] J. S. Kang, M. Li, H. Wu, H. Nguyen, and Y. Hu, Science **361**, 575 (2018).
[41] C. Y. Ho, R. W. Powell, and P. E. Liley, J. Phys. Chem. Ref. Data **1**, 279 (1972).
[42] D. G. Onn, A. Witek, Y. Z. Qiu, T. R. Anthony, and W. F. Banholzer, Phys. Rev. Lett. **68**, 2806 (1992).
[43] A. Sparavigna, Phys. Rev. B **66**, 174301 (2002).
[44] R. E. Taylor, H. Groot, and J. Ferrier, Thermophysical Properties Research Laboratory Report No. TRPL 1336, 1993.
[45] J. Kulda, D. Strauch, P. Pavone, and Y. Ishii, Phys. Rev. B **50**, 13347 (1994).
[46] J. L. Warren, J. L. Yarnell, G. Dolling, and R. A. Cowley, Phys. Rev. **158**, 805 (1967).
[47] M. Schwoerer-Böhning, A. T. Macrander, and D. A. Arms, Phys. Rev. Lett. **80**, 5572 (1998).
[48] J. Kulda, H. Kainzmaier, D. Strauch, B. Dorner, M. Lorenzen, and M. Krisch, Phys. Rev. B **66**, 241202(R) (2002).
[49] J. Serrano, J. Strempfer, M. Cardona, M. Schwoerer-Böhning, H. Requardt, M. Lorenzen, B. Stojetz, P. Pavone, and W. J. Choyke, Appl. Phys. Lett. **80**, 4360 (2002).
[50] F. Widulle, T. Ruf, O. Buresch, A. Debernardi, and M. Cardona, Phys. Rev. Lett. **82**, 3089 (1999).
[51] H. Ma, C. Li, S. Tang, J. Yan, A. Alatas, L. Lindsay, B. C. Sales, and Z. Tian, Phys. Rev. B **94**, 220303(R) (2016).
[52] R. G. Greene, H. Luo, A. L. Ruoff, S. S. Trail, and F. J. DiSalvo, Phys. Rev. Lett. **73**, 2476 (1994).
[53] L. van der Maaten and G. Hinton, J. Mach. Learn. Res. **9**, 2579 (2008).
[54] M. R. G. Marques, J. Wolff, C. Steigemann, and M. A. L. Marques, Phys. Chem. Chem. Phys. **21**, 6506 (2019).
[55] J. Behler, J. Chem. Phys. **134**, 74106 (2011).
[56] F. Müller-Plathe, J. Chem. Phys. **106**, 6082 (1997).
[57] H. Wang, X. Ji, C. Chen, K. Xu, and L. Miao, AIP Adv. **3**, 112102 (2013).
[58] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, J. Chem. Phys. **148**, 241709 (2018).
[59] A. S. Bochkarev, A. van Roekeghem, S. Mossa, and N. Mingo, Phys. Rev. Mater. **3**, 093803-1 (2019).