# Machine Learning for Accurate Force Calculations in Molecular Dynamics Simulations

Punyaslok Pattnaik,[†] Shampa Raghunathan,[†] Tarun Kalluri,[‡] Prabhakar Bhimalapuram,[†] C. V. Jawahar,[‡] and U. Deva Priyakumar[*,†]

[†]*Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Hyderabad 500 032, India*

[‡]*Center for Visual Information Technology, KCIS, International Institute of Information Technology, Hyderabad 500 032, India*

E-mail: deva@iiit.ac.in

## Abstract

The computationally expensive nature of *ab initio* molecular dynamics simulations severely limits its ability to simulate large system sizes and long time scales, both of which are necessary to imitate experimental conditions. In this work, we explore an approach to make use of the data obtained using the quantum mechanical density functional theory (DFT) on small systems and use deep learning to subsequently simulate large systems by taking liquid argon as a test case. A suitable vector representation was chosen to represent the surrounding environment of each Ar atom, and a $\Delta$-NetFF machine learning model where, the neural network was trained to predict the difference in resultant forces obtained by DFT and classical force fields was introduced. Molecular dynamics simulations were then performed using forces from the neural network for various system sizes and time scales depending on the properties we calculated. A comparison of properties obtained from the classical force field and the neural network

model was presented alongside available experimental data to validate the proposed method.

# Introduction

The modeling of a condensed phase system involving chemical processes spanning multiple time and length scales is particularly challenging. *Ab initio* molecular dynamics (AIMD) which explicitly treats electronic degrees of freedom is naturally the first method of choice, however computationally demanding, thus prohibiting its application to large molecular systems.[1,2] Classical molecular dynamics (MD) simulations employing force fields can do a proper sampling of the phase space of large systems (up to million atoms),[3,4] but underlying interatomic potentials are often not accurate enough to obtain quantitavely accurate results. Their transferability to situations that were not originally used in the parameterization is questionable, which further limits their accuracy.

The need to construct a multiscale model (considering electronic, nuclear dynamics[5–7] and their coupling to slower, cooperative motions of the system) to capture accurate dynamics of chemical processes cannot be overstated.[8–10] The fundamental question is: Can one quantify the relevance of atomistic models to electronic interactions employing any numerical formalism and how corresponding MD errors reflect emergent features in *ab initio* driving forces? One of the most successful approaches relies on quantum mechanical (QM) calculations on gas-phase (sometimes considering the implicit solvent model) clusters to parameterize a model meant for bulk phase simulations. Another empirical procedure is based on the minimization of a "loss function" or "objective function" between simulated and experimental physical properties.

With the increasing availability of computational resources and data, machine learning (ML) techniques have been popularly applied to predict quantum mechanical properties.[11–17] A plethora of sophisticated ML approaches exist: For predicting ground state energies, ap-

proaches, such as, boosted regression tree algorithms,[18] high-dimensional neural network potential energy surfaces using symmetry functions,[11] continuous-filter convolutional layers[17] and single-atom atomic environment vectors (AEV)[13] have been used. Atomization energies for molecules have been predicted based on nuclear charges and atomic positions only.[19] Multiple electronic, ground, and excited-state properties have also been predicted simultaneously using Coulomb matrices in conjunction with deep multi-task artificial neural networks.[14,20] The bag-of-bonds model was used to predict accurate electronic properties of molecules, such as, their polarizability and molecular frontier orbital energies.[21,22] Using artificial neural networks (ANNs), energies of molecules have also been predicted as a sum of intrinsic bond energies, while also providing valuable insight into the relative strengths of bonds as a function of their molecular environment.[16] ANNs have also been used along with genetic algorithm (GA) optimization to discover unconventional spin-crossover complexes, which emphasizes their power for discovering new inorganic materials.[23] Recently an ML model was proposed where a novel molecular descriptor inspired by classical force fields terms – bonds, angles, non-bonded interactions and dihedrals to perform geometry optimizations along with predicting their energies.[24] This model employs feed-forward fully connected deep neural networks. Graph neural networks were used to predict solvent-solute interaction map[25] for studying solvation free energies of drug-like molecules/solute. Instead of applying ML techniques to directly compute properties of new molecules through interpolation in chemical compound space, recently, ML of force field parameters was performed for semi-empirical modeling.[26]

In the recent years, machine learning (ML) has emerged as a potential technique for developing a new generation of highly accurate force fields (FFs) for simulations of molecules and materials. Ramprasad and coworkers[27–29] have developed ML-based atomistic force fields for MD simulations. They have mainly focused on bulk solid-state materials. Another approach, on-the-fly ML of QM forces in MD simulations was recently reported by Li and coworkers[30] on bulk Si. The smooth overlap of atomic positions (SOAP) metric has been

used to construct potential energy surfaces, and its performance was evaluated for small silicon clusters.[15] Gaussian approximation potentials have been used to generate trajectories for water dimers, energetics path for a migrating vacancy and the transformation of rhombo-hedral graphite to diamond.[31] Another popularly used class of ML-FFs based on Gaussian process (GP) regression was developed for stduying 19-atom Ni nanocluster[32] as well as adsorption energies of small molecules on NiGa and RhAu nanoclusters.[33,34] Interatomic potentials for metallic aluminium, carbon and dimer potentials for noble gases have been reconstructed using neural networks.[35] The effects of such fitted potentials on the calculation of physical properties obtained from their trajectories, at different physical conditions, such as, temperature and pressure, need to be studied to further reinforce on their future applications. Machine learning is also being successfully used to analyze longtime scale simulation data on large systems.[36,37]

As the area of development of ML-FFs for MD simulations is expanding towards assessing and improving the accuracy and transferability of the model, learning and predicting atomic forces have been receiving notable successes. Because, atomic forces can be seen as true QM observation within the BO-approximation to abide by the Hellmann-Feynman theorem.[38] The energy of a molecular system would then be recovered through appropriate integration of the force-field kernel. ML models are often trained on reference data obtained from QM-based methods, such as, density functional theory (DFT) within the Kohn-Sham formalism. DFT continues to exist as one of the most popular and widely used QM-target from molecular regime[14,19,21,27,39–42] to condensed matter and materials informatics.[11,15,20,31,43–48]

In this article, we explore an approach in which DFT calculations for smaller systems can be used, in conjunction with machine learning to simulate larger systems at a computational effort comparable to classical force fields, while being able to predict forces similar to DFT. A Δ-NetFF model that uses the difference in forces obtained from the molecular mechanical force field and the quantum mechanical DFT approaches to train the NN was introduced for constructing the force field for MD simulations. The predictive power of the present ML

4

model was assessed by calculating dynamical and thermodynamical observables including the liquid-vapor interface, which require sufficient sampling of the configuration space, by employing MD simulations. This method does not require explicit parameterization and does end-to-end prediction of forces using the neural network. One of the main features of this model is that larger systems of multiple sizes can be applied using the same trained model of smaller-sized 96-atom system.

# Methodology

The workflow in constructing the $\Delta$-NetFF model includes: (1) creation of a reference dataset from a large number of diverse atomic environments, and calculating the forces for each environment using DFT method, (2) constructing an input vector for each atom which is then mapped to the resultant force it experiences, (3) learning from the generated data by training a suitable machine learning model to predict forces on each atom, and (4) finally, running large length scale plus long time scale MD simulations using forces predicted from the trained model. The entire workflow is portrayed schematically in Figure 1, which is discussed in detail in the following subsection. As the model learns the difference between a reference force field and DFT forces, we name the method $\Delta$-NetFF . The code for the $\Delta$-NetFF model is freely available from https://github.com/devalab/delNetFF.

## Generating the dataset

The primary system used for generating the data for ML contained 96 argon atoms in a cubic box of size 16.7 x 16.7 x 16.7 Å$^3$. The size of the box was chosen such that the density (1.37 g/cc) was close to that of the experimental density of liquid argon (1.396 g/cc) reported at 87.3 K.[49] We then augmented this dataset with additional structures with sparsely populated argon atoms. To imitate lower densities of argon, the number of atoms was kept the same, and the box size was increased such that, the density decreased in steps of 0.005 g/cc, starting
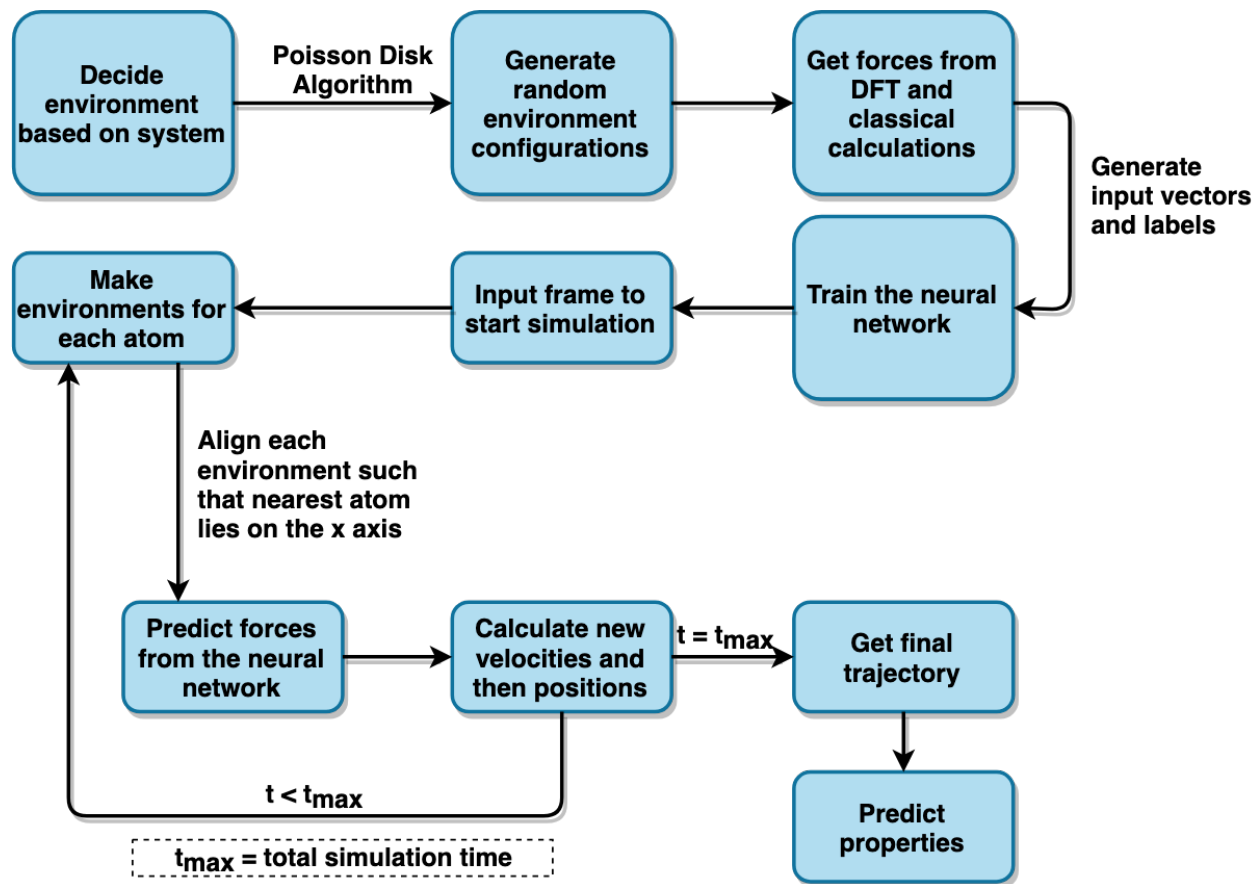
Figure 1: Workflow to train the $\Delta$-NetFF and to perform simulations using the predicted forces.

from 1.37 g/cc upto 0.1 g/cc. This resulted in 255 unique box sizes with a maximum box size of 39.96 x 39.96 x 39.96 $\text{Å}^3$. 2500 frames were generated for every density, which resulted in a total of 637,500 frames.

**Generating frames**

The arrangement of atoms in a frame for a particular box size was obtained using the Poisson disk algorithm.[50] This algorithm is used to pack points inside a $n$-dimensional box where the distance between no two particles should be less than a parameter $r$, called the Poisson disk radius. This algorithm was used because random samples tend to produce clusters and consequently large empty spaces between points. Points generated using uniform random generators guarantee a uniform placement probability of points, but not a uniform coverage of the $n$-dimensional sampling space. Additionally, random sampling makes it very hard to place all points at some distance apart from each other, which is essential in our case to avoid two atoms being too close together resulting in non-physical contacts. A comparison between points generated by a uniform random and the Poisson disk algorithm is shown in Figure 2.

**Calculating forces for each frame**

DFT was chosen as the *ab initio* method to calculate forces because of its speed of calculations, so that a dataset can be generated in a reasonable amount of time. The DFT calculations were performed using the CP2K/Quickstep package[51,52] with the BLYP[53,54] exchange-correlation (XC) functional and the Goedecker-Teter-Hutter (GTH) pseudopotential.[55,56] Each frame had 96 atoms, as described above, and varying cell sizes according to the density. The cell shape was always chosen to be cubic for all densities. The energy cutoff used in the expansion of electron density in plane waves was set to 300 Ry, while the SCF energy convergence threshold was set to $10^{-6}$ Hartree/atom. For optimizing the wave function, the orbital transformation (OT) method[57] was used with the DIIS[58,59] minimizer.

7
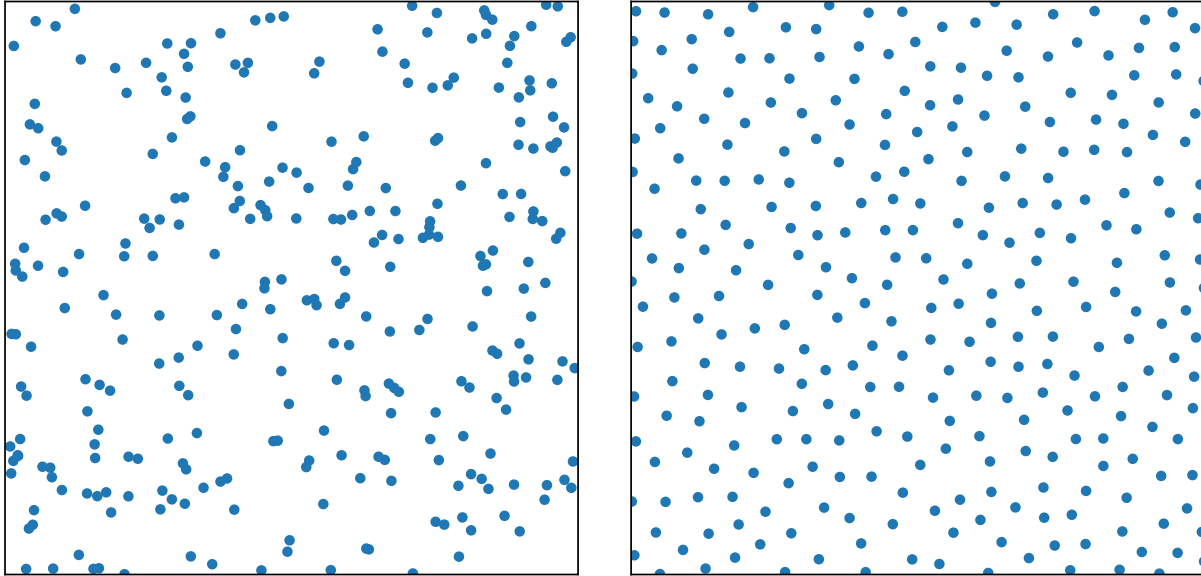
(a) Uniform random samples          (b) Poisson disk samples

Figure 2: Comparison between random and Poisson disk samples for 300 points in a 2-D space.

In addition to the DFT calculations, forces were also obtained using a Lennard-Jones (LJ) argon potential with $\sigma = 3.40$ Å and $\epsilon = 114.99$ K as the Lennard-Jones parameters.[60]

**Construction of the input vector**

Taking each atom as a **central atom** for every frame, we calculated the relative positions of all the other atoms with respect to the central atom in the Cartesian coordinate representation, i.e., the central atom remains at the origin, with all other atoms around it. With respect to the central atom, only the closest image (by applying periodic boundary conditions in all directions) of the surrounding atoms was considered. Then we sort the surrounding atoms with respect to distance from the origin (central atom) and rotate all the coordinates, such that, the nearest atom lies on the positive $X$-axis. The resultant force on the central atom is also rotated by the same amount. In the rare case when there are more than one closest atoms, one of them was randomly chosen for the rotation step.

This rotation step helps the neural network to learn the forces efficiently and predict them more accurately because:

- The state space for all possible configurations is greatly reduced if the nearest atom always lies on the same axis for all the samples. We found this method of representation to be more accurate as compared to including more samples by randomly rotating the configurations as reported before in the Poisson disk algorithm, without aligning the nearest atom.

- The component of the resultant force in the direction of the nearest atom is generally higher than the directions perpendicular to it. This is made clear by Figures 3 and 4 which show the frequency distribution of forces along all axes before and after the rotation step, respectively.

For example, consider a frame $s$ consisting of $N$ atoms, with $j^{\text{th}}$ atom as the central atom. Let $F^s$ be the forces array containing $N$ resultant force vectors. Now, to get feature vector for atom $j$:

1. We get a new coordinate frame $R_i^{s'} = R_i^s - R_j^s \ \forall \ i \in [1, N]$, where $R_i^s$ represents the coordinates of the $i^{\text{th}}$ atom in frame $s$.

2. Discard $R_j^{s'}$(now a zero vector) from $R^{s'}$, as now the central atom lies at the origin.

3. Now we sort $R^{s'}$ based on the Euclidean distance from the origin for each $R_i^{s'}$. As the input to a neural network has to be of a fixed length, we consider only the 50 closest atoms in $R^{s'}$. This is similar to having a distance cutoff used in MD simulations for calculating non-bonded interactions.

4. Rotate the coordinates of all atoms such that the closest atom lies on the positive $X$-axis. This rotation is done along the path which covers the shortest angular distance from the atom's position vector to the positive $X$-axis. While training the model, further rotate the resultant force on atom $j$, i.e. $F_j^s$, by the same amount. This
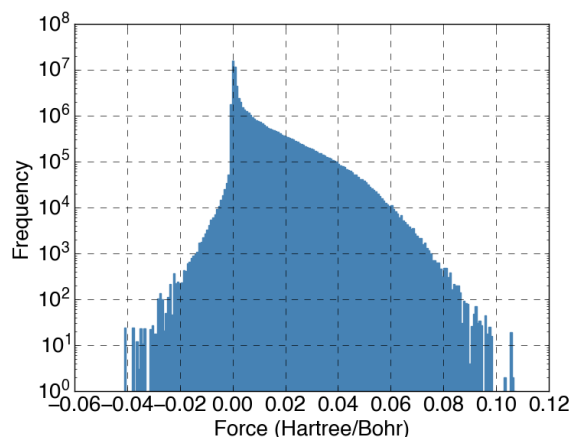
9
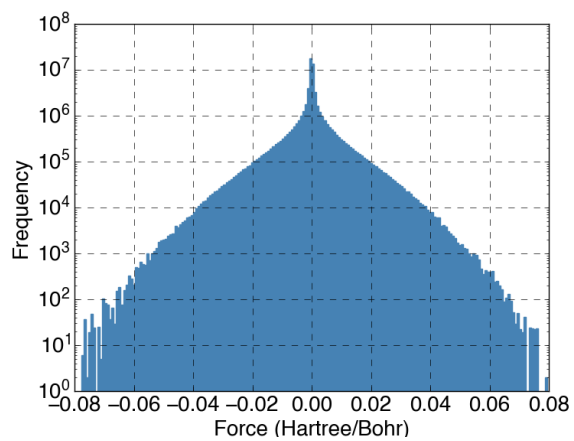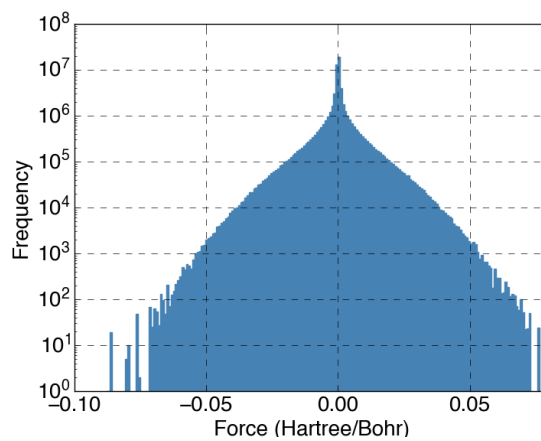
(a) X-component
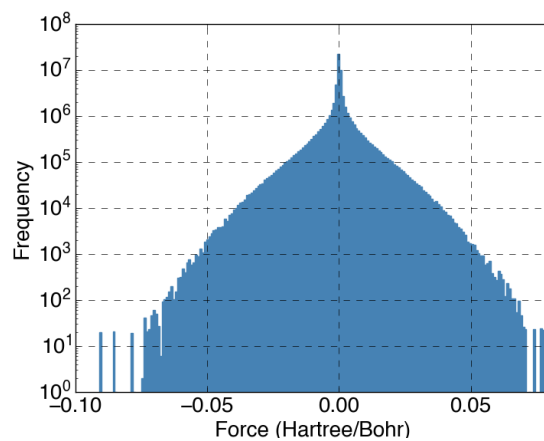

(b) Y-component


(c) Z-component

Figure 3: Distribution of the three components of the force before aligning vector connecting the origin and the nearest atom to the X-axis.

(a) X-component



(b) Y-component
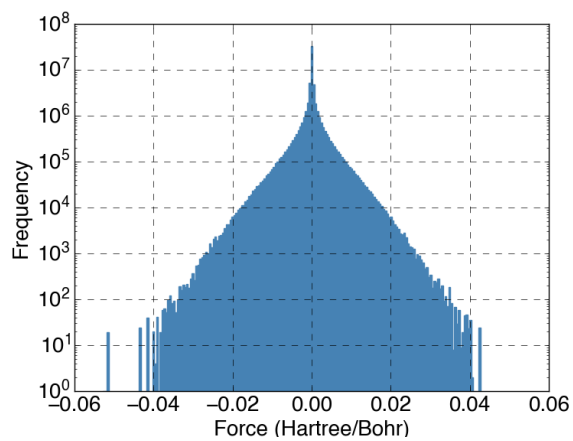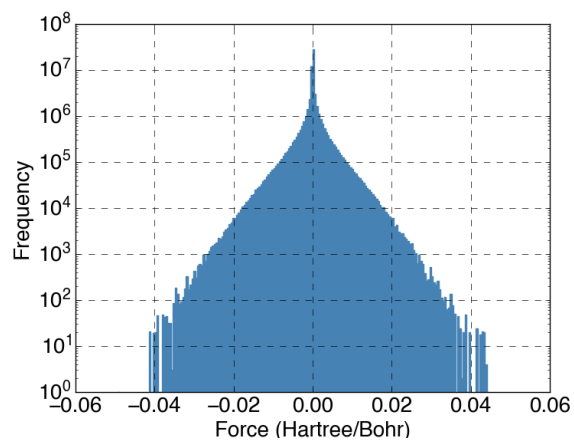


(c) Z-component

Figure 4: Distribution of the three components of the force after aligning vector connecting the origin and the nearest atom to the X-axis.

would enable computing the "loss function" from the predicted forces which are now consistent with the reference forces.

5. Now both $R^{s'}$ and $F_j^s$ are scaled using the reversible transformations:

$$R^{s''} = \frac{R^{s'} - \mu_{pos}}{\sigma_{pos}} \tag{1}$$

$$F_j^{s'} = \frac{F_j^s - \mu_{frc}}{\sigma_{frc}} \tag{2}$$

where $\mu_{pos}$ & $\mu_{frc}$ are the means of the coordinates and forces in the training data, respectively; $\sigma_{pos}$ & $\sigma_{frc}$ are their corresponding standard deviations.

The force used as the target vector is the difference between the force on the central atom when the forces are computed using a classical force field and a DFT force field, respectively ($F_{classical} - F_{DFT}$). This difference was used instead of training the network with just $F_{DFT}$ to prevent atom overlaps while running the simulation using forces from the neural network, since the training data would not include configurations involving atomic positions with non-physical contacts. This approach is philosophically similar to the $\Delta$-machine learning model proposed earlier[61] for predicting thermochemical properties. In this study, the forces from the network are used in the form of a corrective force on top of forces computed from the classical force field. This prevents atoms overlapping or moving into one another. This might happen when the neural network makes successive slightly inaccurate predictions, allowing two atoms to come too close. After that, since they are too close, and this configuration is now unknown to the network, they are not repelled apart as strongly as they should be, causing the problem to continue, and accumulate during the remaining length of the simulation. Hence using the classical force field ensures that the repulsive force is high enough when atoms come too close, while at the same time the corrective force from the network attempts to mimic the DFT force field. Additionally, the difference between

the forces lie in a smaller range than the forces themselves, which enables better capturing of the variations and ultimately, prediction by the network.

## Training the neural network

We used the Keras[62] neural networks API running on top of TensorFlow[63] to construct, train and validate the model. The network has an input layer, six densely connected hidden layers and an output layer. The neural network architecture is depicted in Figure 5. The input coordinates are processed to get a feature vector for each atom. The output forces are scaled back and rotated as described above before using them in the simulation.

The dataset has 637,500 frames, each consisting of 96 atoms, resulting in 61.2 million data points. The model was trained with a batch size of 2048 and an initial learning rate of $5 \times 10^{-4}$, which was reduced in case the learning stagnated over successive epochs. Mean squared error was used as the loss function along with the Adam[64] optimizer for updating the weights. All the hidden layers were activated using the ReLU[65] activation function. 70% of the entire dataset was used for training and the rest 30% was used for validation.

## Running simulations using forces from $\Delta$-NetFF

The OpenMM package[66,67] was used to run the simulations using the forces predicted by $\Delta$-NetFF . The forces obtained from $\Delta$-NetFF for each atom are added to the reference model forces (Lennard-Jones here) before the new positions and velocities are calculated by the integrator. The predicted forces added to the forces calculated using the Lennard Jones potential were fed through the custom force support in OpenMM. At each timestep, the positions of all the particles are used to construct a surrounding vector for each particle and all the vectors are submitted in one batch to the neural network. Output from $\Delta$-NetFF is rotated back into the Lab Reference frame before adding them to the Lennard-Jones forces. Several analyses such as radial distribution function, diffusion coefficient, densities at liquid-
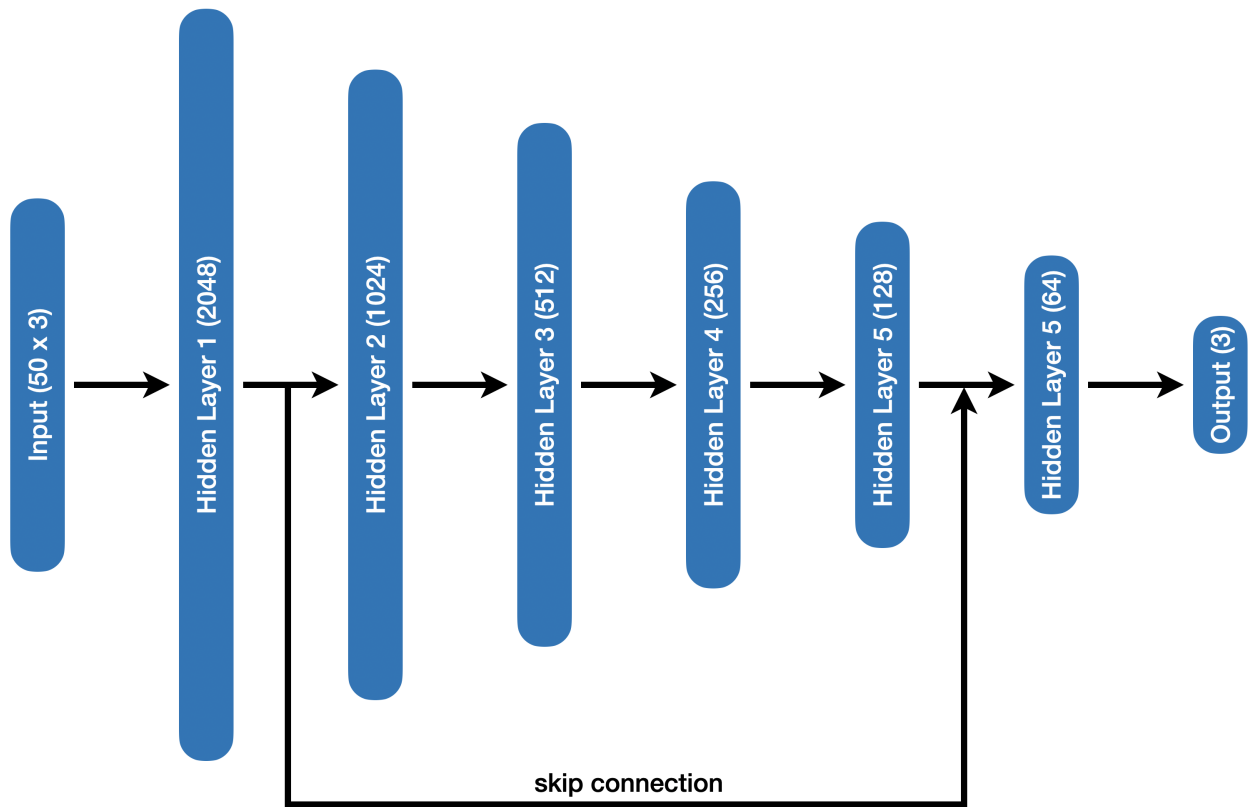
Figure 5: Architecture of the neural network used for the present Δ-NetFF model. The numbers of inputs, outputs and the number of nodes in each hidden layer are given in parenthesis.

vapor equilibrium and coefficient of viscosity have been performed to validate the approach, details of which are presented in the Results and Discussion section.

# Results and Discussion

The effectiveness of the trained model in real world scenarios can be better understood by calculating physical properties using trajectories generated by using the neural network in addition to the ability to predict the target force data. Here, we calculate the radial distribution function of liquid argon and compare it with the DFT data. To test the effectiveness of the trained model on a non-uniform particle distribution scenario, we perform liquid-vapor equilibrium simulations and compare the results with experimental data. The coefficient of diffusion and shear viscosity for argon are also calculated and compared with experimental data under various physical conditions to thoroughly validate the proposed method.

## Accuracy of the model

Figure 6 shows the coefficient of determination and the percentage of predictions within 25% and 50% tolerance of the labels for training and validation. For example, a 25% tolerance of the labels with $F_{classical} = 100 \ nN$ and $F_{DFT} = 120 \ nN$ would mean that $15 \ nN \leq F_{model} \leq 25 \ nN$ and $115 \ nN \leq F_{final}(= F_{classical} + F_{model}) \leq 125 \ nN$. It was observed that the learning stagnated at a coefficient of determination value of around 0.92 after 6 epochs, after which no significant improvements were observed. It was also observed that after 6 epochs, more than 85% of the predictions in the validation set were being predicted within a 50% error tolerance.
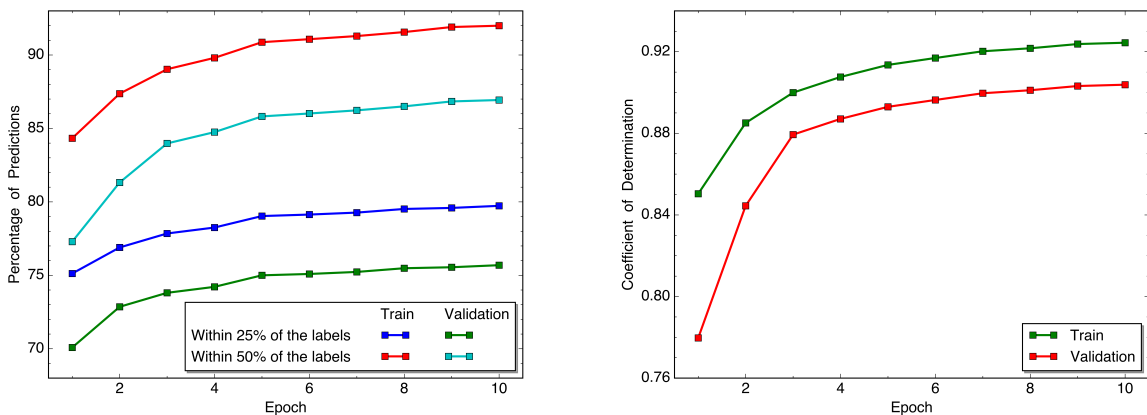
Figure 6: Coefficient of determination and percentage of predictions within 25% and 50% tolerance values.

## Liquid structure

The structure of liquid argon can be examined by using the radial distribution function (RDF). The RDF shows the location of the solvation shells and its deviation from the general shape can indicate the presence of unpredictable artifacts of a new force field. The 96 atom argon system at 85K from Maerzke et al.[60] was taken. A DFT trajectory using this system was generated and its RDF was plotted for reference. The experimental data was obtained from Yarnell et al.[68] The calculated RDF is shown in Figure 7.

The thermodynamic and energetic properties of liquid argon are sensitive to many body contributions but the structure of the liquid (and hence RDF) is not[69] - which explains the good agreement between experimental data and the distributions obtained by classical, DFT and the model. It can be observed in Figure 7 that the RDF generated by the $\Delta$-NetFF model seems to reproduce the DFT RDF satisfactorily, indicating that the simulation is using forces which are similar to what a DFT simulation would produce. Note that the objective of the model is to match DFT accuracy. Calculation of some properties presented below reveal better agreement with experimental data as compared to the classical force field.
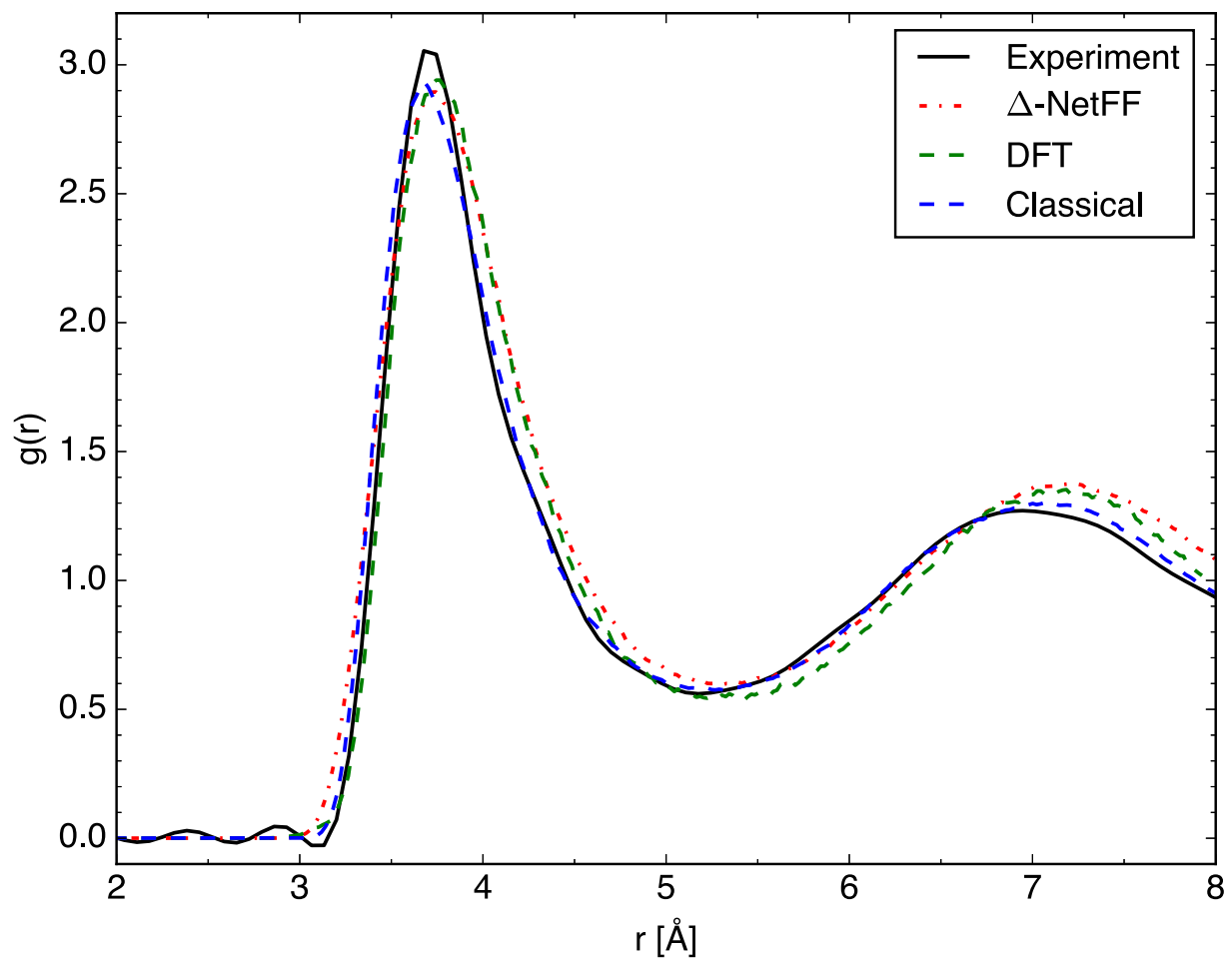
Figure 7: Radial distribution functions of liquid argon generated using trajectories obtained using the LJ potential, DFT and $\Delta$-NetFF along with that obtained using experiment.

## Diffusion Coefficient

The self-diffusion coefficient can be calculated from a molecular dynamics trajectory by using two different methods - the Green-Kubo method (based on the velocities),[70–73] and the Einstein method (based on the positions).[74] These two methods are different but equivalent. The self-diffusion coefficient from the Einstein method is obtained by relating the mean square displacement (MSD) of a certain particle over a certain time interval.[75] When the observation time tends to infinity, the self-diffusion coefficient is given by:

$$D = \lim_{t \to \infty} \frac{1}{6Nt} \langle \sum_{i=1}^{N} [r_i(t) - r_i(0)]^2 \rangle \tag{3}$$

where $t$ is the simulation time, $N$ is the total number of particles in the simulation and $r_i(t)$ is the displacement vector of the $i^{th}$ atom at time $t$, and the angular bracket denotes the ensemble average.

All the calculations were performed using a 500 atom argon system with a step size of 2 fs. A cubic cell was used with periodic boundary conditions in all directions. 50 ps of equilibrium was followed by a production run of 200 ps.

Figure 8 shows an example of log-log plot of MSD with respect to time along with the fitted line from which the diffusion coefficient is obtained. Refer Figures S1–S4 in the Supplementary Information for all the log-log plots. The initial part (short times) of the plot was not included in the calculation because the particles follow a ballistic trajectory, as indicated by the MSD being proportional to $t^2$ before going into the diffusive regime and being proportional to $t$. Long time-separations were also not included in the calculation because of the availability of very few points in the trajectory, and hence the statistics for these points are not optimal.

Table 1 shows the comparison between diffusion coefficients obtained from simulations using the classical force field, $\Delta$-NetFF model, and experimental values of Naghizadeh and
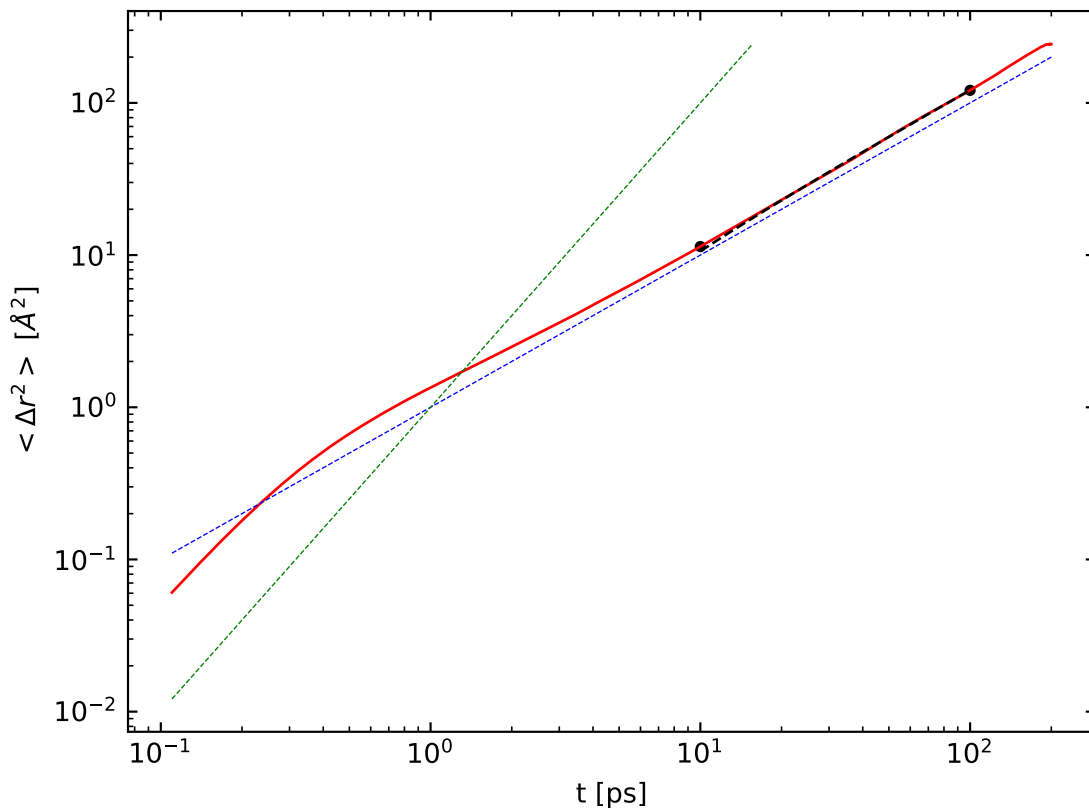
Figure 8: Log-log plot of MSD vs time for classical force field at 13.07 bar and 90 Kelvin. The blue line is the $t$ asymptote and the green line is the $t^2$ asymptote. The black dashed line represents the line whose slope is used to compute the diffusion coefficient.

Table 1: Comparison of self-diffusion coefficients calculated using various methods with the experimental values at different thermodynamic conditions.

| Pressure (bar) | Temperature (K) | Experiment[76] | Classical | $\Delta$-NetFF |
|---|---|---|---|---|
| 13.07 | 90 | 2.32 | 1.98 (-14.66) | **2.24 (-3.45)** |
| | 100 | 3.43 | 2.65 (-22.74) | **3.09 (-9.91)** |
| | 110 | 4.73 | 4.09 (-13.53) | **4.42 (-6.55)** |
| | 120 | 6.17 | 4.59 (-25.61) | **5.33 (-13.61)** |
| | 130 | 7.74 | 6.21 (-19.77) | **7.26 (-6.2)** |
| 58.6 | 94 | 2.38 | 1.95 (-18.07) | **2.465 (3.57)** |
| | 103 | 3.34 | 2.84 (-14.97) | **3.61 (8.08)** |
| | 108 | 4.16 | 3.11 (-25.24) | **3.82 (-8.17)** |

[a] Numbers in the parentheses are % error wrt the experiment value;
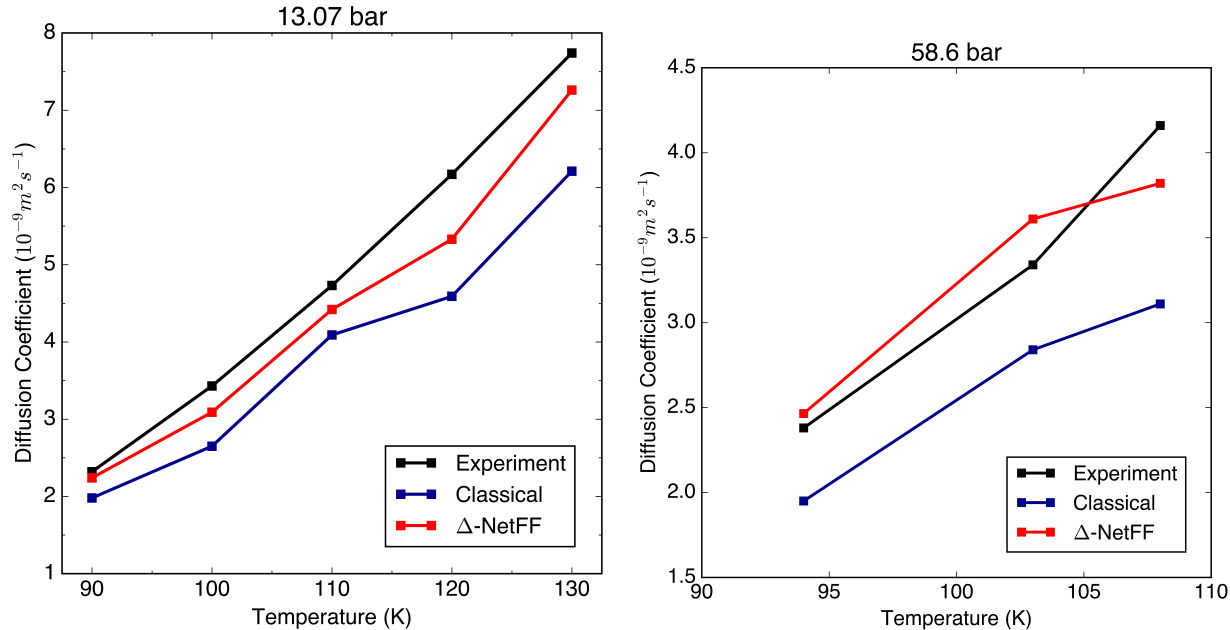[b] Diffusion coefficients units are in $10^{-9} m^2 s^{-1}$

19

Figure 9: Self-diffusion coefficients of argon obtained at different temperatures at 13.07 bar and 58.6 bar.

Rice (1962).[76] Since diffusion coefficient calculations require longer time scale trajectories (especially for gaseous states, owing to the high mean free path),[77] it is not feasible to calculate this property using a DFT trajectory.

The diffusion coefficient calculations indicate that the classical force fields lead to a lower diffusion coefficient than the experimental data. The coefficients obtained using forces from the $\Delta$-NetFF , as shown in Table 1, are closer to the experimental values, indicating that the DFT force field would give values which are closer to experiment.

At 13.07 bar, the values obtained from the classical force field are much lower than the experimental data, with an average error percentage of -19.26%. However, the values obtained from $\Delta$-NetFF have an average error percentage of -7.94% with the maximum error percentage -13.61% occurring at 120K.

At 58.6 bar, the data from the classical force field has an average error percentage of -19.43% while $\Delta$-NetFF produces 6.6% average error, with the maximum deviation of -8.17

K occurring at 108 K. The general shape of the plots at both pressures indicate that the model generated trajectories follow the trend of the classical trajectories, due to the $F_{classical}$ contribution in the force model. However the corrective terms learned by the model, shift the calculated values closer to the experimental data.

## Liquid-vapor density profile

To evaluate the performance of the model on a system with a changing density profile, we simulated a liquid-vapor interface system and compared the density profile with that of classical force fields and experimental data. For simulating the liquid-vapor interface, an equilibrated system (cubic) of liquid argon is placed between two empty regions. The initial configuration is a cubic liquid slab in the middle, and along the z-direction there are two regions of vacuum (see Figure 10).
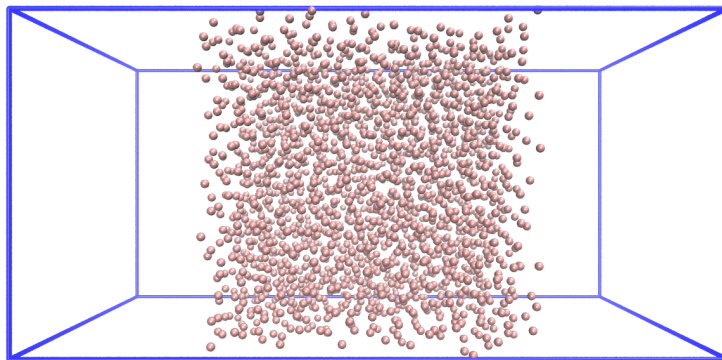


Figure 10: Initial snapshot of the liquid-vapor system.

Molecular dynamics simulations were performed with 2400 argon atoms at constant temperature and volume, and periodic boundary conditions were applied in x, y and z-directions. The step size was taken as 2 fs. The two systems taken in accordance with the density of liquid argon at the specified temperature are shown in Table 2.

Table 2: The two systems chosen for the liquid-vapor interface simulation. $L$ is the length of the side of the simulation box.

| T(K) | $\rho(g/cc)$ | $L_x = L_y$ (Å) $^a$ | $L_z$ (Å) |
|------|--------------|----------------------|-----------|
| 94.4 | 1.374 | 48.753 | 100 |
| 119.8 | 1.176 | 51.348 | 100 |

$^a$ Initial equilibration of the slab was done in a cubic box with dimensions $L_z = L_x$.

Figure 11 shows the density profiles, $\rho(z)$ in $g/cc$ along the z-direction which is the direction normal to the slab surface used in the MD simulation. The density profile being symmetric about z=0, i.e. the midpoint, supports the case of the system being properly equilibrated. Density values obtained in the liquid and vapor regions are compared with experimental data (see Table 3). Indeed the $\Delta$-NetFF model is able to generate different density values for different regions in the system. The model generated values are much closer to the experimental data as compared to the classical force field. It is also evident that the thickness of the liquid slab is approximately the same for both the force fields.
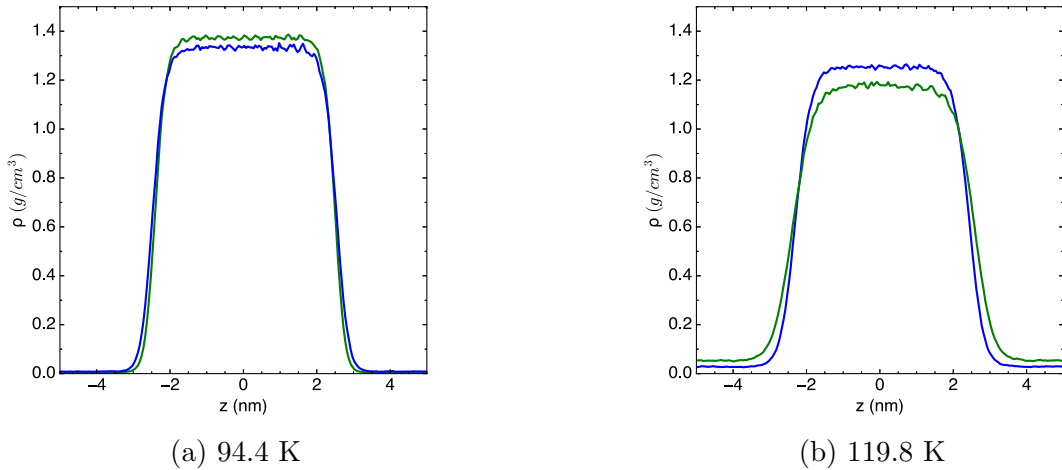


(a) 94.4 K                                   (b) 119.8 K

Figure 11: Vapor-liquid density profile along the z-direction which is normal to the interface in the simulation. Blue line represents the classical force field while the green line represents $\Delta$-NetFF .

Table 3: Liquid-vapor Interface: $\rho_v$ and $\rho_l$ are the vapor and liquid densities at equilibrium.

| Temperature (K) | $\rho_v$(expt) | $\rho_v$ | | $\rho_l$(expt) | $\rho_l$ | |
|---|---|---|---|---|---|---|
| | | Classical | $\Delta$-NetFF | | Classical | $\Delta$-NetFF |
| 94.4 | 0.01089 | 0.00627 | 0.01208 | 1.3507 | 1.326 | 1.378 |
| 119.8 | 0.05947 | 0.02877 | 0.05352 | 1.1645 | 1.251 | 1.171 |

[a] All density values are in $g/cc$

## Shear viscosity

The Green-Kubo approach (Equation 4) uses the running integral over time of the pressure tensor autocorrelation function to calculate shear viscosity.[78]

$$\eta = \frac{V}{k_B T} \int_0^\infty \langle P_{\alpha\beta}(t) \cdot P_{\alpha\beta}(0) \rangle dt \tag{4}$$

where, $V$ is the system volume, $k_B$ is the Boltzmann constant, $T$ is temperature, $P_{\alpha\beta}$ is the $\alpha\beta$ element of the pressure tensor, $\alpha, \beta \in [x, y, z]$, and the angular bracket denotes the ensemble average.

Multiple stress tensor terms can be used to improve averaging statistics across the ensemble.[79–81] For our calculations, we used all six independent components of the stress tensor. The integral is now given by,[81,82]

$$\eta = \frac{V}{10 k_B T} \sum_i \sum_j \int_0^\infty \langle P_{ij}^{OS}(t) \cdot P_{ij}^{OS}(0) \rangle dt \tag{5}$$

where $P_{ij}^{OS}$ is the symmetrized and traceless stress sensor expressed as,

$$P_{ij}^{OS} = \omega_{ij} \left( \frac{P_{ij} + P_{ji}}{2} - \frac{\delta_{ij}}{3} \sum_k P_{kk} \right) \tag{6}$$

Here, $\delta_{ij}$ is the Kronecker delta, $\omega_{ij} = 1$ for $i \neq j$ and $\omega_{ij} = 4/3$ for $i = j$.[83] Due to the weight factors $\omega_{ij}$, the factor of 10 appears in Equation 5.

All the calculations were done using a 500 argon atom system. A cubic cell was used with periodic boundary conditions in all directions. Because the stress autocorrelation function

is a property of the system as a whole, it has a non decaying long time tail. Hence, to improve statistical accuracy, the final viscosity (Figure 12) reported was averaged over 15 individual trajectories (Figures S5–S6 in the Supplementary Information). Due to the well-known,[84] large oscillations at very short times and noise at very long times, these regions are excluded from the fitting procedure. Table 4 illustrates the calculated viscosity at 3 different temperature and pressure combinations. At 94.4 K, the classical force field has a 31% deviation from the experimental value, which goes above 50% at 135 K. The calculations done using the forces from the $\Delta$-NetFF , however, are much closer to the experimental values, evident by the 15% deviation at 94.4 K and going upto 54% at 135 K.

Table 4: Comparison of viscosity coefficients calculated using various methods with the experimental values at different thermodynamic conditions.

| Temperature and Pressure | Experiment | Classical | $\Delta$-NetFF |
|---|---|---|---|
| 94.4 K and 1 atm | 0.197 | 0.258400(31.16) | 0.227747(15.6) |
| 110 K and 4.93 atm | 0.0092 | 0.01428(55.21) | 0.012568(36.6) |
| 135 K and 4.93 atm | 0.01116 | 0.02133(91.12) | 0.01729(54.9) |

[a] Numbers in the parentheses are % error wrt experiment value;
[b] Coefficient of viscosity are in centipoise

**Efficiency**

Comparison between times taken by different simulation-setups are shown in Table 5. All systems were run on a system with a 20 core (40 threads) CPU and 1 Nvidia GeForce GTX 1080 Ti GPU. All trajectories were written onto a solid state drive (SSD) to minimize the time spent on saving the data to persistent storage. It is important to note that when the neural network pipeline was used to run any simulation, optimizations such as cell lists[78] and Verlet lists[85] were not used while constructing atom environments at each timestep. They can greatly improve the speed of the atom environment construction step. It should also be
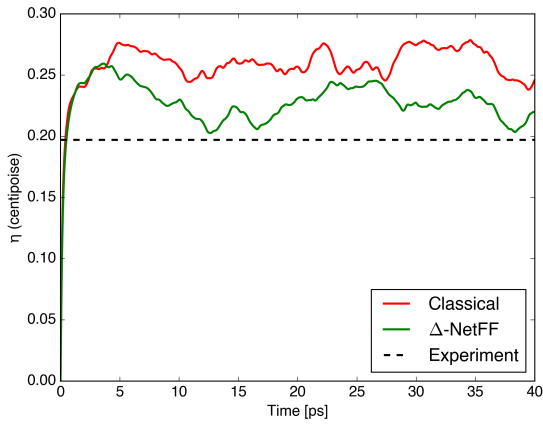
(a) 94.4 K and 1 atm
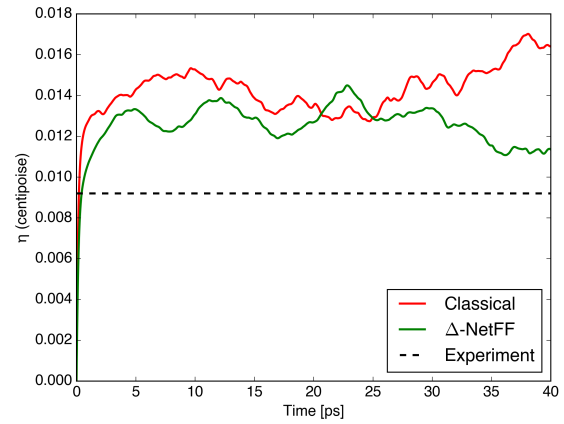
(b) 110 Kelvin and 4.93 atm
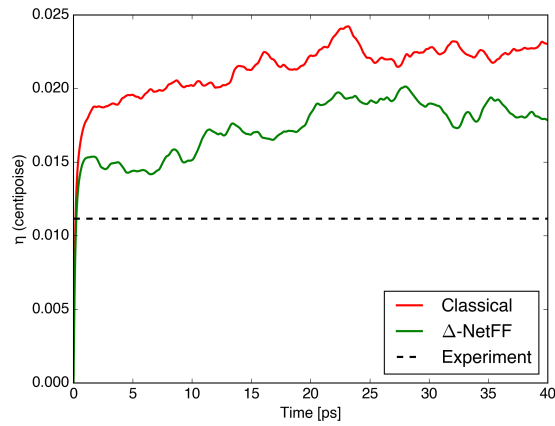
(c) 135 Kelvin and 4.93 atm

Figure 12: Time evolution of viscosity coefficients calculated using various methods at different thermodynamic conditions.

noted that the speed of the 96 atom system does not scale well with respect to the number of CPU cores because of the bottleneck of sending and receiving data between processes.

It can be observed that, as expected, the classical setup is clearly the fastest irrespective of the size of the system. The DFT calculations become computationally expensive for real-time applications for systems with more than a few hundreds atoms. Even if a system can fit within its computational requirements, the time taken to calculate forces for each timestep would be so high that any reasonably long simulation would take a very long time to finish. This is evident by the fact that the 96 atom DFT simulation proceeds at a rate of only 348 timesteps per hour. In contrast, simulations run by the model, along with being significantly faster than DFT, do not have an exponentially increasing demand for computational resources with respect to increasing system sizes. This also allows for the prospect of running multiple independent simulations of the same system to improve statistical accuracy.

Table 5: Simulation time comparison of different simulation-setups.

| System | | Setup | | |
|---|---|---|---|---|
| Atoms | Timesteps | Classical | DFT | $\Delta$-**NetFF** |
| 96 | | 20 minutes | 12 days | **3 hours** |
| 2400 | 100,000 | 70 minutes | infeasible | **14 hours** |
| 10,000 | | 3 hours | infeasible | **44 hours** |

# Conclusion

In this work, we explore a deep learning based method of imitating DFT derived forces to simulate systems with larger sizes and longer time scales. This method involves (a) constructing random configurations of atoms and obtaining forces for each configuration to be used as a dataset; (b) choosing a suitable atomic environment vector representation of each atom; (c) training the neural network; and (d) running simulations using predicted forces

and calculating desired properties by taking suitable system sizes and time scales. The shape of the RDF being similar to that of a DFT trajectory indicates that the forces coming from the neural network are biased in the direction of the resultant DFT forces. Density values of liquid and vapor regions obtained from the liquid slab simulation validate the model's robustness to perform on a changing density profile. Diffusion coefficient and viscosity calculations indicate that the new forces bias the simulation closer towards experimental data, indicating that these properties calculated from a long DFT simulation would most likely be closer to experimental values as compared to a purely classical force field. Finally, the time comparison data further emphasizes the efficiency of the model to run long and multiple replicate simulations which are vital in the calculation of thermodynamic and kinetic properties. Future work to modify the feature vector to extend this method to complex multicomponent systems is in progress.

## Supporting Information

The Supporting Information is available free of charge at xxxxxxx. Plots of mean squared displacements used for calculating diffusion constants and plots of stress tensor autocorrelation functions.

## Acknowledgements

## Acknowledgements

# References

(1) Stanton, J. F. Why CCSD (T) works: a different perspective. *Chem. Phys. Lett.* **1997**, *281*, 130–134.

(2) Møller, C.; Plesset, M. S. Note on an Approximation Treatment for Many-Electron Systems. *Phys. Rev.* **1934**, *46*, 618–622.

(3) Brooks, B. R. et al. CHARMM: The biomolecular simulation program. *J. Comput. Chem.* **2009**, *30*, 1545–1614.

(4) Scott, W. R. P.; Hünenberger, P. H.; Tironi, I. G.; Mark, A. E.; Billeter, S. R.; Fennen, J.; Torda, A. E.; Huber, T.; Krüger, P.; van Gunsteren, W. F. The GROMOS Biomolecular Simulation Program Package. *J. Phys. Chem. A* **1999**, *103*, 3596–3607.

(5) Morrone, J. A.; Car, R. Nuclear quantum effects in water. *Phys. Rev. Lett.* **2008**, *101*, 017801.

(6) Ceriotti, M.; Cuny, J.; Parrinello, M.; Manolopoulos, D. E. Nuclear quantum effects and hydrogen bond fluctuations in water. *Proc. Natl. Acad. Sci. USA* **2013**, *110*, 15591–15596.

(7) Car, R.; Parrinello, M. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* **1985**, *55*, 2471.

(8) MacKerell Jr, A. D. Empirical force fields for biological macromolecules: overview and issues. *J. Comput. Chem.* **2004**, *25*, 1584–1604.

(9) Massobrio, C.; Du, J.; Bernasconi, M.; Salmon, P. S. *Molecular dynamics simulations of disordered materials*; Springer, 2015; Vol. 215.

(10) Sen, F.; Kinaci, A.; Narayanan, B.; Gray, S.; Davis, M.; Sankaranarayanan, S.; Chan, M. Towards accurate prediction of catalytic activity in IrO 2 nanoclusters via first principles-based variable charge force field. *J. Mater. Chem. A* **2015**, *3*, 18970–18982.

(11) Behler, J. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* **2011**, *134*, 074106.

(12) Rupp, M. Machine learning for quantum mechanics in a nutshell. *Int. J. Quantum Chem.* **2015**, *115*, 1058–1073.

(13) Smith, J. S.; Isayev, O.; Roitberg, A. E. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, *8*, 3192–3203.

(14) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **2013**, *15*, 095003.

(15) Bartók, A. P.; Kondor, R.; Csányi, G. On representing chemical environments. *Phys. Rev. B* **2013**, *87*, 184115.

(16) Yao, K.; Herr, J. E.; Brown, S. N.; Parkhill, J. Intrinsic Bond Energies from a Bonds-in-Molecules Neural Network. *J. Phys. Chem. Lett.* **2017**, *8*, 2689–2694, PMID: 28573865.

(17) Schütt, K.; Kindermans, P.-J.; Sauceda Felix, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc., 2017; pp 991–1001.

(18) Himmetoglu, B. Tree based machine learning framework for predicting ground state energies of molecules. *J. Chem. Phys.* **2016**, *145*, 134101.

(19) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **2012**, *108*, 058301.

(20) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754–13769.

(21) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *J. Phys. Chem. Lett.* **2015**, *6*, 2326–2331, PMID: 26113956.

(22) Ramakrishnan, R.; von Lilienfeld, O. A. Machine learning, quantum chemistry, and chemical space. *Rev. Comput. Chem.* **2017**, *30*, 225–256.

(23) Janet, J. P.; Chan, L.; Kulik, H. J. Accelerating Chemical Discovery with Machine Learning: Simulated Evolution of Spin Crossover Complexes with an Artificial Neural Network. *J. Phys. Chem. Lett.* **2018**, *9*, 1064–1071, PMID: 29425453.

(24) Laghuvarapu, S.; Pathak, Y.; Priyakumar, U. D. Band nn: A deep learning framework for energy prediction and geometry optimization of organic small molecules. *J. Comput. Chem.* **2019**,

(25) Pathak, Y.; Laghuvarapu, S.; Mehta, S.; Priyakumar, U. *Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules*; ChemRxiv, 2019.

(26) Dral, P. O.; von Lilienfeld, O. A.; Thiel, W. Machine learning of parameters for accurate semiempirical quantum chemical calculations. *J. Chem. Theory Comput.* **2015**, *11*, 2120–2125.

(27) Botu, V.; Ramprasad, R. Learning scheme to predict atomic forces and accelerate materials simulations. *Phys. Rev. B* **2015**, *92*, 094306.

(28) Huan, T. D.; Batra, R.; Chapman, J.; Krishnan, S.; Chen, L.; Ramprasad, R. A universal strategy for the creation of machine learning-based atomistic force fields. *Npj Comput. Mater.* **2017**, *3*, 1–8.

(29) Ramprasad, R.; Batra, R.; Pilania, G.; Mannodi-Kanakkithodi, A.; Kim, C. Machine learning in materials informatics: recent applications and prospects. *npj Comput. Mater.* **2017**, *3*, 1–13.

(30) Li, Z.; Kermode, J. R.; De Vita, A. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.* **2015**, *114*, 096405.

(31) Bartók, A. P.; Payne, M. C.; Kondor, R.; Csányi, G. Gaussian Approximation Potentials: The Accuracy of Quantum Mechanics, without the Electrons. *Phys. Rev. Lett.* **2010**, *104*, 136403.

(32) Zeni, C.; Rossi, K.; Glielmo, A.; Fekete, Á.; Gaston, N.; Baletto, F.; De Vita, A. Building machine learning force fields for nanoclusters. *J. Chem. Phys.* **2018**, *148*, 241739.

(33) Ulissi, Z. W.; Tang, M. T.; Xiao, J.; Liu, X.; Torelli, D. A.; Karamad, M.; Cummins, K.; Hahn, C.; Lewis, N. S.; Jaramillo, T. F., et al. Machine-learning methods enable exhaustive searches for active bimetallic facets and reveal active site motifs for CO2 reduction. *ACS Catalysis* **2017**, *7*, 6600–6608.

(34) Jinnouchi, R.; Asahi, R. Predicting catalytic activity of nanoparticles by a DFT-aided machine-learning algorithm. *J. Phys. Chem. Lett.* **2017**, *8*, 4279–4283.

(35) Dolgirev, P. E.; Kruglov, I. A.; Oganov, A. R. Machine learning scheme for fast extraction of chemically interpretable interatomic potentials. *AIP Advances* **2016**, *6*, 085318.

(36) Wang, Y.; Ribeiro, J. M. L.; Tiwary, P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Current Opinion in Structural Biology* **2020**, *61*, 139–145.

(37) Chattopadhyay, A.; Zheng, M.; Waller, M. P.; Priyakumar, U. D. A probabilistic frame-

work for constructing temporal relations in replica exchange molecular trajectories. *Journal of chemical theory and computation* **2018**, *14*, 3365–3380.

(38) Stanton, R. Hellmann-Feynman Theorem and Correlation Energies. *J. Chem. Phys.* **1962**, *36*, 1298–1300.

(39) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; Von Lilienfeld, O. A.; Tkatchenko, A.; Müller, K.-R. Assessment and validation of machine learning methods for predicting molecular atomization energies. *J. Chem. Theory Comput.* **2013**, *9*, 3404–3419.

(40) Rupp, M.; Ramakrishnan, R.; Von Lilienfeld, O. A. Machine learning for quantum mechanical properties of atoms in molecules. *J. Phys. Chem. Lett.* **2015**, *6*, 3309–3313.

(41) Hirn, M.; Poilvert, N.; Mallat, S. Quantum energy regression using scattering transforms. *arXiv preprint arXiv:1502.02077* **2015**,

(42) Chmiela, S.; Tkatchenko, A.; Sauceda, H. E.; Poltavsky, I.; Schütt, K. T.; Müller, K.-R. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **2017**, *3*, e1603015.

(43) Koziol, L.; Fried, L. E.; Goldman, N. Using force matching to determine reactive force fields for water under extreme thermodynamic conditions. *J. Chem. Theory Comput.* **2017**, *13*, 135–146.

(44) Behler, J.; Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* **2007**, *98*, 146401.

(45) Behler, J.; Lorenz, S.; Reuter, K. Representing molecule-surface interactions with symmetry-adapted neural networks. *J. Chem. Phys.* **2007**, *127*, 07B603.

(46) Jose, K. J.; Artrith, N.; Behler, J. Construction of high-dimensional neural network potentials using environment-dependent atom pairs. *J. Chem. Phys.* **2012**, *136*, 194111.

(47) Bartók, A. P.; Csányi, G. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.* **2015**, *115*, 1051–1057.

(48) Chen, C.; Deng, Z.; Tran, R.; Tang, H.; Chu, I.-H.; Ong, S. P. Accurate force field for molybdenum by machine learning large materials data. *Phys. Rev. Mater.* **2017**, *1*, 043603.

(49) Lide, D. R. *CRC handbook of chemistry and physics*; CRC press, 2004; Vol. 85.

(50) Bridson, R. Fast Poisson Disk Sampling in Arbitrary Dimensions. ACM SIGGRAPH 2007 Sketches. New York, NY, USA, 2007.

(51) Hutter, J.; Iannuzzi, M.; Schiffmann, F.; VandeVondele, J. cp2k: atomistic simulations of condensed matter systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 15–25.

(52) VandeVondele, J.; Krack, M.; Mohamed, F.; Parrinello, M.; Chassaing, T.; Hutter, J. Quickstep: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach. *Comput. Phys. Commun.* **2005**, *167*, 103 – 128.

(53) Becke, A. D. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A* **1988**, *38*, 3098–3100.

(54) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **1988**, *37*, 785–789.

(55) Goedecker, S.; Teter, M.; Hutter, J. Separable dual-space Gaussian pseudopotentials. *Phys. Rev. B* **1996**, *54*, 1703–1710.

(56) Hartwigsen, C.; Goedecker, S.; Hutter, J. Relativistic separable dual-space Gaussian pseudopotentials from H to Rn. *Phys. Rev. B* **1998**, *58*, 3641–3662.

(57) VandeVondele, J.; Hutter, J. An efficient orbital transformation method for electronic structure calculations. *J. Chem. Phys.* **2003**, *118*, 4365–4369.

(58) Pulay, P. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters* **1980**, *73*, 393 – 398.

(59) Pulay, P. Improved SCF convergence acceleration. *J. Comput. Chem.* **1982**, *3*, 556–560.

(60) Maerzke, K. A.; Murdachaew, G.; Mundy, C. J.; Schenter, G. K.; Siepmann, J. I. Self-Consistent Polarization Density Functional Theory: Application to Argon. *J. Phys. Chem. A* **2009**, *113*, 2075–2085, PMID: 19260723.

(61) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big data meets quantum chemistry approximations: The $\Delta$-machine learning approach. *J. Chem. Theory Comput.* **2015**, *11*, 2087–2096.

(62) Chollet, F., et al. Keras. https://keras.io, 2015.

(63) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; https://www.tensorflow.org/, Software available from tensorflow.org.

(64) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(65) Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on International Conference on Machine Learning. USA, 2010; pp 807–814.

(66) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Comput. Biol.* **2017**, *13*, 1–17.

(67) Eastman, P.; Pande, V. S. Efficient nonbonded interactions for molecular dynamics on a graphics processing unit. *J. Comput. Chem.* **2010**, *31*, 1268–1272.

(68) Yarnell, J.; Katz, M.; Wenzel, R. G.; Koenig, S. Structure factor and radial distribution function for liquid argon at 85 K. *Phys. Rev. A* **1973**, *7*, 2130.

(69) Bukowski, R.; Szalewicz, K. Complete ab initio three-body nonadditive potential in Monte Carlo simulations of vapor–liquid equilibria and pure phases of argon. *J. Chem. Phys.* **2001**, *114*, 9518–9531.

(70) Green, M. S. Brownian Motion in a Gas of Noninteracting Molecules. *J. Chem. Phys.* **1951**, *19*, 1036–1046.

(71) Green, M. S. Comment on a Paper of Mori on Time-Correlation Expressions for Transport Properties. *Phys. Rev.* **1960**, *119*, 829–830.

(72) Kubo, R. Statistical-Mechanical Theory of Irreversible Processes. I. General Theory and Simple Applications to Magnetic and Conduction Problems. *Journal of the Physical Society of Japan* **1957**, *12*, 570–586.

(73) Mori, H. Statistical-Mechanical Theory of Transport in Fluids. *Phys. Rev.* **1958**, *112*, 1829–1842.

(74) Einstein, A. *Investigations on the Theory of the Brownian Movement*; Courier Corporation, 1956.

(75) Mark, P.; Nilsson, L. Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K. *J. Phys. Chem. A* **2001**, *105*, 9954–9960.

(76) Naghizadeh, J.; Rice, S. A. Kinetic Theory of Dense Fluids. X. Measurement and Interpretation of Self-Diffusion in Liquid Ar, Kr, Xe, and CH4. *J. Chem. Phys.* **1962**, *36*, 2710–2720.

(77) Helfand, E. Transport Coefficients from Dissipation in a Canonical Ensemble. *Phys. Rev.* **1960**, *119*, 1–9.

(78) Allen, M. P.; Tildesley, D. J. *Computer simulation of liquids*; Oxford university press, 2017.

(79) Hess, B. Determining the shear viscosity of model liquids from molecular dynamics simulations. *J. Chem. Phys.* **2002**, *116*, 209–217.

(80) Liu, H.; Maginn, E.; Visser, A. E.; Bridges, N. J.; Fox, E. B. Thermal and Transport Properties of Six Ionic Liquids: An Experimental and Molecular Dynamics Study. *Industrial & Engineering Chemistry Research* **2012**, *51*, 7242–7254.

(81) Daivis, P. J.; Evans, D. J. Comparison of constant pressure and constant volume nonequilibrium simulations of sheared model decane. *J. Chem. Phys.* **1994**, *100*, 541–547.

(82) Maginn, E. J.; Messerly, R. A.; Carlson, D. J.; Roe, D. R.; Elliott, J. R. Best Practices for Computing Transport Properties 1. Self-Diffusivity and Viscosity from Equilibrium Molecular Dynamics [Article v1.0]. *Living Journal of Computational Molecular Science* **2018**, *1*, 6324–.

(83) Borodin, O.; Smith, G. D.; Kim, H. Viscosity of a room temperature ionic liquid: predictions from nonequilibrium and equilibrium molecular dynamics simulations. *J. Phys. Chem. B* **2009**, *113*, 4771—4774.

(84) Zhang, Y.; Otani, A.; Maginn, E. J. Reliable Viscosity Calculation from Equilibrium Molecular Dynamics Simulations: A Time Decomposition Method. *J. Chem. Theory Comput.* **2015**, *11*, 3537–3546, PMID: 26574439.

(85) Verlet, L. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.* **1967**, *159*, 98–103.

**Table of Contents Graphic**



$$a_i(t) = \frac{F_i(r_i(\Delta t))}{m_i}$$

$$r_i(t + \Delta t) = 2r_i(t) - r_i(t - \Delta t) + a_i(t)\Delta t^2 + O(\Delta t^4)$$