© 2021 Alireza Moradzadeh

MACHINE-LEARNING OF LIQUIDS: A MULTI-SCALE FRAMEWORK FOR LIQUID PHYSICS AND COARSE-GRAINING USING DEEP LEARNING

BY

ALIREZA MORADZADEH

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2021

Urbana, Illinois

Doctoral Committee:

Professor Narayana R. Aluru, Chair
Associate Professor Elif Ertekin
Associate Professor Charles Sing
Associate Professor Kyle C. Smith

Abstract

Liquids are a condensed and non-crystallized phase of matter and their properties are typically governed by many-body effects. The physics and properties of liquids are a mixture of gas and solid phases, but quite different from any of them. The application of theoretical models developed for solid and gas phases to liquids is difficult and development of theoretical models for liquid is hindered by many-body effects. Computer simulation and experimental methods have played a crucial role in understanding of liquids. In fact, most of our knowledge about liquids are obtained using computer simulation, especially molecular dynamics (MD) and Monte Carlo simulations. MD simulations are successful in finding thermodynamic, structural, dynamical, and transport properties of liquids. However, MD simulation requires accurate potential parameters to model physical phenomena. It is noteworthy to state that with known potential parameters structural properties not only describe local arrangement of atoms, but they are enough to calculate various thermodynamic properties such as pressure and isothermal compressibility. The more intriguing part about liquids is the relationship between structural properties and potential parameters. This question can be seen as the inverse problem of liquid-state theory or as a coarse-graining problem, where the objective is to parameterize a force field to reproduce a reference structure. In this study, a deep neural network is used for atom-agnostic parametrization of the Lennard-Jones potential at different thermodynamic states. The DNN demonstrates good performance for two cases – parameterization of LJ particles and development of single-bead CG LJ potentials for simple multi-atom molecules through transfer learning obtained from LJ particles. The transferability and generalizability of the method are investigated by computing the total variation in the radial distribution function and Kullback-Leibler divergence for the coarse-grained model development. Our results indicate that deep learning can compute the solution to the inverse-problem of liquid-state theory (DeepILST) under the assumption of a predetermined pair potential in a coarse-grained model.

In a follow-up study, statistical and deep learning-based methods are employed to obtain insights into the quasi-universal properties of simple liquids. In the first part, a statistical model is employed to provide a probabilistic explanation for the similarity in the structure of simple liquids interacting with different pair potential forms, collectively known as simple liquids. The methodology works by sampling the radial distribution function and the number of interacting particles within the cut-off distance and it produces the probability density function of the net force. We show that matching the probability distribution of the net force can be a direct route to parameterize simple liquids pair potentials with a similar structure, as the net force is the main component of the Newtonian equations of motion. The statistical model is assessed and validated against various cases. The physics and quasi-universality of simple liquids are also studied through deep learning by finding structurally-equivalent Lennard-Jones liquids with similar reduced RDFs, *i.e.*, isomorphs. Structurally-equivalent Lennard-Jones liquids identify systems with constant order parameters in the space of non-dimensional temperature and density of Lennard-Jones liquids consistent with the approximate theoretical solution derived in the current study and other theoretical models. Considering various investigations performed in this study, we show the successful employment of statistical and deep learning approaches and coarse-graining methods in the physics of simple liquids.

As shown in above examples, MD simulation is a popular and strong computational tool to compute microscopic and macroscopic properties of liquids. However, to determine properties of atomic systems to a good level of accuracy with minimal noise or fluctuation, MD simulations are performed over a long time ranging from a few nanoseconds to several tens to hundreds of nanoseconds depending on the system and the properties of interest. There have been attempts to go around this issue with enhanced sampling and theoretical models. In this study, by considering simple liquids, we explore the feasibility of significantly reducing the MD simulation time to compute various properties of monoatomic systems such as the structure, pressure, and isothermal compressibility. A deep denoising autoencoder network is trained to obtain structural and thermodynamic properties of Lennard-Jones liquids at various thermodynamic states using a single snapshot RDF as input. The algorithm is successful not only in

predicting the RDF of a Lennard-Jones pair potential, but also it is generalizable to other simple liquid pair potentials such as exponential, Yukawa, and inverse-power-law potentials. In terms of computational efficiency, the number of snapshots required from MD simulation to obtain the accuracy of DAE predicted RDF is at least hundred snapshots, making the network highly efficient. The pressure and isothermal compressibility from DAE based RDFs are also comparable with those obtained from longtime MD simulation.

To expand our frameworks for more complex liquids, we investigate development of coarse-grained (CG) models of water, which is far more complex than simple liquids. In this study, we train a neural network-based force field with two- and three-body interactions, which makes the developed force field interpretable. Within our framework, the requirement for accurate forces and energies is eliminated by using the local search algorithm instead of backpropagation. We successfully develop coarse-grained models of both classical and *ab initio* water models. We also investigate the dependency of the coarse-grained force field of water on the number of expansions, which shows that the double-well interaction, known as a signature of water-like behavior among spherically symmetric pairwise interactions, vanishes with the inclusion of three-body interactions. We also notice that the two-body interaction fails to reproduce the angular distribution of water, especially over a short range. Based on our findings, we conclude that water-like behavior is better captured using the three-body interaction, which is consistent with the directional dependency of interactions in water.

Finally, we employ graph neural network in the phase identification of water. Due to the high dimensionality and uninterpretable nature of atomistic simulation data, researchers have developed a wide variety of order parameters to reduce dimensionality and connect data with the phase and structural properties. Motivated by the importance of water in various areas, water is studied through various order parameters such as bond-order parameter (BOP), tetrahedral order parameter, and local-structure index. Even though these order parameters are widely adapted in various studies ranging from ice nucleation, phase discrimination/identification, free energy calculation, and as collective variables of enhanced

sampling simulation, however, they are far from perfect. In several cases, it requires lots of domain expertise and effort to combine multiple order parameters to reach conclusive findings. Our phase classification works by collecting all the pairwise distances of high dimensional data within a cut-off distance, followed by feeding these data into an edge-conditioned convolutional graph neural network. The high accuracy and no need for pre-calculation of other order parameters makes our methods more rigorous and generalizable for more complex cases such as confined water.

To my family and friends

Acknowledgements

I would like to wholeheartedly thank my adviser, Professor Narayana R. Aluru, for his tremendous supervision and guidance. Nothing could have been achieved without his critical and scientific assessment of my work. He has dedicated lots of time and emotional support through my graduate years. My special thanks are extended to Professor Elif Ertekin, Professor Charles Sing, Professor Kyle C. Smith, and Professor Diwakar Shukla for their time and thoughtful feedback.

I would also like to thank my groupmates in the Computational Multiscale Nanosystems group during my Ph.D., as well as my collaborators, Prof. Salehi and Dr. Anh Pham for their wonderful guidance and collaborations.

I would like to acknowledge the financial support by NSF, DOE, NanoMFG, and the Beckman Institute for Advanced Science and Technology. I also need to acknowledge the use of the parallel computing resource Blue Waters provided by the University of Illinois and National Center for Supercomputing Applications (NCSA).

I am deeply thankful to my parents Nader and Batool, and my siblings Zahra and Amirreza. Their unconditional love and support played an undeniable role in keeping me on track towards my goals.

TABLE OF CONTENTS

CHAPTER 1: Introduction

1.1 Liquid Physics

Liquids are a condensed and non-crystallized phase of matter, and their properties are typically governed by many-body effects. The physics and properties of liquids is a complex mixture of gas and solid phases, but quite different from any of them. For example, liquids particles can move around like gas particles and unlike solid particles, but their motion is correlated like solid particles and unlike gas particles.[1,2] The presence of both gas- and solid-like behaviors limits accuracy of theoretical models developed for the gas and solid to liquid. On the other hand, most of the theoretical and analytical methods developed for liquids suffer from low accuracy due to complex interplay between entropic and energetic forces. Therefore, understanding physics of liquids phase of matter is hindered using theoretical framework. Considering that liquids play a crucial role in many physiochemical and biological phenomena such as super-capacitor[3,4], gas capture,[5] lubrication,[6,7] protein-folding,[8,9] it is of outmost important to fundamentally understand the physics and properties of liquids.

Lacking theoretical and analytical methods to understand liquids, computer simulation and experimental methods have played crucial role in understanding of liquids. In fact, most of our knowledge about liquids are obtained using computer simulation, especially molecular dynamics (MD) and Monte Carlo simulations.[10] MD simulations are successful in finding thermodynamic, structural, dynamical, and transport properties of liquids.[11] MD results also provide initial finding for quasi-universal properties among simple liquids.[10] However, MD simulation requires accurate potential parameters to model physical phenomena. It is noteworthy to state that with known potential parameters structural properties not only describe local arrangement of atoms, but they are enough to calculate various thermodynamic properties such as pressure and isothermal compressibility. The more intriguing part about liquid is the relationship between structural properties and potential parameters. This question can be seen as the inverse problem of liquid-state theory or as a coarse-graining problem, where the objective is to

parameterize a force field to reproduce a reference structure.[12–15] From theoretical perspective, the answer lies in the Henderson theorem, which states that the relationship between the pair potential and radial distribution function (RDF) is unique up to a constant at a given thermodynamic state, implying that the potential parameters at a specific thermodynamic state can be determined using the RDF.[16] The theory has been used as a starting point for different studies including design of nanostructure and systematic coarse-graining.[13,17] Both tasks of design of nanostructure and coarse-graining are important, particularly coarse-graining is essential for many practical systems, such as biological entities, where the number of particles (mostly liquid particles) are of the order of several thousands to millions, making all-atom MD (AAMD) simulation forbiddingly expensive for a reasonable length scales and timescales.[18–20]vvM Therefore, development of liquid-state theory can be useful for progress in other related areas such as coarse-graining, biology, etc.

Development of coarse-grained (CG) models and understanding of liquid-state theory relies on the repetitive AAMD and CGMD simulations, leading to repetitive MD simulation of a specific system. As most of the MD packages use a specific functional form to describe particles interaction such Lennard-Jones (LJ) potential, performing MD simulation of the LJ system might seem redundant, but it is the only reliable method available to the researchers. Surrogate models circumvent the difficulty associated with closed form solution of physical problems consisting of complex and many-body interactions by switching to an equation-free framework. It is of great value to apply machine learning and big data tools to provide insight into the behavior and physics of liquids.

1.2 Structure of the Dissertation

The current thesis is composed of five studies on the application of deep learning in the inverse liquid-state theory, coarse-graining, enhanced sampling of liquid properties, quasi-universality of simple liquids, as well as coarse-grained force field development for water and phase identification of water. Each chapter includes a separate section for introduction, methods, results and discussion, and conclusion of

the study. Appendix (A, B, C, D, and E) provides supporting details for the materials discussed in the main chapters of the thesis.

In the second chapter, deep learning methods are applied to find the solution of inverse liquid-state theory of Lennard-Jones particles, where finding an analytical solution is not straightforward. The model is used to coarse-grain simple multi-atom molecules into a single Lennard-Jones particles. In the third chapter, deep denoising autoencoder network is used to facilitate prediction of Lennard-Jones liquid properties from a single snapshot of molecular dynamics simulation. The prediction from a single frame after denoising outperform brute-force averaging by several order of magnitude more samples. In the fourth chapter, we explore quasi-universality of simple liquids, which refers to structural and dynamical similarity among simple liquids with different type of pair potential. We employ a statistical model to explain this similarity, followed by applying deep learning to find hidden scale physics in simple liquids. Additionally, we bridge concepts such as coarse-graining, Henderson theorem, deep learning, and quasi-universality together. In the fifth chapter, coarse-grained force field are developed to study water using neural networks. In the sixth chapter, graph neural network framework is developed alongside a python package to train machine learning models to classify different phases of water. The model uses edge-conditioned convolutional layer to map pairwise distance matrix to the corresponding phase, outperforming traditional methods trained over bond-order parameters or other human-engineering order parameters.

CHAPTER 2: Deep Inverse Liquid State Theory

2.1 Introduction

The Lennard-Jones (LJ) potential form is one of the widely used non-bonded pair potentials to investigate molecular scale phenomena. The LJ potential comprises the repulsive and attractive terms, which are usually represented using a standard 12-6 potential form with two parameters (C_{12}, C_6).[21] The 12-6 LJ potential is prominently used in molecular dynamics (MD) simulations to study physical, chemical, biological, and mechanical systems.[22] Once the underlying potential parameters and the thermodynamic state are specified, MD can compute various quantities of interest such as the radial distribution function (RDF). However, given a specific RDF, MD cannot directly predict the underlying potential parameters. In general, the estimation of potential parameters is a difficult task.[23] As per Henderson's theorem[16], the relationship between the pair potential and RDF is unique up to a constant at a given thermodynamic state, implying that the potential parameters at a specific thermodynamic state can be determined using the RDF. In this work, we explore the feasibility of force field development based on the Henderson's theorem using a data-driven approach combined with deep learning.

The inverse problem of parameterization of the LJ potential to reproduce a given RDF can also be viewed as a coarse-graining problem (where the objective is to develop a pair-potential between coarse-grained (CG) particles such that the RDF of the original system is reproduced) or as a solution to the inverse-problem of liquid-state theory.[24–27] Different frameworks, such as the fundamental measure theory[28,29], and integral equations[30], have been developed to address this problem. However, accuracy is an issue and generalizability to more complex systems is quite involved. An alternative route to reproduce a given RDF relies on MD to refine the potential parameters. For example, MD simulation data are either integrated with a theoretical framework (*e.g.* iterative Boltzmann inversion[24,31]) or used to optimize

statistical or empirical errors between the MD-calculated and given RDFs (*e.g.* relative entropy or the simplex minimization method[14,32]). MD-based approaches require thousands of simulations on a specific system of interest and the data is often not reused for other systems with similar mapping and underlying potential form (see Figure 1a).[33] The main bottleneck in reusing data for parameterization of a new system originates from the inherent complexity of physics-based model development as well as storing the data for a long period of time. Recent advancements in data-driven approaches can solve the reusability issue of MD data as well as obtaining the underlying physics of data, also known as the physics of big data[34], by surrogate models. Surrogate models circumvent the difficulty associated with closed form solution of physical problems consisting of complex and many-body interactions by switching to an equation-free framework. As a result, a data-driven approach avoids the development of a brute force, closed form analytical solution or repetitive MD simulation of a specific system.

During the last decade, data driven and machine learning (ML) methods have received enormous attention in computational analysis of physical problems.[34–37] Even though the usage of ML methods in scientific research and discovery dates back to several decades ago, only in the last five years researchers have embarked on using deep learning (DL)[38] in modeling and understanding of physical phenomena such as data-driven materials discovery[39–44], calculating the thermodynamic and ground states in multi-body systems,[45,46] phase transitions,[47] classification of phases in strongly correlated fermions,[48] quantum entanglement,[49] and many other applications[50]. During the last couple of years, researchers have started to develop deep neural networks (DNN) for CG model development.[51,52] Nevertheless, to the best of our knowledge, DL has not been used to solve the inverse problem of liquid-state theory, especially as a systematic CG method.

In this study, we develop a DNN to learn the relation between RDF of LJ particles at various thermodynamic states with the potential parameters (C_{12} and C_6). The dataset is generated using MD simulation of 26,000 distinct systems (each MD simulation is performed for 2 ns summing to 52 μs of total simulation time) with uniform sampling over a specific range of temperature, density and potential

parameters. Once the training part is complete on the single atom systems (referred to as the purely atomistic case), the knowledge acquired from the single atom systems or LJ particles is transferred to coarse-grain simple multi-atom molecules, with bonded and non-bonded interactions, into a single LJ particle. We refer to the CG model developed through this route as transfer learning based coarse-graining. DL model performance on parametrization of LJ particles and transfer learning based CG models is assessed using different metrics such as the deviation from ground truth potential parameters, total variation between predicted and specified RDFs, and Kullback-Leibler divergence (KL). The study presented here can also be viewed as a data-driven solution to the inverse problem of liquid-state theory (Deep Inverse Liquid-State Theory, DeepILST) for LJ particles and simple multi-atom molecules using DL (see Figure 1c for the DL-based methodology developed in the current study, which is also applicable to other potential forms as well as for other molecular structures). We show that DeepILST is also able to estimate other thermodynamic quantities like pressure and potential energy.

The rest of the chapter is organized as follows. In the next section, we describe the details of MD simulations for both multi-atom and single bead molecules and DNN development, as well as coarse-graining through transfer learning. In Section 2.3, we present the results of DL training as well as coarse-graining through transfer learning. Finally, we summarize the findings of this study.

2.2 Methods

2.2.1 Molecular Dynamics Simulation

The data for DNN training is generated via MD simulation of 26000 distinct systems with interaction potential parameters and thermodynamic states sampled uniformly over the range shown in Table 2.I. The LJ potential form is given by,

$$u(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{2.1}$$

where $C_{12}(=4\epsilon\sigma^{12})$ and $C_6(=4\epsilon\sigma^6)$ are the interaction potential parameters, and ϵ and σ are the energy- and length-scale parameters of the LJ pair potential, respectively. All the MD simulations are performed using GROMACS[53] using a timestep of 1 fs in the NVT ensemble. The temperature is controlled using the Nosé-Hoover thermostat with a time constant of 0.2 ps. Each LJ particle system is simulated for 2 ns and the RDF is calculated from the last 1.8 ns of the MD trajectory. RDF, the corresponding thermodynamic state, and potential parameters are stored as feature vectors to be fed as input data to the input layer of DNN, and the interatomic potential parameters constitute the output layer of DNN.

Three distinct simple multi-atom molecules such as CH_4, F_2, and CO are simulated to evaluate the development of coarse-grained potentials for these multi-atom molecules using the DNN developed for LJ particles. The simple multi-atom molecules, shown in Figure 2.1a, are modeled using the GROMOS force field[54]. The interactions between simple multi-atom molecules are described by both bonded and nonbonded potentials. The bonded interactions are modeled using GROMOS force field for bond and angle potential.[54] The nonbonded potential includes the van der Waals potential, described by the LJ potential given in Eq. 1, and electrostatic interactions given by the Coulomb potential, $u_{Coulomb}(\boldsymbol{r}; s) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$, where ϵ_0 denotes the dielectric permittivity of vacuum, q_i and q_j are the point charges on atoms *i* and *j*, respectively, and r_{ij} is the radial distance. The particle mesh Ewald algorithm is used to treat the long-range part of the electrostatic interaction.[55] Once the initial configuration of atoms is generated, energy minimization is performed, followed by 8 ns of production simulation, from which 6 ns is used to calculate the center of mass (COM) RDFs.

2.2.2 Deep Learning

In general, the many-body nature of interactions in liquids makes it difficult to develop a closed-form analytical relation between the pair-potential and RDF. The complex relationship between the underlying potential and RDF can be expressed as,

$$(C_{12}, C_6) = \boldsymbol{f}(g(r);\ T, \rho) \tag{2.2}$$

where $\boldsymbol{f}$ is a vector valued function, which is a nontrivial function of RDF, and thermodynamic state variables, T and ρ represent the temperature and density, respectively. r is the radial distance between the particles, and $g(r)$ is the RDF between particles. RDF is also related to the potential of mean force (PMF) by the expression, $g(r) = \exp(-\frac{U_{PMF}(r)}{k_B T})$, where k_B is the Boltzmann constant. The explicit relation between PMF and the pair potential is also nontrivial.

In this work, we use a feed-forward neural network (FNN) to estimate the function in Eq. 2.2 based on the universal approximation theorem, which states that FNN with enough capacity can approximate many continuous functions.[56] The data of each MD simulation is fed into DNN as,

$$x_i = \left(\boldsymbol{g}_i\,(r), \rho_i^1, \rho_i^2, \dots, \rho_i^p, T_i, T_i^2, \dots, T_i^p\right)_m \tag{2.3}$$

where x_i is the input vector composed of the concatenation of system i RDF (size of n) and thermodynamic states (each with size of p) in the dataset (D) with a total size of $m(= n + 2p)$, and i refers to the *i-th* LJ system ($x_i \in D$). The first *n* elements correspond to the value of RDF discretized between its minimum and maximum range, i.e., $(x_{i,l} = g_i(r_l < r \leq\ r_l + dr) / \max_{\mathrm{i}\in\mathrm{D}} g_i(r)\ \forall\ l \in \{1, 2, \dots, n\}$, where $dr = \frac{r_{max} - r_{min}}{n-1}$ and $r_l = r_{min} + (l-1)\ dr$). The remaining $2p$ elements of the input vector correspond to the different exponents of the scaled density and temperature, e.g., $x_{i,n+l} = (\rho_i^l - \min_{\mathrm{i}\in\mathrm{D}} \rho_i^l)/(\max_{\mathrm{i}\in\mathrm{D}} \rho_i^l - \min_{\mathrm{i}\in\mathrm{D}} \rho_i^l)\ \forall\ l \in \{1, 2, \dots, p\}$ define the input vector for density.

The simplest unit of a DNN, denoted as a perceptron or node, receives an input signal and applies a linear transformation, followed by a nonlinear activation function resulting in an output signal. Stacking nodes in width (within a layer) and in depth (denoted as stacking layers; hidden layers are shown with blue circles in Figure 2.1b) results in a multilayer perceptron, which can approximate many continuous functions.[38,57] With x_i as the input and the LJ potential parameters as the output, the DNN can be mathematically expressed as,

$$\left(C_{12,i}^{fnn}, C_{6,i}^{fnn}\right) = \phi_o(\boldsymbol{W}_o\phi_{n_h}(\ldots\phi_2(\boldsymbol{W}_2\phi_1(\boldsymbol{W}_1 x_i + \boldsymbol{b}_1) + \boldsymbol{b}_2)) + \boldsymbol{b}_o) \quad (2.4)$$

where ϕ_k is the nonlinear activation function of layer k given by $\phi_k(x) = \phi_k(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k) = \tanh(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k) + \text{sigmoid}(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k)$. Layer k receives a linear transformation of the output of layer $k-1$, given by $\phi_{k-1}(x)$, and applies the activation function ϕ_k to produce the output of layer k. Each layer, k, has weights $\boldsymbol{W}_k$ and bias $\boldsymbol{b}_k$ with dimensions, $\boldsymbol{W}_k \in R^{d_{k-1}\times d_k}$ and $\boldsymbol{b}_k \in R^{d_k}$, where d_k is the number of nodes in layer k. n_h denotes the number of hidden layers. Summation of the tangent hyperbolic and sigmoid nonlinearities ($\tanh x + \text{sigmoid } x$) is used between layers, except for the output layer (ϕ_o) where the sigmoid nonlinearity is used (see the supporting information Eq. A.4-7).

As shown in Figure 2.1c, once the data generation is complete, the DNN is trained to obtain the near optimal weights and biases through minimization of the DNN loss function. The loss function minimization is performed with a backpropagation algorithm. The FNN is trained based on the mean square loss function where the mean-squared error (MSE) between the ground truth and the predicted parameters is optimized. The adaptive moment estimation (Adam) optimizer [58] is used to minimize the loss function ($\epsilon_{\mathcal{L}}(\theta, D)$), which can be expressed as,

$$\epsilon_{\mathcal{L}}(\theta, D) = \frac{1}{2|D|}\sum_{i\in D}\sum_{j\in\{C_{12},C_6\}}\left(v_{j,GT}^{(i)} - v_{j,p}^{(i)}(\theta, x_i)\right)^2 \quad (2.5)$$

where θ represents the free parameters (weights and biases) of DNN. $v_{j,GT}^{(i)}$ and $v_{j,p}^{(i)}(\theta)$ are the ground truth and DNN-predicted scaled LJ interaction parameters (with respect to the minimum and maximum in Table 1) of the i^{th} data point in the dataset. $|D|$ denotes the size of the training dataset. The training dataset corresponds to about 75 percent of the data obtained from MD simulations, and they are used to find optimal weights and biases of the DNN. The validation dataset (about 12.5 percent of the MD dataset) is used as a metric during the loss function minimization to monitor network performance, avoiding overfitting or underfitting in the learning process, *i.e.*, they are fed into the network to determine the value of the loss function, but they do not contribute to the backpropagation. Furthermore, during

training, we use the dropout technique for the second hidden layer to avoid overfitting.[59] The dropout technique drops nodes in layers with a specific probability, therefore, training only a fraction of weights ($\boldsymbol{W}_2$) and biases ($\boldsymbol{b}_2$) of the second hidden layer at each training step. In this study, the FNN units are dropped randomly with a probability of 0.25. Improvements achieved using the dropout technique are mainly due to preventing nodes from undesirable co-adapting and it far outweighs the ones achieved with L1 and L2 regularizations.

The training and design of DNN continues until a good performance is achieved on both the training and validation datasets. Once the network is trained, we check its loss function value on the testing dataset (the remaining 12.5 percent of MD dataset) to ensure its generalizability. If the network performance shows overfitting or underfitting on the validation or training datasets, its architecture involving the number of layers, the number of nodes in each layer, and activation functions are changed by trying rectified linear unit, sigmoid, or tangent hyperbolic, or a combination of them until a reasonable performance is achieved. Based on the network performance on the entire dataset, the network with two hidden layers with 48 and 15 nodes is selected, and an exponent of three for the thermodynamic states is chosen ($p = 3$). All the results presented in the main manuscript are obtained using this network (see the supporting information section SII for more details on training).

2.2.3 Transfer Learning-Based Coarse-Graining

Next, we consider coarse-graining simple multi-atom molecules into single beads to preserve the structure (RDF) of the multi-atom molecule. Specifically, a multi-atom molecule is coarse-grained into a single bead such that the center of mass (COM) RDF of the multi-atom molecule system is preserved in the single bead system. The interactions among the single beads are represented by the 12-6 LJ potential and the interaction potential parameters (C_6, C_{12}) are obtained by transferring the DNN knowledge of LJ particles – this approach is referred to as transfer learning based coarse-graining. As shown in Figure 2.1c (dashed arrows in the figure), the COM RDF and the thermodynamic state of a simple multi-atom

molecule are fed into the DNN as an input. The DNN provides the interatomic potential parameters as an output for the single bead representation of the multi-atom molecule. Once the potential parameters are available, MD simulations can be performed using the single bead representation of the multi-atom molecule to estimate RDF and other properties of the CG multi-atom system. It is important to note that, typically, coarse-graining introduces errors in the estimation of various properties of the original multi-atom system. Here, we estimate the error in the RDF obtained from the DNN predicted potential parameters by using two measures: the first is the error in the total variation between CG and AA RDFs and the second is the KL divergence.[14,33] The error in the total variation between CG and AA RDFs is calculated using the expression,

$$\epsilon_{rdf} = \frac{\int_0^{r_{cf}} |g_{CG}(r) - g_{AA}(r)|\, r^2 dr}{\int_0^{r_{cf}} g_{AA}(r) r^2 dr} \tag{2.6}$$

where $g_{CG}(r)$ and $g_{AA}(r)$ are radial distribution functions of CG and AA models, respectively. The error is also estimated using the KL metric using the expression,

$$\epsilon_{KL} = \beta\langle U_{CG} - U_{AA}\rangle_{AA} - \beta\langle F_{CG} - F_{AA}\rangle_{AA} + \langle S_{map}\rangle_{AA} \tag{2.7}$$

where $U(=\sum_{i<j} u(r_{ij}))$ and F are the potential energy and free energy of the system in the AA and CG systems, and $\langle S_{map}\rangle_{AA}$ is the mapping entropy, which is not a function of the potential parameters. Appendix A.1 contains mathematical details on KL divergence calculation, as well as its convexity and the existence and uniqueness of a global minimum.[60,61]

2.3 Results and Discussions

We examine the performance of DNN by considering two cases – first, the generalizability and transferability of the interatomic potential parameterization for LJ particles (this is in fact the development of atomistic force fields for single atom particles) is investigated, and second, transfer learning for CG force-field development is considered. Generalizability refers to the use of DNN to estimate potential parameters for LJ particles for thermodynamic states that fall within the range of the

training dataset, i.e., the thermodynamic state considered to establish generalizability is not part of the training dataset, but the thermodynamic state lies within training dataset thermodynamic states that are part of the training dataset. Transferability refers to the use of DNN to estimate potential parameters for LJ particles for thermodynamic states that fall outside the range of the training dataset, i.e., the thermodynamic states considered are not within the range shown in Table 2.1. Transfer learning, as discussed in Section 2.4, refers to the use of DNN to estimate potential parameters for CG representation of simple multi-atom molecules shown in Figure 1a.

Case 1. Parameterization of Lennard-Jones Particles

DNN is trained to reproduce the potential parameters of LJ particles using RDFs and thermodynamic state variables. The network architecture (the number of layers and the number of nodes in each layer) is optimized based on its performance on both the training and validation datasets (see Figure A.3 for loss function minimization during each iteration of training; the total number of iterations for each network is about 20,000,000). Generalizability of the network is assessed through its performance on the test dataset, which is not seen until the DNN training step is complete.

One-to-one comparison of the DNN predicted and ground truth potential parameters (parameters used in MD simulation) are shown in Figure 2.2. We note that all the points are distributed almost uniformly around the one-to-one mapping line, *i.e.*, the line on which the ground truth and DNN predicted parameters match exactly. Figure 2.2a and 2.2e show the training dataset results for the prediction of C_{12} and C_6, respectively. While thermodynamic states and RDF vary for each point, the DNN is able to relate them correctly to the underlying potential parameters. Similarly, Figure 2.2b and 2.2f show the validation dataset results for predictability of C_{12} and C_6, respectively. Validation dataset is used for the design of DNN to avoid overfitting and is not part of the dataset used for minimization of the loss function. Figure 2.2c and 2.2g show the generalizability of DNN for the prediction of C_{12} and C_6, respectively, for unforeseen data during the training of DNN.

The accuracy of the model is measured based on the loss function value of the training dataset (see Eqs. A.8-10 for the definition of bandwidth and accuracy, as well as the definition of the mean absolute percentage error (MAPE)[62], defined as $\epsilon_{\boldsymbol{MAPE,j}} = 100 \times \frac{\sum_{i \in D} |v_{j,DNN}^{(i)} - v_{j,GT}^{(i)}|}{\sum_{i \in D} |v_{j,GT}^{(i)}|}$, where i represents the *i-th* data point in the dataset, and v_j is either C_{12} or C_6). The dash-dotted lines, shown in Figure 2.2, are parallel to the one-to-one mapping line and at a distance equal to four times the square of the loss function value of the training dataset. These lines show the interval in which about 99 percent of DNN prediction points are located compared with ground truth of the training dataset. The same accuracy lines are shown for the rest of the datasets, which shows that most of the data of validation and testing datasets lies in this region (close to 99 percent of points). The MAPE is also shown for each dataset and potential parameters. C_{12} and C_6 of training, validation, and testing datasets have about 1.7 and 4.2 % MAPE, respectively, indicating high accuracy of prediction.

The transferability of DNN to thermodynamic states outside the range of the dataset is also investigated. For this, we estimate the potential parameters by considering the temperature range of [200, 500] K and density range of [1.96, 20.49] nm^{-3}, excluding the range shown in Table 2.1 (temperature range of [290, 400] K and density range of [8.0, 19.4] nm^{-3}). One-to-one comparison (prediction versus ground truth) for the transferability dataset is shown in Figure 2.2d and 2.2h, indicating about 8.7 % and 39.8 % MAPE for C_{12} and C_6, respectively. Similarly, about half of the points of transferability dataset lie in the region bounded by the accuracy lines. This result indicates that caution should be exercised in using the DL model outside the range of the training data. However, DNN shows clear a correlation between the predicted and ground truth parameters (see the supporting information A.2 for discussion on the details of the transferability dataset selection).

To evaluate the performance of DNN for other thermodynamic properties of a given system, MD simulations are performed on Argon particles for 121 different thermodynamic states with a uniform sampling on the temperature and density ranges shown in Table 1 with each having 11 points. The

pressure calculated using the DNN estimated potential parameters (shown in Figure 2.3) has a mean error of 17% compared to AA MD pressure for the 121 points used in the dataset. This error is reasonably small compared to the pressure fluctuations observed in MD simulations. The comparison between mean values of LJ parameters predicted by DNN and ground truth LJ parameters of Argon are shown in Table 2.2 (the details on the calculation of the thermodynamic quantities including pressure and total energy are presented in the supporting information SI). As Table 2 indicates, the network can map all 121 points into a single value of LJ parameter with a small deviation, as well as a small error, compared to the MD ground truth parameters (6.18 percent for C_{12} and 0.16 percent for C_6 relative to the ground truth LJ parameters of Argon)

Case 2. Coarse-graining

The DNN results for the parameterization of LJ particles exhibit no more than 4.4% MAPE over the dataset, which implies that FNN is an efficient approach to solve the inverse problem of the liquid-state theory, at least for the LJ particles. To investigate the transferability of knowledge acquired from LJ particles as a new coarse-graining route, single bead CG models of simple multi-atom molecules such as carbon monoxide, fluorine, and methane are developed. As stated in Section 2.3, the COM RDFs of these systems are first obtained using AAMD simulations. Then, the COM RDFs and thermodynamic states (midpoint of Table 1) are fed into the DNN (the procedure is shown with dashed line in Figure 2.1c). The comparison of the COM RDFs of multi-atom molecules and CG model RDFs is shown in Figure 2.4. Figure 2.4 also shows the results of CG RDFs of multi-atom molecules for LJ pair-potential parameterized by simplex and relative entropy methods. The results indicate that DNN parameterized force fields are indistinguishable from the other two methods. Considering that DNN is a single shot method with no iterations, its speed to derive the CG force field is faster compared to the simplex and relative entropy methods (see Figures A.11 and A.12 as well as Table A.1 in the supporting information for further details and additional case studies). Following the procedure shown in Figure 1c, we assess the accuracy of CG models with two additional metrics, *i.e.*, ϵ_{rdf} and ϵ_{KL}. In Figure 2.5a-b, ϵ_{rdf} is shown

on a mesh discretizing the space of LJ potential parameters. The mesh points have different values of C_{12} and C_6, and using this data, one can find the LJ parameters with a minimum value of ϵ_{rdf}. The minimum point of ϵ_{rdf} lies in the vicinity of the LJ parameters predicted by the DNN. Based on Eq. 2.8, ϵ_{KL} between simple multi-atom molecules and the LJ particles is shown in Figure 2.5e-f. Similar to ϵ_{rdf}, the minimum value of ϵ_{KL} is close to the DNN predicted parameters. The convexity of ϵ_{KL} ensures that the local minimum of ϵ_{KL} found within the LJ parameter space is a global minimum (for a more detailed discussion on KL divergence and convexity, see Appendix A.1 or reference [14] and references therein). Both error metrics show a small deviation from the DNN predicted LJ parameters with a distance less than 0.1% of the maximum error in the investigated parameter space. As DNN does not have prior knowledge about the information theory (KL metric) or the statistical mechanics metric (error in the total variation of the RDF), we can conclude that DL is a good coarse-graining strategy as it performs well on both metrics.

2.4 Conclusions

In this study, a deep neural network is used for atom-agnostic parametrization of the Lennard-Jones potential at different thermodynamic states. The DNN demonstrates good performance for two cases – parameterization of LJ particles and development of single-bead CG LJ potentials for simple multi-atom molecules through transfer learning obtained from LJ particles. The transferability and generalizability of the method are investigated by computing the total variation in the radial distribution function and Kullback-Leibler divergence for the coarse-grained model development. Our results indicate that deep learning is able to compute the solution to the inverse-problem of liquid-state theory (DeepILST) under the assumption of a predetermined pair potential in a coarse-grained model.

2.5 Figures and Table

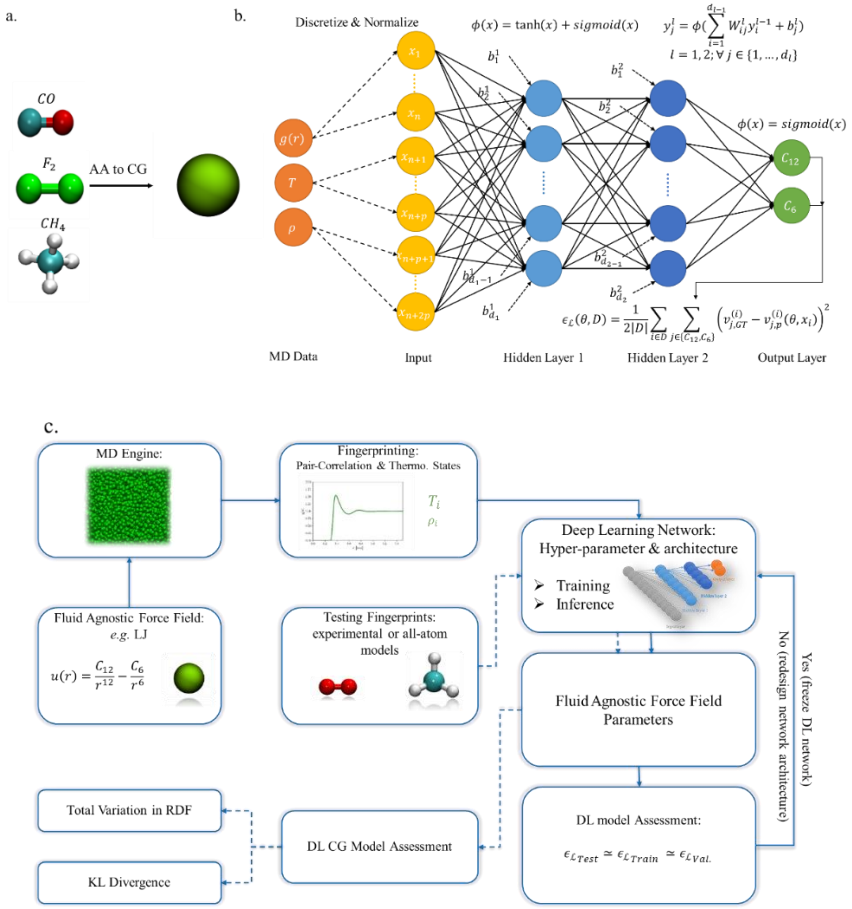


Figure 2.1. a. AA and CG model representations showing how different AA models are mapped to the single bead CG model with different potential parameters. b. Schematic representation of a deep neural network c. Deep learning-based methodology employed in this work to develop an atom agnostic framework for inverse liquid state theory. First, MD simulations are performed on a variety of LJ fluids at different thermodynamic states to generate RDFs, which are used for training, validation, and testing. 75 percent of the RDFs (inputs) are subsequently used for training the DNN to generate atom agnostic force field parameters (outputs). The DNN loss function is monitored on 12.5 percent of the data, and then tested on the rest of the data. The DNN architecture is redesigned (as necessary) until the errors from training, testing and validation are within a specific tolerance (the solid lines denote the DL training and inference stages.). Once the training and assessment of the network is accomplished, either RDFs obtained from AA models or experimental data are used as input to the DNN to predict the force field parameters (transferring learning for coarse-graining is shown with dashed lines). Finally, the accuracy of the force field parameters is assessed using KL divergence and the total variation in RDFs of AA and CG models.

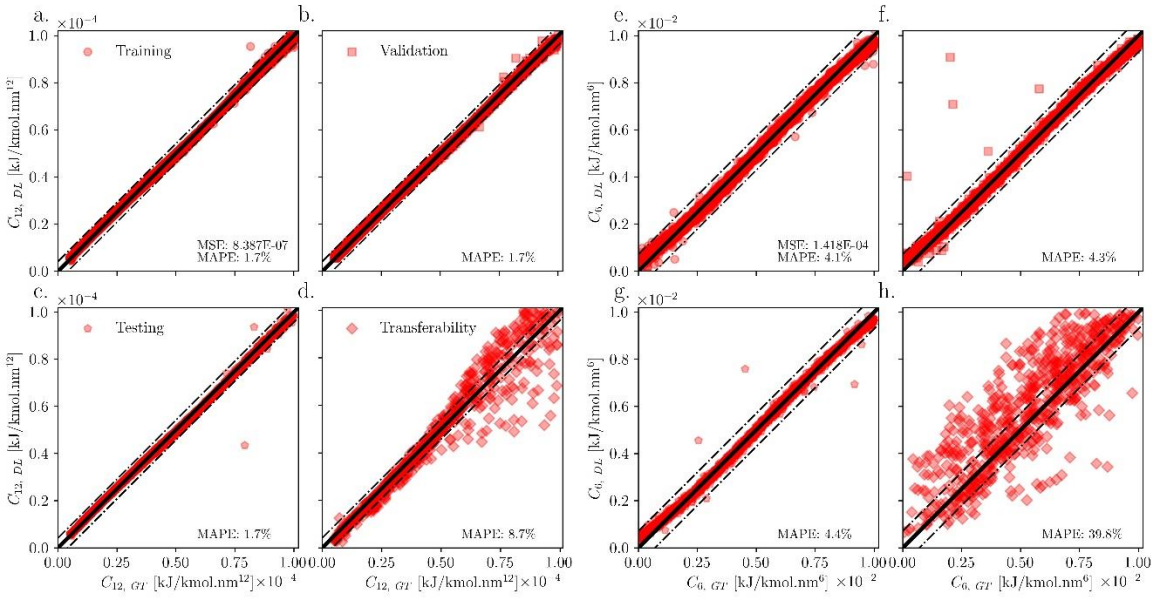


Figure 2.2. Comparison of the pair potential parameters determined from the DNN with the ground truth values for training, validation, testing, and transferability datasets. a-d. C_{12} parameter from training, validation, testing and transferability datasets. e-h. C_6 parameter from training, validation, testing and transferability datasets. The mean absolute percentage error is also shown for each dataset (red circles, square, pentagon, and diamond points represent training, validation, testing, and transferability datasets, respectively, while black solid lines show a one-to-one mapping and black dash-dot lines denote lines parallel to the one-to-one mapping line with a distance of four times the square root of loss function of the training dataset. 99 percent of the training data are enclosed in this region. The region also encloses about 99 percent of data of validation and testing datasets).

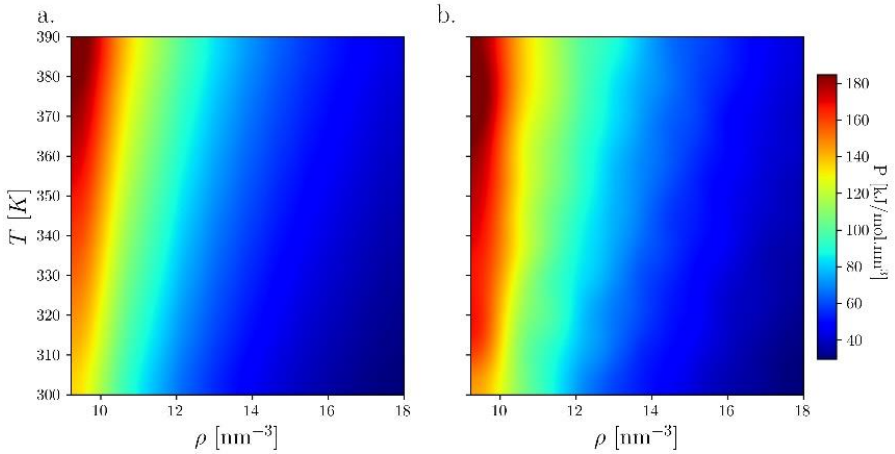


Figure 2.3. Calculation of the bulk pressure of Argon for different thermodynamic states. a. molecular dynamics simulations b. deep learning predicted potential parameters-based pressure.

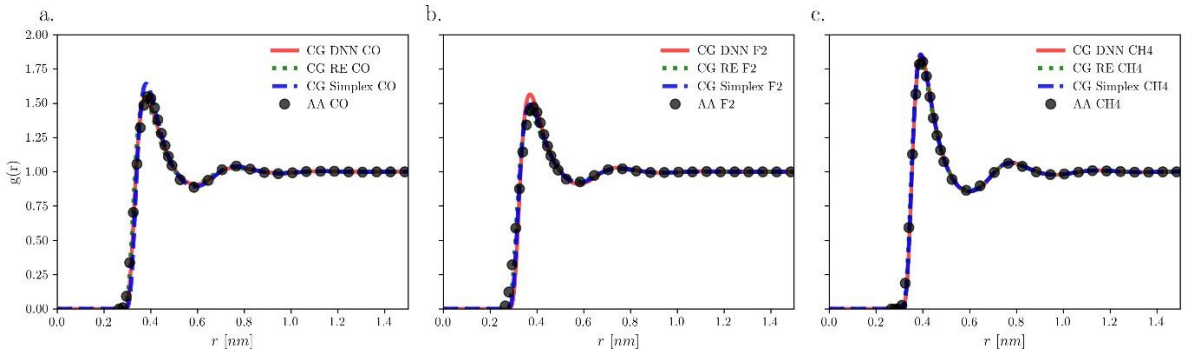


Figure 2.4. Comparison of RDFs obtained with DNN-based (solid red line), relative entropy (dotted green line), simplex (dash-dot blue line) CG models and AA model. All the three methods show excellent match with the AA model results (shown as black circles). a. CO b. F_2 c. CH_4

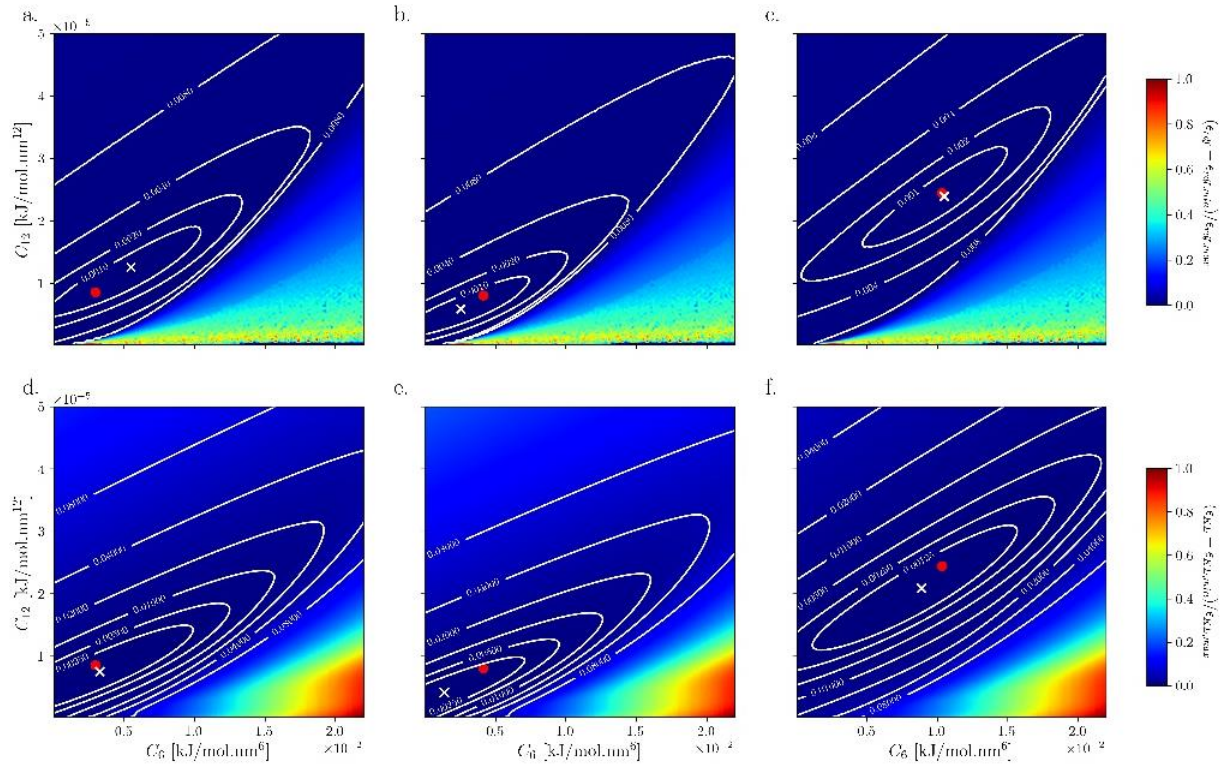


Figure 2.5. Assessment of the coarse-grained force fields through total variation in RDF (first row) and Kullback-Leibler divergence (second row) criteria. The parameter space is discretized into a mesh and ϵ_{rdf} and ϵ_{KL} are computed at each mesh point. White cross points are the minimum values obtained from the two metrics, while red filled circles are predictions from deep learning. a, d. CO b, e. F_2 c, f. CH_4

Table 2.1. Thermodynamic states and the range of parameters used during MD simulation of single bead systems.

	Thermodynamic State		Parameter Range	
	$\rho\ [nm^{-3}]$	$T\ [K]$	$C_{12}\ [\frac{kJ}{mol.nm^{12}}]$	$C_6\ [\frac{kJ}{mol.nm^{6}}]$
min	8.0	290.0	0.000005	0.00005
max	19.4	400.0	0.0001	0.01

CHAPTER 3: Deep Denoising Autoencoder for Liquid Properties

3.1 Introduction

Molecular dynamics (MD) is a popularly used computational tool to compute microscopic and macroscopic properties and understand physics of atomic systems in many areas of science and engineering including soft and hard matter.[63] The Newtonian equations of motion are solved numerically by MD simulation to determine the trajectory of every atom in the system as a function of position and time. In the case of liquids, which is the problem considered in this work, the atomic trajectories are used to compute properties such as the radial distribution function (RDF), pressure, isothermal compressibility, etc. Among these properties, RDF is an important quantity as it not only characterizes the local structure of the atomistic system, but it can also be used to calculate thermodynamic properties such as the pressure, energy, and isothermal compressibility.[64] To estimate these properties, MD simulations are performed over very long times (ranging from several nanoseconds to several tens to hundreds of nanoseconds depending on the property and system of interest). Once equilibrium of the atomic system is attained, physical properties are computed using data from each MD snapshot and averaged over all the MD snapshots to minimize fluctuation in the estimated property. Several attempts have been made to incorporate signal processing and other statistical concepts with MD simulation to reduce fluctuation and noise in estimated properties.[65–70] However, signal processing and Fourier series expansion-based methods generally suffer from mathematical complexity and poor performance in the context of atomistic scale simulation. Reducing the computational time to calculate various properties of atomistic systems can not only accelerate fundamental studies on soft- and hard-matter but also accelerate the development of bottom-up coarse-grained models[33,71–73] where an accurate calculation of structural and thermodynamic properties is required and is commonly achieved through long and repetitive MD simulations.

MD simulations generate a tremendous amount of data, which hasn't been exploited to a great extent until

recently. The advent of concepts such as physics of big data and data-driven methods [34] during the last five years has opened up opportunities to address the problem of obtaining accurate RDF (and other properties, e.g. thermodynamic properties) from a short MD simulation instead of a long one. Furthermore, methods such as hybrid Monte Carlo[74,75] for systems with slow relaxation can generate extensive datasets with rich physics such as the liquid-liquid phase-transition[76,77] and nucleation[78]. Among the various data-driven methods, deep learning is a promising method, which has been successfully employed in various fields, including atomic scale simulation.[34,38,42,79–84] Various deep neural networks have been developed,[57] among which, autoencoder networks learn to encode data into a low-dimension representation ignoring noise in the data.[85] Autoencoder networks map the data into a low-dimension, and then they map the low-dimension encoding back to the dimension of the data in an unsupervised manner. Recent studies have successfully employed various types of autoencoder networks in atomistic scale simulation.[86–88] Among the various autoencoder networks, the deep denoising autoencoder (DAE) is an autoencoder network in which the input data is intentionally corrupted by adding additional noise, which, in turn, improves the robustness of the DAE network training compared with simple autoencoder network.[89]

In this study, we explore the possibility of using the DAE network to estimate RDF of simple liquids using a single snapshot from MD instead of hundreds of snapshots. RDF obtained from a single snapshot of MD simulation has large fluctuations, which mimics the noise used in the DAE training. In particular, the DAE used in this study learns to map a single snapshot RDF into a temporally averaged RDF (obtained through averaging of RDF over a long MD simulation). Note that the fluctuations in a single snapshot RDF are inherent features of MD data. To train the DAE network, large-scale MD simulations of 12000 Lennard-Jones pair potentials at various thermodynamic states are carried out with a total simulation time of $24\,\mu s$. The data is generated through calculation of 800 single snapshots and corresponding temporally averaged RDFs of 12000 Lennard-Jones systems summing up to 9.6 million single snapshot RDFs. Once the data is generated, it is used to train the DAE network through

minimization of a loss function between DAE predicted and corresponding temporally averaged RDFs, *i.e.,* the DAE learns to map any single snapshot RDF to its corresponding temporally averaged RDF. Furthermore, we combine traditional dimension reduction methods, specifically principal component analysis,[90,91] with the DAE network. The objective of this approach is to take advantage of simplicity of a traditional method to reduce the size and training time of DAE network and to increase accuracy of the traditional method to the level of deep learning-based method. In this scheme, we change the DAE input vector from RDF to values of principle components.

Once the DAE network is trained, we assess its performance in prediction of RDF of Lennard-Jones systems based on a single snapshot RDF of MD simulation. We investigate the generalizability of DAE network to other pair potentials describing monoatomic systems belonging to the class of simple liquids. In particular, we investigate the exponential,[92] inverse-power-law (IPL),[93] and Yukawa[94] pair potentials. The methodology can be extended to other phases of matter as well as to complex systems described by, for example, the Stillinger-Weber potential[95] with interesting physics such as liquid-liquid phase transition.[77]

The rest of the chapter is organized as follows. First, we describe the details of MD simulations and DAE network. Then, we present results on the DAE network performance for structure and thermodynamic properties of Lennard-Jones liquids, followed by assessment of generalizability of DAE to other simple liquids. Finally, we summarize the results of this study.

3.2 Methods

3.2.1 Molecular Dynamics Simulation

The training data for DAE is obtained through MD simulation of 12000 standard Lennard-Jones pair potentials at various thermodynamic states (shown in Table I) for a total simulation time of 24 μs. The Lennard-Jones potential form can be expressed as,

$$u(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{3.1}$$

where $C_{12}(= 4\epsilon\sigma^{12})$ and $C_6(= 4\epsilon\sigma^6)$ are the potential parameters, and ϵ and σ are the energy- and length-scale parameters of the Lennard-Jones potential, respectively. Generalizability of the DAE network to other simple liquid pair potentials is assessed for IPL, exponential, and Yukawa pair potentials given by,

$$u_{IPL}(r) = \frac{C_p}{r^p} - \frac{C_q}{r^q} \tag{3.2}$$

$$u_{EXP}(r) = A\exp(-k_D r) \tag{3.3}$$

$$u_{Yukawa}(r) = A\frac{\exp(-k_D r)}{k_D r} \tag{3.4}$$

where $C_p(= 4\epsilon\sigma^p)$ and $C_q(= 4\epsilon\sigma^q)$ are the repulsive and attractive components of the IPL pair potential with exponents p and $q(p > q)$, and A and k_D are the strength and inverse screening length parameters of EXP and Yukawa pair potentials (potential parameters are given in the Supplementary Information S.2). All the MD simulations are performed using GROMACS[53] with a timestep of 1 fs in the NVT ensemble with the temperature being controlled with a time constant of 0.2 ps using the Nosé-Hoover thermostat. 800 single snapshot RDFs of each system (for a total of 9.6 million single snapshots) at various times are calculated from 2 ns of simulation, where the first 200 ps is discarded. Each RDF is stored as an input vector along with its corresponding thermodynamic state, and the DAE output is the temporally averaged RDF. The temporally averaged RDF ($g_{ref}(r)$) is calculated using the expression,

$$g_{ref}(r) = \frac{1}{N}\sum_{j=1}^{N} g_{t_j}(r) \tag{3.5}$$

where $g_{t_j}(r)$ is the RDF at time t_j corresponding to snapshot j, and, j proceeds over all snapshots of MD trajectory, i.e., $j \in \{1, 2, \dots, N\}$. Once the temporally averaged RDF is obtained, the thermodynamic

properties such as pressure (P) and isothermal compressibility ($\mathcal{X}_T$) can be calculated using the expressions,

$$P = \rho k_B T - \frac{2\pi\rho^2}{3}\int_0^r \frac{\partial u}{\partial r} g(r) r^3 dr \tag{3.6}$$

$$\mathcal{X}_T = \frac{1}{\rho T}[1 + 4\pi\rho \int_0^\infty (g(r) - 1) r^2 dr] \tag{3.7}$$

where T and ρ are the temperature and density, respectively, and k_B is the Boltzmann constant. In practice, the integration for calculation of isothermal compressibility is done for a finite cut-off, which we set to 1.5 nm as the RDF is almost homogeneous beyond this distance.

The temporally averaged RDF of a system at a given thermodynamic state (T, ρ) can also be written as the summation of a single snapshot RDF at time t_j ($g_{t_j}(r;\ T, \rho)$) and a distance- and time-dependent fluctuation ($\epsilon_{t_j}(r;\ T, \rho)$) expressed as,

$$g_{ref}(r;\ T, \rho) = g_{t_j}(r;\ T, \rho) + \epsilon_{t_j}(r;\ T, \rho) \tag{3.8}$$

where $\epsilon_{t_j}(r;\ T, \rho)$ is the fluctuation in the RDF of a single snapshot at time t_j. The DAE in this study learns to eliminate the fluctuation term in Eq. 3.8 (discussed in more detail in the next section, see Figure 3.1a for the workflow of current study).

The generalizability of the DAE network to various simple liquids with different pair potentials is also investigated in terms of RDF error[33] expressed as,

$$\epsilon_{rdf}(g_{est}(r), g_{ref}(r)) = \frac{\int_0^{r_{cf}} |g_{est}(r) - g_{ref}(r)|\, r^2 dr}{\int_0^{r_{cf}} g_{ref}(r) r^2 dr} \tag{3.9}$$

where $g_{est}(r)$ and $g_{ref}(r)$ are the estimated and reference (longtime averaged MD simulation data) RDFs, respectively.

3.2.2 Denoising Autoencoder Network

Deep neural networks are generally composed of simple units known as a perceptron or node, which are stacked in width (referred to as a layer) and in depth (successive layers). Each node receives an input signal, applies a linear transformation, followed by a nonlinear activation function resulting in an output signal.[38,57] A simple autoencoder network has two parts, namely encoder and decoder, both of which are usually modeled using deep neural networks. The encoder network receives the input and maps it into a low-dimensional (usually smaller than the input dimension) space, also known as a latent space. Then, the latent space is fed into the decoder network, which maps it back to a dimension equal to the input data dimension. The main difference between a simple autoencoder and a DAE is intentional corruption of the input of DAE through noise, which, in turn, enhances the robustness of DAE compared to a simple autoencoder (schematic representation of a DAE is shown in Figure 3.1b).

The DAE network employed in this study learns to map a single snapshot RDF and the corresponding thermodynamic state to its temporally averaged (long-time average) RDF of an MD simulation. The DAE network is trained over 12000 distinct Lennard-Jones liquids at various thermodynamic states with 9.6 million single snapshot RDFs. It is important to note that noise is not added to the input RDF as the inherent fluctuations in a single snapshot RDF play the role of noise in training the DAE network, i.e., each of the 800 RDFs of a given system is a noisy version of the temporally averaged RDF of the same system. Mathematically, the DAE network performs the following mapping,

$$g_{DAE;\rho,T}(r) = DAE(g_{t_j}(r);\rho,T) \tag{3.10}$$

where $g_{DAE;\ \rho,T}(r)$ and $g_{t_j,\rho,T}(r)$ are temporally averaged RDF predicted by the DAE network and MD RDF at time t_j(corresponding to a specific snapshot), respectively. The parameters (weights and biases of the encoder and decoder) in the DAE network are optimized to minimize the following expectation,

$$\min \mathbb{E}\left[\left|g_{ref;\rho,T}(r) - f_d\left(f_e\left(g_{t_j}(r);T,\rho\right)\right)\right|\right] \tag{3.11}$$

where f_e and f_d are encoder and decoder functions, respectively. Both functions are selected from a flexible class of functions modeled using deep neural network and can be expressed as,

$$f_e(x_{i,t_j}) = \phi_o^e[\boldsymbol{W}_o^e \phi_{n_h}^e \left(\dots \phi_2^e[\boldsymbol{W}_2^e \phi_1^e \left(\boldsymbol{W}_1^e x_{i,t_j} + \boldsymbol{b}_1^e\right) + \boldsymbol{b}_2^e\ ]\right) + \boldsymbol{b}_o^e] \tag{3.12}$$

$$y_{i,t_j} = \phi^l(\boldsymbol{W}^l f_e\left(x_{i,t_j}\right) + b^l) \tag{3.13}$$

$$f_d(y_{i,t_j}) = \phi_o^d[\boldsymbol{W}_o^d \phi_{n_h}^d \left(\dots \phi_2^d[\boldsymbol{W}_2^d \phi_1^d \left(\boldsymbol{W}_1^d y_{i,t_j} + \boldsymbol{b}_1^d\right) + \boldsymbol{b}_2^d\ ]\right) + \boldsymbol{b}_o^d] \tag{3.14}$$

where x_{i,t_j} is the input vector composed of a concatenation of single snapshot RDF at time t_j, temperature, and density ($x_{i,t_j} = [g_{i,t_j}(r), T_i, \rho_i]$) for the Lennard-Jones system i, y_{i,t_j} is the output of encoder, *i.e.*, latent space ($y_{i,t_j} = \phi^l(\boldsymbol{W}^l f_e\left(x_{i,t_j}\right) + b^l)$), and index i specifies a specific Lennard-Jones system in the dataset (D), and index j represents snapshot j RDF of system i at time t_j ($\in \{T\}_i = \{1, 2, \dots, T_n\}_i$, where $n(= 800)$ is the total number of snapshots stored from MD simulation of system i). $g_{i,t_j}(r)$ is composed of RDF values at various radial distances with a bin size of 0.005 nm for the RDF interval [0.25, 1.5] nm. ϕ_k is the nonlinear activation function of layer k, which receives a linear transformation of output of layer *k-1* ($\boldsymbol{W}_j \phi_{k-1} + \boldsymbol{b}_k$). Each layer has its own weights $\boldsymbol{W}_k$ and bias $\boldsymbol{b}_k$with consistent dimensions corresponding to the output of its previous layer and its number of nodes ($\boldsymbol{W}_k \in R^{d_{k-1} \times d_k}$ and $\boldsymbol{b}_k \in R^{d_k}$, where d_k is the number of nodes in layer k). Optimal DAE used for obtaining the results in this study has $tanh$ nonlinearity function for all the layers except the decoder output layer, which has no nonlinearity.

In practice, DAE minimizes the following loss function,

$$\epsilon_{\mathcal{L}_{DAE}} = \epsilon_{\mathcal{L}_R} + \epsilon_{\mathcal{L}_{Reg}} \tag{3.15}$$

where $\epsilon_{\mathcal{L}_R}$ and $\epsilon_{\mathcal{L}_{Reg}}$ are the reconstruction and regularization losses, respectively. $\epsilon_{\mathcal{L}_R}$defines the match between the output of the DAE network with the ground truth (temporally averaged RDF of MD

simulation). In this study, we use the mean absolute error (MAE) (the MAE between output of DAE and temporally averaged RDFs) to model the reconstruction loss, i.e.,

$$\epsilon_{\mathcal{L}_R} = \frac{1}{|D||T|} \sum_{i\in\{D\}} \sum_{t\in\{T_i\}} \left| x_{i;T_i,\rho_i} - f_d\left(f_e\left(x_{i,t_j}; T_i, \rho_i\right)\right)\right| \tag{3.16}$$

where $|D|$ and $|T|$ are the number of Lennard-Jones systems and snapshots, respectively. The regularization loss is defined as,

$$\epsilon_{\mathcal{L}_{Reg}} = \gamma \sum_h \|\boldsymbol{W}_h\|^2 \tag{3.17}$$

where $\boldsymbol{W}_h$ are the weights of hidden layers of both encoders and decoders in Eq. 3.12 and 3.14, and γ is the regularization factor added for stability purpose and preventing over-fitting ($\gamma \in \{0.001, 0.0005, 0.0001\}$). The adaptive moment estimation optimizer [58] is employed to minimize the loss function ($\epsilon_{\mathcal{L}_{DAE}}$). In addition to the above regularization procedure, the dropout technique is also applied to DAE network layers to avoid overfitting.[59] The dropout technique randomly drops nodes in different layers with a specific probability, therefore, training only a fraction of weights ($\boldsymbol{W}$) and biases ($\boldsymbol{b}$) of each layer at each training step. The DAE network nodes are dropped with a probability of 0.25. Various DAE architectures are trained in order to find the optimal network with satisfactory performance over both the training and validation datasets (we divide the data set of 12000 Lennard-Jones systems into two sets, namely training and validation data sets with 9600 and 2400 systems assigned to each set, respectively. The simple liquids RDFs play the role of testing data set in this study, see the Appendix B.1). To find optimal networks, we perform multiple training with various architectures. Based on the results of initial experiments, we guide our architecture selection and narrow down our search space to the three-layer networks with 2 different architectures, which are trained extensively. The optimal network encoder has three fully connected layers with dimensions of 250, 200, and 150, respectively. The latent space has a dimension of 100, which gets fed into the decoder with three fully connected layers each with a dimension of 150, 200, and 250, respectively.

Furthermore, we determine the sampling gain (N_T) as the minimum number of single snapshot RDFs needed in MD such that the average of N_T RDFs ($g_{MD_{N_T}}(r) = (1/N_T)\sum_{j=1}^{N_T} g_{t_j}(r)$) has a smaller error, $\epsilon_{rdf}\left(g_{MD_N}(r)\ , g_{ref}(r)\right)$ compared to the DAE predicted RDF error $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$. As DAE requires only one snapshot, the sampling gain can be expressed as,

$$N_T = \{\min N\,;\ \epsilon_{rdf}\left(g_{MD_N}(r)\ , g_{ref}(r)\right) < \epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)\}. \tag{3.18}$$

3.3 Results and Discussions

Once the DAE network is trained and an optimal architecture is found, we examine the performance of DAE by considering two different cases. First, we investigate the performance of DAE on monoatomic systems described using the Lennard-Jones pair potential. Then, we investigate its generalizability to various simple liquids.

We present a comparison of DAE with traditional approaches such as the principal component analysis and independent component analysis in Supporting Information, B.3. The Supporting Information also provides details on why the principal component analysis and independent component analysis are not very accurate for denoising of RDFs. The assumptions commonly used in these methods, specifically independent and identically distributed Gaussian noise, are assessed for denoising using the departure test from normality[96] and mutual information[97,98]. The supporting Information B.4 provides information on how to combine principal component analysis with DAE to decrease the number of nodes and size of the DAE network.

3.3.1 Lennard-Jones Pair Potential

We feed single snapshot RDFs of three randomly selected Lennard-Jones systems from training and validation datasets into the DAE network (potential parameters and thermodynamic states are provided in the Supporting Information Section B.2). The predicted average RDFs are shown in Figure 3.2, which indicate a good match between DAE predicted and temporally averaged RDFs. The grey region in Figure

3.2 shows the fluctuations ($\epsilon_{t_j}(r;\rho,T)$) of single snapshot RDFs around the temporally averaged RDF of MD. The insets in Figure 2 show the error distribution of single snapshot RDFs and DAE prediction, *i.e.*, $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ and $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$. The error distribution of DAE predicted RDF is narrower compared with the error distribution of single snapshot RDFs of MD. The distribution of $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$ has a smaller mean compared with the mean of $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ distribution, indicating that the error in the DAE predicted RDF is fairly independent of the input (single snapshot RDF) and the DAE network predicts the temporally averaged RDF with good accuracy.

To assess the efficiency of DAE, we estimate the number of single snapshot RDFs required so that the average RDF estimated from the single snapshot RDFs is of the same accuracy as the DAE predicted RDF, *i.e.*, $\epsilon_{rdf}\left(g_{MD_{N_T}}(r), g_{ref}(r)\right) \leq \epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$, where N_T is the sampling gain defined in Eq. 3.17. Figure 3.3 shows the error in RDF (computed using Eq. (3.9)) as a function of the number of snapshots. RDF error from DAE is independent of the number of snapshots, i.e., once the DAE is trained, it takes a noisy RDF as input and predicts the RDF which is close to temporally averaged RDF. As the number of snapshots increase, the averaging gets better in MD and the RDF error decreases. Figure 3 also indicates that the number of snapshots required to reach the accuracy of DAE prediction is at least 100 and in some cases even with 800 snapshots the DAE prediction has a smaller error.

To further assess the DAE performance, we performed MD simulations on Argon ($C_{12} = 9.70 \times 10^{-6}\ \left[\frac{kJ}{mol.nm^{12}}\right], C_6 = 6.22 \times 10^{-3}\ [\frac{kJ}{mol.nm^6}]$[99]) at various thermodynamic states. The single snapshot (used as input to DAE), DAE predicted, and temporally averaged RDFs are shown in Figure 3.4. The error between DAE predicted and temporally averaged RDF (Eq. 3.9) is an order of magnitude smaller compared to the error between single snapshot and temporally averaged RDFs. Table 2 shows the relative pressure and isothermal compressibility obtained from MD (averaged over 300 snapshots) and DAE (same 300 snapshots are used as inputs and the output is averaged). The pressure and isothermal

compressibility are normalized with their corresponding long-time averaged MD data. The mean value of pressure and isothermal compressibility is close to 1. However, the standard deviation in isothermal compressibility from DAE is an order of magnitude smaller compared to that from MD. These results indicate that DAE is a good method to calculate the pressure and isothermal compressibility in addition to RDF. Transferability of the DAE network to unforeseen thermodynamic states is also investigated and the results are shown in appendix B.1.

3.3.2 Simple Liquids

In order to assess the generalizability of the DAE network, we investigate its performance by considering various simple liquids, which are known to obey similar physics with regards to their structure and dynamics (potential parameters and thermodynamic states are provided in the Supporting Information Section B.2). We feed the single snapshot RDFs obtained from exponential, Yukawa, and two different IPLs into the DAE network. The DAE predicted and temporally averaged RDFs, shown in Figure 3.5 are in good agreement. The distributions of $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ and $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$ are also similar to the Lennard-Jones systems with the peak of $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ located further apart from zero compared to the DAE $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$. Furthermore, similar to the Lennard-Jones system, the DAE error has a narrower distribution compared with the error distribution from MD single snapshot RDFs. These results indicate the generalizability of the DAE network to other monoatomic systems, especially those belonging to the class of simple liquids. The average error between DAE predicted and longtime MD RDF is 0.001, 0.002, 0.002, and 0.002 for exponential, 10-4 IPL, 14-8 IPL, and Yukawa pair potential, respectively, Similarly, the average error between single snapshots and longtime MD RDF is 0.037, 0.037, 0.042, and 0.033 for exponential, 10-4 IPL, 14-8 IPL, and Yukawa pair potential, respectively. The results again indicate that the error in DAE predicted RDFs are one order of magnitude smaller compared to the error in single snapshot RDFs.

3.4 Conclusions

A deep denoising autoencoder network is trained to obtain structural and thermodynamic properties of Lennard-Jones liquids at various thermodynamic states using a single snapshot RDF as input. The algorithm is successful not only in predicting the RDF of a Lennard-Jones pair potential, but also it is generalizable to other simple liquid pair potentials such as exponential, Yukawa, and inverse-power-law potentials. For the simple liquids considered in this work, the radial distribution functions (RDFs) predicted by the denoising autoencoder network have an order of magnitude lower RDF error compared to the errors from using single snapshot RDFs. In terms of computational efficiency, the number of snapshots required from MD simulation to obtain the accuracy of DAE predicted RDF is at least hundred snapshots, making the network highly efficient. The pressure and isothermal compressibility from DAE based RDFs are also comparable with those obtained from longtime MD simulation.

3.5 Figures and Table

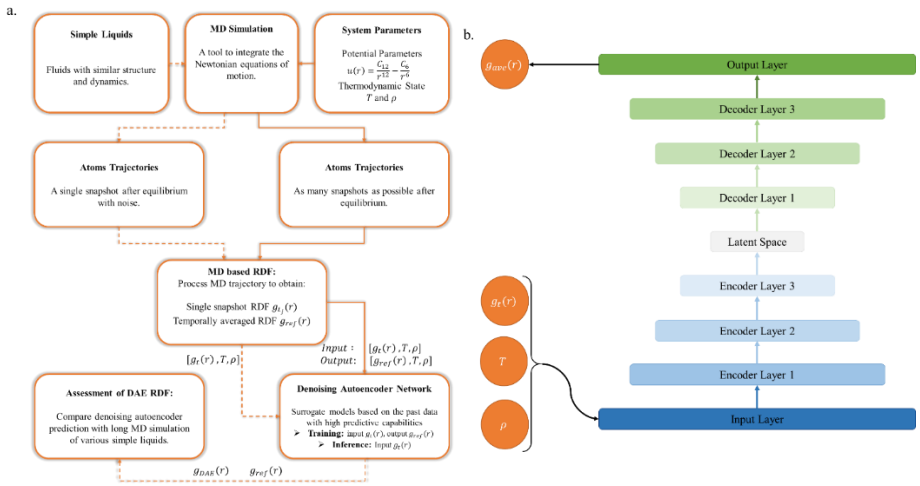


Figure 3.1. a. Schematic representation of workflow of current study. Training (shown with solid lines) is done for the Lennard-Jones liquids with various potential parameters and thermodynamic states and generalizability (shown with dashed lines) is assessed for various simple liquids. b. Schematic representation of the denoising autoencoder (DAE) network (Figure 3.1 (cont.)).

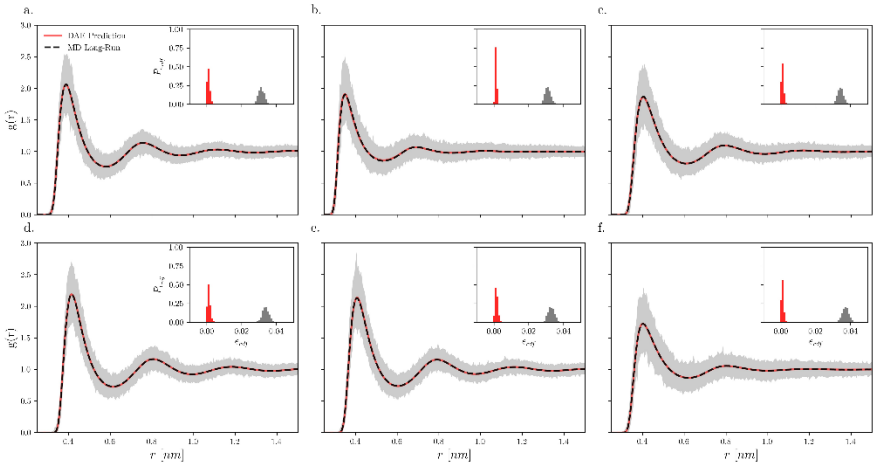


Figure 3.2. Assessment of the performance of DAE for prediction of temporally averaged RDF based on a single snapshot RDF for six randomly selected Lennard-Jones systems (please see Table B.2). a-c. Training dataset d-f. Validation dataset. Solid red lines show the DAE prediction and dashed black lines show temporally averaged MD. The grey region shows fluctuations of single snapshot RDFs around the temporally averaged RDF. The insets show the distribution of $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$ from various single snapshots of MD simulation with red boxes and $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ of single snapshots RDFs with black boxes.

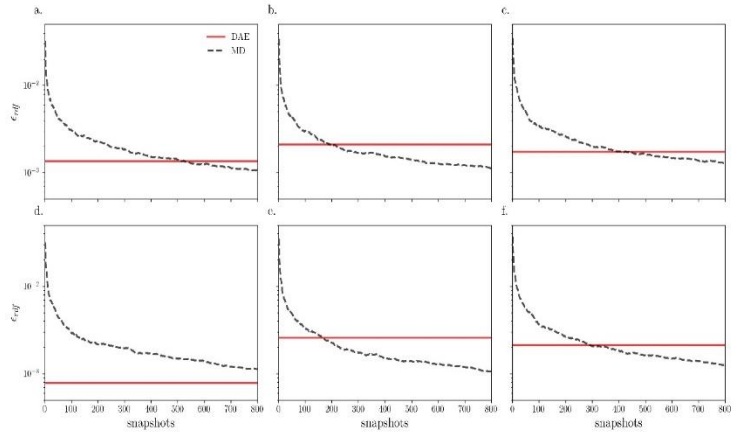


Figure 3.3. Error in RDF as a function of the number of snapshots. Error in RDF from DAE is independent of the number of snapshots. RDF error from MD decreases as the number of snapshots increase. a-c. Training dataset d-f. Validation dataset.

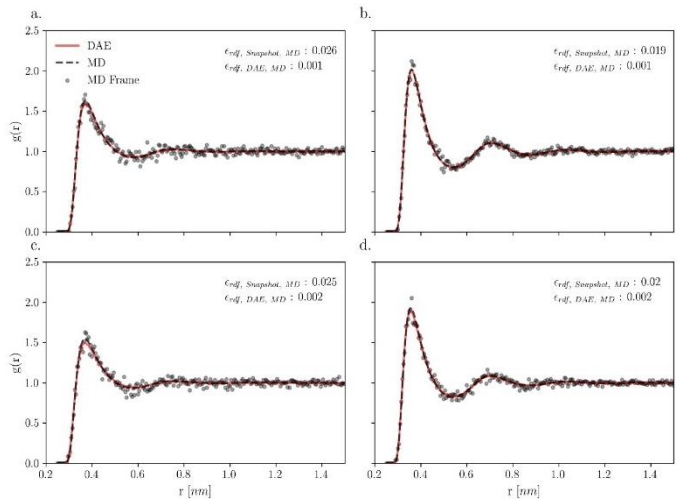


Figure 3.4. Comparison between DAE predicted (shown as solid red line) and temporally averaged (shown as dashed black line) RDFs for Argon Lennard-Jones system at various thermodynamic states. Input RDF (a snapshot from MD) to DAE is shown using grey circles. a. $T = 300\,K, \rho = 10\,nm^{-3}$ b. $T = 300\,K, \rho = 18\,nm^{-3}$ c. $T = 390\,K, \rho = 10\,nm^{-3}$ d. $T = 390\,K, \rho = 18\,nm^{-3}$.

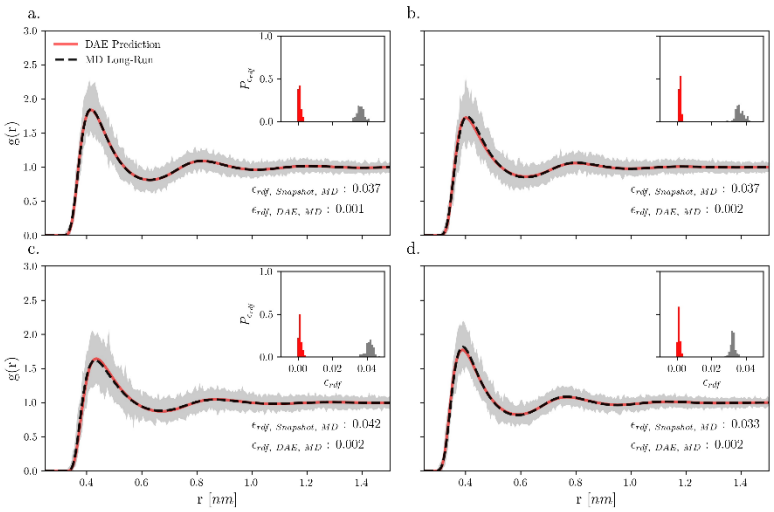


Figure 3.5. Assessment of the generalizability of DAE for prediction of temporally averaged RDF of various simple liquids. a. Exponential pair potential b. IPL pair potential $(p, q) = (10,4)$ c. IPL pair potential $(p, q) = (14,8)$ d. Yukawa pair potential (please see Table S3 of the Supplementary Information Section S.2 for the details of MD simulations). Solid red lines show the DAE predicted RDF, and dashed black lines show the temporally averaged RDF of MD simulation. Grey regions show the fluctuations in RDFs of MD snapshots around the mean (temporally averaged) RDF. The insets show the distribution of $\epsilon_{rdf}\left(g_{DAE}(r), g_{ref}(r)\right)$ from various single snapshots with red boxes and $\epsilon_{rdf}\left(g_{MD_{t_j}}(r), g_{ref}(r)\right)$ of single snapshots with black boxes.

Table 3.1. Mean and standard deviation of pressure and isothermal compressibility from MD (averaged over 300 snapshots) and DAE (same 300 snapshots were used as inputs and the outputs are analyzed). Both pressure and isothermal compressibility are normalized with their corresponding values from long-time MD simulation average.

System	Thermodynamic State		P/P_{MD}		$\mathcal{X}_T/\mathcal{X}_{T,MD}$	
	$\rho\ [nm^{-3}]$	$T\ [K]$	MD_{Frame}	DAE	MD_{Frame}	DAE
1	10.0	300	1.00 ± 0.045	1.00 ± 0.049	1.00 ± 0.0155	1.00 ± 0.0031
2	18.0	300	1.01 ± 0.028	1.00 ± 0.028	1.00 ± 0.0052	1.00 ± 0.0003
3	10.0	390	1.00 ± 0.035	1.00 ± 0.034	1.00 ± 0.0134	1.00 ± 0.0024
4	18.0	390	1.01 ± 0.025	0.99 ± 0.027	1.00 ± 0.0058	1.00 ± 0.0003

CHAPTER 4: Quasi-Universality Through Deep and Statistical Learning

4.1 Introduction

Liquids are a condensed and non-crystallized phase of matter where their properties are typically governed by many-body effects. The inherent complexities associated with many-body systems, particularly liquids, hinder the development of simplified analytical and theoretical approaches, which, in turn, makes it difficult to fundamentally understand the physics of liquids and their properties. On the other hand, atomistic simulations, *e.g.*, molecular dynamics (MD) simulations, have played a crucial role in providing fundamental insights into the structure, dynamics, thermodynamics, and transport of liquids.[100] Computational studies of liquids, for example, have led to the identification of quasi-universal (QU) properties [10,101] of simple liquids (SLs), which refer to the class of liquids having approximately the same structural (*e.g.*, radial distribution function (RDF)) and dynamical properties.[10,64,102–107] The term QU is used instead of universality to delineate the fact that systems with distinct pair potentials have different properties (*e.g.*, thermodynamic properties), however, they exhibit similar behavior concerning other properties (*e.g.*, structural and dynamical properties).[16] Formally, SLs are defined either as liquids whose first coordinate shell determines the structure and dynamics, *i.e.*, reducing the interatomic potential cut-off to the first coordinate shell does not affect the structure and dynamics of liquids or liquids exhibiting a strong energy-virial correlation. Both definitions are used to understand similarities in the underlying physics of SLs with regard to their structure and dynamics.[10,108,109] In particular, understanding the quasi-universality of SLs can be essential to understand the fundamentals of liquid physics as structural quantities, such as RDF, can be used to determine the thermodynamic properties such as pressure, energy, and compressibility, besides quantifying the local structure of atomistic systems.[110]

The structural quasi-universality of SLs only holds in a specific part of the phase diagram of various pair potentials such as exponential (EXP), Yukawa, and inverse-power-law (IPL). Among these potentials, the

standard 12-6 Lennard-Jones (LJ) belonging to the IPL pair potential family and the exponential (EXP) pair potential have played an important role in the computational and theoretical analysis of liquids.[10,111] The LJ potential[93] is composed of repulsive and attractive parts and is widely used in the computational analysis of SLs and various other physicochemical and biological systems.[112–114] On the other hand, EXP potential has been widely used for the theoretical understanding of SLs and recent studies have suggested replacing the hard-sphere model[115,116] with the EXP model as the building block of liquid physics.[10] In general, theoretical models indicate the dominant role of repulsive forces in the long and intermediate wave vectors of structure factor and attraction in the small wave vectors with dependency on the thermodynamic state.[117,118] Even though MD simulations and theoretical models of various SLs have provided insight into the SLs physics and quasi-universality, many aspects of SLs remain to be understood; for example, a method for predicting force distribution without MD simulation.[108,117,119,120]

In this study, we approach the physics and quasi-university of SLs using two perspectives: using statistical methods in the first part and deep learning methods in the second part. In the first part, we develop a simple statistical model to predict the probability distribution function (PDF) of a net force, where developing an analytical solution is intractable, even for SLs, and the problem itself is less explored.[121–123] Our model shows the non-uniqueness of pair potential forms with similar net force PDF among SLs and describes the individual pair force cancellation. Furthermore, it provides a route to parameterize pair potential forms with a similar PDF of the net force. Therefore, our method has applications for coarse-graining and multiscale studies, where both RDF and PDF of forces are available from simulation or experiment.

Before the discussion on the second part of our work, we describe details of the statistical method. We start from the definition of net force and formulate it as a statistical problem. In the statistical formulation, the PDF of the net force is obtained by sampling the PDF of radial distance (which is directly related to RDF, $p(r) = \begin{cases} \frac{g(r)r^2}{\int_0^{r_{cf}} g(r)r^2 dr} & \text{if } r < r_{cf} \\ 0 & \text{if } r \geq r_{cf} \end{cases}$) and the probability mass function (PMF) of the

neighbor list size (number of atoms within the cutoff distance from a tagged particle, $\mathcal{N}(i) = \{j: \forall\, j\ if\ r_{ij} < r_{cf}\}$). The pair potential plays the role of transformation from the PDF of radial distance and PMF of neighbor list size to the PDF of the net force. Next, we show that the transformation from a given variable to another variable is not unique consistent with the quasi-universality. Similarly, the pair potential form should not be unique for a given net force PDF. Furthermore, we show that our statistical model can be used to optimize the parameters of various pair potentials to find SL systems with similar PDF of a net force, which, in turn, leads to having similar RDF with various pair potentials. We also note that like previous approaches,[118,121,124] we have few approximations, e.g., mean-field and independence between random variables of our statistical model. We verify our model by examining its performance for different pair potentials at different thermodynamic states, and we observe good agreement between its results and MD ones. Within our model, it is easy to use the information theory-based methods for the inverse problem, where the PDF of force is known and not potential parameters i.e., coarse-graining with the combination of structure and force.

In the second part, we use a deep learning-based method to investigate thermodynamic states other than the reference simulation, which is a bottleneck for our statistical method. To do so, we connect DeepILST, developed in our prior work[80], to the study of SLs. We start with finding the structurally-equivalent LJ liquids, defined as LJ liquids having the same reduced RDF ($\tilde{g} = g(\xi)$, where $\xi = r\rho^{1/3}$). In other words, we find isomorphs of LJ liquids, curves with invariant structures. After obtaining various structurally-equivalent LJ liquids through the DeepILST, we find constant order-parameter systems in the non-dimensional temperature and density space. Then, we assess generalizability of the DeepILST in parameterizing structurally-equivalent LJ potential for various pair potential forms. The formalization of deep learning model is based on the Henderson theorem, which states the relation between a given RDF and the pair potential is unique up to a constant[16,125] (note even though recent studies show that the small deviation in the structure can lead to a completely different pair potential,[12,126] however, consistent with any structure-based coarse-graining methods[14,24,31,127], our approach converges to the optimal LJ potential

parameters after learning over a large amount of data of LJ potential, especially considering the restriction of the pair potential form in our study). Consistent results between DeepILST and theoretical models show promising applications and potentials of deep learning models in the physics of SLs similar to its success in the other areas of physics of big data[34,45,79,82,128–132]. The workflow of both parts, namely statistical- and deep-learning parts, is shown in Figure 4.1 along with the objectives of each method.

The rest of the paper is organized as follows. First, we describe the details of MD simulations and statistical and deep learning-based methods. Then, we present the results to validate the statistical model, followed by the results to assess consistency and applications of the DeepILST with the structural quasi-universality of SLs, as well as the ability of DeepILST in the study of SLs. Finally, we summarize the findings of this study.

4.2 Methods

4.2.1 Molecular Dynamics Simulation

The training data for DeepILST is obtained through MD simulation of various 12-6 LJ pair potentials at various thermodynamic states with a total simulation time of 52 μs. The LJ potential form can be expressed as,

$$u_{LJ}(r) = \frac{C_{12}}{r^{12}} - \frac{C_6}{r^6} \tag{4.1}$$

where $C_{12}(= 4\epsilon\sigma^{12})$ and $C_6(= 4\epsilon\sigma^6)$ are repulsive and attractive parameters of the LJ pair potential, respectively. In particular, we study the following QU pair potentials,

$$u_{IPL,p,q}(r) = \frac{C_p}{r^p} - \frac{C_q}{r^q} \tag{4.2}$$

$$u_{EXP}(r) = A\exp(-k_D r) \tag{4.3}$$

$$u_{Yukawa}(r) = A\frac{\exp(-k_D r)}{k_D r} \tag{4.4}$$

where $C_p(=4\epsilon\sigma^p)$ and $C_q(=4\epsilon\sigma^q)$ are the repulsive and attractive parameters of the IPL pair potential with an exponent of p and q, respectively. A and k_D are the strength and inverse screening length parameters of EXP and Yukawa pair potentials. All the MD simulations are performed using GROMACS [53] with a timestep of 1 fs in the NVT ensemble with the temperature being controlled by a time constant of 0.2 ps using the Nosé-Hoover thermostat (thermodynamic states and potential parameters are provided in the Supporting Information Section C.1). Each system is simulated for 2 ns after energy minimization. Then, MD trajectories are processed to calculate RDF from the last 1.8 ns.

4.2.2 Statistical Model

A statistical model is employed to obtain the PDF of instantaneous net forces, which governs the motion of atoms. The net force on particle i can be expressed as the sum of the pair forces exerted by all the neighboring particles (denoted by $\mathcal{N}(i) = \{j: \forall\, j\ if\ r_{ij} < r_{cf}\}$, where r_{cf} is the cutoff distance and r_{ij} is the radial distance between particles i and j). Mathematically, the net force can be expressed as,

$$\boldsymbol{F}_i = \sum_{j\,\in\,\mathcal{N}(i)} -\frac{\partial u}{\partial \boldsymbol{r}}\Big|_{\boldsymbol{r}=r_{ij}.\boldsymbol{e}_{ij}} = \sum_{j\,\in\,\mathcal{N}(i)} -u'(r_{ij})\,.\,\boldsymbol{e}_{ij} \tag{4.5}$$

where $\boldsymbol{F}_i$ is the net force on particle i. u' is the first-order derivative of pair potential, *i.e.*, $u'(r_{ij})$ is the value of individual pair force. r_{ij} and $\boldsymbol{e}_{ij}$ $\left(=\frac{\boldsymbol{r}_{ij}}{r_{ij}}\right)$ denote the distance and unit vector of the radial distance vector between particles i and j, respectively. In the statistical model, r_{ij} is treated as a random variable with its PDF obtained directly from MD simulation. Similarly, the size of the neighbor list, *i.e.*, the summation index ($\mathcal{N}(i)$) in Eq. 4.5, is considered as an independent random variable with its PMF calculated using MD simulation.

The isotropic nature of a pair potential leads to a uniform distribution over components of the unit vector of radial distance, *i.e.*, the components of the force are identical. Therefore, the distribution of the unit vector can be written as,

$$\boldsymbol{e}_{ij} = \sqrt{1-v^2}\cos\theta\,\boldsymbol{e}_x + \sqrt{1-v^2}\sin\theta\,\boldsymbol{e}_y + v\,\boldsymbol{e}_z \tag{4.6}$$

where v and θ are independent and uniformly distributed random variables representing the orientation of unit vector ($p(\theta) = \frac{1}{2\pi}\ \forall\ \theta \in [0, 2\pi)$ and $p(v) = \frac{1}{2}\ \forall\ v \in [-1,1]$. Having all the above assumptions set, the PDF of force ($p(F_i) = p\left(\sum_{j\,\in\,\mathcal{N}(i)} u'\left(\left|r_{ij}\right|\right)\right)$ where F_i represents the Cartesian components of $\boldsymbol{F}_i$) can be determined using the law of total probability as expressed below,

$$p(F_i) = \sum_{\forall\ N>0} p(|\mathcal{N}(i)| = N)\, p\Big(\sum_{m=1}^{N} -u'(r_m)\,.\,e_m\Big) \tag{4.7}$$

where the inner most sum is N times convolution of PDF of $u'(r_m)e_m$. e_m is any component of the unit vector, which has a uniform PDF according to Eq. 4.6. $p(-u'(r_m)\boldsymbol{e}_m)$ is obtained through statistical sampling based on the inverse transform sampling of cumulative distribution function (CDF) of r_m (denoted by $cdf(r_m) = \int_{-\infty}^{r_m} p(r)dr$, where $p(r)$ denotes the PDF of r and is related to RDF, $p(r) = \begin{cases} \frac{g(r)r^2}{\int_0^{r_{cf}} g(r)r^2 dr} & if\ r < r_{cf} \\ 0 & else \end{cases}$).[133] Therefore, PDF of F_i can be obtained based on the PMF of $\mathcal{N}(i)$ and the PDF of r_m.

Before discussing the implications of Eq. 4.7 on the QU properties of SLs, we examine its consistency with the equilibrium condition, *i.e.,* the expected value of force equals zero. The expected value of force can be expressed as follows based on Wald's identity,[134]

$$\mathbb{E}[F_i] = \mathbb{E}[\mathcal{N}(i)]\mathbb{E}[-u'\left(r_{ij}\right)\,e_{ij}] = \mathbb{E}[\mathcal{N}(i)]\ \mathbb{E}\left[-u'\left(r_{ij}\right)\right]\mathbb{E}[e_{ij}] \tag{4.8}$$

For the right-hand side of Eq. 8 to be zero, either $\mathbb{E}\left[-u'(r_{ij})\right]$ or $\mathbb{E}\left[\boldsymbol{e}_{ij}\right]$ should be zero (note that the system with $\mathbb{E}[\mathcal{N}(i)]$ of zero has no physical significance). The first case leads to an exceptional condition on the pair potential. On the other hand, $\mathbb{E}\left[e_{ij}\right]$ equal to zero is consistent with the isotropic pair potential with uniformly distributed force components.

An important implication of Eq. 4.7 is that it provides statistical evidence for the existence of systems with similar structural properties but different pair potential forms. Various pair potentials exist such that the PDF of net force remains statistically identical with another system, and, therefore, their structures resemble each other within the Newtonian equations of motion.

Having the statistical model based on the PDF of radial distance and PMF of the number of neighboring particles, we proceed to validate its accuracy in reproducing the PDF of net force from MD simulation, as well as finding optimal potential parameters of another QU pair potential at the same thermodynamic state. The validation step is done through a comparison between the PDF of net force from the statistical model based on Eq. 4.7 with the reference MD simulation. To find optimal parameters of another pair potential, we minimize the KL divergence between the PDFs of net force from MD simulation and of a statistical model with unknown potential parameters. The objective function for finding optimal parameters is expressed as follows,

$$\min_{\boldsymbol{S}} D_{KL}\left(p_{MD}(F_i); p_{Stat}(F_i; \boldsymbol{S})\right) \tag{4.9}$$

where $\boldsymbol{S}$ denotes the pair potential parameters, *e.g.*, $\boldsymbol{S} = (A, k_D)$ of Yukawa and Exponential pair potentials. $D_{KL}\left(p_{MD}(F_i); p_{Stat}(F_i)\right)$ is the KL divergence expressed as follows,

$$D_{KL}\left(p_{MD}(F_i); p_{Stat}(F_i)\right) = \int_{-\infty}^{\infty} p_{MD}(F_i) \log\left(\frac{p_{MD}(F_i)}{p_{Stat}(F_i)}\right) dF_i \tag{4.10}$$

The above minimization is performed using the particle swarm optimization, a derivative-free algorithm in which multiple points of the objective function are evaluated and influence the evaluation of the

objective function in the next iterations.[135] The total run time of the statistical model calculation for simultaneous development of 32 statistical models is in the interval of [1, 3] min on the computing nodes of Blue Waters with a single node while using the multiprocessing package python (note that the computational cost is far smaller than the direct MD simulation).

4.2.3 Deep Learning

Even though the statistical model developed in this study provides a probabilistic explanation for identical behavior of SLs within the framework of Newtonian mechanics, it is, however, limited by the thermodynamic state of the reference MD simulation. To circumvent the difficulties associated with the thermodynamic states as well as to provide further insight into the SLs, we investigate the consistency and capabilities of DeepILST[80] in the study of SLs.

A concise summary of the DeepILST framework is provided here and we refer the reader to Ref. [80] and Chapter 1 for further information. The feature vector x_i of the DeepILST is composed of the concatenation of RDF, density, and temperature for system i, which can be expressed as,

$$x_i = \left(\boldsymbol{g}_i\,(r), \rho_i^1, \rho_i^2, \dots, \rho_i^p, T_i^1, T_i^2, \dots, T_i^p\right)_m \tag{4.11}$$

where the index i refers to the i-th system in the dataset and x_i represents the input into the deep neural network of DeepILST composed of RDF and thermodynamic states. With the above feature vector and based on the universal approximation theorem,[56] DeepILST approximates the function in the inverse liquid-state theory equation using DNN,

$$(C_{12}, C_6) = \boldsymbol{f}(g(r);\, T, \rho) \tag{4.12}$$

where $\boldsymbol{f}$ is a vector-valued function, which is a nontrivial function of RDF and thermodynamic state variables.

The deep neural network, in this study and in general, is composed of simple units known as perceptron or node, which are stacked in width (referred to as a layer) and in depth (successive layers). Each node

receives an input signal and applies a linear transformation, followed by a nonlinear activation function resulting in an output signal.[38,57] With LJ potential parameters as the output of the DeepILST and RDF and its corresponding thermodynamic state as the input vector, the deep neural network can be expressed as,

$$\left(C_{12,i}, C_{6,i}\right) = \phi_o(\boldsymbol{W}_o\phi_{n_h}(\dots\phi_2(\boldsymbol{W}_2\phi_1(\boldsymbol{W}_1 x_i + \boldsymbol{b}_1) + \boldsymbol{b}_2)) + \boldsymbol{b}_o) \tag{4.13}$$

where ϕ_k is the nonlinear activation function of layer k. Layer k receives a linear transformation of the output of layer $k-1$, given by $(W_k\phi_{k-1}(x) + b_k)$, and applies the activation function ϕ_k to produce the output of layer k given by $\phi_k(x) = \tanh(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k) + \text{sigmoid}(\boldsymbol{W}_k\phi_{k-1}(x) + \boldsymbol{b}_k)$. Each layer k, has weights $\boldsymbol{W}_k$ and bias $\boldsymbol{b}_k$ with dimensions, $\boldsymbol{W}_k \in R^{d_{k-1}\times d_k}$ and $\boldsymbol{b}_k \in R^{d_k}$, where d_k is the number of nodes in layer k. n_h denotes the number of hidden layers. Summation of the tangent hyperbolic and sigmoid nonlinearities ($\tanh x + \text{sigmoid } x$) is used between layers, except for the output layer (ϕ_o) where the sigmoid nonlinearity is used. See Ref. [80] and its supplementary material for further details regarding the training and network selection..

To train the deep neural network, the mean-squared loss function, *i.e.*, the mean-squared error between the ground truth and the predicted parameters is optimized by the backpropagation algorithm. The adaptive moment estimation optimizer [58] is employed to minimize the loss function ($\epsilon_{\mathcal{L}}(\theta, D)$), which can be expressed as,

$$\epsilon_{\mathcal{L}}(\theta, D) = \frac{1}{2|D|}\sum_{i\in D}\sum_{j\in\{C_{12},C_6\}}\left(v_{j,GT}^{(i)} - v_{j,DNN}^{(i)}(\theta, x_i)\right)^2 \tag{4.14}$$

where θ represents the free parameters (weights and biases) of a deep neural network. $v_{j,GT}^{(i)}$ and $v_{j,DNN}^{(i)}(\theta)$ are the ground truth and DeepILST-predicted scaled LJ interaction parameters (with respect to the minimum and maximum in Table 1) of the i^{th} data point in the dataset. $|D|$ denotes the size of the training dataset with the size of about 26000 distinct systems[80]). The selected network has two hidden layers with 48 and 15 nodes with the exponent of thermodynamic states being three, *i.e.*, $p = 3$.

4.2.4 Physics of Data

Once the DeepILST network is trained, we find systems with constant order parameters in the space of non-dimensional temperature and density for 12-6 LJ liquids. The order parameter is given by,

$$\tau = \int_0^{\xi_{cf}} |g(\xi) - 1|\, \xi^2 d\xi \tag{4.15}$$

where ξ is the non-dimensional length-scale ($\xi = r\rho^{1/3}$), and ξ_{cf} is the reduced cutoff distance which is taken to be 3.5. This order parameter, known as the translation order parameter, measures the deviation of RDF from that of an ideal gas.[136–138] Note that all the non-dimensional quantities are given for the 12-6 LJ pair potential. The algorithm 4.I defines the procedure used to identify constant order parameter systems.

Algorithm 4.I. Identify structurally equivalent LJ liquids with similar $g(\xi)$

$\forall\, i \in D$

$$g_i(\xi) = g_i\left(r\rho_i^{\frac{1}{3}};\, \rho_i, T_i\right)$$

$$\tau_i = \int_0^{\xi_{cf}} |g_i(\xi) - 1|\, \xi^2 d\xi$$

$\forall\, \rho \in [\rho_i - 2\Delta\rho : \Delta\rho : \rho_i + 2\Delta\rho]$

$$\tau_i^{\rho} = \tau_i$$

$$g_i^{\rho}(\xi^{\rho}) = g_i\left(r\left(\frac{\rho_i}{\rho}\right)^{\frac{1}{3}}\right)$$

$$C_{12}^{QU,\rho}, C_6^{QU,\rho} = \text{DeepILST}\left(\, g_i^{\rho}(r), \rho, T\,\right)$$

where index i refers to a specific RDF in the dataset (D) with density and temperature of ρ_i and T_i, respectively. Here, we derive an approximate theoretical solution to assess the accuracy of DeepILST prediction. To do so, we make two approximations, namely the high temperature ($k_B T \gg 1$) approximation and approximating the potential of mean force with the pair potential ($u_{pmf}(r) \sim u_{LJ}(r)$).

With these assumptions, we start from the relationship between RDF and the potential of mean force given by,

$$g(r) = \exp(-\frac{u_{pmf}(r)}{k_B T}) \tag{4.16}$$

Approximating the potential of mean force with the pair potential, the above equation can be written as,

$$g(r) \sim \exp(-\frac{u_{LJ}(r)}{k_B T}) \tag{4.17}$$

Applying the high-temperature approximation, the first-order Taylor expansion of the above equation can be expressed as,

$$g(r) \sim 1 - \frac{u_{LJ}(r)}{k_B T} \tag{4.18}$$

After a few mathematical steps (see section C.2 of the Supporting Information for details), we arrive at the equation,

$$\tau = k \ \frac{\rho^*}{T^*} \tag{4.19}$$

where $k = \int \left| 4\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| \left(\frac{r}{\sigma}\right)^2 \frac{dr}{\sigma}$, $\rho^* (= \rho\sigma^3)$ and $T^* \left(= \frac{k_B T}{\epsilon}\right)$ are non-dimensional density and temperature, respectively. This relationship, even though it is simple, is physically consistent in the limit of solid and gas phases. For example, assuming a given material with specific potential parameters at a specific temperature (T, ϵ, and σ are given), the value of ρ^*/τ is constant. The solid phase has a long-range and heterogeneous RDF with a large order parameter ($\tau \gg 1$), implying that $\rho^* \gg 1$, which is consistent with the phase diagram of LJ liquids. The gas phase at the same condition has a short-range and homogenous RDF with a small order parameter ($\tau \ll 1$), implying that $\rho^* \ll 1$, which again is consistent with the phase diagram of LJ liquids.[139,140]

We also assess the consistency of our results for the isomorphs of LJ liquids with the theoretical solution developed by Ingebrigtsen *et al.*[141] Their theory leads to a simple mathematical expression for the isomorphs of LJ liquids, which show good agreement with simulation results. The expression is given by,

$$\frac{h(\rho^*)}{T^*} = \frac{A\rho^{*4} - B\rho^{*2}}{T^*} = constant. \tag{4.20}$$

To further assess the consistency between DeepILST and the structural quasi-universality of various SLs, we provide RDFs of SLs with different pair potentials as input to DeepILST to parameterize QU-equivalent LJ pair potential. The term QU-equivalent is used instead of structurally-equivalent to emphasize that the reference system does not have a 12-6 LJ pair potential form. Next, we quantify the error in the structure (RDF) of various SLs and their QU-equivalent LJ potentials based on two measures: the first metric is the RDF deviation between LJ and SL models and the second metric is the KL divergence.[14,33] The RDF deviation between reference and target models is calculated using the following expression,

$$\epsilon_{rdf} = \frac{\int_0^{r_{cf}} |g_{tgt}(r) - g_{ref}(r)|\, r^2 dr}{\int_0^{r_{cf}} g_{ref}(r) r^2 dr} \tag{4.21}$$

where $g_{tgt}(r)$ and $g_{ref}(r)$ are the RDFs of target and reference models, respectively. r_{cf} is the cut-off distance, which is set to 1.2 nm. The KL error is also estimated using the following expression,

$$\epsilon_{KL} = \beta\langle U_{LJ} - U_{SL}\rangle_{SL} - \beta\langle F_{LJ} - F_{SL}\rangle_{SL} + \langle S_{map}\rangle_{SL} \tag{4.22}$$

where $U(=\sum_{i<j} u(r_{ij}))$ and F are respectively the potential energy and free energy of the system in the SL and LJ systems, and $\langle S_{map}\rangle_{SL}$ is the mapping entropy. Appendix A contains mathematical details on the KL divergence calculation, as well as its convexity and uniqueness of a global minimum.[60,61,142]

4.3 Results and Discussions

4.3.1 Statistical Model Assessment

We first assess the statistical model by generating the PDF of the net force on the particles. To do so, we use the PDF of radial distance and the PMF of neighbor list size, both of which are obtained from MD simulations of Yukawa potential. The reference Yukawa pair potential is simulated at a density of

$12.232\ nm^{-3}$ using LAMMPS[143] with A and k_D equal to $1.056 \times 10^5 [Kcal/mole]$ and $3.144\ [1/A^{\circ}]$. Figure 4.2 a,b show the PMF of neighbor list at 50 and 300 K, respectively. The RDFs used to obtain the PDF of radial distance at 50 K and 300 K are shown in Figures 4.2c,d, respectively. The log of the net force PDF is shown in Figure 4.2e,f, which indicates that the PDF of net force between MD and statistical model are in good agreement with each other. The KL errors are 0.012 and 0.042 for 50 K and 300 K, respectively, indicating the validity of the statistical model when the same pair potential as the MD simulation is used. We also observe a linear behavior at the tail of log of the PDF of the net force consistent with the Powles and Flower observations,[121] which is attributed to the forces exerted by atoms in the first coordination shell as they have small distance and therefore large forces.[117,122] Note that the linear behavior is not universal and may deviate from linear. For instance, the theoretical model by Rickayzen et al.[122] predicts an exponentially-decaying probability for large forces of a repulsive potential with $\mathrm{u(r)} = \frac{\epsilon}{\mathrm{r^n}}$ $(\mathrm{P(F} \gg 1) \sim \mathrm{F^v} \exp(-\alpha \mathrm{F})$ with v and α being thermodynamic- and potential-dependent quantities). This approximate PDF of force shows an exponential decay (or close to exponential as $\alpha = \mathrm{n}/(\mathrm{n}+1)$ goes toward unity) for large values of the force consistent with our observations and statistical model predictions for various pair potentials. It is also noticeable from Figure 4.2 that as temperature increases it leads to larger fluctuation of various quantities such as neighbor list size and force, *i.e.*, larger standard deviation due to thermal fluctuation.

4.3.2 DeepILST and Quasi-Universality

In the next step, we optimize the parameters of exponential and LJ pair potentials such that each of them reproduces the same PDF of net force as the MD results of the Yukawa pair potential at 50 K and 300 K. Figure 4.3a-b show the log of PDFs of net force from MD simulation and that of various pair potentials from the statistical model, which show a very good agreement. At 50 K, the errors using the KL measure are 0.00131 and 0.00980 for the EXP and LJ pair potentials, respectively. The errors at 300 K using the KL measure are 0.001884 and 0.00564 for the EXP and LJ pair potentials, respectively. Once the

minimization is done, we perform MD simulations of both the EXP and LJ pair potentials. The comparison between the RDFs of Yukawa and exponential pair potentials is shown in Figure 4.3c-d. The RDF errors (ϵ_{rdf}) are 0.00526 and 0.00876 at 50 K for the EXP and LJ pair potentials, respectively. Similarly, the RDF errors are 0.00213 and 0.00282 at 300 K for the EXP and LJ pair potentials, respectively. Overall, the results establish the accuracy of the statistical model.

1. Lennard-Jones Liquids

We randomly feed six distinct RDFs into DeepILST and apply Algorithm 4.I. The algorithm produces a set of structurally-equivalent LJ potentials at various thermodynamic states for each RDF. Using the set of structurally-equivalent LJ potentials, we perform MD simulations at the corresponding thermodynamic state. As expected, the DeepILST predicted LJ potentials produce almost the same reduced RDF ($g(\xi)$) for each set of structurally-equivalent LJ potentials as shown in Figure 4.4. The accuracy of DeepILST is clear as the reduced RDFs are almost indistinguishable (see Supporting Information Section C.3 for further data).

Now that we have shown that the reduced RDF is similar for each set of structurally-equivalent LJ potential, we check the qualitative accuracy of our results with Eq. 4.13. Figure 4.5a shows the non-dimensional temperature and density of structurally-equivalent LJ potentials with various order parameters using circles with the area of the circle being proportional to the order parameter. In other words, points with a similar area are selected points from a curve with constant reduced RDF. The trend is increasing between non-dimensional temperature and density of given order parameters consistent with Eq. 4.13. Furthermore, at a given non-dimensional density the system with larger non-dimensional temperature has smaller order parameters (see the inset in Figure 4.5a). To compare the DeepILST-predicted isomorphs with the theoretical solution in Eq. 4.20, we fit all data into Eq. 4.20. In Figure 4.5b, we compare the DeepILST results for isomorphs with the theoretical solution, which shows a good agreement as isomorphic LJ liquids have constant values. Additionally, the constant values in the right-

hand side of Eq. 4.20 show an increasing trend with the increase of order parameters among different isomorphs.

2. Quasi-Universal Liquids

We now establish a connection between DeepILST based coarse-graining and quasi-universality of SLs by determining QU-equivalent LJ pair potentials for various SL pair potentials. Figure 4.6 shows the RDFs of reference QU and QU-equivalent LJ pair potentials for four different potentials studied in this work (further case studies are provided in Section C.3 of the Supporting Information; note that we only show results for the LJ pair potentials for the range shown in Table 2.1 as our prior study[80] indicates decreased accuracy for regions which are not sampled during the DeepILST training). The pair potentials shown in Figure 4.6 indicate that even though the RDFs of the four reference QU pair potentials are similar, the LJ pair potentials vary significantly in each case. The change is especially significant in the region where the potential changes from repulsion to attraction (shown in the inset of Figure 4.6).

Figure 6a-b shows that the repulsive part of both the EXP and Yukawa pair potentials starts at a larger radial distance (beyond the cutoff distance, which is set to 1.2 nm) compared to their QU-equivalent LJ pair potential forms (at approximately 0.6 nm). This can be attributed to the presence of an attractive part in the tail of the LJ pair potential, the effects of which are canceled out with the repulsive part starting at a smaller radial distance compared to the EXP and Yukawa potentials. Figure 4.6c-d shows that the QU-equivalent LJ pair potentials change based on the exponent of the repulsion and attraction parts of the reference IPL potential. For IPL with exponents larger than standard LJ potential (12 and 6), the QU-equivalent LJ pair potential becomes more negative compared to the IPL pair potential implying more attraction. For exponents smaller than the standard 12-6 exponents, the QU-equivalent LJ pair potential becomes less negative compared with the IPL pair potential implying less attraction.

As shown in Figure 4.6, the RDFs from reference QU pair potentials and QU-equivalent LJ potentials are nearly indistinguishable. To ensure accuracy, we assess the error between the RDFs using two metrics, establishing a connection between coarse-graining and quasi-universality of SLs as well as showing

optimality of DeepILST predicted QU-equivalent LJ potentials. The two metrics are the RDF deviation (ϵ_{rdf}) and KL error (ϵ_{KL}). Figure 4.7a-d shows ϵ_{rdf} for a specific range (see Table 2.1) of the LJ parametric space considered in this work. DNN prediction of the potential parameters and the minimum value of ϵ_{rdf} are also shown in Figure 4.7. We also calculate ϵ_{KL}, which provides a more rigorous measure (shown in Figure 4.7e-h). Again, the optimal point of the KL error and DNN potential parameters are close to each other. Furthermore, the DNN-predicted potential parameters are enclosed in the region smaller than 0.1 percentile of both error metrics in the investigated parameter space, indicating optimality of DNN potential parameters as well as DeepILST consistency with the structural quasi-universality of the simple liquids. The results, therefore, indicate that DL-based models, particularly DeepILST, are accurate to capture physical laws for systems with universal and QU properties, which, in turn, opens opportunities to transfer knowledge acquired from one class to other classes of problems with similar underlying physics.

4.4 Conclusions

In summary, we propose a statistical model to predict the probability distribution of the net force of simple liquids and explain the pair force cancellation, which explains the non-uniqueness of potential form for a given net force distribution among simple liquids. In addition to computational efficiency, the proposed statistical model is successful in obtaining parameters of various pair potential forms with similar net force distribution among them, which within the Newtonian equations of motion translates into a similar structure. We have verified the structural similarity among simple liquids with similar net force distribution by molecular dynamics simulation. The physics and quasi-universality of simple liquids are also studied through deep learning by finding structurally-equivalent Lennard-Jones liquids with similar reduced RDFs, *i.e.,* isomorphs. Structurally-equivalent Lennard-Jones liquids identify systems with constant order parameters in the space of non-dimensional temperature and density of Lennard-Jones liquids consistent with the approximate theoretical solution derived in the current study

and other theoretical models. Furthermore, DeepILST successfully reproduces the structure of various non-Lennard-Jones simple liquids by feeding their RDFs, shown for four different quasi-universal pair potentials including exponential, Yukawa, and two inverse-power-law pair potentials. The error between the RDFs and Kullback-Leibler errors are quantified using two different coarse-graining metrics, which, in turn, indicates the consistency between the DeepILST and the structural quasi-universality of simple liquids. Considering various investigations performed in this study, we show the successful employment of statistical and deep learning approaches and coarse-graining methods in the physics of simple liquids.

4.5 Figures

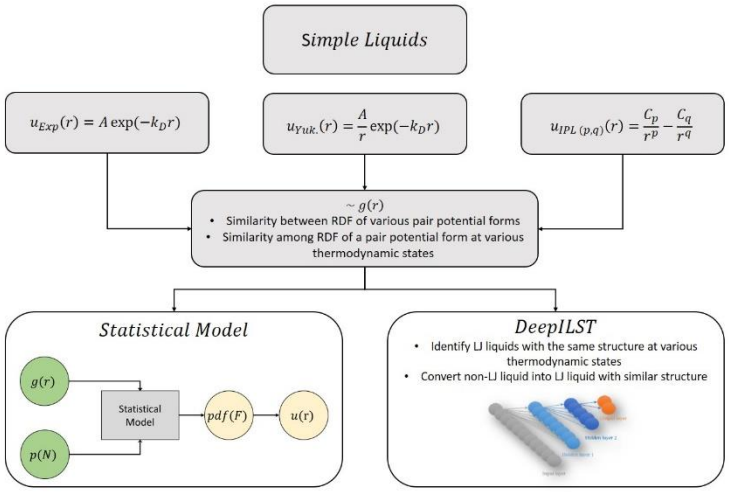


Figure 4.1. Simple liquids are liquids described by different pair potential forms with a similar structure, usually quantified in terms of the radial distribution function (RDF) (shown as grey boxes). Even though the structural similarity is known among these potentials, the mechanism through which pair force cancellation occurs is not clear. A statistical model is developed based on the radial distance and the number of neighboring particle distributions to predict the distribution of net force on the particle. The statistical model provides insight into the force cancellation mechanism and non-uniqueness of the pair potential form. It also provides a route to convert different pair potential forms into each other at a given thermodynamic state, with application in coarse-graining of system with both force and RDF. To solve the problem with thermodynamic state-dependency of the statistical model, we use DeepILST to identify structurally equivalent LJ liquids over various thermodynamic states as well as converting various simple liquids into LJ liquids. In the statistical model box, green circles show the known quantities and yellow ones show the model's predictions.

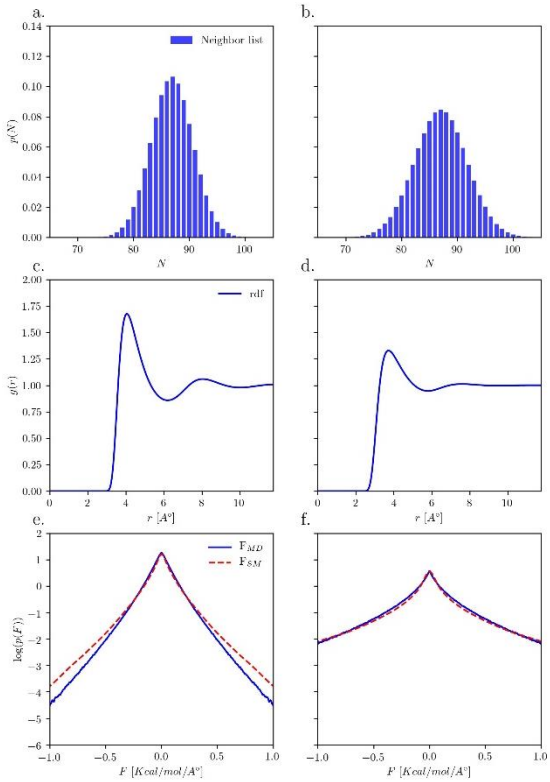


Figure 4.2. Comparison of the net force distribution calculated using the statistical model and the reference MD simulation of the Yukawa pair potential at 50 K and 300 K. **(a) and (b)** probability mass function of neighbor list size within the cut-off distance obtained from MD simulation at 50 K and 300 K, respectively. **(c) and (d)** RDFs used to obtain the probability distribution function of radial distance within the cut-off distance obtained from MD at 50 K and 300 K, respectively. **(e) and (f)** comparison between log of probability distribution function of net force obtained from MD simulation of Yukawa potential and from statistical model at 50 K and 300 K, respectively (SM indicates statistical model in the Figure).

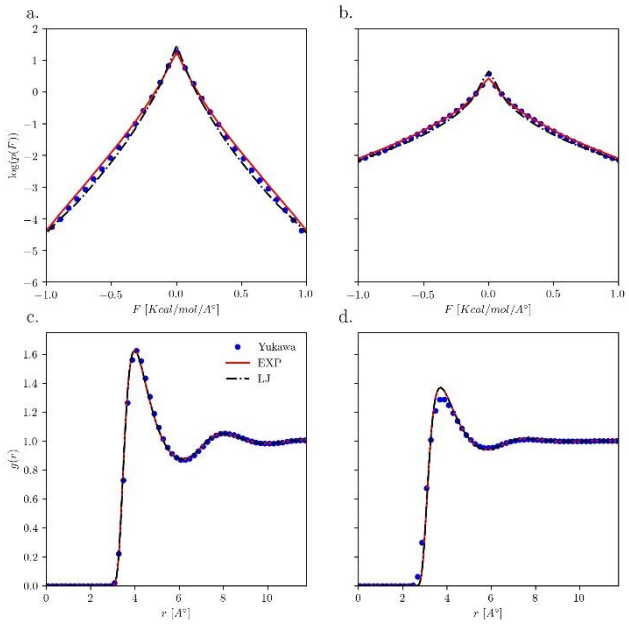


Figure 4.3. Pair force cancellation mechanism described by the statistical model allows to parameterize different SL pair potentials (EXP and LJ pair potentials) such that the net force distribution is similar to the reference MD simulation (Yukawa pair potential at 50 K and 300 K). **(a) and (b)** comparison between log of probability distribution function of net force obtained from MD simulation of Yukawa potential and from the statistical model with EXP and LJ pair potential at 50 K and 300 K, respectively (SM indicates the statistical model in the Figure, blue circles show MD simulation results with Yukawa pair potential, and red solid and black dot-dashed lines represent results of statistical or MD simulations of the EXP and LJ pair potentials, respectively). **(c) and (d)** comparison between the radial distribution function obtained from MD simulation of the reference Yukawa pair potential and that of the statistical model-based parameterized EXP and LJ pair potential at 50 K and 300 K, respectively.

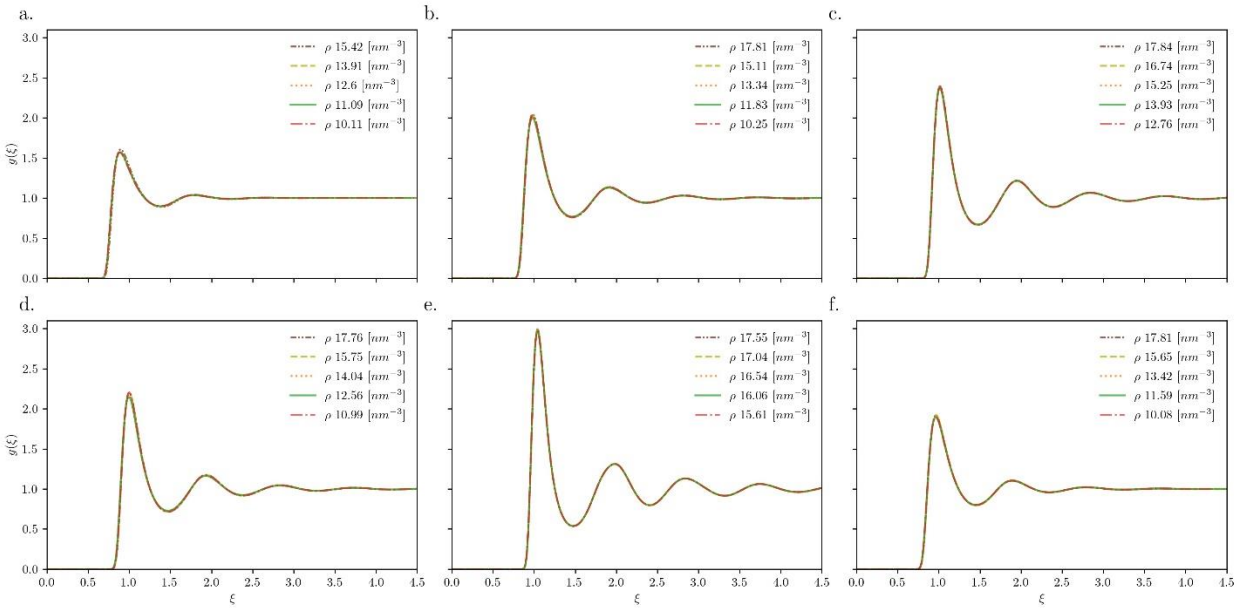


Figure 4.4. Structurally-equivalent LJ liquids obtained using DeepILST based on Algorithm I. The reduced RDFs are indistinguishable between LJ systems at different thermodynamic states. Each plot shows 5 different thermodynamic states with similar reduced RDF and order parameter. **(a)** $\tau = 0.39$, **(b)** $\tau = 1.04$ **(c)** $\tau = 1.73$, **(d)** $\tau = 1.32$, **(e)** $\tau = 3.0$, **(f)** $\tau = 0.82$.

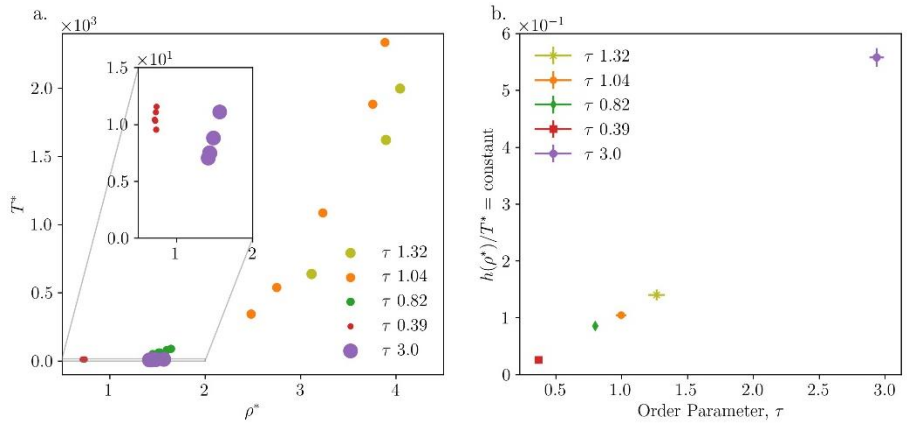


Figure 4.5. **a.** Systems of constant order parameter from Figure 4 in the non-dimensional temperature and density space of 12-6 LJ liquids. The inset zooms into a smaller region of non-dimensional temperature and density. Consistent with the approximate theoretical solution, the non-dimensional temperature and density are linearly related for a given order parameter (reduced RDF). Systems with a larger constant order parameter have a smaller non-dimensional temperature at a given non-dimensional density, as seen in the inset. The area of the circle is proportional to the order parameter. **b.** Comparison between the DeepILST predicted isomorph and theory (Eq. 20) is fitted to the DeepILST predicted isomorphs. Different isomorphic LJ liquids almost collapse onto a single value, and different isomorphs collapse to different constant values with the constant increasing as the order parameter increases. Points show the mean values of order parameters and $\frac{h(\rho^*)}{T^*}$ for each isomorph in part a, and lines show deviation from the mean.

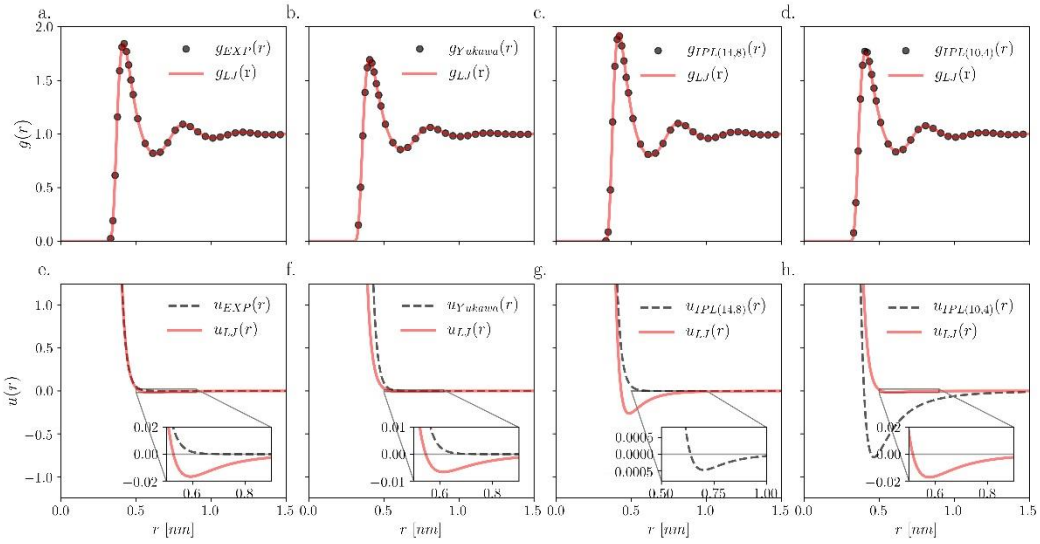


Figure 4.6. Comparison of reference QU pair potentials, QU-equivalent LJ potentials (obtained using DeepILST) as well as their RDFs. The inset shows an enlarged region of the LJ pair potential and its comparison to the reference QU pair potential **(a) and (e)** exponential pair potential as a reference QU pair potential **(b) and (f)** Yukawa pair potential as a reference QU pair potential **(c) and (g)** IPL ($(p, q) = (14, 8)$) pair potential as a reference QU pair potential **(d) and (h)** IPL ($(p, q) = (10, 4)$) pair potential as a reference QU pair potential.

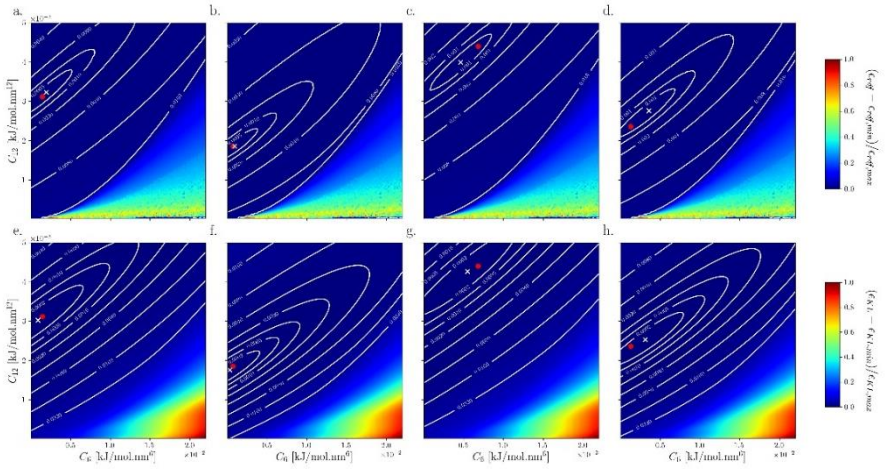


Figure 4.7. Assessment of optimality of QU-equivalent LJ pair potentials (red-filled circles show DeepILST predictions and white cross points show the optimal point obtained from RDF deviation and KL measures). The first row shows the error using the RDF deviation metric. The second row quantifies the error using the KL divergence. **a, e.** exponential pair potential **b, f.** Yukawa pair potential **c, g.** inverse-power-law ($(p, q) = (14, 8)$) pair potential **d, h.** inverse-power-law ($(p, q) = (10, 4)$) pair potential.

CHAPTER 5: Targeting Structural Properties Using Many-body Neural Networks

5.1 Introduction

Classical molecular dynamics (MD) simulation is a strong computational tool to study various physical, chemical, and biological systems[22] by allowing researchers to obtain structural and dynamical properties of these systems with a computational cost far less than the *ab initio* molecular dynamics (AIMD) simulations and resolution far better than the continuum models. The accuracy of the force fields (FFs) used in MD simulations is pivotal to obtain accurate and meaningful results. To date, the most common choice for FFs in MD simulations belongs to some predefined analytical forms, known as empirical FFs, which limit truthful representation of real interactions.[82] The problem is particularly more pronounced in coarse-grained FF developments, where due to many lost degrees of freedom, coarse-grained (CG) models often have features that are not necessarily present in the reference systems.

Various strategies have been developed to improve the accuracy of empirical FFs, such as the inclusion of higher-body interactions, *e.g.,* three-body interactions. For example, a single bead water model with the Stillinger-Weber potential[95], known as the mW model[144], predicts various properties, such as the melting temperature, as accurately as more complex all-atom models. However, such FFs are usually designed with various approximations (*e.g.,* analytical form limitation) motivated by physical intuition, making their application to other systems difficult, if not impossible. In general, the selection of an analytical form and its parameterization requires considerable expertise as well as computer and human time.[145] Motivated by recent progress in the application of machine learning in atomistic-scale simulations[41,80,146], researchers in FF development have embarked on employing various machine learning methods, such as kernel- and neural network-based force fields (NNFFs)[147–151]. Particularly, NNFFs solve the problem of functional form limitation using the universal approximation theorem.

Despite recent success in NNFFs, there are still several issues challenging the widespread usage of NNFFs, such as the arbitrariness of symmetry functions and state dependency of the NNFFs, as well as interoperability. Due to the implementation of NNFFs, they are also less interpretable compared to the empirical FF, as forces and energies are only known once the positions of the whole system are set. Most of the available NNFFs also require accurate force and energy data, which are not always available or measurable. Recent studies have attempted to solve this problem using more interpretable FFs and active learning schemes, but there is still a need for accurate forces and energies.

Recently, researchers started to tackle structure-based coarse-graining methods with machine learning-based force fields such as NNFFs and kernel-based methods[52,152–154]. However, in many cases, the FF is not interpretable. Motivated by the need to overcome challenges with the interpretability of NNFFs and targeting structural properties without energies and forces, we propose a new strategy for developing NNFFs. In this study, instead of building a single NNFF, we build our NNFFs by the expansion of interactions as separate two-, three-, and n-body interactions. We limit our NNFF to a three-body expansion with the possibility of considering higher expansions similarly. Our method, unlike other NNFFs[52,149,150,155], does not necessarily require forces or energies on the particles to train the network and can be trained based on the structural properties of the reference system. This is achieved by replacing backpropagation with a local-search algorithm[156] to train our FFs. To show the capabilities of our method, we study three different models of liquid water, mainly due to the significant importance of water in various physicochemical and biological systems. We start with atomistic-scale force field development using force matching using the forces and positions of the mW model. Showing the success of our method within the force-matching framework, we develop a single bead coarse-grained water model targeting the structural properties of SPCE and *ab initio* water models. The coarse-graining of both the SPCE and AIMD water models in our study is motivated by the drastic dependency of CG models on the structural properties of the reference systems, especially in our study, where we observe slight differences between the location and the value of RDF[12]. We use the strongly constrained and appropriately normed

(SCAN)[157] functional coupled with van der Waals corrections at the D3[158] level to determine the *ab initio* water properties using AIMD simulations (see the Methods section for more details; note that the SCAN functional has shown the ability to produce accurate water properties in close agreement with the experimentally measured properties[159,160]).

We show that our model can reproduce structural properties such as the radial distribution function and angular distribution of both classical and quantum water models. The interpretability of the NNFF in our study allows us to provide insights into the role of two- and three-body interactions in water-like behavior, especially as most of the past studies have focused on the two-body interaction[61,147,161,162]. Even though the SPCE model is one of the widely implemented water models in the literature, the AIMD results slightly differ from the SPCE model in terms of the location of RDF peaks and their values. This slight change might seem trivial, but it is worth observing how it can translate into the CG model. Additionally, coarse-graining[10] has shown that slight differences in RDFs of the two systems can lead to drastic changes in the underlying potentials[12]. Compared with other machine learning-based FFs targeting similar problems, *e.g.*, kernel-based methods, our method can handle complex radial and angular dependencies in the interactions between particles. For example, a spline three-body interaction or kernel-based three-body interaction[147] capturing the complexity of our model with fewer than 1000 free parameters requires at least 10^9 free parameters (1000 for each of the two radial and angular components), which is not possible to train and implement due to its computational and memory costs.

5.2 Methods

5.2.1 Molecular dynamics simulation

The MD simulation of the mW model is based on the Stillinger-Weber potential. It is described by two- and three-body interactions over pairs and triplets, respectively, as follows:

$$u_{sw}^2(r) = A\epsilon\left(B\left(\frac{\sigma}{r}\right)^p - \left(\frac{\sigma}{r}\right)^q\right)\exp\left(\frac{\sigma}{r - a\sigma}\right) \quad (5.1)$$

$$u_{sw}^{3}(r_1, r_2, \theta) = \lambda\epsilon(\cos\theta - \cos\theta_0)^2 \exp\left(\frac{\gamma\sigma}{r_{ij} - a\sigma} + \frac{\gamma\sigma}{r_{ik} - a\sigma}\right) \quad (5.2)$$

where r indicates the radial distance between a pair, r_1, r_2, and θ are the radial distances and the angle between the central atom and two other atoms, respectively (see Figure 5.1c). $\sigma(= 2.3925\ \text{A°})$ and $\epsilon(= 6.189\ \text{kcal/mol})$ are the length and energy scales of the two-body potential, respectively. $\lambda\ (= 23.15)$ is the scaling parameter indicating the strength of tetrahedral interactions. $A = 7.049556277, B = 0.6022245584$, $p = 4$, and $q = 0$ are parameters giving rise to the form and scale of the potential, and the reduced cutoff $a = 1.8$ enforces that all forces and potentials vanish at distances larger than $a\sigma$. The angular term enforces a tetrahedral angle around $\theta_0 = 109.47°$. All MD simulations are performed using LAMMPS[143] with a timestep of 2 fs in the NVT ensemble. The temperature is controlled using the Nosé-Hoover thermostat with a time constant of 0.2 ps.

5.2.2 *Ab initio* molecular dynamics simulation

We performed AIMD simulations using CP2K software[163]. The optimized norm-conserving Gaussian pseudopotentials generated by PBE are used with a double-zeta polarized basis[164,165]. The exchange-correlation energies are treated using the SCAN-D3 approach. The SCAN functional was implemented using the LIBXC library[166]. The Grimme D3 method with zero damping is used to correct the van der Waals interactions. The simulations were performed using an isothermal-isochoric (NVT) ensemble where the constant temperature was controlled using the Nosé-Hoover thermostat[167]. We consider cubic simulation boxes with an edge length of 20.02 Å. Following a 2 ps equilibration time, a production run of 22 ps trajectories is carried out using a 0.5 fs timestep. The radial and angular distribution functions are averaged over the total simulation time with a frequency of 50 fs.

5.2.3 Deep Neural Network

The NNFF developed in this study is composed of two- and three-body interactions, each described by a neural network[38,57]. Note that while we keep the FF limited to two- and three-body interactions, there is no limitation in using higher-order many-body interactions. The two-body term takes the radial distance of a pair as input, and the three-body term takes the radial distances and angle between a central atom and two other atoms in a triplet. Mathematically, the two- and three-body forces of our NNFF can be written as follows:

$$f_{nn}^{2}(\boldsymbol{r}) = \phi_2\big((r_{cf,2} - r)/r_{cf,2}; \mathbf{W}_2, \boldsymbol{b}_2\big).\boldsymbol{e}_r \tag{5.3}$$

$$f_{nn}^{3}(\boldsymbol{r}_1, \boldsymbol{r}_2, \theta) = \mathcal{E}\big(\boldsymbol{e}_{\boldsymbol{r}_1}, \boldsymbol{e}_{\boldsymbol{r}_2}\big)\, \boldsymbol{\phi}_3\big((r_{cf,3} - r_1)/r_{cf,3}, (r_{cf,3} - r_2)/r_{cf,3}, \cos\theta\,; \boldsymbol{W}_3,\ \boldsymbol{b}_3\big) \tag{5.4}$$

where $r_{cf,2}$ and $r_{cf,3}$ are the cutoff distances of two-body and three-body interactions, respectively. $\boldsymbol{e}_{\boldsymbol{r}}$ is the unit vector along the radial vector between two atoms ($\boldsymbol{e}_r = \frac{\boldsymbol{r}}{|\boldsymbol{r}|}$). ϕ_2 is the output of the two-body neural network with $\phi_2 \in \mathbb{R}^1$, and $\boldsymbol{\phi}_3$ is the output of the three-body neural network with $\boldsymbol{\phi}_3 \in \mathbb{R}^4$. $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weights and biases of the neural networks, respectively, and their indices indicate the number of expansions, namely, two- and three-body interactions. We use different neural network depths and widths to model different interactions; however, we use similar nonlinearities, swish nonlinearity ($\text{swish}(x) = \frac{x}{1+\exp(-x)}$) between the hidden layers and tanh nonlinearity for the output layer. $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ is a mapping from $\mathbb{R}^4$ to Cartesian coordinates, written as follows:

$$\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2) = \begin{bmatrix} e_{1,x} & e_{2,x} & e_{1,x} & e_{2,x} \\ e_{1,y} & e_{2,y} & e_{1,y} & e_{2,y} \\ e_{1,z} & e_{2,z} & e_{1,z} & e_{2,z} \end{bmatrix} \tag{5.5}$$

where the first two columns of $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ and the first two elements of $\boldsymbol{\phi}_3$ determine the contribution of force on the central atom from the left particle of a triplet (left, central, right). Similarly, the last two columns of $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2)$ and the last two elements of $\boldsymbol{\phi}_3$ show contributions from the right atom in a triplet

(left, central, right). Note that for n-body interaction within our framework, the following steps are taken: Given the position of the central atom, the n-body interaction depends on the position of $n-1$ particles, described by $n(n-1)/2$ relative distances or angles, and it produces a force vector with a dimension of $\mathbb{R}^{(n-1)^2}$, which gets mapped to the Cartesian coordinates through the mapping matrix, $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{n-1})$. The reaction forces are mapped to the contributing noncentral atom using specific columns of $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{n-1})$.

Having the appropriate force calculation method for a particular pair or triplet, we use two different schemes to train our NNFF, namely, force-matching and structural-matching methods. The force-matching method is a simple method without requiring any CGMD simulations. However, it fails to reproduce the structure of the coarse-grained model, as the effective forces are not equal to instantaneous forces on the CG beads. Additionally, how we map forces on the CG beads changes the FF obtained from the force-matching method[127]. During force matching, one attempts to minimize the variational error between the reference and model forces, which implies minimization of the following loss function using the backpropagation algorithm:

$$\epsilon_{\mathcal{L}}^{f}(\boldsymbol{\theta}, D) = \frac{1}{2|D|}\sum_{i\in D}\left[\boldsymbol{f}_{GT}(i) - \sum_{\forall i \in P}\boldsymbol{f}_{nn}^{2}(\boldsymbol{r}) - \sum_{\forall i \in T}\boldsymbol{f}_{nn}^{3}(\boldsymbol{r}_1, \boldsymbol{r}_2)\right]^2 \tag{5.6}$$

where $\boldsymbol{\theta}$ indicates the free parameter of NNFF, both weights and biases of two- and three-body interactions. D is the training data set. $\boldsymbol{f}_{GT}(i)$ is the ground-truth force on particle i. P and T represent all the unique pairs and triplets, respectively. Summation over pairs and triplets proceeds over those containing atom i without duplicate terms. Note that atom i can be the left, right, or central atom in a triplet, which leads to using the first two, last two, or all the columns of $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2)$, respectively. Note that the above procedure for triplet force calculation enforces Newton's third law of action and reaction. During the training of the network, input data, *i.e.*, various pairs and triplets, along with the corresponding ground truth forces are fed into the neural networks, which are then used to train through backpropagation. Based on our experiments, it is easier to use convolutional neural networks to calculate

forces for the above force-matching method and model each of the two- and three-body interactions as separate filters over pairs and triplets, respectively. To eliminate the effects due to the particles outside of the cutoff distance, we multiply the output of the convolutional neural network with a rectified linear unit with linearly transformed radial distance as input ($\max(0, (r_{cf} - r)/r_{cf})$). The Adam optimizer[58] is used to adjust the weights and biases, and an additional L2 regularization is added to the loss function to avoid overfitting.

Furthermore, we develop a new route to train our NNFF without requiring direct access to forces. We apply our method to reproduce the structural properties of reference systems (note that the method is general and can be applied to various properties). Our method solves the problem with force matching in the reproduction of structural properties and the fact that reference forces are not always target or available during FF development. To do so, the backpropagation algorithm, which is used in the force-matching method, is replaced with the local search algorithm. This is similar to using evolutionary or reinforcement learning methods to train NNFF. In other words, MD acts as an environment where the policy predicts forces and forces are used for MD simulation[168,169]. The cost determined at the end of the MD simulation is the structural agreement between the CG and reference models[73,170]. Within the local-search algorithm, the training for such policy is done through two steps. First, a small fraction of the free parameters of NNFF are selected randomly, followed by a perturbation of selected parameters. The perturbation is accepted if it improves the loss function. Within structure-based coarse-graining method, the loss function is defined as:

$$\epsilon_{\mathcal{L}}^{s}(\theta) = \frac{\sum_{n=1}^{N} |g_{ref}(r_n) - g_{cg}(r_n)|}{\sum_{n=1}^{N} |g_{ref}(r_n)|} + \alpha \sum_{l=1}^{L} \frac{\sum_{m=1}^{M} |a_{ref}^{(l)}(\theta_m) - a_{cg}^{(l)}(\theta_m)|}{L} \tag{5.7}$$

where g_{ref} and g_{cg} represent the radial distribution function (RDF) of the reference and CG models. a_{ref} and a_{cg} show the angular distribution function (ADF) of the reference and CG models. α indicates the importance of ADF in the structural similarity of the reference and CG models (with only 2-body interaction α is set to 0). The RDF is discretized into *N* segments from 0 to the cutoff distance of two-

body interactions. Similarly, the ADF is discretized into *M* segments from 0 to 180 degrees. Furthermore, ADF is calculated for L different cutoff distances (see the supporting information for more discussion on the discretization). Additionally, note that for each iteration of local search training, a short MD simulation is performed to obtain the RDF and ADF of CG models and therefore the loss function.

5.3 Results and Discussions

5.3.1 Force-Matching

The objective of the force-matching method is to reproduce the ground truth forces. In this study, we use force matching to show that our scheme of training NNFF leads to retrieving the ground-truth force field of the mW model in the form of NNFF. To do so, we train our NNFF based on forces obtained at multiple temperatures. We perform an NPT simulation at 300 K and 1 bar to obtain the correct density, followed by a simulation in the NVT ensemble at T = 270, 300, 330 K to obtain the positions and forces of the mW model particles.

Once the data were obtained, we trained our NNFF with a cutoff distance of 4.310, equal to the mW model cutoff distance. The two- and three-body interactions in NNFF form are modeled with 3 and 4 hidden layers, respectively. All the hidden layers except the output layer have swish nonlinearity, and the output layer has tanh nonlinearity. We trained the network for approximately $2x10^6$ iterations with a batch size of 64. In Figure 5.2a, we compare the RDF between the mW and NNFF models, followed by a comparison of the two-body forces of the ground truth mW and NNFF models at different radial distances, as shown in Figure 5.2b. Similarly, we compare the ADF of the mW and NNFF models with various cutoff distances in Figure 5.3a-b. We also compare the norm of the three-body forces from the mW and NNFF models at different angles for the equidistant configurations of neighboring atoms in Figure 5.3c. The method shows good agreement in terms of RDF and ADF reproducibility, as well as recovering the ground truth forces.

5.3.2 Structure-Matching

The force-matching scheme is not able to reproduce the structural properties of the reference system. To solve this problem, we train different NNFFs pretrained by force-matching or iterative Boltzmann inversion methods using the local search method to match the structural properties of the reference system. The local-search algorithm is started by randomly picking a small fraction of NNFF parameters, followed by adding a random perturbation to the selected parameters. The new candidate NNFF is then used for running coarse-grained MD simulations. If the results are improved, then the new candidate becomes the best candidate, and additional perturbations are applied to it; otherwise, a new fraction of NNFF parameters are selected. The procedure continues until desired results are obtained.[156]

In our study, we use two different reference systems, classical water modeled using the SPCE model and AIMD water modeled using the SCAN functional. For each reference system, we develop two different NNFFs, one with only two-body interactions (NN2) and one with two- and three-body interactions (NN3). In addition to the interpretability of both the NN2 and NN3 models, the models allow us to understand the role of 2body and 3body interactions in water-like behavior. Both NN2 and NN3 model the two-body interaction with a cutoff distance of 0.8 nm, and NN3 models the three-body interaction with a cutoff of 0.45 nm. The NN2 model with only two-body interaction has 3 hidden layers with widths of 4, 16, and 4. In the NN3 model, the two-body interaction has 3 hidden layers with widths of 6, 12, and 6, and the three-body interaction has 4 hidden layers with widths of 3, 6, 12, and 8. All the hidden layers except the output layer have swish nonlinearity, and the output layer has tanh nonlinearity. Note that both RDF and ADF are calculated for the oxygen atoms of water molecules, *i.e.,* the oxygen atom of water molecules is the mapped CG model.

Both NN2 and NN3 reproduce the RDF of oxygen-oxygen with high accuracy with an error of less than 0.01 (see Figure 5.4a). In Figure 5.4b, we compare the two-body interaction between the NN2 and NN3 models. As seen in Figure 4b, the NN3 model has a single-well form, while NN2 has a double-well

interaction. The double-well interaction observed in the NN2 model is consistent with previous studies that used pairwise interactions[171,172]. The interesting finding is that the double-well potential vanishes for the NN3 models. This indicates that the pairwise interaction might not be the best representation of water-like behavior. This is particularly pronounced in the ADF comparison between the SPCE, NN2, and NN3 models. In Figure 5.5a-b, we show that NN3 captures the ADF behavior of SPCE water; however, NN2 fails, especially at shorter distances. Various studies allude to the importance of angular distribution for water-like behavior, such as liquid-liquid transition. Therefore, capturing it through the NN3 model is significant progress towards deciphering the angular signature of water[173]. In Figure 5.5c, we show the norm of the three-body forces of the NN3 model at different angles for the equidistant configurations of neighboring atoms. Figure 5.5c indicates that the three-body forces present in the NN3 model have far more complex behavior than those of the mW model, which is limited by its analytical form.

Although the SPCE model is one of the widely adapted water models in the literature, the AIMD results slightly differ from the SPCE model in terms of the location of RDF peaks and their values. This slight change might seem trivial, but it is worth observing how it translates into the CG model. Following similar training steps as the SPCE-based CG model, we train the NN2 and NN3 models to reproduce the structural properties of the AIMD water model. In Figure 5.6a, we compare the RDF of the NN2 and NN3 models with the AIMD reference with an RDF error less than 0.03 for both cases. In Figure 5.6b, we show the two-body interactions of both the NN2 and NN3 models. Similar to CG models of SPCE, we observe double-well and single-well interactions for NN2 and NN3, respectively. A one-to-one comparison between the NN2 CG models of the AIMD and SPCE water models shows slight differences in the location of wells between the NN2 and NN3 models, as well as the transition between the wells. We contribute this difference to the differences between the RDFs of the reference systems. In Figures 5.7a-b, we compare the ADF of the reference AIMD water and CG models. Again, NN2 fails to capture the ADF behavior of the reference system, and NN3 captures it. Quantitatively, the ADF error for the

NN3 model lies in the range 0.017 to 0.041, depending on the cutoff distance, and the NN2 ADF error is in the range 0.14 to 0.38, which is one order of magnitude larger than the NN3 CG model.

We briefly recap the water-like behavior of CG NN2 and NN3 CG models. We observe that both the NN2 and NN3 models can capture the water RDF; however, the NN3 model captures the angular distribution of water, and NN2 fails to capture it. We observe a double-well two-well interaction in NN2 CG models; however, NN3 only has a single well. The fact that in the presence of a three-body interaction, the double-well structure of the two-body interaction vanishes shows that waterlike behavior is best described by a higher-order expansion of interaction rather than a pairwise interaction with double-well. In particular, our NNFF is not limited to a particular form. This is also consistent with the directional dependency of hydrogen bonds[174,175], which is best described by a three-body interaction rather than a double-well two-body interaction.

5.4 Conclusion

In this study, we train a neural network-based force field with two- and three-body interactions, which makes the developed force field interpretable. Within our framework, the requirement for accurate forces and energies is eliminated by using the local search algorithm instead of backpropagation. Therefore, our method is suitable for application in structure-based coarse-graining. To show the capability of our method, we successfully develop coarse-grained models of both classical and *ab initio* water models. We also investigate the dependency of the coarse-grained force field of water on the number of expansions, which shows that the double-well interaction, known as a signature of water-like behavior among spherically symmetric pairwise interactions, vanishes with the inclusion of three-body interactions. We also notice that the two-body interaction fails to reproduce the angular distribution of water, especially over a short range. Based on our findings, we conclude that water-like behavior is better captured using the three-body interaction, which is consistent with the directional dependency of interactions in water.

5.5 Figures

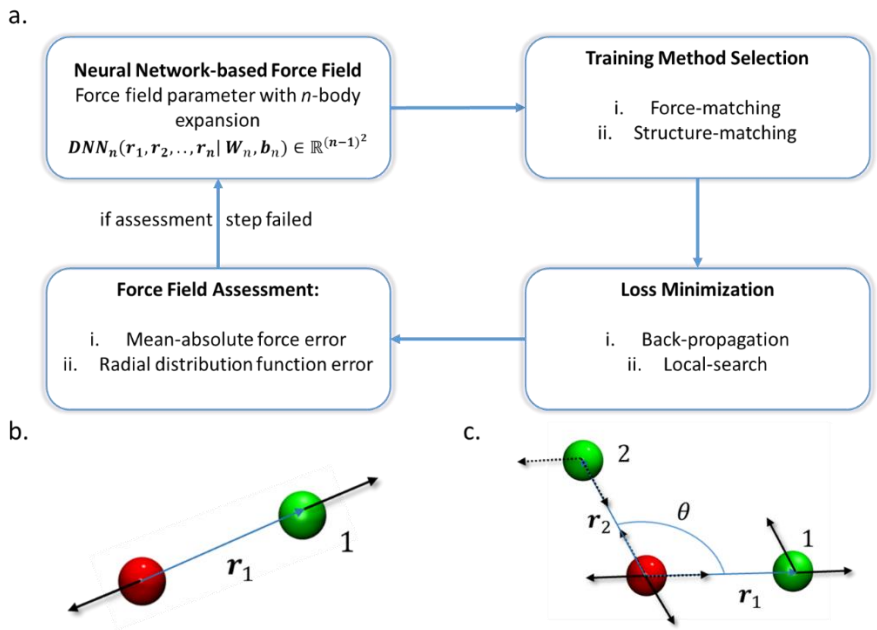


Figure 5.1. Neural network-based force-field training. a) Force-field development begins by selection of n-body expansion and initialization of neural network architecture, weights, and biases. Based on the objective of force field development and its differentiability, either back-propagation or local-search methods are employed to minimize the loss function. Once the loss function is minimized, its generalizability is assessed. If the neural network force field did not pass the criteria for its assessment, the cycle repeats. The n-body term in the neural-network force-field depends on the position of $n-1$ particles, described by $n(n+1)/2$ relative distances or angles, and it produces a force vector with a dimension of $\mathbb{R}^{(n-1)^2}$, which gets mapped to the Cartesian coordinates through the mapping matrix, $\mathcal{E}(\boldsymbol{e}_1, \boldsymbol{e}_2, \dots, \boldsymbol{e}_{n-1})$. b) The two-body term for a tagged particle (red particle) only depends on the radial distance ($|\boldsymbol{r}_1|$) of its neighboring particle (green particle) and the mapping is the unit vector between two particles ($\boldsymbol{e}_1 = \boldsymbol{r}_1/|\boldsymbol{r}_1|$). c) The three-body term depends on the position of three-particles best described by two radial distances ($|\boldsymbol{r}_1|$ and $|\boldsymbol{r}_2|$) and the angle between them (θ). The neural network predicts four values, acting on the central red particle, where the first two values, shown with the solid black lines, correspond to action-and-reaction from the first neighboring particle and the second two, shown with dashed black lines, correspond to action-and-reaction from the second neighboring particle. The mapping matrix for a three-body term is described in Eq. 5.

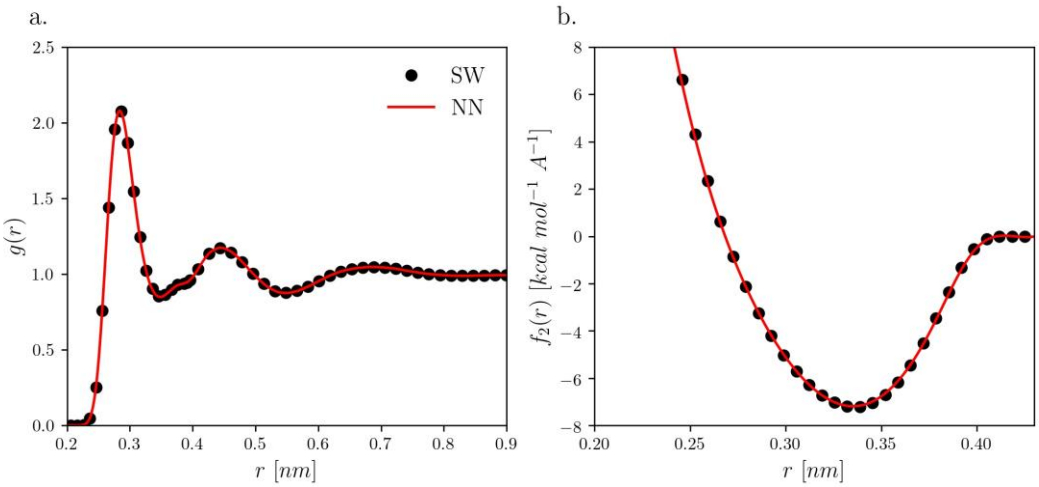


Figure 5.2. Radial distribution function and two-body force comparison between neural network force field and ground truth mW model. a. Comparison between RDFs of mW and NN-based models b. Comparison between 2body force of mW and NN-based models. Black circles show the mW model results and red lines show NN-based force field results.

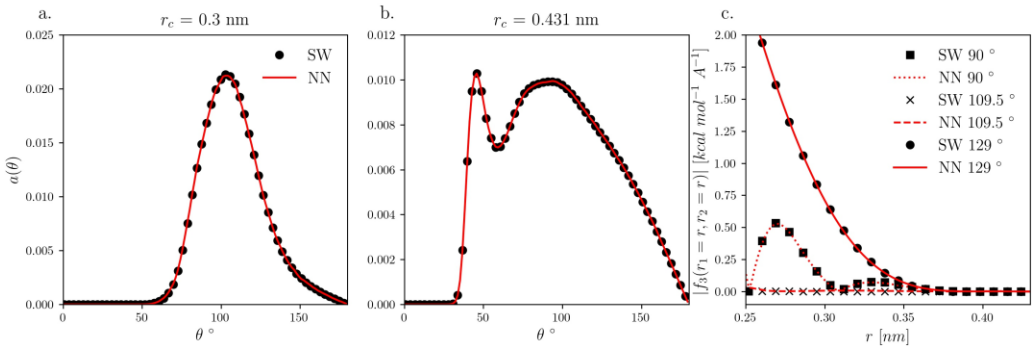


Figure 5.3. Angular distribution functions and three-body force comparison between neural network force field and ground truth mW model. The agreement between the two models is excellent. a. Comparison between ADF mW and NN-based models with cutoff 0.3 nm b. Comparison between ADF mW and NN-based models with cutoff 0.43 nm c. Comparison between the norm of 3body force on the central particle of mW and NN-based models at different angles in the equidistant radial configuration. Black points show the mW model results and red lines show NN-based model results.

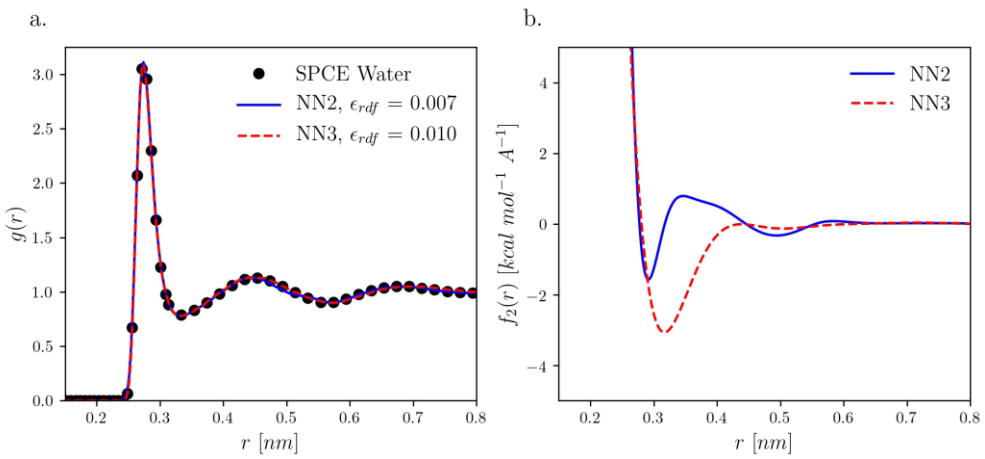


Figure 5.4. Radial distribution function and two-body force comparison between different neural network force fields and SPCE model. Both NN2 and NN3 models have a two-body force field, however, NN3 has additional NN modeling its 3-body interaction. Both models produce the RDF properties of SPCE models. a. Comparison between RDFs of SPCE model and NN-based models b. Comparison between 2body force of NN2 and NN3, NN2 shows the double-well force profile, while NN3 has a single well, and it is interaction range is far shorter than the NN2 model. Black circles show the SPCE model results and blue solid and red dashed lines show NN2 and NN3 model results, respectively.

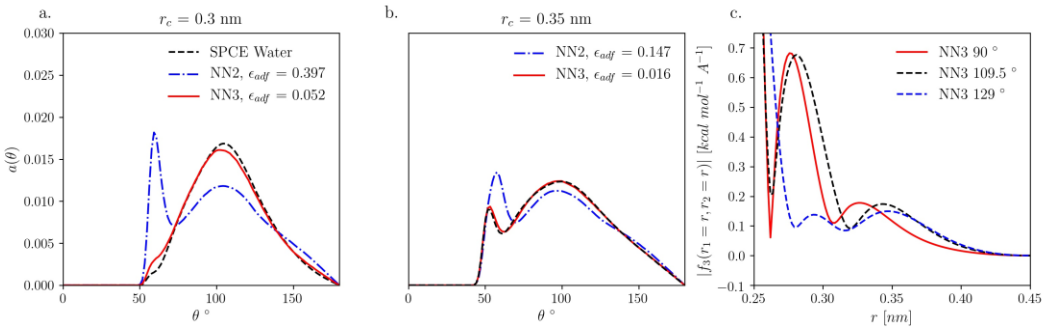


Figure 5.5. Angular distribution functions and three-body force comparison between different neural network force fields and SPCE model. NN3 models with three-body interaction capture ADF behavior of water far better than the NN2 model. a. Comparison between ADFs of SPCE, NN2, and NN3 models within cutoff 0.3 nm b. Comparison between ADFs of SPCE, NN2, and NN3 models within cutoff 0.35 nm c. Norm of 3body force on the central particle of NN3 models at different angles in the equidistant radial configuration. In a and b, black dashed lines show the SPCE model results and dashed blue and solid red lines show the NN2 and NN3 models results. All lines in part c show the norm of the force of 3body interaction.

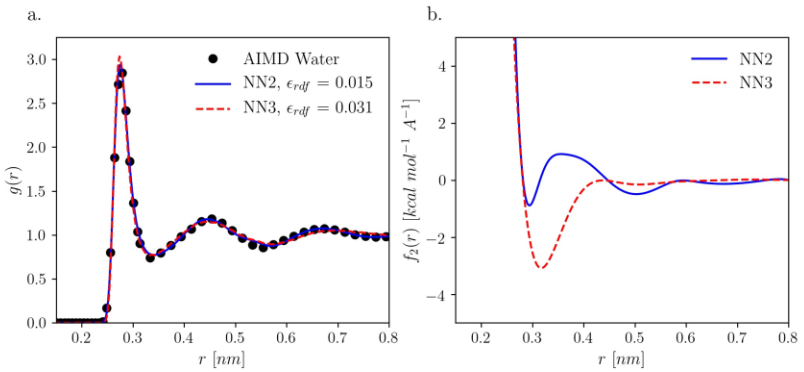


Figure 5.6. Radial distribution function and two-body force comparison between different neural network force fields and AIMD model. Both NN2 and NN3 models have a two-body force field, however, NN3 has additional NN modeling its 3-body interaction. Both models produce the RDF properties of AIMD models. a. Comparison between RDFs of AIMD model and NN-based models b. Comparison between 2body force of NN2 and NN3, NN2 shows the double-well force profile, while NN3 has a single well, and it is interaction range is far shorter than the NN2 model. Black circles show the AIMD model results and blue solid and red dashed lines show NN2 and NN3 models results, respectively.

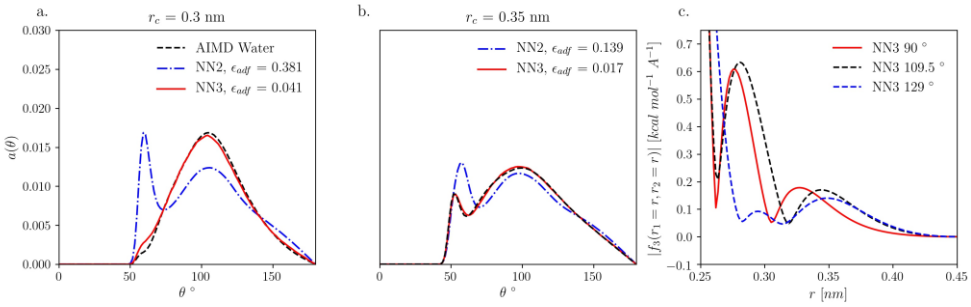


Figure 5.7. Angular distribution functions and three-body force comparison between different neural network force fields and AIMD model. NN3 models with three-body interaction capture ADF behavior of water far better than the NN2 model. a. Comparison between ADFs of AIMD, NN2, and NN3 models within cutoff 0.3 nm b. comparison between ADFs of AIMD, NN2, and NN3 models within cutoff 0.35 nm c. Norm of 3body force on the central particle of NN3 models at different angles in the equidistant radial configuration. In a and b, black dashed lines show the AIMD model results, and dashed blue and solid red lines show the NN2 and NN3 model results. All lines in part c show the norm of the force of 3body interaction.

CHAPTER 6: Graph Neural Network-based Water Phase Identification

6.1 Introduction

Water is an indispensable part of many physicochemical and biological phenomena, and its phase can significantly alter physicochemical and biological phenomena such as CO2 reduction,[176] proton transport,[177] power generation,[178], and water desalination[179]. Furthermore, the properties of water itself, as the liquid sustaining life, depend on its phase such as diffusion coefficient, dielectric permittivity, and density as well as structural properties.[180] Therefore, it is of great importance to accurately identify different water phases. Like most of the other liquids, most of our knowledge about water is obtained through computational studies, where atomistic level data about the positions and velocities are available.[2,11] Therefore, the prediction of water phases from water molecules topology, *i.e.* configuration of other molecules around a tagged water molecule, is a task worth studying and understanding, especially for confined systems such as carbon nanotubes (CNTs) due to CNTs technological applications.

Due to the high dimensionality and uninterpretable nature of atomistic simulation data, researchers have developed a wide variety of order parameters to reduce dimensionality and predict the phase of a system from reduced dimensions. Motivated by the importance of water in various areas, water is studied through various order parameters (OPs) such as bond-order parameter (BOP),[181,182] tetrahedral order parameter[173], and local-structure index.[137,183] Even though these order parameters are widely adopted in various studies ranging from ice nucleation[184], phase discrimination/identification,[181] liquid-liquid transitions,[185,186] free energy calculation[187], however, they are far away from perfect. In many cases, it requires lots of domain expertise and efforts to combine multiple order parameters to reach conclusive findings or come up with new order parameters.[187,188] The problem is particularly pronounced for confined systems as OPs are usually defined for homogenous and bulk systems, which is not the case for

confined water. Furthermore, due to the interplay between fluid-fluid and fluid-wall interactions in the confined system,[189] confined systems have a richer physics accompanied with anomalous behavior in the phase transition region, where both continuous and discontinuous phase transition can occur (the discontinuous phase transition characterized by a sharp change in the potential energy, enthalpy, or OP of the system, while continuous phase transition shows only a critical point).[190,191] Various computational and experimental studies are performed to investigate and identify phase behavior of confined water;[192,193] however, there are remaining pieces to fundamentally understand confined water phase behavior, one of which is the design of a route to predicting the phase of water directly from positional information, especially for the confined system.

Reminiscent of order parameter design in the phase identification task is the kernel and feature engineering in the image, speech, and text processing, which required lots of domain expertise and human time.[38,57,89] During the last decades, however, the process of kernel and feature engineering has been revolutionized by deep learning-based methods, which are adapted for a wide variety of applications in physics, chemistry, and biology. Particularly, water, as one of the most complex and important liquids, has been successfully studied through various deep learning methods in application such as force field development and phase-identifications.[80,130,149,150,155,194] However, the recent deep-learning method still tries to use traditional OPs as a feature for phase-identification of water, which still does not solve the issue with their definition.

The main bottleneck of phase identification stems from the nature of the data obtained from MD simulation. The data used for deep learning should pose several properties, permutation, rotational, and translational invariance.[57] The atomic coordinates obtained from MD simulation do not pose these properties, which hinders the application of many conventional deep learning algorithms unless some sort of arbitrary transformation is applied. Initial attempts for classification of water phase using deep learning-based methods started by the study where multiple features requiring multiple transformations are fed into a multilayer-perceptron. However, the method requires arbitrary rotational transformations of

the dataset to enforce the rotational invariance. The recent progress in graph neural network (GNN) provide a suitable tool to deal with the atomistic data as they are best described in a non-Elucidation space.[195] In addition to addressing permutation, rotational, and translation invariance, GNN addresses the variable size of the data, which is the case for confined water with different numbers of neighbors depending on water phase and its distance from the wall. GCIceNet is developed to solve the problem of rotational and permutational invariance using GNN. Even though GCIceNet is particularly successful, however, GCIce constructs node features using OPs, which is an edge feature (it depends on the distance between atoms). Additionally, OPs used as node features are not well-defined for confined systems.

In this study, we use the latest advances in GNN, particularly edge-conditioned convolutional (ECC) graph neural network, to address the problem of phase identification of water in bulk, interface, and confined systems in an end-to-end fashion.[196] In short, ECC is successfully applied to the point Cloud dataset, which mimics the problem of phase identifications in many ways. We formulate the phase-identification problem as a graph classification task and use the ECC layers to remove the need for human-engineered order parameters. To do so, we construct our graphs $G = \{V, E\}$ by collecting the oxygen atoms within the cut-off distance of a tagged oxygen atom, based on the performance and computational cost, we keep all the oxygens or several closet oxygen atoms. The oxygen atoms form the nodes (V) of our graph, and the pairwise distance between all oxygen atoms (nodes) is the edge feature. The node feature ($X \in \mathbb{R}^{|V|\times 2}$) is the one-hot encoded vector $\{0,1\}$, where the tagged oxygen atom *i.e.*, the water molecule we want to classify its phase has a different node feature compared to its neighboring oxygen atoms. We use a full adjacency matrix to classify phases ($A = [1]_{|V|\times|V|} - I_{|V|}$). We collect all the pairwise distances between all nodes as the edge feature of our graphs. Additionally, we collect information regarding the hydrogen bonds between water molecules by determining whether an edge corresponds to a donor-acceptor or acceptor-donor hydrogen-bond as well as no-hydrogen-bond. The hydrogen bonding is only information incorporated as a feature in neural network training, as nodes corresponding to it are coarse-grained in graph representation. The dimension of an edge, therefore, is a

vector of size 4 ($E \in \mathbb{R}^{|V|\times|V|\times 4}$). The output of the graph classification task for the bulk water is a vector of dimension n_c, where n_c is the number of different water phases in the dataset. Particularly, we study 9 different phases of water such as Ih, Ic, II, III, VI, VII, VIII, and IX Ices as well as liquid water. For interface and confined systems, we use similar inputs as bulk systems, but our output is a binary value indicating whether water is liquid or solid. GNNs are trained to predict whether a particular configuration of atoms is liquid-like or solid-like for various CNTs or interface water at various temperatures. We study CNT 10x10, inside which both continuous and discontinuous phase transition can occur. Our reference solid and liquid systems used for training are picked from temperatures away from phase transition temperature. The model successfully shows both sharp and smooth changes in the fraction of liquid-like molecules near the phase transition, allowing us to predict phase transition temperature faster than normal methods. This is an advantage of our method as previous studies need averaging over lots of trajectories to calculate OP or thermodynamic properties to obtain phase transition temperatures.

The rest of this section is organized as follows. First, we describe the details of MD simulation and calculation of order parameters, followed by training of graph neural network and random forest models and comparisons between their performance. Finally, we summarize the findings of our study.

6.2 Methods

6.2.1 Molecular Dynamics Simulation

Molecular dynamics (MD) simulations of water in bulk and confinement are performed using GROMACS package.[197] Water is modeled using TIP4P/Ice model as it performs better for phase transitions.[198] For interface system, we use the interface of Ice h/vapor, which is known as a challenging phenomenon, where a quasi-like layer can form due to the missing hydrogen-bonding in the interface. For confined cases, carbon-water interactions are modeled using the parameters from the reference[199]. The temperature and pressure of the systems are controlled using the Nose-Hoover thermostat with a time constant of 0.2 ps and Parrinello-Rahman barostat with a time constant of 2.0 ps, respectively.[200] Initial

configurations of the bulk system are generated using GenIce packages.[201] After energy minimization steps on the initial bulk configurations, MD simulation is performed for 25 ns at corresponding temperature and pressure of phase. The data for machine learning models training as well as OPs are obtained from the last 10 ns of simulation. For the confined systems, we fill CNTs with reservoir. Later, the isolated periodic CNT mimicking an infinite CNT is simulated at different temperatures, by gradually decreasing temperature from 390 K to 10 K with the rate of 1 K/ns. For every 10 K decrease, we simulate the system for 20 ns, the last 10 ns are used for post-processing.

To compare the performance of GNN with conventional machine learning methods as a baseline, we calculate OPs including LSI, BOP, and tetrahedral OP. The LSI indicates the translational order of the system, and it considers $\left|\mathcal{N}\left(i, r_{cf} = 0.37\ nm\right)\right|$ neighboring water molecules by ordering them in ascending pairwise distances ($r_{j+1} > r_j \ \forall\, j \in \mathcal{N}\left(i, r_{cf} = 0.37\ nm\right)$). Mathematically, it is defined as,

$$LSI = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} [\Delta(j) - \overline{\Delta}]^2 \tag{6.1}$$

where $\Delta(j)$ is the difference between the pairwise distance of two neighboring water molecules *i.e.* ($\Delta(j) = r_{j+1} - r_j$) and $\overline{\Delta}$ is the average value of $\Delta(j)$.

The BOP of order l (Q_l) is the other OP used in the baseline machine learning method, where it is a coarse-grained representation of Steinhardt parameter q_{lm}, which can be expressed as follows,

$$q_{lm}(i) = \frac{1}{\left|\mathcal{N}\left(i,\ r_{cf} = r_{cf,6}\right)\right|} \sum_{j \in \mathcal{N}(i)} Y_{lm}(\theta_{ij}, \phi_{ij}) \tag{6.2}$$

Where Y_{lm} is the spherical harmonic function of degree l and order m. θ_{ij} and ϕ_{ij} are polar angles. The cut-off distance ($r_{cf} = r_{cf,6}$) of the neighbor list is chosen such that $\left|\mathcal{N}\left(i,\ r_{cf} = r_{cf,6}\right)\right|$ equals 6. The BOP of order l and degree m is defined as,

$$Q_{lm}(i) = \frac{1}{\left|\mathcal{N}\left(i,\ r_{cf} = r_{cf,6}\right)\right| + 1}\left(q_{lm}(i) + \sum_{j \in \mathcal{N}(i)} q_{lm}\left(\theta_{ij}, \phi_{ij}\right)\right) \tag{6.3}$$

which is coarse-grained by averaging over degree through the following expression,

$$Q_l(i) = \sqrt{\frac{4\pi}{2l+1}\sum_{m=-l}^{l} |Q_{lm}|^2} \tag{6.4}$$

The tetrahedral OP is defined based on the four nearest molecules and takes a value between 0 and 1, where 0 and 1 correspond to an ideal gas and perfect tetrahedron, respectively. It can be expressed as,

$$q_{te} = 1 - \frac{3}{8}\sum_{j=1}^{3}\sum_{k=j+1}^{4}\left(\cos\psi_{jk} + \frac{1}{3}\right) \tag{6.5}$$

where ψ_{jk} is the angle formed from the tagged molecule and two of the four closet water molecules. BOPs and tetrahedral OP calculated using PyBoo package are used to train the baseline machine learning models.[202] In this study, we use the random forest as our baseline.[203] We also store the pairwise distance between atoms within a cut-off distance from a tagged water molecule as an edge feature for GNN training along with the presence and type of hydrogen-bonding as a one-hot-encoded vector.

6.2.2 Machine Learning

To show the superiority of GNN over other machine learning algorithms, we train a baseline machine learning method *i.e.,* random forest (RF). Before going into the details of our training, we describe the task and procedure we have taken. We treat the problem as a classification problem, and for the bulk system, our output is a one-hot encoded vector of 9 different phases. The task in the confined and interface system, however, is simplified to the identification of solid and liquid phases, where the output is binary *i.e.,* 0 or 1. The data for classification is selected from multiple temperatures, both above and below the melting temperature of TIP4P/Ice models, exposing the model to both liquid and solid phases. The performance particularly is interesting for confined systems, where determination of phase transition temperature from conventional method (first-order change in order parameters) is not straightforward and

requires long simulation. It can also suffer from lots of noise as the timescale of phase transition can be large for continuous phase transition. However, the model can still predict phase behavior well (more quantitative analyses are provided later).

Random Forest

Random forest algorithm (RF) as an ensemble learning method selects a subset of features (in our cases a vector of dimension 7, composed of LSI and tetrahedral order parameters as well as BOPs $\{4,6,8,10,12\}$. For each selected subset, a decision tree is trained by randomly selecting a subset of features and dataset, followed by the construction of decision trees. Once the training of various trees is done, a majority vote is taken to determine the class for a given data. We apply grid search with 5-fold cross-validation to obtain optimal depth and number of trees for RF.

Graph Neural Network

The graphs in this study are denoted by $G = \{X, E\}$, where $X = \{x_i \in \mathbb{R}^2 | i = 1, \dots, N\}$ is the set of all nodes (oxygen atom of water molecules) with a dimension of 2 corresponding to one-hot-encoding based on whether the node is a tagged oxygen atom or a neighboring oxygen atom, and $E = \left\{e_{ij} \in \mathbb{R}^4 \middle| x_i, x_j \in X\right\}$ is the set of edges with 4 attributes *i.e.*, the pairwise distance and the one-hot-encoding of hydrogen-bonding corresponding to donor-acceptor, acceptor-donor, or no-hydrogen-bond cases. The graph can also be described using the adjacency matrix, a binary matrix of dimension $|X|^2$. Elements of the adjacency matrix determine connections between nodes $A = \left[\delta_{ij}\right]_{|X|\times|X|}$. Note that depending on the computational cost and classification performance, we use different numbers of water molecules to form the graph (see Figure 6.1 for schematic representations of graph construction).

In general, most of the GNN methods belong to the message-passing networks, which utilize combinations of message, aggregation, and update.[195] In this study, we use the ECC layer to build our GNN model.[196] The hidden representation of nodes h^l at layer l is equal to a weighted sum of hidden

representation h^{l-1} in its neighborhood. The weights in the ECC are generated by another network, also known as filter generating networks, which is usually modeled with a MLP with trainable parameters. Mathematically, the following operations are performed in the *l*-th layer,

$$h^l(i) = h^{l-1}(i)W^l_{root} + \sum_{j \in N(i)} h^{l-1}(j)\mathrm{MLP}\left(e_{ji}, w^l\right) + b^l \tag{6.6}$$

where b^l is the *l*-the layer bias, and $N(i)$ is the neighborhood of node i ($N(i) = \{j; (j,i) \in E\}$). Note that h^0 corresponds to input feature x and w^l are learnable parameters of the multi-layer perceptron. After 3 or 4 ECC layers, we use a pooling function (sum pooling) to find a representation for each graph. The role of pooling is to reduce node embeddings of the whole graph into a single vector. Additionally, the pooling layer should be invariant to the permutation of nodes, we use sum function as our pooling layer. The pooled representation is fed to a multi-layer perceptron with 1 hidden layer. All the layers, except the last of multi-layer perceptron, use the ReLU activation function.

To train the parameter of the GNN we use either binary- or categorical-cross-entropy losses defined as,

$$\mathcal{L} = -\sum_{i \in n_c} p_i \log q_i \tag{6.7}$$

Where p_i an $\log q_i$ are *i*-th element of vectors with dimension equal to the number of classes (n_c) representing one-hot-encoding and GNN predictions, respectively. The Adam optimizer is used to train the model with a learning rate of 0.00005 for 100000 epochs with early stopping if the loss is saturated for 5 consecutive epochs on the validations dataset (0.2 of the dataset). The batch size of 16, 32, or 64 is used depending on the computational cost. Spektral and TensorFlow packages are used to perform GNN training, and the top2phase package developed as a python package for broader usage.

6.3 Results and Discussions

We first perform data-exploratory analysis to examine the sufficiency of BOPs to separate different bulk water phases (bulk systems are shown in Figure 6.2a). To do so, we obtain 2D scatter plots of BOPs for two sets of degrees, namely (q_6, q_8) and (q_{10}, q_{12}). In Figure 6.2b, we show the results of the analysis, where we observe a large overlap between any selected BOPs. Following this step, we train RF methods with different numbers of trees and depth and find the optimal RF parameters. The model reaches an average accuracy of 89.2 percent. Training the GNN model with the same dataset leads to an average accuracy of 99.9 percent. Figure 6.3 shows more quantitative analysis on the accuracy of both GNN and RF models based on the confusion matrix. The confusion matrix shows the percentage of dataset misclassified for off-diagonal elements, and diagonal elements show the percentage of correctly classified samples per class. The GNN confusion matrix (Figure 6.3b) shows far superior behavior as off-diagonal elements are far less than their counterparts in the RF confusion matrix (Figure 6.3a).

Following on the same line, we study Ice h/vapor system, where we simulate system for the temperature range of 10 K to 300 K with a 10 K step (see Figure 6.4 for explanatory data analysis as well as the schematic representation of waters at different temperatures before and during phase transition). The experimental and computational investigation show formation of a quasi-liquid layer at the interface of Ice h /vapor. The predicted melting temperature from the experiment and simulation is around 270 ± 5 K. We select temperatures of [10,140] K and [290, 300] K as our reference solid and liquid systems for ML training. After training the model, which shows an accuracy of 99% for GNN and 0.97% for RF, we feed the data from other temperatures to predict melting temperature and compare classification results obtained from RF and GNN. To do so, we compare the liquid fraction at each temperature using both RF and GNN models in Figure 6.5. Additionally, we show the potential energy of the system, which shows a sharp change near melting temperature. The GNN predicts a melting temperature of 275 K and RF predict it as 275 K, and potential energy indicates a melting temperature of 275.0 ± 2.5 K. The behavior of GNN

is monotonic within the temperature range of our study, showing an increasing number of liquid-like molecules, while RF shows a non-monotonic and inconsistent behavior with temperature increase.

Now, we study confined systems, where we simulate water confined inside CNT 10x10 for the temperature range of 10 K to 390 K with a 10 K step (see Figure 6.6 for explanatory data analysis as well as the schematic representation of waters in liquid and solid phase for water with average densities of 16.75 nm^{-3} and 19.14 nm^{-3}). Confined water in general shows more complex behavior compared to bulk water, for example with the increase in density phase transition becomes continuous, especially for CNTs with a smaller diameter. This phenomenon is usually attributed to the interplay between the interactions of interface-water and water-water. As shown in figure 6.6a-d, the solid phase of water inside CNT 10x10 show both heptagon and heptagon with single-file water, respectively at densities of 16.75 nm^{-3} (low) and 19.14 nm^{-3} (high) and liquid phase of both CNT looks like each other. The larger overlap of scatter plots of BOPs in high-density cases BOPs shown in Figure 6.6e-h, indicates difficulty in using BOPs for high-density cases. Note that the high-density case corresponds to a continuous phase transition. The predicted melting temperatures for low- and high-density cases are around 270 ± 10 K and 290 ± 10, respectively. The trend is consistent with previous computational studies. Like the Ice h/vapor case, we select two representative temperatures for both liquid and solid, in this case, we use [10,150] K and [310, 380] K as our reference solid and liquid system, respectively. After training the model, the GNN model achieves 0.994 and 0.949 accuracy, respectively for low- and high-density cases, the accuracy of RF is lower than of GNN, achieving 0.997 and 0.809 accuracies, respectively for low and high densities (see SI for confusion matrix and more details on model performance). The GNN outperforms RF in the high-density case, which shows the capabilities of GNN in complex environments. The lower accuracy of RF is attributed to the large overlap of BOPs as shown in Figure 6.6 e-h. GNN, however, learns its featurization based on data and does not face many difficulties in distinguishing solid and liquid phases apart. Once the models are trained, we feed the data from other temperatures to predict melting temperature and compare the results obtained from RF and GNN. In Figure 6.7a-b, we compare

the liquid fraction at each temperature using both RF and GNN models, respectively for low- and high-density cases. Additionally, we show potential energy and axial diffusion coefficient, which shows a sharp change near melting temperature for low density and smooth transition for high density, signatures of discontinuous and continuous phase transition, respectively. The predicted melting temperatures are close to MD simulation results using both models. However, the behavior of GNN is again more monotonic with temperature change as shown in Figure 6.7a-b the fractions of liquid-like molecules are a non-decreasing function of liquid-like molecules, while RF shows a non-monotonic and inconsistent behavior with temperature increase. We also note that the larger deviation of both GNN and RF for the high-density case can be attributed to the difficulty in reaching complete equilibrium in the high density as the timescale of relaxation of simulation is large. Overall, the results of the high-density case prove the abilities of GNN in more complex environments.

6.4 Conclusion

In this study, we train a graph neural network model to classify different phases of water in bulk, interfacial, and confined environments. To address the issue with the definition of order parameters in the confined environments, we train the model to learn features from the positional data *i.e.,* the distance between the oxygen atom of tagged molecules and all other water molecules oxygen atoms within a cut-off distance. We augmented the edges features with the hydrogen-boding information (acceptor-donor, donor-acceptor, or lack of hydrogen-bonding) as hydrogen atoms are coarse-grained in the graph representation. The results show successful employment of model in bulk, interfacial, and confined water inside carbon nanotube, especially in terms of its generalization compared to baseline method trained using classical order parameters model. Furthermore, the predicted melting temperature and behavior of the model in both continuous and discontinuous phase transition inside carbon nanotube were in good agreement with the change in the potential energy and dynamics of waters. In summary, the methodology and accompanied code provide a robust data-driven tool to classify and study the phase behavior of complex systems.

6.5 Figures

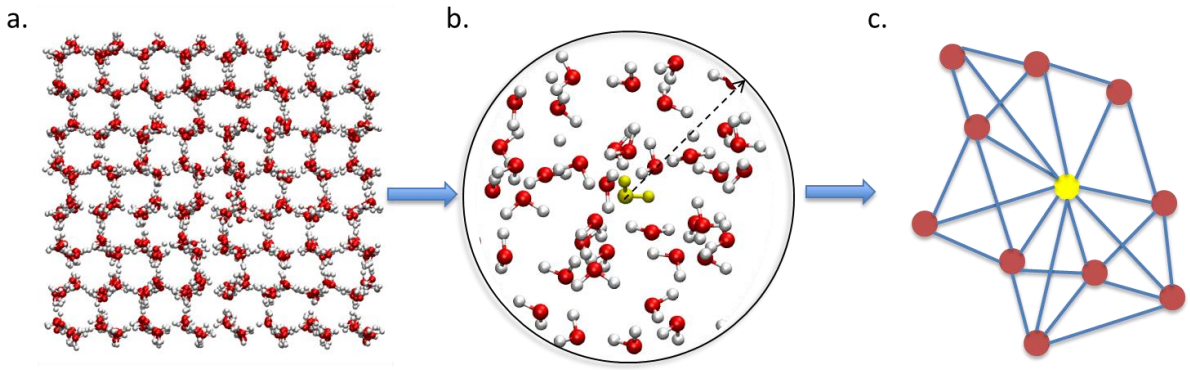


Figure 6.1. Schematic representation of water and corresponding graph structure representing the structure. a. atomistic configuration b. neighbor list formation based on a tagged water molecular c. graph representation with nodes as oxygen atoms, and edges representing connection with blue color. Each edge has four dimensions, representing distance and H-bond. Node color represents whether it is the tagged molecule or not.

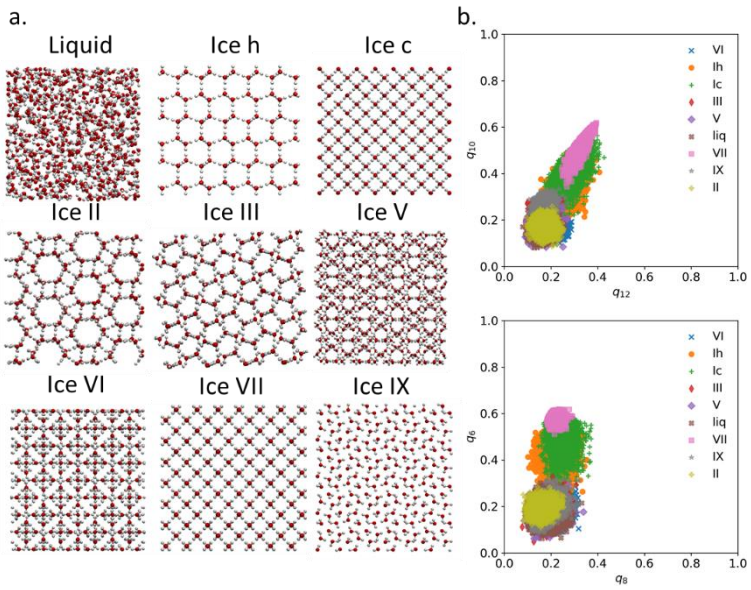


Figure 6.2. Explanatory data of analysis of bulk water phases. a. Schematic representation of different bulk water phases b. scatter plot of (q_{10}, q_{12}) and (q_6, q_8) of bulk phases of water.

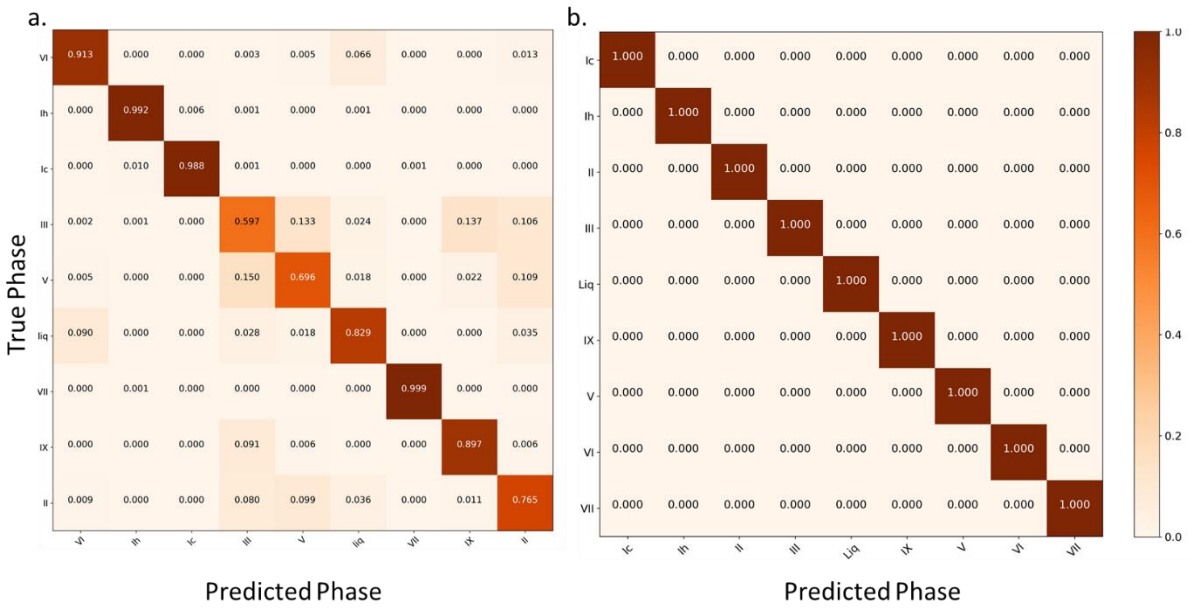


Figure 6.3. Confusion matrix for classification of bulk water. a. using RF. b. using GNN.

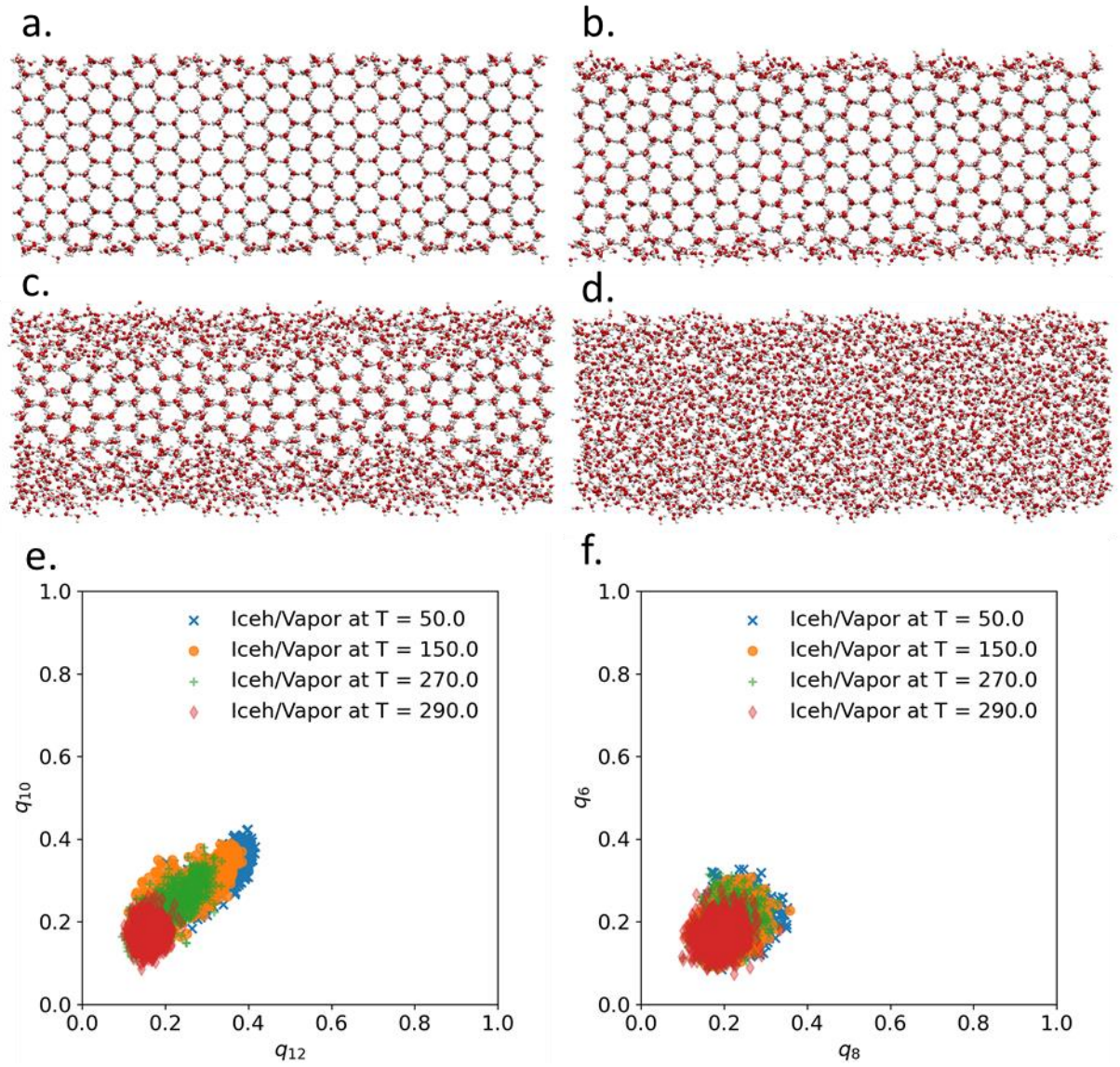


Figure 6.4. Schematic representation of Ice h/Vapor system at different temperature along with explanatory data of analysis. a. configuration at 50 K b. configuration at 150 K. c. configuration at 270 K d. configuration at 290 K e. scatter plot of (q_{10}, q_{12}) at different temperatures f. scatter plot of (q_6, q_8) at different temperatures. Distribution of order parameter pairs show lots of overlap for different temperature, which are at different temperatures.

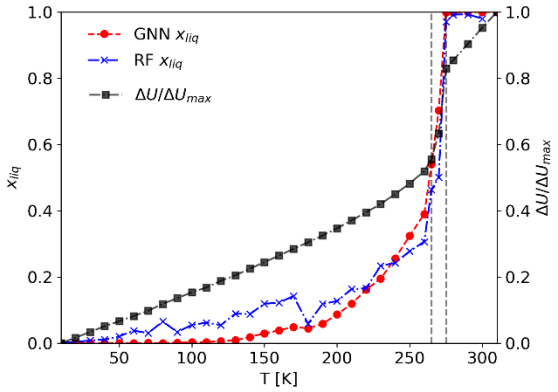


Figure 6.5. Phase transition of Ice h/vapor system. The black line with squares shows scaled potential energy of water at different temperatures. The red circle and blue cross represent fraction of liquid-like molecules at different temperatures obtained using RF and GNN, respectively. Dashed vertical lines indicate temperature range at which all Ice-like molecules disappear by temperature increase.

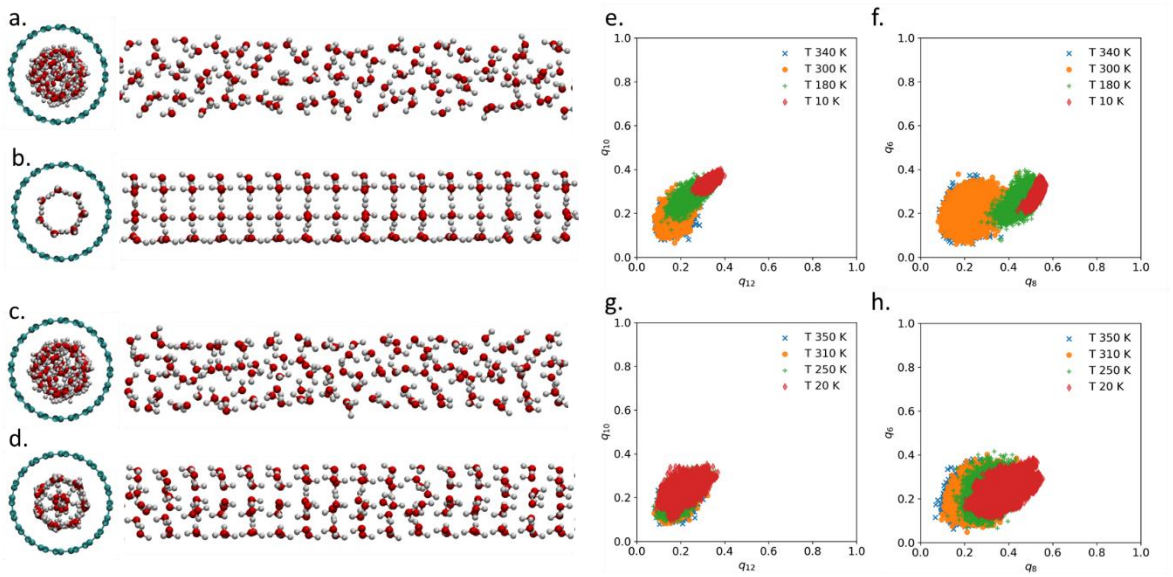


Figure 6.6. Schematic representation of confined water inside CNT 10x10 system at different temperat ures and densities along with explanatory data of analysis. a. liquid water configuration at 350 K and 1 6.75 nm^{-3}. b. solid water configuration at 20 K and 16.75 nm^{-3}. c. liquid water configuration at 350 K and 19.14 nm^{-3}. d. solid water configuration at 20 K and 19.14 nm^{-3}. e. scatter plot of (q_{10}, q_{12}) at different temperatures and density of 16.75 nm^{-3}. f. scatter plot of (q_6, q_8) at different temperatures an d density of 16.75 nm^{-3}. g. scatter plot of (q_{10}, q_{12}) at different temperatures and density of 19.14 $n m^{-3}$. h. scatter plot of (q_6, q_8) at different temperatures and density of 19.14 nm^{-3}. Distribution of or der parameter pairs show lots of overlap for different phases.

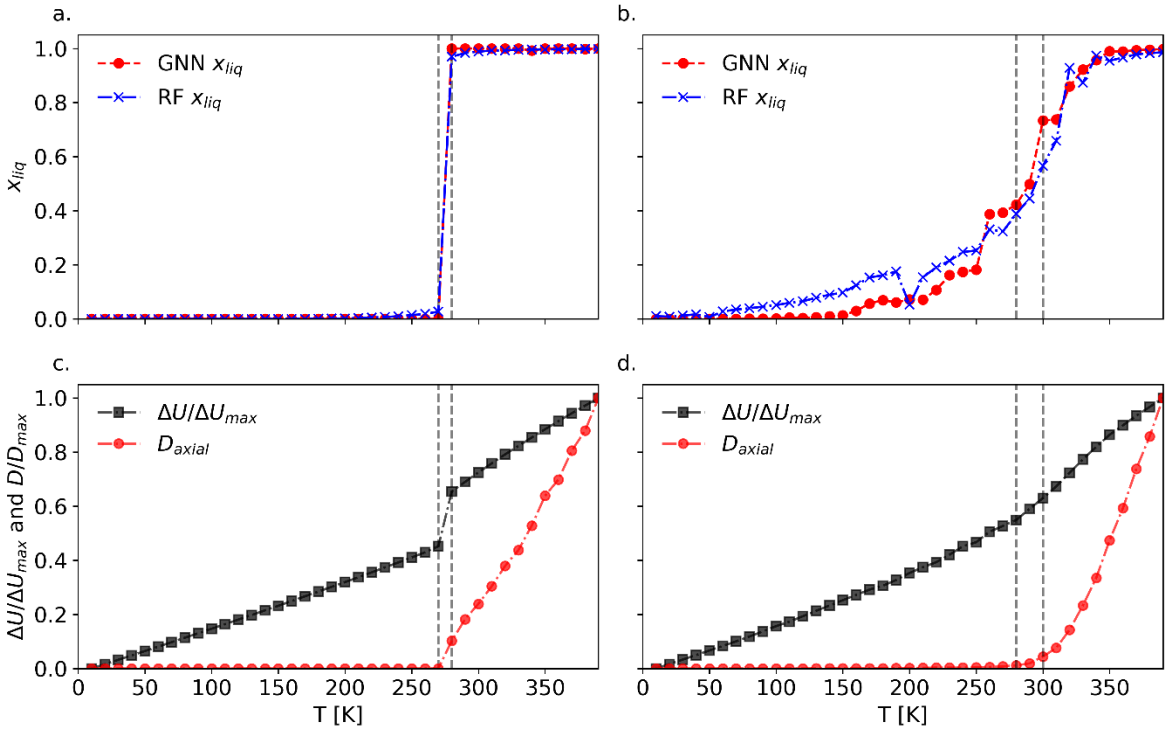


Figure 6.7. Phase transition of confined water with discontinuous and continuous phase transition. Comparison between the normalized potential energy and diffusion coefficient change at different temperatures and two densities.

CHAPTER 7: Conclusion

In conclusion, we studied simple liquids and water through statistical and deep learning approaches, providing routes to solve liquid-state theory, develop coarse-grained force fields, enhance sampling, provide insight into quasi-universal behavior of liquids, and identify phase of water. To do so, we employed extensive MD simulation to obtain big data, and train deep neural network to deduce properties and potential parameters of liquids. In addition, the quasi-universal properties of liquids are studies using statistical and deep learning method providing insight into hidden scale behavior of liquids. Followed by investigation into role of water-like behavior in coarse-grained model and phase identification of water from pairwise distance matrix.

In the second chapter, we showed that deep neural network models can learn the solution to inverse liquid-state theory, which relates structure to the potential parameters. Followed by application of the deep learning model in the coarse-graining, where we observed performance close to conventional coarse-graining methods with no need for repetitive MD simulations *i.e.*, a single-shot method. In the third chapter, we showed applications of deep denoising autoencoder in prediction of liquid structural properties from a single snapshot structure, which matches with the accuracy of several hundreds of frame average. The model also outperformed other baseline machine learning methods such as principal component and independent component methods by huge margin. The model also successfully predicts other thermodynamics properties. In the fourth chapter, we employed statistical learning methods to explain statistical similarity among simple liquids with different pair potential forms. We applied the DeepILST framework developed in the second chapter to obtain hidden scale physics properties. Additionally, mapping among different simple liquid pair potential parameters is provided. The success of the model promotes its application to the cases where analytical solution is still obscure.

In the fifth chapter, we employed local-search method to train neural network-based coarse-grained force field targeting structural properties of all-atom and *ab initi*o water model. Unlike previous methods, our

method does not accurate force and energies. The water-like behavior is studies in the single bead coarse-grained water model with two- and three-body interactions, which showed the double well potential in water vanishes when a three-body interaction is present. This finding indicates the role of directionality in water-like behavior. In the sixth chapter, we investigated the possibility to use modern graph neural network architectures to classify different phases of water without any feature design part. The model reaches high accuracy without any need to calculate traditional order parameters, and outperform baseline ML method, random forest, in the calculation of melting temperature.

References

(1) Bøhling, L.; Veldhorst, A. A.; Ingebrigtsen, T. S.; Bailey, N. P.; Hansen, J. S.; Toxvaerd, S.; Schrøder, T. B.; Dyre, J. C. Do the Repulsive and Attractive Pair Forces Play Separate Roles for the Physics of Liquids? *J. Phys. Condens. Matter* **2013**, *25* (3), 032101. https://doi.org/10.1088/0953-8984/25/3/032101.

(2) Bacher, A. K.; Schrøder, T. B.; Dyre, J. C. Explaining Why Simple Liquids Are Quasi-Universal. *Nat. Commun.* **2014**, *5* (1), 5424. https://doi.org/10.1038/ncomms6424.

(3) Vatamanu, J.; Vatamanu, M.; Bedrov, D. Non-Faradaic Energy Storage by Room Temperature Ionic Liquids in Nanoporous Electrodes. *ACS Nano* **2015**, *9* (6), 5999–6017. https://doi.org/10.1021/acsnano.5b00945.

(4) Kondrat, S.; Wu, P.; Qiao, R.; Kornyshev, A. A. Accelerating Charging Dynamics in Subnanometre Pores. *Nat. Mater.* **2014**, *13* (4), 387–393. https://doi.org/10.1038/nmat3916.

(5) Faucher, S.; Aluru, N.; Bazant, M. Z.; Blankschtein, D.; Brozena, A. H.; Cumings, J.; Pedro De Souza, J.; Elimelech, M.; Epsztein, R.; Fourkas, J. T.; Rajan, A. G.; Kulik, H. J.; Levy, A.; Majumdar, A.; Martin, C.; McEldrew, M.; Misra, R. P.; Noy, A.; Pham, T. A.; Reed, M.; Schwegler, E.; Siwy, Z.; Wang, Y.; Strano, M. Critical Knowledge Gaps in Mass Transport through Single-Digit Nanopores: A Review and Perspective. *J. Phys. Chem. C* **2019**, *123* (35), 21309–21326. https://doi.org/10.1021/acs.jpcc.9b02178.

(6) Capozza, R.; Benassi, A.; Vanossi, A.; Tosatti, E. Electrical Charging Effects on the Sliding Friction of a Model Nano-Confined Ionic Liquid. *J. Chem. Phys.* **2015**, *143* (14). https://doi.org/10.1063/1.4933010.

(7) Perkin, S. Ionic Liquids in Confined Geometries. *Phys. Chem. Chem. Phys.* **2012**, *14* (15), 5052. https://doi.org/10.1039/c2cp23814d.

(8) Chen, X.; Weber, I.; Harrison, R. W. Hydration Water and Bulk Water in Proteins Have Distinct Properties in Radial Distributions Calculated from 105 Atomic Resolution Crystal Structures. *J. Phys. Chem. B* **2008**, *112* (38), 12073–12080. https://doi.org/10.1021/jp802795a.

(9) Bellissent-Funel, M.-C.; Hassanali, A.; Havenith, M.; Henchman, R.; Pohl, P.; Sterpone, F.; van der Spoel, D.; Xu, Y.; Garcia, A. E. Water Determines the Structure and Dynamics of Proteins. *Chem. Rev.* **2016**, *116* (13), 7673–7697. https://doi.org/10.1021/acs.chemrev.5b00664.

(10) Bacher, A. K.; Schrøder, T. B.; Dyre, J. C. Explaining Why Simple Liquids Are Quasi-Universal. *Nat. Commun.* **2014**, *5* (1), 5424. https://doi.org/10.1038/ncomms6424.

(11) P, S.; J, K. Computer Simulation of Liquids. *J. Mol. Liq.* **1988**, *38* (3–4), 267. https://doi.org/10.1016/0167-7322(88)80022-9.

(12) Wang, H.; Stillinger, F. H.; Torquato, S. Sensitivity of Pair Statistics on Pair Potentials in Many-Body Systems. *J. Chem. Phys.* **2020**, *153* (12), 124106. https://doi.org/10.1063/5.0021475.

(13) Wang, Y.-L.; Lyubartsev, A.; Lu, Z.-Y.; Laaksonen, A. Multiscale Coarse-Grained Simulations of Ionic Liquids: Comparison of Three Approaches to Derive Effective Potentials. *Phys. Chem. Chem. Phys.* **2013**, *15* (20), 7701. https://doi.org/10.1039/c3cp44108c.

(14) Chaimovich, A.; Shell, M. S. Relative Entropy as a Universal Metric for Multiscale Errors. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **2010**, *81* (6), 1–4. https://doi.org/10.1103/PhysRevE.81.060104.

(15) Noid, W. G.; Liu, P.; Wang, Y.; Chu, J.-W.; Ayton, G. S.; Izvekov, S.; Andersen, H. C.; Voth, G. A. The Multiscale Coarse-Graining Method. II. Numerical Implementation for Coarse-Grained Molecular Models. *J. Chem. Phys.* **2008**, *128* (24), 244115. https://doi.org/10.1063/1.2938857.

(16) Henderson, R. L. L. A Uniqueness Theorem for Fluid Pair Correlation Functions. *Phys. Lett. A* **1974**, *49* (3), 197–198. https://doi.org/10.1016/0375-9601(74)90847-0.

(17) Hansen, J. P.; McDonald, I. R. *Theory of Simple Liquids: With Applications to Soft Matter: Fourth Edition*; Elsevier Ltd, 2013. https://doi.org/10.1016/C2010-0-66723-X.

(18) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *J. Phys. Chem. B* **2007**, *111* (27), 7812–7824. https://doi.org/10.1021/jp071097f.

(19) Monticelli, L.; Kandasamy, S. K.; Periole, X.; Larson, R. G.; Tieleman, D. P.; Marrink, S.-J. The MARTINI Coarse-Grained Force Field: Extension to Proteins. *J. Chem. Theory Comput.* **2008**, *4* (5), 819–834. https://doi.org/10.1021/ct700324x.

(20) Qiang Shi; Sergei Izvekov, and; Voth*, G. A. Mixed Atomistic and Coarse-Grained Molecular Dynamics: Simulation of a Membrane-Bound Ion Channel. **2006**. https://doi.org/10.1021/JP062700H.

(21) Galliéro, G.; Boned, C.; Baylaucq, A.; Montel, F. Molecular Dynamics Comparative Study of Lennard-Jones α -6 and Exponential α -6 Potentials: Application to Real Simple Fluids (Viscosity and Pressure). *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.* **2006**, *73* (6), 061201. https://doi.org/10.1103/PhysRevE.73.061201.

(22) Hospital, A.; Goñi, J. R.; Orozco, M.; Gelpí, J. L. Molecular Dynamics Simulations: Advances and Applications. *Adv. Appl. Bioinform. Chem.* **2015**, *8*, 37–47. https://doi.org/10.2147/AABC.S70333.

(23) Stroet, M.; Koziara, K. B.; Malde, A. K.; Mark, A. E. Optimization of Empirical Force Fields by Parameter Space Mapping: A Single-Step Perturbation Approach. *J. Chem. Theory Comput.* **2017**, *13* (12), 6201–6212. https://doi.org/10.1021/acs.jctc.7b00800.

(24) Reatto, L.; Levesque, D.; Weis, J. J. Iterative Predictor-Corrector Method for Extraction of the Pair Interaction from Structural Data for Dense Classical Liquids. *Phys. Rev. A* **1986**, *33* (5), 3451–3465. https://doi.org/10.1103/PhysRevA.33.3451.

(25) W. G. Noid; Jhih-Wei Chu; Gary S. Ayton, and; Voth*, G. A. Multiscale Coarse-Graining and Structural Correlations: Connections to Liquid-State Theory. **2007**. https://doi.org/10.1021/JP068549T.

(26) D'Alessandro, M.; Cilloco, F. Information-Theory-Based Solution of the Inverse Problem in Classical Statistical Mechanics. *Phys. Rev. E* **2010**, *82* (2), 021128. https://doi.org/10.1103/PhysRevE.82.021128.

(27) Bereau, T.; Rudzinski, J. F. Accurate Structure-Based Coarse Graining Leads to Consistent Barrier-Crossing Dynamics. *Phys. Rev. Lett.* **2018**, *121* (25), 256002.

https://doi.org/10.1103/PhysRevLett.121.256002.

(28) Hansen-Goos, H.; Mecke, K. Fundamental Measure Theory for Inhomogeneous Fluids of Nonspherical Hard Particles. *Phys. Rev. Lett.* **2009**, *102* (1), 018302. https://doi.org/10.1103/PhysRevLett.102.018302.

(29) Tang, Y.; Wu, J. Modeling Inhomogeneous van Der Waals Fluids Using an Analytical Direct Correlation Function. *Phys. Rev. E* **2004**, *70* (1), 011201. https://doi.org/10.1103/PhysRevE.70.011201.

(30) Rogers, F. J.; Young, D. A. New, Thermodynamically Consistent, Integral Equation for Simple Fluids. *Phys. Rev. A* **1984**, *30* (2), 999–1007. https://doi.org/10.1103/PhysRevA.30.999.

(31) Reith, D.; Pütz, M.; Müller-Plathe, F. Deriving Effective Mesoscale Potentials from Atomistic Simulations. *J. Comput. Chem.* **2003**, *24* (13), 1624–1636. https://doi.org/10.1002/jcc.10307.

(32) D'Alessandro, M. Maximum-Entropy Monte Carlo Method for the Inversion of the Structure Factor in Simple Classical Systems. *Phys. Rev. E* **2011**, *84* (4), 041130. https://doi.org/10.1103/PhysRevE.84.041130.

(33) Moradzadeh, A.; Motevaselian, M. H.; Mashayak, S. Y.; Aluru, N. R. Coarse-Grained Force Field for Imidazolium-Based Ionic Liquids. *J. Chem. Theory Comput.* **2018**, *14* (6), 3252–3261. https://doi.org/10.1021/acs.jctc.7b01293.

(34) Byers, J. The Physics of Data. *Nat. Phys. 2017 138* **2017**.

(35) Kulakova, L.; Arampatzis, G.; Angelikopoulos, P.; Hadjidoukas, P.; Papadimitriou, C.; Koumoutsakos, P. Data Driven Inference for the Repulsive Exponent of the Lennard-Jones Potential in Molecular Dynamics Simulations. *Sci. Rep.* **2017**, *7* (1). https://doi.org/10.1038/s41598-017-16314-4.

(36) Bartók, A. P.; De, S.; Poelking, C.; Bernstein, N.; Kermode, J. R.; Csányi, G.; Ceriotti, M. Machine Learning Unifies the Modeling of Materials and Molecules. *Sci. Adv.* **2017**, *3* (12), e1701816. https://doi.org/10.1126/sciadv.1701816.

(37) Krumscheid, S.; Pradas, M.; Pavliotis, G. A.; Kalliadasis, S. Data-Driven Coarse Graining in Action: Modeling and Prediction of Complex Systems. *Phys. Rev. E* **2015**, *92* (4), 042139. https://doi.org/10.1103/PhysRevE.92.042139.

(38) LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521* (7553), 436–444. https://doi.org/10.1038/nature14539.

(39) Tabor, D. P.; Roch, L. M.; Saikin, S. K.; Kreisbeck, C.; Sheberla, D.; Montoya, J. H.; Dwaraknath, S.; Aykol, M.; Ortiz, C.; Tribukait, H.; Amador-Bedolla, C.; Brabec, C. J.; Maruyama, B.; Persson, K. A.; Aspuru-Guzik, A. Accelerating the Discovery of Materials for Clean Energy in the Era of Smart Automation. *Nat. Rev. Mater.* **2018**, *3* (5), 5–20. https://doi.org/10.1038/s41578-018-0005-z.

(40) 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* (5), 1064–1071. https://doi.org/10.1021/acs.jpclett.8b00170.

(41) Putin, E.; Asadulaev, A.; Vanhaelen, Q.; Ivanenkov, Y.; Aladinskaya, A. V.; Aliper, A.; Zhavoronkov, A. Adversarial Threshold Neural Computer for Molecular *de Novo* Design. *Mol.*

Pharm. **2018**, acs.molpharmaceut.7b01137. https://doi.org/10.1021/acs.molpharmaceut.7b01137.

(42) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268–276. https://doi.org/10.1021/acscentsci.7b00572.

(43) Kim, E.; Huang, K.; Jegelka, S.; Olivetti, E. Virtual Screening of Inorganic Materials Synthesis Parameters with Deep Learning. *npj Comput. Mater.* **2017**, *3* (1), 53. https://doi.org/10.1038/s41524-017-0055-6.

(44) 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), 54. https://doi.org/10.1038/s41524-017-0056-5.

(45) Carleo, G.; Troyer, M. Solving the Quantum Many-Body Problem with Artificial Neural Networks. *Science* **2017**, *355* (6325), 602–606. https://doi.org/10.1126/science.aag2302.

(46) Nomura, Y.; Darmawan, A. S.; Yamaji, Y.; Imada, M. Restricted Boltzmann Machine Learning for Solving Strongly Correlated Quantum Systems. *Phys. Rev. B* **2017**, *96* (20), 205152. https://doi.org/10.1103/PhysRevB.96.205152.

(47) Van Nieuwenburg, E. P. L.; Liu, Y. H.; Huber, S. D. Learning Phase Transitions by Confusion. *Nat. Phys.* **2017**, *13* (5), 435–439. https://doi.org/10.1038/nphys4037.

(48) Ch'Ng, K.; Carrasquilla, J.; Melko, R. G.; Khatami, E. Machine Learning Phases of Strongly Correlated Fermions. *Phys. Rev. X* **2017**, *7* (3), 031038. https://doi.org/10.1103/PhysRevX.7.031038.

(49) Deng, D.-L.; Li, X.; Das Sarma, S. Quantum Entanglement in Neural Network States. *Phys. Rev. X* **2017**, *7* (2), 021021. https://doi.org/10.1103/PhysRevX.7.021021.

(50) Wang, H.; Yang, W. Force Field for Water Based on Neural Network. *J. Phys. Chem. Lett.* **2018**, *9* (12), 3232–3240. https://doi.org/10.1021/acs.jpclett.8b01131.

(51) Bejagam, K. K.; Singh, S.; An, Y.; Deshmukh, S. A. Machine-Learned Coarse-Grained Models. *J. Phys. Chem. Lett.* **2018**, *9* (16), 4667–4672. https://doi.org/10.1021/acs.jpclett.8b01416.

(52) Zhang, L.; Han, J.; Wang, H.; Car, R.; Weinan, W. E. DeePCG: Constructing Coarse-Grained Models via Deep Neural Networks. *J. Chem. Phys.* **2018**, *149* (3), 034101. https://doi.org/10.1063/1.5027645.

(53) James, M.; Murtola, T.; Schulz, R.; Smith, J. C.; Hess, B.; Lindahl, E. ScienceDirect GROMACS : High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. **2015**, *2*, 19–25. https://doi.org/10.1016/j.softx.2015.06.001.

(54) Malde, A. K.; Zuo, L.; Breeze, M.; Stroet, M.; Poger, D.; Nair, P. C.; Oostenbrink, C.; Mark, A. E. An Automated Force Field Topology Builder (ATB) and Repository: Version 1.0. *J. Chem. Theory Comput.* **2011**, *7* (12), 4026–4037. https://doi.org/10.1021/ct200196m.

(55) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A Smooth Particle Mesh Ewald Method. *J. Chem. Phys.* **1995**, *103* (19), 8577–8593. https://doi.org/10.1063/1.470117.

(56) Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* **1989**, *2* (5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8.

(57) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep Learning for Computational Chemistry. *J. Comput. Chem.* **2017**, *38* (16), 1291–1307. https://doi.org/10.1002/jcc.24764.

(58) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. **2014**.

(59) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

(60) Chaimovich, A.; Shell, M. S. Coarse-Graining Errors and Numerical Optimization Using a Relative Entropy Framework. *J. Chem. Phys.* **2011**, *134* (9), 094112. https://doi.org/10.1063/1.3557038.

(61) Shell, M. S. The Relative Entropy Is Fundamental to Multiscale and Inverse Thermodynamic Problems. *J. Chem. Phys.* **2008**, *129* (14), 1–7. https://doi.org/10.1063/1.2992060.

(62) Hyndman, R. J.; Koehler, A. B. Another Look at Measures of Forecast Accuracy. *Int. J. Forecast.* **2006**, *22* (4), 679–688. https://doi.org/10.1016/J.IJFORECAST.2006.03.001.

(63) Chandler, D. From 50 Years Ago, the Birth of Modern Liquid-State Science. *Annu. Rev. Phys. Chem.* **2017**, *68* (1), 19–38. https://doi.org/10.1146/annurev-physchem-052516-044941.

(64) Hansen, J.; McDonald, I. *Theory of Simple Liquids: With Applications to Soft Matter*; 2013.

(65) Masoori, M.; Greenfield, M. L. Reducing Noise in Computed Correlation Functions Using Techniques from Signal Processing. *Mol. Simul.* **2017**, *43* (18), 1485–1495. https://doi.org/10.1080/08927022.2017.1321753.

(66) Berg, B. A.; Harris, R. C. From Data to Probability Densities without Histograms. *Comput. Phys. Commun.* **2008**, *179* (6), 443–448. https://doi.org/10.1016/J.CPC.2008.03.010.

(67) Schofield, J. Optimization and Automation of the Construction of Smooth Free Energy Profiles. *J. Phys. Chem. B* **2017**, *121* (28), 6847–6859. https://doi.org/10.1021/acs.jpcb.7b04871.

(68) van Zon, R.; Schofield, J. Constructing Smooth Potentials of Mean Force, Radial Distribution Functions, and Probability Densities from Sampled Data. *J. Chem. Phys.* **2010**, *132* (15), 154110. https://doi.org/10.1063/1.3366523.

(69) Goulko, O.; Gaenko, A.; Gull, E.; Prokof'ev, N.; Svistunov, B. Implementation of the Bin Hierarchy Method for Restoring a Smooth Function from a Sampled Histogram. *Comput. Phys. Commun.* **2019**, *236*, 205–213. https://doi.org/10.1016/J.CPC.2018.09.019.

(70) Goulko, O.; Prokof'ev, N.; Svistunov, B. Restoring a Smooth Function from Its Noisy Integrals. *Phys. Rev. E* **2018**, *97* (5), 053305. https://doi.org/10.1103/PhysRevE.97.053305.

(71) Rosenberger, D.; van der Vegt, N. F. A. Relative Entropy Indicates an Ideal Concentration for Structure-Based Coarse Graining of Binary Mixtures. *Phys. Rev. E* **2019**, *99* (5), 053308. https://doi.org/10.1103/PhysRevE.99.053308.

(72) Moore, T. C.; Iacovella, C. R.; McCabe, C. Derivation of Coarse-Grained Potentials via Multistate Iterative Boltzmann Inversion. *J. Chem. Phys.* **2014**, *140* (22), 1–10.

https://doi.org/10.1063/1.4880555.

(73) Mashayak, S. Y.; Jochum, M. N.; Koschke, K.; Aluru, N. R.; Rühle, V.; Junghans, C. Relative Entropy and Optimization-Driven Coarse-Graining Methods in VOTCA. *PLoS One* **2015**, *10* (7). https://doi.org/10.1371/journal.pone.0131754.

(74) Guo, J.; Haji-Akbari, A.; Palmer, J. C. Hybrid Monte Carlo with LAMMPS. *J. Theor. Comput. Chem.* **2018**, *17* (03), 1840002. https://doi.org/10.1142/S0219633618400023.

(75) Prokhorenko, S.; Kalke, K.; Nahas, Y.; Bellaiche, L. Large Scale Hybrid Monte Carlo Simulations for Structure and Property Prediction. *npj Comput. Mater.* **2018**, *4* (1), 80. https://doi.org/10.1038/s41524-018-0137-0.

(76) Palmer, J. C.; Poole, P. H.; Sciortino, F.; Debenedetti, P. G. Advances in Computational Studies of the Liquid–Liquid Transition in Water and Water-Like Models. *Chem. Rev.* **2018**, *118* (18), 9129–9151. https://doi.org/10.1021/acs.chemrev.8b00228.

(77) Sastry, S.; Austen Angell, C. Liquid–Liquid Phase Transition in Supercooled Silicon. *Nat. Mater.* **2003**, *2* (11), 739–743. https://doi.org/10.1038/nmat994.

(78) Haji-Akbari, A.; Debenedetti, P. G. Direct Calculation of Ice Homogeneous Nucleation Rate for a Molecular Model of Water. *Proc. Natl. Acad. Sci.* **2015**, *112* (34), 10582–10588. https://doi.org/10.1073/PNAS.1509267112.

(79) Wu, D.; Wang, L.; Zhang, P. Solving Statistical Mechanics Using Variational Autoregressive Networks. *Phys. Rev. Lett.* **2019**, *122* (8), 080602. https://doi.org/10.1103/PhysRevLett.122.080602.

(80) Moradzadeh, A.; Aluru, N. R. Transfer-Learning-Based Coarse-Graining Method for Simple Fluids: Toward Deep Inverse Liquid-State Theory. *J. Phys. Chem. Lett.* **2019**, *10* (6), 1242–1250. https://doi.org/10.1021/acs.jpclett.8b03872.

(81) Xie, T.; France-Lanord, A.; Wang, Y.; Shao-Horn, Y.; Grossman, J. C. Graph Dynamical Networks for Unsupervised Learning of Atomic Scale Dynamics in Materials. *Nat. Commun.* **2019**, *10* (1), 2667. https://doi.org/10.1038/s41467-019-10663-6.

(82) Zhang, L.; Han, J.; Wang, H.; Car, R.; E, W. Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Phys. Rev. Lett.* **2018**, *120* (14), 143001. https://doi.org/10.1103/PhysRevLett.120.143001.

(83) Lamim Ribeiro, J. M.; Tiwary, P. Toward Achieving Efficient and Accurate Ligand-Protein Unbinding with Deep Learning and Molecular Dynamics through RAVE. *J. Chem. Theory Comput.* **2019**, *15* (1), 708–719. https://doi.org/10.1021/acs.jctc.8b00869.

(84) Cheng, S.; Wang, L.; Xiang, T.; Zhang, P. Tree Tensor Networks for Generative Modeling. *Phys. Rev. B* **2019**, *99* (15), 155131. https://doi.org/10.1103/PhysRevB.99.155131.

(85) Kingma, D. P.; Welling, M. Auto-Encoding Variational Bayes. **2013**.

(86) Hernández, C. X.; Wayment-Steele, H. K.; Sultan, M. M.; Husic, B. E.; Pande, V. S. Variational Encoding of Complex Dynamics. *Phys. Rev. E* **2018**, *97* (6), 062412. https://doi.org/10.1103/PhysRevE.97.062412.

(87) Chen, W.; Sidky, H.; Ferguson, A. L. Capabilities and Limitations of Time-Lagged Autoencoders

for Slow Mode Discovery in Dynamical Systems. *J. Chem. Phys.* **2019**, *151* (6), 064123. https://doi.org/10.1063/1.5112048.

(88) Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted Autoencoded Variational Bayes for Enhanced Sampling (RAVE). *J. Chem. Phys.* **2018**, *149* (7), 072301. https://doi.org/10.1063/1.5025487.

(89) Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P.-A. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th international conference on Machine learning - ICML '08*; ACM Press: New York, New York, USA, 2008; pp 1096–1103. https://doi.org/10.1145/1390156.1390294.

(90) Anderson, T. W. (Theodore W. *An Introduction to Multivariate Statistical Analysis*; Wiley-Interscience, 2003.

(91) Jolliffe, I. *Principal Component Analysis*; John Wiley & Sons, Ltd: Chichester, UK, 2005. https://doi.org/10.1002/0470013192.bsa501.

(92) Bacher, A. K.; Dyre, J. C. The Mother of All Pair Potentials. *Colloid Polym. Sci.* **2014**, *292* (8), 1971–1975. https://doi.org/10.1007/s00396-014-3290-0.

(93) Hansen, J.-P.; Verlet, L. Phase Transitions of the Lennard-Jones System. *Phys. Rev.* **1969**, *184* (1), 151–161. https://doi.org/10.1103/PhysRev.184.151.

(94) Kremer, K.; Robbins, M. O.; Grest, G. S. Phase Diagram of Yukawa Systems: Model for Charge-Stabilized Colloids. *Phys. Rev. Lett.* **1986**, *57* (21), 2694–2697. https://doi.org/10.1103/PhysRevLett.57.2694.

(95) Stillinger, F. H.; Weber, T. A. Computer Simulation of Local Order in Condensed Phases of Silicon. *Phys. Rev. B* **1985**, *31* (8), 5262–5271. https://doi.org/10.1103/PhysRevB.31.5262.

(96) PEARSON, E. S.; D'''AGOSTINO, R. B.; BOWMAN, K. O. Tests for Departure from Normality: Comparison of Powers. *Biometrika* **1977**, *64* (2), 231–246. https://doi.org/10.1093/biomet/64.2.231.

(97) Ross, B. C. Mutual Information between Discrete and Continuous Data Sets. *PLoS One* **2014**, *9* (2), e87357. https://doi.org/10.1371/journal.pone.0087357.

(98) Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating Mutual Information. *Phys. Rev. E* **2004**, *69* (6), 066138. https://doi.org/10.1103/PhysRevE.69.066138.

(99) Rutkai, G.; Thol, M.; Span, R.; Vrabec, J. How Well Does the Lennard-Jones Potential Represent the Thermodynamic Properties of Noble Gases? *Mol. Phys.* **2017**. https://doi.org/10.1080/00268976.2016.1246760.

(100) Ingebrigtsen, T. S.; Schrøder, T. B.; Dyre, J. C. What Is a Simple Liquid? *Phys. Rev. X* **2012**, *2* (1), 011011. https://doi.org/10.1103/PhysRevX.2.011011.

(101) López-Flores, L.; Ruíz-Estrada, H.; Chávez-Páez, M.; Medina-Noyola, M. Dynamic Equivalences in the Hard-Sphere Dynamic Universality Class. *Phys. Rev. E* **2013**, *88* (4), 042301. https://doi.org/10.1103/PhysRevE.88.042301.

(102) Dyre, J. C. Perspective: Excess-Entropy Scaling. *J. Chem. Phys.* **2018**, *149* (21), 210901. https://doi.org/10.1063/1.5055064.

(103) Pond, M. J.; Errington, J. R.; Truskett, T. M. Communication: Generalizing Rosenfeld's Excess-Entropy Scaling to Predict Long-Time Diffusivity in Dense Fluids of Brownian Particles: From Hard to Ultrasoft Interactions. *J. Chem. Phys.* **2011**, *134* (8), 081101. https://doi.org/10.1063/1.3559676.

(104) Rosenfeld, Y. Relation between the Transport Coefficients and the Internal Entropy of Simple Systems. *Phys. Rev. A* **1977**, *15* (6), 2545–2549. https://doi.org/10.1103/PhysRevA.15.2545.

(105) Rosenfeld, Y.; Ashcroft, N. W. Theory of Simple Classical Fluids: Universality in the Short-Range Structure. *Phys. Rev. A* **1979**, *20* (3), 1208–1235. https://doi.org/10.1103/PhysRevA.20.1208.

(106) Rosenfeld, Y. A Quasi-Universal Scaling Law for Atomic Transport in Simple Fluids. *J. Phys. Condens. Matter* **1999**, *11* (28), 5415–5427. https://doi.org/10.1088/0953-8984/11/28/303.

(107) Rosenfeld, Y.; Ashcroft, N. W. Mean-Spherical Model for Soft Potentials: The Hard Core Revealed as a Perturbation. *Phys. Rev. A* **1979**, *20* (5), 2162–2169. https://doi.org/10.1103/PhysRevA.20.2162.

(108) Dyre, J. C. Hidden Scale Invariance in Condensed Matter. *J. Phys. Chem. B* **2014**. https://doi.org/10.1021/jp501852b.

(109) Dyre, J. C. Isomorphs, Hidden Scale Invariance, and Quasiuniversality. *Phys. Rev. E* **2013**, *88* (4), 042139. https://doi.org/10.1103/PhysRevE.88.042139.

(110) Chandler, D. *Introduction to Modern Statistical Mechanics*; Oxford University Press, 1987.

(111) The Classical Equation of State of Gaseous Helium, Neon and Argon. *Proc. R. Soc. London. Ser. A. Math. Phys. Sci.* **1938**, *168* (933), 264–283. https://doi.org/10.1098/rspa.1938.0173.

(112) Pedersen, U. R.; Bailey, N. P.; Schrøder, T. B.; Dyre, J. C. Strong Pressure-Energy Correlations in van Der Waals Liquids. *Phys. Rev. Lett.* **2008**, *100* (1), 015701. https://doi.org/10.1103/PhysRevLett.100.015701.

(113) Craven, G. T.; Lubbers, N.; Barros, K.; Tretiak, S. Machine Learning Approaches for Structural and Thermodynamic Properties of a Lennard-Jones Fluid. *J. Chem. Phys.* **2020**. https://doi.org/10.1063/5.0017894.

(114) Craven, G. T.; Lubbers, N.; Barros, K.; Tretiak, S. Ex Machina Determination of Structural Correlation Functions. *J. Phys. Chem. Lett.* **2020**. https://doi.org/10.1021/acs.jpclett.0c00627.

(115) BERNAL, J. D. A Geometrical Approach to the Structure Of Liquids. *Nature* **1959**, *183* (4655), 141–147. https://doi.org/10.1038/183141a0.

(116) Haxton, T. K.; Schmiedeberg, M.; Liu, A. J. Universal Jamming Phase Diagram in the Hard-Sphere Limit. *Phys. Rev. E* **2011**, *83* (3), 031503. https://doi.org/10.1103/PhysRevE.83.031503.

(117) Weeks, J. D.; Chandler, D.; Andersen, H. C. Role of Repulsive Forces in Determining the Equilibrium Structure of Simple Liquids. *J. Chem. Phys.* **1971**, *54* (12), 5237–5247. https://doi.org/10.1063/1.1674820.

(118) Vollmayr-Lee, K.; Katsov, K.; Weeks, J. D. Using Mean Field Theory to Determine the Structure of Uniform Fluids. *J. Chem. Phys.* **2001**, *114* (1), 416–425. https://doi.org/10.1063/1.1329881.

(119) Tejero, C. F.; Daanoun, A.; Lekkerkerker, H. N. W.; Baus, M. Phase Diagrams of '"Simple"' Fluids with Extreme Pair Potentials. *Phys. Rev. Lett.* **1994**, *73* (5), 752–755. https://doi.org/10.1103/PhysRevLett.73.752.

(120) Berthier, L.; Tarjus, G. The Role of Attractive Forces in Viscous Liquids. *J. Chem. Phys.* **2011**. https://doi.org/10.1063/1.3592709.

(121) Powles, J. G.; Fowler, R. F. A Simple Property of a Simple Liquid. *Mol. Phys.* **1987**, *62* (5), 1079–1084. https://doi.org/10.1080/00268978700102791.

(122) Rickayzen, G.; Braka, A. C.; Pieprzyk, S.; Heyes, D. M. Single Particle Force Distributions in Simple Fluids. *J. Chem. Phys.* **2012**, *137* (9), 094505. https://doi.org/10.1063/1.4748103.

(123) Rickayzen, G.; Heyes, D. M. The Force Distribution Probability Function for Simple Fluids by Density Functional Theory. *J. Chem. Phys.* **2013**, *138* (8), 084509. https://doi.org/10.1063/1.4792640.

(124) Katsov, K.; Weeks, J. D. On the Mean Field Treatment of Attractive Interactions in Nonuniform Simple Fluids. *J. Phys. Chem. B* **2001**, *105* (28), 6738–6744. https://doi.org/10.1021/jp010893x.

(125) Frommer, F.; Hanke, M.; Jansen, S. A Note on the Uniqueness Result for the Inverse Henderson Problem. *J. Math. Phys.* **2019**, *60* (9), 093303. https://doi.org/10.1063/1.5112137.

(126) Wang, H.; Junghans, C.; Kremer, K. Comparative Atomistic and Coarse-Grained Study of Water: What Do We Lose by Coarse-Graining? *Eur. Phys. J. E* **2009**, *28* (2), 221–229. https://doi.org/10.1140/epje/i2008-10413-5.

(127) Wang, Y. T.; Izvekov, S.; Yan, T. Y.; Voth, G. A. Multiscale Coarse-Graining of Ionic Liquids. *J. Phys. Chem. B* **2006**, *110* (8), 3564–3575. https://doi.org/Doi 10.1021/Jp0548220.

(128) Gao, X.; Duan, L.-M. Efficient Representation of Quantum Many-Body States with Deep Neural Networks. *Nat. Commun.* **2017**, *8* (1), 662. https://doi.org/10.1038/s41467-017-00705-2.

(129) Schneider, E.; Dai, L.; Topper, R. Q.; Drechsel-Grau, C.; Tuckerman, M. E. Stochastic Neural Network Approach for Learning High-Dimensional Free Energy Surfaces. *Phys. Rev. Lett.* **2017**, *119* (15), 150601. https://doi.org/10.1103/PhysRevLett.119.150601.

(130) Moradzadeh, A.; Aluru, N. R. Molecular Dynamics Properties without the Full Trajectory: A Denoising Autoencoder Network for Properties of Simple Liquids. *J. Phys. Chem. Lett.* **2019**, 7568–7576. https://doi.org/10.1021/acs.jpclett.9b02820.

(131) Prezhdo, O. V. Advancing Physical Chemistry with Machine Learning. *J. Phys. Chem. Lett.* **2020**, 9656–9658. https://doi.org/10.1021/acs.jpclett.0c03130.

(132) Ha, M. Y.; Yoon, T. J.; Tlusty, T.; Jho, Y.; Lee, W. B. Universality, Scaling, and Collapse in Supercritical Fluids. *J. Phys. Chem. Lett.* **2020**, *11* (2), 451–455. https://doi.org/10.1021/acs.jpclett.9b03360.

(133) Devroye, L. *Non-Uniform Random Variate Generation*; Springer New York: New York, NY, 1986. https://doi.org/10.1007/978-1-4613-8643-8.

(134) Wald, A. On Cumulative Sums of Random Variables. *Ann. Math. Stat.* **1944**, *15* (3), 283–296. https://doi.org/10.1214/aoms/1177731235.

(135) Juneja, M.; Nagar, S. K. Particle Swarm Optimization Algorithm and Its Parameters: A Review. In *2016 International Conference on Control, Computing, Communication and Materials (ICCCCM)*; IEEE, 2016; pp 1–5. https://doi.org/10.1109/ICCCCM.2016.7918233.

(136) Errington, J. R.; Debenedetti, P. G.; Torquato, S. Quantification of Order in the Lennard-Jones System. *J. Chem. Phys.* **2003**, *118* (5), 2256–2263. https://doi.org/10.1063/1.1532344.

(137) Truskett, T. M.; Torquato, S.; Debenedetti, P. G. Towards a Quantification of Disorder in Materials: Distinguishing Equilibrium and Glassy Sphere Packings. *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.* **2000**, *62* (1 B), 993–1001. https://doi.org/10.1103/PhysRevE.62.993.

(138) Zhang, G.; Stillinger, F. H.; Torquato, S. The Perfect Glass Paradigm: Disordered Hyperuniform Glasses down to Absolute Zero. *Sci. Rep.* **2016**, *6* (1), 1–12. https://doi.org/10.1038/srep36963.

(139) Smit, B. Phase Diagrams of Lennard-Jones Fluids. *J. Chem. Phys.* **1992**, *96* (11), 8639–8640. https://doi.org/10.1063/1.462271.

(140) Watanabe, H.; Ito, N.; Hu, C.-K. Phase Diagram and Universality of the Lennard-Jones Gas-Liquid System. *J. Chem. Phys.* **2012**, *136* (20), 204102. https://doi.org/10.1063/1.4720089.

(141) Ingebrigtsen, T. S.; Bøhling, L.; Schrder, T. B.; Dyre, J. C. Communication: Thermodynamics of Condensed Matter with Strong Pressure-Energy Correlations. In *Journal of Chemical Physics*; American Institute of PhysicsAIP, 2012; Vol. 136, p 061102. https://doi.org/10.1063/1.3685804.

(142) Bertsekas, D.; Homer, M.; Logan, D.; SD. *Nonlinear Programming*; Belmont: Athena scientific, 1995.

(143) Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **1995**, *117* (1), 1–19. https://doi.org/10.1006/JCPH.1995.1039.

(144) Molinero, V.; Moore, E. B. Water Modeled as an Intermediate Element between Carbon and Silicon. *J. Phys. Chem. B* **2009**, *113* (13), 4008–4016. https://doi.org/10.1021/jp805227c.

(145) Mayne, C. G.; Saam, J.; Schulten, K.; Tajkhorshid, E.; Gumbart, J. C. Rapid Parameterization of Small Molecules Using the Force Field Toolkit. *J. Comput. Chem.* **2013**, *34* (32), 2757–2770. https://doi.org/10.1002/jcc.23422.

(146) Moradzadeh, A.; Aluru, N. R. Molecular Dynamics Properties without the Full Trajectory: A Denoising Autoencoder Network for Properties of Simple Liquids. *J. Phys. Chem. Lett.* **2019**, 7568–7576. https://doi.org/10.1021/acs.jpclett.9b02820.

(147) Scherer, C.; Scheid, R.; Andrienko, D.; Bereau, T. Kernel-Based Machine Learning for Efficient Simulations of Molecular Liquids. *J. Chem. Theory Comput.* **2020**, *16* (5), 3194–3204. https://doi.org/10.1021/acs.jctc.9b01256.

(148) Glielmo, A.; Sollich, P.; De Vita, A. Accurate Interatomic Force Fields via Machine Learning with Covariant Kernels. *Phys. Rev. B* **2017**, *95*. https://doi.org/10.1103/PhysRevB.95.214302.

(149) Behler, J.; Parrinello, M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Phys. Rev. Lett.* **2007**, *98* (14), 146401. https://doi.org/10.1103/PhysRevLett.98.146401.

(150) Nguyen, T. T.; Székely, E.; Imbalzano, G.; Behler, J.; Csányi, G.; Ceriotti, M.; Götz, A. W.;

Paesani, F. Comparison of Permutationally Invariant Polynomials, Neural Networks, and Gaussian Approximation Potentials in Representing Water Interactions through Many-Body Expansions. *J. Chem. Phys.* **2018**, *148* (24), 241725. https://doi.org/10.1063/1.5024577.

(151) Hernandez, A.; Balasubramanian, A.; Yuan, F.; Mason, S. A. M.; Mueller, T. Fast, Accurate, and Transferable Many-Body Interatomic Potentials by Symbolic Regression. *npj Comput. Mater.* **2019**, *5* (1), 1–11. https://doi.org/10.1038/s41524-019-0249-1.

(152) Chan, H.; Cherukara, M. J.; Narayanan, B.; Loeffler, T. D.; Benmore, C.; Gray, S. K.; Sankaranarayanan, S. K. R. S. Machine Learning Coarse Grained Models for Water. *Nat. Commun.* **2019**, *10* (1), 379. https://doi.org/10.1038/s41467-018-08222-6.

(153) Wang, J.; Olsson, S.; Wehmeyer, C.; Pérez, A.; Charron, N. E.; De Fabritiis, G.; Noé, F.; Clementi, C. Machine Learning of Coarse-Grained Molecular Dynamics Force Fields. *ACS Cent. Sci.* **2019**, *5* (5), 755–767. https://doi.org/10.1021/acscentsci.8b00913.

(154) Wang, W.; Gómez-Bombarelli, R. Coarse-Graining Auto-Encoders for Molecular Dynamics. *npj Comput. Mater.* **2019**, *5* (1), 1–9. https://doi.org/10.1038/s41524-019-0261-5.

(155) Wang, H.; Zhang, L.; Han, J.; E, W. DeePMD-Kit: A Deep Learning Package for Many-Body Potential Energy Representation and Molecular Dynamics. *Comput. Phys. Commun.* **2018**, *228*, 178–184. https://doi.org/10.1016/j.cpc.2018.03.016.

(156) Aly, A.; Guadagni, G.; Dugan, J. B. Derivative-Free Optimization of Neural Networks Using Local Search. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2019*; Institute of Electrical and Electronics Engineers Inc., 2019; pp 0293–0299. https://doi.org/10.1109/UEMCON47517.2019.8993007.

(157) Sun, J.; Ruzsinszky, A.; Perdew, J. Strongly Constrained and Appropriately Normed Semilocal Density Functional. *Phys. Rev. Lett.* **2015**, *115* (3), 036402. https://doi.org/10.1103/PhysRevLett.115.036402.

(158) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A Consistent and Accurate Ab Initio Parametrization of Density Functional Dispersion Correction (DFT-D) for the 94 Elements H-Pu. *J. Chem. Phys.* **2010**, *132* (15), 154104. https://doi.org/10.1063/1.3382344.

(159) Chen, M.; Ko, H. Y.; Remsing, R. C.; Calegari Andrade, M. F.; Santra, B.; Sun, Z.; Selloni, A.; Car, R.; Klein, M. L.; Perdew, J. P.; Wu, X. Ab Initio Theory and Modeling of Water. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114* (41), 10846–10851. https://doi.org/10.1073/pnas.1712499114.

(160) Sun, J.; Remsing, R. C.; Zhang, Y.; Sun, Z.; Ruzsinszky, A.; Peng, H.; Yang, Z.; Paul, A.; Waghmare, U.; Wu, X.; Klein, M. L.; Perdew, J. P. Accurate First-Principles Structures and Energies of Diversely Bonded Systems from an Efficient Density Functional. *Nat. Chem.* **2016**, *8* (9), 831–836. https://doi.org/10.1038/nchem.2535.

(161) Motevaselian, M. H.; Mashayak, S. Y.; Aluru, N. R. Extended Coarse-Grained Dipole Model for Polar Liquids: Application to Bulk and Confined Water. *Phys. Rev. E* **2018**, *98* (5), 052135. https://doi.org/10.1103/PhysRevE.98.052135.

(162) Chaimovich, A.; Shell, M. S. Anomalous Waterlike Behavior in Spherically-Symmetric Water Models Optimized with the Relative Entropy. *Phys. Chem. Chem. Phys.* **2009**, *11* (12), 1901. https://doi.org/10.1039/b818512c.

(163) Kühne, T. D.; Iannuzzi, M.; Del Ben, M.; Rybkin, V. V.; Seewald, P.; Stein, F.; Laino, T.; Khaliullin, R. Z.; Schütt, O.; Schiffmann, F.; Golze, D.; Wilhelm, J.; Chulkov, S.; Bani-Hashemian, M. H.; Weber, V.; Borštnik, U.; Taillefumier, M.; Jakobovits, A. S.; Lazzaro, A.; Pabst, H.; Müller, T.; Schade, R.; Guidon, M.; Andermatt, S.; Holmberg, N.; Schenter, G. K.; Hehn, A.; Bussy, A.; Belleflamme, F.; Tabacchi, G.; Glöß, A.; Lass, M.; Bethune, I.; Mundy, C. J.; Plessl, C.; Watkins, M.; VandeVondele, J.; Krack, M.; Hutter, J. CP2K: An Electronic Structure and Molecular Dynamics Software Package -Quickstep: Efficient and Accurate Electronic Structure Calculations. *Journal of Chemical Physics*. American Institute of Physics Inc. May 21, 2020, p 194103. https://doi.org/10.1063/5.0007045.

(164) VandeVondele, J.; Hutter, J. Gaussian Basis Sets for Accurate Calculations on Molecular Systems in Gas and Condensed Phases. *J. Chem. Phys.* **2007**, *127* (11), 114105. https://doi.org/10.1063/1.2770708.

(165) Krack, M. Pseudopotentials for H to Kr Optimized for Gradient-Corrected Exchange-Correlation Functionals. *Theor. Chem. Acc.* **2005**, *114* (1–3), 145–152. https://doi.org/10.1007/s00214-005-0655-y.

(166) Lehtola, S.; Steigemann, C.; Oliveira, M. J. T.; Marques, M. A. L. Recent Developments in LIBXC — A Comprehensive Library of Functionals for Density Functional Theory. *SoftwareX* **2018**, *7*, 1–5. https://doi.org/10.1016/j.softx.2017.11.002.

(167) Nosé, S. A Unified Formulation of the Constant Temperature Molecular Dynamics Methods. *J. Chem. Phys.* **1984**, *81* (1), 511. https://doi.org/10.1063/1.447334.

(168) Al-Sahaf, H.; Bi, Y.; Chen, Q.; Lensen, A.; Mei, Y.; Sun, Y.; Tran, B.; Xue, B.; Zhang, M. A Survey on Evolutionary Machine Learning. *Journal of the Royal Society of New Zealand*. Taylor and Francis Asia Pacific April 3, 2019, pp 205–228. https://doi.org/10.1080/03036758.2019.1609052.

(169) Drugan, M. M. Reinforcement Learning versus Evolutionary Computation: A Survey on Hybrid Algorithms. *Swarm Evol. Comput.* **2019**, *44*, 228–246. https://doi.org/10.1016/j.swevo.2018.03.011.

(170) Moradzadeh, A.; Motevaselian, M. H.; Mashayak, S. Y.; Aluru, N. R. Coarse-Grained Force Field for Imidazolium-Based Ionic Liquids. *J. Chem. Theory Comput.* **2018**, *14* (6), 3252–3261. https://doi.org/10.1021/acs.jctc.7b01293.

(171) Cho, C. H.; Singh, S.; Robinson, G. W. Liquid Water and Biological Systems: The Most Important Problem in Science That Hardly Anyone Wants to See Solved. *Faraday Discuss.* **1996**, *103* (0), 19–27. https://doi.org/10.1039/fd9960300019.

(172) Barros De Oliveira, A.; Neves, E. B.; Gavazzoni, C.; Paukowski, J. Z.; Netz, P. A.; Barbosa, M. C. Liquid Crystal Phase and Waterlike Anomalies in a Core-Softened Shoulder-Dumbbells System. *J. Chem. Phys.* **2010**, *132* (16), 164505. https://doi.org/10.1063/1.3386384.

(173) Monroe, J. I.; Shell, M. S. Decoding Signatures of Structure, Bulk Thermodynamics, and Solvation in Three-Body Angle Distributions of Rigid Water Models. *J. Chem. Phys.* **2019**, *151* (9), 094501. https://doi.org/10.1063/1.5111545.

(174) Chen, B.; Ivanov, I.; Klein, M. L.; Parrinello, M. Hydrogen Bonding in Water. *Phys. Rev. Lett.* **2003**, *91* (21), 215503. https://doi.org/10.1103/PhysRevLett.91.215503.

(175) Starr, F. W.; Nielsen, J. K.; Stanley, H. E. Hydrogen-Bond Dynamics for the Extended Simple Point-Charge Model of Water. *Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip. Top.* **2000**, *62* (1 A), 579–587. https://doi.org/10.1103/PhysRevE.62.579.

(176) Asadi, M.; Kim, K.; Liu, C.; Addepalli, A. V.; Abbasi, P.; Yasaei, P.; Phillips, P.; Behranginia, A.; Cerrato, J. M.; Haasch, R.; Zapol, P.; Kumar, B.; Klie, R. F.; Abiade, J.; Curtiss, L. A.; Salehi-Khojin, A. Nanostructured Transition Metal Dichalcogenide Electrocatalysts for CO2 Reduction in Ionic Liquid. *Science* **2016**, *353* (6298), 467–470. https://doi.org/10.1126/science.aaf4767.

(177) Janssen, G. J. M. A Phenomenological Model of Water Transport in a Proton Exchange Membrane Fuel Cell. *J. Electrochem. Soc.* **2001**, *148* (12), A1313. https://doi.org/10.1149/1.1415031.

(178) Feng, J.; Graf, M.; Liu, K.; Ovchinnikov, D.; Dumcenco, D.; Heiranian, M.; Nandigana, V.; Aluru, N. R.; Kis, A.; Radenovic, A. Single-Layer MoS2 Nanopores as Nanopower Generators. *Nature* **2016**, *536* (7615), 197–200. https://doi.org/10.1038/nature18593.

(179) Heiranian, M.; Farimani, A. B.; Aluru, N. R. Water Desalination with a Single-Layer MoS2 Nanopore. *Nat. Commun.* **2015**, *6*, 8616. https://doi.org/10.1038/ncomms9616.

(180) Errington, J. R.; Debenedetti, P. G. Relationship between Structural Order and the Anomalies of Liquid Water. *Nature* **2001**, *409* (6818), 318–321. https://doi.org/10.1038/35053024.

(181) Lechner, W.; Dellago, C. Accurate Determination of Crystal Structures Based on Averaged Local Bond Order Parameters. *J. Chem. Phys.* **2008**, *129* (11), 114707. https://doi.org/10.1063/1.2977970.

(182) Steinhardt, P. J.; Nelson, D. R.; Ronchetti, M. Bond-Orientational Order in Liquids and Glasses. *Phys. Rev. B* **1983**, *28* (2), 784–805. https://doi.org/10.1103/PhysRevB.28.784.

(183) Duboué-, E.; Dijon, D.-; Laage, D. Characterization of the Local Structure in Liquid Water by Various Order Parameters. *J. Phys. Chem. B* **2015**, *119*, 47. https://doi.org/10.1021/acs.jpcb.5b02936.

(184) Haji-Akbari, A.; Debenedetti, P. G. Computational Investigation of Surface Freezing in a Molecular Model of Water. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114* (13), 3316–3321. https://doi.org/10.1073/pnas.1620999114.

(185) Hamm, P. Markov State Model of the Two-State Behaviour of Water. *J. Chem. Phys.* **2016**, *145* (13), 134501. https://doi.org/10.1063/1.4963305.

(186) Singh, R. S.; Biddle, J. W.; Debenedetti, P. G.; Anisimov, M. A. Two-State Thermodynamics and the Possibility of a Liquid-Liquid Phase Transition in Supercooled TIP4P/2005 Water. *J. Chem. Phys.* **2016**, *144* (14), 144504. https://doi.org/10.1063/1.4944986.

(187) Niu, H.; Yang, Y. I.; Parrinello, M. Temperature Dependence of Homogeneous Nucleation in Ice. *Phys. Rev. Lett.* **2019**, *122* (24), 245501. https://doi.org/10.1103/PhysRevLett.122.245501.

(188) Piaggi, P. M.; Parrinello, M. Calculation of Phase Diagrams in the Multithermal-Multibaric Ensemble. *J. Chem. Phys.* **2019**, *150* (24), 244119. https://doi.org/10.1063/1.5102104.

(189) Aydin, F.; Moradzadeh, A.; Bilodeau, C. L.; Lau, E. Y.; Schwegler, E.; Aluru, N. R.; Pham, T. A. Ion Solvation and Transport in Narrow Carbon Nanotubes: Effects of Polarizability, Cation-ΠInteraction, and Confinement. *J. Chem. Theory Comput.* **2021**.

https://doi.org/10.1021/acs.jctc.0c00827.

(190) Raju, M.; Van Duin, A.; Ihme, M. Phase Transitions of Ordered Ice in Graphene Nanocapillaries and Carbon Nanotubes. *Sci. Rep.* **2018**, *8* (1), 1–11. https://doi.org/10.1038/s41598-018-22201-3.

(191) Takaiwa, D.; Hatano, I.; Koga, K.; Tanaka, H. Phase Diagram of Water in Carbon Nanotubes. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105* (1), 39–43. https://doi.org/10.1073/pnas.0707917105.

(192) Algara-Siller, G.; Lehtinen, O.; Wang, F. C.; Nair, R. R.; Kaiser, U.; Wu, H. A.; Geim, A. K.; Grigorieva, I. V. Square Ice in Graphene Nanocapillaries. *Nature* **2015**, *519* (7544), 443–445. https://doi.org/10.1038/nature14295.

(193) Pugliese, P.; Conde, M. M.; Rovere, M.; Gallo, P. Freezing Temperatures, Ice Nanotubes Structures, and Proton Ordering of TIP4P/ICE Water inside Single Wall Carbon Nanotubes. *J. Phys. Chem. B* **2017**, *121* (45), 10371–10381. https://doi.org/10.1021/acs.jpcb.7b06306.

(194) Moradzadeh, A.; Aluru, N. R. Understanding Simple Liquids through Statistical and Deep Learning Approaches. *J. Chem. Phys.* **2021**, *154* (20), 204503. https://doi.org/10.1063/5.0046226.

(195) Xu, K.; Jegelka, S.; Hu, W.; Leskovec, J. How Powerful Are Graph Neural Networks? In *7th International Conference on Learning Representations, ICLR 2019*; International Conference on Learning Representations, ICLR, 2019.

(196) Simonovsky, M.; Komodakis, N. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017* **2017**, *2017-January*, 29–38.

(197) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. C. GROMACS: Fast, Flexible, and Free. *J. Comput. Chem.* **2005**, *26* (16), 1701–1718. https://doi.org/10.1002/jcc.20291.

(198) Abascal, J. L. F.; Sanz, E.; Fernández, R. G.; Vega, C. A Potential Model for the Study of Ices and Amorphous Water: TIP4P/Ice. *J. Chem. Phys.* **2005**, *122* (23), 234511. https://doi.org/10.1063/1.1931662.

(199) Wu, Y.; Aluru, N. R. Graphitic Carbon-Water Nonbonded Interaction Parameters. *J. Phys. Chem. B* **2013**, *117* (29), 8802–8813. https://doi.org/10.1021/jp402051t.

(200) Nosé, S. A Unified Formulation of the Constant Temperature Molecular Dynamics Methods. *J. Chem. Phys.* **1984**, *81* (1), 511–519. https://doi.org/10.1063/1.447334.

(201) Matsumoto, M.; Yagasaki, T.; Tanaka, H. GenIce: Hydrogen-Disordered Ice Generator. *J. Comput. Chem.* **2018**, *39* (1), 61–64. https://doi.org/10.1002/jcc.25077.

(202) Leocmach, M. Pyboo. November 26, 2017. https://doi.org/10.5281/zenodo.1066568.

(203) Zdeborová, L. Machine Learning: New Tool in the Box. *Nat. Phys.* **2017**, *13* (5), 420–421. https://doi.org/10.1038/nphys4053.

(204) Michael R. Shirts*, †; Mobley†, D. L.; and, J. D. C.; Pande, V. S. Accurate and Efficient Corrections for Missing Dispersion Interactions in Molecular Simulations. **2007**. https://doi.org/10.1021/JP0735987.

(205) Abraham, M.; Van Der Spoel, D.; Lindahl, E.; Hess, B. GROMACS User Manual Version 5.0.4.

2014, 312. https://doi.org/10.1007/SpringerReference_28001.

(206) Lee, S. H.; Rasaiah, J. C.; Hubbard, J. B. Molecular Dynamics Study of a Dipolar Fluid between Charged Plates. II. *J. Chem. Phys.* **1987**, *86* (4), 2383–2393. https://doi.org/10.1063/1.452086.

(207) Motevaselian, M. H.; Aluru, N. R. An EQT-Based CDFT Approach for Thermodynamic Properties of Confined Fluid Mixtures. *J. Chem. Phys.* **2017**, *146*, 154102. https://doi.org/10.1063/1.4979896.

(208) Geiger, B. C.; Kubin, G. Signal Enhancement as Minimization of Relevant Information Loss. In *Proceedings of 2013 9th International ITG Conference on Systems, Communication and Coding, SCC 2013*; 2013.

(209) Shen, H.; George, D.; Huerta, E. A.; Zhao, Z. Denoising Gravitational Waves Using Deep Learning with Recurrent Denoising Autoencoders. **2017**.

Appendix A: Supplementary Information for Chapter 2

A.1 Thermodynamic Properties

Given that the deep neural network (DNN) obtains the solution to the inverse-problem of liquid-state theory, we investigate the estimation of thermodynamic properties of Lennard-Jones particles using the DNN and compare them with values obtained from MD simulations of Argon case as discussed in the main manuscript (Argon at various temperatures and densities, see Figure A1 for RDFs of Argon at various thermodynamic states). For example, pressure of a given molecular system can be calculated using,

$$P = \frac{1}{V}\left(\frac{1}{3}N_{DOF}k_BT + < \frac{1}{3}\sum_{j>i} f_{ij}.r_{ij} > \right) \qquad (A.1)$$

where N_{DOF} is the number of degrees of freedom in the system, and V is the volume of the simulation box. The second term in Eq. A.1 is the virial part of the pressure and f_{ij} is the force between two beads, i and j, which can be expressed in terms of the RDF and the pair potential. The pressure can then be expressed as,

$$P = \rho k_BT - \frac{4\pi\rho^2}{6}\int_0^{r_{cut}} \frac{\partial u}{\partial r} g(r) r^3 dr \qquad (A.2)$$

In a similar manner, one can calculate the total potential energy of a system based on the radial distribution function and the pair potential, as expressed below,

$$U = \frac{N}{2} 4\pi\rho \int_0^{\infty} u(r) g(r) r^2 dr \qquad (A.3)$$

where N is the number of particles in the system. Note that during the calculation, we consider dispersion correction for the virial and energy (see Reference)[204] in a similar manner to that implemented in GROMACS.[205] The comparison between pressure and energy predicted using MD and DNN is shown in

Figure A2 and A3, respectively. The mean relative error for deep learning is less than 14% and 25% for the potential energy and pressure, respectively, with an average relative error of 13% and 17%, indicating the ability of the deep learning approach to predict the underlying physics and obtaining the solution to the inverse-problem of liquid-state theory for Lennard-Jones particles.

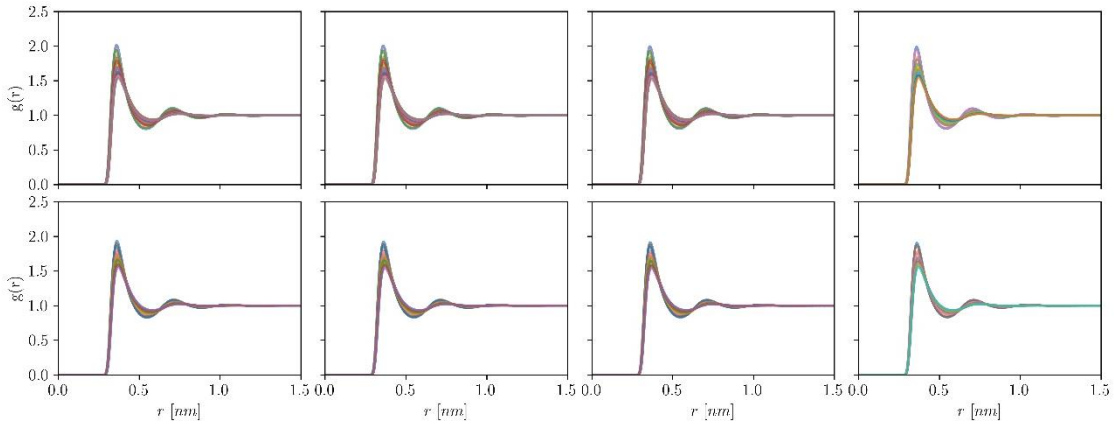


Figure A.1. Radial distribution function of Argon at 121 different thermodynamic states obtained by uniformly sampling both temperature and density range shown in Table 1 of the main manuscript.

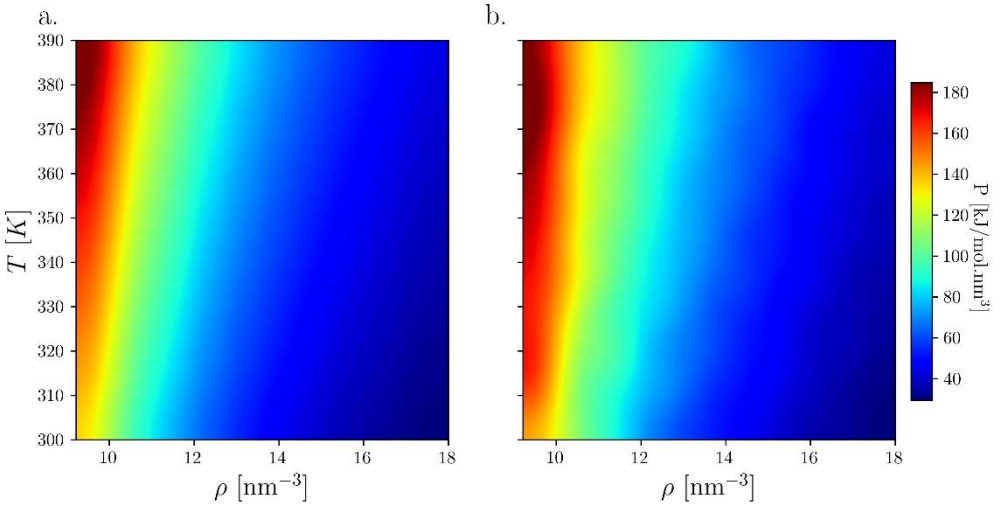


Figure A.2. Bulk pressure of Argon for different thermodynamic states. a. MD results b. deep learning results.

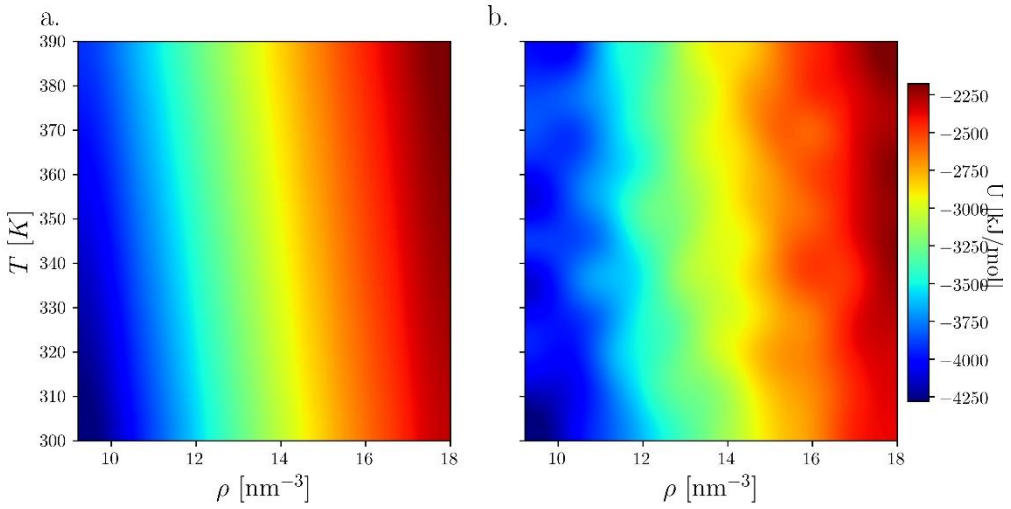


Figure A.3. Total potential energy of Argon for different thermodynamic states. a. MD results b. deep learning results

A.2 Training of Deep Learning Model

The mean-squared error is used as the loss function to train the DNN. Optimization is performed using the Adam optimizer. The dropout technique is used for the second hidden layer, which resembles the regularization algorithm in reducing the overfitting significantly (see reference for more details on the dropout technique).[59] The dropout technique randomly drops nodes in layers with a specified probability, thereby reducing the number of active nodes in that layer. Dropout technique prevents nodes from undesirable co-adapting. The key idea in dropout is that in each minimization step of DNN instead of training the whole network only a fraction of the active nodes is used in training (see reference [57] for more information regarding deep learning).

We have tested multiple networks with different activation functions and size to select the DNN architectures according to Figure 2.1c of the main manuscript. A total of 36 networks are trained to choose the best architecture with the smallest loss function and least overfitting. All the networks used similar input (material fingerprint) except for the expansion power in temperature and density. Figure A4

shows the DL network with the best performance with two hidden layers and 48x15 size and a power of 3 ($p = 3$).

We also trained networks with 3 hidden layers, but these networks had an overfitting problem, as their performance over training and validation datasets diverged, *i.e.*, it has a similar loss function value as the 2 layers network on the training dataset, but the loss function value is higher for the validation and testing datasets, indicating overfitting.

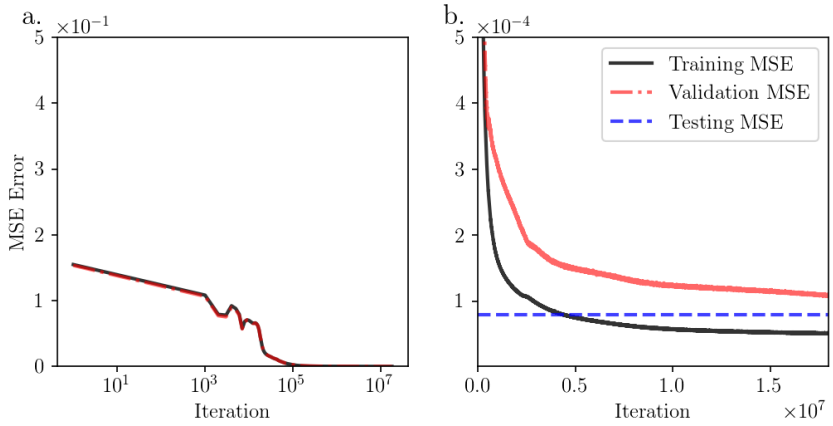


Figure A.4. The loss function value of DL network for both training (black line) and validation (red dash-dot line) datasets. The value of loss function for testing dataset (blue dash-dash line). a. initial steps of training shown in log-scale b. final steps of training.

The output of FNN with three hidden layers uses the following sequence of equations,

$$\boldsymbol{\mu}_1 = \tanh(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \text{sigmoid}(\mathbf{W}_1\boldsymbol{x} + \boldsymbol{b}_1) \quad (A.4)$$

$$\boldsymbol{\mu}_2 = \tanh(\mathbf{W}_2\boldsymbol{\mu}_1 + \boldsymbol{b}_2) + \text{sigmoid}(\mathbf{W}_2\boldsymbol{\mu}_1 + \boldsymbol{b}_2) \quad (A.5)$$

$$\boldsymbol{\mu}_3 = \tanh(\mathbf{W}_3\boldsymbol{\mu}_2 + \boldsymbol{b}_3) + \text{sigmoid}(\mathbf{W}_3\boldsymbol{\mu}_2 + \boldsymbol{b}_3) \quad (A.6)$$

$$\boldsymbol{v}_o = \text{sigmoid}(\mathbf{W}_o\boldsymbol{\mu_3} + \boldsymbol{b}_o) \quad (A.7)$$

The accuracy of DNN is determined using the equation,

$$\boldsymbol{a_{DNN,j}} = \frac{\sum_{\text{i}\in\text{D}} f_{ij}}{|\text{D}|} \quad (A.8)$$

where $\boldsymbol{a_{DNN,j}}$ is the accuracy of the DNN for the potential parameter j (C_6, C_{12}). |D| is the size of the dataset. f_{ij} is determined based on how far is the prediction from the ground truth compared to square root of the loss function value on the training dataset, i.e.,

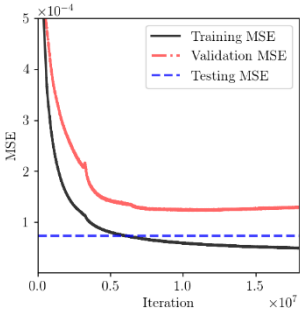
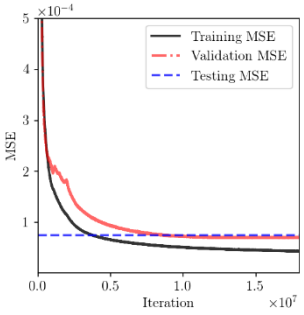
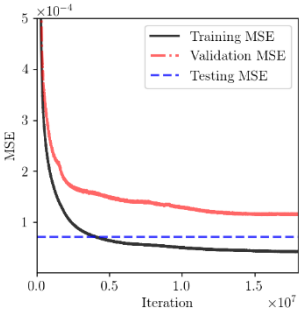
$$f_{ij} = \begin{cases} 1 & \text{if} \quad |v_{j,DNN}^{(i)} - v_{j,GT}^{(i)}| \leq 4\epsilon_{\mathcal{L}}^{0.5} \\ 0 & \text{otherwise} \end{cases} \tag{A.9}$$

where $\epsilon_{\mathcal{L}}$ is the loss function value on the training dataset, which shows the interval for which 99 percent of DNN predictions are located from one-to-one mapping line, *i.e.*, f_{ij} counts the number of data points within two parallel lines with the one-to-one mapping line, both of which have an offset of $\pm 4.0\epsilon_{\mathcal{L}}^{0.5}$.

The mean absolute percentage error[62] (MAPE) used in the main manuscript can be expressed as,

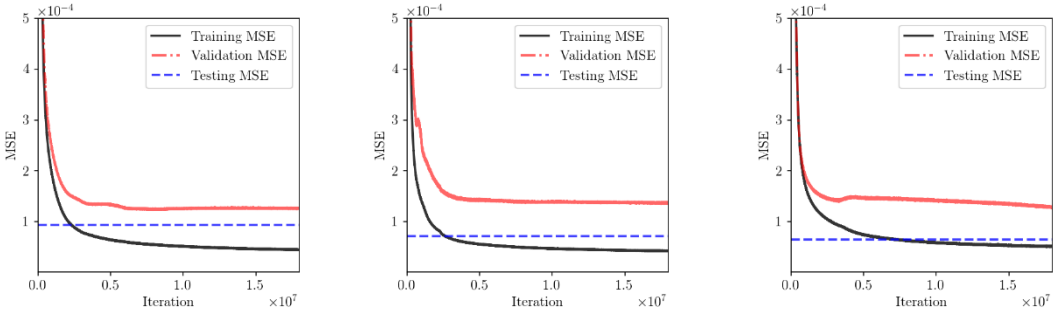
$$\epsilon_{\boldsymbol{MAPE,j}} = 100 \times \frac{\sum_{\mathrm{i}\in \mathrm{D}} \quad |v_{j,DNN}^{(i)} - v_{j,GT}^{(i)}|}{\sum_{i\in \mathrm{D}} |v_{j,GT}^{(i)}|} \tag{A.10}$$

The MSEs during training for both 2 and 3 layer networks are shown in Figure A.5 and A.6.

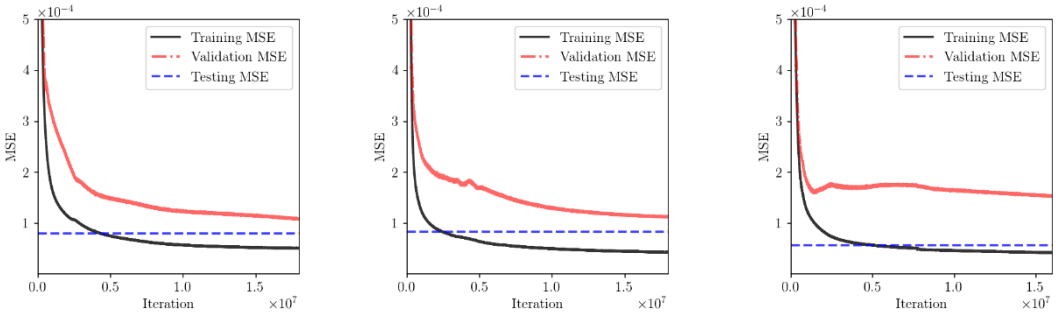
Size/p	48x15	60x15	75x15
1			

(Figure A.5 (cont.))

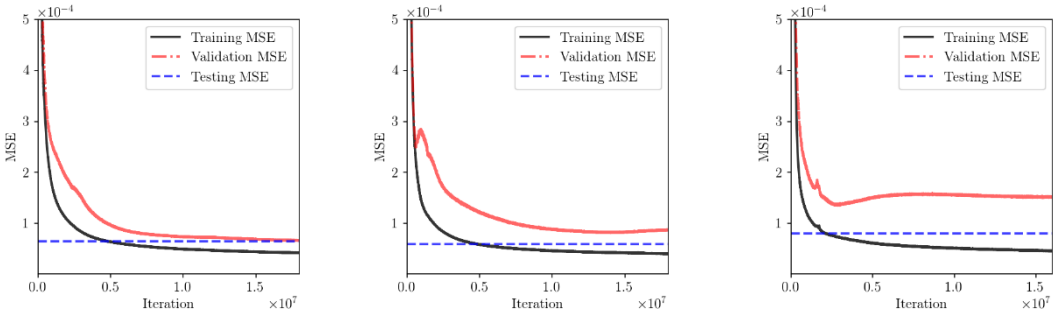
2



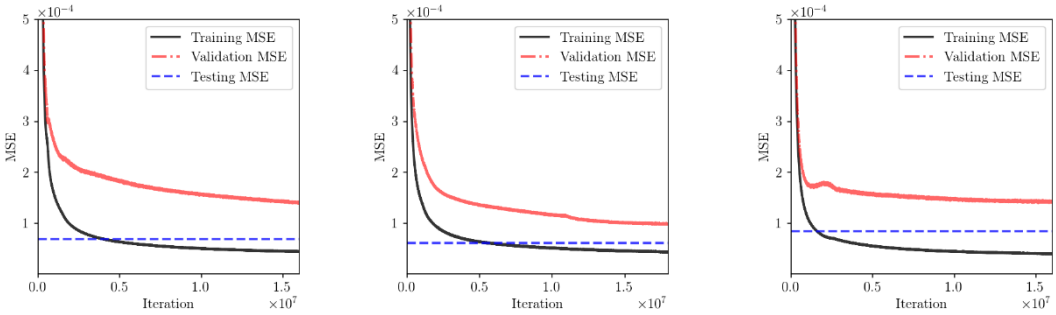
3



4



5



(Figure A.5 (cont.))

6

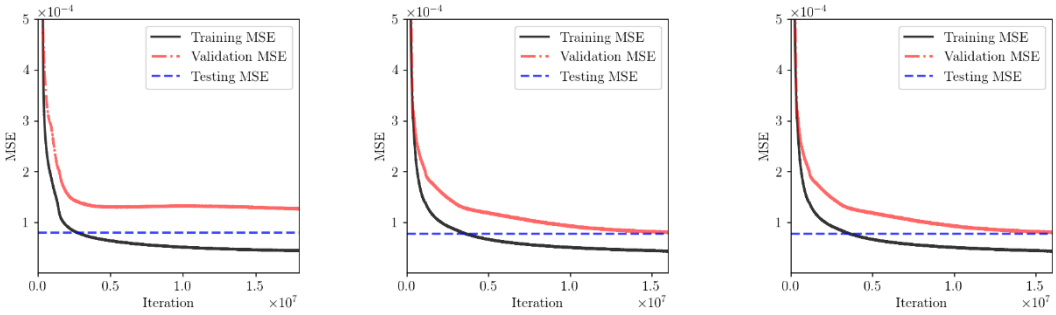
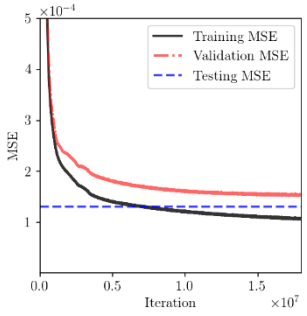
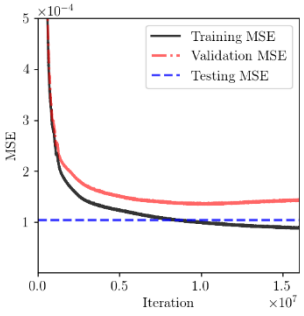
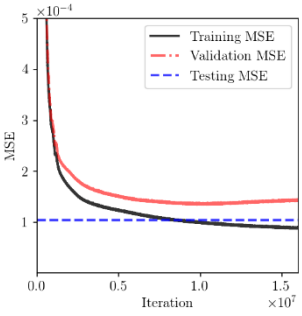
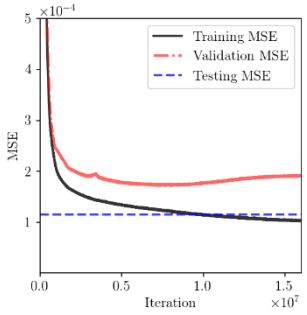
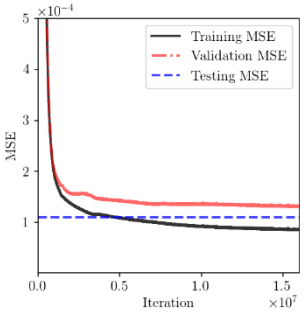
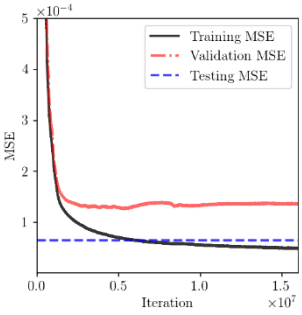


Figure A.5. Loss function during training of 2 hidden layer networks. First column shows the *p* (power of density and temperature for input) and first row shows the number of nodes in the first and second layers, respectively.

Size (1st, 2nd, and 3rd layers)	48x24x12	60x30x15	80x40x20
p			
1			
2			

(Figure A.6 (cont.))

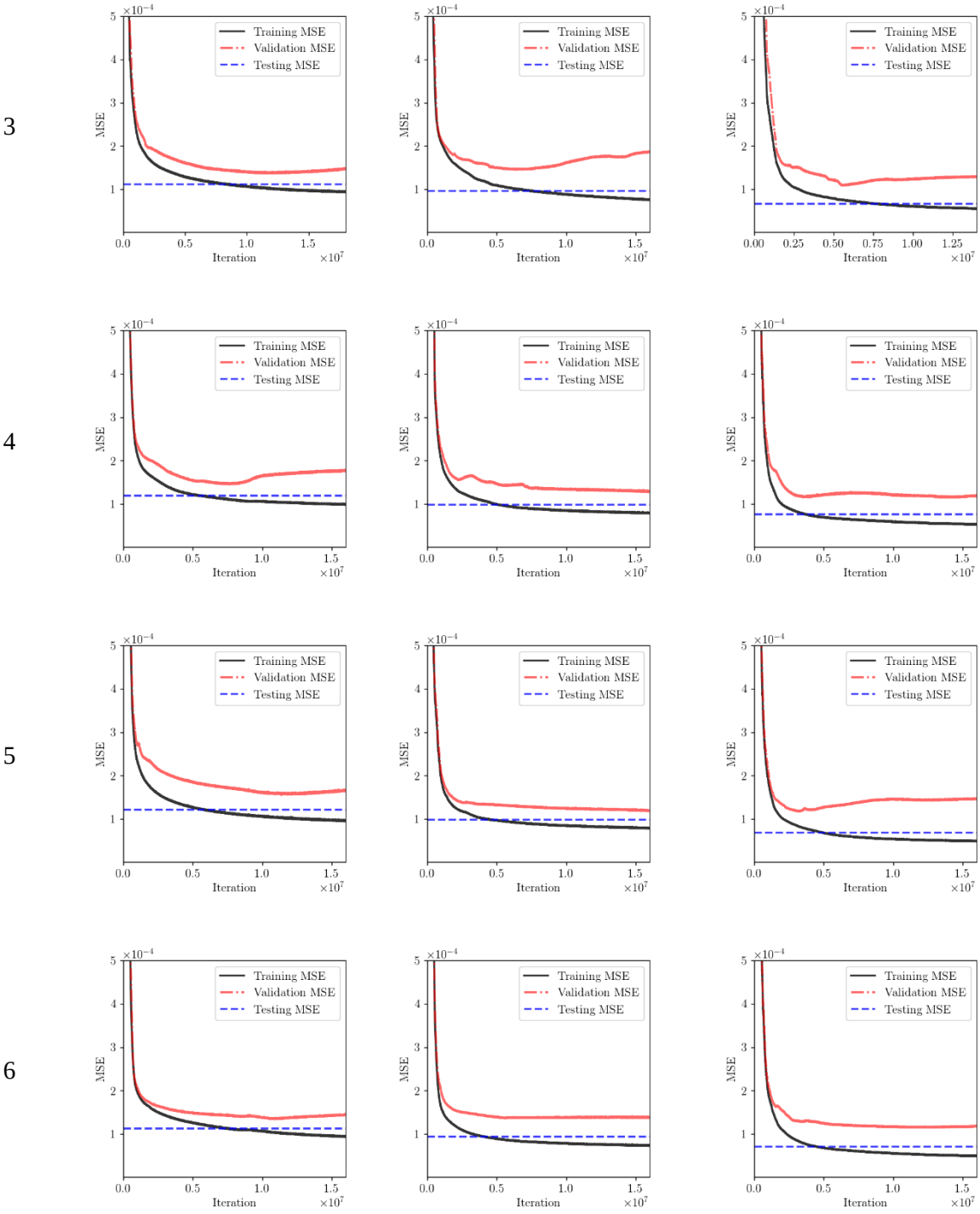
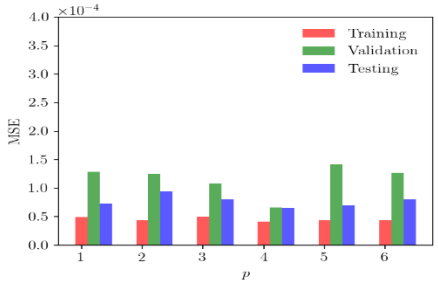


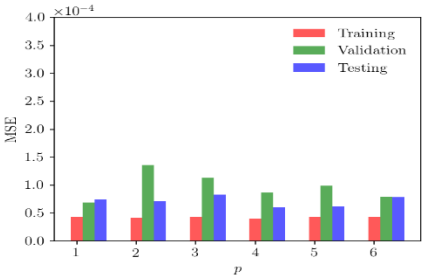
Figure A.6. Loss function during training of 3 hidden layer networks. First column shows the *p* (power of density and temperature for input) and first row shows the number of nodes in the first, second, and third layers, respectively.

Based on the above results, we have selected three 2-hidden-layer networks (60x15 with power of 1 and 6 and 48x15 with powers of 3). The performance of all these networks is similar, but the second network performs better for the transferability case. Therefore, we used it for all the results shown in the main manuscript. Figure A7 and A8 show final values of MSE on each dataset for different network sizes and different features, which, in turn, justifies the selection of the aforementioned networks.

48x15



60x15



75x15

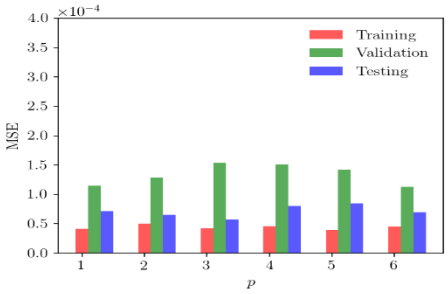
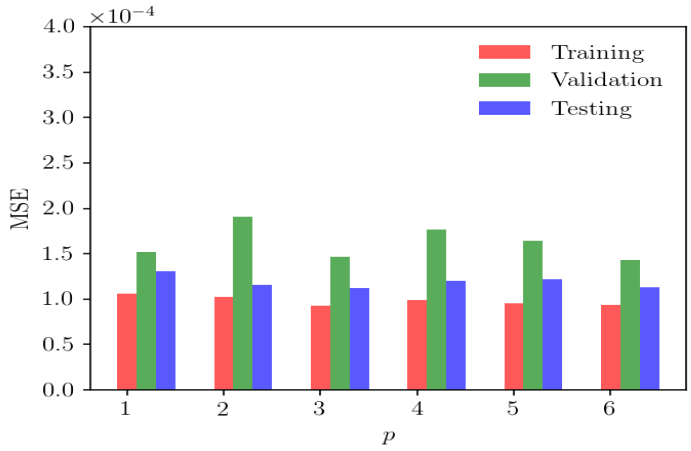
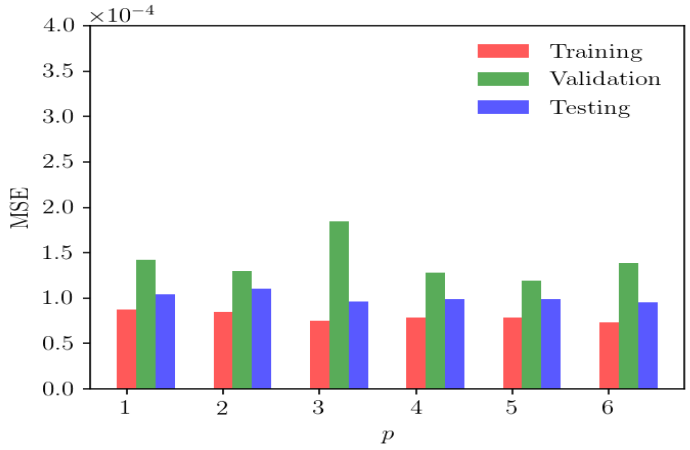


Figure A.7. Final loss function values for training, validation, and testing datasets for various exponents (p) of temperature and pressure.

48x24x12



60x30x15



80x40x20

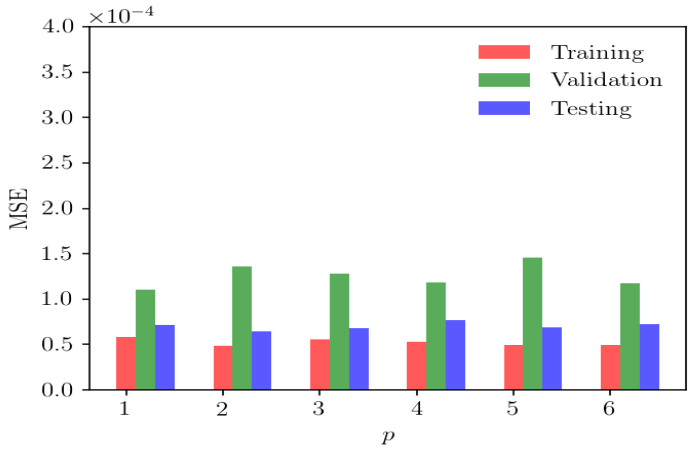
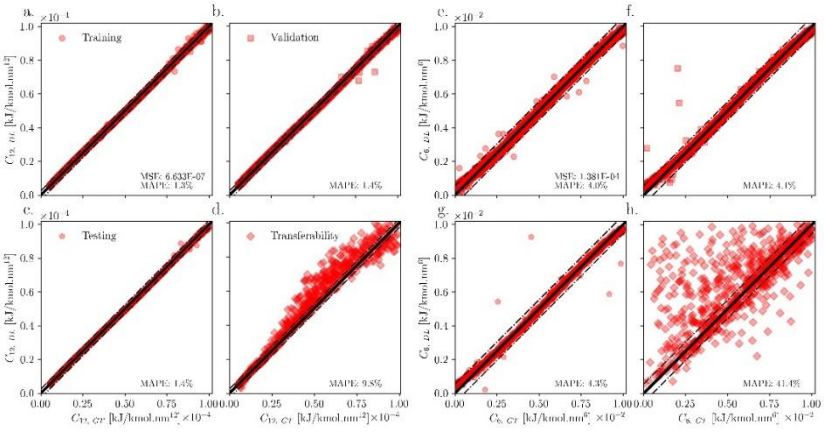


Figure A.8. Final loss function values for training, validation, and testing datasets for various exponents (p) of temperature and pressure.

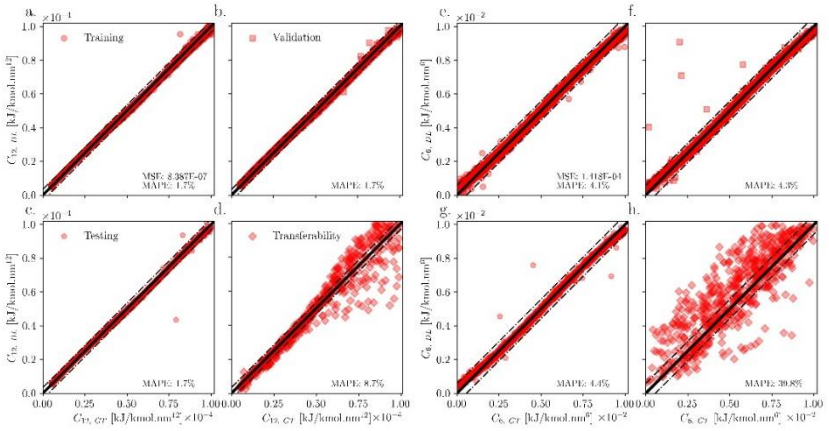
The transferability dataset includes data outside the range mentioned in Table A1 of the main manuscript. The data shown for the one-to-one mapping of the transferability case are obtained using the following steps. Based on the loss function value of the training dataset, data which are bounded by the accuracy line are selected. Then, minimum and maximum of the non-dimensional temperature ($T^* = \frac{k_B T}{C_{6,p}^2/4C_{12,p}}$, k_B is the Boltzmann constant and subscript p indicates predicted value by DL network) and density ($\rho^* = \frac{N(C_{12,p}/C_{6,p})^{\frac{1}{2}}}{L^3}$, L and N are the simulation box length and number of atoms, respectively) are determined for the selected data within two accuracy lines in order to obtain criteria to create a validity range for the non-dimensional temperature and density. Data points in the transferability dataset, which are within the validity range, are shown for each network including the results shown in Figure A.2d and A.2h of the main manuscript with non-dimensional temperature and density of $[0.42, 39.20]$ and $[0.0781, 0.802]$, respectively. The error distributions of all the three networks are shown in Figure A.9.

A.



(Figure A.9 (cont.))

B.



C.

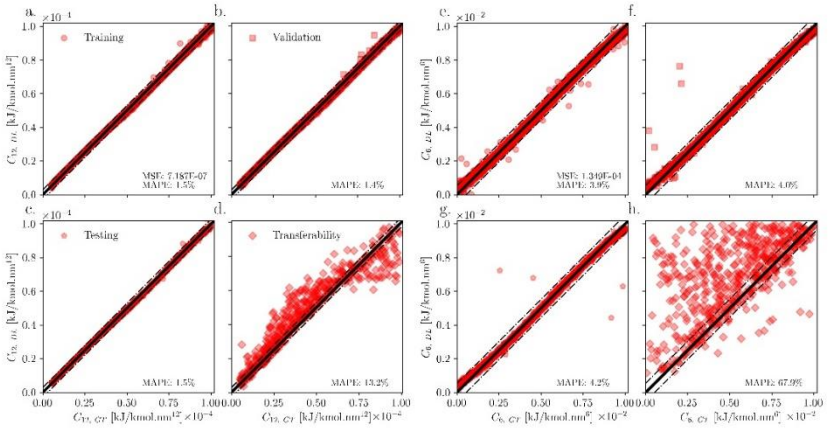


Figure A.9. The error distribution for three 2-hidden-layer networks. The error for B follows a more uniform distribution compared to the other cases. A. 60x15 with Power 1 B. 48x15 with Power 3 C. 60x15 with Power 6.

Further assessment for the best DNN is performed for randomly selected data points (both potential parameters and thermodynamics states) of LJ atomistic particles, shown in Figure A10. We observe that DNN predicted potential parameters lead to almost indistinguishable RDFs compared to corresponding reference data.

Figure A.10. Comparison between RDFs of DNN-based (solid red line) parameterized LJ potential and reference MD (dash-dot black line). Points are selected randomly from the dataset.

Comparison between DNN based coarse-graining and two other coarse-graining methods, namely, relative entropy and simplex, are shown in Figure A11 as well as in Figure A4 of the main manuscript. Furthermore, we compare the relative pressure (with respect to the AA reference) from the potentials parameterized using various methods, including the additional united-atom (UA)[206,207] force field for methane (CH4) (see Table A1). Figure A12 shows comparison of RDF between UA, AA and DNN models. Even though the pressure obtained using the UA method is closer to AA model compared to the DNN method (the pressure of UA and DNN models are 0.84 and 0.72 of AA pressure), the RDF of UA model differs in magnitude and location of peak from the AA model. The RDF of DNN model matches the magnitude and location of the peaks of the AA model.

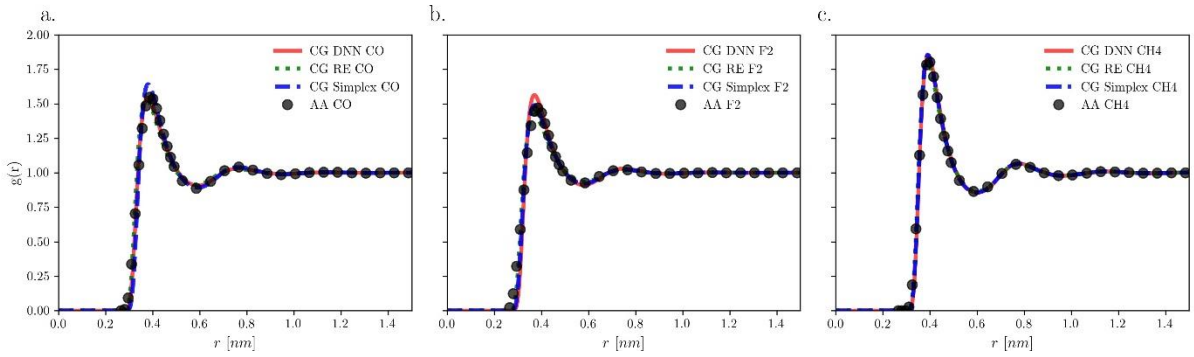


Figure A.11. Comparison between DNN-based (solid red line), relative entropy (dotted green line) and simplex (dash-dot blue line) coarse-graining models. All the three methods show excellent match with the all-atom model results (shown as black circles).

Table A.1. Comparison of the relative pressure obtained with various methods.				
Relative Pressure	$P_{DeepILST}/P_{AA}$	$P_{Simplex}/P_{AA}$	P_{RE}/P_{AA}	P_{UA}/P_{AA}
CO	1.46	**1.23**	1.29	
F2	**1.30**	1.49	1.60	
CH4	0.72	0.66	0.80	**0.84**

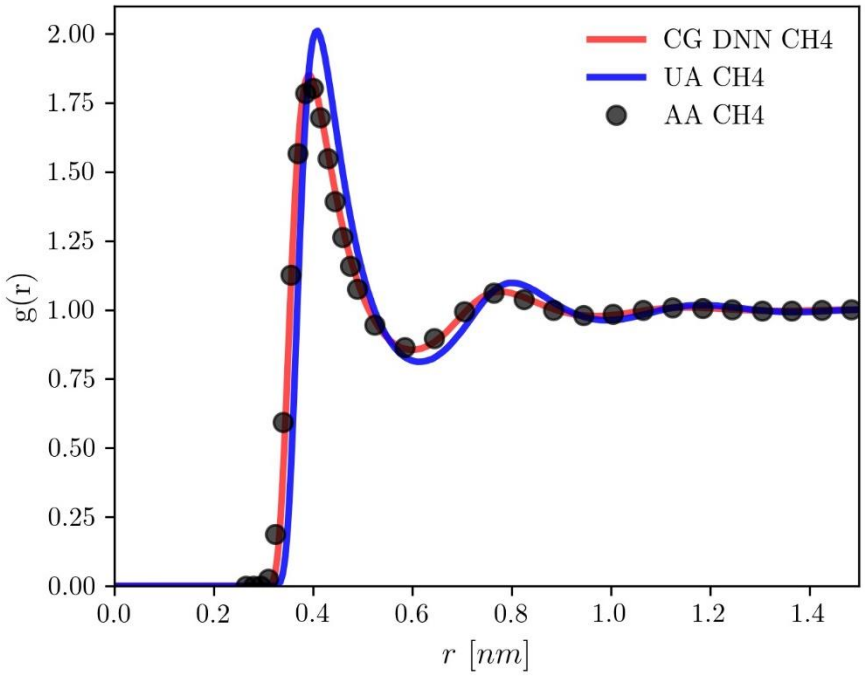


Figure A.12. Comparison of RDF obtained with DNN (solid red line) and UA (dash-dot blue line) models with the all-atom methane (CH4) model (shown with black circles).

Appendix B: Supplementary Information for Chapter 3

B.1 Model Selection and Training

In order to select the deep neural network (DNN) architecture, we have tried various architectures both with different nonlinearities and depths. Based on initial experiments, we have pursued full training of only six DNNs for almost 6,000,000 steps. The six DNNs are shown in Table B1. The loss function for Net 4 is smaller compared with the remaining DNNs, and it has a smaller overfitting as shown in Figure B.1. The overfitting is measured as the distance of loss function value on validation and training dataset. Note that during training of deterministic networks, data usually gets divided into three sets, namely training, validation, and testing.[80] However, in the case of generative networks, data is usually divided into only two sets, training and validation. In this study, we use data from simulation of other simple liquids[100] as the test dataset and generalizability assessment.

Table B1. The DNNs trained for selecting the optimal network.

	Encoder	Latent Space	Decoder	Regularization factor
Net 1	$250 \times 200 \times 125$	75	$125 \times 200 \times 250$	10^{-5}
Net 2	$250 \times 200 \times 125$	75	$125 \times 200 \times 250$	5×10^{-5}
Net 3	$250 \times 200 \times 125$	75	$125 \times 200 \times 250$	10^{-4}
Net 4	$250 \times 200 \times 150$	100	$150 \times 200 \times 250$	10^{-5}
Net 5	$250 \times 200 \times 150$	100	$150 \times 200 \times 250$	5×10^{-5}
Net 6	$250 \times 200 \times 150$	100	$150 \times 200 \times 250$	10^{-4}

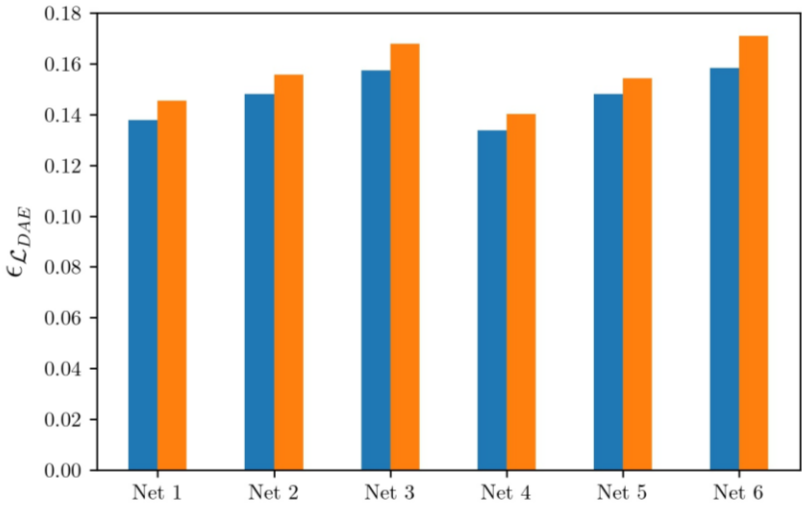


Figure B.1. Loss function for six different DNNs trained for about 6,000,000 iterations. Net 4 is selected for obtaining the results in the main manuscript. Blue and orange boxes show loss function of training and validation data sets, respectively.

The loss functions decrease with training steps for the six DNNs as shown in Figure B2.

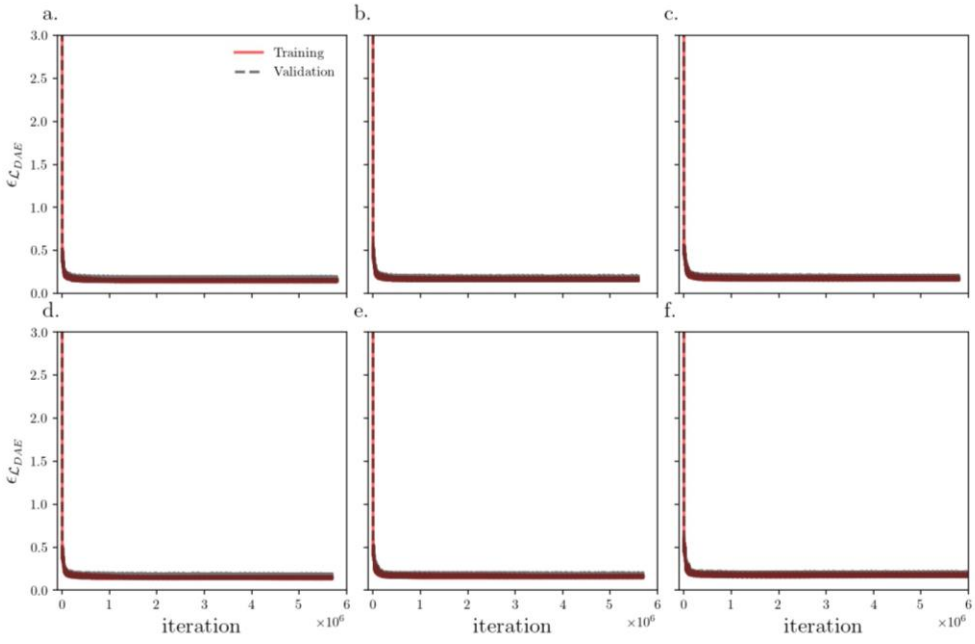


Figure B.2. Loss function with training steps for six different DNNs. a. Net 1 b. Net 2 c. Net 3 d. Net 4 e. Net 5 f. Net 6.

Here, we also invesigate the transferability of DAE network to unforeseen thermodynamic states. To do so, we feed RDF of various Lennard-Jones liquids into frozed DAE network. We observe that the performance of DAE detoriates as we explore thermodynamic state far away from dataset thermodynamic states (shown in Figure B.3). However, we observe a clear correlation between peak radial distance between DAE prediction and MD ground truth RDFs.

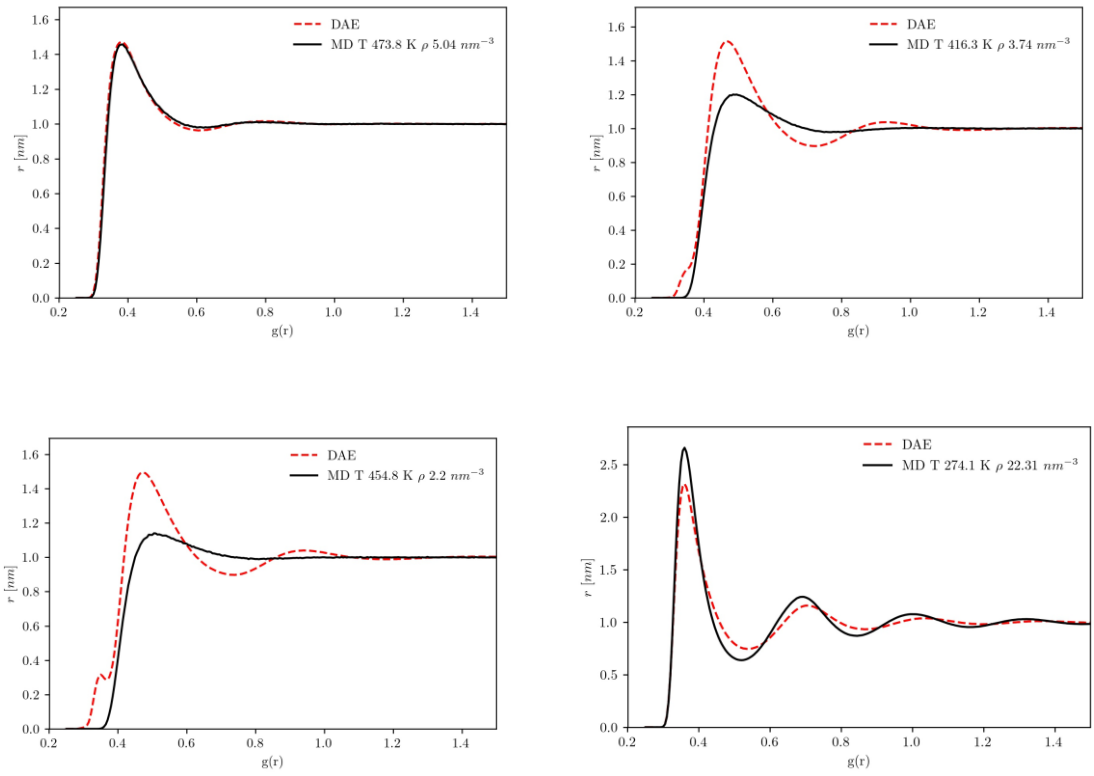


Figure B.3. Transferability of DAE network for prediction of RDF for unforseen thermodynamic states.

B.2 Simulation Details

The potential parameters and thermodynamic states of 6 randomly selected LJ systems studied in the main manuscirpt are given in Table B2.

Table B.2. Potential parameters and thermodynamic states of LJ systems studied in Figures 3.2-3 of the main manuscript. a, b, c, d, e , and f refer to the results with the same alphabetic order.

System	Potential Parameters		Thermodynamic States	
	$C_{12}\ [\frac{kJ}{kmol\ nm^{12}}]$	$C_6\ [\frac{kJ}{kmol\ nm^6}]$	$T\ [K]$	$\rho\ [nm^{-3}]$
a	2.215×10^{-5}	3.06×10^{-3}	385.42	16.05
b	8.160×10^{-6}	7.26×10^{-3}	346.45	16.44
c	2.232×10^{-5}	2.03×10^{-3}	325.98	13.40
d	6.190×10^{-5}	7.59×10^{-3}	399.86	13.53
e	4.086×10^{-5}	3.27×10^{-3}	399.86	14.33
f	2.395×10^{-5}	6.74×10^{-3}	318.46	11.40

Potential parameters used in MD simulation of simple liquids are given in Table B.3. The thermodynamic states of all the simple liquid systems are identical with temperature and density of 350 K and 12.23 nm^{-3}, respectively.

Table B.3. Potential parameters of simple liquids studied in the main manuscript.

	Parameters	Values
Exponential Potential	$A\ [\frac{kJ}{kmol}]$	1.34×10^{6}
	$k_D\ [nm^{-1}]$	3.424×10^{1}
Yukawa Potential	$A\ [\frac{kJ}{kmol}]$	3.17×10^{7}
	$k_D\ [nm^{-1}]$	3.203×10^{1}
Inverse-Power-Law Potential (14-8)	$C_{14}\ [\frac{kJ}{kmol.nm^{14}}]$	4.142×10^{-6}
	$C_8\ [\frac{kJ}{kmol.nm^{8}}]$	6.077×10^{-5}
Inverse-Power-Law Potential (10-4)	$C_{10}\ [\frac{kJ}{kmol.nm^{10}}]$	2.500×10^{-4}
	$C_4[\frac{kJ}{kmol.nm^{8}}]$	8.521×10^{-2}

B.3 Comparison with Baselines

Principal component analysis (PCA) and independent component analysis (ICA) are dimensionality reduction and denoising algorithms in the machine learning for high dimensional and large dataset.[90,91] In particular, PCA method is developed for data with Gaussian distribution, and further study is in progress for non-Gaussian data.[91] The denoising performance of PCA also requires additional condition of independent and identically distributed (iid) Gaussian noise components. We assess validity of these conditions, namely being Gaussian and iid.[208] To do so, we test departure from normality with calculation

of p-value for null hypothesis of noise being drawn from Gaussian distribution with $\alpha = 0.05$ (the test combines skew and kurtosis to assess departure of normality, we use scipy stats package of python to perform this test).[96] We observe that at various radial distances the null hypothesis is rejected (shown below with red circles in Figure B4a). Figure B4b shows the noise distribution of Argon Lennard-Jones liquids at radial distance of 0.46 nm, which clearly shows a non-Gaussian distribution.

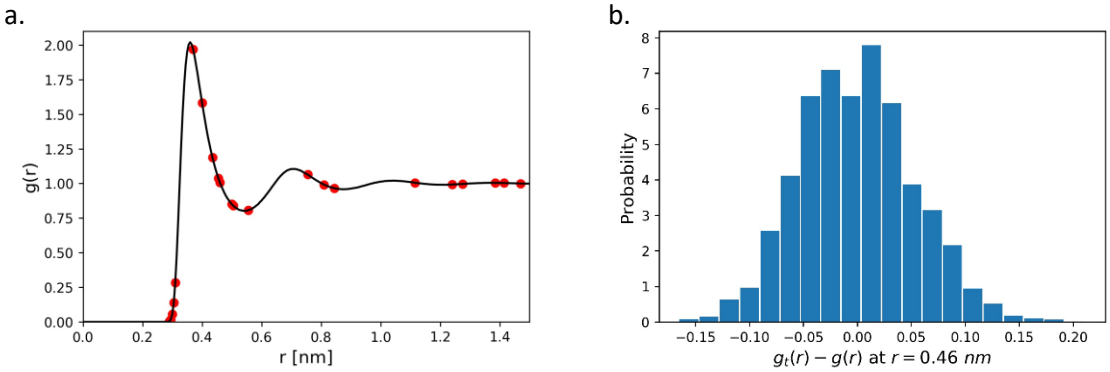


Figure B.4. a. Non-Guassian noise at various radial distance b. noise distribution at a specific radial distance

To check the iid condition of noise components, mutual information[97,98] between the noise component at each radial distance with the noises at other radial distances is calculated. The mutual information can be expressed as,

$$I(X,Y) = \sum_{x} \int \log\left(\frac{p(x,y)}{p(x)p(y)}\right) dy \qquad \text{B.1}$$

where (x, y) are the data points with underlying probability distribution $p(x, y)$. For iid data, mutual information is equal to zero, and otherwise its non-zero. The maximum value of estimated mutual information at various radial distances are shown at figure below (the test is performed using scikit-learn package of python).

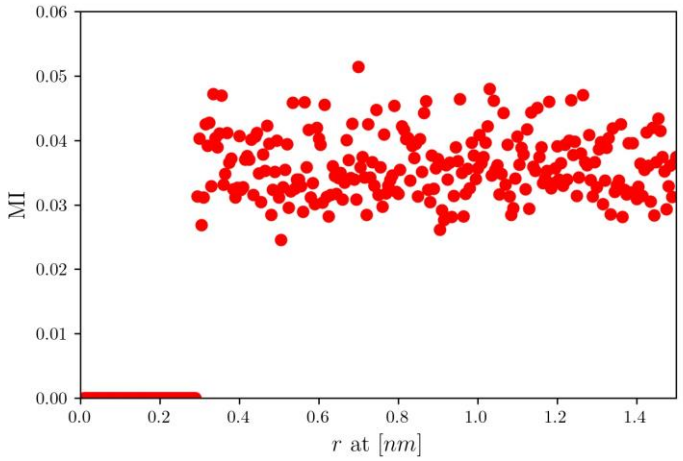


Figure B.5. Mutual information at various radial distance. The maximum value of mutual information between noise at each radial distance with other radial distances are shown with circles. The non-zero circles show dependency with other points.

As shown by the above figure mutual information has a non-zero value, which indicates dependency between noises, which again violates the iid condition.

Even though the conditions for validity of PCA and ICA are not fully-satisfied, we compare performance of PCA and ICA with different numbers of component with the DAE method developed in the main manuscript. In general, we observe that the DAE method is better than both methods consistent with recent studies reporting a better performance for denoising autoencoder compared with PCA in physical problems.[86,209] The main problem associated with PCA and ICA is unphysical negative values as these methods do not guarantee non-negative value for the non-negative dataset. We investigate both methods up to 19 components with dataset of 1500 RDFs. For any number of components, the DAE method outperforms both methods. RDFs of various system reconstructed by PCA and ICA with number of components between 3 and 11 are shown in Figure B6-14. Similar trends are observed for the number of components between 12 and 18. Further increase in the number of components, in fact, deteriorates

denoising performance of both PCA and ICA methods as shown in Figure B15 for number of component equal to 19.

PCA with 3 components

ICA with 3 components

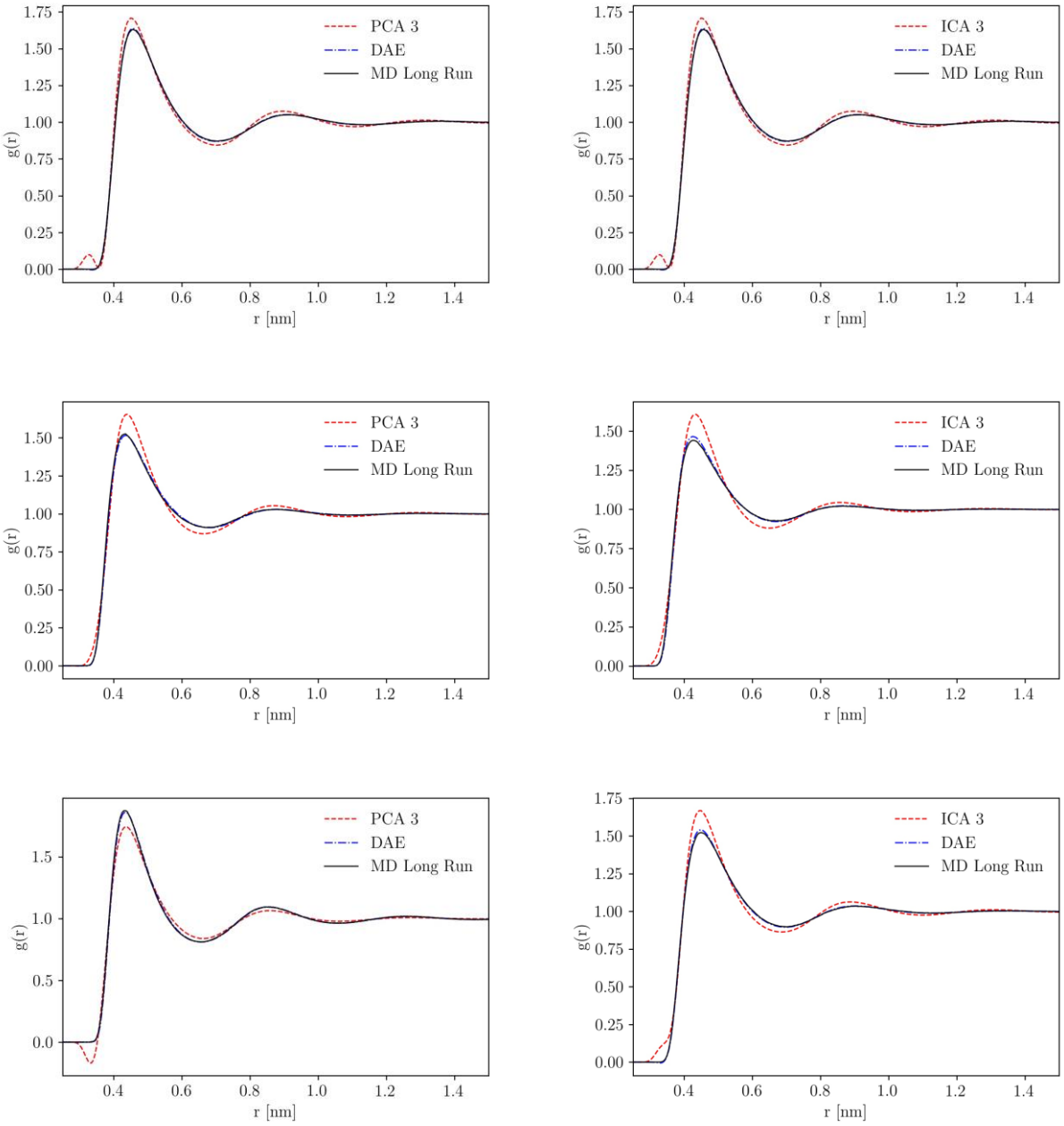


Figure B.6. Comparison between DAE, PCA, and ICA (ICA and PCA with 3 components).

PCA with 4 components

ICA with 4 components

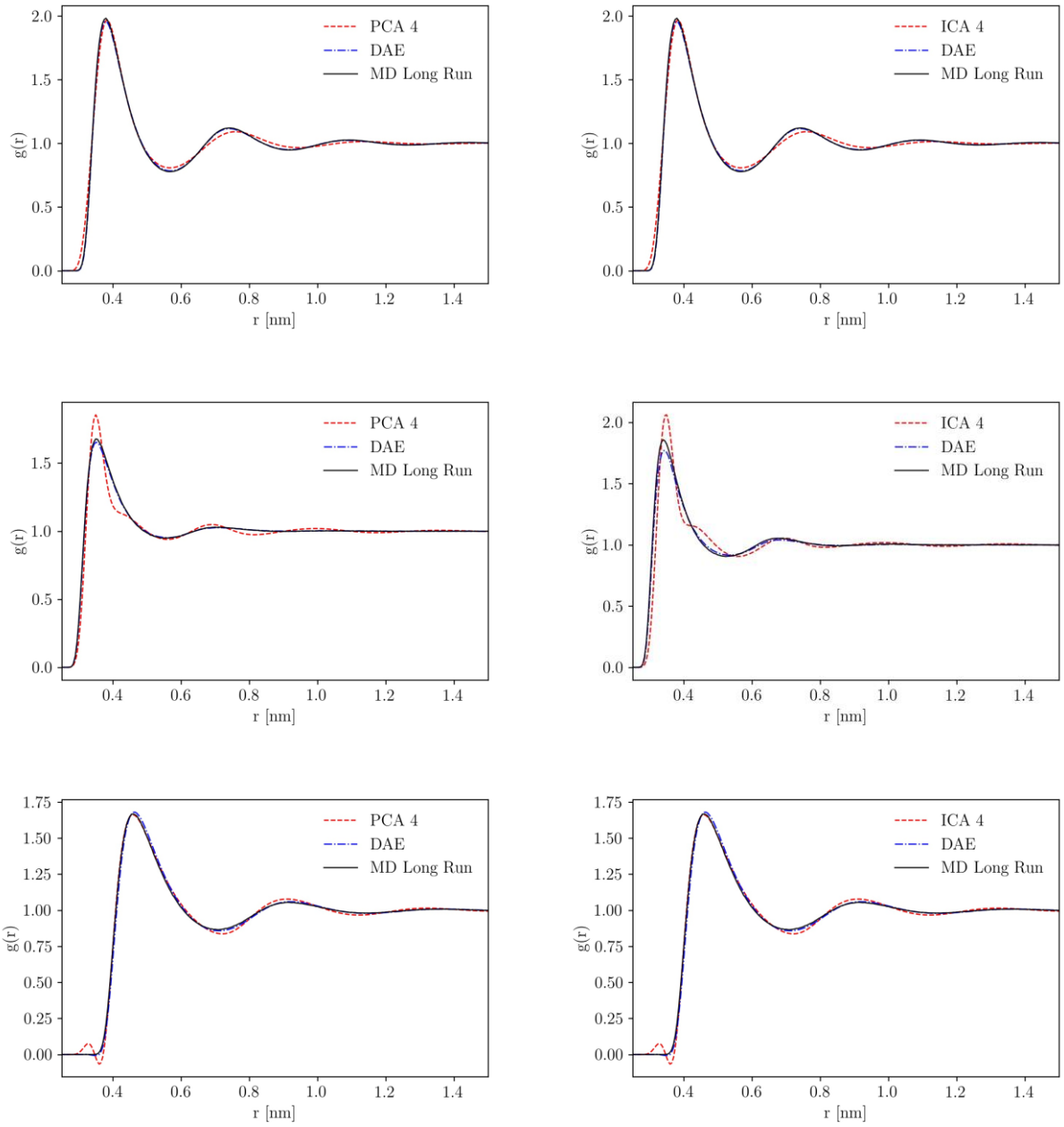


Figure B.7. Comparison between DAE, PCA, and ICA (ICA and PCA with 4 components).

PCA with 5 components

ICA with 5 components

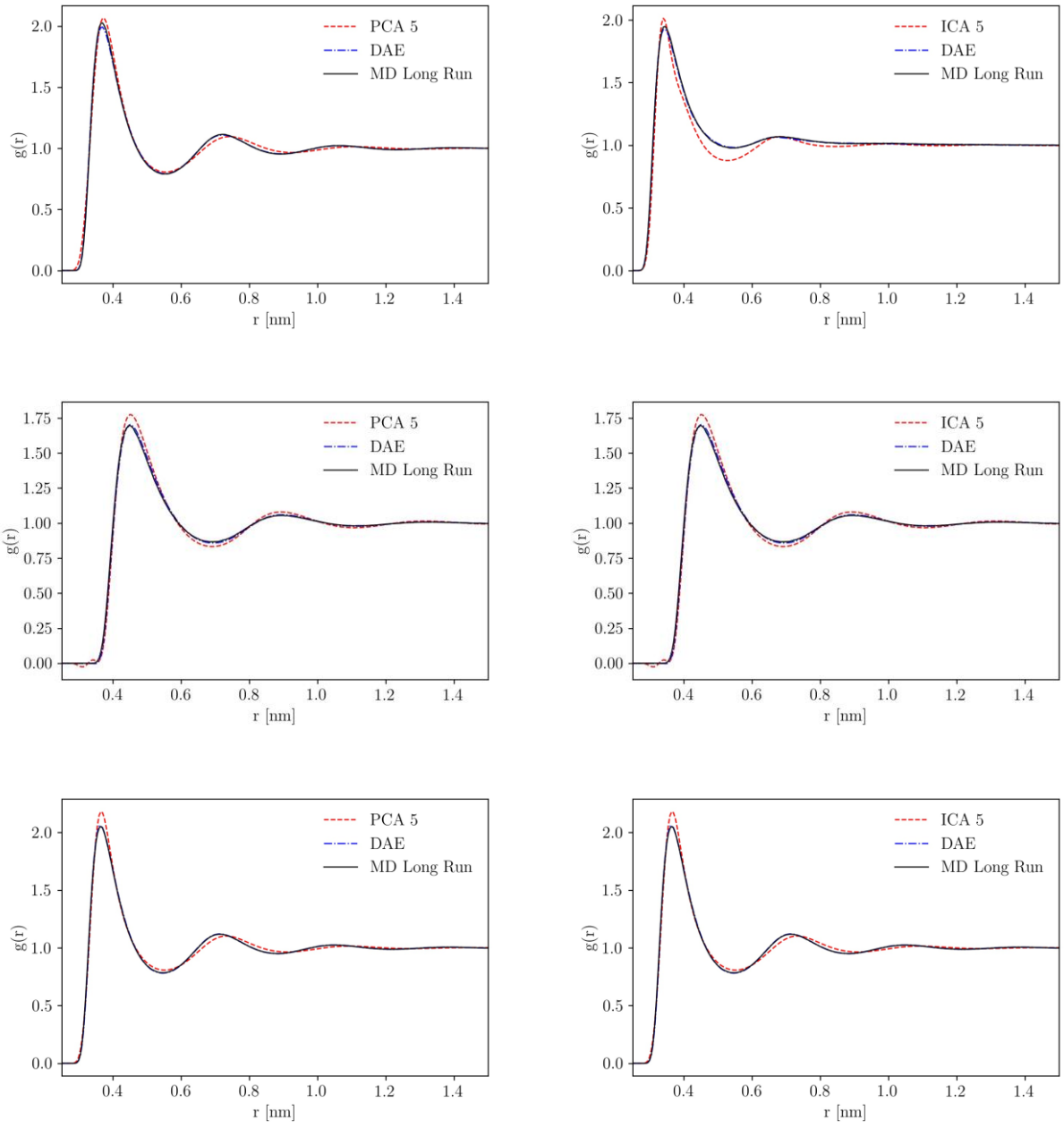


Figure B.8. Comparison between DAE, PCA, and ICA (ICA and PCA with 5 components).

PCA with 6 components

ICA with 6 components

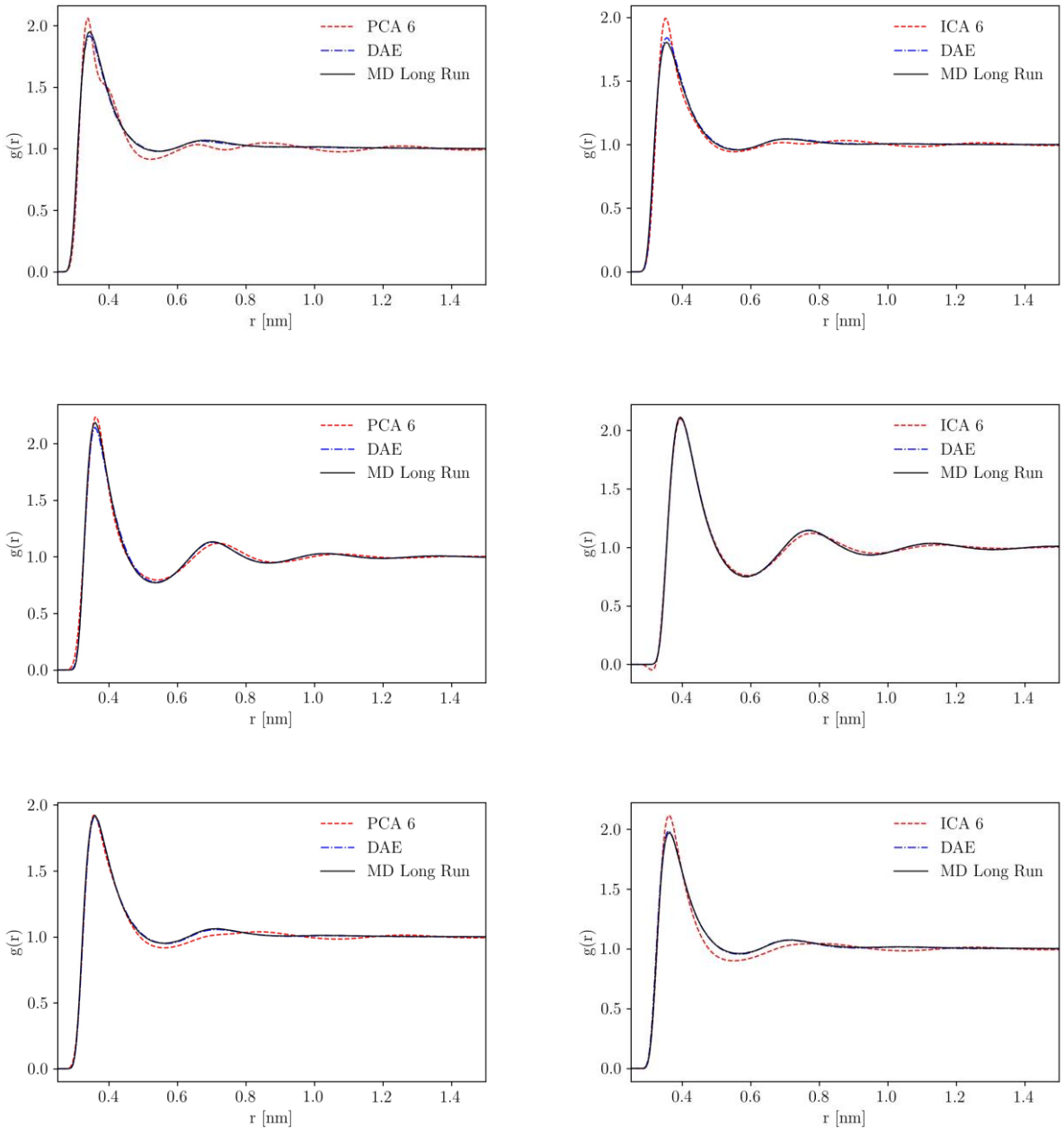


Figure B.9. Comparison between DAE, PCA, and ICA (ICA and PCA with 6 components).

PCA with 7 components

ICA with 7 components

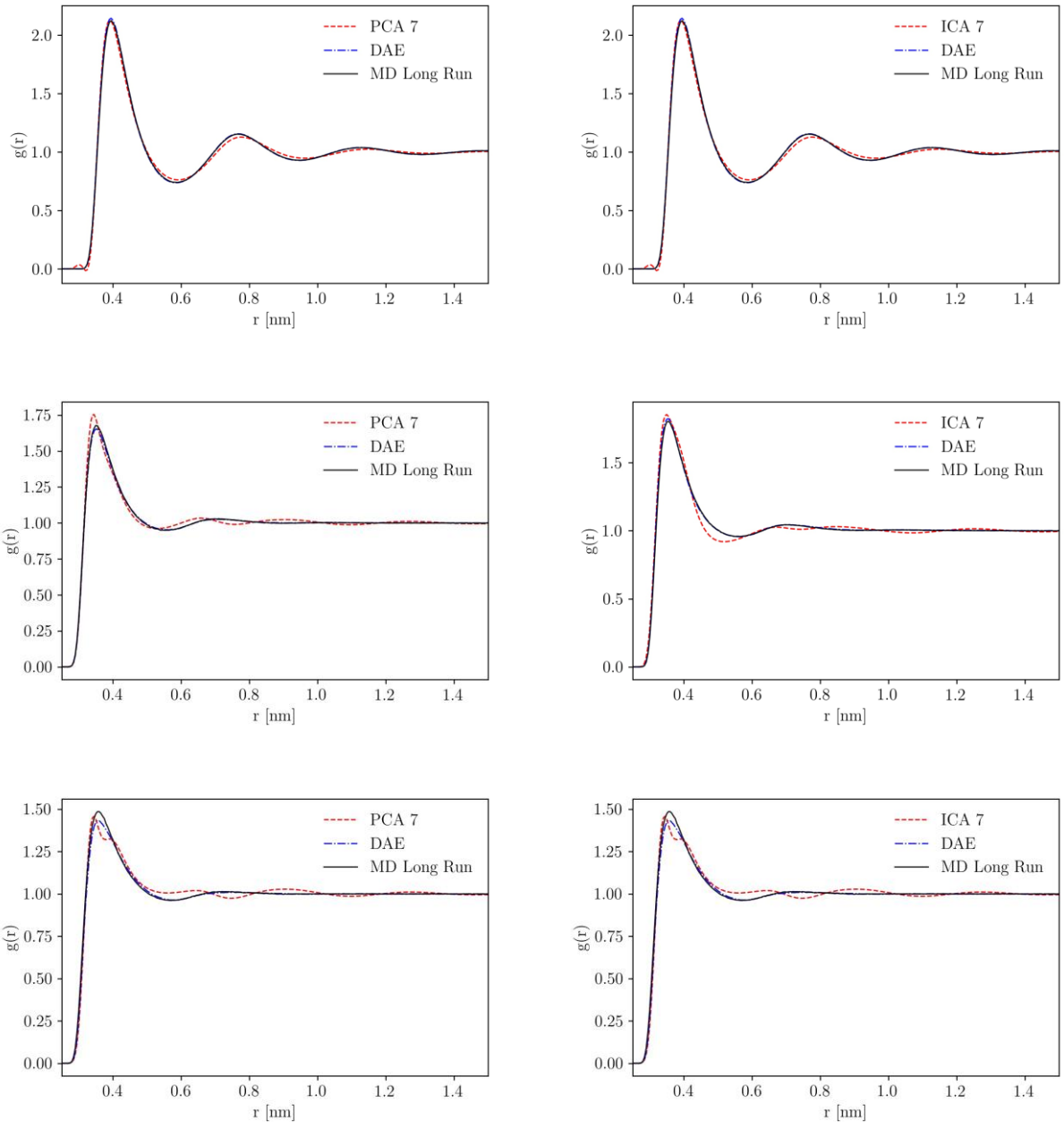


Figure B.10. Comparison between DAE, PCA, and ICA (ICA and PCA with 7 components).

PCA with 8 components

ICA with 8 components

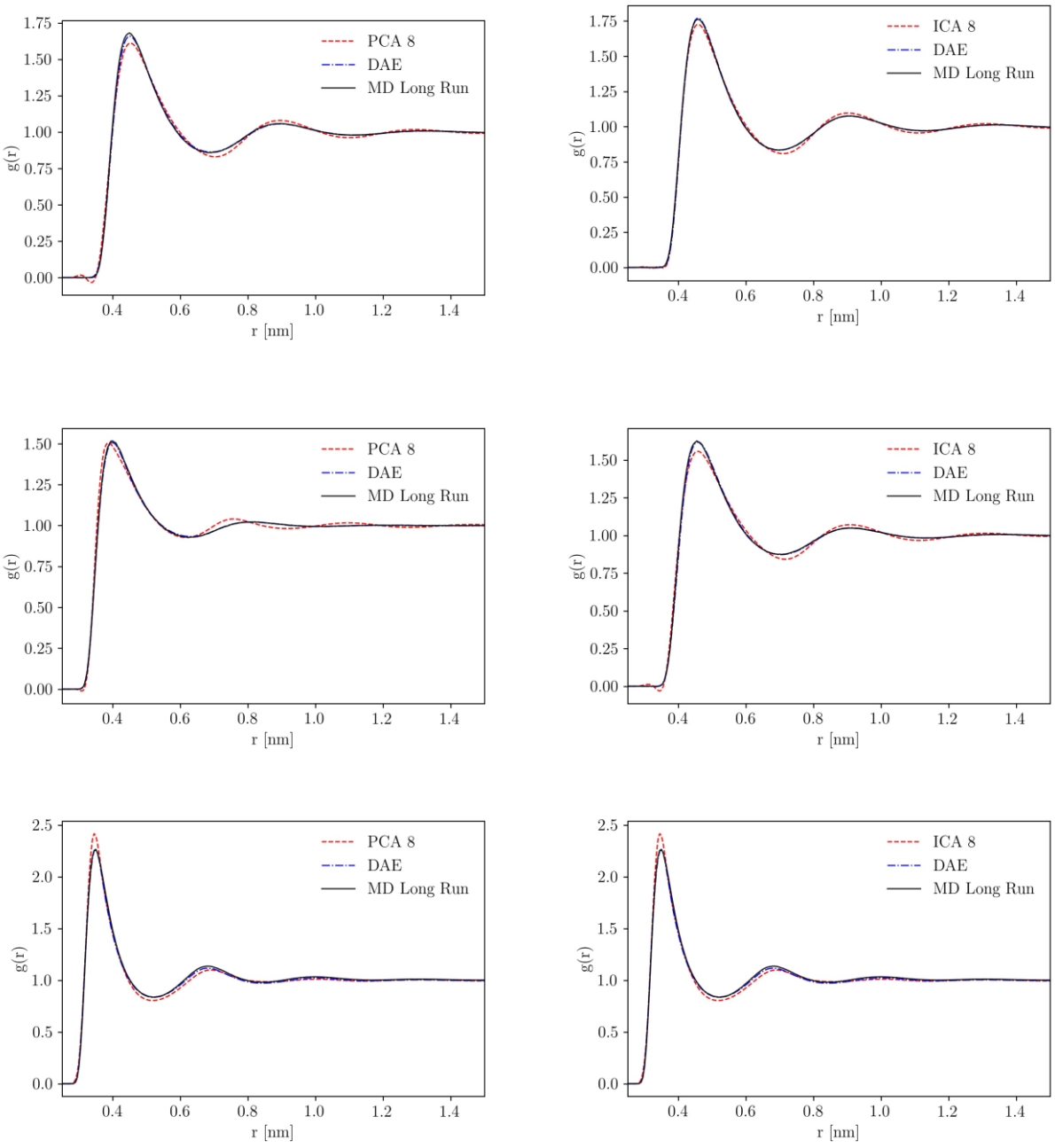


Figure B.11. Comparison between DAE, PCA, and ICA (ICA and PCA with 8 components).

PCA with 9 components

ICA with 9 components

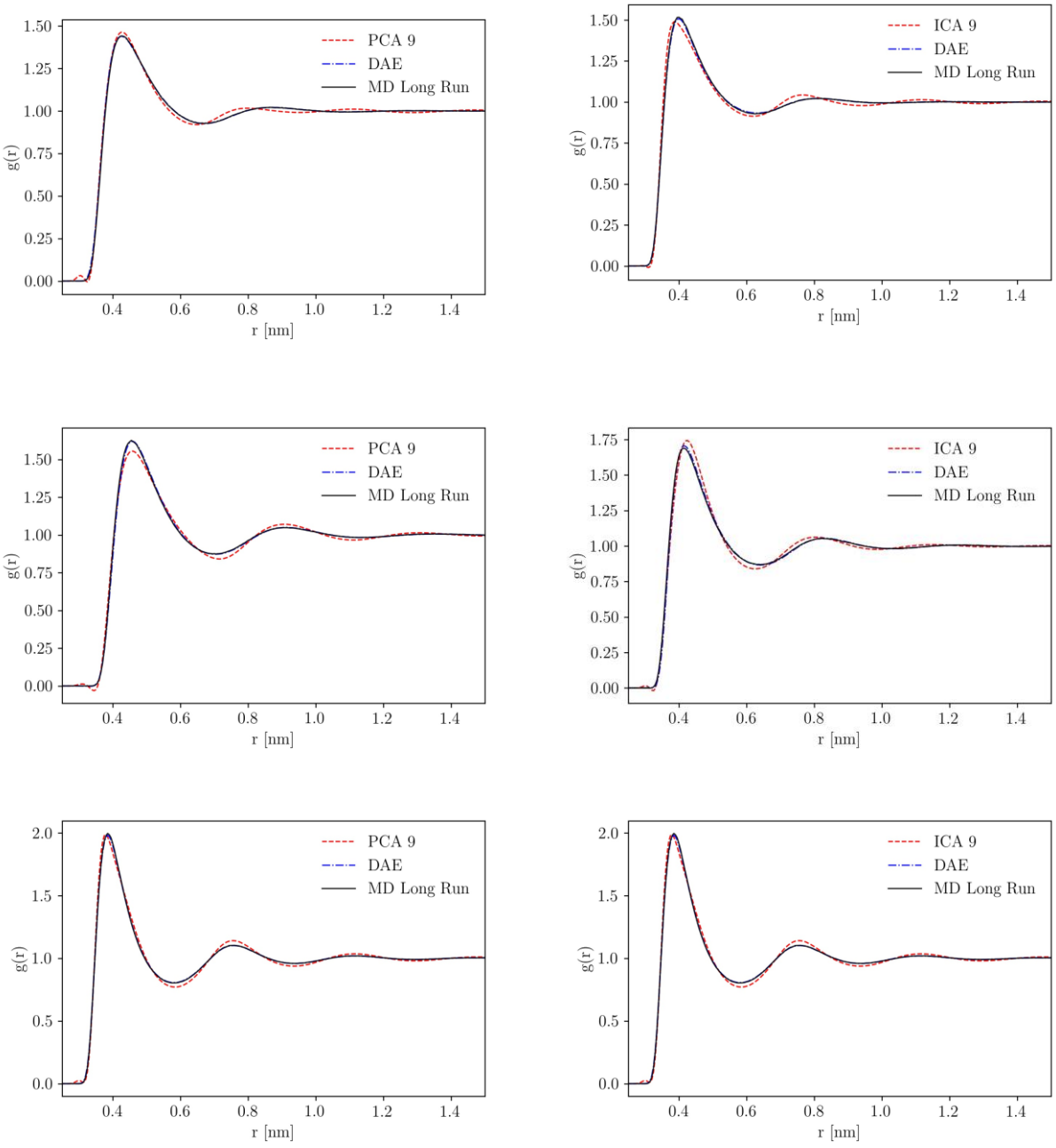


Figure B.12. Comparison between DAE, PCA, and ICA (ICA and PCA with 9 components).

PCA with 10 components

ICA with 10 components

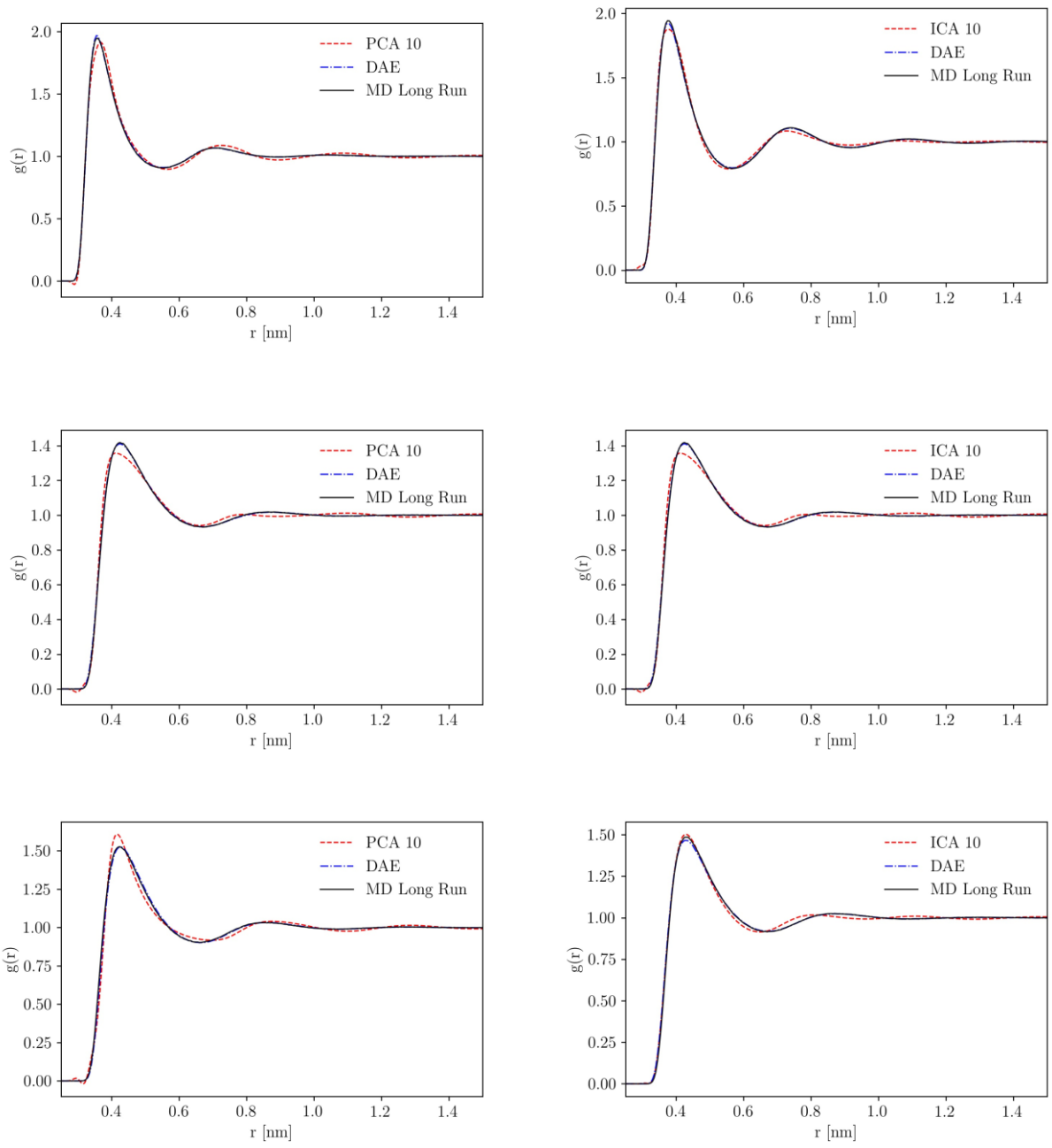


Figure B.13. Comparison between DAE, PCA, and ICA (ICA and PCA with 10 components).

PCA with 11 components

ICA with 11 components

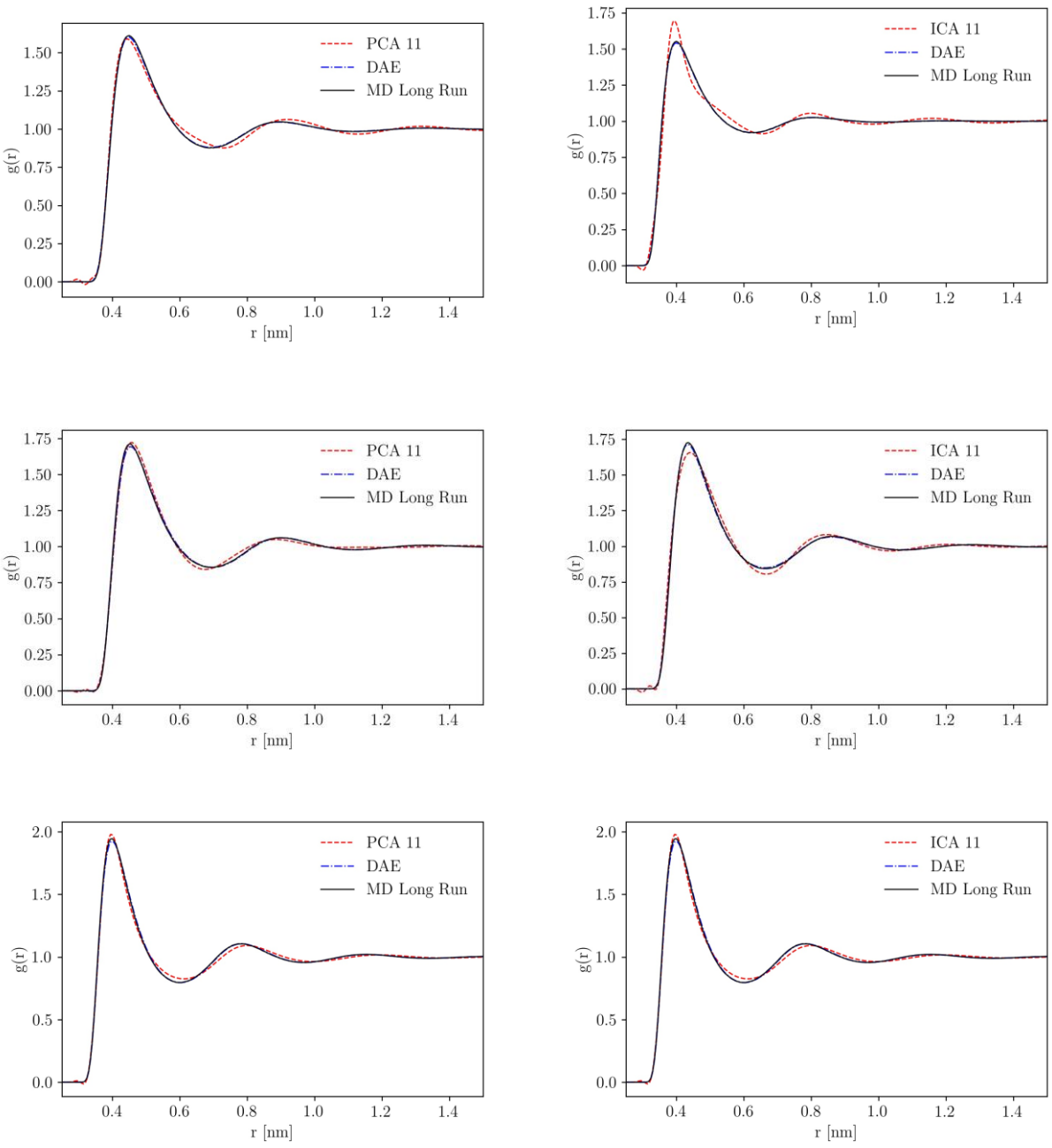


Figure B.14. Comparison between DAE, PCA, and ICA (ICA and PCA with 11 components).

PCA with 19 components ICA with 19 components

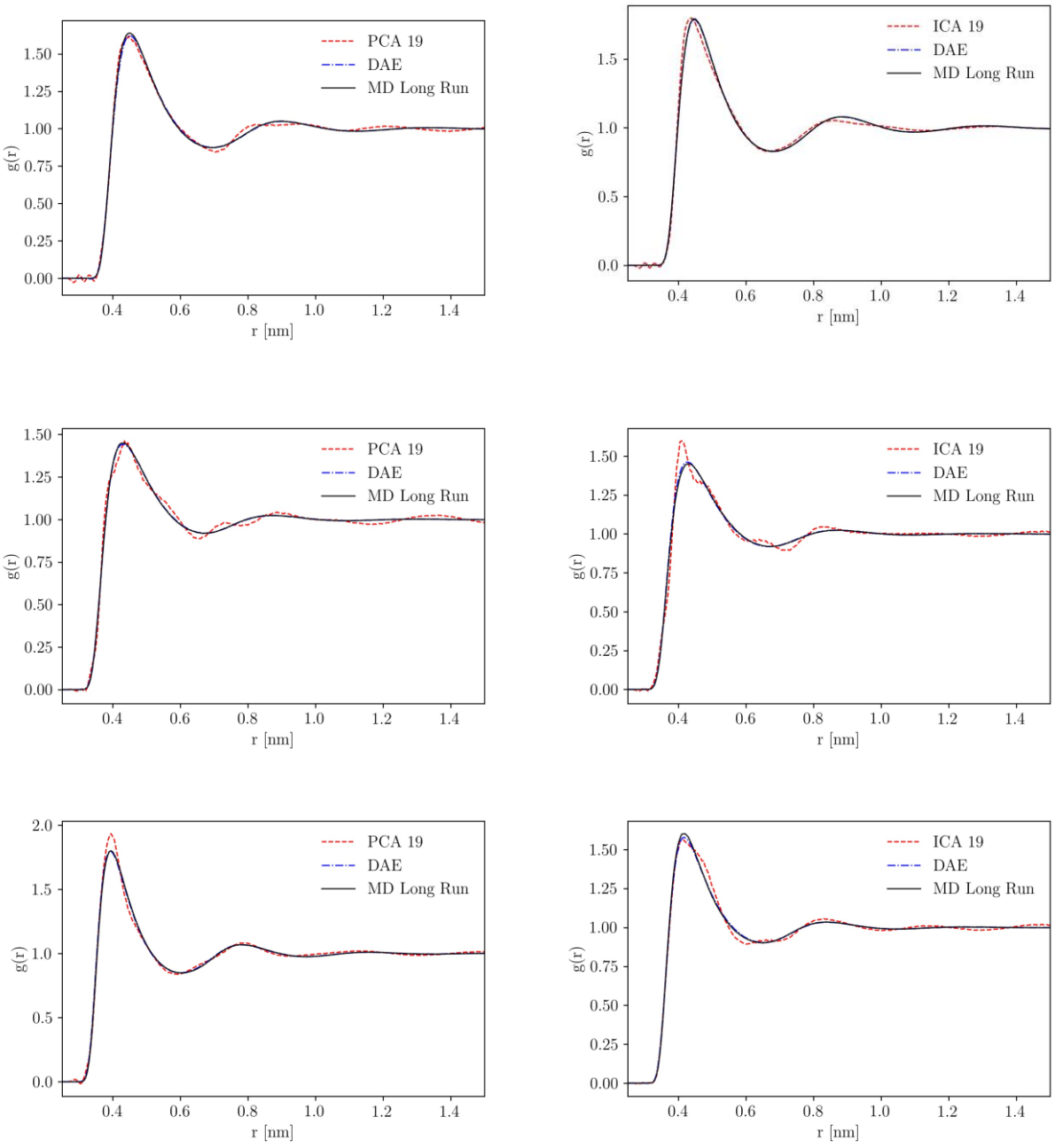


Figure B.15. Comparison between DAE, PCA, and ICA (ICA and PCA with 19 components).

B.4 DAE with PCA Input-Output

We employ PCA algorithm with 20 components, which show excellent ability in the dimensionality reduction of average temporally averaged RDF, however, it comes at the cost of losing denoising

functionality for single snapshot RDF. We augment the denoising deficiency of PCA with large number of components by using it as input and output of the DAE network. This methodology, in turn, reduce the size of the DAE network. We find PCA free parameter and freeze the PCA model. Once both single snapshot and temporally averaged RDFs are dimensionally reduced with PCA, we train a small DAE network with two 2 hidden layers (encoder and decoder networks have two hidden layer with 20x15x10x15x20 architecture for DAE, we don't go through rigorous optimization of network architecture, as the main objective is to show how DAE network can augment traditional method, and in return gets smaller combing two methods with each other). Then, the DAE learns to map noisy PCA component of single snapshot RDF back to PCA components of temporally averaged RDF. The results shown in Figure B16 clearly indicates that DAE combined with PCA outperform PCA denoising ability. The loss function of the training of DAE network with PCA input-output is also shown in Figure B17.

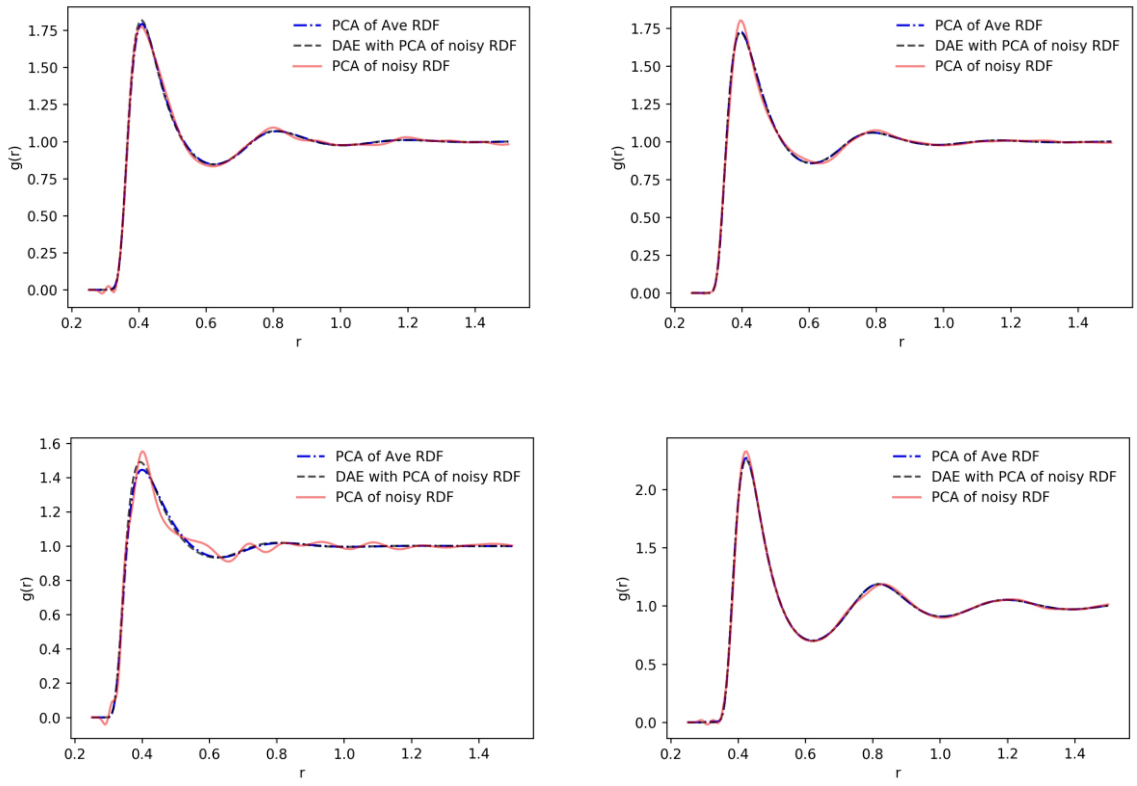


Figure B.16. DAE with PCA input/output RDF prediction for four randomly selected systems. PCA

with 20 components. The dash-dotted blue lines show PCA output for temporally averaged RDF. Solid red line show the PCA of a single snapshot RDF, which is feed into DAE network to reproduce DAE predicted temporally averaged RDF shown with dash-dot blue lines.

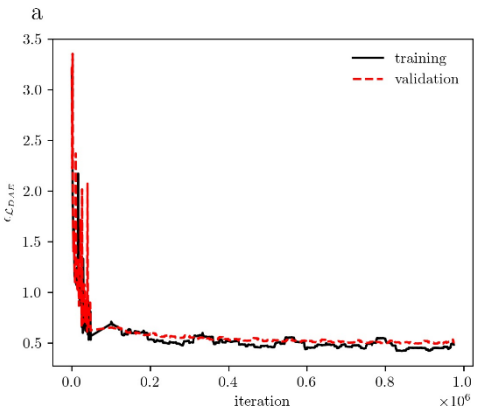


Figure B.17. DAE with PCA input/output loss function during training steps.

Appendix C: Supplementary Information for Chapter 4

C.1 Statistical Model

As mentioned in the main manuscript, to the best of our knowledge, there is no theorem to prove the uniqueness of a transformation (function) of random variable such that it transforms the probability density function (PDF) of a given random variable into a specific PDF. Here, we provide a few counterexamples to establish the non-uniqueness.

Assume that there exists a theorem stating that there is a unique function transforming the PDF of a given random variable into a specific PDF. The theorem should hold for all PDFs; however, we can consider random variable X with a uniform PDF in the interval of [-1, 1] ($p(X = x) = \begin{cases} \frac{1}{2} & if -1 < x < 1 \\ 0 & else \end{cases}$). Then all the following transformations produce the same PDF,

$$f(x) = (x + 1)^2 \quad \text{(C.1)}$$

$$g(x) = (x - 1)^2 \quad \text{(C.2)}$$

$$h(x) = x^2 \quad \text{(C.3)}$$

Therefore, the non-uniqueness of function is proved. Note that none of the above functions, in general, is equal to another one, but they have the same PDF as shown in Figure C1, and expressed below,

$$p(Y = y) = \begin{cases} \frac{1}{2\sqrt{y}} & if\ 0 < y\ < 1 \\ 0 & else \end{cases} \quad \text{(C.4)}$$

where Y can be any of the functions mentioned in Eq. C 1-3.

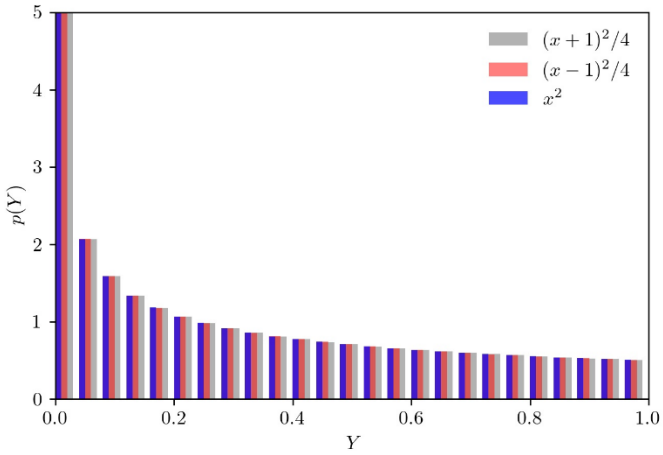


Figure C.1. Comparison between PDF of three different transformations of a uniform random variable between -1 and 1, which gets mapped to a new random variable Y with the same PDF.

C.2 Derivation of Relation for Order Parameter

Eq. 19 in the main manuscript ($\tau = k\ \frac{\rho^*}{T^*}$) is obtained through following mathematical manipulation on Eq. 18 $g(r) \sim 1 - \frac{u_{LJ}(r)}{k_B T}$

$$g(r) - 1 = -\frac{4\epsilon}{k_B T}\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{C.5}$$

$$|g(r) - 1| = \frac{4\epsilon}{k_B T}\left|\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| \tag{C.6}$$

$$\rho|g(r) - 1|r^2 dr = \rho\ \frac{4\epsilon}{k_B T}\left|\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| r^2 dr \tag{C.7}$$

$$\int \rho|g(r) - 1|r^2 dr = \int \rho\ \frac{4\epsilon}{k_B T}\left|\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| r^2 dr \tag{C.8}$$

$$\tau = \frac{\rho\sigma^3}{T^*}\int \left|4\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| \left(\frac{r}{\sigma}\right)^2 \frac{dr}{\sigma} \tag{C.9}$$

$$\tau = \frac{\rho^*}{T^*}\int \left|4\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| \left(\frac{r}{\sigma}\right)^2 \frac{dr}{\sigma} \tag{C.10}$$

Finally, defining k as $\int \left|4\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\right| \left(\frac{r}{\sigma}\right)^2 \frac{dr}{\sigma}$, linear relationship between order parameter and non-dimensional temperature and density can be written as,

$$\tau = k\ \frac{\rho^*}{T^*} \tag{C.11}$$

C.3. Case Studies

To further understand the consistency between DeepILST[80] and structural quasi-universality[2], we performed calculations on several quasi-universal (QU) pair potentials at various thermodynamic states.

In particular, for each QU pair potential, 10 systems with different potential parameters and various thermodynamic states are studied.

Case.1. Exponential

The exponential pair potential with various potential parameters and thermodynamic states is simulated. The list of systems and their parameters is provided in Table C1. Comparison between RDFs of each system and its QU-equivalent LJ system is shown in Figure C2. From the total deviation in the RDFs (ϵ_{rdf}~0.0011), we see a very similar structure between the QU pair potential and QU-equivalent LJ potential indicating consistency of DeepILST with the structural quasi-universality of simple liquids. Note that the results of the 1st system for each pair potential case are not shown in the supplementary as these are provided in the main manuscript.

Case.2. Yukawa

RDFs of the Yukawa pair potential, also known as the screened Coulombic pair potential, with various potential parameters and thermodynamic states are obtained. The list of systems and the parameters in each system are provided in Table C2. Comparison between RDFs of each system and its QU-equivalent LJ system is shown in Figure C3. From the total deviation in the RDFs (ϵ_{rdf}~0.0009), we again see a very similar structure between both systems.

Case.3. Inverse-power-law $(\boldsymbol{p}, \boldsymbol{q}) = (\mathbf{14}, \mathbf{8})$

RDFs of the inverse-power-law (IPL) pair potential with repulsive and attractive exponents of 14 and 8, respectively, are obtained by considering various potential parameters and thermodynamic states. The list of systems and the parameters are provided in Table C3. Comparison between RDFs of each system and its QU-equivalent LJ system is shown in Figure C4. With ϵ_{rdf}~0.002, we observe a similar structure between the LJ system and the IPL system.

Case.4. Inverse-power-law $(\boldsymbol{p}, \boldsymbol{q}) = (\mathbf{10}, \mathbf{4})$

RDFs of the IPL pair potential, with repulsive and attractive exponents of 10 and 4, respectively, are obtained by considering various potential parameters and thermodynamic states. The list of systems and the parameters are provided in Table C4. Comparison between RDFs of each system and its QU-equivalent LJ system is shown in Figure C5. The total deviation between RDFs is again small ($\epsilon_{rdf} \sim 0.0024$) implying a similar structure between both systems.

Table C.1. Exponential pair potential systems with different thermodynamic states and potential parameters

System	Potential Type	$T\ [K]$	$\rho\ [nm^{-3}]$	$A\ [\frac{kJ}{\quad}]$	$k_{\quad}[\frac{1}{\quad}]$
1	EXP	350.00	12.23	$1.34e^6$	34.24
2	EXP	394.42	17.23	$4.88e^4$	29.52
3	EXP	333.00	8.21	$1.15e^6$	31.25
4	EXP	328.68	17.22	$7.62e^5$	34.79
5	EXP	386.41	14.00	$1.24e^6$	33.90
6	EXP	357.52	11.41	$8.16e^5$	33.39
7	EXP	376.61	10.79	$8.16e^5$	34.60
8	EXP	352.28	12.17	$1.31e^6$	33.50
9	EXP	304.23	10.56	$7.67e^5$	34.82
10	EXP	341.5	10.33	$1.50e^6$	35.83

Table C.2. Yukawa pair potential systems with different thermodynamic states and potential parameters

System	Potential Type	$T\ [K]$	$\rho\ [nm^{-3}]$	$A\ [\frac{kJ}{mol}]$	$k_D[\frac{1}{nm}]$
1	Yukawa	350.00	12.23	$1.055e^5$	31.44
2	Yukawa	364.79	14.86	$9.82e^4$	32.03
3	Yukawa	373.45	9.32	$9.50e^4$	33.07
4	Yukawa	367.32	12.80	$8.67e^4$	33.51
5	Yukawa	326.88	16.18	$9.95e^4$	31.65
6	Yukawa	386.41	14.00	$1.11e^5$	34.38
7	Yukawa	328.68	17.22	$9.75e^4$	33.12
8	Yukawa	305.76	16.41	$7.64e^4$	32.67
9	Yukawa	344.14	11.35	$1.04e^5$	34.84
10	Yukawa	399.48	17.97	$1.07e^5$	32.38

Table C.3. Inverse-power-law (14,8) pair potential systems with different thermodynamic states and potential parameters

System	Potential Type	$T\ [K]$	$\rho\ [nm^{-3}]$	$C_{14}[\frac{kJ}{mol.nm^{14}}]$	$C_8\ [\frac{kJ}{mol.nm^8}]$
1	IPL (14,8)	350.00	12.23	$4.340e^{-6}$	$6.424e^{-5}$
2	IPL (14,8)	350.00	12.23	$3.32e^{-6}$	$5.16e^{-5}$
3	IPL (14,8)	312.27	9.28	$4.14e^{-6}$	$6.08e^{-5}$
4	IPL (14,8)	360.57	10.20	$4.31e^{-6}$	$4.75e^{-5}$
5	IPL (14,8)	392.82	9.52	$5.07e^{-6}$	$4.71e^{-5}$
6	IPL (14,8)	324.01	8.29	$7.46e^{-6}$	$2.97e^{-5}$
7	IPL (14,8)	350.00	12.23	$6.57e^{-6}$	$3.69e^{-5}$
8	IPL (14,8)	312.27	9.28	$8.02e^{-6}$	$2.64e^{-5}$
9	IPL (14,8)	303.86	12.43	$3.26e^{-6}$	$7.02e^{-5}$
10	IPL (14,8)	333.00	8.21	$7.40e^{-6}$	$4.53e^{-5}$

Table C.4. Inverse-power-law (10,4) pair potential systems with different thermodynamic states and potential parameters

System	Potential Type	$T\ [K]$	$\rho\ [nm^{-3}]$	$C_{10}\ [\frac{kJ}{mol.nm^{10}}]$	$C_4\ [\frac{kJ}{mol.nm^4}]$
1	IPL (10,4)	350.00	12.23	$2.492e^{-4}$	$6.090e^{-2}$
2	IPL (10,4)	350.00	12.23	$1.80e^{-4}$	$8.28e^{-2}$
3	IPL (10,4)	303.86	12.43	$2.47e^{-4}$	$9.50e^{-2}$
4	IPL (10,4)	344.80	14.99	$2.79e^{-4}$	$8.34e^{-2}$
5	IPL (10,4)	339.18	14.61	$2.67e^{-4}$	$9.97e^{-2}$
6	IPL (10,4)	352.28	12.17	$2.79e^{-4}$	$6.14e^{-2}$
7	IPL (10,4)	322.92	13.69	$2.28e^{-4}$	$8.45e^{-2}$
8	IPL (10,4)	367.32	12.80	$2.45e^{-4}$	$7.34e^{-2}$
9	IPL (10,4)	383.48	11.78	$2.50e^{-4}$	$8.52e^{-2}$
10	IPL (10,4)	386.41	13.40	$1.94e^{-4}$	$6.51e^{-2}$

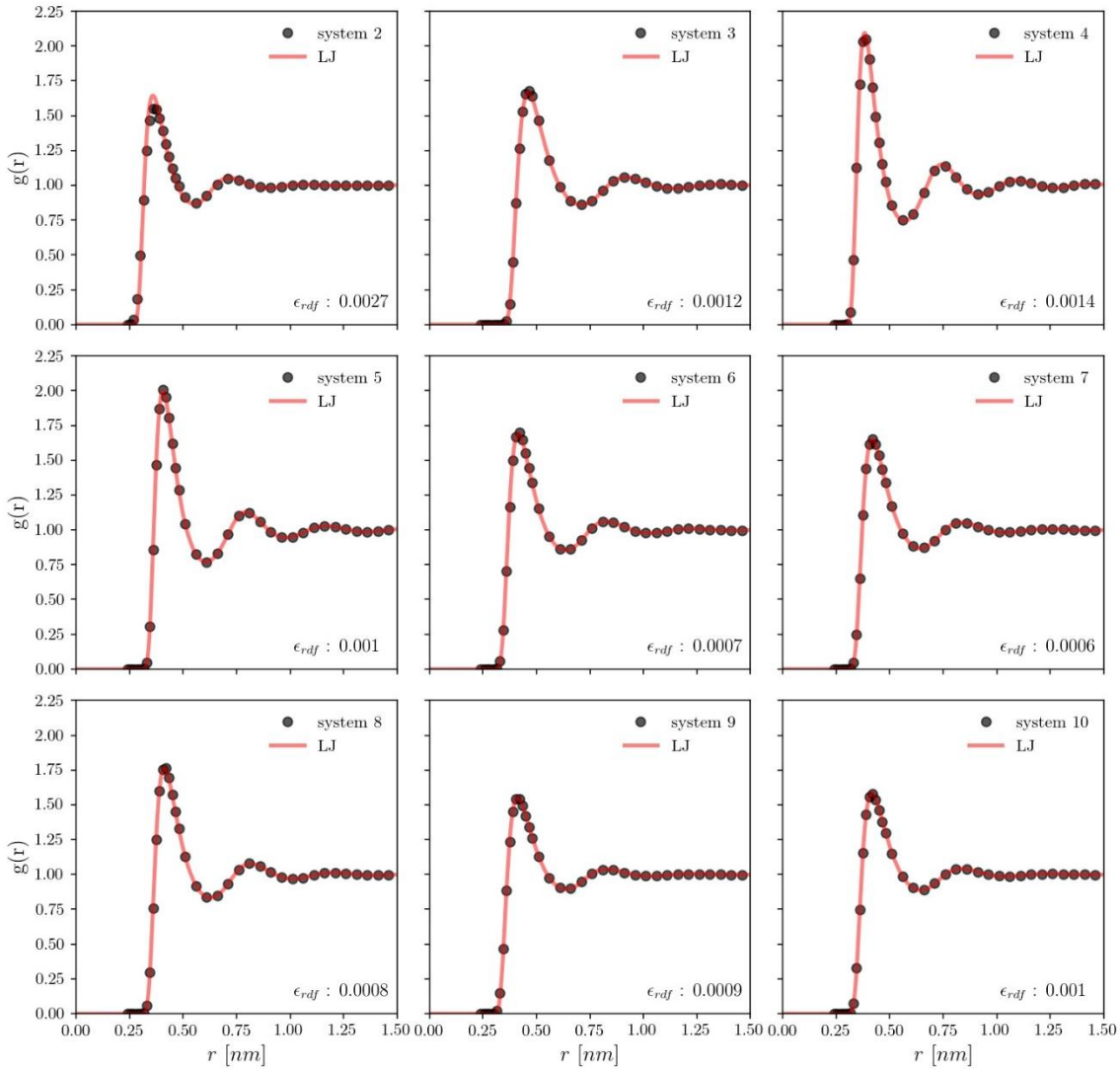


Figure C.2. Consistency between DeepILST and structural quasi-universality for exponential pair potentials. RDFs of exponential pair potentials with different potential parameters and thermodynamic states. RDFs of QU-equivalent LJ pair potential are shown with a solid red line, while black circles show RDFs with the exponential pair potential.

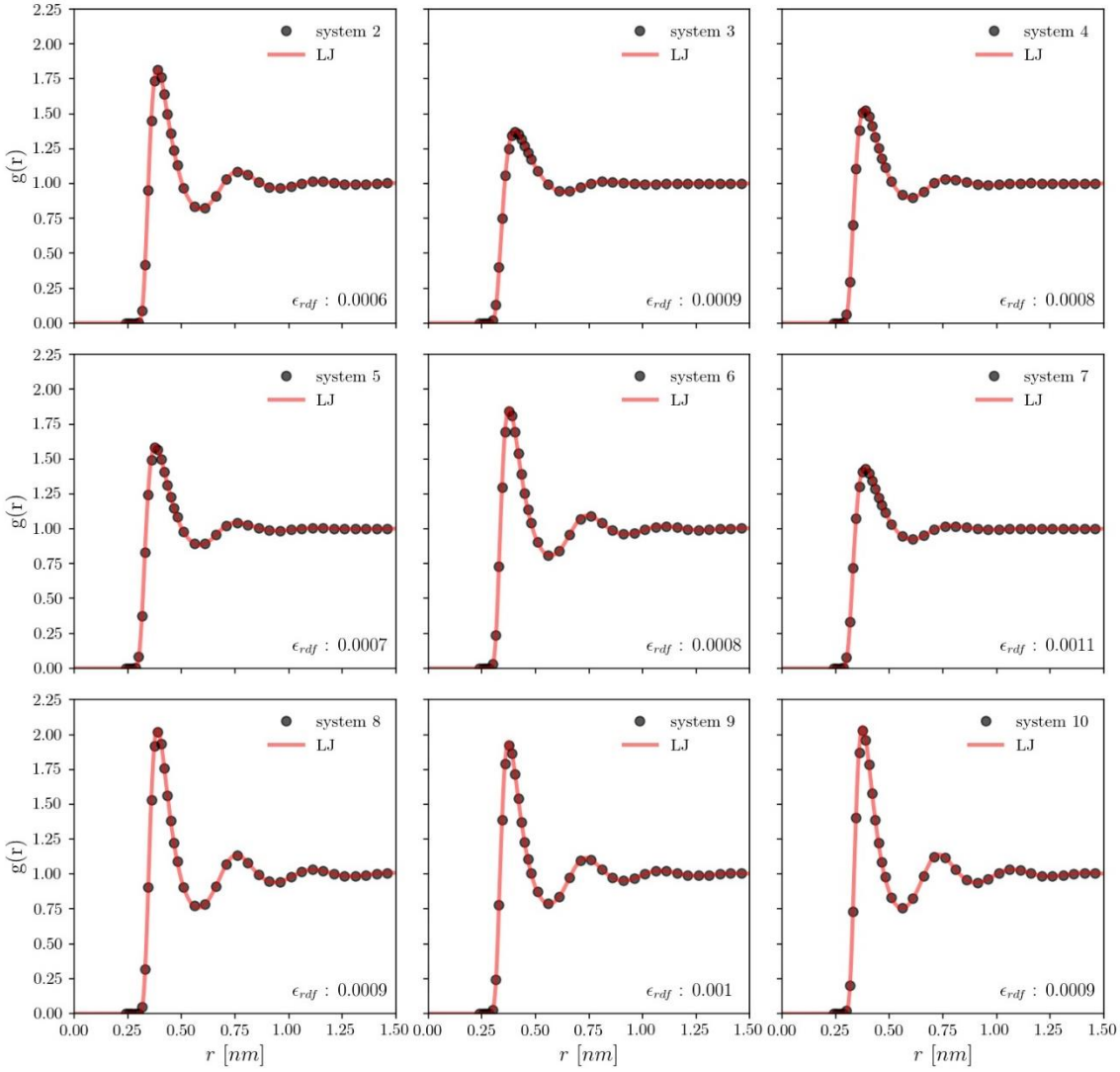


Figure C.3. Consistency between DeepILST and structural quasi-universality for Yukawa pair potentials. RDFs of Yukawa pair potentials with different potential parameters and thermodynamics states. RDFs of QU-equivalent LJ pair potential are shown with a solid red line, while black circles show RDFs with the Yukawa pair potential.

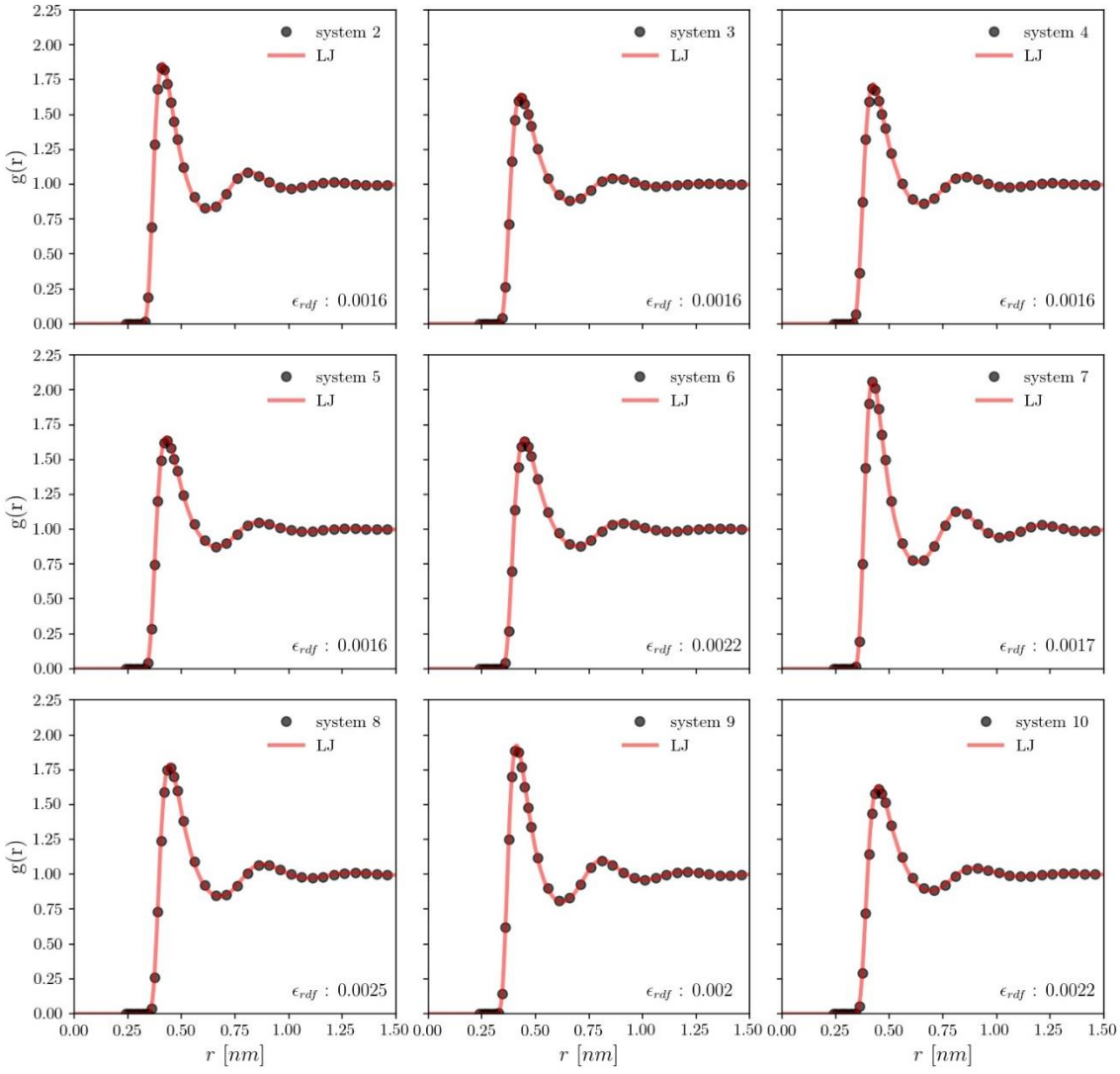


Figure C.4. Consistency between DeepILST and structural quasi-universality for inverse-power-law (14, 8) pair potential. RDFs of inverse-power-law (14, 8) pair potentials with different potential parameters and thermodynamics states. RDFs of QU-equivalent LJ pair potential are shown with a solid red line, while black circles show RDFs with inverse-power-law (14, 8) pair potential.

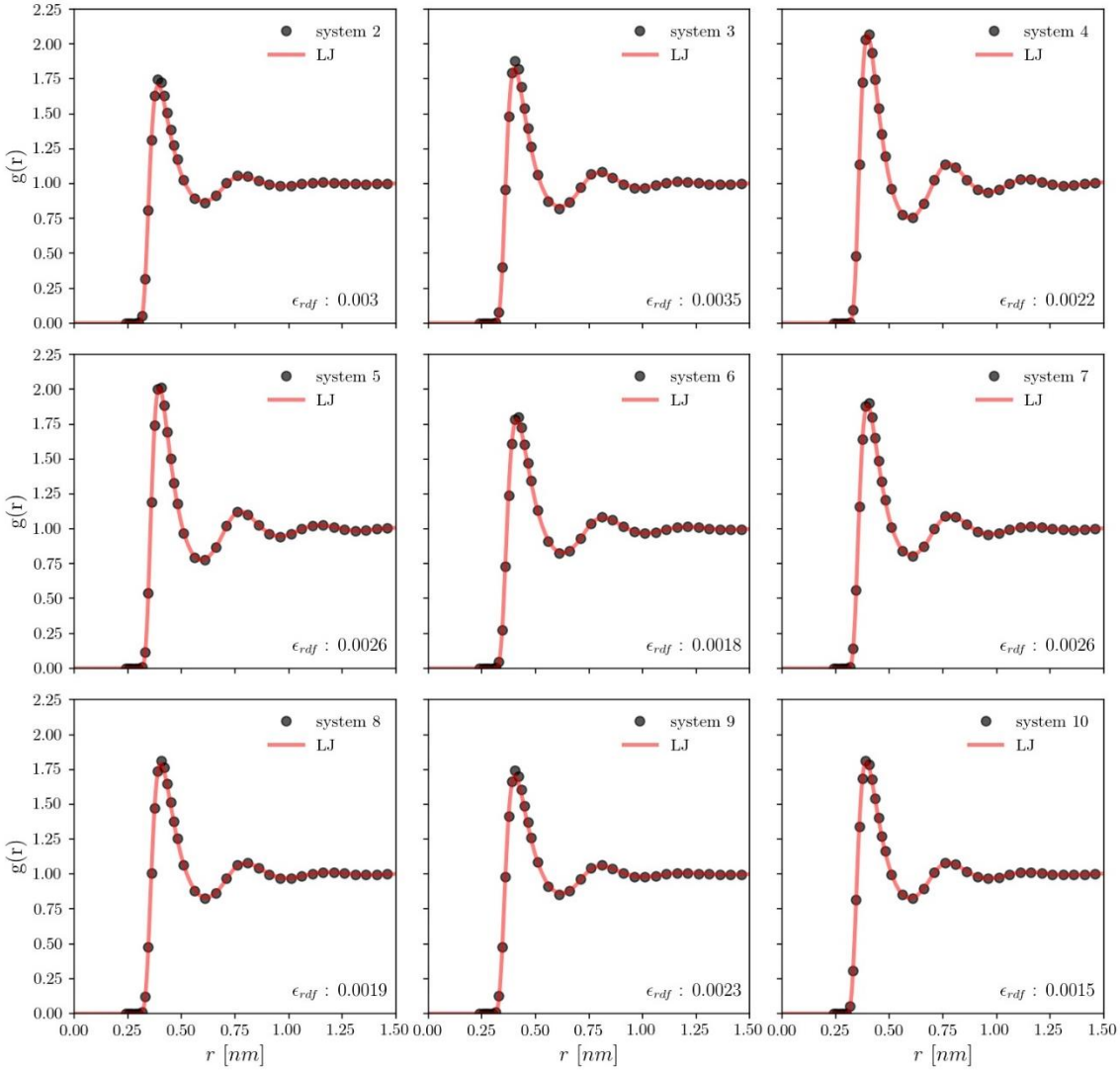


Figure C.5. Consistency between DeepILST and structural quasi-universality for inverse-power-law (10, 4) pair potentials. RDFs of inverse-power-law (10, 4) pair potentials with different potential parameters and thermodynamics states. RDFs of QU-equivalent LJ pair potential are shown with a solid red line, while black circles show RDFs with inverse-power-law (10, 4) pair potential.